NHẬP MÔN THỐNG KÊ HƯỚNG TỚI MÁY HỌC

Phạm Minh Hoàng

Diễn đàn VBA Việt Nam

Website: tuhocvba.net

Ngày 11 tháng 7 năm 2022

LỜI GIỚI THIỆU

Nói tới diễn đàn THVBA, người ta nghĩ tới những bài viết dễ hiểu, vì phương châm của chúng tôi là viết làm sao để người ngu nhất cũng hiểu được. Triết lý này là nền móng căn cơ phát triển diễn đàn cho tới nay. THVBA chuyên về lĩnh vực VBA và đã đóng góp rất nhiều bài viết về VBA được cộng đồng đánh giá rất cao. Nhiều người khi muốn tìm hiểu một lĩnh vực nào đó thường nói đùa rằng, ước gì anh em quản trị viên THVBA cũng mở thêm mảng mà họ quan tâm, họ mong chờ những bài viết với phong cách dễ hiểu.

Lập trình AI hay Khoa học dữ liệu là lĩnh vực khó mặc dù Python đã hỗ trợ chúng ta rất nhiều. Nếu chỉ vận dụng code có sẵn và làm theo thì chẳng có gì để nói, vấn đề ở đây là làm sao hiểu được bản chất để từ đó làm chủ tri thức. Kiến thức toán thống kê được sử dụng nhiều, nhưng phần lớn mọi người giảng cho nhau nghe thì chỉ nói tới khái niệm chung chung, không minh họa bằng những ví dụ cụ thể để mọi người hình dung đúng bản chất. Một số khác mặc nhiên công nhận điều người khác nói mà không hiểu rõ ràng những khái niệm thống kê này.

Theo tôi biết, sách về toán thống kê ở Việt Nam không thiếu, nhưng liên kết nó tới các vấn đề về AI, về khoa học dữ liệu, thì chưa có cuốn sách giáo khoa nào làm tốt. Trước nhu cầu cấp bách đó, tôi ấp ủ tạo nên cuốn sách này, với lối trình bày dễ hiểu, tôi sẽ làm rõ những khái niệm cơ bản về xác suất, đồng thời sử dụng Python làm công cụ kiểm chứng các kết quả tính toán thống kê. Tôi không mong muốn gì hơn đó là cuốn sách thực sự có ích cho các bạn. Và nếu trong tương lai nó trở thành cuốn sách gối đầu giường của các bạn sinh viên theo học ngành khoa học dữ liệu, hay máy học, thì đó là niềm vui đối với tôi.

Nội dung trong cuốn sách này được tôi biên dịch từ Chúc các bạn gặt hái nhiều thành công!

Admin Forum THVBA
nickname: tuhocvba
Phạm Minh Hoàng
Tốt nghiệp ĐH Bách Khoa Hà Nội khóa 2003-2008
Cựu học sinh chuyên toán Chuyên Hùng Vương Phú Thọ khóa 2000-2003

Mục lục

Phần I

Nhập Môn Thống Kê Hướng Tới Data Science

Bài 1

Thống kê mô tả và Suy luận thống kê

1.1 Phạm vi khóa học

Ở khóa học này chúng ta sẽ học các nội dung cơ bản về thống kê học.
Những kiến thức tôi trình bày trong khóa học này là những kiến thức cơ bản. Tôi coi các bạn là những người chưa bao giờ nghiên cứu thống kê!!! Tôi muốn bạn thực hiện khóa học này trước khi đọc một cuốn sách khó về thống kê.
Phạm vi của khóa học này sẽ bắt đầu đi từ thống kê mô tả, tạm thời xem nhẹ ước lượng và kiểm định. Sau đó, tôi ước mình có thể kết nối những kiến thức thống kê với các vấn đề trong khóa học Machine Learning.
Tuy nhiên, tôi nghĩ rằng sau khi trải qua khóa học này, sau đó nếu bạn đọc các sách thống kê khác, bạn sẽ cảm thấy đơn giản hơn vì bạn đã được trang bị những kiến thức thống kê cơ bản nhất mà khóa học này đã mang lại cho bạn. Những kiến thức thống kê trong khóa học này, đó sẽ là những kiến thức cần thiết để bạn tiếp cận với Machine Learning, vì vậy hãy nắm chắc những kiến thức cơ bản trong khóa học này!

1.2 Sử dụng Python trong khóa học này

Trong khóa học này tôi sẽ sử dụng Python để tiếp cận với thống kê, tuy nhiên dù cho bạn không hiểu gì về code python thì bạn có thể bỏ qua các phần chứa code python.
Do mục đích nhắm tới sau này là khoa học dữ liệu (Data Science), do đó dù thế nào đi nữa, nếu bạn có chút kiến thức cơ bản về Python thì vẫn tốt hơn đấy.
Ở khóa học này, các thư viện trong Python được sử dụng chủ yếu là `NumPy`, `Pandas`, `matplotlib`, `seaborn`, và tôi muốn giới thiệu tới hai thư viện mới là `SciPy` (`stats`) và `scikit−learn`.

SciPy là thư viện mở của python trong khoa học, đọc là sai-pai. Nó dựa trên **NumPy** và có các mô-đun rất hữu ích trong khoa học và kỹ thuật, chẳng hạn như các bài toán thống kê và tối ưu hóa, tích phân và đại số tuyến tính.
Trong khóa học này, chúng ta sẽ sử dụng module stats có trong SciPy để nghiên cứu về thống kê. **scikit-learn** là thư viện mở của python ứng dụng trong máy học. Nó có mặt ở trong hầu hết các thuật toán chính về máy học.
Vì scikit xuất phát từ SciPy Toolkit , nó là một thư viện mở rộng của SciPy, nhưng từ quan điểm của người dùng, bạn có thể coi nó như một thư viện riêng biệt từ SciPy.
scikit-learn là thư viện ứng dụng trong máy học nhưng cũng được sử dụng một chút trong thống kê. Cả SciPy và scikit-learn đều có trong Anaconda.
Trong khóa học này, chúng ta thực sự sẽ viết code Python, nhưng xin lưu ý rằng code trong khóa học này không nhất thiết phải là "tối ưu". (Bởi vì nó không phải là một khóa học “Python”).

1.3 Thống kê là gì?

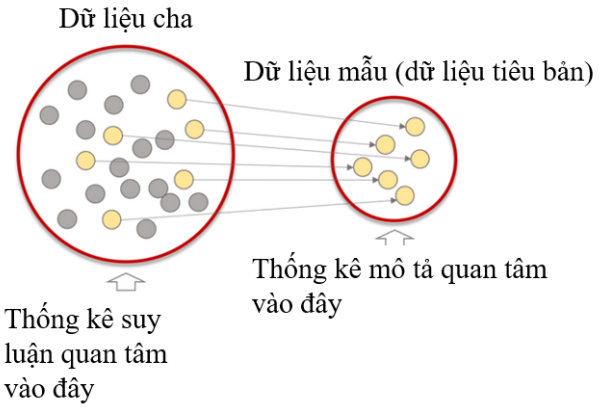
Nào, hãy bắt đầu với câu hỏi, thống kê là gì? Thống kê là bộ môn khoa học nghiên cứu các phương pháp, cách nghĩ về phân tích dữ liệu thống kê. Dữ liệu thống kê là: Khi chúng ta có một vật mà chúng ta muốn quan sát, chúng ta tập hợp các giá trị quan sát được, các giá trị đo được từ đối tượng đó.
Chẳng hạn, nếu chúng ta muốn biết thu nhập năm của một người làm khoa học dữ liệu ở Nhật Bản, thực tế, trước hết chúng ta phải tìm kiếm dữ liệu có tên như thế nào đó liên quan tới thu nhập hàng năm của người làm khoa học dữ liệu. Giá trị mà chúng ta thu thập, tập hợp lại, đó gọi là dữ liệu thống kê.
Trong khóa học này, tôi chia thống kê làm hai nhóm lớn:

- Thống kê mô tả (descriptive statistics).
- Thống kê luận lý (thống kê suy luận) (inferential statistics).

Nghe qua những từ ngữ trên bạn đừng nghĩ là khó nhé, nó không khó đâu.
Tầm quan trọng của việc trực quan hóa dữ liệu bằng cách sử dụng các biểu đồ phân tán cũng đã được thực hiện trong khóa học Python.
Tuy nhiên, thực tế nếu chỉ có biểu đồ phân tán, ta có thể hiểu khuynh hướng của dữ liệu,

nhưng nếu chỉ có thế thì nó vẫn còn đó những giới hạn, do đó tôi muốn sử dụng kiến thức số học cơ bản để nói dữ liệu này là ⋯
Ví dụ, khi nhìn vào thu nhập năm của những người bạn làm về khoa học dữ liệu tại Nhật Bản, có cảm nhận rằng thu nhập này cao hơn người bình thường, nhưng cao hơn bao nhiêu, thì cần có các chỉ số dễ hiểu.
Chẳng hạn nhìn vào mức thu nhập bình quân (trung bình) cũng được, hoặc có thể sử dụng giá trị trung vị.
Nếu sử dụng các chỉ số như thế, ta có thể có những hình dung đúng hơn về dữ liệu thực tế.
À, bạn có thể nghĩ ồ việc như thế cũng có thể làm được hay sao, vậy thì phải cố học thống kê thôi!

Từ những dữ liệu có ý nghĩa mà bạn thu thập trong thực tế, gọi là dữ liệu mẹ(cha) . Tiếng anh là population, từ dữ liệu quan sát thực tế này ta lấy ra một phần dữ liệu để nghiên cứu, phần dữ liệu này gọi là dữ liệu mẫu, hay dữ liệu tiêu bản, tiếng anh là sample. Suy luận thống kê là sử dụng dữ liệu mẫu để từ đó suy luận ra đặc tính của dữ liệu cha.



Suy luận thống kê có hai kiểu là Ước lượng và Kiểm định. Ước lượng là suy ra tỷ lệ trong dữ liệu cha hoặc giá trị trung bình nằm trong một khoảng nào đó nhờ quan sát dữ liệu mẫu, là dữ liệu trích xuất từ dữ liệu cha. (Ví dụ: "Thu nhập trung bình hàng năm của các nhà khoa học dữ liệu Nhật Bản chắc chắn là XX!" Hoặc "Tỷ lệ nam-nữ của các nhà khoa học dữ liệu Nhật Bản chắc chắn là XX!")

Kiểm định là đưa ra câu trả lời Có / Không cho bất kỳ "câu hỏi" nào dựa trên kết quả khảo sát của mẫu. (Ví dụ: trả lời Có / Không cho "Thu nhập hàng năm của các nhà khoa học dữ liệu Nhật Bản có tăng so với năm ngoái không?")

Nói chung, khi mà nói tới "thống kê", hầu hết mọi người nghĩ về "thống kê suy luận". Trên thực tế, "thống kê mô tả" là phần kiến thức không thể thiếu khi thực hiện ước lượng hay kiểm định.

Ồ từ từ đã tôi đã nhớ hết đâu!!! Bạn không cần phải biết, không hiểu cũng được! Bạn

không cần phải nhớ bất kỳ từ nào! Hãy làm quen dần dần nhé!
Sau đó, từ lần sau, tôi sẽ giải thích "giá trị đại diện" đầu tiên của thống kê mô tả!

Bài 2

Giá trị đại diện

Ở bài học trước chúng ta đã chia ra gồm có thống kê mô tả và thống kê suy luận. Thống kê mô tả là lĩnh vực phân tích dữ liệu thống kê từ những dữ liệu quan sát được. Tuy nhiên để hiểu được kết quả phân tích dữ liệu, ta cần có chỉ số thể hiện đặc tính của dữ liệu. Ở bài học này sẽ giới thiệu các chỉ số nói lên đặc tính của dữ liệu.
Cũng từ đây, chúng ta sẽ sử dụng Python để tính toán chỉ số của dữ liệu.

2.1 Giá trị đại diện của dữ liệu

Nó có một cái tên khá khó hiểu, nhưng mấu chốt là giá trị được sử dụng để giải thích ý nghĩa của dữ liệu.
Ví dụ, giả sử bạn mua 5 quả táo ở siêu thị. Khi được hỏi một quả táo nặng bao nhiêu, bạn thường trả lời là trọng lượng trung bình của năm quả táo phải không? "Giá trị trung bình" được sử dụng ở đây là giá trị đại diện thể hiện trọng lượng của quả táo đã mua. Nó chỉ được đại diện bởi một con số và được sử dụng để đánh giá tài sản tổng thể.
Bạn vô tình sử dụng "giá trị đại diện" này trong cuộc sống hàng ngày của mình để xác định xem tổng thể những quả táo bạn mua hôm nay có to không, đúng không? Lần này, chúng ta sẽ xem xét kỹ hơn "giá trị trung bình" này.

2.2 Trung bình số học thường được biết đến là gì (trung bình cộng)

Ồ, nói như vậy chắc là có lắm kiểu giá trị trung bình lắm đúng không? Đúng vậy. Và giá trị trung bình cộng (arithmetic mean) chỉ là một trong số nhiều giá trị trung bình. Ví dụ: Tôi có 5 quả táo có trọng lượng như sau $295g$, $300g$, $300g$, $310g$, $311g$. Chúng có khối lượng trung bình là:

$$\frac{(295g + 300g + 300g + 310g + 311g)}{5} = 303.2g$$

Trong python, ta có thể dùng thư viện numpy để tính toán giá trị trung bình cộng như sau:

```
import numpy as np
apple_weights = [295, 300, 300, 310, 311]
np.mean(apple_weights)
```

Kết quả ta được:

```
303.2
```

Giá trị trung bình cộng của một dãy số là tổng các số của dãy số chia cho số số hạng. Nếu bạn viết điều này trong một công thức toán học, nó sẽ như sau:

$$\bar{x} = \frac{x_1 + x_2 + \cdots + x_n}{n}$$

2.3 Sử dụng tỷ suất (trung bình hình học)

Chúng ta hãy xem xét ví dụ sau:
Người ta thống kê được rằng nhân viên một công ty từ khi gia nhập vào một công ty có độ thăng tiến về thu nhập như sau:

-Sau một năm làm việc thì lương tăng so với năm trước là 5%.

-Sau hai năm làm việc thì lương tăng so với năm trước là 10%.

-Sau ba năm làm việc thì lương tăng so với năm trước là 30%.

Giả sử khi mới gia nhập một công ty, lương anh nhân viên là 500 đô-la. Vậy bây giờ (sau 3 năm) thì lương của anh ấy là bao nhiêu?

$$500\,(1 + 0.05)\,(1 + 0.1)\,(1 + 0.3) = 750.75$$

Như vậy mức tăng trung bình hàng năm là bao nhiêu % là câu hỏi chúng ta sẽ nghĩ tới. Nếu tính bằng công thức tính giá trị trung bình 5%, 10%, 30% thì chẳng phải mức tăng lương bình quân hàng năm là: $\frac{5\% + 10\% + 30\%}{3} = 15\%$ hay sao?
Nào, bây giờ tôi giả định tỷ suất tăng lương hàng năm là g. Khi đó ta có:

$$500\,(1 + g)\,(1 + g)\,(1 + g) = 500\,(1 + g)^3 = 750.75$$

Nếu ta coi $m_g = (1 + g)$, ta có: $x_1 = (1 + 0.05)$,$x_2 = (1 + 0.01)$,$x_3 = (1 + 0.3)$.
Ta có công thức: ${m_g}^3 = x_1 x_2 x_3 \Rightarrow m_g = \sqrt[3]{x_1 x_2 x_3}$
Giá trị này gọi là trung bình hình học (geometric mean). Nó được sử dụng khi tính tỷ suất trung bình.
Như vậy nếu là trung bình cộng thì ta có công thức: $\frac{x_1 + x_2 + x_3}{3}$
Trong trường hợp tính tỷ suất trung bình thì là: $\sqrt[3]{x_1 x_2 x_3}$
Tổng quát ta có:

$$m_g = \sqrt[n]{x_1 x_2 x_3 \cdots x_n}$$

Ta hãy thử tính giá trị trung bình hình học bằng Python. Ở đây tôi sẽ sử dụng module stats trong SciPy.
Các bạn có thể sử dụng `scipy.stats.gmean()` để tính.

```
from scipy import stats
salary_growth = [1.05, 1.1, 1.3]
salary_growth_mean = stats.gmean(salary_growth)
print(salary_growth_mean)
```

Kết quả:

```
1.1450956868476592
```

Như vậy từ nay ai đó hỏi bạn tỷ suất tăng lương trung bình hàng năm, các bạn hãy sử dụng công thức tính giá trị trung bình hình học.

2.4 Trung bình điều hòa

Ở phần này tôi xin giới thiệu một khái niệm trung bình nữa, đó là trung bình điều hòa, có tên tiếng anh là harmonic mean.
Nó còn có tên gọi khác là nghịch đảo của trung bình nghịch đảo. Nghe tên thì có cảm giác khó hiểu nên sau đây tôi đi vào ví dụ.
Chẳng hạn một người đi từ A tới B với vận tốc là $x_1$km/h.Ở chiều về anh ta đi từ B tới A với vận tốc là $x_2$km/h.
Như vậy vận tốc trung bình là bao nhiêu? Nhiều người sẽ nghĩ rằng đó chẳng phải là $\frac{x_1 + x_2}{2}$km/h hay sao? Tuy nhiên chúng ta hãy xem xét vấn đề dưới góc độ thời gian như sau:
Thời gian anh ta đi quãng đường $2.AB$ là bao lâu nếu coi d (km) là khoảng cách AB, ta có:$\frac{d}{x_1} + \frac{d}{x_2}$ (đơn vị thời gian).
Như vậy vận tốc trung bình mà anh ta đã đi là quãng đường chia cho thời gian:

$$\frac{2d}{\frac{d}{x_1} + \frac{d}{x_2}} = \frac{1}{\frac{1}{2}\left(\frac{1}{x_1} + \frac{1}{x_2}\right)}\text{km/h}$$

Nhìn vào công thức này ta thấy nó có hình thù là nghịch đảo của trung bình cộng của nghịch đảo.
Chẳng hạn chiều đi anh ta đi với vận tốc 20km/h và chiều về đi với vận tốc 60km/h, như thế nếu tính công thức trung bình cộng sẽ là $(20 + 60)/2 = 40$km/h, tuy nhiên không phải vậy, nếu tính bằng công thức nghịch đảo như đã nói ở trên, nó sẽ là nghịch đảo của $(1/20 + 1/60)/2 = 1/30$, tức là vận tốc trung bình là 30km/h.
Một cách tổng quát, nếu ta có n số hạng $x_1, x_2 . \cdots, x_n$, khi đó trung bình điều hòa là:

$$m_h = \frac{1}{\frac{1}{n}\left(\frac{1}{x_1} + \frac{1}{x_2} + \cdots + \frac{1}{x_n}\right)}$$

Trong Python ta có cách tính trung bình điều hòa bằng cách sử dụng scipy.stats.hmean() như sau:

```
velocities = [20, 60]
velocities_mean = stats.hmean(velocities)
print(velocities_mean)
```

Kết quả:

```
30.0
```

Thành thật mà nói, tôi không nghĩ có nhiều trường hợp nó được sử dụng như giá trị đại diện, nhưng nó xuất hiện vài lần khi học về lý thuyết máy học, do đó tôi đã giới thiệu cho các bạn công thức này.

2.5 Tính chất quan trọng của giá trị trung bình cộng

Đối với giá trị trung bình cộng, nó có rất nhiều tính chất nhưng ở đây tôi giới thiệu hai tính chất quan trọng.

Đầu tiên phải nói tới tính chất, **tổng độ lệch (sai khác) từ các điểm dữ liệu tới giá trị trung bình cộng là** 0.

Ví dụ với 5 quả táo lần lượt có trọng lượng là 295g, 300g, 300g, 310g, 311g có giá trị trung bình là 303.2g. Lần lượt lấy trọng lượng từng quả táo trừ cho giá trị trung bình cộng rồi cộng các kết quả này với nhau ta sẽ được tổng là 0.

```
apple_weights = np.array([295, 300, 300, 310, 311])
apple_w_mean = np.mean(apple_weights)
deviations = apple_weights - apple_w_mean
print(deviations)
print(deviations.sum())
```

Kết quả:

```
[-8.2 -3.2 -3.2  6.8  7.8]
5.684341886080802e-14
```

$e-14$ có nghĩa là 10^{-14},nó có giá trị xấp xỉ là 0.

Tóm lại, giá trị trung bình có nghĩa là nó ở vị trí chính giữa trong toàn bộ dữ liệu, do đó độ lệch từ các điểm dữ liệu tới nó sẽ có giá trị dương và âm, chúng triệt tiêu lẫn nhau và dẫn tới tổng độ sai khác này có kết quả là 0 .

Độ sai khác tới giá trị trung bình được gọi là Thiên Sai (deviation). Thiên trong từ Thiên Vị-Thiên Kiến ý nói thiên lệch về một phía, Sai trong từ Sai Khác. Tôi nghĩ một từ thuần việt là **độ lệch** có lẽ dễ hiểu hơn.

Một tính chất quan trọng khác đó là **tổng các bình phương của độ lệch từ các điểm dữ liệu tới giá trị trung bình là giá trị nhỏ nhất**. Từ một giá trị X bất kỳ, ta có tổng các bình phương của độ sai khác từ các dữ liệu tới giá trị X là $S(X) = \sum_{i=1}^{n}(x_i - X)^2$.

Vậy X nhận giá trị là bao nhiêu thì tổng trên nhận giá trị nhỏ nhất? Trước hết ta hãy tính đạo hàm của biểu thức trên và tìm hiểu xem nó nhận giá trị 0 khi nào:

$$\frac{dS(X)}{dX} = -2\sum_{i=1}^{n}(x_i - X) = 0$$

Tiếp tục biến đổi:

$$0 = \sum_{i=1}^{n} (x_i - X) = (x_1 + x_2 + \cdots + x_n) - nX = n\bar{x} - nX$$

Biến đổi tới đây tôi nghĩ nhiều người đã hiểu. Tuy nhiên nếu tới đây mà vẫn có người không hiểu thì cũng không sao đâu. Tôi nghĩ bạn có thể công nhận điều này mà không cần phải chứng minh.

Những tính chất trên có vẻ như là hiển nhiên nhưng chúng lại rất quan trọng khi sử dụng nó để giải thích lý thuyết thống kê trong tương lai.

2.6 Tổng kết

Trong bài học này, để giải thích dữ liệu bằng các chỉ số dễ hiểu, tôi đã giới thiệu các giá trị đại diện được sử dụng để giải thích dữ liệu, đặc biệt là giá trị trung bình được sử dụng nhiều nhất.
Thông thường, giá trị trung bình được đề cập thường được hiểu đó là giá trị trung bình cộng trừ khi có ghi chú khác. Trong khuôn khổ khóa học này cũng vậy, thuật ngữ "trung bình" đề cập đến đó là trung bình số học-hay còn gọi là trung bình cộng.

- Giá trị đại diện là giá trị đại diện cho dữ liệu. Chỉ một giá trị được sử dụng để đánh giá bản chất của toàn bộ dữ liệu.
- Trung bình là một trong những chỉ số thường được sử dụng làm giá trị đại diện.
- Chú ý rằng giá trị trung bình hình học được sử dụng trong trường hợp giá trị dữ liệu là tỷ suất.
- Để tính giá trị trung bình hình học ta dùng `scipy.stats.gmean()`, và để tính giá trị trung bình điều hòa ta sử dụng `scipy.stats.hmean()` .
- Tổng độ lệch từ các điểm dữ liệu tới giá trị trung bình cộng là 0.
- Giá trị trung bình cộng là giá trị mà tại đó tổng các bình phương của độ lệch từ các điểm dữ liệu tới nó là nhỏ nhất.

Chỉ toàn chữ và công thức, quả thật là khó nhớ, vì vậy hãy cố gắng học nó. Lần tới, tôi sẽ giới thiệu những giá trị đại diện khác.

Bài 3

Giá trị đại diện khác

Tôi đã đề cập đến chỉ số quan trọng nhất, "trung bình", trong "giá trị đại diện", là "giá trị giải thích các đặc điểm của toàn bộ dữ liệu", nhưng vì có các giá trị khác có thể được sử dụng làm giá trị đại diện, tôi sẽ giới thiệu ngắn gọn về chúng.

3.1 Giá trị trung vị, giá trị giữa

Đây là giá trị xuất hiện nhiều lần khi các bạn học Data Science-Khoa học dữ liệu (Trong NumPy hay boxplot của Seaborn). Tôi không nghĩ mình cần phải giải thích về giá trị trung vị **(median)**, hay còn gọi là giá trị giữa, vì nó là một chỉ số thường được dùng thường xuyên. Khi chúng ta sắp xếp dữ liệu từ bé đến lớn, giá trị này nằm ở chính giữa. (Nếu có một số lượng dữ liệu chẵn thì người ta thường lấy trung bình cộng của cả hai số ở giữa làm giá trị trung vị).

```
import numpy as np
arr = np.array([7, 2, 4])
x = np.median(arr)
y = np.mean(arr)
print(f"Gia tri trung vi la {x}")
print(f"Gia tri trung binh la {y}")
```

Kết quả:

```
Gia tri trung vi la 4.0
Gia tri trung binh la 4.333333333333333
```

Để làm rõ trong trường hợp dữ liệu có số chẵn phần tử:

```
import numpy as np
arr = np.array([7, 2, 4,6])
x = np.median(arr)
print(f"Gia tri trung vi la {x}")
```

Kết quả:

```
Gia tri trung vi la 5.0
```

Như trong ví dụ ở phần trước khi đề cập tới các trái táo, nếu các giá trị dữ liệu gần giống nhau, bạn có thể sử dụng giá trị trung bình làm giá trị đại diện, tuy nhiên có những giá trị ngoại lệ (giá trị lệch khỏi phân phối tổng thể)**(outlier)**, nếu sử dụng giá trị trung

bình làm giá trị đại diện, việc hiểu sai về bản chất dữ liệu có thể xảy ra. Ví dụ khi nói tới thu nhập hàng năm hoặc về tài sản, tiền tệ, có lẽ giá trị trung vị sẽ tốt hơn là giá trị trung bình khi ta phân tích các dữ liệu như vậy. Dù có thể có những giá trị nằm ra ngoài phân phối tổng thể thì giá trị trung vị hầu như không bị ảnh hưởng trong khi đó nếu như có một (hoặc nhiều) giá trị đột biến lớn hay đột biến nhỏ nằm ngoài phân phối tổng thể của dữ liệu sẽ có tác động rất lớn tới giá trị trung bình cộng. Ngoài ra cần lưu ý thêm, lượng phép tính khi tính toán trung vị sẽ tốn nhiều hơn so với tính toán giá trị trung bình cộng. Giá trị trung bình không quá tốn kém khi nói tới chi phí tính toán, ta có thể cộng lần lượt từng giá trị, sau cùng chia cho tổng số các điểm dữ liệu sẽ có được giá trị trung bình. Tuy nhiên đối với giá trị trung vị, đầu tiên chúng ta mất công xử lý sắp xếp dữ liệu từ bé tới lớn, sau đó lấy giá trị ở giữa. Với lượng dữ liệu lớn, bạn sẽ cảm nhận điều này rõ hơn hết. Sau đây ta sẽ sử dụng NumPy để tính toán giá trị trung bình cộng và giá trị trung vị, cũng như so sánh thời gian tính toán bằng `time.time()` .

```
import numpy as np
import time
# Xuat ngau nhien 1 trieu so tu phan bo tieu chuan
randoms = np.random.randn(10**7)
# Thoi gia truoc khi tinh toan(sec)
before_mean = time.time()
# Tinh gia tri trung binh cong
mean = np.mean(randoms)
# Thoi gian sau khi tinh toan gia tri trung binh cong(sec)
after_mean = time.time()
print('mean is {} (time:{:.2f}s)'.format(mean, after_mean-before_mean))
# Thoi gian truoc khi tinh toan(sec)
before_median = time.time()
# Tinh toan gia tri trung vi
median = np.median(randoms)
# Thoi gian sau khi tinh toan gia tri trung vi
after_median = time.time()
print('median is {} (time:{:.2f}s)'.format(median, after_median-before_median))
```

Kết quả:

```
mean is 0.00015996733320892628 (time:0.01s)
median is -4.2001016778634444e-05 (time:0.14s)
```

`time.time()` sẽ trả về đại lượng thời gian có đơn vị là giây khi nó được thực thi. Hãy nhớ lấy nó khi bạn cần tính toán thời gian thực thi một xử lý nào đó trong Python. Ngoài ra `{:.2f}` có nghĩa là sẽ hiển thị giá trị số thập phân tới hai chữ số sau dấu phẩy. Khi tính toán thời gian và sau đó hiển thị thời gian, nó trở nên tiện lợi, hãy nhớ lấy nó. Từ phân phối tiêu chuẩn lấy ngẫu nhiên các giá trị thì tôi nghĩ giá trị trung bình có lẽ sẽ là số xấp xỉ 0. Và giá trị trung vị cũng xấp xỉ là 0. (Phân phối tiêu chuẩn là phân phối chuẩn với giá trị trung bình là 0 và phương sai-mức độ phân tán là 1).

3.2 Giá trị xuất hiện nhiều lần nhất: Tối Tần Trị

Với tư cách là một giá trị đại diện, không phải là giá trị trung bình, cũng không phải là giá trị trung vị, trong phần này tôi giới thiệu một giá trị khác, đó là Tối Tần Trị **(mode)**. Tối tần trị có nghĩa là giá trị được quan sát thấy xuất hiện nhiều lần nhất trong bảng

phân bố dữ liệu. Nếu ta sắp xếp dữ liệu theo số lần xuất hiện của chúng, có thể thấy tối tần trị giống như là một đỉnh núi cao nhất. Trong xác suất thống kê, có thể nói đây là giá trị có mật độ xuất hiện nhiều lần nhất.

Ví dụ nếu dữ liệu thống kê cực kỳ tập trung vào một giá trị nhất định, khi đó có thể sử dụng giá trị tối tần trị như là một giá trị đại diện thì thích hợp hơn giá trị trung bình hoặc trung vị.

Ta có thể sử dụng `scipy.stats.mode()` để tính tối tần trị.

```
from scipy import stats
# Tra ve Toi Tan Tri (mode) va so lan xuat hien cua no
mode, count = stats.mode([6, 2, 4, 5, 1, 3, 5, 3, 4])
# Neu tim thay hai gia tri Toi Tan Tri, no se tra ve gia tri nho hon. Trong vi du nay so
    3 xuat hien 2 lan, so 4 xuat hien 2 lan.
# Vi vay gia tri tra ve la 3, vi 3 < 4
print(mode)
print(count)
```

Kết quả:

```
[3]
[2]
```

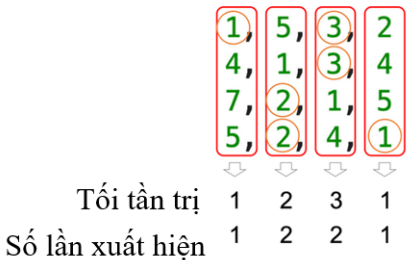
Nếu đầu vào là một mảng **NumPy Array** hai chiều. Khi đó mode sẽ được tính như thế nào?

```
array = np.array([[1, 5, 3, 2],
[4, 1, 3, 4],
[7, 2, 1, 5],
[5, 2, 4, 1]])
mode, count = stats.mode(array, axis=0)
print(mode)
print(count)
```

Kết quả:

```
[[1 2 3 1]]
[[1 2 2 1]]
```

Ở trên ta chỉ định `axis=0` có nghĩa là nó sẽ duyệt từng cột dữ liệu trong ma trận. Trong mỗi cột dữ liệu sẽ tồn tại một giá trị mode. Và ta có bốn cột dữ liệu, cho nên sẽ có bốn giá trị mode. Tương ứng với mỗi mode lại có số lần xuất hiện của giá trị mode đó trong cột dữ liệu chứa nó.



Chế độ mặc định là `axis=0`, tuy nhiên bạn có thể chỉ định `axis=1`, khi đó nó sẽ duyệt theo hàng. Bạn hãy thử thực hành nhé.

3.3 Tổng kết

Ngoài giá trị trung bình, tôi đã giới thiệu thêm hai giá trị đại diện mà các bạn có thể sử dụng đó là trung vị và tối tần trị. Tối Tần Trị: Tối có nghĩa là tối đa, cực hạn. Tần trong từ Tần Suất. Trị trong từ Giá Trị. Có nghĩa là giá trị xuất hiện nhiều lần nhất.

- Giá trị trung vị là giá trị nằm ở trung tâm trong dữ liệu sau khi đã được sắp xếp tuần tự từ nhỏ đến lớn.
- Giá trị trung vị ít bị ảnh hưởng bởi giá trị nhiễu nằm xa vùng phân bố của dữ liệu. Đây là điểm nổi trội so với giá trị trung bình.
- So với giá trị trung bình thì để tính toán giá trị trung vị cần phải sắp xếp lại dữ liệu, vì thế mà phép tính tìm trung vị sẽ mất nhiều thời gian xử lý hơn.
- Để tính toán thời gian xử lý trong `Python` có thể sử dụng `time.time()`
- Tối tần trị là giá trị xuất hiện nhiều lần nhất trong dữ liệu.

Như vậy câu chuyện của chúng ta đã kết thúc, không có gì khó hiểu đúng không mọi người. Tất nhiên các giá trị này sẽ được sử dụng khi giải thích các phân phối khác nhau trong thống kê trong tương lai, vì vậy hãy ghi nhớ chúng.

Bài 4

Mức độ phân tán (Sử dụng phạm vi và vị trí phần tư)

Nếu sử dụng giá trị đại diện thì giá trị như thế nào sẽ mang tính đặc trưng của toàn bộ dữ liệu? Đây hẳn nhiên là điều mà chúng ta muốn biết. Tuy nhiên nếu chỉ nói về giá trị đại diện thì cũng có trường hợp chúng ta không thể biết hết các tính chất của toàn bộ dữ liệu. Quay trở lại bài toán về trọng lượng các quả táo. Do trọng lượng của chúng có độ chênh lệch không nhiều, do đó chỉ với giá trị trung bình cộng đã truyền đạt đầy đủ thông tin về trọng lượng của những trái táo trong vụ thu hoạch này. Thế nhưng giả sử chúng ta có năm con chó với những trọng lượng khác nhau. Nếu chỉ đánh giá qua trọng lượng trung bình thì thật khó phải không nào. Trong trường hợp có những con chó to, con chó nhỏ, thì chỉ với trọng lượng trung bình chúng ta không thể đánh giá được hết tính chất của toàn bộ dữ liệu. Những trường hợp như thế, khi mà giá trị của các dữ liệu có sự chênh lệch nhiều, chúng ta cần một chỉ số mà qua đó có thể đánh giá được độ phân bố của dữ liệu, cùng với các giá trị đại diện qua đó đánh giá đúng đắn các tính chất của toàn bộ dữ liệu. Và trong phần này chúng ta sẽ nói tới độ phân tán để biểu thị mức độ phân tán của dữ liệu.

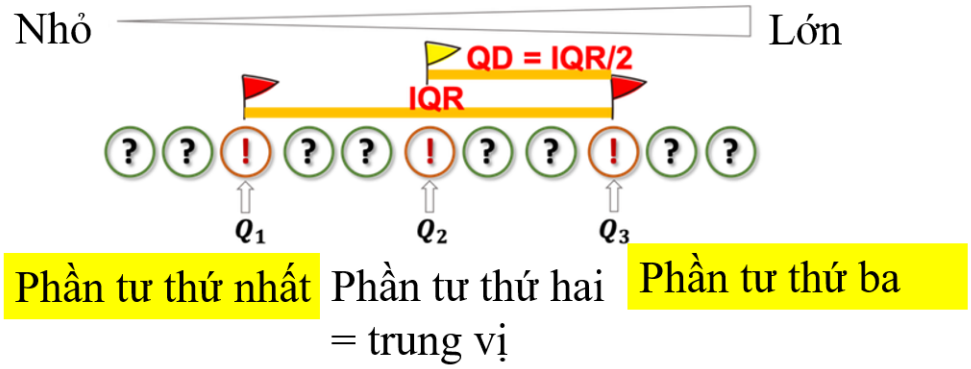
4.1 Phạm vi của giá trị

Trong các con chó mà bạn nuôi, chúng có trọng lượng lần lượt là 10kg, 13kg, 17kg, 20kg, 29kg. Khi đó trọng lượng trung bình là 17.8kg nhưng nếu chỉ dựa vào chỉ số này mà nói các con chó có trọng lượng khoảng 17.8kg thì chưa thỏa đáng, vì từ con chó nhỏ nhất là 10kg tới con chó lớn nhất là 29kg là một khoảng cách khá lớn, mức độ chênh lệch giữa con nhỏ nhất và lớn nhất là 19kg, do đó nếu chỉ nói về trọng lượng trung bình thì thật khó hình dung thực tế ra sao.
Thay vì trả lời con chó tôi nuôi có trọng lượng trung bình là 17.8kg, trả lời một cách đầy đủ rằng: Con chó tôi nuôi có trọng lượng trung bình là 17.8kg và trong phạm vi từ 10kg đến 29kg, khi đó người nghe sẽ hình dung ra được mức độ phân tán của dữ liệu. Độ phân tán của dữ liệu có thể biểu thị bằng Giá Trị Lớn Nhất - Giá Trị Nhỏ Nhất. Giá trị này được gọi là phạm vi (range).

Giá trị trung bình là 17.8kg và có phạm vi là 19kg, lúc này người nghe có thể hình dung được thực tế ra sao. Tuy nhiên nếu như chỉ thế mà không suy xét các tình huống có những giá trị gây nhiễu, tức các trường hợp ngoại lệ, các thông tin không có ý nghĩa bị lẫn vào trong các dữ liệu của chúng ta, thì chúng ta sẽ không giải quyết triệt để từ các vấn đề có trong thực tế.
Do đó tiếp theo tôi sẽ giới thiệu một chỉ số thường được sử dụng, đó là vị trí phần tư.

4.2 Phạm vi sử dụng phần tư và độ lệch phần tư

Các giá trị cực trị như giá trị tối thiểu và tối đa dễ bị ảnh hưởng bởi các giá trị ngoại lệ, do đó, chiến lược là sử dụng các giá trị gần giá trị trung bình hơn một chút để xác định mức độ phân tán.
Dữ liệu sau khi được sắp xếp theo chiều tăng dần, ta chia chúng thành bốn phần bằng nhau dựa vào ba cột mốc Q_1, Q_2, Q_3. Chúng được gọi là vị trí phần tư (quartile).
Giá trị tại vị trí chia dữ liệu thành bốn phần bằng nhau khi dữ liệu được sắp xếp (đây được gọi là phần tư) được sử dụng để thu được độ lệch phần tư (thiên sai phần tư).

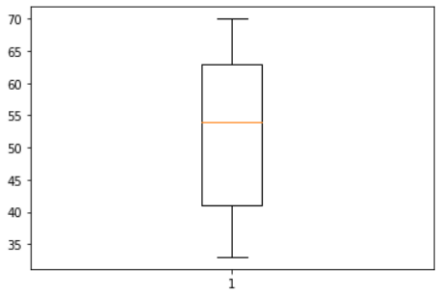


Phương pháp chia bốn phần bằng nhau rất đơn giản. Đầu tiên tìm tới giá trị trung vị, nó chia dữ liệu làm hai nửa trước và nửa sau.
Với mỗi nửa này ta lại tìm trung vị của chúng, vậy là xong. Nếu dữ liệu có chẵn số hạng thì thông thường sẽ lấy giá trị trung bình cộng hai số ở giữa.
Mỗi một phần tư sẽ có các điểm Q_1, Q_2, Q_3 gọi là phần tư thứ nhất, phần tư thứ hai, phần tư thứ ba. Chúng ta lưu ý phần tư thứ hai chính là giá trị trung vị.
Vậy thì $Q_3 - Q_1$ sẽ được coi là phạm vi phần tư (interquartile range: IQR), một nửa giá trị này là $\frac{Q_3 - Q_1}{2}$ được gọi là thiên sai phần tư hay độ lệch phần tư (QD: quartile deviation). Nhìn vào biểu đồ trên có thể thấy QD bằng $Q_3 - Q_2$, hãy cẩn thận vì điều đó không chính xác.
IQR, QD là một chỉ số được sử dụng để biểu thị độ phân tán của dữ liệu. Nó không chịu ảnh hưởng như phạm vi (range) cho dù có giá trị gây nhiễu-hay còn gọi là giá trị ngoại lệ xuất hiện trong dữ liệu của chúng ta.
Ta có thể sử dụng `scipy.stats.iqr()` để tính IQR.

```
from scipy import stats
import matplotlib.pyplot as plt
%matplotlib inline
data = [33, 35, 36, 39, 43, 49, 51, 54, 54, 56, 62, 64, 64, 69, 70]
iqr = stats.iqr(data)
qd = iqr/2
print('IQR: {}'.format(iqr))
print('QD: {}'.format(qd))
# Chieu cao cua box la IQR
plt.boxplot(data)
plt.show()
```

Kết quả:

```
IQR: 22.0
QD: 11.0
```



4.3 Điểm hạn chế của phạm vi và phạm vi phần tư

Bạn thấy thế nào? Chỉ số này có lẽ là khó hiểu đúng không?
Phạm vi có nhược điểm mang tính quyết định là dễ bị ảnh hưởng bởi các yếu tố ngoại lệ, các dữ liệu gây nhiễu sẽ làm cho phạm vi không phản ánh chính xác bản chất của dữ liệu, nhưng phạm vi phần tư và độ lệch giữa các phần tư mặc dù được hiểu là nơi mà dữ liệu phân tán tập trung nhất ở đó cũng không đủ để mô tả mức độ phân tán dữ liệu vì không phải tất cả dữ liệu đều được sử dụng ở đây. Tuy nhiên bản thân nó cũng chưa đủ tốt và có thể bạn cũng chưa cảm thấy thích nó hoàn toàn.
Do đó chúng ta có một chỉ số dễ hiểu hơn có thể sử dụng cho toàn bộ dữ liệu, đó là trung bình thiên sai, hay là trung bình độ sai khác. Ở phần tiếp theo tôi sẽ thuyết minh về phân tán và độ lệch chuẩn.
Nói về độ phân tán có thể nói rằng khái niệm phân tán và độ lệch chuẩn xuất hiện áp đảo, tôi nghĩ nhiều người có thể đã nghe về nó. Đây là một nội dung vô cùng quan trọng, tôi xin phép sẽ trình bày ở phần sau.

4.4 Tổng kết

Trong bài học này, tôi đã giới thiệu cho các bạn về phạm vi phần tư và độ lệch phần tư (thiên sai phần tư) , phạm vi từ giá trị cực đại tới giá trị cực tiểu để biểu thị mức độ dàn trải (phân tán) của dữ liệu.

- Phạm vi (range) = Giá Trị Cực Đại - Giá Trị Cực Tiểu: Dễ bị ảnh hưởng bởi giá trị ngoại lệ (dữ liệu gây nhiễu).
- Phần tư (quartile): Giá trị tại vị trí chia dữ liệu thành bốn phần bằng nhau khi mà dữ liệu đã được sắp xếp từ nhỏ tới lớn.
- Phần tư thứ hai là giá trị trung vị.
- Phạm vi phần tư $(IQR) = Q_3 - Q_1$. Ít bị ảnh hưởng bởi giá trị ngoại lệ.
- Độ lệch phần tư (thiên sai phần tư) (QD) : là một nửa của IQR.
- Những khái niệm nói trên không biểu diễn toàn bộ dữ liệu, chưa đầy đủ.
- Khái niệm tứ phân vị trong các sách thống kê ở Việt Nam và vị trí phần tư trong bài học này có ý nghĩa tương đương.

Phần tư là một nội dung vô cùng quan trọng trong thống kê, hãy ghi nhớ các khái niệm này. Trong Data Science, sử dụng phần tư để đánh giá mức độ phân tán dữ liệu là rất thường xuyên. Tuy nhiên, từ quan điểm của độ phân tán, thì phạm vi phần tư và thiên lệch phần tư chưa thích hợp để đánh giá mức độ dàn trải của dữ liệu.
Phần tiếp theo sẽ giới thiệu một chỉ số quan trọng trong các chỉ số của mức độ phân tán, đó là **phân bố** và **độ lệch chuẩn**. Phân tán và độ lệch chuẩn là nội dung thường thấy trong thống kê học và trong máy học. Vì quan trọng như vậy nên các bạn hãy gắng học nhé.

Bài 5

Nhất định hiểu về phân tán và độ lệch chuẩn

Nói về độ phân tán, ở bài trước chúng ta đã biết tới các khái niệm phạm vi, độ lệch phần tư QD và phạm vi phần tư IQR. Đó là những thứ rất cơ bản nhưng nó có khuyết điểm là không được sử dụng trong tính toán về chỉ số của toàn bộ dữ liệu.
Do đó trong bài học này, để bổ sung khiếm khuyết đó tôi sẽ giới thiệu thêm về phân tán và độ lệch chuẩn, nó là một trong những lý thuyết quan trọng nhất trong thống kê học.

- Độ lệch trung bình
- Phân tán
- Độ lệch chuẩn

Từng khái niệm sẽ được nhắc tới. Sau đó chúng ta sẽ sử dụng Python để tính toán. Ngoài ra sẽ còn nói tới phương sai bất thiên(phân tán bất thiên).

5.1 Độ lệch trung bình

Vì phạm vi cũng như IQR, QD không được sử dụng trong tính toán chỉ số của toàn bộ dữ liệu, do đó chúng ta đã có câu chuyện đó là chưa có một giá trị biểu thị đầy đủ về tính phân tán của toàn bộ dữ liệu. Vậy để nói tới độ phân tán của toàn bộ dữ liệu, thì đơn giản nhất là cách nào đây?

"Mỗi giá trị cách giá trị trung bình bao xa" được gọi là độ lệch và câu chuyện rằng việc cộng các độ lệch thường dẫn đến kết quả bằng 0 đã được đề cập trong bài học 2.

Điều này là đương nhiên vì khi ta thực hiện phép tính độ sai khác từ các điểm dữ liệu tới giá trị trung bình $(x_i - \bar{x})$ khi thì có giá trị dương, khi thì có giá trị âm, cuối cùng chúng triệt tiêu lẫn nhau và kết quả sẽ là 0. Như vậy, nếu tôi chỉ quan tâm khoảng cách là bao xa mà không quan tâm giá trị âm hay dương, tức là lấy giá trị tuyệt đối của độ lệch từ

các điểm dữ liệu tới các giá trị trung bình thì có được không?

Công thức $|x_i - \bar{x}|$ có vẻ tốt. Nếu lấy các giá trị này cộng lại và chia cho số điểm dữ liệu để lấy giá trị trung bình, chỉ số này có thể được sử dụng để biểu diễn mức độ phân tán. Nói tóm lại, ta có thể sử dụng trung bình cộng của giá trị tuyệt đối thiên sai từ các điểm dữ liệu tới giá trị trung bình cộng để biểu thị mức độ phân tán dữ liệu. Giá trị này được gọi là độ lệch trung bình (thiên sai trung bình) **(mean deviation)** hay độ lệch trung bình tuyệt đối **(mean absolute deviation)**. Nó còn được biết đến với tên MD.
Ta có công thức:

$$MD = \frac{1}{n}\left(|x_1 - \bar{x}| + |x_2 - \bar{x}| + \cdots + |x_n - \bar{x}|\right) = \frac{1}{n}\sum_{i=1}^{n}|x_i - \bar{x}|$$

Điều này thật dễ hiểu khi đại diện cho sự thay đổi trong dữ liệu, phải không? Nếu mỗi dữ liệu nằm rải rác, giá trị tuyệt đối của độ lệch của mỗi giá trị đương nhiên sẽ lớn, do đó MD sẽ lớn, và nếu nó nhỏ, MD sẽ nhỏ.
Khi toàn bộ dữ liệu giống nhau thì độ phân tán MD sẽ là 0. Khi đó đương nhiên có thể nói rằng độ phân tán là 0.
Như vậy chúng ta đã giải quyết xong vấn đề về độ phân tán hay độ phân tán của dữ liệu. Tuy nhiên còn một vấn đề nữa đó là trong công thức trên sử dụng dấu giá trị tuyệt đối. Ồ, dùng dấu giá trị tuyệt đối thì có sao đâu?

Không chỉ trong MD, nói chung trong thống kê học người ta có khuynh hướng tránh dùng giá trị tuyệt đối. Tại sao? Bởi vì việc tính toán giá trị khi âm hay khi dương làm cho phép toán thay đổi, điều đó thật là phiền hà.

Vậy thì làm thế nào để tránh được rắc rối này? Đó chính là thực hiện lũy thừa bậc 2 (bình phương), khi đó giá trị âm sẽ trở thành giá trị dương mà không cần sử dụng tới dấu giá trị tuyệt đối.

5.2 Phân tán và độ lệch chuẩn

Phân tán (variance) -các tài liệu khác gọi nó là phương sai, là bình phương độ lệch (thiên sai).
Phân tán $= \frac{1}{n}\left((x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \cdots + (x_n - \bar{x})^2\right)$
Thật vui khi dấu giá trị tuyệt đối đã biến mất. Tuy nhiên, nếu nó được bình phương, nó sẽ lệch khỏi thang của giá trị ban đầu. (Ví dụ, giả sử trọng lượng trung bình là 10kg và độ lệch là 2kg. Nếu bạn bình phương nó, nó sẽ là 4kg, điều này làm cho nó khó giải thích giá trị, phải không?)
Sẽ rất tuyệt nếu lấy căn bậc hai của phân tán (phương sai) để phù hợp với tỷ lệ như đã nói ở trên, phải không? Căn bậc hai của phương sai. Đó là độ lệch chuẩn**(standard deviation)**! Độ lệch chuẩn thường là từ viết tắt s của độ lệch chuẩn. (Nói chung, sử dụng

σ (sigma) cho độ lệch chuẩn của dữ liệu tổng thể và s cho độ lệch chuẩn của mẫu dữ liệu nhỏ hơn.)

$$s = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

Độ lệch chuẩn và phân tán (còn gọi là phương sai) là cơ sở lý luận rất quan trọng trong thống kê học, các bạn hãy nắm vững về nó.

5.3 Sử dụng Python để tính toán phân tán và độ lệch chuẩn

Ta có thể sử dụng `.std()` của thư viện NumPy để tính độ lệch chuẩn. Ngoài ra để tính độ phân tán, ta có thể sử dụng hàm `.var()` trong thư viện NumPy. Nào, đầu tiên ta sẽ tính độ phân tán từng bước như sau:

```
import numpy as np
def get_variance(samples):
    # Tinh gia tri trung binh
    mean = np.mean(samples)
    # Tinh do lech (thien sai)
    deviations = samples - mean
    # Luy thua bac 2 doi voi thien sai
    square_deviations = deviations * deviations
    # Tinh tong cac luy thua bac 2 cua thien sai
    sum_square_deviations = np.sum(square_deviations)
    # Lay tong luy thua bac 2 cua thien sai chia cho so luong du lieu(do phan tan)
    variance = sum_square_deviations/len(samples)
    return variance
```

Trông có vẻ dài nhỉ, nhưng không hề khó chút nào đúng không. Hãy xác nhận từng dòng code thử xem sao. Nếu bạn cảm thấy lo lắng, hãy sử dụng JupyterLab để chạy từng dòng code và xác nhận kết quả.
Nào bây giờ ta hãy xác nhận kết quả:

```
samples = [10, 10, 11, 14, 15, 15, 16, 18, 18, 19, 20]
# Tinh do phan tan bang ham tu tao:
print(get_variance(samples))
# Tinh do phan tan bang ham trong NumPy:
print(np.var(samples))
```

Kết quả:

```
11.537190082644628
11.537190082644628
```

Như vậy kết quả là giống nhau, bây giờ ta hãy cùng nhau tính toán độ lệch chuẩn.

```
#Tinh do lech chuan thong qua do phan tan
print(np.sqrt(get_variance(samples)))
# Tinh do lech chuan thong qua ham co san trong NumPy
print(np.std(samples))
```

Kết quả:

```
3.3966439440489826
3.3966439440489826
```

Như vậy kết quả là giống nhau đúng như những gì chúng ta đã dự đoán từ trước. Ngoài NumPy, người ta cũng có thể tính độ lệch chuẩn và độ phân tán thông qua stats của SciPy hay Pandas.
Đầu tiên hãy thử với `scipy.stats`. Đối với SciPy, độ phân tán và độ lệch chuẩn lần lượt được tính bởi các hàm `scipy.stats.tvar()` và `scipy.stats.tstd()`. Chữ "t" xuất hiện trong tên của các hàm có ý nghĩa là **t**rimmed. Bạn có thể chỉ định một phạm vi giá trị để sử dụng cho phép tính để bạn có thể xử lý các giá trị ngoại lệ-các giá trị gây nhiễu (giống với hình ảnh cắt bỏ các phần thừa ở hai đầu dữ liệu). Lần này tôi sẽ sử dụng nó như nó vốn có.

```
from scipy import stats
# Tinh do phan tan
print(stats.tvar(samples))
# Tinh do lech chuan
print(stats.tstd(samples))
```

Kết quả:

```
12.690909090909091
3.562430222602134
```

Ồ, sao kỳ vậy, kết quả khác với ban nãy khi ta tính toán bằng NumPy. Với NumPy độ phân tán là 11.5 và độ lệch chuẩn là 3.4 trong khi đó với SciPy thì độ phân tán là 12.7 và độ lệch chuẩn là 3.6, chúng có giá trị cao hơn một chút so với kết quả của NumPy.

5.4 Phân tán và Phân tán bất thiên

Cái giá trị phân tán được tính toán bởi hàm `tvar()` và `tstd` trong Module stats của thư viện SciPy thực ra là giá trị phân tán bất thiên. Phân tán bất thiên có vẻ là một từ ngữ khó hiểu, tôi sẽ giải thích ở bài sau. Công thức tính phân tán bất thiên khác phân tán trước đó chúng ta đã biết đó là phép tính không chia cho n mà chia cho $n-1$.

$$\text{Phân tán bất thiên} = \frac{1}{n-1}\left((x_1-\bar{x})^2+(x_2-\bar{x})^2+\cdots+(x_n-\bar{x})^2\right) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})^2$$

Sao kỳ vậy? Dây có lẽ là phản ứng thường thấy từ mọi người khi nhìn vào công thức trên. Tuy nhiên không hiểu thì cũng không sao đâu. Tôi nghĩ rằng đây là thời điểm khó khăn cho mọi người từ bỏ công thức trước đây để làm quen với công thức mới này. Do đó tôi sẽ giải thích trong bài học tiếp theo. `scipy.stats.tvar()` và `scipy.stats.tstd()` có kết quả lớn hơn `np.var()` và `np.std()` bởi vì chúng không phải chia cho n mà là chia cho $n-1$. Với Pandas cũng vậy, chúng cũng tính toán cho kết quả là phân tán bất thiên. Hãy tham khảo code dưới đây.

```
import pandas as pd
samples = [10, 10, 11, 14, 15, 15, 16, 18, 18, 19, 20]
df = pd.DataFrame({'sample':samples})
```

```
print(df['sample'].var())
print(df['sample'].std())
```

```
12.690909090909093
3.5624302226021345
```

Kết quả này giống với khi chúng ta sử dụng `scipy.stats`. Tại sao `scipy.stats` và `Pandas` lại chia cho $n-1$ để tính phân tán bất thiên, trong khi đó `NumPy` lại chia cho n để tính độ phân tán? Tại sao lại có hai loại phân tán? Chúng ta sẽ làm rõ điều này ở bài học tiếp theo.

5.5 Tổng kết

Trong bài học này, với tư cách là độ phân tán (phân bố), tôi đã giới thiệu trung bình thiên sai (độ lệch trung bình), phân tán, độ lệch chuẩn (thiên sai tiêu chuẩn) cho các bạn. Không giống như IQR và QD sử dụng phạm vi và vi trí phần tư, chúng có nguyên tắc cũng như đặc điểm là sử dụng tất cả dữ liệu để tính toán.
Đặc biệt, phân tán (phương sai) và độ lệch chuẩn là một trong những mục quan trọng nhất trong lý thuyết thống kê, vì vậy hãy lưu ý ghi nhớ chúng nhé.

- Độ lệch trung bình (MD) : Là giá trị trung bình của giá trị tuyệt đối độ lệch ($|x_i - \bar{x}|$). Xử lý dấu giá trị tuyệt đối là một rắc rối cần giải quyết.
- Phân tán còn gọi là phương sai (s^2) : Trung bình lũy thừa bậc 2 của độ lệch $\left((x_i - \bar{x})^2\right)$.
- Độ lệc chuẩn (s): Căn bậc hai của phân tán.
- `np.var()` và `np.std()` tính độ phân tán và độ lệch chuẩn.
- `scipy.stats.tvar()` và `scipy.stats.tstd()` có thể tính toán độ phân tán và độ lệch chuẩn nhưng kết quả tính toán là phân tán bất thiên.
- Phân tán bất thiên sử dụng công thức phân tán nhưng không chia cho n mà chia cho $n-1$.

Bài học này là tương đối dài nhưng đây là nội dung vô cùng quan trọng. Tôi nghĩ các bạn nên theo dõi câu chuyện từ: Phạm vi $\longrightarrow$ IQR/QD $\longrightarrow$ MD $\longrightarrow$ Phân tán $\longrightarrow$ Độ lệch chuẩn.
Tôi thích từ **phân tán** hơn từ *phương sai*, vì từ này gợi cho ta hình dung về **mức độ rải rác** của dữ liệu. Rõ ràng khi các điểm dữ liệu bằng nhau, dữ liệu tập trung về một điểm, ta có phân tán (phương sai) là 0. Dữ liệu càng rải rác(phân tán) xa nhau thì công thức phân tán (phương sai) sẽ cho kết quả càng lớn.

Bài 6

Phân tán bất thiên là gì? Tại sao phân tán từ dữ liệu tiêu bản lại nhỏ hơn phân tán từ dữ liệu cha?

Trong bài trước với tư cách là chỉ số biểu hiện mức độ phân tán hay phân bố, ta đã nói về những chỉ số rất quan trọng như phân tán s^2 và độ lệch chuẩn s. Chúng ta có thể sử dụng `NumPy` hay `scipy.stats` hay `Pandas` để tính giá trị phân tán và độ lệch chuẩn. Tuy nhiên chúng ta cũng nhận ra rằng kết quả tính toán của `scipy.stats` và `Pandas` cho kết quả khác với `NumPy`. Điều này được giải thích đó là vì `scipy.stats` và `Pandas` đã đưa ra kết quả tính toán phân tán là phân tán bất thiên.
Phân tán bất thiên có nghĩa là sử dụng công thức phân tán nhưng không chia cho n mà chia cho $n - 1$.
Trong bài học lần này chúng ta sẽ làm rõ rút cuộc phân tán bất thiên là gì?

- Phân tán bất thiên là chỉ số được sử dụng để ước tính độ phân tán của dữ liệu tiêu bản (dữ liệu mẫu) từ dữ liệu cha.

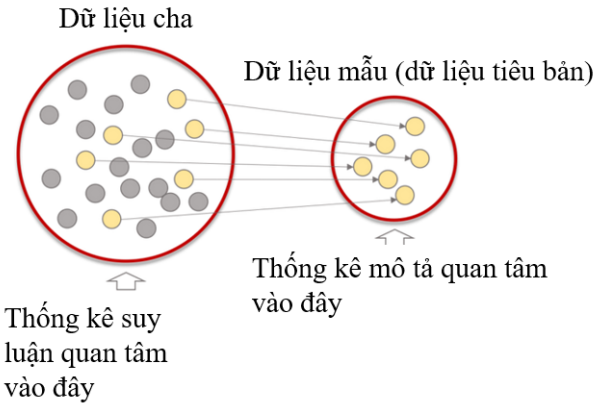
- Như lý giải ở trên thì lý do mà `NumPy` không sử dụng phân tán bất thiên là vì hóa ra `NumPy` chỉ đang tính toán chỉ số mô tả về phân tán của dữ liệu được lấy làm đối số.

- Mặt khác, `scipy.stats` và `Pandas` được cho là được sử dụng trong thống kê và khoa học dữ liệu, vì vậy chúng trả về độ phân tán bất thiên.

- Phân tán bất thiên dễ sử dụng hơn phân tán thông thường trong việc xây dựng lý thuyết thống kê, do đó, có nhiều công cụ và thư viện trả về phân tán bất thiên theo mặc định.

6.1 Ước lượng phân tán của dữ liệu cha như thế nào thì tốt?

Phần chính của thống kê học là tìm ra đặc tính của dữ liệu cha bằng cách sử dụng mẫu dữ liệu có giới hạn mà ta gọi là dữ liệu tiêu bản. Tóm lại đây là suy luận thống kê.



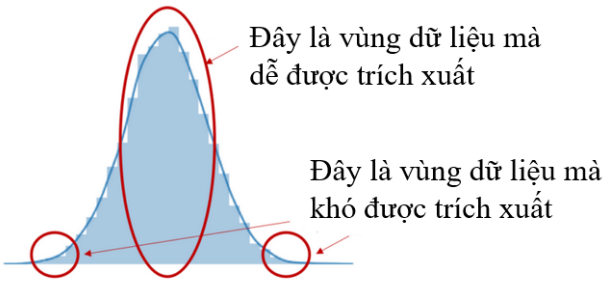
Từ các bài học trước cho đến lúc này những gì tôi đã diễn giải cho các bạn đó là thống kê mô tả. Thống kê mô tả đó là mô tả đặc tính của dữ liệu từ dữ liệu vốn có trong tay. Hãy tách biệt một cách rõ ràng sự khác biệt này khi các bạn học thống kê.
Chúng ta sẽ xem xét chi tiết ở phần sau, chúng ta sẽ sử dụng chỉ số của dữ liệu tiêu bản để suy ra chỉ số dân số. Ví dụ, chúng ta biết rằng giá trị trung bình của một dữ liệu tiêu bản có thể được sử dụng để suy ra giá trị trung bình của cả một tập hợp (dữ liệu cha). Chi tiết sẽ được giải thích lại ngay trong khóa học này.
Nói cách khác, ngay cả khi bạn nói "trung bình" thì nó sẽ có nghĩa khác nhau trong những ngữ cảnh thống kê mô tả và trong ngữ cảnh thống kê suy luận.
Theo cách này, giá trị của dữ liệu tiêu bản được sử dụng để ước tính giá trị đặc trưng của dữ liệu cha, và phép thống kê được sử dụng để ước tính đó được gọi là ước lượng. Bạn không cần thiết phải nhớ những từ ngữ này. Trong công việc không có nhiều trường hợp mà bạn cần phải phân biệt rõ ràng điều này. Đặt tên trong khi học thống kê là rất hữu ích, vì vậy trong cuốn sách này tôi sẽ giới thiệu các từ này và tôi sẽ sử dụng chúng.
Tóm lại, giá trị trung bình được tính từ dữ liệu tiêu bản (dữ liệu mẫu) được sử dụng bằng công cụ ước lượng để từ đó ước tính giá trị trung bình của dữ liệu cha. Vậy khi ước tính độ phân tán của dữ liệu cha có thể sử dụng ước lượng hay không? Có thể sử dụng ước lượng độ phân tán của dữ liệu tiêu bản hay không?
Trong thực tế, độ phân tán của dữ liệu tiêu bản nhỏ hơn một chút so với độ phân tán của dữ liệu cha. Đầu tiên bạn hãy lý giải điều này bằng cảm giác cũng được.

6.2 [Lý giải bằng hình ảnh]Tại sao độ phân tán của dữ liệu tiêu bản lại nhỏ hơn độ phân tán của dữ liệu cha?

Cho tới bây giờ, đây hẳn là câu hỏi các bạn vẫn còn đang băn khoăn. Nhưng nếu suy nghĩ kỹ một chút ta sẽ thấy điều này là đương nhiên.
Khi mà chúng ta lấy ngẫu nhiên từ dữ liệu cha để hình thành nên một dữ liệu mẫu, có thể nói rằng dữ liệu mẫu (dữ liệu tiêu bản) là một phần của dữ liệu cha. Vì chúng ta không có lấy giá trị nào nằm ngoài phạm vi của dữ liệu cha, vì vậy độ phân tán của dữ liệu mẫu rõ ràng có thể nhỏ hơn độ phân tán của dữ liệu cha, đây là điều đương nhiên.
Sau đây chúng ta hãy cùng thực hiện đoạn code dưới đây:

```
population = np.array([1, 5, 10, 11, 14, 15, 15, 16, 18, 18, 19, 20, 25, 30])
# Trich xuat du lieu:
samples = np.random.choice(population, size=3)
print('population variance is {}'.format(np.var(population)))
print('sample variance is {}'.format(np.var(samples)))
print('samples: {}'.format(samples))
```

Bằng `np.var()` ta sẽ tính độ phân bố của `population` với tư cách là dữ liệu cha ($[1, 5, 10, 11, 14, 15, 15, 16, 18, 18, 19, 20, 25, 30]$) và tính độ phân bố của `sample` với tư cách là dữ liệu mẫu hay dữ liệu tiêu bản.
Bạn hãy thực thi code trên nhiều lần, trong nhiều trường hợp chẳng phải độ phân tán của dữ liệu mẫu nhỏ hơn độ phân tán của dữ liệu cha là 51.39 hay sao?
Trừ khi bạn lấy sự kết hợp ở các đầu mút, chẳng hạn [1,5,30] hoặc [1,25,30], còn bình thường thì độ phân tán của dữ liệu mẫu có xu hướng thấp hơn độ phân tán của dữ liệu cha. Do lấy ngẫu nhiên, dữ liệu mẫu được trích xuất từ dữ liệu cha, vì vậy phân bố của dữ liệu mẫu thường có giá trị lệch so với phân bố của dữ liệu cha. Ngược lại xác xuất mà một giá trị nằm ngoài phân bố thường có khả năng thấp, và tất nhiên độ phân tán dữ liệu mẫu có khuynh hướng nhỏ hơn độ phân tán của dữ liệu cha $\left(\frac{n-1}{n}\right)$.



Nhân tiện hình này có thể được vẽ như sau:

```
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
sns.distplot(np.random.randn(int(1e4)), bins=30)
plt.axis('off')
```

`seaborn` là thư viện của Python được sử dụng trong khoa học dữ liệu. Về `displot` nó được giới thiệu trong khóa học nhập môn Python cho Data science do diễn đàn tuhocvba.net tổ chức, bạn hãy tham gia đăng ký học nhé. Tôi nghĩ điều này đã cho các bạn thấy rằng phương sai (độ phân tán) của dữ liệu mẫu hay còn gọi là dữ liệu tiêu bản có xu hướng nhỏ hơn phương sai của dữ liệu cha. Số lượng dữ liệu của dữ liệu mẫu (n) càng nhỏ thì xác suất các giá trị lệch khỏi phân bố được đưa vào dữ liệu mẫu càng ít, vì vậy bạn có thể thấy một hình ảnh rõ ràng rằng độ phân tán cũng sẽ trở nên nhỏ hơn.

6.3 [Lý giải bằng Số Học]Tại sao độ phân tán của dữ liệu tiêu bản lại nhỏ hơn độ phân tán của dữ liệu cha?

Nào, bây giờ chúng ta hãy thử giải thích bằng kiến thức toán. Vì các giá trị đặc trưng của dữ liệu cha thường được biểu thị bằng các chữ cái Hy Lạp, nên trong khóa học này, giá trị trung binh của dữ liệu cha sẽ được ký hiệu là μ. Và độ phân tán sẽ được ký hiệu là σ^2.
Độ phân tán của dữ liệu cha la σ^2 và của dữ liệu mẫu là s^2 và chúng được áp dụng công thức như dưới đây, nếu như bạn không hiểu chúng, xin hãy xem lại bài học trước.

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$$

$$s^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

Độ phân tán của dữ liệu cha là σ^2 được tính thông qua giá trị trung bình của dữ liệu cha là μ, trong khi đó độ phân tán của dữ liệu mẫu là s^2 được tính thông qua giá trị trung bình của dữ liệu mẫu là $\bar{x}$.
Tuy nhiên, khi tính độ phân tán của dữ liệu mẫu chúng ta sử dụng giá trị trung bình, giá trị trung bình này là $\bar{x}$, nếu thay vào đó ta sử dụng giá trị trung bình của dữ liệu cha thì có được không?

$$s^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$$

Nếu như chúng ta tính toán độ phân tán của dữ liệu mẫu dùng cho việc ước lượng độ phân tán của dữ liệu cha, như thế thì nếu sử dụng giá trị trung bình của dữ liệu cha μ thì kết quả có vẻ như sẽ tiệm cận tới độ phân tán của dữ liệu cha là σ^2.
Tuy nhiên, giá trị trung bình của dữ liệu cha là μ thông thường chúng ta không biết, do đó dù muốn sử dụng cũng không có cách nào mà sử dụng nó. Đó là lý do chúng ta phải ước lượng giá trị trung bình $\bar{x}$ của dữ liệu mẫu thay cho giá trị trung bình của dữ liệu cha là μ.
Như đã nói trong các bài học trước, giá trị trung bình cộng là giá trị mà tổng bình phương độ sai khác từ các điểm dữ liệu tới nó là nhỏ nhất.

Chẳng hạn ta có giá trị X bất kỳ. Tổng bình phương độ sai khác từ các điểm dữ liệu trong dữ liệu mẫu tới X là:

$$\sum_{i=1}^{n} (x_i - X)^2$$

Dữ liệu mẫu có giá trị trung bình cộng là $\bar{x}$. Và tổng trên đạt giá tị nhỏ nhất khi $X = \bar{x}$. Độ lệch chuẩn của dữ liệu mẫu chính là giá trị nhỏ nhất của tổng trên.

$$s^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2$$

Nếu như X là một giá trị khác thì tổng trên sẽ có giá trị lớn hơn. Như vậy:

$$\frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2 \leq \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2$$

Đến đây chúng ta đã làm sáng tỏ vấn đề. Lưu ý rằng giá trị trung bình của dữ liệu cha μ thường là giá trị mà chúng ta không biết, và vì thế chúng ta thường ước lượng bằng cách sử dụng giá trị trung bình của một mẫu dữ liệu nhỏ hơn là $\bar{x}$, và vì vậy mà độ phân tán khi tính toán mà sử dụng giá trị trung bình $\bar{x}$ sẽ cho ra giá trị độ phân tán nhỏ hơn một chút.

6.4 Phân tán bất thiên (Phương sai không chệch) có thể được sử dụng làm công cụ ước tính cho phương sai tổng thể (phân tán của dữ liệu cha)

Đến bây giờ có lẽ các bạn đều đã hiểu, thông thường khi tính toán độ phân tán của dữ liệu mẫu s^2 sẽ nhỏ hơn độ phân tán của dữ liệu cha là σ^2.
Độ phân tán của dữ liệu mẫu s^2 không thể sử dụng để ước lượng giá trị độ phân tán của dữ liệu cha σ^2. Phân tán bất thiên được sử dụng làm công cụ, nó có thể nhỏ hơn hay lớn hơn một chút so với độ phân tán của dữ liệu mẫu s^2. Tóm lại phân tán bất thiên là giá trị ước lượng của độ phân tán dữ liệu cha σ^2.
Vì vậy, chúng ta không sử dụng `scipy.stats` hoặc `Pandas` để tính độ phân tán một cách nhanh chóng, thay vào đó chúng ta sử dụng lý luận của thống kê học để tính độ phân tán bất thiên.
Trong lý luận của thống kê học, độ phân tán bất thiên dễ xử lý hơn độ phân tán thông thường, vì vậy mà nhiều công cụ thống kê trả về giá trị độ phân tán bất thiên một cách mặc định khi tính toán liên quan tới độ phân tán.
Thành thật mà nói, tôi không nghĩ rằng có ai đó thực sự quan tâm đến việc sử dụng độ phân tán thông thường hay độ phân tán bất thiên trong thống kê mô tả như một công cụ ước tính. Tuy nhiên để chắc chắn, bạn nên xem công cụ hoặc thư viện nào đang tính toán độ phân tán, nếu n lớn thì sự khác biệt là rất nhỏ, vì vậy tôi không nghĩ chúng ta cần phải lo lắng về điều đó, đặc biệt là với các mẫu dữ liệu lớn.

6.5 Tại sao lại là $n-1$, bất thiên là gì?

Chúng ta đã lý giải được rằng độ phân tán của dữ liệu mẫu có xu hướng nhỏ hơn độ phân tán của dữ liệu cha. Tuy nhiên có một câu hỏi đặt ra, tại sao lại chia cho $n-1$? Tại sao không chia cho $n-2$ hoặc $n-3$? Đúng vậy, đây là một thắc mắc rất tự nhiên. Để giải thích việc này trước hết chúng ta cần làm rõ "Tính Bất Thiên". Chúng ta nói về độ phân tán bất thiên, vậy thì "bất thiên" có nghĩa là gì? Để giải thích điều này sẽ rất dài và tốn một chút thời gian, do đó tôi xin phép được trình bày trong bài học tới. Tôi nghĩ rằng với người nhập môn, các bạn hãy cố gắng nhớ những điều dễ hiểu trước đã. Không cần phải cố gắng nhớ quá nhiều kiến thức một lúc, nó sẽ khiến các bạn cảm thấy quá tải.

6.6 Tổng kết

Trong bài học lần này tôi đã giải thích cho các bạn về độ phân tán bất thiên. Vậy rút cục độ phân tán bất thiên là cái gì? Tại sao khi tính toán độ phân tán của dữ liệu mẫu ta không sử dụng cách tính thông thường? Chúng ta đã lý giải được rằng nó có giá trị nhỏ hơn độ phân tán của dữ liệu cha.
Đây được coi là một nội dung vô cùng quan trọng. Nếu như các bạn vẫn còn chưa hiểu, xin hãy đọc lại nội dung bài học này thêm một lần nữa. Và hãy thử tra cứu google để tìm hiểu thêm. Sau đây tôi xin tóm tắt nội dung bài học này:

- Độ phân tán bất thiên là đại lượng được để ước lượng độ phân tán của dữ liệu cha.
- Khi tính toán độ phân tán của dữ liệu mẫu theo cách thông thường, nó sẽ có khuynh hướng có giá trị nhỏ hơn độ phân tán của dữ liệu cha.
- Lý do mà độ phân tán của dữ liệu mẫu nhỏ hơn dữ liệu cha đó là vì khi tính toán độ phân tán ta không sử dụng được giá trị trung bình của dữ liệu cha mà sử dụng giá trị trung bình của dữ liệu mẫu để tính toán.

Chúng ta vẫn còn một khúc mắc về lý do chia cho $n-1$ mà tôi chưa giải thích nhưng tôi nghĩ các bạn đã hình dung ra được độ phân tán bất thiên là gì.
Trong bài học tới tôi sẽ trình bày về tính bất thiên cũng như lý do chia cho $n-1$.

Bài 7

Lý do độ phân tán bất thiên được tính bằng phép chia cho $n-1$. Tính bất thiên nghĩa là gì?

Ở bài học trước chúng ta đã nói về câu chuyện độ phân tán bất thiên, nó là giá trị ước lượng độ phân tán của dữ liệu cha. Nếu độ phân tán của dữ liệu mẫu là s^2 và độ phân tán của dữ liệu cha là σ^2 thì độ phân tán của dữ liệu mẫu s^2 có khuynh hướng nhỏ hơn σ^2.
Do đó trong bài học này chúng ta sẽ làm rõ tính bất thiên rút cuộc là gì? Và tại sao không phải là n mà lại là chia cho $n-1$. Tôi nghĩ bất cứ ai khi bắt đầu học thống kê sẽ cùng có thắc mắc này, do đó việc làm rõ khúc mắc này là cần thiết.
Nhìn riêng về mặt ngôn ngữ ta có thể thấy:

- Độ phân tán bất thiên là một công cụ để ước tính bất thiên (không thiên vị) của độ phân tán tổng thể (độ phân tán của dữ liệu cha), nên từ "bất thiên" được sử dụng.
- Giá trị ước lượng nếu tính trung bình sẽ cho giá trị trùng với tham số của dữ liệu mẫu, khi đó công cụ ước tính này được cho là không thiên sai (không chệch) và một công cụ ước tính như thế được gọi là công cụ ước tính bất thiên.
- Tuy nhiên tại sao không phải là $n-2$ hay $n-3$ mà lại là $n-1$ thì đây là điều cần phải chứng minh làm rõ.

Sẽ có nhiều bạn băn khoăn, nãy giờ nói gì tôi cũng không hiểu nữa. Được rồi, chúng ta sẽ bắt tay vào ngay bây giờ đây, các bạn hãy kiên nhẫn một chút.

7.1 Tính bất thiên là gì?

Bạn nghĩ rằng bạn đã tính toán độ phân tán của dữ liệu cha và của dữ liệu mẫu. Tuy nhiên vì độ phân tán của dữ liệu mẫu thường có khuynh hướng nhỏ hơn độ phân tán của dữ liệu cha, chúng ta hiểu rằng độ phân tán của dữ liệu mẫu có khuynh hướng nhỏ hơn độ phân tán bất thiên, là giá trị mà chúng ta sử dụng để ước lượng độ phân tán của dữ

liệu cha, nên có thể nói rằng độ phân tán của dữ liệu mẫu không bằng với độ phân tán của dữ liệu cha, nó có một độ lệch sai khác nhất định. Cũng đúng thôi, vì dữ liệu mẫu là dữ liệu mà chúng ta trích xuất từ dữ liệu cha, tùy thuộc sự trích xuất khác nhau mà độ phân tán sẽ có giá trị khác nhau, do đó dù thế nào đi nữa, giá trị độ phân tán mà chúng ta tính được từ dữ liệu mẫu là một giá trị mang tính ước lượng.

Một ví dụ đơn giản, chúng ta thử suy nghĩ về việc ước lượng giá trị trung bình của dữ liệu cha là μ. Rõ ràng vì dữ liệu cha là một khối dữ liệu lớn cho nên giá trị trung bình thực tế là không thể tính được, do đó mới nói μ chỉ có tính ước lượng dựa vào dữ liệu mẫu có giá trị trung bình là $\bar{x}$.

Nào, các bạn hãy thử nhìn vào code dưới đây, đây là đoạn code đã sử dụng trước đây, lần này tôi biến đổi một chút. Từ dữ liệu cha, tôi trích xuất ngẫu nhiên 5 giá trị, sau đó tính giá trị trung bình cho dữ liệu mẫu.

```
import numpy as np
population = np.array([1, 5, 10, 11, 14, 15, 15, 16, 18, 18, 19, 20, 25, 30])
# Trich xuat ngau nhien
samples = np.random.choice(population, size=5)
print('population mean is {}'.format(np.mean(population)))
print('sample mean is {}'.format(np.mean(samples)))
print('samples: {}'.format(samples))
```

Nào bây giờ bạn hãy thực thi chạy code trên nhiều lần, và bạn sẽ thấy rằng giá trị trung bình của dữ liệu mẫu không trùng khớp với giá trị trung bình của dữ liệu cha. Tùy thuộc vào mẫu dữ liệu được trích xuất mà giá trị trung bình sẽ khác nhau, điều này là đương nhiên đúng không nào.

Nếu thử với vòng lặp với một vạn lần, chúng ta có thể tính giá trị trung bình của một vạn dữ liệu mẫu, giá trị trung bình của một vạn mẫu này sẽ như thế nào?

```
import numpy as np
population = np.array([1, 5, 10, 11, 14, 15, 15, 16, 18, 18, 19, 20, 25, 30])
# Trich xuat mau du lieu ngau nhien
##Gia tri trung binh cua du lieu cha
print('population mean is {}'.format(np.mean(population)))
## Nap gia tri ngau nhien cua cac mau du lieu vao list nay
sample_mean_list = []
count = 10000
## Thuc hien trich xuat mau du lieu 10000 lan
for i in range(count):
###Trich xuat mau du lieu ngau nhien
samples = np.random.choice(population, size=5)
###Nap gia tri trung binh cua mau du lieu vao list
sample_mean_list.append(np.mean(samples))
print('sample_mean_list mean is {}'.format(np.mean(sample_mean_list)))
```

Kết quả là:

```
population mean is 15.5
sample_mean_list mean is 15.5386
```

Sau khi trích xuất dữ liệu mẫu tới một vạn lần và tính giá trị trung bình của một vạn mẫu này được một vạn giá trị trung bình. Sau đó lại lấy giá trị trung bình của một vạn giá trị trung bình nói trên, ta được một giá trị rất gần với giá trị trung bình của dữ liệu cha. Nếu tăng lên là 10 vạn lần, 100 vạn lần trích xuất dữ liệu mẫu, thì giá trị này càng tiến gần tới giá trị trung bình của dữ liệu mẫu. Bạn có thể thay giá trị cho biến số `count`

trong code trên.
Như vậy giá trị trung bình của dữ liệu mẫu khi thì lớn hơn, khi thì nhỏ hơn giá trị trung bình của dữ liệu cha, nhưng nếu như chúng ta lặp vô hạn lần thì giá trị trung bình của dữ liệu mẫu $\bar{x}$ sẽ xấp xỉ bằng giá trị trung bình của dữ liệu cha μ.
Giống như nói ở trên, nếu có một giá trị ước lượng mà khi lấy giá trị trung bình các giá trị này thì nó trùng với tham số của dữ liệu cha, ta nói giá trị ước lượng này là bất thiên (unbiased), nó có nghĩa là không chệch, hay không thiên vị. Công cụ ước lượng như thế gọi là công cụ ước lượng bất thiên (unbiased estimator: ước lượng không thiên vị).
Tóm lại giá trị trung bình $\bar{x}$ của dữ liệu mẫu là công cụ ước lượng bất thiên cho giá trị trung bình μ của dữ liệu cha.
Đến đây các bạn cũng đã hiểu, cho dù chúng ta có lấy vô hạn số dữ liệu mẫu để tính giá trị trung bình độ phân tán s^2 của các dữ liệu mẫu đó thì giá trị này cũng không trùng khớp với độ phân tán của dữ liệu cha σ^2.
Nếu lấy dữ liệu mẫu vô hạn lần rồi tính giá trị trung bình của **độ phân tán bất thiên** thì nó sẽ xấp xỉ bằng với độ phân tán của dữ liệu cha σ^2. Vì vậy, độ phân tán bất thiên ở đây được định nghĩa là chia cho $n-1$.
Bạn hãy thử sửa code trên, thử tính một vạn lần độ phân tán bất thiên của dữ liệu mẫu rồi lấy giá trị trung bình của chúng, sau đó so sánh với độ phân tán của dữ liệu cha xem sao.

```
import numpy as np
from scipy import stats
population = np.array([1, 5, 10, 11, 14, 15, 15, 16, 18, 18, 19, 20, 25, 30])
# Trich xuat mau du lieu ngau nhien
##Gia tri trung binh cua du lieu cha
print('Do phan tan cua population la {}'.format(np.var(population)))
## Nap gia tri ngau nhien cua cac mau du lieu vao list nay
sample_mean_list = []
count = 10000
## Thuc hien trich xuat mau du lieu 10000 lan
for i in range(count):
###Trich xuat mau du lieu ngau nhien
samples = np.random.choice(population, size=5)
###Nap do phan tan bat thien cua mau du lieu vao list
sample_mean_list.append(stats.tvar(samples))
#Tinh gia tri trung binh cua 10000 gia tri do phan tan bat thien
print('sample_mean_list co do phan tan la {}'.format(np.mean(sample_mean_list)))
```

7.2 Cách nghĩ về giá trị kỳ vọng

Trước khi giải thích tại sao lại chia cho $n-1$, ta cần hiểu về giá trị kỳ vọng. Vậy giá trị kỳ vọng được hiểu như thế nào?

Giá trị kỳ vọng

Khi chúng ta mua vé số với số tiền 1000 yên, chúng ta kỳ vọng sẽ nhận được 400 yên. Điều này là vì thông thường những người mua vé số thì trung bình sẽ được nhận lại 400 yên. Tất nhiên cũng có một bộ phận trúng thưởng lớn và nhận về số tiền lớn nhưng nếu tính trung bình những người đã mua vé số thì số tiền trúng thưởng trung bình là 400 yên nếu bạn bỏ ra số tiền 1000 yên để mua vé số.

Nói tóm lại đây là giá trị trung bình dựa trên suy luận logic. Dù nói là trung bình thì nếu đã có khái niệm về xác suất, cái giá trị trung bình này được gọi là giá trị kỳ vọng **(expected value)**.

Giá trị biến động thay đổi do xác suất gọi là biến số xác suất. Ví dụ số tiền trúng xổ số, giá trị khi tung xúc sắc ... được gọi là biến số xác suất, ta ký hiệu là x, khi đó giá trị kỳ vọng được biểu diễn là $E(x)$.

Ví dụ một con xúc sắc có các điểm là 1,2,3,4,5,6, vậy giá trị kỳ vọng khi ta tung xúc sắc sẽ là:

$$E(x) = \frac{1+2+3+4+5+6}{6} = 3.5$$

Có thể thấy, giá trị trung bình của dữ liệu mẫu, hay độ phân tán của dữ liệu mẫu, chúng đều là các biến số xác suất, có giá trị thay đổi. Bởi vì khi ta lấy ngẫu nhiên dữ liệu mẫu từ dữ liệu cha, và tính toán giá trị trung bình thì đương nhiên giá trị trung bình này sẽ biến động thay đổi tùy thuộc vào mẫu dữ liệu, vì vậy có thể hiểu giá trị trung bình này là một biến xác suất.

Giá trị kỳ vọng $E(\bar{x})$ của giá trị trung bình $\bar{x}$ của dữ liệu mẫu là giá trị mà ta mong muốn nó trùng khớp với giá trị trung bình μ của dữ liệu cha. Bởi vì giá trị trung bình của dữ liệu mẫu là giá trị ước lượng bất thiên của giá trị trung bình của dữ liệu cha.

$$E(\bar{x}) = \mu$$

Ta có chứng minh đơn giản như sau:

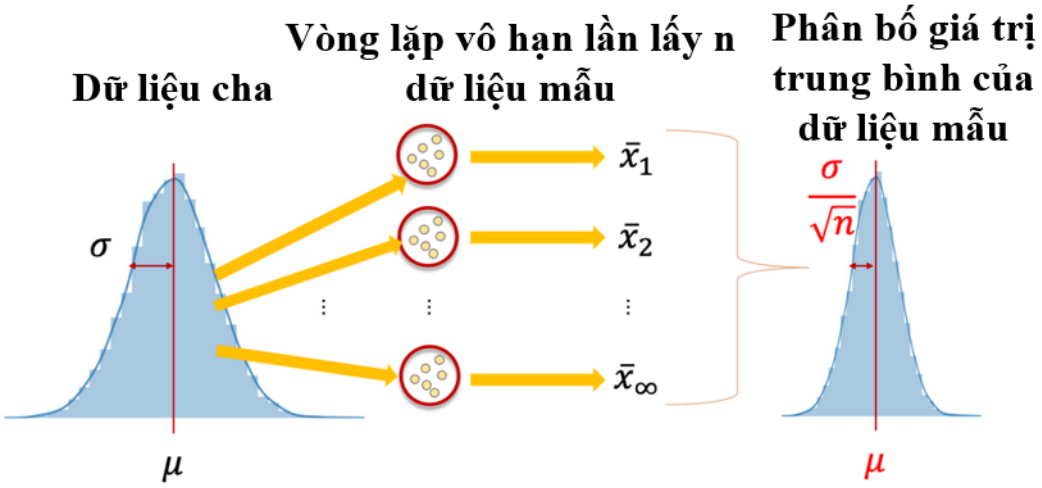
$$\begin{aligned} E(\bar{x}) &= E\left(\left[\frac{1}{n}(x_1 + x_2 + \cdots + x_n)\right]\right) \\ &= \frac{1}{n}E([(x_1 + x_2 + \cdots +_n)]) \\ &= \frac{1}{n}(E(x_1) + E(x_x) + \cdots + E(x_n)) \\ &= \frac{1}{n}n\mu = \mu. \end{aligned}$$

Dữ liệu có giá trị là x được lấy ngẫu nhiên từ dữ liệu cha thì đương nhiên giá trị kỳ vọng của x là $E(x)$ sẽ bằng với μ, vì vậy mà $E(x_1)$ hay $E(x_2)$ ...tất cả đều bằng μ.

Ngoài ra, dữ liệu mẫu có giá trị trung bình là $\bar{x}$, giá trị kỳ vọng của độ phân tán của nó là $E\left(\left[\frac{1}{n}\sum(\bar{x}-\mu)^2\right]\right)$ bằng $\frac{\sigma^2}{n}$.

Đây là phần rất quan trọng nên các bạn hãy ghi nhớ nhé. Khi $n = 1$ thì chúng ta chỉ lấy một dữ liệu mẫu, giá trị kỳ vọng chính là giá trị trung bình của dữ liệu mẫu. Độ phân tán khi đó bằng với độ phân tán của dữ liệu cha σ^2. Ngược lại khi n vô cùng lớn, mỗi lần lấy dữ liệu mẫu thì giá trị trung bình của dữ liệu mẫu lại gần với giá trị trung bình của dữ liệu cha μ nên độ phân tán trở nên nhỏ là điều chúng ta đã lý giải được.

Nhìn vào biểu đồ dưới đây sẽ dễ hiểu hơn. Ở đây không mô tả độ phân tán mà mô tả độ lệch chuẩn.



7.3 Lý do độ phân tán bất thiên chia cho $n-1$

Thành thực mà nói thì cái chứng minh này không quá quan trọng, tức là bạn không biết thì cũng không sao. Thông thường trong công việc làm về khoa học dữ liệu thì tính cần thiết phải hiểu chứng minh này là không có, tức là không biết chứng minh này thì cũng không sao đâu.
Do đó với người bình thường thì các bạn có thể lướt nhanh tới phần tổng kết của bài học này. Còn với người có hứng thú tìm hiểu, chúng ta sẽ tiếp tục cùng nhau với những suy luận như dưới đây.
Có thể nói rằng độ phân tán của dữ liệu cha σ^2 là giá trị kỳ vọng của giá trị trung bình của phép tính tổng bình phương độ sai khác từ các điểm dữ liệu trong dữ liệu cha tới giá trị trung bình của nó là μ.

$$\sigma^2 = E\left(\left[\frac{1}{n}\sum(x_i-\mu)^2\right]\right)$$

Dữ liệu mẫu có giá trị trung bình là $\bar{x}$ nên ta sẽ đưa nó vào biểu thức bên trái, biến đổi như sau:

$$
\begin{aligned}
\sigma^2 &= E\left[\frac{1}{n}\sum (x_i-\mu)^2\right] \\
&= E\left[\frac{1}{n}\sum \left(x_i \boxed{-\bar{x}+\bar{x}} -\mu\right)^2\right] \\
&= E\left[\frac{1}{n}\sum (x_i-\bar{x})^2 + \sum 2\,(x_i-\bar{x})\,(\bar{x}-\mu) + \sum (\bar{x}-\mu)^2\right] \\
&= E\left[\frac{1}{n}\sum (x_i-\bar{x})^2 + 2(\bar{x}-\mu)\underbrace{\sum (x_i-\bar{x})}_{\text{Tổng độ sai khác từ các điểm dữ liệu tới giá trị trung bình là 0}} + \sum (\bar{x}-\mu)^2\right] \\
&= E\left[\frac{1}{n}\sum (x_i-\bar{x})^2\right] + \underbrace{E\left[\frac{1}{n}\sum (\bar{x}-\mu)^2\right]}_{\text{Độ phân tán của giá trị trung bình của dữ liệu mẫu là } \frac{\sigma^2}{n}} \\
&= E\left[\frac{1}{n}\sum (x_i-\bar{x})^2\right] + \frac{\sigma^2}{n}.
\end{aligned}
$$

$(a+b)^2 = a^2 + 2ab + b^2$

Đến đây chúng ta chuyển vế và tính được như sau:

$$
\begin{aligned}
E\left[\frac{1}{n}\sum (x_i-\bar{x})^2\right] &= \sigma^2 - \frac{\sigma^2}{n} \\
&= \frac{n-1}{n}\sigma^2
\end{aligned}
$$

Công thức cuối cùng chính là điều mà tôi muốn nói. Khi chúng ta lấy vô số dữ liệu mẫu và tính toán giá trị kỳ vọng của độ phân tán dữ liệu mẫu $\frac{1}{n}\sum (x_i-\bar{x})^2$ thì nó nhỏ hơn độ phân tán dữ liệu cha σ^2 và có tỷ số là $\frac{n-1}{n}$.

Nói cách khác nếu lầy độ phân tán của dữ liệu mẫu $\frac{1}{n}\sum (x_i-\bar{x})^2$ nhân với $\frac{n}{n-1}$ ta sẽ thu đươc giá trị bằng với độ phân tán của dữ liệu cha σ^2.

Đến đây tôi nghĩ các bạn đã hiểu lý do mà độ phân tán bất thiên lại chia cho $n-1$.

Cách chứng minh này không khó, những người có hứng thú tìm hiểu hãy nhớ công thức này. Tuy nhiên cách chứng minh công thức này không quan trọng cho nên dù bạn không chứng minh được nó thì cũng không sao, các bạn vẫn có thể tiếp tục học khoa học dữ liệu.

7.4 Tổng kết

Bài học này tương đối dài nhưng có mấy điểm quan trọng dưới đây mà tôi muốn các bạn ghi nhớ.

- Giá trị ước lượng mà khi ta lấy giá trị khi lấy trung bình của nó ta thu được tham số của dữ liệu cha (là giá trị kỳ vọng), giá trị ước tính này được gọi là bất thiên và công cụ ước tính như vậy gọi là công cụ ước tính bất thiên.

- Độ phân tán bất thiên (Phương sai bất thiên) là một công cụ ước tính bất thiên của độ phân tán dữ liệu cha.

- Giá trị trung bình của dữ liệu mẫu là một công cụ ước tính bất thiên của giá trị trung bình của dữ liệu cha.
- Độ phân tán của giá trị trung bình của dữ liệu mẫu là $\frac{\sigma^2}{n}$.
- Đại lượng thống kê từ dữ liệu mẫu là một biến xác suất, giá trị cảu nó biến động có tính xác suất.

Mục nào ở trên cũng vô cùng quan trọng, đặc biệt là mục cuối cùng, giá trị ước tính thống kê từ dữ liệu mẫu là một điểm rất quan trọng khi chúng ta lý giải cũng như tìm hiểu về thống kê học. Tôi nghĩ nếu có điều kiện tôi sẽ bổ sung thêm những chú ý về nó.
Đến đây tôi đã trình bày về độ phân bố. Khái niệm phân tán bất thiên đã dẫn dắt chúng ta đi chệch khỏi câu chuyện về độ phân tán (phân bố) ban đầu.
Thực tế khi giải thihs về mức độ phân tán (phân bố), chúng ta thường sử dụng độ lệch chuẩn. Nhưng mà ngay cả khi bạn đươc cho biết rằng độ lệch chuẩn là từng này, tôi không nghĩ rằng bạn có thể biết các giá trị dữ liệu thay đổi bao nhiêu, như thế nào.
Trong bài tiếp theo, tôi ngĩ mình muốn nói về cách sử dụng độ phân tán (phân bố) này, nó được sử dụng như thế nào, đó sẽ là câu chuyện mà tôi muốn cùng các bạn thảo luận.

Bài 8

Làm thế nào để đọc độ phân tán từ độ lệch chuẩn?

Trong các bài học trước, chúng ta đã tìm hiểu về độ phân tán phân bố.
Bài học 4: Phạm vi, IQR(Phạm vi phần tư), QD(Độ lệch phần tư).
Bài học 5: Trung bình độ lệch, phân tán, độ lệch chuẩn.
Bài học 6, Bài học 7: Phân tán bất thiên.
Phạm vi là khoảng giá trị từ vị trí dữ liệu có giá trị nhỏ nhất tới vị trí dữ liệu có giá trị lớn nhất, do đó nó có nhược điểm là chứa cả những giá trị ngoại lệ, đó là những giá trị gây nhiễu không có ý nghĩa. Với IQR, QD có điểm mạnh là lược bớt những giá trị ngoại lệ nhưng vì nó không sử dụng toàn bộ dữ liệu trong tính toán do đó mà độ tin cậy thấp. Vì lý do đó có thể suy nghĩ tới trung bình độ lệch nhưng vì nó chứa dấu giá trị tuyệt đối nên rất khó để xử lý, cúng ta thực hiện bình phương chúng và từ đó có khái niệm về độ phân tán. Vì khi bình phương sẽ làm giá trị biến đổi do đó, do đó chúng ta lấy căn bậc hai của độ phân tán và lúc này chúng ta có giá trị có tên gọi là độ lệch chuẩn. Trường hợp chúng ta ước lượng độ phân tán của dữ liệu cha, chúng ta có phân tán bán bất thiên.

Xâu chuỗi các sự việc như trên chúng ta sẽ thấy kiến thức trở nên dễ nhớ hơn.
Chúng ta đã nói rất nhiều về độ phân bố thế nhưng quan trọng nhất đó là độ lệch chuẩn. Tôi nghĩ rằng trong hầu hết các lý luận thì độ lệch chuẩn đều được sử dụng để biểu thị về độ phân tán, phân bố của dữ liệu.

8.1 Có bao nhiêu dữ liệu nằm trong khoảng: Trung bình $\pm$ độ lệch chuẩn

Vì độ lệch chuẩn là $\cdots$ cho nên sẽ có chừng này dữ liệu nằm trong khoảng từ giá trị trung bình $\pm$ độ lệch chuẩn. Nếu như hiểu được điều này thì ta có thể hiểu được phân bố của dữ liệu, đúng không nào?
Ví dụ, điểm trung bình là 50 điểm, độ lệch chuẩn là 10 điểm, như vậy nói về độ phân bố của dữ liêu, giả sử trong khoảng từ 50 ± 10 điểm, thì có bao nhiêu dữ liệu nằm trong khoảng này? Nếu như hiểu được điều này thì bạn có thể nắm bắt được dữ liệu được phân

tán như thế nào rồi đúng không?

Trong thực tế, điều này phụ thuộc vào sự phân bố của dữ liệu, vì vậy không thể tính toán chính xác tỷ lệ dữ liệu có trong khoảng từ giá trị trung bình ± độ lệch chuẩn.

Tính toán chính xác có thể là không cần thiết, ở đây tôi đưa ra giá trị mang tính cảm quan, trong đó $\bar{x}$ là giá trị trung bình, s là độ lệch chuẩn, khi đó:

Nằm trong phạm vi $\bar{x} \pm s$ ước chừng có $\frac{2}{3}$ dữ liệu.

Nằm trong phạm vi $\bar{x} \pm 2s$ ước chừng có 95% dữ liệu.

Nằm trong phạm vi $\bar{x} \pm 3s$ ước chừng có $99\% \sim 100\%$ dữ liệu.

Như vậy có ba trường hợp mà tôi đã nêu ở trên, điều này có nghĩa là gì? Ví dụ trong một lớp học, kết quả bài kiểm tra của học sinh có điểm trung bình là 50 điểm và độ lêch chuẩn là 10 điểm. Khi đó ta dự đoán có khoảng 67% học sinh có số điểm nằm trong phạm vi 50 ± 10 điểm.

Tóm lại, trong trường hợp này (điểm trung bình 50 điểm, độ lệch chuẩn là 10 điểm), có lẽ không có ai có điểm 0 và cũng không có ai có số điểm là 100.

Nó khá tiện lợi phải không? Tuy nhiên điều này không phải lúc nào cũng chính xác. Hãy thử nghiệm với các số ngẫu nhiên với đoạn code dưới đây.

```
import numpy as np
#Lay ngau nhien cac so co gia tri trong khoang 0~1
randoms = np.random.rand(1000)
mean = np.mean(randoms)
std = np.std(randoms)
#Dem xem co bao nhieu gia tri nam trong khoang tu Gia Tri Trung Binh cong tru Do Lech
    Chuan
count = 0
coef = 1
thresh = coef * std
for num in randoms:
if num > mean - thresh and num < mean + thresh:
count += 1
print('{}% of the numbers are included within mean ± {} std'.format(int(count/len(
    randoms)*100), coef))
```

Kết quả:

```
57% of the numbers are included within mean ± 1 std
```

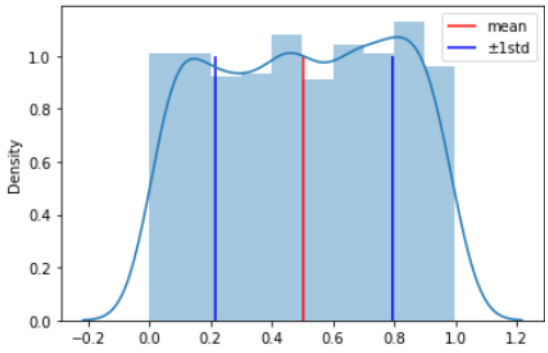
Chương trình trên không khó, các bạn hãy tự thử tự mình code xem sao nhé. `np.random.rand(1000)` sẽ tạo ra 1000 số có giá trị trong khoảng $0 \sim 1$, sau đó ta đếm xem trong phạm vi từ giá trị trung bình của các số ngẫu nhiên trên ± độ lệch chuẩn, có bao nhiêu số? Tôi nghĩ có thể viết code với hiệu suất cao hơn nhưng ở đây tôi viết code sao cho dễ hiểu nên tập trung vào việc sử dụng vòng lặp.

Dù chạy code nhiều lần thì có các kết quả khác nhau nhưng nhìn chung kết quả đều xoay quanh giá trị 57%.

Chúng ta thử vẽ đồ thị cho dễ hình dung, sau đây tôi sẽ sử dụng thư viện `seaborn` và `mathplotlib` và thực hiện code như sau:

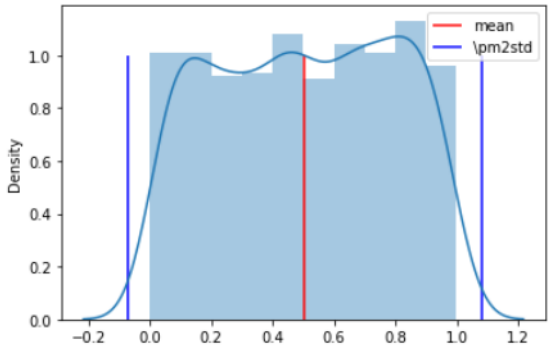
```
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

sns.distplot(randoms)
```

```
plt.vlines(mean, 0, 1, 'r', label='mean')
plt.vlines(mean+coef*std, 0, 1, 'b', label='±{}std'.format(coef))
plt.vlines(mean-coef*std, 0, 1, 'b')
plt.legend()
```



Nhìn vào biểu đồ trên ta thấy rằng dữ liệu nằm trong vùng hai đường kẻ màu xanh chiếm 57%. Biến số `coef` (hệ số của độ lệch chuẩn (coefficient) có tên viết tắt là coef) nếu thay đổi giá trị `coef=2` thì kết quả sẽ như thế nào, các bạn thử xem sao nhé. Chúng ta thấy rằng khi `coef=2` thì toàn bộ dữ liệu sẽ nằm trong hai đường kẻ màu xanh.

```
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
coef = 2
sns.distplot(randoms)
plt.vlines(mean, 0, 1, 'r', label='mean')
plt.vlines(mean+coef*std, 0, 1, 'b', label='\pm{}std'.format(coef))
plt.vlines(mean-coef*std, 0, 1, 'b')
plt.legend()
```



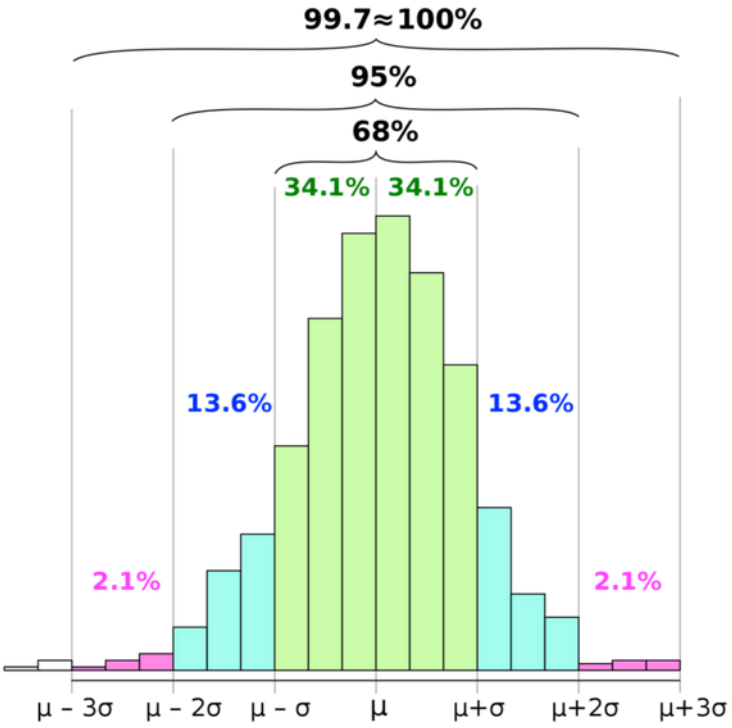
Như vậy giá trị cảm quan mà tôi nói ở trên trong trường hợp này không còn đúng nữa. Tôi đã nói rằng " Nằm trong phạm vi $\bar{x} \pm s$ ước chừng có $\frac{2}{3}$ dữ liệu". Và còn nói " Nằm trong phạm vi $\bar{x} \pm 2s$ ước chừng có 95% dữ liệu". Nhưng điều ấy đã không đúng trong ví dụ mà chúng ta vừa mới thực nghiệm.

Do đó hãy nhớ rằng giá trị cảm quan này không đúng hoàn toàn tùy thuộc vào sự phân bố của dữ liệu.

8.2 Nên nhớ về phân bố chuẩn

Rút cuộc sẽ chẳng có ý nghĩa gì nếu những điều chúng ta suy đoán thay đổi tùy thuộc vào phân bố của dữ liệu. Đây là điều mà có lẽ các bạn đang nghĩ tới. Nhưng không hẳn là không có ý nghĩa. Trước tiên cần biết rằng độ lệch chuẩn không phỉa là thứ hoàn mỹ để chúng ta có thể nắm rõ về dữ liệu. Tuy nhiên cái cảm giác rằng nằm trong phạm vi $\bar{x} \pm 3s$ ước chừng có $99\% \sim 100\%$ dữ liệu là rất tốt đấy chứ.
Điều thứ hai, đúng là nó không hoàn toàn tuân theo các giá trị cảm quan như đã thấy ở ví dụ thực nghiệm vừa qua, nhưng trong thống kê, việc giả định rằng dữ liệu phân bố như $\cdots$ vẫn thường xảy ra.Trong hầu hết các trường hợp chúng ta giả định một phân bố, phân bố giả định đó được gọi là phân bố chuẩn hay một số tài liệu khác gọi là phân phối chuẩn. Nếu dịch đúng nghĩa âm hán việt, thì người ta gọi phân bố chuẩn là phân bố chính quy.
Tôi sẽ viết một bài bình luận về phân bố chuẩn một lần nữa, nhưng nói tóm lại, đó là phân phối rất cơ bản, là một phân phối tiêu biểu trong thống kê học, nó rất quan trọng, bởi vì nhiều sự kiện tồn tại trong tự nhiên tuân theo phân bố chuẩn này.

Một phân bố được gọi là phân bố chuẩn nếu như có khoảng 68% dữ liệu nằm trong phạm vi giá trị trung bình ± độ lệch chuẩn. Và ước chừng có 95% dữ liệu nằm trong phạm vi giá trị trung bình ± 2 * độ lệch chuẩn. Và 99.7% dữ liệu sẽ nằm trong phạm vi giá trị trung bình ± 3 * độ lệch chuẩn.
Luật [68-95-99.7] đươc giới thiệu trong bài học số 9 của khóa học Python cho Data Science do diễn đàn tuhocvba.net tổ chức.



Quả nhiên để nhớ chi tiết những con số này thật là khó, nhưng cũng không cần thiết phải nhớ chi tiết như vậy. Bạn chỉ cần nhớ một cách đại khái [68%, 95%,khoảng toàn bộ dữ liệu]

là được rồi.

8.3 Với phân bố chuẩn, 95% dữ liệu nằm trong phạm vi Giá Trị Trung Bình ±1.96 *Độ Lệch Chuẩn

Tôi muốn các bạn hãy nhớ điều này, đó là với phân bố chuẩn, 95% dữ liệu nằm trong phạm vi Giá Trị Trung Bình ±1.96 *Độ Lệch Chuẩn.
Có bạn sẽ thắc mắc, vậy không phải là 95% dữ liệu sẽ nằm trong phạm vi Giá Trị Trung Bình± Độ Lệch Chuẩn hay sao?
Không phải 95% dữ liệu sẽ nằm vừa vặn trong phạm vi Giá Trị Trung Bình±Độ Lệch Chuẩn, theo wikipedia thì **ước chừng** có 95%, chính xác hơn phải nói là 95.4%. Nếu bạn có độ lệch chuẩn trong tay và tự hỏi rằng dữ liệu được phân bố như thế nào, khi đó chỉ cần nghĩ tới quy tắc $[68\%, 95\%, 99.7\%]$ thì cũng không sao, không phải lúc nào ta cũng có hứng thú quan tâm tới phân bố của dữ liệu, trong nhiều trường hợp chúng ta quan tâm đến đường ranh giới nào của phân phối chuẩn đạt 95%.
Vì vậy các bạn hãy nhớ lấy số 1.96 vì nó sẽ được sử dụng thường xuyên.

Bổ sung

Chúng ta có những đường ranh giới sử dụng hệ số khác ±1.96 nhưng bạn chỉ cần nhớ một hệ số, đối với các hệ số khác bạn có thể tra cứu. Tôi không nghĩ các bạn cần phải ghi nhớ. Trong thời đại này, nếu có thể tìm kiếm ra là được rồi. Tuy nhiên chúng ta không thể tìm hiểu những thứ mà chúng ta hoàn toàn không có khái niệm về nó. Do đó chỉ cần nhớ ±1.96 và 95% vì nó thường được sử dụng nhất.

8.4 Tổng kết

Lần này ta đã sử dụng độ lệch chuẩn và đã hiểu dữ liệu được phân bố như thế nào. Từ bây giờ khi có các thông tin này, chúng ta có thể hình dung được lượng dữ liệu bao nhiêu được phân bố như thế nào. Nó được kiểm tra bằng cách tính toán từ giá trị trung bình tới bao nhiêu lần độ lệch chuẩn, tương ứng với khoảng cách đó sẽ có bao nhiêu dữ liệu được phân bố. Điều này thật là tiện lợi phải không nào.

- Thông thường trong phạm vi $\bar{x} \pm s$ ước chừng có $\frac{2}{3}$ dữ liệu, trong phạm vi $\bar{x} \pm 2s$ ước chừng có 95% dữ liệu, trong phạm vi $\bar{x} \pm 3s$ ước chừng có $99\% \sim 100\%$ dữ liệu được phân bố.

- Điều này không phải lúc nào cũng đúng, nhưng có thể xem xét nó ở một mức độ nào đó.

- Trong phân bố chuẩn có luật $[68 - 95 - 99.7]$. Bạn không cần phải nhớ chi tiết các con số này.

- Đường ranh giới mà ở đó có 95% dữ liệu được phân bố là Giá Trị Trung Bình ±1.96*Độ lệch chuẩn. Hệ số 1.96 được sử dụng nhiều trong thống kê học, hãy ghi nhớ lấy nó.

Trong thống kê, không có nhiều con số để nhớ, nhưng 1.96 là một trong số ít những con số đáng nhớ, vì vậy hãy ghi nhớ nó.
Các bạn có thể vận dụng lý thuyết này để kiểm chứng rất nhiều vấn đề trong thực tế. Chẳng hạn trong một trường học có bài kiểm tra tiếng anh, và một trường học khác cũng có bài kiểm tra tiếng anh. Nếu chúng ta chỉ nhìn vào điểm số thì không thể so sánh thành tích của trường nào tốt hơn. Tuy nhiên nếu sử dụng độ lệch chuẩn, và ở mỗi trường chúng ta tính toán các vị trí, từ đó có thể so sánh tương đối thành tích học sinh của mỗi trường với nhau. Tất nhiên mỗi trường thì trình độ của các học sinh không giống nhau hoàn toàn, do đó cũng không thể nói đây là phép so sánh hoàn mỹ nhưng vẫn có thể mang một ý nghĩa nào đó.
Những người có thể ứng dụng điều này đương nhiên đã biết về độ lệch (Thiên Sai Trị). Số liệu thống kê được sử dụng rất nhiều trong cuộc sống xung quanh ta.

Bài 9

Rất quan trọng! Chuẩn hóa và trị số lệch là gì? Tính điểm z và tính điểm T

Ở bài học trước ta đã sử dụng độ lệch chuẩn khi nói về mức độ phân bố, đây là một khái niệm vô cùng quan trọng vì nó giúp chúng ta phán đoán được ở trong phạm vi nào đó sẽ có bao nhiêu dữ liệu được phân bố. Chúng ta có thể so sánh các nhóm dữ liệu khác nhau khi sử dụng trung bình cộng và độ lệch chuẩn. Ví dụ chúng ta có hai ngôi trường cùng cho học sinh làm bài kiểm tra. Nếu chỉ nhìn vào điểm số, chúng ta không thể so sánh kết quả trường nào thì tốt hơn. Khi sử dụng trung bình cộng và độ lệch chuẩn, chúng ta có thể tính toán và so sánh thành tích điểm số ở một vị trí nào đó trong bảng phân bố dữ liệu.
Trong bài học này, chúng ta sẽ có những khái niệm rất quan trọng và tôi muốn các bạn sẽ ghi nhớ.

9.1 So sánh giữa các nhóm có dữ liệu khác nhau bằng cách tính điểm z.

Tính điểm z là gì? Hẳn là điều các bạn đang thắc mắc.
Trong bài học trước, ta có thể xác định vị trí của dữ liệu nằm trong phạm vi nào, từ độ trung bình tới bao nhiêu lần độ lệch chuẩn, từ đó có thể biết được phạm vi đó có mức độ phân bố dữ liệu là bao nhiêu phần trăm.
Và như tôi đã nói, lý thuyết này rất quan trọng, nó giúp chúng ta có thể so sánh hai dữ liệu khác nhau. Ví dụ, bạn Bình ở trường A có điểm tiếng anh là 40, bạn An ở trường B có điểm tiếng anh là 60. Nếu chỉ nhìn vào điểm kiểm tra như vậy thì không thể so sánh được mức độ thành thạo tiếng anh. Chẳng hạn trường A ra bài kiểm tra quá khó, mặc dù Bình được 40 điểm nhưng có thể đó là học sinh tốp đầu của trường A.
Vậy thì ta phải tính toán với mỗi bạn Bình và bạn An thì các bạn ấy phải tính toán điểm của mình nằm trong phạm vi nào từ giá trị trung bình tới bao nhiêu lần độ lệch chuẩn thì các bạn ấy mới biết trình độ của mình ở mức nào trong trường mình.

Từ vấn đề đặt ra đó, chúng ta có thể đề xuất công thức sau đây, ta gọi nó là công thức tính điểm z.

$$z = \frac{x - \bar{x}}{s}$$

trong đó $\bar{x}$ là giá trị trung bình, s là độ lệch chuẩn.
Ví dụ, chúng ta có hai lớp học và có điểm môn thi Tiếng Anh và môn Toán như sau:

Lớp 1	A	B	C	D	E	Trung Bình	Độ lệch chuẩn
Tiếng Anh	40	30	80	70	69	56	18.5472
Lớp 2	F	G	H	I	J	-	-
Toán	30	50	40	30	20	34	10.198

Vậy câu hỏi đặt ra, ai là người giỏi nhất? Nếu nhìn vào điểm số có lẽ các bạn sẽ khẳng định C là người giỏi nhất, thế nhưng điều đó có đúng hay không?
Dữ liệu khác nhau mà so sánh thông thường sẽ không có ý nghĩa gì. Vậy thì muốn so sánh ta phải chuẩn hóa.
Chuẩn hóa ở đây có nghĩa là ta phải thực hiện biến đổi để dữ liệu có giá trị trung bình là 0 và độ lệch chuẩn là 1.
Công thức ở trên có nghĩa là: Điểm mới = Điểm thực tế−Điểm trung bình÷Độ lệch chuẩn
Như vậy áp dụng công thức trên ta sẽ có dữ liệu mới sau khi chuẩn hóa như sau:

Lớp 1	A	B	C	D	E
Tiếng Anh	−0.8626	−1.4018	1.2939	0.7548	0.2156
Lớp 2	F	G	H	I	J
Toán	−0.39223	1.5689	0.5883	−0.3922	−1.3728

Các bạn cũng có thể tính toán bằng code Python như sau:

```
import numpy as np
data_tienganh = [40,30,80,70,60]
mean_tienganh = np.mean(data_tienganh)
std = np.std(data_tienganh)
#Chuan hoa
z = (data_tienganh - mean_tienganh) / std
print('standardized data tieng anh(z): {}'.format(z))
print('mean mean_tienganh: {:.2f}'.format(np.mean(z)))
print('std mean_tienganh: {}'.format(np.std(z)))
```

Kết quả cũng tương tự như trên:

```
standardized data tieng anh(z): [-0.86266219 -1.40182605  1.29399328  0.75482941
    0.21566555]
mean mean_tienganh: -0.00
std mean_tienganh: 1.0
```

Như vậy ta thấy $C = -1.29$ trong khi đó $G = 1.56$, cho nên so sánh một cách tương đối ta thấy rằng G là người đạt điểm cao nhất.

Chuẩn hóa (standardize) như thế nào?

Mục tiêu chuẩn hóa là cố gắng làm sao để dữ liệu có giá trị trung bình là 0.
Và một mục tiêu nữa đó là làm sao để độ phân tán (đồng thời cả độ lệch chuẩn) có giá

trị là 1.

▌Tính chính xác ra sao?

Điều lo ngại là mức độ tin cậy của dữ liệu sau khi đã được chuẩn hóa. Thật sự thì cách so sánh nói trên có ổn không?

Hãy cẩn thận trong những trường hợp sau đây.

▌Quy mô dữ liệu thống kê quá nhỏ

Đương nhiên nhưng vẫn cần nói, nếu mẫu số quá nhỏ thì sẽ không còn tính chính xác.

Chẳng hạn ta có ba người A có số điểm 20, B có điểm 50, C có điểm 80.

Giá trị trung bình là 50.

Trong trường hợp này thì điểm số chuẩn hóa của C trở nên ít đi đáng kể. Giả sử bài kiểm tra này là rất khó, thậm chí B và C là những chuyên gia trong lĩnh vực này.

Khi đó nếu so sánh với kết quả của những người làm bài kiểm tra thông thường, thậm chí C còn bị đánh giá là kém.

Khi quy mô dữ liệu thống kê quá ít thì các giá trị cực đoan sẽ xuất hiện.

▌Không phải là phân bố chuẩn

Giả sử ta có 3 người đạt số điểm là 0, và có 3 người đạt số điểm là 100. Trong trường hợp này điểm trung bình cũng là 50. Với thí nghiệm như thế này ta cũng thấy rằng việc chuẩn hóa cũng không có ý nghĩa gì. Các phân bố mà dữ liệu tập trung vào hai đầu mút, hoặc quá lớn hay quá bé, thì độ chính xác cũng không có.

Chuẩn hóa được sử dụng rất nhiều trong Machine Learning. Có nhiều trường hợp không thể xử lý được nếu dữ liệu để nguyên như ban đầu mà không thực hiện chuẩn hóa. Khi đào tạo một mô hình bằng Machine Learning, dữ liệu thường được chuẩn hóa trước đó, ta gọi là tiền xử lý (preprocessing).

```
from sklearn.preprocessing import StandardScaler
data = np.array([30,50,40,30,20])
print('data shape: {}'.format(data.shape))
data = np.expand_dims(data, axis=-1)
print('reshaped data shape: {}'.format(data.shape))
# Instance creation
scaler = StandardScaler()
# Doi so cho fit_transform phai la mot mang hai chieu
scaled = scaler.fit_transform(data)
print(scaled)
```

Kết quả:

```
data shape: (5,)
reshaped data shape: (5, 1)
[[-0.39223227]
[ 1.56892908]
[ 0.58834841]
[-0.39223227]
[-1.37281295]]
```

Cách tính ở đây cũng sẽ đưa ra kết quả tính toán giống với code đã giới thiệu ở trên.

Lần này tôi đã sử dụng `sklearn.preprocessing.StandardScaler` , nó được gọi là một lớp (Class) và được viết theo cách hướng đối tượng.

`scaler = StandardScaler()` khi được thực thi, nó sẽ tạo ra các mô hình (instance) dựa

trên Class `StandardScaler` .
Có lẽ các bạn chưa quen với hướng đối tượng, nói một cách đơn giản, từ cái sơ đồ thiết kế đó sẽ tạo ra một thứ (instance) được gọi là `scaler`. Dương nhiên cái `scaler` này có chức năng chuẩn hóa bằng cách gọi hàm `.fit_transform()`. Tôi nghĩ code không khó, các bạn hãy làm quen với code như vậy.
`.fit_transform()` sẽ chuẩn hóa đối số đầu vào mà mảng `ndarray`. Và vì nó chỉ tiếp nhận mảng hai chiều cho nên trước đó chúng ta thông qua `np.expand_dims()` để chuyển thành mảng hai chiều. Về điều này đã được nêu chi tiết trong bài học số 9 của khóa học Data Science. Bạn có thể vừa xem tài liệu tham khảo vừa tiếp tục theo dõi bài học này. Bạn cũng không cần phải nhớ, vì khi thực thi sẽ xuất hiện lỗi do đầu vào không phải là mảng hai chiều, và khi đó bạn sẽ phải xoay sở để chuyển mảng một chiều thành mảng hai chiều, bằng cách này hay cách khác bạn cũng sẽ khắc phục được vì google sẽ giúp bạn tìm ra giải pháp. `sklearn.preprocessing.StandardScaler` được sử dụng rất nhiều, từ giờ các bạn hãy nhớ nó nhé.

9.2 Trị số lệch là gì?

Kết quả kỳ thi hôm trước thế nào?

So với lần trước, điểm tiếng anh cao hơn 10 điểm.

Chúc mừng nhé. Trị số lệch thì thế nào?

À, so với kỳ thi trước thì giảm một chút.

Tôi thì điểm toán giảm nhưng trị số lệch thì tăng. Vậy thì sẽ thế nào nhỉ?

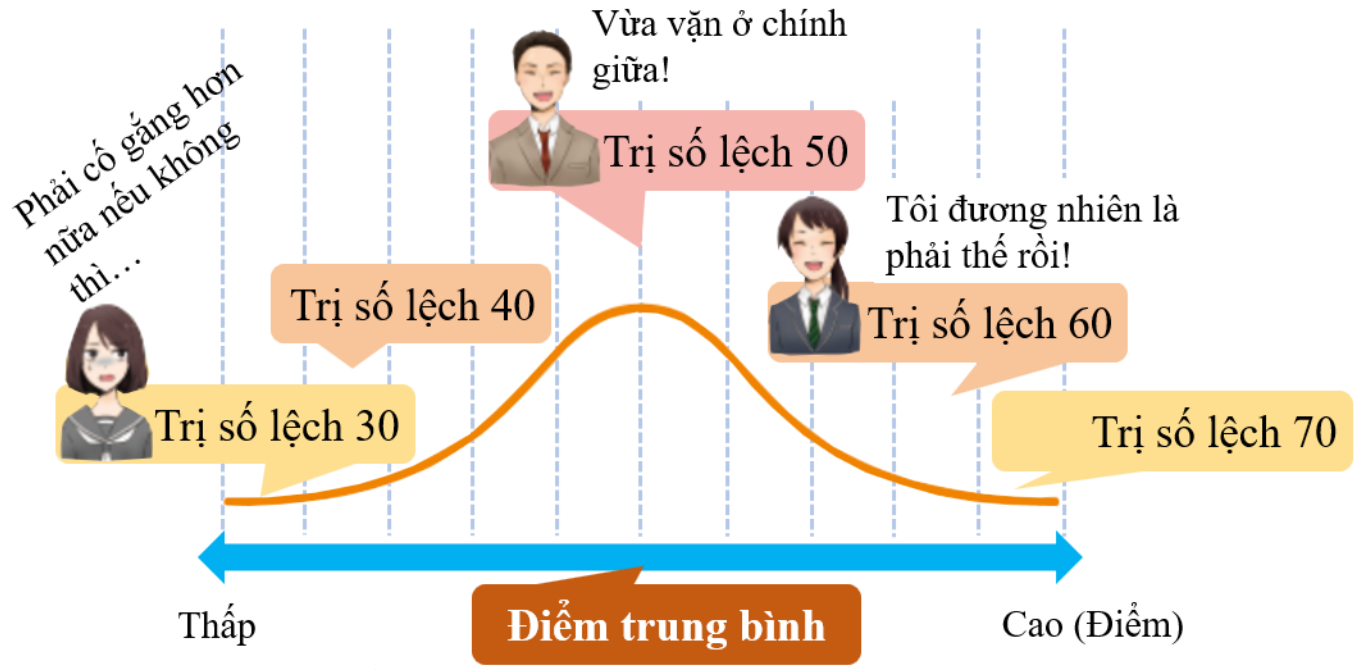
Trị số lệch (giá trị độ lệch) có liên quan rất lớn đến điểm trung bình của bài kiểm tra và sự phân bố điểm của tất cả các thí sinh tham dự kỳ thi. Vì điểm trung bình được biểu thị bằng 50 nên nó phụ thuộc vào phân bố điểm của toàn bộ nhóm.

Phức tạp quá nhỉ!

Hãy cùng suy nghĩ về ý nghĩa của trị số lệch và cách sử dụng nó.

Ví dụ, có hai kỳ thi diễn ra và cả hai kỳ thi tôi đều có 80 điểm. Tuy nhiên mức độ khó của mỗi kỳ thi là khác nhau, do đó điểm số trung bình của mỗi kỳ thi cũng khác nhau. Vậy thì kết quả của kỳ thi lần một và của lần hai, kết quả nào mới là tốt khi mà tôi có cùng số điểm là 80 ? Rõ ràng nếu chỉ dựa vào điểm số thì chúng ta không thể đưa ra phán đoán chính xác. Khi đó, chúng ta cần tới một khái niệm mới, đó là **trị số lệch**. Bằng cách biến đổi dữ liệu để giá trị trung bình có giá trị là 50 và có độ lệch chuẩn là

10. Điểm số mới sau khi đi qua bộ biến đổi như vậy gọi là **trị số lệch**. Trị số lệch càng cao có nghĩa rằng điểm số của tôi có thứ hạng cao. Và ngược lại, trị số lệch thấp có ngĩa rằng thứ hạng của tôi trong số những người tham dự kỳ thi là thứ hạng thấp.



Trị số lệch được tính như thế nào?

Bước 1: Chuyển điểm số về cách tính điểm z. Cách tính này sẽ tạo ra điểm số mới có giá trị trung bình là 0 và có độ lệch chuẩn là 1.

Bước 2: Trị số lệch = Điểm z $*$ 10 + 50 Như vậy điểm số mới chính là trị số lệch, chúng sẽ có giá trị trung bình là 50 và có độ lệch chuẩn là 10.

```
from sklearn.preprocessing import StandardScaler
data = np.array([30,50,40,30,20])
print('data shape: {}'.format(data.shape))
data = np.expand_dims(data, axis=-1)
print('reshaped data shape: {}'.format(data.shape))
# Instance creation
scaler = StandardScaler()
# Doi so cho fit_transform phai la mot mang hai chieu
scaled = scaler.fit_transform(data)
# Tinh diem T
scaled = scaled*10+50
print(f"mean of scaled {np.mean(scaled)}")
print(f"std of scaled {np.std(scaled)}")
print(scaled)
```

Kết quả:

```
data shape: (5,)
reshaped data shape: (5, 1)
mean of scaled 49.99999999999999
std of scaled 9.999999999999998
[[46.0776773 ]
[65.68929081]
[55.88348405]
[46.0776773 ]
[36.27187054]]
```

Giá trị trung bình khi tính sang trị số lệch sẽ là bao nhiêu? Trị số lệch có giá trị cao nhất là bao nhiêu?

Do việc chuẩn hóa để tính điểm T là làm sao để giá trị trung bình là 50 do đó nếu điểm của bạn ngoài thực tế có giá trị bằng số điểm trung bình thì khi chuyển sang cách tính điểm T, trị số lệch của bạn khi đó nhất định sẽ là 50.

Ngoài ra trị số lệch ở một kỳ thi thông thường sẽ nằm trong phạm vi từ $25 \sim 75$ nhưng khi tính toán sang điểm T, vẫn có trường hợp trị số lệch âm hoặc trị số lệch lớn hơn 100. Tôi lấy ví dụ trong một kỳ thi có 100 người dự thi trong đó 99 người có số điểm là 0, và chỉ có một người có số điểm là 100, khi đó nếu tính điểm T thì những người điểm 0 sẽ có trị số lệch là -49 và người có điểm 100 có trị số lệch là 149.5.

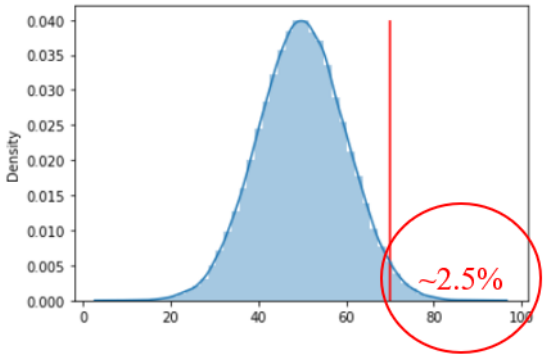
Những trường hợp cực đoan, khi mà dữ liệu tập trung phân bố vào hai đầu mút, hoặc nhỏ nhất, hoặc lớn nhất, trường hợp như thế thực sự là không thể suy nghĩ được, và kết quả khi tính trị số lệch cũng không có ý nghĩa gì.

Cách xem trị số lệch và tỷ lệ của nó

Bảng dưới đây mô tả mối quan hệ về thứ hạng với trị số lệch. Nếu trị số lệch là 75 thì nó ở vị trí 0.62% tính từ trên xuống. Điều đó tức là nếu có 1000 người dự thi thì nó sẽ có thức hạng trước hoặc sau thứ hạng thứ 6. Nếu trị số lệch là 35 thì thứ hạng tính từ trên xuống là 93.32% (tính từ dưới lên sẽ là 6.68%), tức là thứ hạng tính từ cao xuống thấp, nó sẽ ở vị trí trước hoặc sau thứ hạng thứ 933. Lợi điểm của trị số lệch là bạn có thể biết được vị trí năng lực học tập của mình trong nhóm bất kể điểm số hay thứ hạng.

Trị số lệch	Tỷ lệ từ trên xuống	Thứ hạng trong 1000 người
80	0.13%	Vị trí 1.3
75	0.62%	Vị trí 6.2
70	2.28%	Vị trí 22.8
65	6.68%	Vị trí 66.8
60	15.87%	Vị trí 158.7
55	30.85%	Vị trí 308.5
50	50.00%	Vị trí 500.0
45	69.15%	Vị trí 691.5
40	84.13%	Vị trí 841.3
35	93.32%	Vị trí 933.2
30	97.72%	Vị trí 977.2

Trong bài học trước tôi cũng đã nói rằng trong phạm vi Giá trị trung bình$\pm 2*$Độ lệch chuẩn sẽ có 95% dữ liệu, như vậy có thể tính toán được rằng trị số lệch 70 (khi tính sang điểm T thì độ lệch chuẩn là 10 do đó phạm vi dữ liệu lúc này nằm từ $30 \sim 70$) sẽ có khoảng 2.5% dữ liệu, bởi vì $100\% - 95\% = 5\%$, và vì bảng phân bố chuẩn đối xứng trái phải nên ta có con số 2.5%. Tức là người có trị số lệch 70 sẽ nằm ở tốp 25 người cao nhất nếu có 1000 người dự thi. Điều này cũng phù hợp với bảng trên.



Hình vẽ trên các bạn có thể vẽ bằng code như sau:

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

samples = np.random.randn(100000)
tscore = samples*10 + 50
sns.distplot(tscore)
plt.vlines(70, 0,0.04, 'r')
```

Lợi điểm sử dụng trị số lệch

Thầy ơi, kết quả kỳ thi lần này, môn Toán em được 70 điểm và môn Tiếng Anh em được 90 điểm. Môn Toán khó nên điểm không tốt, tuy nhiên môn Tiếng Anh được điểm cao nên em rất vui.

Chúc mừng em, vậy trị số lệch thì thế nào?

Cả hai môn đều có trị số lệch là 62.2 ạ.

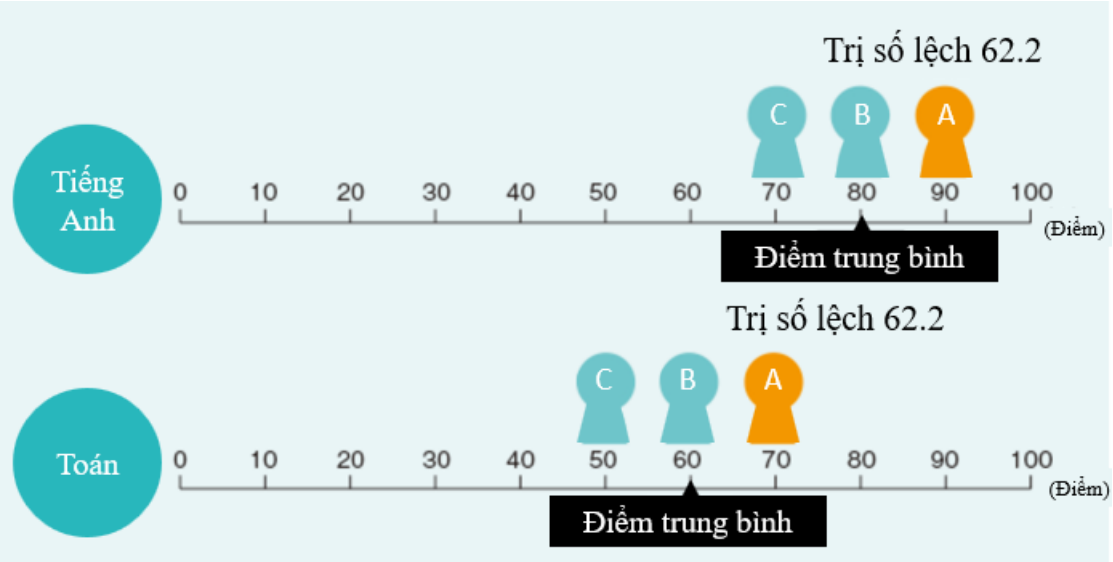
Môn Toán quả nhiên là khó nên điểm thấp hơn nhỉ.

Dù kỳ thi có điểm trung bình khác nhau, tuy nhiên sau khi chuẩn hóa và tính trị số lệch, chúng ta có thể so sánh điểm số của hai kỳ thi khác nhau. Đây là một lợi điểm của trị số lệch. Giống như bạn học sinh ở trên, tuy điểm Tiếng Anh và điểm Toán khác nhau, nhưng khi tính toán trị số lệch thì đều có giá trị là 62.2.
Để dễ hình dung ta hãy xem bạn học sinh A:

Người dự thi	Tiếng Anh	Toán
A	90	70
B	80	60
C	70	50
Điểm trung bình	80	60
Trị số lệch của A	62.2	62.2

Điểm trung bình của môn Tiếng Anh là 80 và điểm trung bình của môn Toán là 60, có thể thấy vì môn Toán khó hơn môn Tiếng Anh, nên trị số lệch của cả hai môn thi đều có giá trị như nhau mặc dù điểm thi của bạn A ở môn Tiếng Anh là cao hơn môn Toán. Xét

về mặt thứ hạng thì không thay đổi. Như vậy có thể thấy, chúng ta không chỉ so sánh được điểm số hay thứ hạng trong một kỳ thi, mà còn có thể so sánh điểm của hai kỳ thi khác nhau.



Chú ý đến sự thay đổi trong điểm số

Để đánh giá kết quả bài thi một cách công bằng hơn, cần xem xét không chỉ sự chênh lệch so với điểm trung bình, mà cần quan tâm cả sự phân bố của điểm số. Ví dụ dưới đây là tóm tắt kết quả của năm người đã làm bài kiểm tra Tiếng Anh và Toán.

Người dự thi	Tiếng Anh	Toán
D	70	70
E	36	16
F	50	38
G	64	82
H	80	94
Điểm trung bình	60	60
Độ lệch chuẩn	15.4	28.8
Trị số lệch của D	56.5	53.5

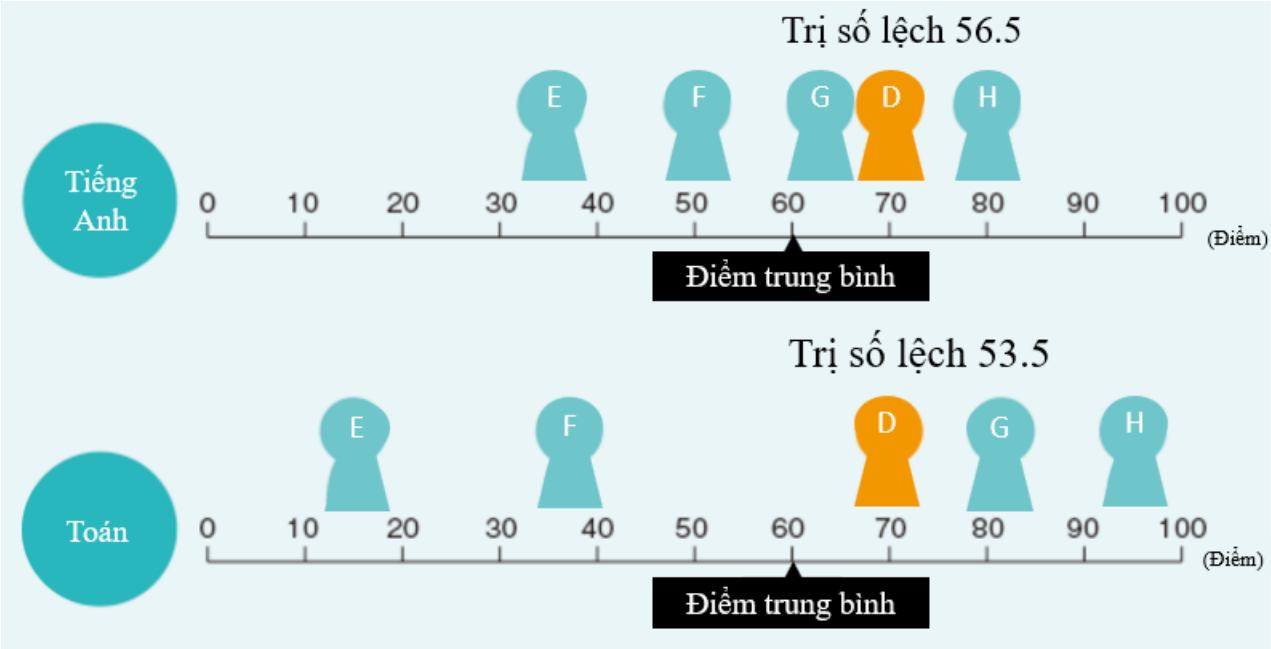
Điểm của bạn D ở cả môn Toán và Tiếng Anh đều là 70 điểm, và điểm trung bình của cả Toán và Tiếng Anh đều là 60 điểm. Tuy nhiên trong môn Tiếng Anh, 5 người có điểm số rất gần điểm số trung bình. Trong khi đó với môn Toán thì điểm của 5 người khác nhau rất nhiều.

Trong một kỳ thi nếu như phần lớn mọi người đều có số điểm tập trung quanh điểm số trung bình với mật độ dày đặc thì điều đó chứng tỏ việc có được điểm số vượt qua con số trung bình này là một việc khó khăn. Do đó nếu ai đó có điểm cao hơn mức trung bình thì trị số lệch sẽ trở nên rất lớn.

Ngược lại trong một kỳ thi mà điểm thi của nhiều người cách xa mốc điểm trung bình, khi đó dù điểm số cao hơn mức điểm trung bình thì trị số lệch hầu như không gia tăng.

Do đó để đánh giá công bằng thành tích trong một kỳ thi, cần xem xét cả hai yếu tố, độ lệch so với điểm số trung bình, và phân bố điểm.

Do đó, D có điểm số giống nhau trong cả hai kỳ thi mà các kỳ thi này đều có điểm trung bình giống nhau, tuy nhiên trị số lệch lại có sự khác nhau.



9.3 Tổng kết

Lần này tôi đã giải thích cho các bạn về cách tính điểm z và trị số lệch.
Nhờ tính toán từ điểm dữ liệu tới giá trị trung bình, khoảng cách đó bằng bao nhiêu lần độ lệch chuẩn, chúng ta có thể biết được phân bố của toàn bộ dữ liệu và vận dụng nó để tính được vị trí thứ hạng của điểm dữ liệu trong toàn bộ tổng thể dữ liệu, thông qua đó ta có thể so sánh được hai dữ liệu khác nhau cho dù chúng có độ trung bình cũng như độ lệch chuẩn khác nhau. Để có thể thực hiện được việc so sánh như thế chúng ta cần một bước biến đổi dữ liệu, công việc đó gọi là **tiêu chuẩn hóa** hoặc **chuẩn hóa**.

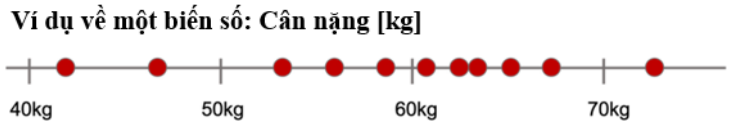
- Chuẩn hóa là biến đổi dữ liệu để giá trị trung bình là 0 và độ lệch chuẩn là 1, ta gọi đây là cách tính điểm z.
- Công thức chuẩn hóa là $z = \frac{x - \bar{x}}{s}$.
- Trị số lệch là giá trị mới của dữ liệu sau khi biến đổi dữ liệu ban đầu sao cho giá trị trung bình là 50 và có độ lệch chuẩn là 10.

Trị số lệch trong bài này chỉ mang tính giới thiệu, sau này trong thống kê học hay trong máy học thì nó không thực sự quan trọng.
Về cơ bản các bạn hãy nắm rõ cách tính điểm z. Từ công thức này ta có thể biến đổi dữ liệu sao cho nó có giá trị trung bình cũng như độ lệch chuẩn tùy ý mà ta mong muốn.

Bài 10

Hãy hiểu về phân tán chung

Khóa học thống kê nhập môn đã tới bài học thứ 10 và chưa có dấu hiệu sẽ kết thúc. Tôi xin thuyết minh trình tự một cách đơn giản cho tới thời điểm này. Ở Bài học 1, chúng ta nói về thống kê mô tả và thống kê suy luận, và cho tới bây giờ chúng ta vẫn đang nói về thống kê mô tả.
Với tư cách là giá trị đại diện, ở Bài học 2, chúng ta nói về giá trị trung bình, ở Bài học 3 chúng ta nói về trung vị và tối tần trị. Với tư cách mô tả độ phân tán phân bố, ta có khái niệm phạm vi, IQR, QD được nêu ở Bài học 4. Từ trung bình cộng độ lệch chúng ta có độ phân tán và độ lệch chuẩn được nêu ở Bài học 5, phân tán bất thiên được nêu ở Bài học 6. Cho tới thời điểm này, chúng ta đã có khá nhiều thứ.
Tuy nhiên những cái đó là trong thống kê mô tả cho một biến số.
Ví dụ, trong một lớp học, có điểm môn Tiếng Anh, hay là tập hợp thông tin chiều cao của mỗi người, những thông tin đó chỉ là một loại thông tin và nó là một biến số, chúng ta có thể tính được độ phân tán hay trung bình cộng của những thông tin đó.



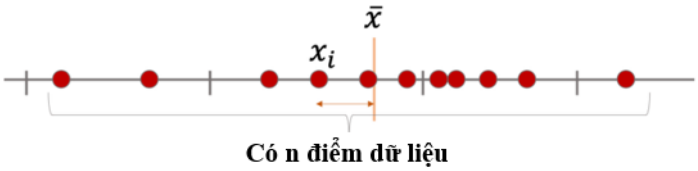
Tuy nhiên thực tế trong khoa học dữ liệu, chúng ta không mấy khi chỉ xử lý dữ liệu có một biến số, thông thường là phải xử lý với nhiều biến số. Chẳng hạn chúng ta có 3 biến số là cân nặng, chiều cao, tuổi.
Trong bài học này tôi sẽ giải thích về phân tán chung trong thống kê mô tả trong trường hợp có hai biến số. Nó thực sự không khó nên các bạn hãy nhớ nó nhé.

10.1 Phân tán chung là phiên bản phân tán cho hai biến số

Có thể nói rằng phân tán chung (covariance) là một phiên bản phân tán cho hai biến số.

Ta nhắc lại trong trường hợp một biến số, độ phân tán thông thường được tính là:

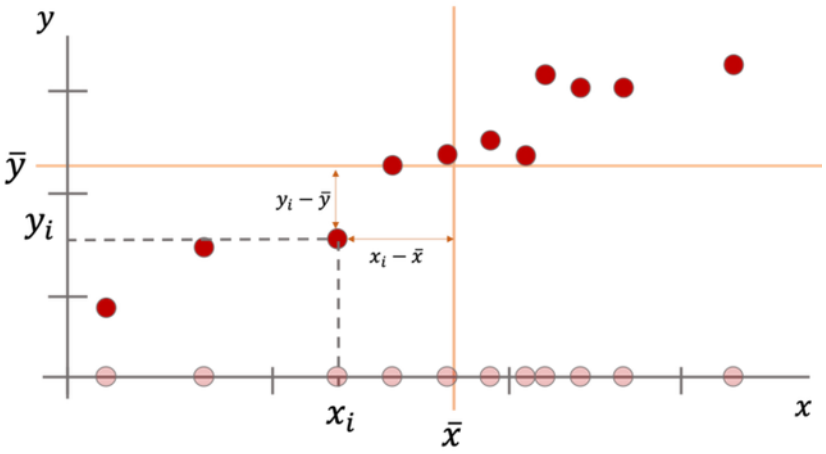
$$s^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$$



Công thức trên có thể viết lại thành:

$$s^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})$$

Giả sử như dữ liệu của chúng ta không chỉ có biến x mà còn có cả biến y thì công thức tính phân tán sẽ được tính như thế nào?
Ví dụ, dữ liệu dưới đây không chỉ có thông tin trọng lượng mà còn có cả thông tin chiều cao. Và nó được biểu diễn như biểu đồ dưới đây.



Khi chỉ có một biến thì ta có độ lệch từ điểm dữ liệu tới giá trị trung bình $\bar{x}$.Khi có hai biến số (x, y) ta sẽ có độ lệch từ điểm dữ liệu tới giá trị trung bình là cặp số $(x - \bar{x}), (y - \bar{y})$. Tích các cặp số như vậy ứng với mỗi điểm dữ liệu cộng lại với nhau rồi lấy trung bình cộng ta sẽ có **phân tán chung (covariance)**, thông thường được ký hiệu là s_{xy}.

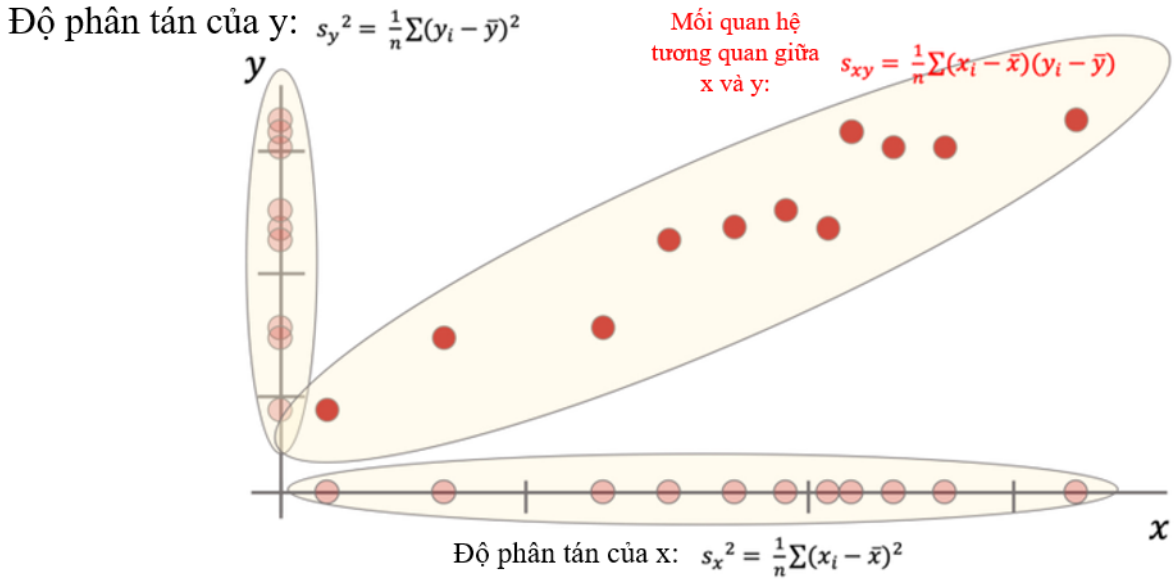
$$s_{xy} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

Phân tán chung còn có tên gọi khác là *hiệp phương sai*. Cái tên *hiệp phương sai* không có chú thích gì thêm, tôi nghĩ chữ "*hiệp*" là trong từ hiệp lực, chung sức. Chữ "*phương*" xuất phát từ khái niệm phương sai (phân tán) do công thức này có biểu thức bình phương để triệt tiêu dấu âm nếu có, và chữ "*sai*" trong từ sai lệch, sai khác, ý nghĩa là độ lệch.

10.2 Phân tán chung là chỉ số nói về mối tương quan giữa hai biến

> Tôi đã hiểu về sự tồn tại của phân tán chung, nhưng rút cuộc nó là gì vậy nhỉ?

Đây là nghi vấn lớn nhất phải không nào. Ở trên chúng ta đi từ công thức phân tán thông thường và suy ra công thức phân tán chung. Nhưng rút cuộc nó là gì, hẳn là nghi vấn mà mọi người đang nghĩ tới.
Nói một cách đơn giản, phân tán chung là chỉ số biểu thị mối tương quan giữa hai biến số.
Ồ, chẳng phải nó biểu thị mức độ phân tán của dữ liệu hay sao? Không, nếu nói về độ phân tán thì chúng ta có thể nhìn vào phân tán trên mỗi trục tọa độ là đủ rồi. Hãy thử nhìn vào biểu đồ dưới.



Để biết phân tán của x, y ta dựa vào ${s_x}^2, {s_y}^2$ trên mỗi trục tọa độ. Vậy thì phân tán chung **biểu thị mối quan hệ** giữa x, y như thế nào?
Ví dụ, chiều cao càng lớn thì cân nặng của người ấy cũng lớn, hoặc điểm Tiếng Anh mà cao thì có khuynh hướng là điểm môn Tiếng Việt cũng sẽ cao. Phân tán chung chính là biểu thị mối tương quan như thế.

10.3 Tương quan thuận, tương quan nghịch và không tương quan

Trong mối tương quan có tương quan thuận và tương quan nghịch.
Giá trị này càng lớn thì giá trị khác cũng có xu hướng lớn theo, đó là mối **tương quan thuận**.
Ngược lại, giá trị này càng lớn thì giá trị khác có xu hướng nhỏ đi, đó là mối **tương quan**

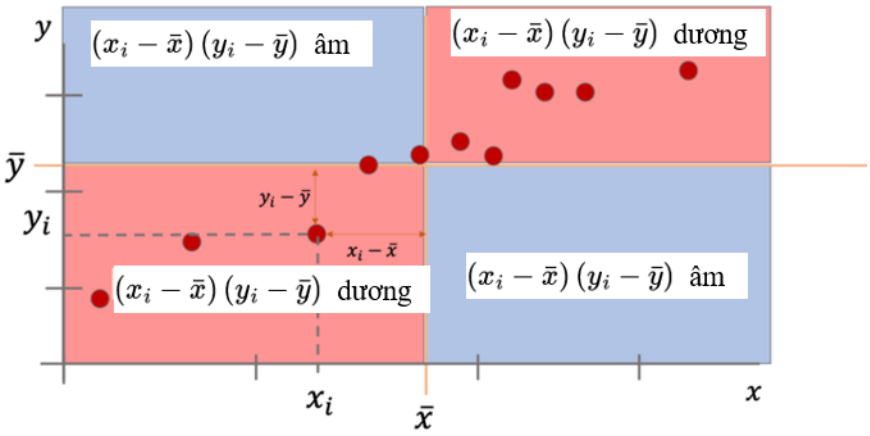
nghịch.

Trong mối tương quan thuận, phân tán chung sẽ có giá trị dương. Và trong mối tương quan nghịch, phân tán chung sẽ có giá trị âm. Trường không không tương quan thì phân tán chung có giá trị la 0.

Như vậy chúng ta thử suy nghĩ xem khi nào thì tích $(x_i - \bar{x})(y_i - \bar{y})$ dương, khi nào thì âm.

Trường hợp âm, đó là khi $(x_i - \bar{x})$ và $(y_i - \bar{y})$ có một số hạng âm. Nếu x_i nhỏ hơn $\bar{x}$ thì y_i lớn hơn $\bar{y}$, hoặc ngược lại.

Trường hợp dương, đó là khi $(x_i - \bar{x})$ và $(y_i - \bar{y})$ cùng dương hoặc cùng âm.



Như vậy tổng các tích $(x_i - \bar{x})(y_i - \bar{y})$ trong phần màu xanh mà lớn thì kết quả cuối cùng, phân tán chung sẽ có giá trị âm. Ngược lại, phần màu đỏ cho kết quả lớn, thì phân tán chung sẽ có giá trị dương. Điều này thật đơn giản phải không nào.

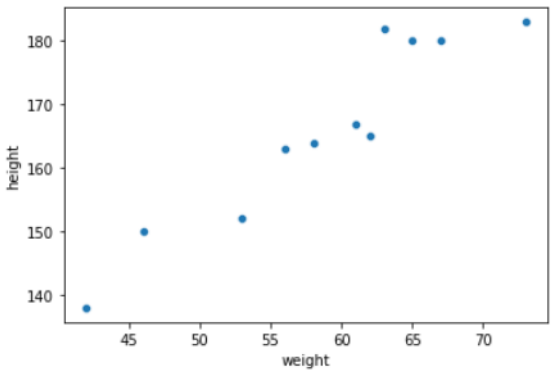
Vậy thì không tương quan là trường hợp như thế nao? Đó là khi phần màu xanh và phần màu đỏ giống nhau và chúng triệt tiêu lẫn nhau khi chúng ta tính tổng các tích $(x_i - \bar{x})(y_i - \bar{y})$, cuối cùng phân tán chung sẽ có giá trị là 0.

Đến đây có lẽ các bạn đã lý giải được công thức của phân tán chung và lý giải được ý nghĩa của nó. Trong bài học tiếp theo tôi sẽ trình bày về mối quan hệ tương quan khi thể hiện độ mạnh yếu của mối tương quan.

10.4 Tính phân tán chung bằng Python

Chúng ta có thể sử dụng hàm `.cov()` của `NumPy` hay `Pandas` để tính phân tán chung. Trước hết hãy nhìn phân bố dữ liệu dưới đây.

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

weight = np.array([42, 46, 53, 56, 58, 61, 62, 63, 65, 67, 73])
height = np.array([138, 150, 152, 163, 164, 167, 165, 182, 180, 180, 183])
sns.scatterplot(weight, height)
plt.xlabel('weight')
plt.ylabel('height')
```



Đầu tiên chúng ta sử dụng `np.cov()` để tính phân tán chung.

```
np.cov(weight, height)
```

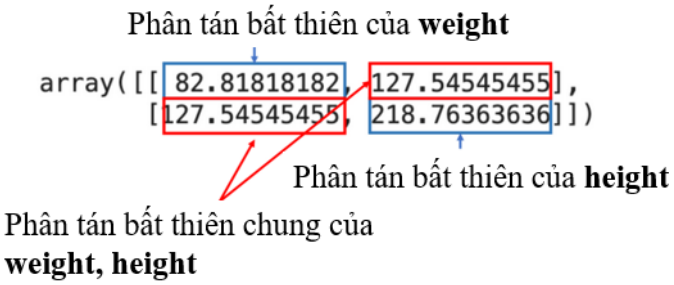
Kết quả:

```
array([[ 82.81818182, 127.54545455],
[127.54545455, 218.76363636]])
```

Nó trả về cho chúng ta một ma trận. Ma trận này gọi là **ma trận phân tán chung**. (Các tài liệu tiếng việt hiện hành gọi là *ma trận hiệp phương sai*). Có tên tiếng anh là **variance-covariance matrix**.
Không có gì là khó cả. Ví dụ biến số lần này giống như cân nặng và chiều cao (weight, height) là $x_1, x_2, x_3, \cdots x_i$. Khi đó ma trận phân tán chung được biểu diễn như dưới đây.

$$\begin{bmatrix} s_1^2 & s_{12} & \cdots & s_{1i} \\ s_2^2 & s_2 & \cdots & s_{2i} \\ \cdots & \cdots & \cdots & \cdots \\ s_{i1} & s_{i2} & \cdots & s_i^2 \end{bmatrix}$$

Số hạng có chỉ số ii được biểu diễn ngắn gọn là s_i^2, số hạng có chỉ số ij được biểu diễn là s_{ij}.
Ngoài ra với `NumPy` thì phân tán và phân tán chung mặc định sẽ trả về **phân tán bất thiên** và **phân tán bất thiên chung** của dữ liệu cha với phép chia cho $n-1$. Do đó với ví dụ cân nặng và chiều cao (weight, height), chúng ta có thể đọc như sau:



Do cả phân tán và phân tán chung đều được viết trong cùng một hàng trong ma trận, do đó mà người ta gọi đó là **mà trận phân tán và phân tán chung**, để ngắn gọn ta gọi nó là **ma trận phân tán chung**. Nếu bạn chỉ muốn tính phân tán chung và phân tán của dữ liệu mẫu mà không dùng công cụ ước lượng bất thiên, bạn có thể chuyển đổi số thành `bias = True`.

```
np.cov(weight, height, bias = True)
```

```
    array([[ 75.2892562 , 115.95041322],
    [115.95041322, 198.87603306]])
```

Trong trường hợp này vì chia cho n cho nên giá trị ta nhận được đã giảm đi một chút so với trước. Nếu bạn muốn xem lại kiến thức về ước lượng bất thiên thì hãy xem lại Bài học 6.

Ta cũng có thể tính ma trận phân tán chung bằng `Pandas`.

```
import pandas as pd
df = pd.DataFrame({'weight':weight, 'height':height})
df
```

	weight	height
0	42	138
1	46	150
2	53	152
3	56	163
4	58	164
5	61	167
6	62	165
7	63	182
8	65	180
9	67	180
10	73	183

```
df.cov()
```

	weight	height
weight	82.818182	127.545455
height	127.545455	218.763636

Kết quả trả về là một `DataFrame`. Nó trông thật dễ nhìn. Như thế thì với nhiều biến số, việc sử dụng `DataFrame` sẽ cho kết quả dễ xử lý hơn `NumPy`. Trong trường hợp ví dụ này thì chỉ có hai biến số, nhưng nếu tăng lên 3 hay 4 biến số, do `NumPy` cho kết quả khó nhìn, nên chúng ta hãy sử dụng `DataFrame` nhé.

`.cov()` của `DataFrame` cũng chia cho $n-1$, nó cũng trả về là phân tán bất thiên và phân tán bất thiên chung.

10.5 Tổng kết

Bài học này tôi đã giới thiệu cho các bạn vè phân tán chung biểu thị môi quan hệ tương quan giữa hai biến số trong thống kê mô tả. Đó là một cái tên không quen thuộc, vì vậy những người mới bắt đầu không thích thống kê, nhưng tôi nghĩ nó không khó chút nào.

- Phân tán chung là một phiên bản công thức phân tán cho hai biến số.

- Phân tán chung không biểu diễn phân tán hay phân bố của dữ liệu, nó được sử dụng với tư cách là một chỉ số cho mối quan hệ tương quan giữa hai biến số.
- Phân tán chung của hai biến số là dương nếu mối quan hệ tương quan là thuận, và âm nếu mối quan hệ tương quan là nghịch.
- Ma trận phân tán chung là ma trận biểu diễn phân tán chung và phân tán thông thường bằng tên các biến số.
- Có thể dùng `np.cov()` hay `df.cov()` để tính ma trận phân tán chung.
- `np.cov()` hay `df.cov()` khi được sử dụng, chúng trả về phân tán bất thiên và phân tán chung bất thiên.

Qua bài học này hy vọng các bạn đã hiểu về phân tán chung. Tôi nghĩ cái tên hiệp phương sai mà các tài liệu hiện nay đang dùng không dễ hiểu bằng phân tán chung, cho nên trong bài viết này, tôi sử dụng từ phân tán chung rất nhiều lần. Tôi hi vọng những người đã tiếp xúc với các tài liệu thống kê khác trước đó ở Việt Nam không cảm thấy khó chịu về điều này. Khi chúng ta đã hiểu ý nghĩa, tôi nghĩ cái tên là gì không còn thực sự quan trọng nữa.
Phân tán chung được sử dụng để tìm hệ số tương quan, thể hiện mối quan hệ tương quan là mạnh hay yếu.
Chúng ta đã biết rằng phân tán chung sẽ trở thành dương khi có tương quan thuận và âm khi có tương quan nghịch, và trở thành 0 nếu không có tương quan, nhưng làm thế nào chúng ta có thể tìm thấy độ mạnh yếu của mối tương quan?
Trước đó chúng ta đã có ví dụ về weight và height có phân tán chung là 115.9 và 127.5 (bất thiên), rút cuộc nó có ý nghĩa như thế nào?
Câu trả lời cho câu hỏi này, đó là chỉ số được gọi là hệ số tương quan, sẽ được giải thích ở bài học tiếp theo.

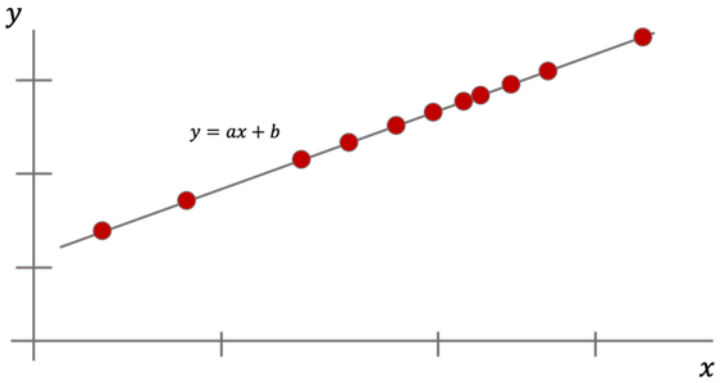
Bài 11

Giải thích dễ hiểu về hệ số tương quan

Trong Bài học 10, chúng ta đã được biết về phân tán chung với tư cách như là một chỉ số biểu thị mối quan hệ tương quan giữa hai biến số.
Khi nhìn vào phân tán chung, bạn có nghĩ mình có chút nghi vấn nào không? Mặc dù đã hiểu chỉ số biểu thị mối quan hệ tương quan giữa hai biến số, khi tương quan thuận sẽ có giá trị dương, và khi tương quan nghịch sẽ có giá trị âm, và nếu không tương quan sẽ có giá trị 0, thhé nhưng giá trị phân tán chung sau khi được tính toán cần phải diễn giải như thế nào?
Ở bài học trước, chúng ta đã tính phân tán chung của weight và height là 115, giá trị đó là cao hay thấp, mối tương quan đó là mạnh hay yếu, chúng ta vẫn chưa lý giải được.
Ngoài ra đơn vị của weght [kg] và height [cm] cũng khác nhau, chúng ta cũng chưa thực hiện chuẩn hóa.
Nếu nhhư chúng ta có thể chuẩn hóa các giá trị mà phân tán chung có thể nhận, ta có thể sử dụng các giá trị đó để so sánh phải không nào?

11.1 Khi mối tương quan đạt cực đại, biến số xếp trên một đường thẳng

Chúng ta hãy cùng suy ghĩ về trường hợp mối tương quan giữa hai biến số đạt cực đại.
Mối tương quan giữa biến số x,y đạt cực đại có nghĩa là một khi biến số x được xác định thì giá trị biến số y cũng được xác định hoặc ngược lại. Giống như ví dụ dưới đây, khi x và y có mối quan hệ tuyến tính, tức là dữ liệu xếp trên một đường thẳng, có thể nói đây là mối quan hệ tương quan thuận hoàn toàn. Khi đó, phân tán chung đạt cực đại.



Nào, vậy thì khi chúng ta có mối tương quan thuận hoàn toàn, phân tán chung sẽ trở nên ra sao? Chúng ta hãy xem biến đổi sau với chú ý rằng $y = ax + b$.

$y_i = ax_i + b$
$\bar{y} = a\bar{x} + b$

$$\begin{aligned} s_{xy} &= \frac{1}{n}\sum (x_i - \bar{x})\left(y_i - \bar{y}\right) \\ &= \frac{1}{n}\sum (x_i - \bar{x})\left(ax_i + \cancel{b} - \left(a\bar{x} + \cancel{b}\right)\right) \\ &= \frac{a}{n}\sum_{i=1}^{n} (x_i - \bar{x})^2. \end{aligned}$$

Đây là công thức phân tán

Ở đây y được suy ra từ x với công thức là $y = ax + b$ cho nên độ lệch chuẩn $s_y = ax$. Tất cả các giá trị khi nhân với a thì độ lệch chuẩn cũng nhân lên với a. Và khi di chuyển dữ liệu song song (thành phần b của $y = ax + b$) thì độ lệch chuẩn không bị ảnh hưởng. Tức là các giá trị đồng thời cùng $+b$ thì độ phân tán của dữ liệu ấy không thay đổi.
Tóm lại công thức trên có nghĩa là $s_{xy} = as_x^2 = \frac{s_y}{s_x}s_x^2 = s_x s_y$, khi tương quan thuận hoàn toàn thì:

$$s_{xy} = s_x s_y$$

Tương tự như vậy, khi tương quan nghịch hoàn toàn, công thức đường thẳng lúc này là $y = -ax + b$, do hệ số $-a$ cho nên lúc này ta có:

$$s_{xy} = -s_x s_y$$

Tóm lại phân tán chung lấy giá trị từ $-s_x s_y$ đến $s_x s_y$.

$$-s_x s_y \leq s_{xy} \leq s_x s_y$$

11.2 Đây là hệ số tương quan

$$-s_x s_y \leq s_{xy} \leq s_x s_y$$

Các giá trị có thể có của phân tán chung (*hiệp phương sai*) thay đổi tùy thuộc vào thang giá trị, do đó rất khó so sánh giá trị của các phân tán chung (*hiệp phương sai*).

Chia các vế của bất đẳng thức trên cho $s_x s_y$ ta có:

$$-1 \leq \frac{s_{xy}}{s_x s_y} \leq 1$$

Tỷ số $\frac{s_{xy}}{s_x s_y}$ này chính là hệ số tương quan (correlation coefficient). Do phân tán chung có vấn đề vì giá trị có độ lớn thay đổi, vì vậy người ta lấy tỷ số này như một chỉ số để biểu thị mối quan hệ tương quan, gọi là **hệ số tương quan**.
Hệ số tương quan có giá trị từ -1 tới 1, khi mối tương quan thuận hoàn toàn thì hệ số tương quan là 1 và khi tương quan nghịch hoàn toàn thì hệ số tương quan là -1. Và khi không có mối tương quan thì hệ số tương quan là 0.
Nếu có hai biến số x và y thì người ta ký hiệu r là hệ số tương quan của hai biến số trên, khi đó:

$$r = \frac{s_{xy}}{s_x s_y}$$

Bổ sung

Ngoài chỉ số này còn có một hệ số khác cũng được gọi là hệ số tương quan, trong trường hợp đó thì chỉ số này được gọi là **hệ số tương quan Pearson**. Thông thường khi nói tới hệ số tương quan, thì mọi người thường nghĩ đó là *hệ số tương quan Pearson*. Do đó trong khuôn khổ cuốn sách này, nó được gọi đơn giản là hệ số tương quan.

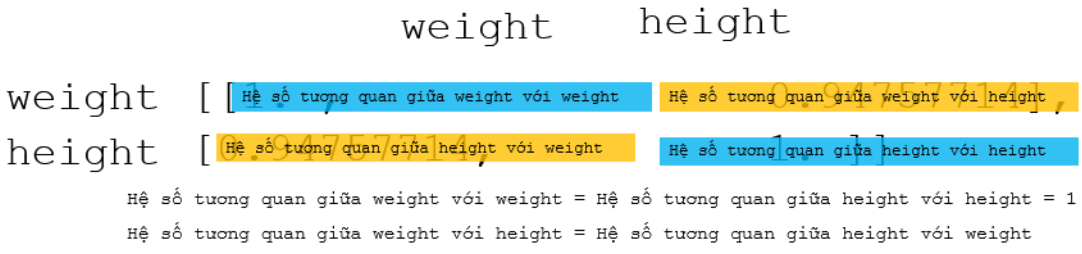
11.3 Thử tính hệ số tương quan bằng Python

Nào, bây giờ chúng ta hãy thử tính hệ số tương quan bằng `Python`. Đầu tiên ta sẽ dùng `.corrcoef()` của `NumPy`.

```
import numpy as np
weight = np.array([42, 46, 53, 56, 58, 61, 62, 63, 65, 67, 73])
height = np.array([138, 150, 152, 163, 164, 167, 165, 182, 180, 180, 183])

r = np.corrcoef(weight, height)
r
```

```
array([[1.        , 0.94757714],
[0.94757714, 1.        ]])
```

Giống như bài trước, kết quả trả về là mảng `ndarray`. Hàm `corrcoef()` trả về ma trận tương quan **(correlation matrix)**.



Đương nhiên hệ số tương quan giữa các giá trị giống nhau sẽ là 1, ví dụ hệ số tương quan giữa weight với weight sẽ là 1. Khi hoán đổi các biến số thì giá trị tương quan giữa các biến số là như nhau. Ví dụ hệ số tương quan giữa weight với height giống với hệ số tương quan giữa height với weight. Do đó ma trận tương quan là ma trận đối xứng với đường chéo là 1.

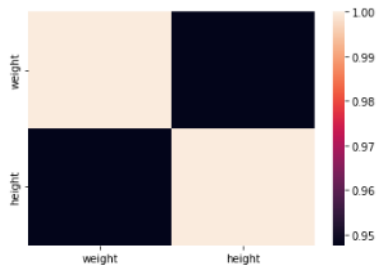
Trong công việc thực tế thường sẽ sử dụng `Pandas` để tạo ma trận tương quan.

```
import pandas as pd
df = pd.DataFrame({'weight':weight, 'height':height})
df.corr()
```

	weight	height
weight	1.000000	0.947577
height	0.947577	1.000000

Ở bài học 26 khóa học Python hướng tới Data Science, tôi đã giới thiệu cách tổ chức số liệu với `Heatmap`. Tất nhiên trong trường hợp chỉ có hai biến số thì Heatmap là không cần thiết, nhưng trong thức tế thường chúng ta sẽ có rất nhiều biến số cần quan sát tương quan của chúng với nhau.

```
import seaborn as sns
sns.heatmap(df.corr())
```



11.4 Tổng kết.

Trong bài học này tôi đã giới thiệu **hệ số tương quan** là một chỉ số quan trọng để biểu thị mối quan hệ tương quan là mạnh hay yếu giữa hai biến số.

- Hệ số tương quan có giá trị từ -1 tới 1, nó biểu thị mối quan hệ tương quan mạnh hay yếu giữa hai biến số.
- Hệ số tương quan là phân tán chung giữa các biến số chia cho tích độ phân tán (tài liệu gốc: độ lệch chuẩn-nhưng xem xét lại công thức tôi cho rằng chưa chính xác) của các biến số. Nói tóm lại nó được định nghĩa là $r = \frac{s_{xy}}{s_x s_y}$.
- Ma trận tương quan là ma trận biểu diễn mối tương quan giữa các biến số, với thành phần đường chéo là 1.

- Có thể sử dụng `.corrcoef()` trong `NumPy` hoặc `.corr()` trong `Pandas` để tính ma trận tương quan.

Trong bài học trước các bạn đã được giới thiệu về phân tán chung, các bạn đã hiểu chưa? Trong thực tế, cơ hội bắt gặp hệ số tương quan là rất nhiều. Đặc biệt là trong khóa học này.
Ví dụ, khi học một mô hình bằng máy học, một biến sẽ bị loại trừ khỏi quá trình học nếu mối tương quan giữa các biến cao. Phần này sẽ được giải thích lại trong khóa học Machine Learning.
Tôi nghĩ rằng nhiều người hiểu định nghĩa nhưng không hiểu cách xử lý. Bài học tới chúng ta sẽ xem xét kỹ hơn về **hệ số tương quan**.

Bài 12

Ba điểm nên biết về hệ số tương quan

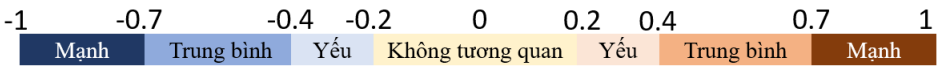
Trong Bài học 11, chúng ta đã nói tới hệ số tương quan, và ở bài học này chúng ta tiếp tục bàn về nó.
Chúng ta đã hiểu rằng, hệ số tương quan có thể được sử dụng như là một chỉ số biểu thị độ mạnh yếu của mối quan hệ tương quan giữa các biến số.
Trong bài học này, tôi sẽ giới thiệu tới các bạn ba điểm chú ý về hệ số tương quan.

- Tiêu chuẩn tương quan mạnh, tương quan yếu. Hệ số tương quan là bao nhiêu thì có thể nói đó là tương quan mạnh?
- Có nguy hiểm nếu chỉ nhìn vào hệ số tương quan? Nên quan tâm cả biểu đồ phân bố phân tán.

Nội dung trong bài học này rất quan trọng, các bạn hãy ghi nhớ.

12.1 Tiêu chuẩn tương quan mạnh yếu

Hệ số tương quan có giá trị từ -1 đến 1, việc chuẩn hóa thang đo giá trị như vậy thật là dễ sử dụng. Tuy nhiên hệ số tương quan như thế nào thì được coi đó là mối tương quan mạnh? Đó là điều chúng ta vẫn chưa biết.
Thực ra không có một tiêu chuẩn chung để có thể đưa ra một kết luận kiểu như là đây là mối tương quan mạnh. Nói rộng hơn, tùy thuộc vào lĩnh vực, chủng loại dữ liệu mà tiêu chuẩn này sẽ thay đổi, do đó trong thực tế không có một tiêu chuẩn chung.
Vậy câu hỏi đặt ra, có thể đưa ra những ví dụ cụ thể nào đó mà chúng ta có thể tham khảo hay không? Vâng, tôi sẽ giới thiệu một thang đo mà chúng ta có thể tham khảo trong những trường hợp thông thường.

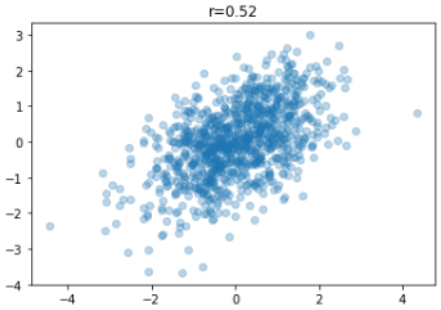


Đại khái, hệ số tương quan (giá trị tuyệt đối) từ 0.7 trở lên, ta nói đó là tương quan mạnh. Từ 0.2 trở xuống, có thể nghĩ rằng hầu như không có tương quan.
Chúng ta sẽ sử dụng **Python** và thử nhìn vào biểu đồ phân bố phân tán để kiểm chứng.
Đầu tiên, tôi sẽ tạo ra một chương trình tạo một bộ dữ liệu ngẫu nhiên với hệ số tương quan tùy ý.

```
import numpy as np
def generate_values(r=0.5, num=1000):

    # Tao gia tri ngau nhien voi phan bo chuan
    a = np.random.randn(num)
    # Do lech 1: Ta se tang them 'a' vao gia tri x
    e1 = np.random.randn(num)
    # Do lech 2: Ta se tang them 'a' vao gia tri y
    e2 = np.random.randn(num)
    # Neu he so tuong quan am, vi khong the lay can bac hai cua so am, cho nen ta se lay -
     x
    if r < 0:
    r = -r
    x = -np.sqrt(r)*a - np.sqrt(1-r)*e1
    else:
    x = np.sqrt(r)*a + np.sqrt(1-r)*e1
    y = np.sqrt(r)*a + np.sqrt(1-r)*e2
    # Lay he so tuong quan giua x va y thong qua ma tran tuong quan
    actual_r = np.corrcoef(x, y)[0][1]

    return x, y, actual_r
```

`np.random.randn()` đã được giới thiệu trong bài học 8 nhập môn Python hướng tới Data Science. Đây là hàm tạo các giá trị ngẫu nhiên với phân bố chuẩn. `np.sqrt()` lấy căn bậc hai.
Tôi không giải thích tại sao lại sử dụng công thức `np.sqrt(r)*a + np.sqrt(1-r)*e1`. Trong thực tế cũng không có công việc tạo ra số ngẫu nhiên từ hệ số tương quan tùy ý. Ở đây chỉ có tính chất tham khảo, nếu bạn có hứng thú, hãy thử tự mình suy nghĩ.
Nào, bây giờ chúng ta sẽ vẽ biểu đồ phân tán phân bố.

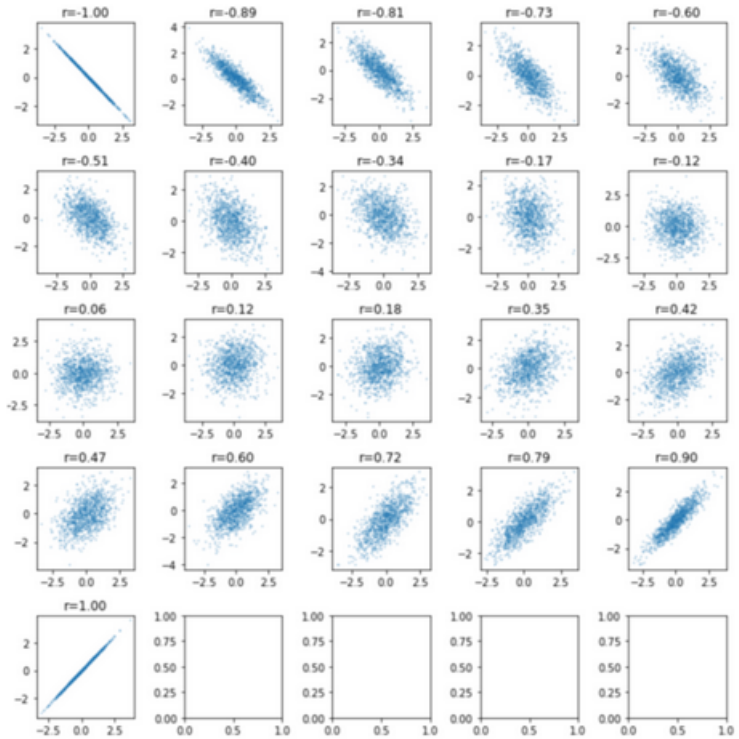
```
mport matplotlib.pyplot as plt
%matplotlib inline

x, y ,actual_r = generate_values()
plt.scatter(x, y, alpha=0.3)
plt.title('r={:0.2f}'.format(actual_r))
```



Số ngẫu nhiên đã được tạo ra do đó hệ số tương quan thực tế không nhất thiết phải là 0.52.

Vậy thì bây giờ chúng ta sẽ sử dụng hàm này để tạo ra các trường hợp với hệ số ngẫu nhiên đi từ -1 tới 1 xem sao:

```
# Toi muon hien thi bieu do phan bo phan tan cua he so tuong quan tu -1 den 1 voi don vi
    la 0.1 (tong cong la 21 lan)

#Hien thi voi kich thuoc 7x3 (Vi co 21 diem moc)
rows = 5
cols = 5
figsize = (10, 10)
fig, ax = plt.subplots(rows, cols, figsize=figsize)
for idx, r in enumerate(np.arange(-1, 1.1, 0.1)):
   # Tinh toan dat cac plot o dau
   row_i = idx//cols
   col_i = idx%cols
   #Tinh toan gia tri ngau nhien ung voi he so tuong quan
   x, y, actual_r = generate_values(r=r)
   #Hien thi he so tuong quan tu cac gia tri ngau nhien da duoc tao ra
   title = 'r={:.2f}'.format(actual_r)
   ax[row_i, col_i].set_title(title)
   ax[row_i, col_i].scatter(x, y, alpha=0.3, s=1)
# Ngan chan cac tieu de va truc chong cheo len nhau
fig.tight_layout()
```



Việc tạo số ngẫu nhiên với hệ số tương quan tùy ý như ban nãy thực sự không quan trọng nhưng việc sử dụng **for** để vẽ biểu đồ với nhiều `plot` là vô cùng quan trọng. Các bạn hãy luyện tập vì kỹ thuật này rất quan trọng, thường sử dụng trong Data Science.
Nhìn vào biểu đồ trên, ta thấy khi hệ số tương quan từ 0.7 trở lên, ta thấy dữ liệu rất tập trung như một quần thể. Hãy so sánh với biểu đồ khi hệ số tương quan bằng 0.2 hay 0.

12.2 Có nguy hiểm nếu chỉ nhìn vào hệ số tương quan? Nên quan tâm cả biểu đồ phân bố phân tán

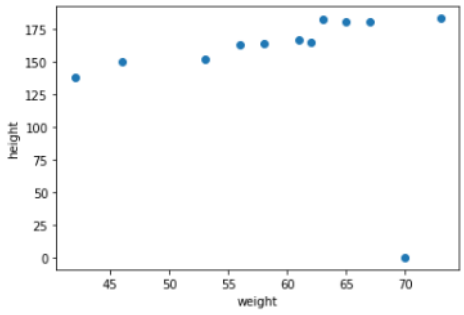
Trước đó, chúng ta đã thấy biểu đồ phân tán phù hợp với hệ số tương quan, nhưng nếu bạn muốn xem mối tương quan giữa các biến mà bạn quan tâm, hãy cố gắng xem cả biểu đồ phân tán càng nhiều càng tốt.

Giá trị của hệ số tương quan có xu hướng thay đổi đáng kể tùy thuộc vào các giá trị ngoại lệ lớn hay nhỏ (các giá trị nhiễu). Chúng ta hãy lấy ví dụ về weight và height trong ví dụ trước đây.

Chúng ta hãy tính hệ số tương quan trong trường hợp chúng ta thêm vào dữ liệu một giá trị ngoại lệ.

```
import numpy as np
weight = np.array([42, 46, 53, 56, 58, 61, 62, 63, 65, 67, 73])
height = np.array([138, 150, 152, 163, 164, 167, 165, 182, 180, 180, 183])
#Truoc khi them gia tri ngoai le
r = np.corrcoef(weight, height)[0][1]
print('r={:.2f} (before adding outlier)'.format(r))
#Them gia tri ngoai le
weight = np.append(weight, 70)
height = np.append(height, 0)
r = np.corrcoef(weight, height)[0][1]
print('r={:.2f} (after adding outlier)'.format(r))
plt.scatter(weight, height)
plt.xlabel('weight')
plt.ylabel('height')
```

```
r=0.95 (before adding outlier)
r=-0.09 (after adding outlier)
```



Chỉ với một giá trị ngoại lệ, hệ số tương quan thay đổi khá mạnh, từ 0.95 thay đổi thành 0.09. Do đó bạn nên xem biểu đồ phân tán phân bố để xác nhận có giá trị ngoại lệ nào hay không.

Định nghĩa giá trị ngoại lệ như thế nào?

Để hiểu về giá trị ngoại lệ hay giá trị dị thường cần có kiến thức miền, tức là cần kiến thức về lĩnh vực mà mình thu thập dữ liệu. Ví dụ, trong trường hợp này, khi thiết định trong lượng là 0 thì đây là giá trị dị thường. Do đó cái giá trị như thế, có thể

dự đoán rằng có sự bất thường trong thiết bị, lỗi của con người khi thu thập dữ liệu, hoặc một số lỗi do xử lý dữ liệu.

12.3 Phân biệt hệ số tương quan với quan hệ nhân quả

Có sự ngộ nhận rằng quan hệ tương quan giống với quan hệ nhân quả.
Ví dụ, trong ví dụ lần này ta thấy trọng lượng (weight) và chiều cao (height) có mối quan hệ tương quan. Vậy rút cuộc nó có ý nghĩa như thế nào?
Bạn có thể nói rằng , người càng cao thì cân nặng có khuynh hướng càng lớn. Nhưng bạn không thể nói rằng, vì tôi cao nên cân nặng của tôi cũng lớn.
Tóm lại ta không thể nói rằng, vì tôi cao nên cân nặng của tôi lớn, hoặc ngược lại.
Nếu bạn muốn chỉ ra rằng, bạn càng cao thì cân nặng của bạn càng lớn, bạn cần phải làm gì đó để tăng chiều cao của bạn và quan sát sự thay đổi của cân nặng đối với từng dữ liêu mà bạn thu thập được.
Thứ hai, vì bạn cao nên bạn nặng, đây là mối quan hệ nhân quả, nhưng điều này cực kỳ khó chứng minh từ các dữ liệu. Ngay cả khi bạn quan sát dữ liệu trong một khoảng thời gian về chiều cao và cân nặng của một ai đó. Bạn có thể tưởng tượng ra rằng, tăng cân không thể quy cho tăng chiều cao. Có thể một lúc nào đó bạn chỉ tăng cân mà không tăng chiều cao.
Muốn làm thí nghiệm xác định mối quan hệ nhân quả thì cần xét đến các biến số khác, thực tế khó có thể làm rõ mối quan hệ nhân quả.
Dù sao đi nữa, điều quan trọng ở đây là, mối quan hệ tương quan chỉ nói lên xu hướng đối với tập hợp chứ không phải là mối quan hệ nhân quả trong mỗi dữ liệu. Chú ý rằng đây là một quan niệm sai lầm nhưng rất phổ biến.

12.4 Tổng kết

Trong bài học này tôi đã giới thiệu ba điểm chính. Nó rất quan trọng trong Data Science.

- Hệ số tương quan từ 0.7 trở lên được coi là tương quan mạnh. Từ 0.2 trở xuống hầu như không có tương quan.
- Nếu chỉ quan tâm tới hệ số tương quan thì rất nguy hiểm, hãy cố gắng quan tâm tới cả biểu đồ phân bố phân tán.
- Hệ số tương quan chỉ có thể nói lên khuynh hướng của dữ liệu chứ không phải là mối quan hệ nhân quả.

Bài 13

Giải thích phân tích hồi quy bằng biểu đồ! Trung bình cộng có điều kiện và phương pháp bình phương tối thiểu

Trong bài học này chúng ta sẽ nói về một khái niệm rất gần gũi với hệ số tương quan, nó có tên là "**hồi quy**" (regression).
Hồi quy là một chủ đề dẫn tới Machine Learning, và có thể nói rằng nó là phần cơ bản nhất của Data Science. Vì vậy chúng ta hãy nắm vững nó.

13.1 Hồi quy là gì?

Trong Bài học 12 chúng ta đã biết tới hệ số tương quan, và chúng ta đã lý giải được giá trị của một biến khi cao thì biến số còn lại cũng có khuynh hướng có giá trị cao (hoặc thấp). Nhưng chúng ta chưa bàn tới mức độ đó là giá trị của biến số cao tới mức nào thì giá trị của biến số còn lại sẽ trở nên cao tới một mức nào đó hay thấp tới mức nào đó.
Nếu hiểu được điều này, khi đó chỉ cần biết giá trị của một biến số, ta có thể suy luận ra giá trị của biến số còn lại. Nói cách khác, chúng ta có thể đưa ra một dự đoán! Đó là một trong những lý do cho Data Science, đó là có thể dự đoán giá trị của một biến khác từ giá trị của một biến biết trước. Điều này rất có ích trong thế giới thực.
Ví dụ, khi đưa ra quyết định về giá trị của một căn nhà, từ những thông tin biến số như là ngôi nhà bao nhiêu tuổi, diện tích bao nhiêu, khoảng cách từ nhà ga tới nó là bao xa, từ những thông tin như thế nếu có thể dự đoán được giá nhà thì việc mua bán nhà thật sự rất tiện lợi.
Chúng ta có thể chỉ ra giá căn nhà đại khái là khoảng bao nhiêu.
Từ những thông tin số tuổi của căn nhà, diện tích, khoảng cách từ nhà ga tới nó ... rồi đưa ra giá nhà thì giá nhà này được gọi là **giá nhà trung bình**. Hãy cùng nhau suy nghĩ một ví dụ đơn giản hơn. Trong bài học trước ta đã có một ví dụ về trọng lượng cơ thể và chiều cao. Hãy thử dự đoán trọng lượng cơ thể được suy ra từ chiều cao xem sao.

Ví dụ, những người có chiều cao 170 cm thì trọng lượng cơ thể trung bình là bao nhiêu? Việc đưa vào điều kiện [chiều cao 170 cm] rồi tính trung bình trọng lượng cơ thể, ta gọi đó là trung bình có điều kiện.
Hồi quy là việc tính giá trị trung bình có điều kiện.
Giả định rằng dữ liệu được biểu diễn bởi đường thẳng, ta có công thức như sau:

$$\hat{y} = a + bx$$

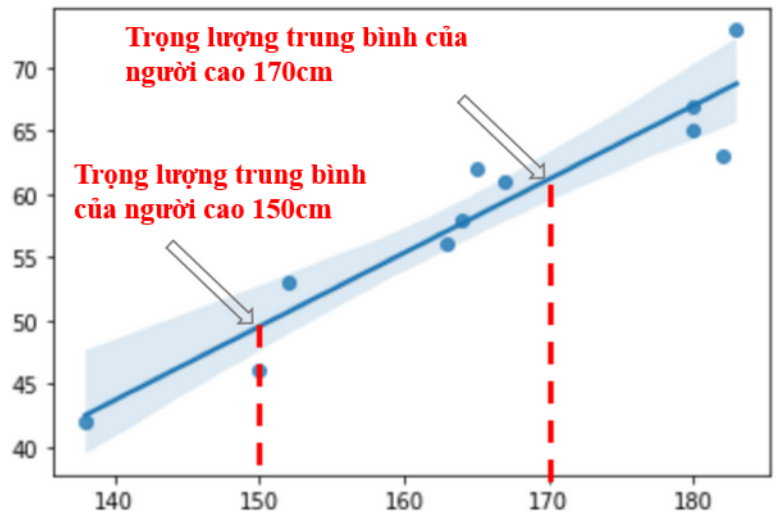
Trong đó $\hat{y}$ là trung bình có điều kiện của y đối với x. Tức là từ giá trị của x chúng ta có thể dự doán được giá trị của y.
Điều này gọi là hồi quy tuyến tính của y đến x **(regression line)**. Đường hồi quy còn được gọi là hồi quy tuyến tính **(Linear Regression)**.
Công thức tính y được biểu diễn là một đường thẳng, đường thẳng này được vẽ khi sử dụng các giá trị của x, cho nên nó được hiểu là đến x hay tới x.

Bổ sung

Công thức đường thẳng thông thường được viết là $y = ax + b$ nhưng trong hồi quy tuyến tính thông thường được viết là $y = a + bx$.

Quay trở lại ví dụ về weight(trọng lượng) và height (chiều cao), chúng ta có thể vẽ đường hồi quy như sau. (Đường hồi quy có thể được vẽ một cách đơn giản bằng `regplot()` của `seaborn`).

```
import numpy as np
import seaborn as sns
%matplotlib inline

weight = np.array([42, 46, 53, 56, 58, 61, 62, 63, 65, 67, 73])
height = np.array([138, 150, 152, 163, 164, 167, 165, 182, 180, 180, 183])
sns.regplot(height, weight)
```



Đường thẳng trên biểu diễn trọng lượng trung bình với điều kiện là chiều cao.
Ví dụ, đối với chiều cao 150 cm, theo dữ liệu trọng lương tương ứng là 46 kg. Nhưng

đường hồi quy chỉ đơn thuần tính toán giá trị trung bình dựa trên dữ liệu đưa vào, do đó tất nhiên sẽ có sai khác với dữ liệu thực tế. (Với chiều cao 150 cm trọng lượng khoảng 48 kg).

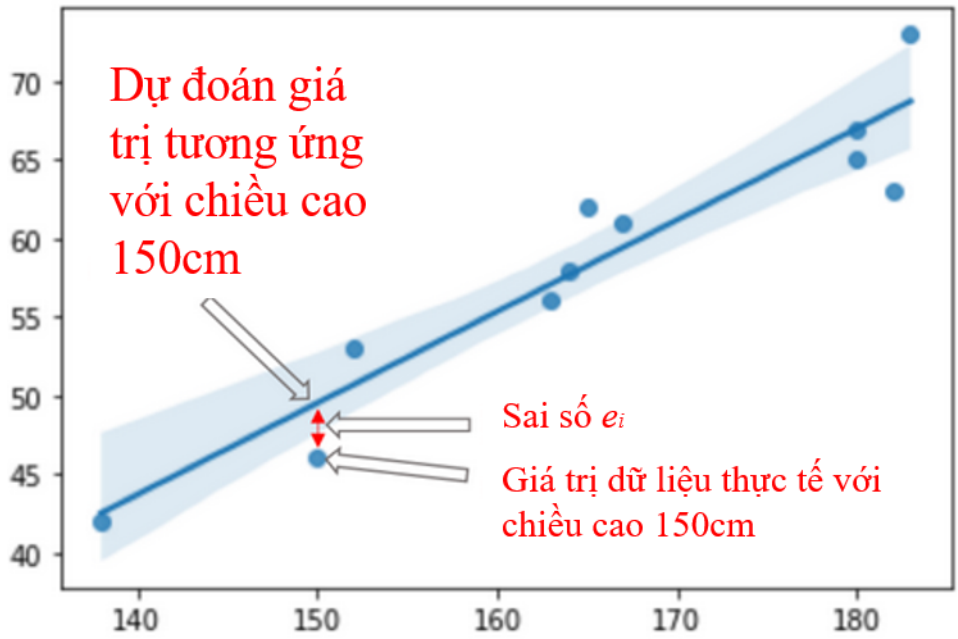
Bổ sung

Phần màu xanh lam gần đường hồi quy được hiển thị bởi `sns.regplot()` được gọi là khoảng tin cậy 95%. Bạn có thể vẽ một đường thẳng tương đối trong phạm vi này. Tôi sẽ giải thích chi tiết trong một bài viết khác.

Biến số x sẽ quyết định giá trị trung bình của biến số y, biến số x được gọi là biến số hồi quy **(regressor)**, biến số y gọi là biến số bị hồi quy **(regressand)**.
Có những cách gọi khác, biến x gọi là biến độc lập **(independent variable)**, biến y gọi là biến phụ thuộc **(dependent variable)**, cách đặt tên biến số dựa vào mục đích của biến số.
Việc kẻ đường thẳng như thế này để phân tích mối quan hệ hồi quy được gọi là phân tích hồi quy **(regression analysis)**, bạn hãy ghi nhớ nó.

13.2 Quyết định công thức đường thẳng như thế nào? Phương pháp bình phương tối thiểu

Bằng cách kéo dài đường tuyến tính, từ giá trị của biến số, ta có thể dự đoán được giá trị của biến số còn lại. Điều này thật tiện lợi.
Ta đã biết phương trình đường thẳng $y = a + bx$.
Tuy nhiên làm thế nào để xác định được đường thẳng này?
Nếu như có thể xác định được a và b ta sẽ biết được đường hồi quy $y = a + bx$. Vậy làm thế nào để xác định được a và b ?
Việc sử dụng phương pháp bình phương tối thiểu **(least squares method)** có thể giúp chúng ta xác định được đường hồi quy. Đây là điều vô cùng quan trọng, bạn tuyệt đối hãy ghi nhớ nó!
Nếu chỉ nghe tới cái tên, nhiều người sẽ cảm thấy mệt mỏi với thống kê học vì từ đầu tới giờ đã có rất nhiều khái niệm rồi, có thể các bạn muốn từ bỏ. Thế nhưng phương pháp này thật ra rất đơn giản.
Nào, đầu tiên chúng ta sẽ xem xét giá trị dự đoán lệch so với giá trị dữ liệu thực tế như thế nào? Hãy nhìn vào biểu đồ dưới đây.



Ví dụ, chiều cao 150cm ($x = 150$) ta có giá trị dự đoán ($y = a + b \times x$) với giá trị thực tế 46 có độ lệch t gọi đó là sai số (Residual error), hay còn gọi là chênh lệch sai sót, chênh lệch sai sót tại điểm dữ liệu thứ i gọi là e_i.
Với toàn bộ dữ liệu, ta tính tổng các số e_i, đường thẳng hồi quy cần tìm là đường thẳng làm sao để tổng này có giá trị nhỏ nhất.
Tuy nhiên sai sót chênh lệch khi thì lớn, khi thì nhỏ, khi tính toán Giá trị dự đoán − Giá trị thực tế có thể có giá trị âm, hoặc giá trị dương. Điều này thật nguy hiểm. Để khắc phục điều này ta sẽ bình phương chúng lên. Bởi vì trong thống kê học người ta không thích sử dụng giá trị tuyệt đối.
Tóm lại công thức tính tổng lúc này sẽ là:

$$\sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} \{y_i - (a + bx_i)\}^2$$

Ta cần tìm a, b để tổng trên có giá trị nhỏ nhất. Chúng ta có thể tìm được chúng bằng cách lập phương trình vi phân cho từng biến. Phương pháp tính toán này tôi nghĩ không thực sự quan trọng nên tôi sẽ lược bỏ ở đây. Nhưng điều quan trọng đó là chúng ta vẽ ra một đường để thu nhỏ bình phương của sai sót chênh lệch và đây được gọi là **phương pháp bình phương tối thiểu**.
Vậy thì bằng phương pháp này, a và b sẽ nhận giá trị như thế nào? Do bài học đến đây đã dài nên tôi sẽ trình bày nó trong bài học tiếp theo.

13.3 Tổng kết

Trong bài học này tôi đã giải thích về phân tích hồi quy. Phân tích hồi quy có thể giúp chúng ta tiếp cận các vấn đề khác nhau trong thế giới thực. Nó là một trong những phần rất cơ bản của Data Science, cho nên các bạn hãy ghi nhớ nó.

- Hồi quy là trung bình có điều kiện.

- Đường hồi quy có dạng $y = a + bx$.
- Đường hồi quy là đường làm sao để bình phương sai sót chênh lệch nhỏ nhất.

Bài 14

Lý giải công thức đường hồi quy bằng hình ảnh. Mối liên quan giữa hệ số hồi quy và hệ số tương quan

Trong Bài học 13, chúng ta đã biết rằng thông qua phương pháp bình phương tối thiểu, ta có thể tìm được công thức đường hồi quy $y = a + bx$.

- Thực tế a và b có giá trị như thế nào?
- Thực tế tìm công thức đường hồi quy bằng `Python` như thế nào?

Trong bài học này chúng ta sẽ cùng nhau giải quyết những khúc mắc còn tồn đọng ở trên. Tính lý luận trong bài học này là vô cùng quan trọng. Tôi sẽ sử dụng biểu đồ để giải thích giúp các bạn dễ hình dung. Điều quan trọng nhất là các bạn có thể hình dung và lý giải được.

14.1 Tìm công thức đường hồi quy $y = a + bx$

Chúng ta cần tìm a và b. Việc suy ra bằng công thức toán học không quan trọng, vì vậy chúng ta sẽ sử dụng hình ảnh để hình dung ra kết quả và ý nghĩa của kết quả này. Điều quan trọng là có thể tưởng tượng và hình dung hơn là đi sâu vào biến đổi toán học.

$$\sum_{i=1}^{n} \{y_i - (a + bx_i)\}^2$$

Trong công thức trên ta coi a và b là hai biến số. Để tìm giá trị nhỏ nhất, ta cần tính đạo hàm và xem xét khi nào đạo hàm có giá trị là 0. Trong trường hợp này, biểu thức có hai biến, cho nên trước hết ta coi đây là phương trình với ẩn là a, khi đó thì đạo hàm bằng 0 khi nào, tương tự như thế ta coi đây là phương trình ẩn b và cũng tính đạo hàm và xem xét khi nào thì đạo hàm bằng 0. Tất nhiên nếu bạn có thể giải cùng lúc với hai biến số thì cũng tốt, hãy thử suy nghĩ nó nếu bạn có hứng thú.
Ồ, nói gì mà chẳng hiểu gì hết!!! Không sao, không hiểu cũng được. Ta sẽ có:

$$a = \bar{y} - b\bar{x}$$

$$b = r\frac{s_y}{s_x}$$

Trong đó r là hế số tương quan giữa x và y. Và: $\bar{y}$ và $\bar{x}$ lần lượt là giá trị trung bình của y và x.

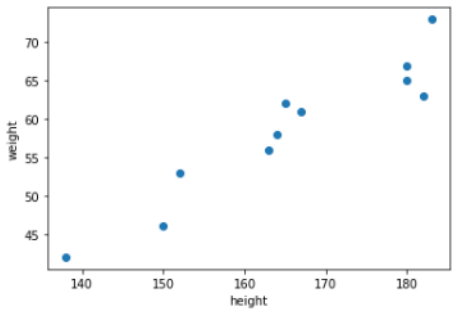
Bạn bất ngờ nhận được kết quả trên và tỏ ra không ổn lắm đúng không. Bây giờ tôi sẽ giải thích ý nghĩa của hai công thức này và hình ảnh của chúng một cách siêu dễ hiểu. Điều quan trọng nhất là bạn hình dung ra ý nghĩa của chúng hơn là một công thức toán học.

Đầu tiên, ở công thức thứ nhất $a = \bar{y} - b\bar{x}$, điều đó có nghĩa rằng $\bar{y} = a + b\bar{x}$. **Đường thẳng hồi quy của x và y đi qua điểm có tọa độ là** $(\bar{x}, \bar{y})$.

Thế nghĩa là thế nào, chúng ta hãy quay trở lại ví dụ về `height` và `weight` trước đây. Từ chiều cao, ta dự đoán trọng lượng, do đó ta coi trục Ox là chiều cao (height) và trục Oy là trọng lượng (weight).

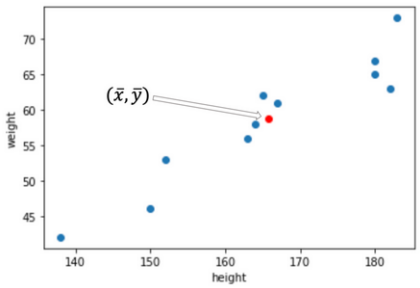
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

weight = np.array([42, 46, 53, 56, 58, 61, 62, 63, 65, 67, 73])
height = np.array([138, 150, 152, 163, 164, 167, 165, 182, 180, 180, 183])
plt.scatter(height, weight)
plt.xlabel('height')
plt.ylabel('weight')
```

Khi thực thi code trên ta sẽ có biểu đồ phân bố phân tán.



Ta cần dựng một đường thẳng mà từ đó ta có thể dự đoán được trọng lượng thông qua chiều cao. Như thế đương nhiên đường thẳng này sẽ đi qua điểm chính giữa trong số các điểm dữ liệu trên biểu đồ phân bố phân tán nói trên. Hình ảnh dễ thấy về điểm chính giữa này, không còn nghi ngờ gì nữa đó chính là điểm $(\bar{x}, \bar{y})$.

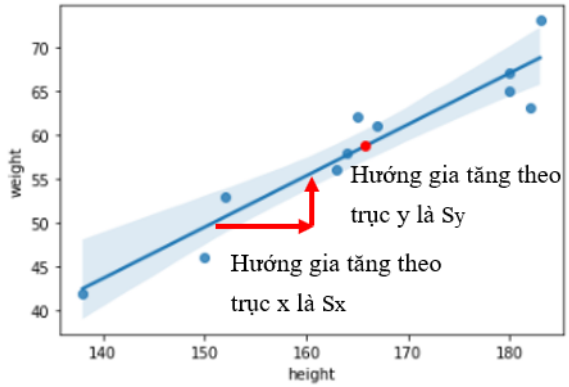
```
plt.scatter(height, weight)
plt.plot(np.mean(height), np.mean(weight), 'ro')
plt.xlabel('height')
plt.ylabel('weight')
```



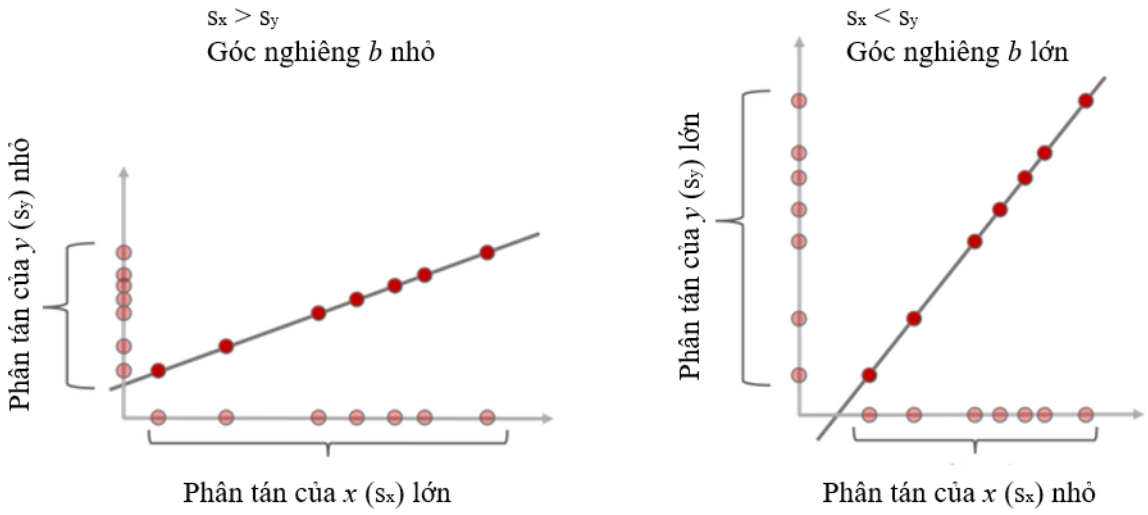
Đường thẳng này đi qua điểm $(\bar{x}, \bar{y})$ và có góc nghiêng là $b = r\frac{s_y}{s_x}$. Vì đây là góc nghiêng, nên nói một cách đơn giản nó là Phần gia tăng của phương $y \div$ Phần gia tăng của phương x.

```
sns.regplot(height, weight)
plt.plot(np.mean(height), np.mean(weight), 'ro')
plt.xlabel('height')
plt.ylabel('weight')
```

Lần trước tôi đã giới thiệu về `sns.regplot()`, nó dùng để vẽ đường hồi quy.

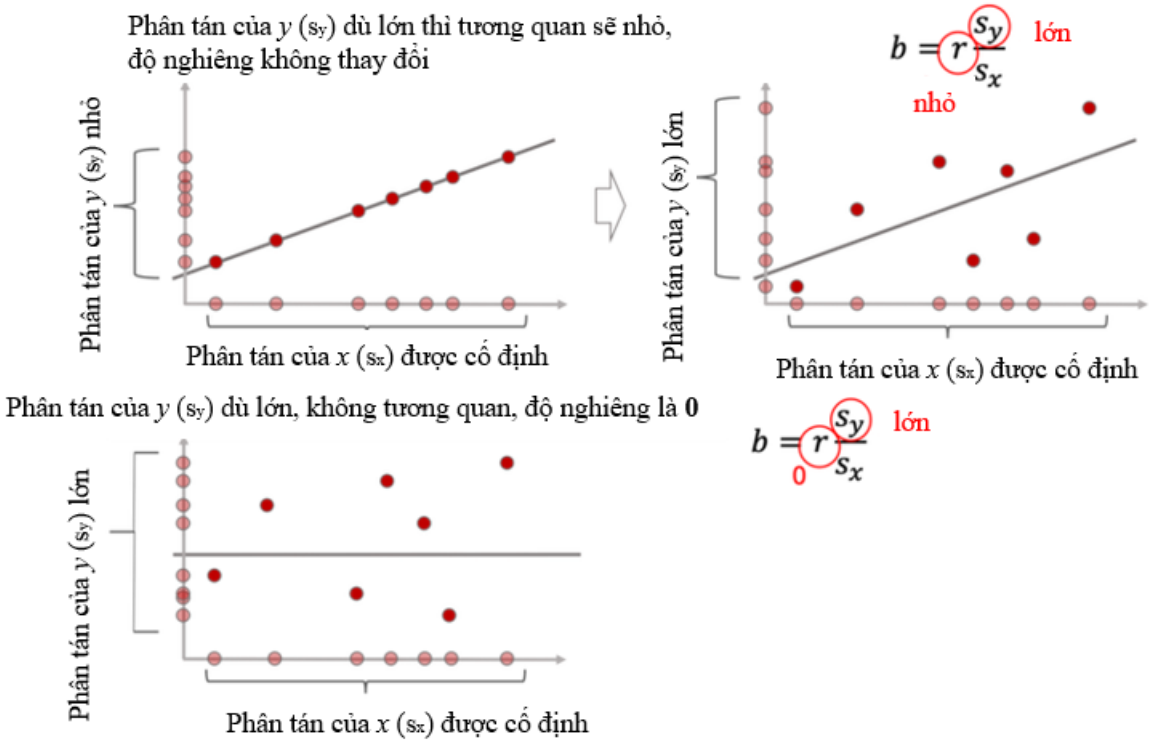


Đường mũi tên đỏ không thực sự chính xác về tỷ lệ kích cỡ, nó chỉ mang tính chất tượng trưng, $b = r\frac{s_y}{s_x}$ có ý nghĩa trục Ox gia tăng là s_x thì trục Oy gia tăng rs_y.



Trong hình minh họa này, chúng ta có ba biến số là (s_x, s_y, r) trong đó ta giả định $r = 1$. Khi hệ số tương quan là 1 thì tỷ số phân tán của y chia cho x tức là $\frac{s_y}{s_x}$ sẽ quyết định góc

nghiêng b lớn hay nhỏ. Tất nhiên nếu s_y là 0 thì khi đó dữ liệu sẽ chỉ được sắp xếp trên trục Ox, góc nghiêng đương nhiên sẽ là 0.
Tiếp theo chúng ta hãy xe xét hệ số tương quan r.



Đầu tiên hãy nhìn vào phần bên trên và so sánh hai bên trái phải của hình minh họa bên trên. Khi độ phân tán x được cố định, ở hình bên trái, phân tán của y nhỏ và hệ số tương quan là 1, ở hình bên phải thì phân tán của y lớn và hệ số tương quan < 1.
Không thể nói rằng khi s_y lớn thì góc nghiêng cũng lớn. Điều này có thể kiểm chứng bằng hình ảnh minh họa ở trên. Tương quan thấp thì góc nghiêng cũng nhỏ. Tuy nhiên giữa s_y và r có sự bù trừ cho nên bằng hình ảnh ta thấy góc nghiêng không thay đổi.
s_y lớn đến mức nào đó mà không có tương quan, thì góc nghiêng đường hồi quy sẽ tiệm cận với 0. Hãy tham khảo hình minh họa ở trên.
Dù sao đi nữa chúng ta cũng đã hiểu rằng đường hồi quy và hệ số tương quan có mối quan hệ mật thiết với nhau.
Dù nói thế nào đi nữa, thì điều quan trọng tôi muốn các bạn ghi nhớ, **đó là đường hồi quy luôn đi qua điểm $(\bar{x}, \bar{y})$ và có hệ số góc là $r\frac{s_y}{s_x}$.**
Góc nghiêng $b = r\frac{s_y}{s_x}$ gọi là hệ số hồi quy (regression coefficient).

14.2 Tìm đường hồi quy bằng Python

Nào, bây giờ chúng ta hãy thử tìm đường hồi quy, dự đoán weight từ height bằng Python.
Trong `Python` để tìm đường hồi quy ta sẽ sử dụng class `LinearRegression` trong Module `linear_model` của thư viện `scikit-learn`.

Hồi quy được sử dụng như là một trong những thuật toán của Machine Learning, vì vậy chúng ta sử dụng `scikit—learn`. Ví dụ nếu bạn có thể vẽ được đường hồi quy để dự đoán cân nặng từ chiều cao, bạn có thể sử dụng đường hồi quy đó cho dữ liệu chiều cao mới không có trong dữ liệu đó, đúng không? Nghĩa là đường hồi quy này đã được đào tạo với dữ liệu hiện có, bạn có thể coi nó như một mô hình. Để biết chi tiết, tôi mong muốn nói về nó trong một khóa học về Machine Learning.

```
from sklearn.linear_model import LinearRegression
#Bien so thuyet minh,shape la ndarray. So luong dac tinh lan nay la chi la chieu cao nen
    la 1.
X = np.expand_dims(height, axis=-1)
#shape cua Bien so muc dich la OK nen giu nguyen va su dung
y = weight
#Create instance
reg = LinearRegression()
#Ham .fit()la thuat toan hoc may voi du lieu duoc dua vao
reg.fit(X, y) #Khi thuc thi thi reg se la mo hinh may hoc

#He so b,a:
# .coef_ tra ve la ndarray. shape la X.shape
print("b={}".format(reg.coef_))
print("a={}".format(reg.intercept_))
```

Tham số của mô hình của `sklearn` là `X` và `y`, khi thực thi hàm `.fit()` có nghĩa là ta đang tiến hành cho máy học.
Hồi quy là một loại thuật toán của máy học (Machine Learning), là mô hình dự đoán có thể được dự đoán bằng việc cho máy học. Thông thường `X` là một ma trận (shape = số lượng mẫu, số lượng đặc tính) cho nên nó được viết hoa, `y` là danh sách các biến mục đích, nó được coi như là một vector cho nên ta đặt tên biến bằng chữ in thường.

Bổ sung

Các biến được sử dụng làm biến thuyết minh như lần này được gọi là các tính năng trong ngữ cảnh học máy (feature). Các tính năng sẽ được giải thích chi tiết trong phần học máy.

Lần này biến số đặc trưng (biến thuyết minh) chỉ là height nên chúng ta sử dụng `np.expand_dims()` và cho `rank` là 2. `NumPy` được nói tới trong bài học 9 khóa học Python nhập môn hướng tới Data Science.
Cách sử dụng class `LinearRegression()` (không giới hạn ở các mô hình `sklearn`) nói chung.

- Tạo `instance` (cụ thể lần này là `reg`).
- Gọi hàm `.fit()`. Khi được thực thi thì instance đó sẽ trở thành mô hình máy học.

Nếu việc học tập thành công thì ta sẽ lấy được hệ số hồi quy b và a thông qua `.coef_` và `.intercept_`.
Hệ số hồi quy được trả về bằng `ndarray`. Lĩnh vực này sẽ được giải thích chi tiết trong phần Machine Learning. Trong khuôn khổ bài học này, tôi nghĩ ở mức này là được rồi.
Nào, chúng ta hãy nhìn kết quả:

```
b=[0.58302859]
a=-37.94946808510638
```

Chúng ta quay trở lại công thức:

$$a = \bar{y} - b\bar{x}$$

$$b = r\frac{s_y}{s_x}$$

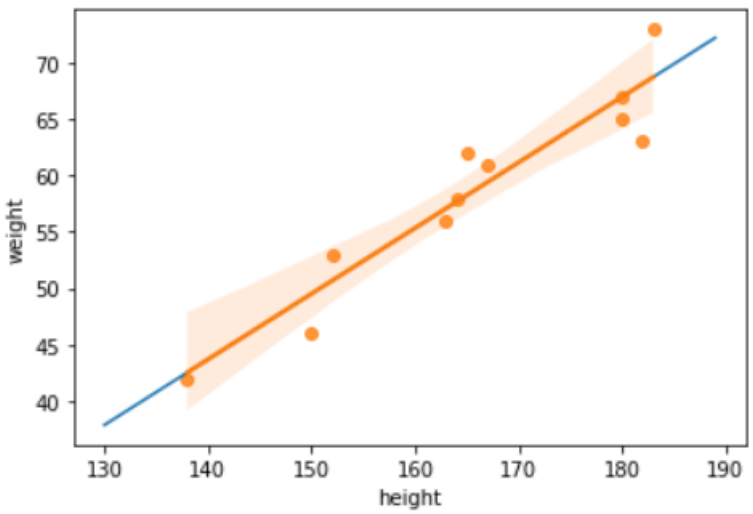
và xác nhận kết quả:

```
#Do lech chuan
s_x = np.std(height)
s_y = np.std(weight)
#Trung binh cong
mean_x = np.mean(height)
mean_y = np.mean(weight)
#He so tuong quan
r = np.corrcoef(weight, height)[0][1]
b = r * s_y/s_x
a = mean_y - b*mean_x
print("b={}".format(b))
print("a={}".format(a))
```

```
b=0.5830285904255318
a=-37.94946808510635
```

Kết quả này trùng khớp với kết quả của **sklearn**.
(Về **np.corrcoef()** có thể tham khảo Trong Bài học 11)
Nào bây giờ chúng ta hãy thử plot đường thẳng $y = a + bx$.

```
#Tao truc x
x = np.arange(130, 190, 1)
plt.plot(x, b*x+a, )
#Ve bieu do cung voi regplot seaborn
sns.regplot(height, weight)
plt.xlabel('height')
plt.ylabel('weight')
```



Đường màu xanh ta sử dụng các giá trị a, b đã được tính toán trước đó và vẽ. Đường màu cam ta sử dụng **regplot()** của **seaborn** để vẽ. Bạn có thể thấy hai đường thẳng

trùng lặp khớp với nhau hoàn toàn. Nói cách khác `regplot()` của `seaborn` chạy một thuật toán hồi quy tuyến tính bên trong và vẽ nó.
Nào, vậy thì điều gì làm bạn hài lòng với đường hồi quy tuyến tính này?
Đúng vậy, đối với giá trị chiều cao, ta có thể dự đoán và tính toán đưa ra trọng lượng tương ứng.
Ban nãy ta đã cho máy học, cho nên ta có thể sử dụng hàm `.predict()` của instance(`reg`), cái mà đã được học (`.fit()`) của `LinearRegression()` để dự đoán trọng lượng của người có chiều cao 175 cm.

```
X = np.array([[175]])
y = reg.predict(X)
print(X, y)
```

```
[[175]] [64.08053524]
```

Bạn có thể tạo một ma trận có tên là `X` và sử dụng `predict()` để dự doán.

14.3 Tổng kết

Bài học này tương dối là dài. Các thuật ngữ về thống kê cho tới bây giờ càng lúc càng nhiều hơn, thật là vất vả để nhớ chúng. Dưới đây tôi xin đưa ra mấy điểm tóm tắt chính.

- a và b của đường hồi quy $y = a + bx$ đuwọc xác định bằng: $a = \bar{y} - b\bar{x}$ và $b = r\frac{s_y}{s_x}$.
- Thay vì sử dụng công thức đạo hàm, để nhớ công thức đường hồi quy, chúng ta sử dụng hình ảnh minh họa, đây là điểm quan trọng.
- Đường hồi quy đi qua điểm $(\bar{x}, \bar{y})$ và có hệ số góc là $r\frac{s_y}{s_x}$.
- Đường hồi quy $y = a + bx$ trong đó hệ số b được gọi là hệ số hồi quy.
- Chúng ta có thể tạo ra đường hồi quy trong Python bằng `sklearn.linear_model.LinearRegression()`.
- Nếu chỉ vẽ đồ thị thì `seaborn.regplot()` thật tiện lợi.

Bài học lần này đã dài nhưng nó vô cùng quan trọng. `sklearn` không thể thiếu trong Machine Learning, nên các bạn hãy ghi nhớ nó.
Hồi quy tuyến tính (đường thẳng hồi quy) là kiến thức cơ bản trong cơ bản của Data Science. Cũng có rất nhiều thuật toán hồi quy khác nhưng cái cơ bản mà tôi muốn các bạn nhớ, đó là **hồi quy tuyến tính nêu trong bài học này và phương pháp bình phương tối thiểu**.
Lần này tôi thực hiện tạo ra đường thẳng hồi quy để dự đoán cân nặng từ chiếu cao. Nhưng điều này cũng có thể được sử dụng để dự đoán chiều cao từ cân nặng hay không? Đứng từ quan điểm kết luận, hai đường này là các đường hồi quy khác nhau, đường hồi quy mà chúng ta đã tạo ra trong bài học này không thể dùng để dự đoán chiều cao.
Bài học tới, tôi sẽ giải thích về điều này, và khi đó chúng ta sẽ có một chỉ số vô cùng quan trọng, đó là **hệ số quyết định**.

Bài 15

Hồi quy tuyến tính có thể dự đoán x từ y hay không?

Trong Bài học 14, ta đã tìm đường hồi quy $y = a + bx$ từ hai biến x và y. Thông qua phương pháp bình phương tối thiểu hệ số hồi quy b và a được xác định là:

$$a = \bar{y} - b\bar{x}$$

$$b = r\frac{s_y}{s_x}$$

Chúng ta đã dùng nó để dự đoán trọng lượng từ chiều cao. Đối với DataSet của trọng lượng và chiều cao, đường hồi quy được tính toán là $y = -37.95 + 0.58x$, bằng đường thẳng này ta có thể dự đoán được y từ giá trị của x.
Vậy câu hỏi đặt ra, ta có thể tìm được x nếu biết giá trị y hay không? Thực sự thì điều này không thực hiện được, cách sử dụng đường hồi quy như thế là cách sử dụng sai.

15.1 Có thể dự đoán y từ x bằng đường hồi quy hay không?

Từ ví dụ ban nãy, bằng đường hồi quy $y = -37.95 + 0.58x$ ta có thể sử dụng nó để dự đoán chiều cao từ trọng lượng hay không?
Câu trả lời là Không!
Tất nhiên chiều cao x có thể nhận được bằng cách thay thế trọng lượng y vào công thức đường hồi quy rồi tính toán tìm ra x. Tuy nhiên giá trị này không phải là tối ưu.
Để dự đoán giá trị chiều cao x từ giá trị trọng lượng y ta giả sử có đường hồi quy $x = a' + b'y$. Trong bài học trước, ta có công thức tìm a' và b' như sau:

$$a' = \bar{x} - b\bar{y}$$

$$b' = r\frac{s_x}{s_y}$$

Ta chỉ đổi vị trí của x và y cho nhau trong công thức trước đây, tôi nghĩ đây là điều hiển nhiên.
Bây giờ so sánh với đường hồi quy $y = a + bx$ ta sẽ thấy công thức khác với trên:

$$a = \bar{y} - b\bar{x}$$

$$b = r\frac{s_y}{s_x}$$

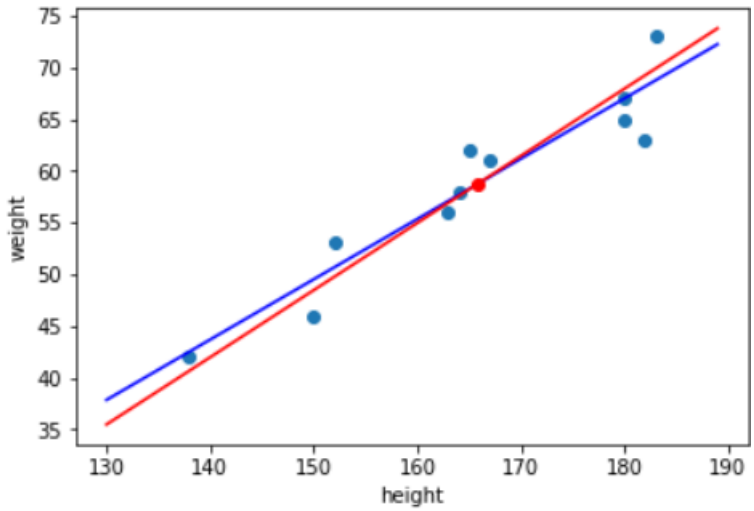
Những ai có hứng thú với thống kê, hãy thử tính toán xem sao. Nói cách khác, việc hoán đổi biến giải thích và biến mục đích sẽ tạo ra một đường hồi quy khác.

15.2 Sử dụng Python để vẽ hai đường hồi quy

Ta sẽ sử dụng code trong Bài học 14 để vẽ hai đường hồi quy.

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
%matplotlib inline

weight = np.array([42, 46, 53, 56, 58, 61, 62, 63, 65, 67, 73])
height = np.array([138, 150, 152, 163, 164, 167, 165, 182, 180, 180, 183])

# reg1: Duong hoi quy tu height du doan weight
X = np.expand_dims(height, axis=-1)
y = weight
reg1 = LinearRegression()
reg1.fit(X, y)

# reg2: Duong hoi quy tu weight du doan height
X = np.expand_dims(weight, axis=-1)
y = height
reg2 = LinearRegression()
reg2.fit(X, y)

# Tao danh sach gia tri truc x de ve duong thang
x = np.arange(130, 190, 1)
# Ve cac gia tri duoc su dung de hoc trong mot bieu do phan tan
plt.scatter(height, weight)
# Ve reg1
plt.plot(x, x*reg1.coef_+reg1.intercept_, 'b')
# Ve reg2 Chu y: (Bien doi x=a'+b'y thanh: y=(x-a')/b')
plt.plot(x, (x-reg2.intercept_)/reg2.coef_, 'r')
#Diem giao nhau giua reg1 va reg2
#la diem trung binh cua can nang weight va chieu cao height
plt.plot(np.mean(height), np.mean(weight), 'ro')
plt.xlabel('height')
plt.ylabel('weight')
```

Trục ngang **`height`** và trục dọc **`weight`** do đó khi vẽ đường thẳng $x = a' + b'y$ ta cần biến đổi nó thành $y = \dfrac{(x - a')}{b'}$. Khi thực thi code trên ta sẽ có kết quả:



Đường màu xanh là đường hồi quy dự đoán trọng lượng từ chiều cao, đường màu đỏ là đường hồi quy dự đoán chiều cao từ trọng lượng.
Chúng ta thấy rằng hai đường hồi quy này không trùng khớp, chúng là hai đường thẳng khác nhau. Tuy nhiên đường thẳng nào cũng đi qua tọa độ $(\bar{x}, \bar{y})$, và đây chính là điểm giao nhau. Chúng ta lưu ý rằng kết quả của đường hồi quy thay đổi phụ thuộc vào việc chúng ta lấy cái gì làm biến mục đích.
Nó có vẻ hiển nhiên nhưng khi bạn làm việc với nhiều dữ liêu khác nhau và tìm thấy hệ số hồi quy, thật ngạc nhiên khi bạn thay đổi các biến mà giữ nguyên hệ số hồi quy trước đó mà không suy nghĩ quá nhiều, nhưng việc sử dụng nó sẽ không còn chính xác, hãy cẩn thận.

15.3 Tổng kết

Trong bài học này chúng ta đã vẽ hai đường hồi quy từ hai biến số.
Thông thường thì **công thức đường hồi quy dự đoán y từ x thì không sử dụng được trong việc dự đoán x từ y**. Vì vậy các bạn hãy lưu ý.
Trong bài học tới chúng ta sẽ nói về hệ số quyết định. Bằng việc sử dụng hệ số quyết định, có thể đánh giá mức độ phù hợp của đường hồi quy thu được với dữ liệu. Nó là một chỉ số thường được sử dụng trong luận văn, nhưng tôi nghĩ rằng ít người có thể hiểu đúng về nó. Chúng ta hãy học một cách đúng đắn và có hệ thống.

Bài 16

Hệ số quyết định R2

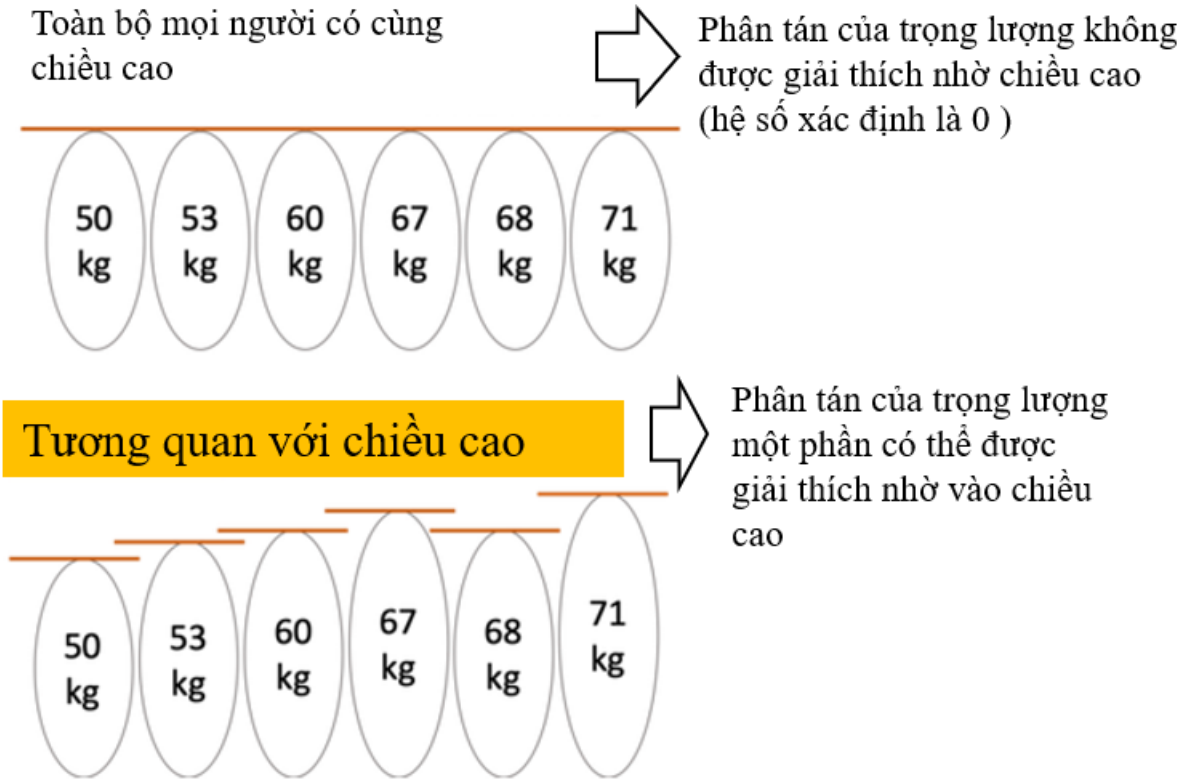
Trong Bài học 15, chúng ta đã biết về hồi quy tuyến tính (đường thẳng hồi quy). Trong bài học này, liên quan tới phân tích hồi quy, tôi sẽ giải thích chỉ số quan trọng, đó là **hệ số quyết định**.
Hệ số quyết định không chỉ quan trọng trong thống kê học mà còn quan trọng trong Machine Learning.

16.1 Hệ số quyết định là gì?

Hệ số quyết định là chỉ số quyết định giá trị của biến giải thích là bao nhiêu cho giá trị của biến mục đích.
Nghe có vẻ khó hiểu, nhưng thực ra không khó đâu. Chúng ta hãy xem xét hai người có trọng lượng lần lượt là 50kg và 60kg.
Từ sự chênh lệch trọng lượng này thì chênh lệch chiều cao sẽ như thế nào? Giả sử nếu hai người có cùng chiều cao, như vậy chiều cao không có mối quan hệ nào với trọng lượng. Nhưng giả sử người nặng 50kg có chiều cao là 165cm, người nặng 60kg có chiều cao là 170cm, khi đó có thể nói rằng sự chênh lệch trọng lượng một phần là vì có sự chênh lệch về chiều cao.
Tôi xin đưa một ví dụ khác.
Ví dụ có một nhóm người có trọng lượng lần lượt là 50kg, 53kg, 60kg, 67kg, 68kg, 71kg. Từ sự chênh lệch về trọng lượng này, câu hỏi là chiều cao giải thích bao nhiêu cho sự thay đổi trọng lượng này? Đó là ý nghĩa của hệ số xác định.

> Tôi hoàn toàn không hiểu gì. Nếu tất cả mọi người có chiều cao giống nhau thì sao?

Chiều cao không nhất thiết giải thích cho sự phân tán phân bố của trọng lượng. Trong trường hợp này hệ số xác định sẽ là 0.
Thế nhưng trong trường hợp người có trọng lượng lớn thì chiều cao cũng lớn, khi đó có thể nói phân bố của trọng lượng có phần nào đó phụ thuộc vào chiều cao.
Tóm lại nó có nghĩa là, **phân tán y được giải thích bao nhiêu từ x.**

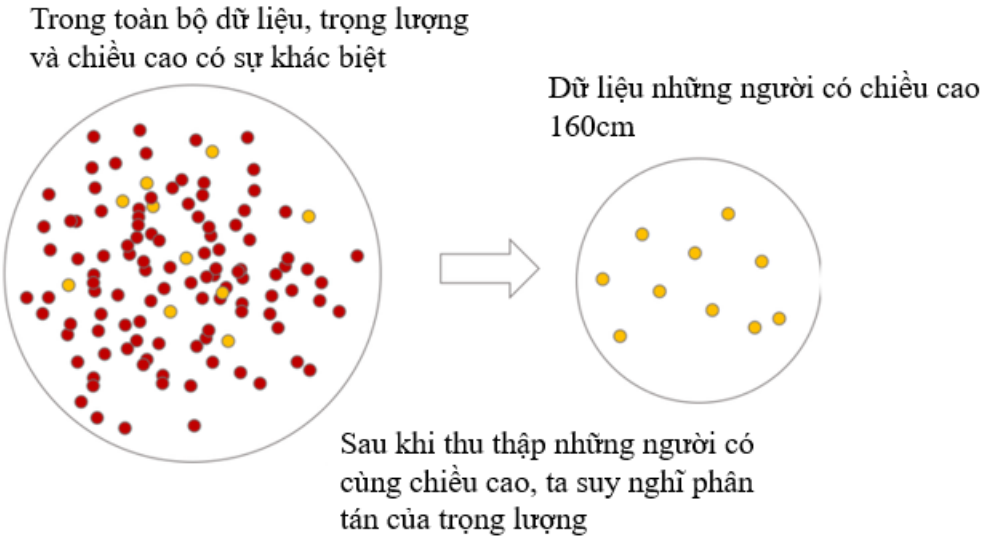


16.2 Định nghĩa hệ số quyết định

Như các ví dụ trước đây, ta coi x là biến chiều cao và y là biến trọng lượng. Xét dữ liệu có n người, chúng ta chỉ quan tâm tới trọng lượng mà không để ý tới chiều cao. Chúng ta có thể tính toán phân tán trọng lượng như dưới đây:

$$s_y^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2$$

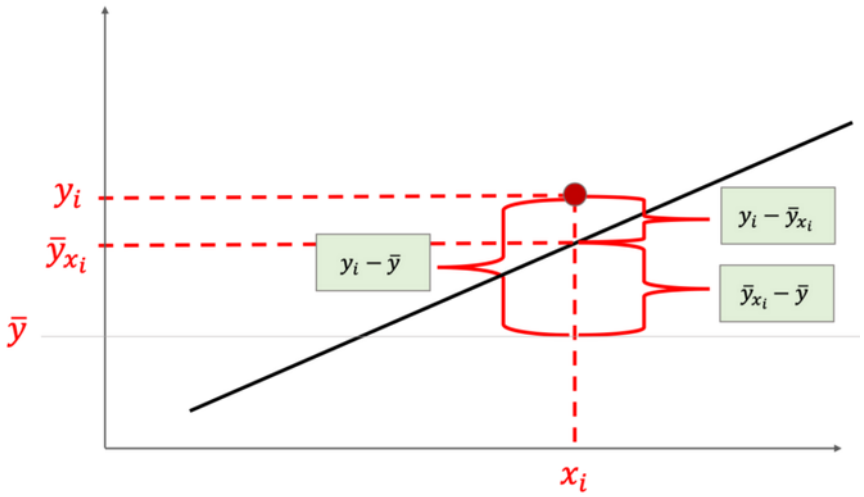
Tóm lại, phân tán của trọng lượng sẽ là trung bình cộng tổng bình phương độ lệch từ các điểm dữ liệu (trọng lượng) tới giá trị trung bình $\bar{y}$.
Tuy nhiên một phần của s_y^2 chắc chắn phụ thuộc vào sự sai khác của chiều cao x. Cái tỷ lệ đó được gọi là hệ số xác định.
Vậy thì trong s_y^2 được giải thích bao nhiêu phần nhờ vào chiều cao x, làm thế nào để biết được điều này?
Để biết điều này có lẽ chúng ta nên xem xét phân tán của trọng lượng y không được giải thích bởi chiều cao x, sau đó lấy s_y^2 trừ cho nó.
Vậy thì chúng ta cần tìm, phân tán trọng lượng y không được giải thích nhờ vào chiều cao x.
Để tìm điều này, chúng ta nghĩ tới, khi mà tất cả mọi người có cùng chiều cao thì trọng lượng của họ phân bố như thế nào?
Ví dụ, trong dữ liệu, chúng ta thu thập những người có chiều cao là 160cm. Sau đó tính phân tán của trọng lượng trong mẫu dữ liệu này.



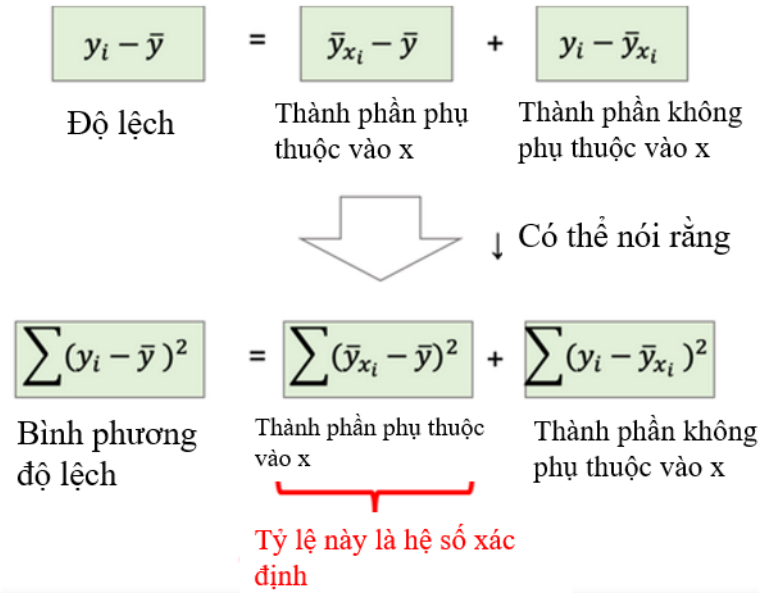
Phân tán trọng lượng của nhóm người có chiều cao 160cm, cái phân tán này sẽ nhỏ hơn phân tán trọng lượng của toàn bộ dữ liệu là s_y^2.
Như vậy ta có thể suy nghĩ rằng cái nhóm như vậy sẽ có phân tán trọng lượng không bị ảnh hưởng bởi chiều cao.
Ứng dụng điều này, công thức $s_{y \cdot x}$ cho phân bố của những người có cùng chiều cao:

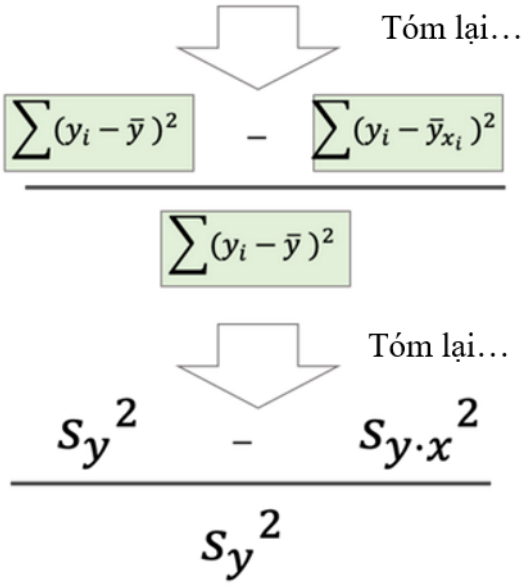
$$s_{y \cdot x} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \bar{y}_{x_i})^2$$

$\bar{y}_{x_i}$ là trọng lượng trung bình ứng với những người có chiều cao là x_i. Tóm lại ta tính độ lệch trọng lượng với người thứ i, không phải từ giá trị trọng lượng trung bình $\bar{y}$ của tất cả mọi người, mà ở đây ta tính độ lệch từ trọng lượng trung bình $\bar{y}_{x_i}$ của những người có cùng chiều cao x_i.
Nó có vẻ hơi khó hiểu nhưng bạn không cần phải nhớ công thức này. Điều quan trọng là $s_{y \cdot x}$ là thành phần không bị phụ thuộc vào chiều cao trong phân tán của trọng lượng.
Trong Bài học 13, chúng ta có đường hồi quy là câu chuyện trung bình có điều kiện. Có nghĩa là với $x = x_i$ ta tính được $y = a + bx_i$, giá trị y này là giá trị trung bình của y khi $x = x_i$.
Tóm lại, trọng lượng trung bình $\bar{y}_{x_i}$ trong những người có cùng chiều cao x_i được tính toán bằng đường hồi quy $y = a + bx_i$. Nếu bạn hiểu điều này, hãy thử lý giải hình vẽ dưới đây xem sao?



Hình vẽ này là đường hồi quy dự đoán giá trị y từ giá trị x.
Điểm màu đỏ có tọa độ (x_i, y_i) là một dữ liệu thực tế. Độ lệch của y_i là $y_i - \bar{y}$ được chia làm các phần là $\bar{y}_{x_i} - \bar{y}$ và $y_i - \bar{y}_{x_i}$.
Giá trị $\bar{y}_{x_i} - \bar{y}$ là độ lệch tính từ giá trị trung bình $\bar{y}$ của y bằng đường hồi quy. Tóm lại có thể nói đó là độ lệch từ giá trị trung bình của y khi $x = x_i$.
Trong ví dụ về trọng lượng về chiều cao, giá trị giá trị trọng lượng trung bình của những người có chiều cao x_i là $\bar{y}_{x_i}$, độ lệch $y_i - \bar{y}$ là thành phần phụ thuộc vào chiều cao.
Hãy nhìn vào hình vẽ trên ở điểm màu đỏ (x_i, y_i), giá trị y_i ở trên một chút so với vị trí của $\bar{y}_{x_i}$.
Điều này có nghĩa là trọng lượng của người này nặng hơn một chút so với trọng lượng trung bình của những người có cùng chiều cao với người này. Độ lệch so với giá trị trung bình này $(y_i - \bar{y}_{x_i})$ là thành phần không phụ thuộc vào chiều cao.
Một cách dễ hình dung, ta lấy ví dụ người có trọng lượng là 60kg và chiều cao 160cm. Nếu nói "Tôi nặng vì tôi cao", thế nhưng cân nặng trung bình so với chiều cao của bạn chỉ là 50kg, vậy phần còn lại là 10kg là một phần không thể giải thích bằng chiều cao, tức không phụ thuộc vào chiều cao.





Tôi sẽ bỏ qua phần chứng minh vì nó không quan trọng nhưng hãy nhớ rằng từ phân tích độ lệch trong hình trên dẫn đến việc phân tích độ phân tán (phương sai).
Từ phân tích trên, trong phân tán của y thành phần được giải thích nhờ x (phụ thuộc vào x) có thể được định nghĩa, đó là hệ số quyết định **(coefficient of determination)**, thường được gọi là R^2 **(R squared)**.

$$R^2 = \frac{s_y^2 - s_{y \cdot x}^2}{s_y^2} = 1 - \frac{s_{y \cdot x}^2}{s_y^2}$$

Nếu đường hồi quy thu được bằng phương pháp bình phương nhỏ nhất thì R này bằng với hệ số tương quan r. Hãy ghi nhớ điều rất quan trọng này.
$R^2 = 0$ có nghĩa là $s_y^2 = s_{y \cdot x}^2$. Tóm lại x hoàn toàn không giải thích được y, tức là y không phụ thuộc vào x.
Ngược lại $R^2 = 1$ nghĩa là $s_{y \cdot x}^2 = 0$, như hình vẽ đường hồi quy lúc trước, ta thấy rằng trong trường hợp này với mọi i thì $y_i - \bar{y}_{x_i}$ là 0, điều đó có nghĩa là toàn bộ điểm dữ liệu đều nằm trên đường hồi quy, nói cách khác trong trường hợp này ta có tương quan thuận hoàn toàn, trường hợp này có thể nói x **hoàn toàn quyết định** y.

16.3 Tổng kết

Trong bài học này tôi đã giải thích về hệ số xác định R^2.

- Hệ số quyết định là chỉ số giải thích biến số mục đích y phụ thuộc vào biến x bao nhiêu.
- Độ lệch (Phân tán) của y được chia ra làm các phần, phần phụ thuộc vào biến x và phần không phụ thuộc vào biến x.
- Hệ số quyết định R^2 được định nghĩa là thành phần tỷ lệ có thể giải thích trong độ phân tán của y có phần phụ thuộc vào biến x.

- Nếu đường hồi quy thu được bằng phương pháp bình phương tối thiểu thì hệ số quyết định R^2 có R bằng với hệ số tương quan r.

Nếu bài học này mà bạn chưa hiểu thì bạn hãy đọc lại vài lần nhé.
Trong bài học tiếp theo chúng ta sẽ xem xét hệ số xác định được sử dụng như thế nào, cũng như cách tính hệ số xác định bằng Python ra sao.

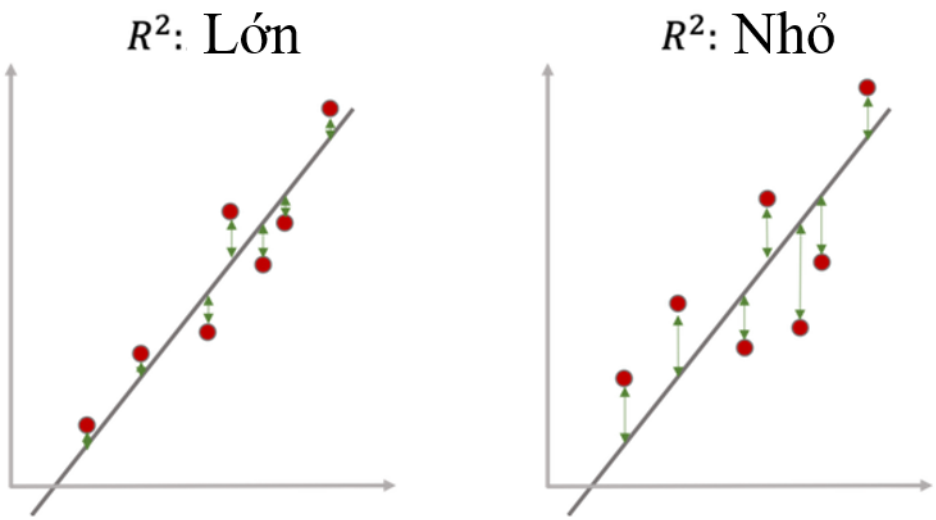
Bài 17

Đánh giá độ chính xác của phương trình hồi quy với hệ số quyết định R2

Trong Bài học 16, chúng ta đã biết hệ số quyết định R^2 là chỉ số cho biết biến số giải thích x diễn giải biến số mục đích y là bao nhiêu. Nói cách khác biến số mục đích y phụ thuộc vào biến số giải thích x như thế nào.
Giải thích này là chính xác khi chỉ nhìn vào công thức xác định hệ số quyết định R^2, nhưng khi sử dụng hệ số quyết định R^2 trong công việc, nó được sử dụng như một chỉ số thể hiện độ chính xác của phương trình hồi quy.
Tại sao hệ số quyết định R^2 có thể được sử dụng như một chỉ số về độ chính xác của phương trình hồi quy? Lần này tôi muốn giải thích cho các bạn biết điều này.
Nếu đã đọc bài viết trước thì việc tiếp cận bài học này sẽ dễ hiểu hơn, vì vậy nếu chưa đọc bài viết trước, thì các bạn hãy đọc lại nhé.

17.1 Hế số quyết định đánh giá đường hồi quy là phù hợp hay không

Đầu tiên bạn hãy nhìn và so sánh hai biểu đồ dưới đây.



Đường hồi quy bên trái có độ lệch từ các điểm dữ liệu tới nó là nhỏ hơn so với đường hồi quy bên phải. Từ công thức về hệ số quyết định R^2 nêu trong bài học trước, khi $y_i - \bar{y}_{x_i}$ nhỏ thì $s_{y \cdot x}$ sẽ nhỏ.

$$s_{y \cdot x} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \bar{y}_{x_i})^2$$

$$R^2 = \frac{s_y^2 - s_{y \cdot x}^2}{s_y^2} = 1 - \frac{s_{y \cdot x}^2}{s_y^2}$$

Như vậy có thể thấy $s_{y \cdot x}$ nhỏ tức là, **thành phần trong y mà không được giải thích bởi x (không phụ thuộc vào x) là nhỏ**. Từ công thức định nghĩa hệ số quyết định R^2, có nghĩa là có nhiều thành phần trong y có thể được giải thích bởi x khi đó **hệ số quyết định R^2 sẽ lớn**.

Chúng ta hãy nhìn vào hình trên một lần nữa, tôi nghĩ rằng so với hình bên phải thì đường hồi quy bên trái phù hợp với dữ liệu hơn.

"**Đường hồi quy phù hợp với dữ liệu**" và "**hệ số quyết định R^2 lớn**", điều này có nghĩa là hệ số quyết định R^2 có thể được sử dụng để phán đoán đường hồi quy là phù hợp với dữ liệu hay không.

Bạn có thể thấy rằng "mức độ phù hợp của đường hồi quy với dữ liệu" đại diện cho "đường hồi quy có thể sử dụng để dự đoán tốt như thế nào".

Tất nhiên sẽ không có ý nghĩa gì nếu chúng ta nhìn vào dữ liệu và tự vẽ đường hồi quy sao cho trông nó có vẻ tốt, đúng không nào.

Đúng là như thế, với dữ liệu mà chúng ta chưa biết, để đánh giá đường hồi quy là tốt hay không bằng hệ số quyết định, chúng ta có thể đưa ra đánh giá đường hồi quy đó phù hợp với dữ liệu này hay không.

17.2 Dữ liệu huấn luyện và dữ liệu đánh giá

Chúng ta sẽ nói một chút về Machine Learning. Về lĩnh vực này tôi sẽ giải thích chi tiết sau, trước hết tôi muốn nói về các khái niệm cơ bản. Thông thường, khi chúng ta xây dựng một mô hình dự đoán cho Máy Học (Machine Learning), chúng ta sẽ sử dụng

dữ liệu huấn luyện (Data Training) để cho máy nhận biết với dữ liệu input như thế thì kết quả mong muốn là gì, dữ liệu này ta gọi là **dữ liệu huấn luyện** (Data Training), bao gồm dữ liệu đầu vào và cả kết quả mong muốn. Và để đánh giá mô hình này tốt hay không, ta sẽ đưa **dữ liệu đánh giá** vào để máy luyện tập và đưa ra kết quả.

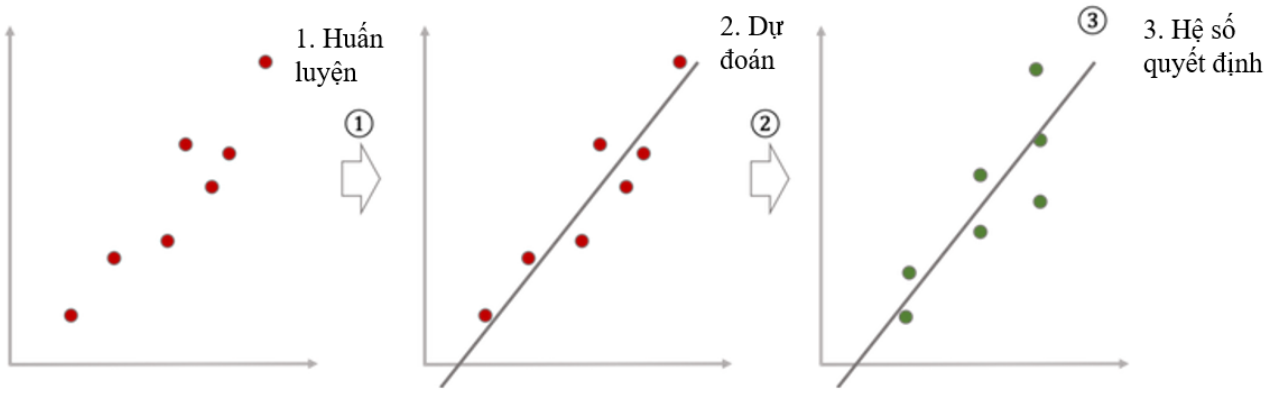
- Dữ liệu huấn luyện: Được sử dụng để cho máy học khi xây dựng mô hình dự đoán.
- Dữ liệu đánh giá: Là dữ liệu không sử dụng khi cho máy học trong khi xây dựng mô hình dự đoán, chúng ta sử dụng để đo đếm sự chính xác khi máy đưa ra dự đoán dựa trên dữ liệu này.

Vì dữ liệu huấn luyện đã được sử dụng trong thời điểm huấn luyện cho máy học, nên khả năng cao là mô hình dự đoán sẽ cho kết quả rất chính xác với dữ liệu này. Do đó, để đo đếm sự chính xác của mô hình máy học này, ta cần đưa vào một dữ liệu mà nó chưa hề biết, và xem nó dự đoán chính xác được bao nhiêu.
Do đó khi đánh giá đường hồi quy bằng hệ số quyết định R^2, chúng ta xem xét mức độ phù hợp với dữ liệu đánh giá chứ không phải mức độ phù hợp với dữ liệu huấn luyện.

17.3 Tính toán hệ số quyết định bằng dữ liệu đánh giá

Nào, bây giờ chúng ta đã tính toán được đường hồi quy từ dữ liệu huấn luyện, vậy thì hãy thử đánh giá đường hồi quy đó chính xác như thế nào với dữ liệu đánh giá bằng cách tính toán hệ số quyết định.
Các bước làm theo quy trình dưới đây:

- Sử dụng dữ liệu huấn luyện để tạo ra đường hồi quy.
- Sử dụng dữ liệu đánh giá để tính giá trị dự đoán của mô hình hồi quy.
- Tính hệ số quyết định bằng cách sử dụng giá trị dự đoán với giá trị dữ liệu đánh giá thực tế.



Trong hình trên, điểm màu đỏ là dữ liệu huấn luyện, điểm màu xanh là dữ liệu đánh giá.
Một lần nữa chúng ta hãy nhìn lại công thức tính hệ quyết định:

$$s_{y \cdot x} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \bar{y}_{x_i})^2$$

$$s_y^2 = \frac{1}{n} \sum_{i=1}^{n} (y_i - \bar{y})^2$$

$$R^2 = \frac{s_y^2 - s_{y \cdot x}^2}{s_y^2} = 1 - \frac{s_{y \cdot x}^2}{s_y^2}$$

Chúng ta có thể thay bằng cách viết như sau:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \bar{y}_{x_i})^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$$

$\bar{y}_{x_i}$ là giá trị được tính toán dựa vào công thức hồi quy $y = a + bx_i$ mà tôi đã giải thích trước đây, có thể nói đây là **giá trị dự đoán**.
Nếu ta gọi các giá trị dự đoán dựa vào công thức hồi quy là $\hat{y}$, dữ liệu thực tế dùng để đánh giá là y_i có giá trị trung bình là $\bar{y}$, công thức hệ số quyết định sẽ được viết lại thành:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$$

Công thức này giống với công thức đã giới thiệu trong bài học trước.

Nhìn vào công thức này nhiều lần các bạn sẽ cảm thấy quen mắt.

Hừm, tôi đang bối rối bởi có quá nhiều công thức.

Tiếp theo chúng ta sẽ thử tính hệ số quyết định bằng Python.
Khi học lý luận về thống kê, đôi lúc các bạn sẽ cảm thấy khó hiểu, tôi cũng có lúc như vậy.
Những lúc như vậy, chúng ta viết code và thự mình kiểm chứng. Tự mình lập nên trình tự các bước cần làm rồi sau đó viết code theo trình tự ấy, các bạn có thể lý giải được.

17.4 Tính hệ số quyết định bằng Python

Để tính hệ số quyết định bằng Python, chúng ta sẽ sử dụng hàm `r2_score()` trong Module `metrics` của `scikit−learn`.
`scikit−learn` là một thư viện được biết đến rộng rãi trong Machine Learning.
Nếu bạn biết rằng hệ số quyết định được dùng như là một chỉ số để đánh giá mô hình hồi quy, tôi nghĩ bạn có thể lý giải được hàm tìm hệ số xác định nằm trong Module `metrics` của `scikit−learn`.
Nào, bây giờ chúng ta sẽ code theo 3 bước như đã nói trước đây.

- Sử dụng dữ liệu huấn luyện để tạo ra đường hồi quy.

- Sử dụng dữ liệu đánh giá để tính giá trị dự đoán của mô hình hồi quy.
- Tính hệ số quyết định bằng cách sử dụng giá trị dự đoán với giá trị dữ liệu đánh giá thực tế.

Lần này, dữ liệu huấn luyện và dữ liệu đánh giá đều là các dữ liệu giả lập. Một cách khác thường được làm đó là sử dụng 80% dữ liệu nào đó làm dữ liệu đào tạo và 20% dữ liệu còn lại được sử dụng như dữ liệu đánh giá. Những vấn đề xung quanh phương pháp đó sẽ giới thiệu chính thức trong phần Machine Learning.
Quay trở lại công việc của chúng ta, vẫn như mọi khi, chúng ta sẽ tạo ra mô hình hồi quy dự đoán trọng lượng từ chiều cao. Trong đó chiều cao x (height) và y là trọng lượng (weight).

Bước 1: Sử dụng dữ liệu huấn luyện để tạo ra đường hồi quy

```
import numpy as np
from sklearn.linear_model import LinearRegression
weight = np.array([42, 46, 53, 56, 58, 61, 62, 63, 65, 67, 73])
height = np.array([138, 150, 152, 163, 164, 167, 165, 182, 180, 180, 183])
# shape tra ve (So luong du lieu, so luong thuoc tinh) cua ndarray
X = np.expand_dims(height, axis=-1)
y = weight
# Tao mo hinh duong hoi quy
reg = LinearRegression()
# Huan luyen
reg.fit(X, y)
```

Code trên đã được giải thích trong Bài học 14.

Bước 2: Sử dụng dữ liệu đánh giá để tính giá trị dự đoán của mô hình hồi quy

Lần này, dữ liệu đánh giá (weight_test, height_test) đã được tạo ra một cách thích hợp. Từ mô hình đã được huấn luyện, chúng ta tìm giá trị dự đoán bằng cách sử dụng hàm `.predict()` đã từng được nói tới trong Bài học 14.

```
#Data danh gia
weight_test = np.array([43, 45, 50, 58, 58, 60, 66, 63, 65, 70, 72])
height_test = np.array([140, 148, 152, 163, 160, 170, 163, 177, 177, 185, 180])
#Gia tri du doan
X_test = np.expand_dims(height_test, axis=-1)
y_test_pred = reg.predict(X_test)
```

Bước 3: Tính hệ số quyết định bằng cách sử dụng giá trị dự đoán với giá trị dữ liệu đánh giá thực tế

Nào, bây giờ chúng ta sẽ tính hệ số quyết định.
Chú ý lần này chúng ta sử dụng dữ liệu đánh giá với giá trị dự đoán của nó. Chúng ta không sử dụng dữ liệu huấn luyện.
Chúng ta sẽ sử dụng hàm `r2_score()` của Module `metrics` của `scikit−learn`.

```
from sklearn.metrics import r2_score
# He so quyet dinh
r2_score(weight_test, y_test_pred)
```

```
0.8586682113144612
```

Thật đơn giản phải không nào. Những ai có hứng thú có thể tham khảo về trực quan hóa dữ liệu visualization trong Bài học 15.
Không có cơ sở chính xác về hệ số quyết định, về mặt cảm giác có thể nói rằng:

- Dưới 0.6 : Khó có thể nói rằng dự đoán là chính xác
- Từ 0.6 đến 0.8 : Phản ánh đúng hiện thực
- Từ 0.8 đến 0.8 : Độ chính xác cao
- Từ 0.9 trở lên: Khả năng trang bị quá mức kỳ vọng (Over fitting)

Bổ sung

Over fitting có nghĩa là đối với dữ liệu có tính chất đặc thù nó vẫn có thể đưa ra dự đoán có độ chính xác cao. Lĩnh vực này sẽ được đề cập trong phần Machine Learning.

17.5 Tổng kết

Trong bài học này, tôi đã giải thích về hệ số quyết định với tư cách như là một chỉ số đánh giá mô hình hồi quy.
Trong công việc thực tế nó được sử dụng với tư cách như là một chỉ số đánh giá mô hình hồi quy.
Tuy nhiên cũng giống như cách nghĩ đã nói trước đây, biến số giải thích x sẽ giải thích được bao nhiêu về biến số mục đích y, chúng ta chú ý dữ liệu mục tiêu (dữ liệu y) sẽ trở thành dữ liệu để đánh giá.

- Có thể sử dụng hệ số quyết định R^2 là chỉ số đánh giá mô hình hồi quy.
- Hệ số quyết định R^2 là chỉ số đánh giá mức độ phù hợp với dữ liệu đánh giá của mô hình hồi quy.
- Có thể tính hệ số quyết định bằng cách sử dụng hàm `r2_score()` trong Module `metrics` của `scikit−learn`.

Như vậy ở bài viết này, phạm vi liên quan tới thống kê mô tả đã kết thúc.
Từ bài viết tới sẽ là phần cuối (phần kết). Có thể nói bản chất của phần này là lĩnh vực thống kê suy luận.
Khi nói tới Thống Kê Học, thông thường nó có nghĩa là Thống Kê Suy Luận. Do đó, nó là phần có giá trị trọng tâm của thống kê, vì vậy hãy tiếp tục đọc tiếp phần sau.

Bài 18

Nhất định hiểu về biến số xác suất, phân bố xác suất, mật độ xác suất

Cho tới Bài học 17, chúng ta đã tìm hiểu xong các vấn đề về thống kê mô tả.
Trong thống kê học có rất nhiều chỉ số, nhưng từ bài viết này, tôi sẽ đề cập tới một lĩnh vực thống kê khác, đó là "thống kê suy luận".
Khi hiểu về thống kê suy luận, với dữ liệu tiêu bản (dữ liệu mẫu) có trong tay, chúng ta có thể biết được các đặc tính của dữ liệu cha. Đây cũng là lý do tại sao hiện tại có nhiều người nghiên cứu thống kê đến thế.
Ví dụ, để điều tra về sức bền của một sản phẩm, ta tiến hành thả rơi sản phẩm đó nhiều lần, đương nhiên chúng ta không thể làm rơi tất cả những sản phẩm mà chúng ta có, như thế thì hỏng hết sản phẩm, còn có thể bán cho ai được nữa, đúng không nào. Ví dụ, chúng ta lấy ngẫu nhiên ra 100 sản phẩm và tiến hành thực nghiệm sức bền của các sản phẩm này.
Một ví dụ khác, để tìm hiểu mức độ thỏa mãn của người dùng về sản phẩm của công ty mình, rõ ràng việc điều tra tất cả người dùng là điều không thể nào làm được, do đó chúng ta sẽ lấy phản hồi của một bộ phận người dùng, dựa trên cơ sở điều tra đó ta có thể đo đạc được độ thỏa mãn của toàn thể người dùng đối với sản phẩm của công ty của mình.
Công việc như thế, **tức là lấy một phần dữ liệu (sản phẩm) từ dữ liệu cha, điều tra dữ liệu mẫu này để đo đạc ra đặc tính của dữ liệu cha** là việc có thể làm được và vô cùng cần thiết trong cuộc sống.
Trong ví dụ trên, khả năng tất cả 100 sản phẩm thả rơi đều bị lỗi không phải là không có.
Những khái niệm không thể thiếu trong thống kê học, đó là **biến số xác suất**, **phân bố xác suất** sẽ được tôi trình bày trong bài học này. Các thuật ngữ này sẽ được sử dụng thường xuyên trong các bài học thống kê từ nay về sau, hãy kiên nhẫn để hiểu nó là gì nhé.

18.1 Biến số xác suất là gì?

Biến xác suất là khái niệm đã được nói tới, nó có tên tiếng anh là random variable. Biến số xác suất (biến ngẫu nhiên) tưởng chừng khó hiểu vì tên gọi dễ gây nhầm lẫn nhưng thực ra không có gì khó cả.

Đơn giản chúng ta hãy nghĩ rằng biến xác suất (biến ngẫu nhiên) có giá trị dao động ngẫu nhiên.

Ví dụ khi ta tung xúc sắc, thì số điểm trên mặt con xúc sắc là biến xác suất. Nó có giá trị từ 1 tới 6. Khi tung xúc sắc thì xác suất của mỗi mặt xúc sắc xuất hiện đều là $\frac{1}{6}$.

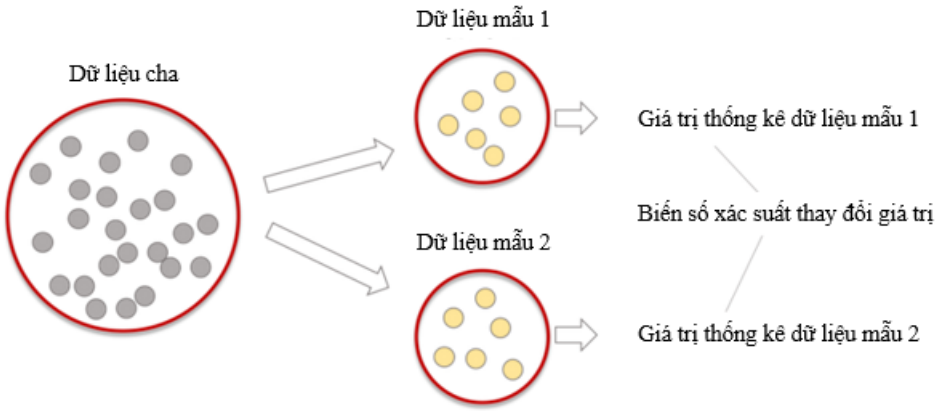
Khi thảo luận về các sự kiện có tính dao động(có xác suất), sẽ rất tiện lợi khi có khái niệm về biến xác suất (biến ngẫu nhiên) vì nó giúp chúng ta dễ dàng thảo luận với nhau hơn.

Và một điều quan trọng trong thông kê học các bạn cần biết, đó là thống kê dữ liệu mẫu cũng là một biến ngẫu nhiên.

Thống kê dữ liệu mẫu là cái gì cơ?

Thống kê dữ liệu mẫu đó là chúng ta đo đạc đặc tính của một quần thể (dữ liệu cha) bằng cách tìm đặc tính của dữ liệu mẫu (dữ liệu được trích xuất từ dữ liệu cha). Ví dụ, các chỉ số như giá trị trung bình, phân tán, độ lệch chuẩn ... Những chỉ số này đã được giới thiệu trong các bài học trước đây.

Từ dữ liệu cha, ta lấy ngẫu nhiên dữ liệu mẫu, giá trị khi ta thực hiện thống kê dữ liệu mẫu tất nhiên sẽ thay đổi tùy thuộc vào dữ liệu mẫu được trích xuất, do đó nó là một biến xác suất.



Ví dụ ta coi những người đàn ông trưởng thành trong đất nước Việt Nam là dữ liệu cha, trong đó ta lấy ngẫu nhiên một số người đàn ông trưởng thành, ta gọi đây là dữ liệu mẫu. Ta thực hiện lấy 2 lần nên sẽ có hai dữ liệu mẫu.

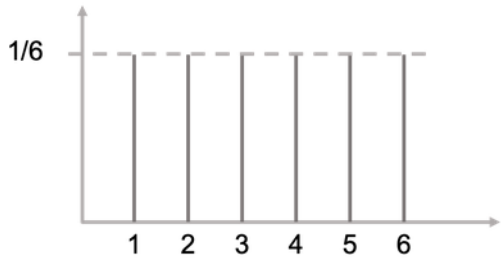
Giả định rằng chiều cao trung bình của những người đàn ông trưởng thành trên toàn quốc là 172 cm. Giá trị chiều cao trung bình của hai dữ liệu mẫu (trung bình dữ liệu mẫu) đương nhiên không có chuyện vừa khít con số 172 cm. Mỗi dữ liệu mẫu sẽ có giá trị trung bình có độ lệch nhất định so với con số 172 cm.

Tuy nhiên, khả năng mà giá trị trung bình của dữ liệu mẫu sai khác lớn so với giá giá trung bình của dữ liệu cha (172 cm) là rất thấp. Ví dụ ít có khả năng sẽ là 160 cm hay

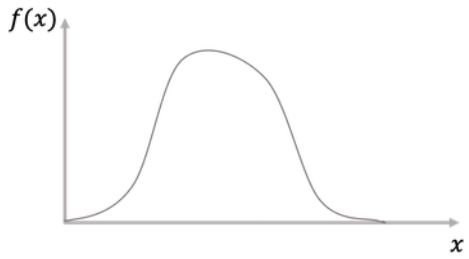
180cm, mà có thể ở trước và sau con số 172 cm một chút, chẳng hạn là 170 cm, cái khuynh hướng ấy sẽ có khả năng cao hơn.
Nói cách khác, có thể nói rằng trung bình mẫu dao động ngẫu nhiên. Hay giá trị trung bình của dữ liệu mẫu là một biến xác suất.
Cách nghĩ cho rằng thống kê mẫu là biến ngẫu nhiên dao động là rất quan trọng, vì vậy xin hãy lưu ý.

18.2 Phân bố xác suất là gì?

Phân bố xác suất (probability distribution) nói một cách đơn giản đó là phân bố biểu diễn xác suất ứng với mỗi giá trị của biến ngẫu nhiên (biến xác suất).
Ta hãy quay trở lại ví dụ về tung con xúc sắc.
Con xúc sắc có các mặt, chúng có các điểm lần lượt là 1 đến 6 điểm, xác suất cho mỗi mặt này xuất hiện khi tung con xúc sắc đều là $\frac{1}{6}$. Do đó phân bố xác suất có thể được biểu diễn như sau.



Đây không phải là điều gì khó hiểu. (Nhân tiện, hãy nhớ rằng phân bố xác suất cho mỗi giá trị mà xác suất của mỗi giá trị này đều giống nhau thì được gọi là hay phân bố đồng đều(uniform distribution). Hãy ghi nhớ điều này.)
Trường hợp các giá trị rời rạc giống như số điểm của con xúc sắc (từ 1 đến 6 diểm), chúng không có tính liên tục, khi đó biến ngẫu nhiên được gọi là các biến ngẫu nhiên rời rạc. Mặt khác một biến ngẫu nhiên nhận các giá trị liên tục được gọi là biến ngẫu nhiên liên tục, và phân bố xác suất trong trường hợp đó cũng là một đường liên tục, ví dụ như đường cong dưới đây.



Đây là đường biểu diễn phân bố xác suất đối với biến ngẫu nhiên (biến xác suất) x. Có thể nói rằng đó là một hàm số với biến xác suất x.
Hàm số này được gọi là hàm mật độ xác suất (probability density function). Người ta hay nói ngắn gọn là **mật độ xác suất**.
Đây là một thuật ngữ vô cùng quan trọng mà tôi muốn các bạn nhớ tuy nhiên

về hàm mật độ xác suất thì tôi sẽ giải thích sau.

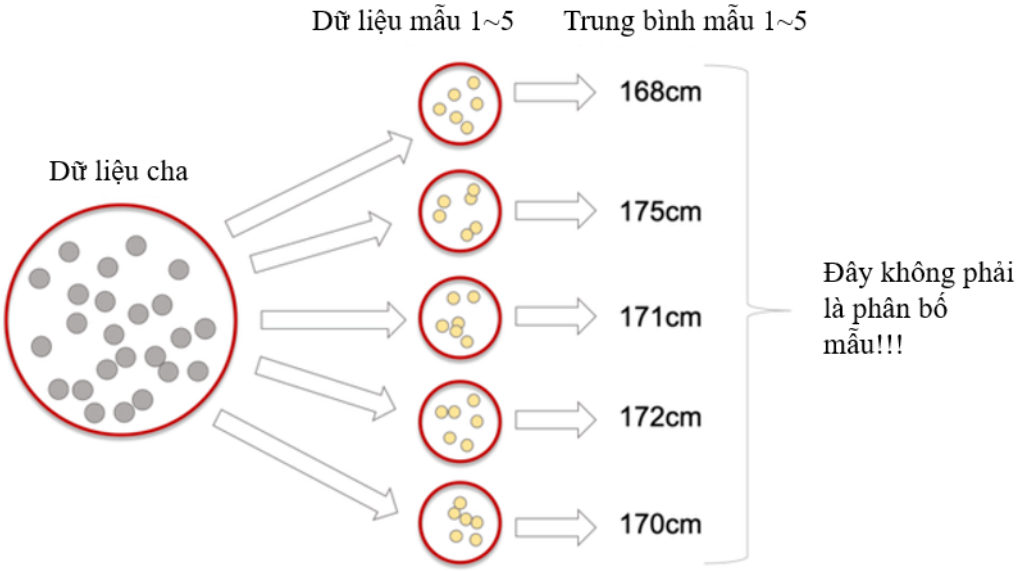
Chúng ta quay trở lại với ví dụ trước đây, thử nhìn vào giá trị trung bình của dữ liệu mẫu đối với chiều cao của những người đàn ông trưởng thành ở Việt Nam. Giá trị thống kê từ dữ liệu mẫu là biến số xác suất tức là thống kê mẫu là một biến ngẫu nhiêu, điều đó có nghĩa là một thống kê mẫu sẽ tuân theo một phân bố xác suất nhất định đúng không nào?

Phân bốc xác suất của giá trị thống kê dữ liệu mẫu được gọi là phân bố mẫu hay phân bố lấy mẫu (sampling distribution). Đây là điều **vô cùng quan trọng**, các bạn hãy lưu ý ghi nhớ.

Lưu ý rằng việc chúng ta lấy nhiều dữ liệu mẫu sau đó tính giá trị thống kê của các dữ liệu mẫu này, cái phân bố của nhiều giá trị được tính toán đó không phải là phân bố mẫu.

Ví dụ, những người đàn ông trưởng thành Việt Nam có chiều cao trung bình là 172 cm. Từ những người đàn ông trưởng thành Việt Nam chúng ta lấy dữ liệu mẫu. Với mỗi dữ liệu mẫu ta tính ra chiều cao trung bình. Ví dụ ta có 5 dữ liệu mẫu và tương ứng có 5 giá trị trung bình của 5 dữ liệu mẫu này.

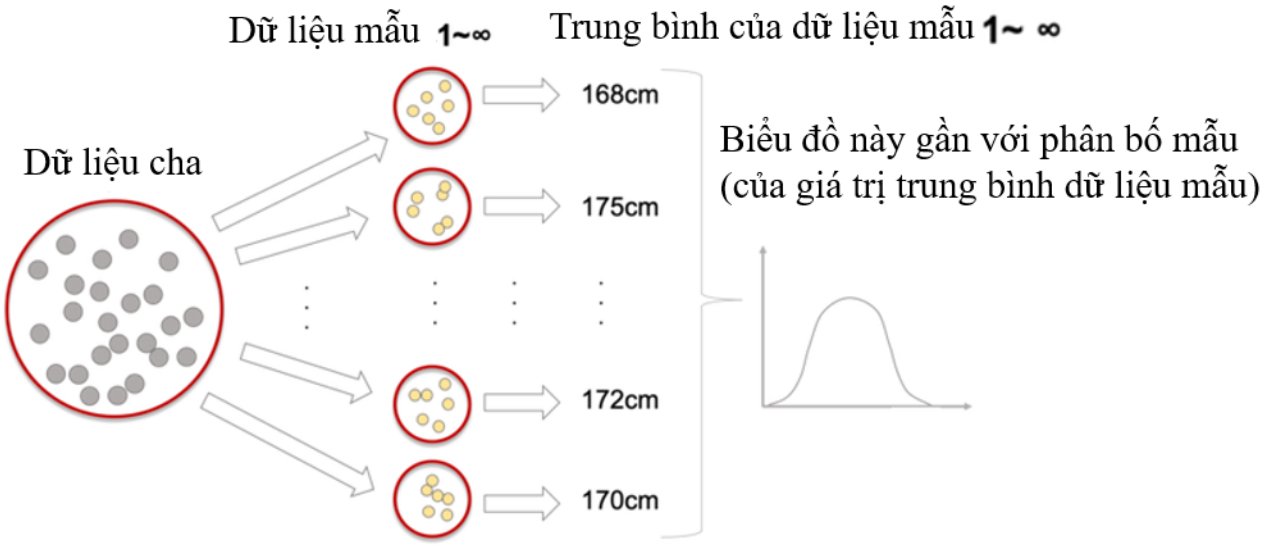
Giá trị trung bình giả sử là 168cm, 170cm, 171cm, 172cm, 175cm.



Những giá trị này hông thể gọi là phân bố mẫu.

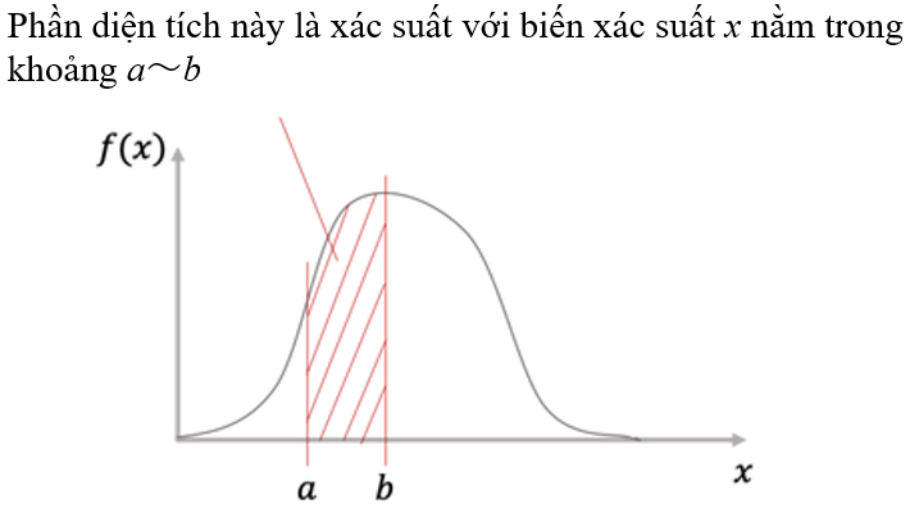
Phân bố mẫu nghĩa là phân bố xác suất của giá trị thống kê từ dữ liệu mẫu (trong bài học này là giá trị trung bình của dữ liệu mẫu) tuân theo và không thể nhìn thấy được.

Giả sử chúng ta lấy dữ liệu mẫu vô hạn lần, khi đó giá trị thống kê dữ liệu mẫu của vô hạn dữ liệu mẫu này (đã chuẩn hóa nghiêm ngặt) sẽ có biểu đồ càng gần với phân bố mẫu.

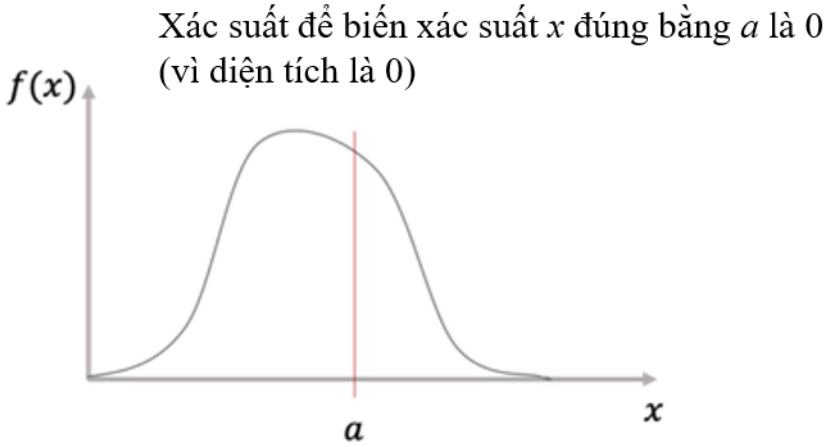


18.3 Tính xác suất từ hàm mật độ xác suất

Trong phân bố xác suất của biến xác suất liên tục, xác suất để biến ngẫu nhiên nhận giá trị trong một khoảng nào đó là diện tích của đường cong được biểu diễn bằng phân bố xác suất.



Những ai còn nhớ môn Toán cấp 3 sẽ nhận thấy rằng giá trị nhận được khi lấy tích phân hàm mật độ xác suất $f(x)$ trong khoảng từ a đến b là diện tích được biểu diễn bằng đường chéo màu đỏ trong hình trên.
Ngược lại xác suất để biến số xác suất x nhận một giá trị cụ thể nào đó là 0.
Tại sao lại như vậy là bởi vì, ví dụ chúng ta lấy ngẫu nhiên một người đàn ông trưởng thành Việt Nam, xác suất để người này có chiều cao đúng bằng 170cm là 0.
Vì chiều cao có nghĩa là độ dài, nó có tính liên tục. Đúng là không có ai có chiều cao đúng bằng 170.000000cm phải không nào.



Thế nhưng trong khoảng từ 170 cm ~ 171 cm, diện tích của hàm mật độ xác suất không phải là 0 (xác suất không phải là 0%), nó sẽ có giá trị nào đó.

Bổ sung

Không chỉ đối với chiều cao, nói chung khi lưu dữ liệu, độ chính xác của dữ liệu có thể bị cắt, chẳng hạn nếu lấy chính xác có thể tính tới đơn vị mm, cho nên cũng có dữ liệu "vừa đúng 170 cm". Đây đơn thuần chỉ là chuyển đổi biến liên tục thành biến rời rạc để thuận tiện xử lý.

Diện tích của đường cong phân bố xác suất là xác suất cho nên hàm số đường cong này gọi là hàm mật độ.
Tương tự như khi ta tính số dân dựa trên mật độ dân số, người ta cũng tính xác suất từ mật độ xác suất.
Ngoài ra, thông thường toàn bộ diện tích cảu hàm mật độ xác suất là xác suất của biến xác suất nhận toàn bộ các giá trị cho nên nó là 1 tức xác suất là 100%.

18.4 Tổng kết

Trong bài học này, tôi đã giải thích về biến ngẫu nhiên (biến số xác suất), phân bố xác suất, phân bố lấy mẫu (phân bố mẫu) và mật độ xác suất.
Những thuật ngữ này tương đối giống nhau dễ gây nhầm lẫn, vì vậy hãy cẩn thận và chú ý các bạn nhé. Tuy nhiên chúng cũng là các thuật ngữ rất quan trọng trong xác suất thống kê, vì vậy hãy nắm vững thật chắc.

- Biến ngẫu nhiên (biến số xác suất) là biến số có xác suất tương ứng với từng giá trị có thể có và giá trị của chúng biến động ngẫu nhiên.
- Phân bố xác suất là phân bố của biến ngẫu nhiên, là cái mà biểu diễn xác suất tương ứng đối với từng giá trị của biến ngẫu nhiên.
- Giá trị thống kê mẫu là biến số xác suất có giá trị thay đổi ngẫu nhiên. Ví dụ thống kê chiều cao trung bình của một mẫu dữ liệu, thì chiều cao trung bình lấy được từ mẫu là biến xác suất, có giá trị thay đổi ngẫu nhiên tùy thuộc vào mẫu dữ liệu.

- Phân bố xác suất của các giá trị mẫu được gọi là phân bố mẫu. Lưu ý rằng việc lấy một số lượng mẫu rồi thực hiện tính toán, phân bố của giá trị thống kê này không phải là phân bố mẫu (Do lấy một hữu hạn dữ liệu mẫu).
- Hàm mật độ xác suất là hàm số biểu diễn phân bố xác suất của biến số xác suất liên tục.
- Diện tích của đường cong hàm mật độ xác suất trong một khoảng giá trị là xác suất của biến số xác suất trong khoảng giá trị đó.
- Các sách về thống kê ở Việt Nam gọi là phân phối xác suất, khái niệm này tương đương với phân bố xác suất trong bài học này.

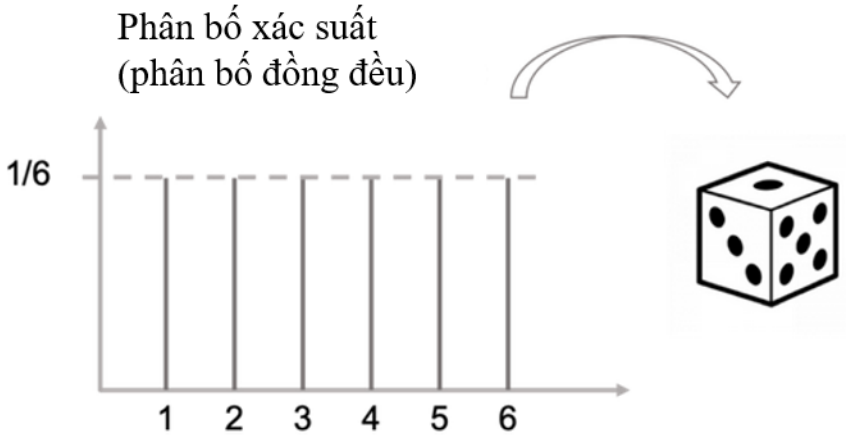
Bài 19

Sử dụng Python để lý giải phân bố nhị thức

Trong Bài học 18, tôi đã giới thiệu về phân bố xác suất, biến số xác suất.
Từ thời điểm này, tôi sẽ giới thiệu nhiều loại phần bố xác suất khác nhau.
Trong bài này, một kiến thức rất cơ bản về phấn bố xác suất đó là phân bố nhị thức(binominal distribution).

19.1 Hãy suy nghĩ phân bố xác suất như một thiết bị tạo dữ liệu

Trong Bài học 18 chúng ta đã nói về ví dụ tung xúc sắc. Số điểm trên mặt con xúc sắc có thể xuất hiện là từ 1 đến 6. Với lần lượt mỗi giá trị này đều có xác suất xuất hiện là $\frac{1}{6}$.
Phân bố xác suất mà ta nói trong ví dụ này là một loại phân bố nhưng được gọi chung là phân bố xác suất. Tuy nhiên hãy suy nghĩ rằng đây là phân bố đồng đều, phân bố này là phân bố mà mỗi giá trị từ $1 \sim 6$ đều có xác suất xuất hiện là $\frac{1}{6}$.
Tóm lại có thể nói rằng khi ta tung xúc sắc và mặt xúc sắc xuất hiện là giá trị được tạo ra phân bố xác suất đồng đều.
Cách nghĩ này dùng để lý giải phân bố xác suất và biến số xác suất (biến số ngẫu nhiên) là vô cùng dễ hiểu nên các bạn hãy lý giải điều này.



Và phân bố xác suất có một số kiểu cố định.
Nếu bạn biết loại phân bố xác suất nào (nghĩa là kiểu nào), bạn có thể biết giá trị nào được lấy với xác suất là bao nhiêu.
Nói như thế thì quá tiện lợi nhỉ! Nếu như bạn biết rằng xúc sắc tuân theo phân bố xác suất ở trên, bạn không cần phải có kinh nghiệm chơi trò chơi xúc sắc, bạn cũng có thể biết rằng nếu tung xúc sắc 1000 lần thì dường như các giá trị đều có xác suất là $\frac{1}{6}$.
Ví dụ trong một tổng đài điện thoại, cứ trung bình 10 phút thì có 3 cuộc điện thoại gọi tới. Vậy thì xác suất để trong 10 phút tới đây có 5 cuộc gọi tới là bao nhiêu? Hay trong một dây chuyền sản xuất, trung bình cứ 100 sản phẩm thì có 1 sản phẩm lỗi, vậy thì xác suất để không có sản phẩm lỗi nào khi cho 1000 sản phẩm vào hộp là bao nhiêu?
Một khi chúng ta biết phân bố xác suất mà một sự kiện tuân theo, về mặt lý thuyết, chúng ta có thể nói rằng **nó xảy ra với xác suất bao nhiêu**.
Quả là tiện lợi phải không? Các phân bố xác suất liên quan trực tiếp đến các vấn đề trong thực tế và nó rất hữu ích.
Trong bài học này, tôi sẽ giới thiệu về "phân bố nhị thức", có thể nói nó là cơ sở của các phân bố xác suất khác nhau.

19.2 Phân bố nhị thức là tổ tiên của phân bố xác suất

Có rất nhiều phân bố xác suất khác nhau nhưng hầu hết chúng đều bắt nguồn từ phân bố nhị thức.
Xác suất mà một sự kiện xảy ra trong một lần quan sát, giả sử có xác suất là p. Khi quan sát (hoặc thử nghiệm) này được thực hiện n lần, số lần (x) sự kiện này xảy ra tuân theo phân bố xác suất gọi là phân bố nhị thức. Trong đó x là một biến xác suất (biến ngẫu nhiên).

Vâng, tôi chẳng hiểu gì.

Bình tĩnh, tôi sẽ cho một ví dụ cụ thể.

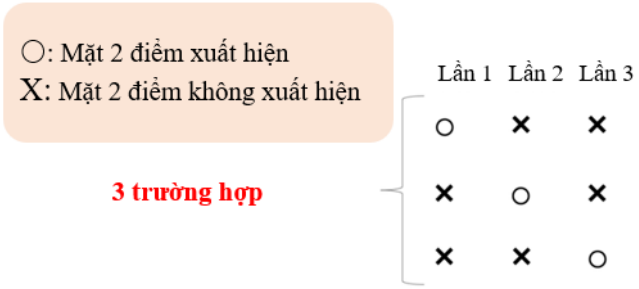
Ví dụ khi ta tung xúc sắc 3 lần, số lần mặt 2 điểm xuất hiện là x lần.

- 1 lần quan sát $\longrightarrow$ Tung xúc sắc 1 lần.
- Xác suất để sự kiện xảy ra $\longrightarrow$ Mặt 2 điểm xuất hiện có xác suất là $\frac{1}{6}$.
- Thực hiện n lần $\longrightarrow$ Thực hiện 3 lần
- x lần xảy ra sự kiện $\longrightarrow$ x lần mặt 2 điểm xuất hiện.

Đó, nó kiểu như vậy.
Khi tung xúc sắc 3 lần, số lần mặt 2 điểm xuất hiện đương nhiên sẽ nằm trong khả năng từ lần tung số 0 tới lần tung xúc sắc thứ 3.
Xác suất của mỗi lần được tính toán như trên.

Số lần xuất hiện: 0	Số lần xuất hiện: 1	Số lần xuất hiện: 2	Số lần xuất hiện: 3
$\left(\frac{5}{6}\right)^3 = \frac{125}{216}$	$3\left(\frac{1}{6}\right)\left(\frac{5}{6}\right)^2 = \frac{75}{216}$	$3\left(\frac{1}{6}\right)^2\left(\frac{5}{6}\right) = \frac{15}{216}$	$\left(\frac{1}{6}\right)^3 = \frac{1}{216}$

Tôi nghĩ số lần xuất hiện 0 lần và 3 lần không có vấn đề gì. Xác suất để mặt 2 điểm của xúc sắc không xuất hiện là $\frac{5}{6}$. Trong ba lần tung liên tục không xuất hiện mặt 2 điểm dù chỉ một lần là $\left(\frac{5}{6}\right)^3 = \frac{125}{216}$. Nếu mặt 2 điểm của xúc sắc xuất hiện 3 lần trong cả ba lần tung xúc sắc liên tục, cũng tương tự. Xác suất để mặt 2 điểm xuất hiện là $\frac{1}{6}$, xác suất để xuất hiện liên tục trong cả ba lần tung xúc sắc là $\left(\frac{1}{6}\right)^3 = \frac{1}{216}$.
Trường hợp mặt 2 điểm của xúc sắc xuất hiện một lần hoặc hai lần trong ba lần tung xúc sắc cách lập luận như nhau. Do đó tôi sẽ đi vào giải thích trường hợp mặt 2 điểm xuất hiện đúng một lần trong cả ba lần tung liên tục.
Để dễ hiểu, tôi minh họa bằng hình vẽ dưới đây:



Mặt 2 điểm xuất hiện đúng một lần cho nên có 3 trường hợp xảy ra, nó có thể xuất hiện ở lần tung thứ nhất, hoặc lần tung thứ hai, hoặc lần tung thứ ba. Ngoài ra xác suất để mặt 2 điểm xuất hiện là $\frac{1}{6}$. Nếu đã xuất hiện một lần thì hai lần còn lại phải không xuất hiện, mà xác suất mặt 2 điểm không xuất hiện là $\frac{5}{6}$, do đó xác suất hai lần không xuất hiện sẽ là $\left(\frac{5}{6}\right)^2$. Vậy xác suất để mặt 2 điểm xuất hiện đúng một lần là $3\left(\frac{1}{6}\right)\left(\frac{5}{6}\right)^2 = \frac{75}{216}$.
Tổng quát hóa, số trường hợp để mặt 2 điểm xuất hiện x lần trong n lần tung là:

$$C_n^x = \frac{n!}{x!\,(n-x)!}$$

Bổ sung

C là chữ cái đầu của từ **C**ombination. Nếu bạn muốn xem lại kiến thức về tổ hợp, bạn có thể google từ khóa bằng tiếng việt "tổ hợp, chỉnh hợp, hoán vị". Đây là kiến thức toán phổ thông nên tôi xin phép không nhắc lại.

Xác suất một sự kiện xảy ra là p, nếu thực hiện thực nghiệm n lần thì xác suất x lần sự kiện xảy ra là $P(x)$:

$$P(x) = C_n^x p^x q^{n-x} = \frac{n!}{x!\,(n-x)!} p^x q^{n-x}$$

Trong đó $q = 1 - p$, nó là xác suất mà sự kiện không xảy ra.
Phân bố như thế này gọi là **phân bố nhị thức (binominal distribution)**.
Biểu thức nói trên không thực sự quan trọng, vì nó không khó nên có thể nhớ được. Mà tôi nghĩ cũng không cần phải nhớ, phân bố nhị thức là phần cơ bản trong những phần cơ bản, nhìn mãi rồi thì cũng sẽ thấy quen mắt, tôi nghĩ vậy.
Điều quan trọng đó là bạn hiểu phân bố nhị thức là gì, hãy nhớ đó là một sự kiện có xác suất là p được quan sát trong n lần thì xác suất để sự kiện xảy ra x lần là gì thì phân bố nhị thức cho ta biết điều đó.
Đặc biệt hãy lưu ý là sự phân bố tập trung vào **"số lần"**.
Tôi nghĩ rằng có rất nhiều điều trong thực tế có thể áp dụng, tất nhiên việc tung xúc sắc và tung đồng xu là giống nhau. Ví dụ trò chơi bánh xe nhỏ trong sòng bạc có thể sử dụng phân bố nhị thức để lý giải số lần hiển thị.
Phân bố nhị thức là cơ sở để chúng ta lý giải các phân bố xác suất khác trong các bài học sau, vì vậy các bạn hãy ghi nhớ phân bố nhị thức.

Bổ sung

Trong phân bố nhị thức, khi $n = 1$, nó được gọi là phân bố Béc-no-li (Bernoulli distribution). Trường hợp $n = 1$ là một trường hợp đặc biệt và nó được gắn một cái tên khác.
Nhân tiện, một thử nghiệm chỉ đưa ra hai kết quả "thành công/ thất bại" được gọi là thử nghiệm Béc-no-li. Tự bản thân phân bố Béc-no-li ít khi được sử dụng cho bất cứ điều gì.

19.3 Từ phân bố xác suất tạo giá trị bằng Python

Có thể coi phân bố xác suất như một thiết bị tạo dữ liệu.
Bây giờ tôi sẽ sử dụng Python để tạo dữ liệu từ phân bố xác suất trong thực tế.
Trong công việc thực tế, code như thế này thật ra không nhiều. Thỉnh thoảng từ những phân bố đặc biệt, chúng ta cần tạo dữ liệu giả lập để tiến hành thực nghiệm, vì vậy các bạn hãy nhớ cách làm này. Ngoài ra việc lý giải bản thân phân bố xác suất là gì cũng rất

có ích, tôi nghĩ vậy.
Trong Module `stats` của `scipy` được trang bị rất nhiều loại phân bố xác suất, do đó tôi sẽ sử dụng nó.
Phân bố nhị thực được trang bị với cái tên là `binom`. Từ phân bố xác suất ta sẽ lấy giá trị, nói cách khác, ta sẽ tạo ra biến số xác suất tuân theo phân bố xác suất, chúng ta sẽ sử dụng hàm `.rvs()`
rvs là viết tắt tên của **r**andom **v**ariables (biến số xác suất).

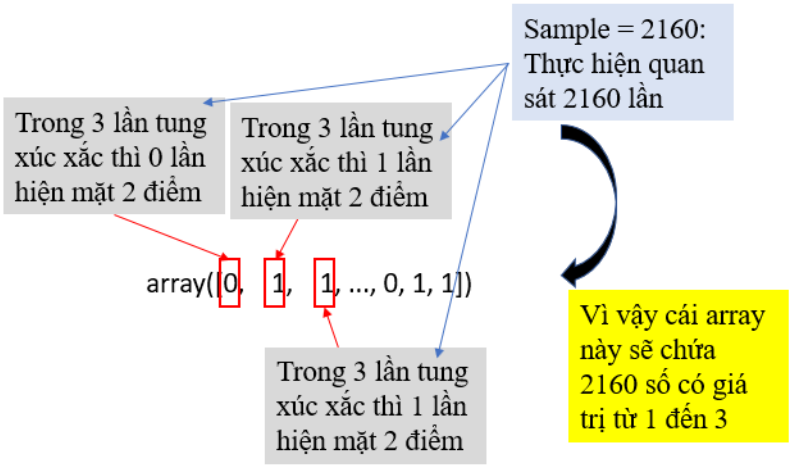
```
from scipy.stats import binom
data_binom = binom.rvs(n=3,p=1/6,size=2160)
```

Tôi xin nói lại về phân bố nhị thức, xác suất một sự kiện xảy ra trong một lần quan sát là p, xác suất để sự kiện đó xảy ra x lần trong n lần quan sát tuân theo một phân bố xác suất, phân bố xác suất này gọi là phân bố nhị thức.
$n = 3, p = 1/6$ có nghĩa là phân bố xác suất xảy ra x lần sau khi thử 3 lần biết rằng xác suất xảy ra sự kiện là 1/6. Nó giống như bạn tung xúc xắc 3 lần và số lần mà mặt xúc sắc hiện lên là 2 điểm như đã đề cập trước đây.
Vậy thì hãy thử quan sát xem nội dung của `data_binom` là gì?

```
data_binom
```

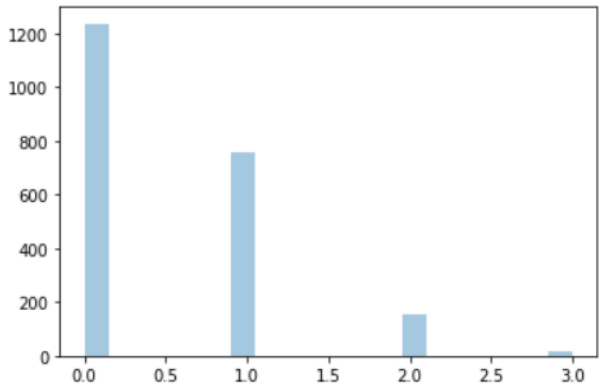
```
array([0, 1, 1, ..., 0, 1, 1])
```

Kết quả sẽ khác nhau tùy thuộc vào mỗi người khi chạy code trên, nhưng tôi nghĩ nó chứa 2160 số (từ 0 đến 3).
Ví dụ này cũng như ví dụ ta tung xúc sắc trước đây. `sample=2160` có nghĩa là ta thực hiện quan sát 2160 lần xem trong ba lần tung xúc sắc liên tục thì số lần mặt xúc sắc hiện 2 điểm là như nào.



Đến đây tôi nghĩ các bạn đã hiểu. Và chúng ta cũng có thể biết được trong `data_binom` thì các số $0, 1, 2, 3$ xuất hiện với tỷ lệ bao nhiêu.
Sau đây ta sẽ trực quan hóa dữ liệu bằng `seaborn`.

```
import seaborn as sns
%matplotlib inline

sns.distplot(data_binom, kde=False)
```



Hãy nhìn vào bảng xác suất ta đã tính trước đây và thấy rằng nó phù hợp với biểu đồ ở trên. Số lần xuất hiện 0 lần chiếm tỷ lệ cao nhất. Trong ba lần tung xúc sắc để mà xuất hiện cả ba lần mặt xúc sắc đều là 2 điểm có tỷ lệ rất thấp. Biểu đồ này cùng với bảng tính toán trước đó là phù hợp với nhau.
Nếu bạn lấy số lượng mẫu (sample) vô hạn lần, biểu đồ chúng ta thu được như thế này sẽ tiệm cận với phân phối xác suất trong thực tế.
Trong công việc thực tế, ta có thể thực hiện tính toán theo cách này để tạo ra dữ liệu và sử dụng nó để báo cáo cho ai đó, và nói "Nhìn xem! Hiếm khi mặt điểm 2 xuất hiện cả ba lần", như thế thật là tiện lợi.

Bổ sung

Tất nhiên tôi không nghĩ tất cả mọi người đều có hứng thú với xúc sắc. Nhưng có rất nhiều ví dụ thực tế mà phân phối nhị thức có thể được sử dụng. Trong thự tế không có nhiều sự kiện mà chúng ta biết trước xác suất nó xảy ra nhưng · Hãy ghi nhớ rằng, ở đây bạn có thể tạo ra dữ liệu từ phân phối xác suất bằng cách viết code Python giống như ở trên.

19.4 Tổng kết

Có rất nhiều kiểu phân bố xác suất, trong bài học này tôi đã giới thiệu về phân bố nhị thức.

- Có nhiều loại phân bố xác suất, có thể nói rằng phân bố nhị thức là tổ tiên của phân bố xác suất.
- Phân bố nhị thức là phân bố xác suất một sự kiện xảy ra x lần trong n lần quan sát biết rằng xác suất sự kiện đó xảy ra một lần là p.
- Phân bố xác suất được định nghĩa trong Module `scipy.stats`.

- Phân bố nhị thức sử dụng `scipy.stats.binorm`.
- Để tạo dữ liệu từ phân bố xác suất có thể dùng hàm `.rvs()`.
- Việc nhớ biểu thức tính xác suất nêu trong bài là không cần thiết nhưng hãy lý giải được phân bố nhị thức là gì.

Từ sau, tôi sẽ giới thiệu các phân bố xác suất khác nhau trong đó nền tảng vẫn là phân bố nhị thức. Các phân bố xác suất mà tôi sẽ giới thiệu trong các bài viết tiếp theo có thể được sử dụng trong các bài toán trong thế giới thực, vì vậy hãy chắc chắn biết đến chúng. Mong rằng chúng sẽ hữu ích cho công việc của bạn.

Bài 20

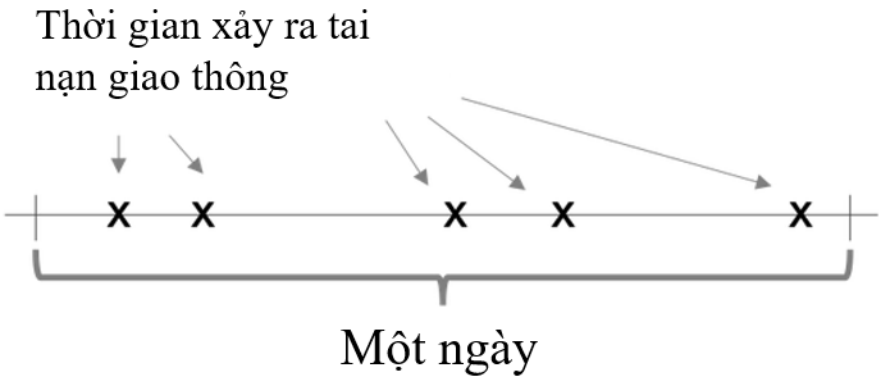
Giải thích về phân bố Poisson, mối quan hệ giữa phân bố nhị thức và phân bố chuẩn

Trong Bài học 19, tôi đã giới thiệu phân bố xác suất cơ bản nhất, đó là phân bố nhị thức. Trong bài viết này, tôi sẽ giới thiệu về phân bố Poisson **(Poisson Distribution)**, nó được phát triển từ phân bố nhị thức.
Phân bố Poisson là phân bố xác suất một sự kiện xảy ra x lần trong một khoảng thời gian, biết rằng trong khoảng thời gian này thì trung bình có μ lần sự kiện xảy ra.
Tôi nghĩ rằng nếu chỉ đọc định nghĩa như thế này thì thật sự khó hiểu, cho nên tôi sẽ đi vào giải thích dễ hiểu hơn ngay sau đây.

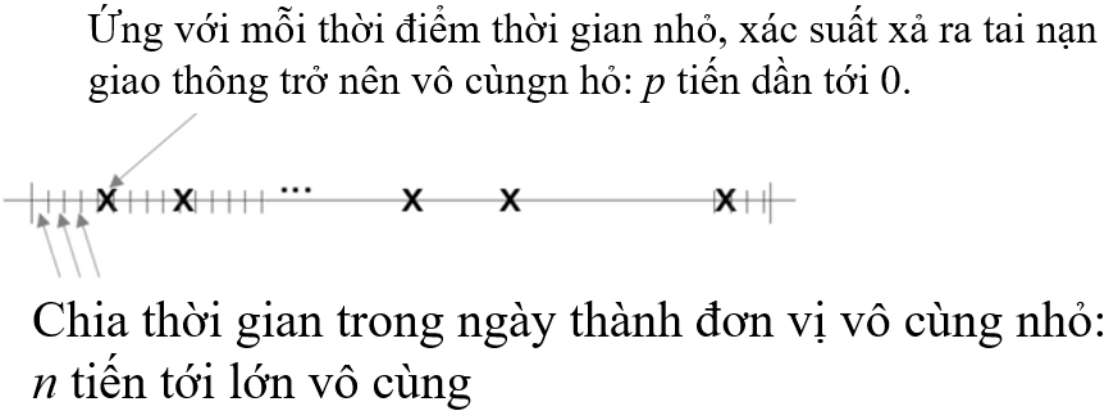
20.1 Phân bố Poisson là cực hạn của phân bố nhị thức

Phân bố Poisson là một trong những phân bố xác suất nổi tiếng nhất và là phân bố xác suất thường được áp dụng trong các bài toán thực tế.
Ví dụ, trong một ngày trung bình có 10 vụ tai nạn giao thông. Xác suất để hôm nay không có vụ tai nạn giao thông nào, hoặc, xác suất xảy ra 20 vụ tai nạn giao thông được tính như thế nào? Dựa vào phân bố Poisson chúng ta có thể tính được xác suất trong những bài toán như vậy.
Ví dụ khác, trung bình cứ 100 sản phẩm thì có 1 sản phẩm lỗi, vậy nếu ta cho 200 sản phẩm, xác suất để không có sản phẩm lỗi nào, hay xác suất có 5 sản phẩm lỗi là bao nhiêu?
Đây quả nhiên là những bài toán mang tính thực chiến rất cao, vì nó xảy ra trong cuộc sống thường ngày của chúng ta.
Nào, bây giờ chúng ta sẽ thử suy nghĩ dựa trên nền tảng là phân bố nhị thức.
Phân bố nhị thức là phân bố xác suất một sự kiện xảy ra x lần trong n lần thực nghiệm biết rằng xác suất để sự kiện này xảy ra một lần là p.
Nếu chúng ta cố định np trong phân bố nhị thức, coi nó là hằng số, trong đó cho n tiến

tới lớn vô cùng, và cho p tiến tới nhỏ vô cùng. Ở đây ta coi np là μ. (Thực ra không phải là u, thường thì ký hiệu λ thường được dùng nhiều hơn nhưng chúng ta sử dụng ký hiệu μ để phù hợp với thư viện `scipy`).
Thế nghĩa là thế nào? Chúng ta hãy thử suy nghĩ về ví dụ, trong một ngày, trung bình có 5 vụ tai nạn giao thông.
Như vậy ta có trục thời gian trong một ngày, trong đó có các thời điểm xảy ra tai nạn giao thông.



Nếu ta chi thời gian trong một ngày thành các khoảng thời gian vô cùng nhỏ, từng thời khắc ta chỉ ra ở thời điểm đó **"đã xảy ra tai nạn giao thông"** hoặc **"không xảy ra tai nạn giao thông"**, như vậy chúng ta có *hai sự kiện.*



Giải thích dễ hiểu hơn, chẳng hạn nếu lấy một giây làm đơn vị thời gian, và một ngày có $24 \times 60 \times 60 = 86400$ giây. Với mỗi giây, ta liệt kê tại thời điểm đó "đã xảy ra tai nạn giao thông" hoặc "không xảy ra tai nạn giao thông".
Bây giờ liên tưởng tới phân bố nhị thức, xác suất xảy ra tai nạn giao thông là p, chúng ta quan sát trong một ngày ($n = 84600$) xem thức thế có xảy ra $x = 5$ vụ tai nạn giao thông hay không?
Ở đây, xác suất p để một vụ tai nạn giao thông xảy ra trong khoảng thời gian một giây là vô cùng nhỏ đúng không nào.
Trong một ngày xảy ra 5 vụ tai nạn giao thông tức là trong 86400 giây chỉ xảy ra 5 vụ tai nạn giao thông. Xác suất để một vụ tai nạng thông xảy ra trong 1 giây chắc chắn là vô cùng nhỏ.

Ta đang giả định các vụ tai nạn giao thông không xảy ra đồng thời vào cùng một thời điểm. Nếu bạn băn khoăn về giả định này, vậy thì hãy chia nhỏ thời gian hơn nữa, không phải là một giây, có thể chia đơn vị thời gian là 0.001 giây. Hãy chia nhỏ thời gian hơn nữa. Khi đó xác suất quan sát được một vụ tai nạn giao thông trong một khoảng thời gian càng trở nên nhỏ hơn.

Có thể nói rằng đây là phân bố nhị thức với n tiến tới vô cùng lớn và p tiến tới 0 và $\mu = np$, để cố định μ thì n càng lớn thì p càng nhỏ.

Khi tiến hành làm như vậy người ta gọi đó là phân bố xác suất rời rạc **Poisson (Poisson Distribution)**.

Tôi sẽ giản lược bước chứng minh, mà xin được đưa ra công thức như dưới đây:

$$P\left(x\right) = \frac{\mu^x e^{-\mu}}{x!}$$

Với ai có hứng thú tìm hiểu chứng minh công thức trên thì hãy theo dõi biến đổi dưới đây, từ phân bố nhị thức ta có:

$$\begin{aligned}
P\left(x\right) &= C_n^x p^x q^{n-x} = \frac{n!}{x!\left(n-x\right)!} p^x q^{n-x} \\
P\left(x\right) &= C_n^x p^x q^{n-x} \left(\frac{\mu}{n}\right)^x \left(1-\frac{\mu}{n}\right)^{n-x} \\
P\left(x\right) &= \frac{n(n-1)(n-2)\cdots(n-x+1)}{x!} \frac{\mu^x}{n^x} \left(1-\frac{\mu}{n}\right)^n \left(1-\frac{\mu}{n}\right)^{-x}, x = 0, 1, 2, \ldots \\
&= \frac{\mu^x}{x!} \left(1-\frac{\mu}{n}\right)^n \left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)\cdots\left(1-\frac{x-1}{n}\right)\left(1-\frac{\mu}{n}\right)^{-x}
\end{aligned}$$

Do n vô cùng lớn nên ta có:

$$\lim_{n\to\infty} \left(1-\frac{\mu}{n}\right)^n = e^{-\mu}$$

$$\lim_{n\to\infty} \left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)\cdots\left(1-\frac{x-1}{n}\right)\left(1-\frac{\mu}{n}\right) = 1$$

Việc nhớ chứng minh trên là không cần thiết.

Chúng ta sử dụng phân bố Poisson để từ đó có thể tính được xác suất một sự kiện xảy ra x lần trong một khoảng thời gian, biết rằng trong khoảng thời gian ấy thì trung bình có μ lần sự kiện xảy ra.

Thông thường phân bố Poisson được sử dụng đối với thời gian nhưng tất nhiên nó cũng có thể sử dụng với yếu tố khác không phải là thời gian.

Chẳng hạn số lỗi chính tả trên một trang có thể được tính bằng cách sử dụng phân bố Poisson.

Chú ý

Thời gian thường được sử dụng trong phân bố Poisson vì nó có thể chia nhỏ vô hạn lần, khi đó xác suất p nhỏ vô hạn tiến dần tới 0. Về cơ bản phân bố Poisson có khuynh hướng sử dụng cho các sự kiện hiếm khi xảy ra (xác suất gần bằng 0).

20.2 Sử dụng Python tính phân bố Poisson

Giống với bài học trước, ta sẽ thử sử dụng Module `scipy.stats` để tính phân bố Poisson.
Sử dụng `poisson` để tính phân bố Poisson.
Lần trước ta đã sử dụng hàm `.rvs()` để tạo giá trị cho biến số xác suất từ phân bố xác suất. Lần này ta sẽ sử dụng `.pmf()` để vẽ phân bố Poisson.
pmf là tên viết tắt của từ **Probability Mass Function**(Hàm số chất lượng xác suất, các tài liệu thống kê ở Việt Nam gọi là Hàm Khối Xác Suất). Hàm chất lượng xác suất lấy xác suất của biến số xác suất rời rạc. Với biến số xác suất liên tục ta có hàm mật độ xác suất (Probability Density Function:PDF) đã nói trong Bài Học 18.
Đối số đầu vào của hàm `.pmf()` là `mu,k` trong đó k là số lần sự kiện xảy ra.
Ví dụ, cứ 10 phút trung bình có 5 cuộc gọi tới trung tâm tổng đài. Xác suất để 1 giờ có 40 cuộc gọi đến là $\mu = 5 \times \frac{60}{10} = 30, k = 40$.
Cách tính μ là ta lấy số sự kiện trung bình xảy ra trong thời gian mà chúng ta muốn tìm xác suất (trong ví dụ này là 1 giờ), với giả thiết cứ 10 phút trung bình có 5 cuộc gọi đến, do đó trong 1 giờ trung bình có $\mu = 30$ cuộc gọi đến.

k là gì?

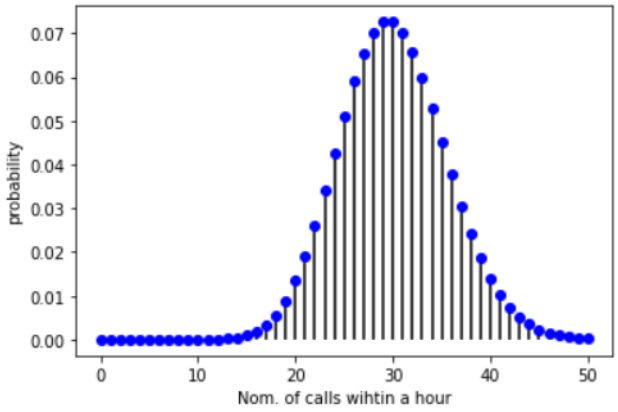
Trong phân bố xác suất rời rạc, x là giá trị rời rạc, biến số xác suất (biến số ngẫu nhiên) $x = k(k = 0, 1, 2, ..., k)$. Thông thường người ta viết $P(x = k)$, trong sách này bạn không cần bận tâm nếu tôi viết ngắn gọn là $P(x)$.

```
from scipy.stats import poisson
import math
import numpy as np
mu = 30
k = 40
p1 = poisson.pmf(k=k, mu=mu)
#Tinh theo cong thuc Poisson
p2 = (mu**k * np.e**(-mu))/math.factorial(k)
print(p1, p2)
```

```
0.013943463479967897 0.013943463479967761
```

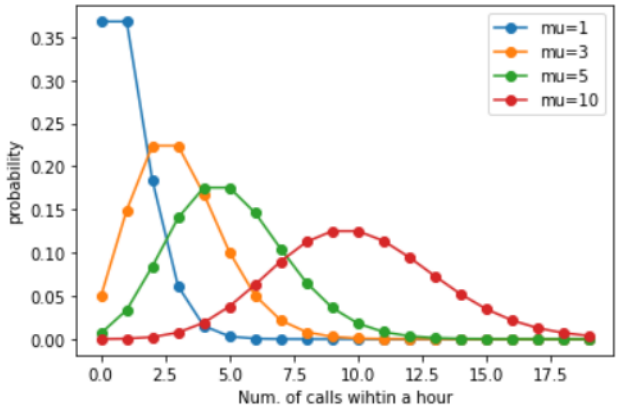
Kết quả của `scipy.stats.poisson` với cách tính thông thường có giá trị giống nhau. 0.0139... có nghĩa là xác suất chỉ ở khoảng 1.4%. Kết quả này là lớn hay bé? Chúng ta cho trục hoành với $k = 0, 1, 2, ..., 50$ và dùng `plot` để vẽ phân bố Poisson.

```
import matplotlib.pyplot as plt
%matplotlib inline

x = np.arange(51)
plt.plot(x, poisson.pmf(k=x, mu=mu), 'bo')
plt.vlines(x, 0, poisson.pmf(k=x, mu=mu))
plt.xlabel("Nom. of calls wihtin a hour")
plt.ylabel("probability")
```



Vì là phân bố xác suất rời rạc cho nên trong `plt.plot()` không kết nối các điểm thành đường liên tục. Trong code trên đã sử dụng `plt.vlines()`để vẽ từ trục `x` lên các điểm tạo thành các đưởng kẻ thẳng đứng.
Nhìn vào biểu đồ trên ta thấy xác suất cao nhất (đương nhiên) là 1 giờ có 30 cuộc gọi đến, nó có xác suất là 7%.
Phân bố Poisson đã được vẽ nhưng nó không giống với phân bố chuẩn? Chúng ta thử thay đổi `mu` và vẽ thử xem sao. Tôi nghĩ `mu` càng lớn thì đồ thị càng gần với phân bố chuẩn.

```
x = np.arange(20)
mu_list = [1, 3, 5, 10]
for idx, mu in enumerate(mu_list):
plt.plot(x, poisson.pmf(k=x, mu=mu), 'o-', label='mu={}'.format(mu))
plt.legend()
plt.xlabel("Num. of calls wihtin a hour")
plt.ylabel("probability")
```



Các điểm đường nối thành các đường liên tục nhưng lưu ý rằng thực tế đây là phân bố rời rạc.

20.3 Tổng kết

Trong bài học này tôi đã giới thiệu phân bố Poisson.

- Phân bố Poisson là phân bố xác suất thường được sử dụng trong các bài toán thực tế (trong cuộc sống thường ngày).

- Phân bố Poisson được định nghĩa như là phân bố nhị thực có giới hạn, trong đó $\mu = np$ là hằng số cố định, n lớn vô hạn và p nhỏ vô cùng gần bằng 0.
- Phân bố Poisson biểu diễn xác suất x lần một sự kiện xảy ra trong một khoảng thời gian, biết rằng trong khoảng thời gian này có trung bình μ lần sự kiện xảy ra.
- Có thể dùng `scipy.stats.poisson` để tính phân bố Poisson.
- Hàm chất lượng xác suất (PMF: Probability Mass Function) là hàm biểu diễn xác suất với biến xác suất là biến rời rạc. Hãy thử so sánh nó với hàm mật độ xác suất ((PDF: Probability Density Function) làm việc với biến xác suất liên tục.

Phân bố Poisson giống như phân bố nhị thức, nó là phân bố tập trung vào **số lần quan sát**.
Lần tới tôi sẽ giới thiệu phân bố xác suất tập trung vào "khoảng thời gian" tức là mất bao nhiêu thời gian để quan sát. Giống như phân bố Poisson, nó là một phân bố xác suất có thể được áp dụng trực tiếp cho nhiều bài toán trong thế giới thực, vì vậy tôi muốn các bạn sẽ ghi nhớ nó.

Bài 21

Phân bố hình học, phân bố mũ! Mối liên quan với phân bố Poisson là gì?

Trong Bài học 19 và Bài học 20, ta đã biết phân bố Poisson được phát triển từ phân bố nhị thức. Trong đó phân bố nhị thức là nền móng phát triển cho nhiều phân bố xác suất khác.
Phân bố Poisson là phân bố xác suất có tính thực chiến, có thể áp dụng trong các vấn đề trong thế giới thực, tức trong cuộc sống thường ngày. Trong bài học này, tôi tiếp tục giới thiệu tới các bạn về phân bố hình học (Geometric Distribution) và phân bố mũ (Exponential Distribution).
Nếu như phân bố nhị thức và phân bố Poisson tập trung vào "số lần" thì phân bố hình học và phân bố mũ tập trung vào "thời gian" và "khoảng thời gian". Cả hai đều giống như Poisson, tức là nó rất quan trọng, vì vậy các bạn hãy nắm chắc về chúng.
Nếu các bạn đã hiểu bài học lần trước, tôi nghĩ bài học lần này sẽ không khó.
Nói một cách đơn giản:

- Phân bố hình học được định nghĩa là phân bố xác suất rời rạc của một biến ngẫu nhiên, là xác suất số lần thử nghiệm x từ khi bắt đầu cho tới khi sự kiện xảy ra, biết rằng xác suất xảy ra sự kiện xảy ra là p.

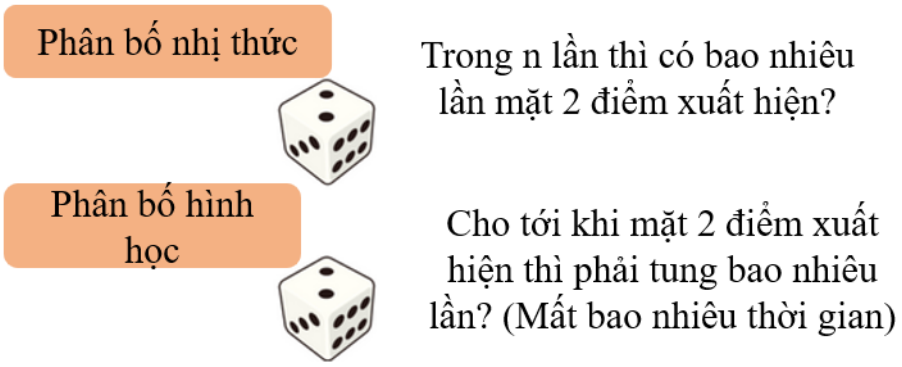
- Phân bố mũ là phiên bản của phân bố hình học nhưng là phân bố xác suất liên tục

Tôi nghĩ nếu chỉ đọc định nghĩa như thế này sẽ cảm thấy khó hiểu, do đó chúng ta sẽ đi vào chi tiết ngay bây giờ.

21.1 Phân bố nhị thức chú ý vào Số Lần, Phân bố hình học chú ý vào Thời Gian

Phân bố nhị thức quan tâm tới **số lần**, ví dụ nếu tung xúc sắc 3 lần thì số lần mặt xúc sắc 2 điểm xuất hiện có xác suất là như thế nào. Cụ thể, nếu mặt 2 điểm nếu chỉ xuất hiện một lần thì xác suất ra sao? Nếu xuất hiện đúng hai lần thì xác suất là bao nhiêu? Nếu xuất hiện cả ba lần thì xác suất là bao nhiêu?

Phân bố Poisson cũng quan tâm tới **số lần** xảy ra sự kiện trong một đơn vị thời gian nào đó. Có thể nói nó quan tâm tới **số lần thành công**.
Phân bố hình học chú ý vào **số lần thực nghiệm** cho tới khi mặt xúc sắc 2 điểm xuất hiện. Về mặt ngôn ngữ, chúng đều giống nhau vì có từ **"số lần"** nhưng ý nghĩa là khác nhau. Để dễ hình dung, có thể nghĩ phân bố hình học là **thời gian chờ** cho tới khi mặt xúc sắc 2 điểm xuất hiện.
Phân bố hình học (Geometric distribution) là phân bố xác suất của thời gian chờ từ lúc bắt đầu cho tới khi thành công (sự kiện xảy ra). Nó được gọi là phân bố rời rạc của thời gian chờ.



Khái niệm "thời gian" ở đây có nghĩa là *đã tiến hành thực nghiệm bao nhiêu lần*?
Hàm chất lượng xác suất của phân bố hình học có thể được tính đơn giản từ công thức phân bố nhị thức.
Công thức phân bố nhị thức đã nêu trong Bài học 19.

$$P\left(x\right)=C_n^x p^x q^{n-x}=\frac{n!}{x!\left(n-x\right)!}p^x q^{n-x}$$

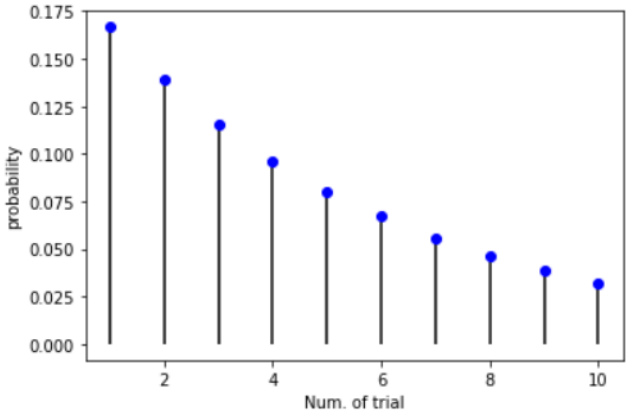
Trong đó q là xác suất sự kiện không xảy ra, $q=1-p$.
Với một sự kiện có xác suất xảy ra là p, ta hãy suy nghĩ xác suất mà từ lúc bắt đầu cho tới khi tiến hành thực nghiệm x lần thì sự kiện xảy ra.
Trong x lần thực nghiệm thì có $x-1$ lần thất bại và chỉ 1 lần thành công, cho nên:

$$P\left(x\right)=q^{x-1}p$$

Đây là công thức của hàm chất lượng xác suất biểu diễn phân bố hình học.
Nó cực kỳ đơn giản. Với sự kiện có xác suất p chỉ xảy ra ở lần cuối cùng và các lần trước đó đều thất bại, công thức trên là hoàn toàn dễ hiểu.
Công thức trên không mấy quan trọng vì nó không phải là công thức khó, **nếu hiểu phân bố hình học là gì, bạn có thể suy ra nó một cách tự nhiên**.
Nào, bây giờ ta sẽ thử dùng Python để vẽ phân bố xác suất hình học.
Trong Bài học 20 tôi đã giới thiệu `.pmf()` để vẽ.
Lần này ta sẽ cho $p=\frac{1}{6}$ và vẽ xác suất khi $x=1,2,...,10$.
Cái này cũng giống như ví dụ tung xúc sắc trước đây.

```
from scipy.stats import geom
```

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

x = np.arange(1, 11)
y = geom.pmf(k=x, p=1/6)
plt.plot(x, y, 'bo')
plt.vlines(x, 0, y)
plt.xlabel("Num. of trial")
plt.ylabel("probability")
```

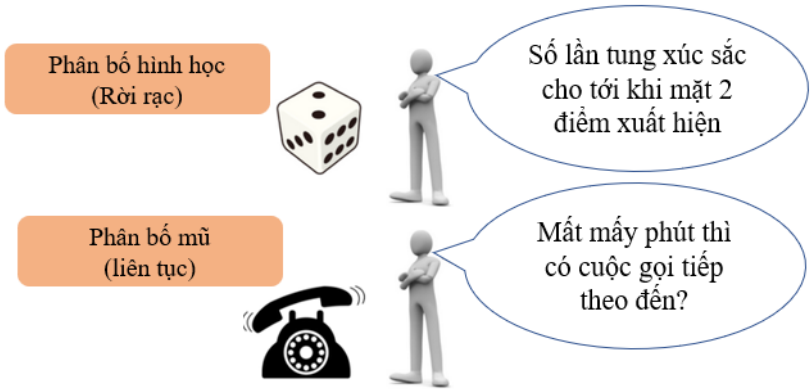


Nhìn vào kết quả tôi nghĩ x càng lớn thì xác suất càng giảm là điều chúng ta có thể hiểu được.
Suy nghĩ này rất tự nhiên, ví dụ khi suy nghĩ xác suất để mặt xúc sắc 2 điểm sau x lần tung xúc sắc, như vậy số lần tung xúc sắc càng nhiều thì phải đảm bảo rằng các lần tung trước đó không được xuất hiện mặt xúc sắc 2 điểm, cái xác suất cho điều ấy càng ngày càng nhỏ nếu số lần tung xúc sắc càng lớn.
Trong ví dụ trên, xác suất khi $x = 10$ tức là tìm xác suất mặt xúc sắc 2 điểm xuất hiện sau 10 lần tung xúc sắc. Tức là 9 lần liên tục trước đó mặt xúc sắc 2 điểm không được xuất hiện. Một ví dụ cực đoan hơn, trường hợp $x = 100$ thì tức là tìm xác suất để mặt xúc sắc 2 điểm xuất hiện sau 100 lần tung xúc sắc. Có lẽ nó sẽ có giá trị gần với 0.
Vì xác suất có tính suy giảm chuỗi hình học, nên nó được gọi là phân bố hình học.

21.2 Phân bố mũ là phiên bản liên tục của phân bố hình học

Phân bố hình học là phân bố xác suất rời rạc. Bởi vì biến số xác suất lấy giá trị rời rạc, không có tính liên tục, ví dụ là 1 hoặc 2 hoặc 3.
Trong ví dụ về tung xúc sắc trước đây đã nói số lần tung xúc sắc là giá trị rời rạc, không có tính liên tục.
Phân bố mũ (exponential distribution) là phiên bản liên tục của phân bố hình học.
Vậy thì phiên bản liên tục của phân bố hình học nghĩa là thế nào?
Phân bố hình học biến xác suất đếm số lần thực nghiệm (1 lần, 2 lần,...) cho tới khi thành công.

Phân bố mũ là biến xác suất **thời gian** cho tới khi thành công.
Ví dụ trong 1 giờ trung bình có 10 cuộc gọi đến tổng đài. Vậy thì xác suất thời gian cho cuộc gọi tiếp theo gọi đến là bao nhiêu?

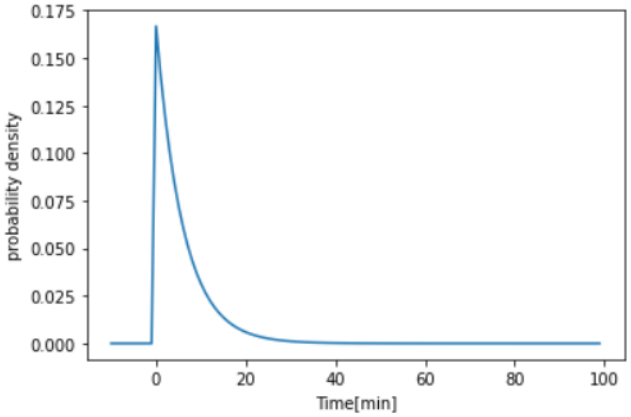


Hàm mật độ xác suất của phân bố mũ có công thức như dưới đây:

$$P(x) = \begin{cases} \lambda e^{-\lambda x} & (x \geqq 0) \\ 0 & (x < 0) \end{cases}$$

Trong đó λ là số lần trung bình mà sự kiện xảy ra trong thời gian kỳ vọng (có đơn vị là thời gian). Như vậy biến số xác suất liên tục x là thời gian cho tới khi sự kiện xảy ra.
Đơn vị thời gian có thể là 1 phút hay 1 giờ. Ví dụ trung bình cứ 1 giờ thì có 10 cuộc điện thoại gọi đến tổng đài. Nếu nhìn vào xác suất để cuộc gọi tiếp theo gọi đến trong bao nhiêu phút, ta có với 1 phút sẽ có $\lambda = \frac{10}{60} = \frac{1}{6}$.
Phân bố mũ cũng giống như phân bố hình học, ta có thể sử dụng `scipy.stats` để vẽ.
Phân bố mũ là **`expon`**. Trở lại với ví dụ cuộc gọi đến tổng đài, ta sẽ vẽ như sau:

```
from scipy.stats import expon
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

x = np.arange(-10, 100)
y = expon.pdf(x=x, scale=6)
plt.plot(x, y)
plt.xlabel("Time[min]")
plt.ylabel("probability density")
```



Khác với `geom` của phân bố hình học, `expon` của phân bố mũ là phân bố xác suất liên tục cho nên không phải là `.pmf()` mà sử dụng `.pdf()`, xin hãy lưu ý. (PMF là Probability Mass Function, PDF là Probability Desnity Function, chúng lần lượt là hàm chất lượng xác suất và hàm mật độ xác suất).
Tham số λ sử dụng cho `scale` của `.pmf()`. Cụ thể ta sẽ đưa giá trị $\frac{1}{\lambda}$ cho đối số `scale`. Trong ví dụ lần này vì $\lambda = \frac{1}{6}$ nên ta sẽ cho `scale` là 6.
Ví dụ lần này đơn vị thời gian là phút nên trục hoành thời gian sẽ là phút, cứ 1 giờ trung bình có 10 cuộc gọi đến, ta cần vẽ xác suất sau x phút thì có cuộc gọi tiếp theo gọi tới tổng đài.
x nhỏ hơn 0 thì đương nhiên xác suất sẽ là 0.
Thời gian cho tới cuộc gọi tiếp theo gọi đến, rõ ràng thời gian này càng dài thì xác suất càng nhỏ. Trung bình cứ 1 giờ có 10 cuộc gọi đến, sau 30 phút mà không có cuộc gọi nào thì xác suất cho sự kiện như vậy xảy ra rõ ràng là thấp.

21.3 Tổng kết

Trong bài học này tôi đã giới thiệu về phân bố hình học và phân bố mũ. Phân bố nào cũng nổi tiếng ngang với phân bố Poisson, có thể ứng dụng trong rất nhiều vấn đề trong thực tế. Hãy nắm vững nó.

- Phân bố nhị thức, phân bố Poisson tập trung vào số lần thành công. Phân bố hình học, phân bố mũ tập trung vào thời gian chờ đợi cho tới khi thành công (Sự kiện xảy ra).
- Phân bố hình học là phân bố xác suất rời rạc tuân theo số lần x thực nghiệm kể từ khi quan sát thì sự kiện xảy ra, biết rằng xác suất để sự kiện xảy ra là p.
- Phân bố mũ là phiên bản liên tục của phân bố hình học.
- Phân bố mũ là xác suất sau một khoảng thời gian x thì sự kiện xảy ra, biết rằng trung bình có λ lần sự kiện xảy ra trong một khoảng thời gian nào đó (đơn vị thời gian).
- Phân bố hình học, phân bố mũ bằng Python lần lượt là `scipy.stats.geom` và `scipy.stats.epon`

Bài 22

Khám phá bản chất của phân bố chuẩn

Trong Bài học 19, tôi đã giới thiệu về phân bố nhị thức. Khi $n = 1$ ta có phân bố Bernoulli. Trong Bài học 20, phân bố Poisson được định nghĩa là phân bố nhị thức trong giới hạn vô cùng, phân bố hình học và phân bố mũ thì không phải là số lần thành công mà là phân bố thời gian chờ cho tới khi thành công đã giới thiệu ở Bài học 21.
Trong bài học này tôi sẽ giới thiệu về phân bố chuẩn (Normal Distribution), hay còn được gọi với cái tên là phân bố chính quy.

22.1 Bản chất của phân bố chuẩn

Phân bố chuẩn được biểu diễn bằng giới hạn vô cùng của phân bố nhị thức. Ồ, phân bố xác suất nào cũng bắt nguồn từ phân bố nhị thức nhỉ. Như đã nói phân bố nhị thức là cụ tổ của phân bố xác suất.

22.2 Xem lại phân bố nhị thức

Nếu bạn đang thắc mắc phân bố nhị thức là phân bố như thế nào, bạn có thể xem lại Bài học 19.
Một sự kiện với xác suất p. Gọi $P(x)$ là xác suất khi tiến hành thực nghiệm (quan sát) n lần thì có x lần xảy ra sự kiện. Cái phân bố của xác suất này gọi là **phân bố nhị thức**.

$$P(x) = C_n^x p^x q^{n-x} = \frac{n!}{x!\,(n-x)!} p^x q^{n-x}$$

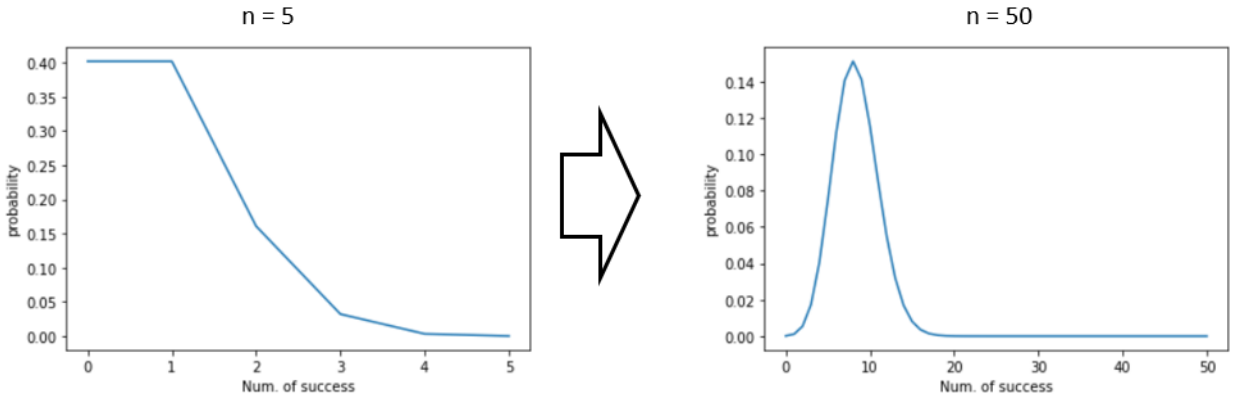
Nếu chúng ta cố định $\mu = np$ và cho n tiến tới lớn vô cùng, p tiến tới gần 0, phân bố xác suất khi ấy được gọi là phân bố Poisson. Các bạn có thể xem lại Bài học 21.
Lúc này nếu như p không giảm, ta có phân bố chuẩn (gần với phân bố chuẩn).

22.3 Hãy xem phân bố nhị thức gần giống với phân bố chuẩn bằng Python

Tôi sẽ sử dụng code Python đã được giới thiệu trong Bài học 19 và Bài học 20, ta thử vẽ biểu đồ của phân bố nhị thức.

Lấy ví dụ về việc tung xúc sắc. Xác suất $p = \frac{1}{6}$ và thử thay đổi gí trị n.

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

from scipy.stats import binom
n = 50 #->50
p = 1/6
x = np.arange(n + 1)
y = binom.pmf(k=x, p=p, n=n)
plt.plot(x, y)
plt.xlabel("Num. of success")
plt.ylabel("probability")
```



Ví dụ ta so sánh khi $n = 5$ với khi $n = 50$, ta thấy đồ thị càng lúc càng gần với phân bố chuẩn.(Để cho dễ hiểu ta đã nối các điểm lại với nhau nhưng xin lưu ý rằng trong thực tế phân bố nhị thức là phân bố rời rạc.)

Tôi sẽ giản lược bước chứng minh, mà đưa ra ngay công thức hàm mật độ xác suất của phân bố chuẩn $f(x)$:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Không cần thiết phải nhớ công thức này bởi vì nó là một trong những công thức nổi tiếng, bạn sẽ còn gặp nó rất nhiều khi học thống kê, và nhìn mãi thì cũng sẽ quen.

Điểm quan trọng của công thức này đó là giá trị trung bình μ và độ phân tán σ^2 sẽ quyết định hoàn toàn hình dạng của phân bố chuẩn.

Đến đây ta cũng hiểu được giá trị trung bình μ và độ phân tán σ^2 trong phân bố nhị thức lần lượt là np và npq.

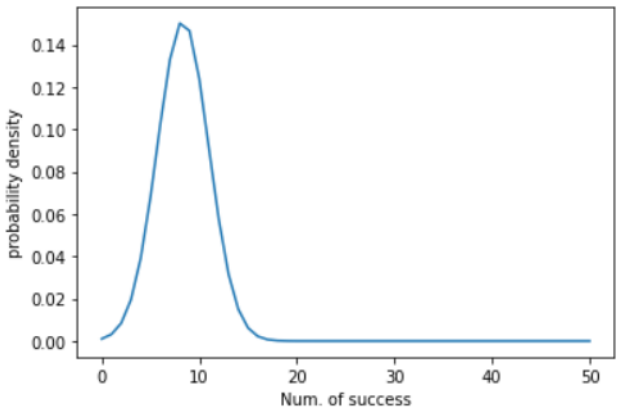
Tôi nghĩ các bạn hiểu tại sao giá trị trung bình μ bằng với np. Ví dụ khi ta tung xúc sắc 100 lần, ta có thể trả lời khả năng cao nhất số lần mặt điểm 2 xuất hiện đơn giản sẽ là

$100 \times \frac{1}{6}$. (Trong phân bố chuẩn, giá trị trung bình có đỉnh cao nhất (mật độ xác suất lớn nhất).)
Có lẽ sẽ khó hình dung về độ phân tán σ^2 nhưng có thể thấy khi n lớn thì đương nhiên mức độ rải rác của x cũng sẽ lớn theo.
Về giá trị phân tán này, việc nhớ cũng là không cần thiết nhưng các bạn hãy biết rằng, việc sử dụng các tham số của phân bố nhị thức để từ đó có thể vẽ được phân bố chuẩn.

22.4 Vẽ đường cong phân bố chuẩn bằng Python

Chúng ta có thể sử dụng `NumPy` để tạo ra giá trị từ phân bố chuẩn, điều này được nói chi tiết trong nhập môn Python hướng tới Data Science.
Trong bài học này ta sẽ sử dụng `norm.pdf` của `scipy.stats` để vẽ phân bố chuẩn. Về `.pdf()` tôi đã từng sử dụng trong Bài học 21.

```
from scipy.stats import norm
n = 50
mu = n * p
std = np.sqrt(n * p * (1-p))
x = np.arange(n + 1)
y = norm.pdf(x=x, loc=mu, scale=std)
plt.plot(x, y)
plt.xlabel("Num. of success")
plt.ylabel("probability density")
```



Ta vẽ biểu đồ với $n = 50$, đại để thì cũng giống với biểu đồ của phân bố nhị thức.
Ta truyền cho đối số của `.pdf()` là `loc` và `scale`. `loc` là `location`, trong trường hợp phân bố chuẩn thì đó chính là giá trị trung bình μ. `scale` có ý nghĩa như là bề rộng của biểu đồ, trong trường hợp của phân bố chuẩn thì đó là σ, xem lại độ lệch chuẩn ở đây.
p như đã nói, ta giữ nguyên giá trị của phân bố nhị thức và sử dụng.
Nhân tiện, phân bố xác suất với giá trị trung bình $\mu = 0$ và độ phân tán (phương sai) $\sigma = 1$ được gọi là phân bối chuẩn tắc (Standard normal distribution).
Bạn có thể xem lại chuẩn hóa.
Phân bố chuẩn, phân bố chuẩn tắc là phân bố xác suất không thể thiếu trong việc hình thành cơ sở lý luận của xác suất thống kê. Ngay cả cho tới thời điểm này cũng vậy. Sau này các bạn cũng sẽ thấy trong thống kê suy luận, rất nhiều trường hợp chúng ta giả định

phân bố của dữ liệu cha là phân bố chuẩn.
Phân bố xác suất sau này cũng sử dụng nó trong lý luận, vì chúng ta sử dụng những thứ được sinh ra từ phân bố chuẩn cho nên có thể nói phân bố chuẩn là phân bố xác suất vô cùng quan trọng.

22.5 Tổng kết

- Trong phân bố nhị thức, khi cho n tiến tới lớn vô cùng, nó gần đúng với phân bố Poisson hoặc phân bố chuẩn.
- Phân bố Poisson là phân bố nhị thức trong đó ta cố định np và cho p nhỏ về gần với 0.
- Nếu p không giảm, nó gần với phân bố chuẩn.
- Giá trị trung bình trong phân bố chuẩn là $\mu = np$ và phân tán (phương sai) là $\sigma^2 = npq$.
- Chúng ta sử dụng hàm `.pdf()` của `scipy.stats.norm` để vẽ phân bố chuẩn. Đưa giá trị trung bình vào `loc` và đưa độ lệch chuẩn vào `scale`.

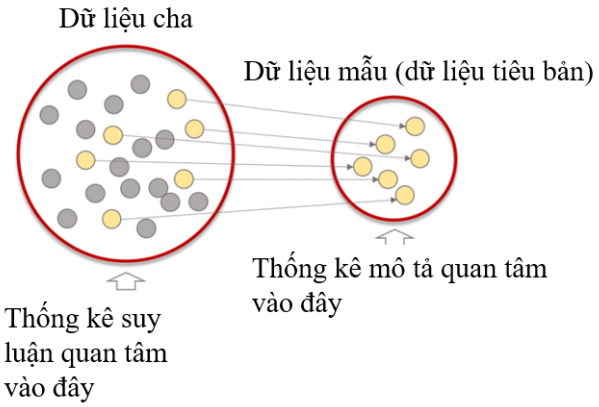
Ta có thể tạo ra các giá trị ngẫu nhiên để từ đó vẽ nên biểu đồ của phân bố chuẩn. Nhưng tôi nghĩ, việc lấy xấp xỉ phân bố chuẩn từ phân bố nhị thức thì nó sẽ mang lại cho các bạn cảm giác quen thuộc ngay lập tức.
Có thể coi rằng phân bố xác suất của số lần mặt 2 điểm xuất hiện khi tung xúc sắc rất nhiều lần là hình ảnh của một phân bố chuẩn.
Từ bài sau chúng ta sẽ đi sâu hơn vào thống kê suy luận. Nội dung sẽ trở nên thiết thực hơn, vì vậy hãy cố gắng học tập nhé!

Bài 23

Thống kê suy luận là gì?

23.1 Thống kê suy luận là gì?

Như đã đề cập trong Bài học 1, thống kê suy luận quan tâm tới dữ liệu cha. Từ dữ liệu cha trích xuất ra các dữ liệu mẫu (dữ liệu tiêu bản), và tìm ra các đặc điểm của chúng rồi suy ra đặc điểm của dữ liệu cha.



Nếu như có thể sử dụng thông tin có giới hạn là cái mà được gọi là dữ liệu mẫu rồi từ đó suy luận ra đặc tính của dữ liệu cha, cái điều như thế mà trở thành hiện thực, nó hẳn là rất có ích phải không nào?
Ví dụ, để tìm chiều cao trung bình của nam giới trưởng thành trên toàn Việt Nam, chúng ta không thể đo chiều cao của tất cả nam giới trưởng thành.
Để tìm hiểu sức bền của những chiếc điện thoại được sản xuất, ta không thể cho thả rơi tất cả những chiếc điện thoại tại nhà máy đó.
Khi đo đạc tìm ra tỷ lệ lỗi của sản phẩm, mối quan tâm của chúng ta không phải là những sản phẩm đã được tạo ra trong quá khứ mà là những sản phẩm sẽ được tạo ra trong tương lai.
Dù sao đi nữa, "dữ liệu cha" chỉ đơn thuần là một khái niệm, trong hầu hết các trưởng hợp, chúng ta không thể điều tra toàn bộ dữ liệu cha.
Thống kê học chính là có ích ở những nơi như thế. Khi học về thống kê suy luận, từ thông tin giới hạn của dữ liệu mẫu, chúng ta có thể truy tìm ra đặc tính của dữ liệu cha.

Lấy mẫu ngẫu nhiên(random sampling)

Tiền đề (điều tiên quyết) của lý luận thống kê suy luận đó là giả định rằng một mẫu dữ liệu là dữ liệu được trích xuất ngẫu nhiên từ dữ liệu cha (lấy mẫu ngẫu nhiên). Thế nhưng trong thực tế việc lấy mẫu ngẫu nhiên là một việc không đơn giản.
Ví dụ, để ước tính nhận thức của học sinh tiểu học trên toàn quốc thông qua kết quả bài kiểm tra nhưng việc chọn ngẫu nhiên học sinh từ các trường tiểu học lại không thể thực hiện được.
Vì vậy, hãy tạo ra một mẫu trong phạm vi mà bạn có thể chọn ngẫu nhiên một số trường tiểu học từ một thành phố nhất định.
Trong trường hợp như thế, việc thiết định đâu là dữ liệu cha trong một phạm vi mà chúng ta có thể thực hiện được, đó là việc quan trọng. Ví dụ, không phải là toàn quốc, chúng ta coi dữ liệu cha là học sinh tiểu học tại một thành phố cụ thể nào đó.
Không thể đoán được đặc điểm của học sinh tiểu học trên toàn quốc từ một vài trường tiểu học cụ thể.

23.2 Ước lượng và Kiểm định

Thống kê suy luận rất rộng nhưng có thể chia làm hai kiểu: Ước lượng (estimation) và kiểm định (statistical test).
Ước lượng có nghĩa là từ giá trị được tính toán từ dữ liệu mẫu, ta ước lượng đặc tính của dữ liệu cha. Ví dụ, để ước lượng chiều cao trung bình của nam giới trưởng thành trên toàn quốc, ta tính giá trị trung bình của dữ liệu mẫu và suy ra chiều cao trung bình của nam giới trưởng thành trên toàn quốc.
Như đã nói trong Bài học 6, để suy đoán giá trị đặc tính dữ liệu cha, ta sử dụng giá trị của dữ liệu mẫu, đó được gọi là **đại lượng thống kê**. Ta gọi đại lượng thống kê được sử dụng trong việc ước lượng là **công thức ước lượng (estimator)** để chỉ ra đặc tính của dữ liệu cha.
Đặc tính của dữ liệu cha gọi là **đối số (tham số) (parameter)**. "Đối số" là thuật ngữ được sử dụng nhiều, nên các bạn hãy ghi nhớ.

Đối số của dữ liệu cha

Nếu có một hàm số $f(x, y, z)$ thì người ta gọi x, y, z là các đối số hay tham số của hàm số f. Nếu biết các giá trị x, y, z thì có thể xác định được giá trị của hàm số. Một hình dung không hoàn toàn chính xác, nhưng có thể coi dữ liệu cha là một hàm số, các giá trị đại diện giống như là tham số hay đối số của dữ liệu cha. Nếu biết thông tin các giá trị đại diện này, ta có thể hình dung ra ít nhiều về dữ liệu cha. Tùy vào từng trường hợp mà đối số dữ liệu cha mà ta cần quan tâm là gì, đó có thể là tỷ suất, hay giá trị trung bình .v.v. Dù thế nào đi nữa thì ta cũng phải trích xuất các mẫu dữ liệu từ dữ liệu cha, trong việc trích xuất này ta cố gắng trích xuất sao cho nó có tính ngẫu nhiên, dù việc này là khó, ít nhất cũng phải trông cho có vẻ ngẫu nhiên. Nếu

chúng ta quan tâm tới tỷ suất của của dữ liệu cha, ta gọi đó là đối số của dữ liệu cha. Để tìm tỷ suất của dữ liệu cha, ta thông qua mẫu dữ liệu nhỏ hơn đã được trích xuất, và tìm tỷ suất của mẫu dữ liệu. Như vậy tỷ suất của mẫu dữ liệu sẽ gọi là đại lượng thống kê. Khi đã có tỷ suất của các mẫu dữ liệu khác nhau, nếu sử dụng nó để đưa ra phán đoán ước lượng rằng tỷ suất của dữ liệu cha là như nào, khi đó tỷ suất của dữ liệu mẫu được sử dụng để đưa ra phán đoán như thế sẽ gọi là công thức ước lượng.

Ý nghĩa của từ "kiểm định" khác so với những gì chúng ta hình dung về từ ngữ này. Từ bài viết này chúng ta sẽ nói riêng về ước lượng và chúng ta sẽ giải quyết phần thử nghiệm ở phần sau của khóa học.
Về một giả thiết nào đó đối với tính chất của dữ liệu cha, bằng kết quả thực nghiệm (quan sát) ở dữ liệu mẫu, ta có thể nói giả thiết này đúng hay không, có mâu thuẫn hay không, kiểm định chính là công việc điều tra những cái đó.
Nghe thì có vẻ khó nhưng ví dụ, giả thiết là "chiều cao trung bình của nam giới trưởng thành trên toàn quốc là 170 cm", đối với giả thiết như vậy, ta thử quan sát một mẫu dữ liệu nào đó, điều tra một cách ngẫu nhiên (xác suất), xem chiều cao trung bình có đúng là 170 cm hay không.
Giả thiết hay giả định này còn gọi là **giả thuyết, giả sử (hypothesis)**.
Sử dụng kiểm định trong việc kiểm định đại lượng thống kê ta gọi nó một cách đơn giản là đại lượng kiểm định.

23.3 Ước lượng điểm và ước lượng khoảng

Ước lượng có hai cách, đó là **ước lượng điểm (point estimation)** và **ước lượng khoảng (interval estimation)**. Ước lượng điểm là một phương pháp ước lượng tham số tổng thể bằng một công cụ ước lượng từ mẫu dữ liệu. Ước lượng khoảng là một phương pháp ước lượng lỏng hơn, rằng "nó sẽ nằm trong khoảng này".
Sẽ rất hay nếu ước lượng điểm có thể sử dụng để ước lượng nhanh, nhưng trong thực tế, ước lượng điểm được cho là không chắc chắn, vì vậy ước lượng khoảng được sử dụng để ước lượng.
Nói tóm lại, **ước lượng điểm thường không sử dụng trong thực tế, do đó người ta thường sử dụng ước lượng khoảng được coi như là phương pháp ước lượng điểm mở rộng**.

23.4 Cách ước lượng khoảng

Như đã đề cập ở trên, đối số (tham số) của dữ liệu cha có thể được tính toán bằng ước lượng. Nói tóm lại, giá trị trung bình hay tỷ suất, phân tán (phương sai) của dữ liệu cha v.v. có thể tính toán được.

Ta sẽ sử công thức ước lượng của dữ liệu mẫu để đưa ra đối số (tham số) của dữ liệu cha.

- Lấy mẫu ngẫu nhiên từ dữ liệu cha. (Trong thực tế, thông thường dữ liệu mẫu đã được chuẩn bị sẵn. Trong trường hợp đó hãy thiết định dữ liệu cha là gì ở bước này. Ví dụ là học sinh tiểu học toàn quốc, hay học sinh tiểu học của tỉnh A.)
- Từ dữ liệu mẫu tính toán công thức ước lượng. (Công thức ước lượng phụ thuộc vào đối số của dữ liệu cha. Nếu ta quan tâm tỷ suất của dữ liệu cha, thì thông thường công thức ước lượng sẽ là tỷ suất, tức là ta phải điều tra tỷ suất của dữ liệu mẫu.)
- Tôi muốn ước lượng dữ đối số của dữ liệu cha bằng công thức ước lượng nhưng tôi không làm được vì vậy tôi đã thiết lập một khoảng tin cậy, ví dụ khoảng tin cậy 95%. Tôi sẽ giải thích trong các bài viết sau.
- Xem xét phân bố của dữ liệu mẫu bằng công thức ước lượng. Phân bố dữ liệu mẫu đã được nói Bài học 18.
- Tính toán các giá trị có thể có của khoảng tin cậy từ phân bố dữ liệu mẫu.

Nói một cách đại khái, nó như những gì tôi đã tóm tắt ở trên. Sau khi hoàn thành các bước này, bạn có thể nói rằng, đối số của dữ liệu cha nằm trong khoảng từ A đến B với khoảng tin cậy là $XX\%$.
Ví dụ, chiều cao trung bình của nam giới trưởng thành nằm trong khoảng từ $168cm \sim 172cm$ với độ tin cậy là 95%. 95% có nghĩa là trong đó có 5% có thể có sai sót.
Việc đột ngột chỉ ra những trình tự như trên có lẽ sẽ là khó hiểu. Do đó trong bài viết sau, tôi sẽ lấy ví dụ cụ thể để chúng ta ước lượng khoảng, từ đó sẽ hiểu hơn. Khi thử làm thực tế vài lần, các bạn sẽ nhớ ngay.

23.5 Tổng kết

Bài viết này nói về thống kê suy luận, và nó toàn chữ là chữ, nhưng có thể tóm tắt như sau.

- Thống kê suy luận có hai phần chính là ước lượng và kiểm định.
- Ước lượng có nghĩa là ta trích xuất lấy ngẫu nhiên dữ liệu mẫu từ dữ liệu cha và sử dung nó để ước lượng đối số (tham số) của dữ liệu cha.
- Kiểm định có nghĩa là giả định tính chất nào đó của dữ liệu cha, sử dụng dữ liệu mẫu để điều tra xem có mâu thuẫn nào hay không.
- Trong ước lượng có ước lượng điểm và ước lượng khoảng.
- Ước lượng điểm là sử dụng công thức ước lượng của dữ liệu mẫu để ước lượng tham số tổng thể. Trong thực tế thường sử dụng ước lượng khoảng.

Bạn không cần phải nhớ các trình tự đã nêu trong bài học này, sau khi thực hành thực tế, các bạn sẽ hiểu cách làm.
Trong bài học tới chúng ta sẽ thử ước lượng khoảng "tỷ suất" của dữ liệu cha.

Bài 24

Ước lượng khoảng tỷ suất

24.1 Thử ước lượng khoảng tỷ suất

Trong Bài học 23 chúng ta đã nói với nhau về việc sử dụng dữ liệu mẫu thông qua công thức ước lượng để ước lượng đối số của dữ liệu cha.
Ví dụ, để ước lượng tỷ suất của dữ liệu cha, ta phải làm như thế nào?
Tỷ suất (ratio) là gì? Ví dụ trong một lớp học có tỷ lệ nam nữ, hay duy trì tỷ suất nam nữ trong nội các chính phủ. Hay tỷ lệ sản phẩm lỗi trong một nhà máy. Tỷ suất là một chỉ số thường được dùng trong cuộc sống hàng ngày của chúng ta.
Ví dụ bản tin bầu cử của một nước phát đi khi mà cuộc bầu cử còn đang diễn ra, người ta có thể ứng dụng tỷ suất để ước lượng kết quả bầu phiếu. Tại sao lại có thể đánh giá chắc chắn chiến thắng của một ứng cử viên trong khi tỷ lệ kiểm phiếu mới chỉ là 1%. Có ổn không khi còn 99% phiếu bầu chưa được kiểm tra?
Điều này là vì người ta coi dữ liệu cha là toàn bộ những phiếu bầu. Và dữ liệu mẫu là 1% phiếu bầu trong dữ liệu cha. Từ tỷ suất của dữ liệu mẫu, có thể ước lượng tỷ suất của dữ liệu cha với xác suất là bao nhiêu.
Ví dụ, nếu trong 99% phiếu bầu chưa được kiểm tra mà duy trì được tỷ suất hơn 50% thì dự đoán người chiến thắng trong cuộc bầu cử là hoàn toàn chính xác.
Thống kê học có thể được ứng dụng trong những trường hợp như vậy. Có thể nói rằng thống kê là bộ môn có tính thực chiến (ứng dụng trong thực tiễn) là vì vậy.

24.2 Các bước cụ thể khi ước lượng khoảng tỷ suất

Trong bài viết trước tôi đã nêu các bước để ước lượng khoảng nhưng trong bài học này chúng ta sẽ thử ước lượng khoảng thông qua thao tác làm thực tế.
Trong bài học này, tôi sẽ lấy ví dụ thường thấy trong thực tế, đó là tỷ suất trong bầu cử. Hãy xem một ví dụ đơn giản về việc bỏ phiếu cho ứng cử viên A và ứng cử viên B.
Chúng ta xem xét trong trường hợp tất cả các phiếu bầu đều hợp lệ, không có phiếu bầu nào phạm quy. Người chiến thắng đơn giản là người chiếm được đa số phiếu bầu (tức là phải đạt hơn 50%).

Giả định rằng có 100.000 người đã bỏ phiếu. Để kiểm tra toàn bộ phiếu bầu này thật sự rất mất thời gian. Mọi người không thể chờ đợi điều đó. Họ muốn biết kết quả càng sớm càng tốt.

Chúng ta cần tạo ra dữ liệu mẫu để từ đó ước lượng khoảng cho đối số của dữ liệu cha (trong trường hợp này là tỷ suất của dữ liệu cha). Bằng cách không kiểm tra toàn bộ tất cả phiếu bầu, chúng ta lấy ra một số lượng phiếu bầu trong đó một cách ngẫu nhiên, ta gọi là dữ liệu mẫu. Từ dữ liệu mẫu này, thử tiến hành ước lượng tỷ suất của dữ liệu cha (100.000 phiếu bầu).

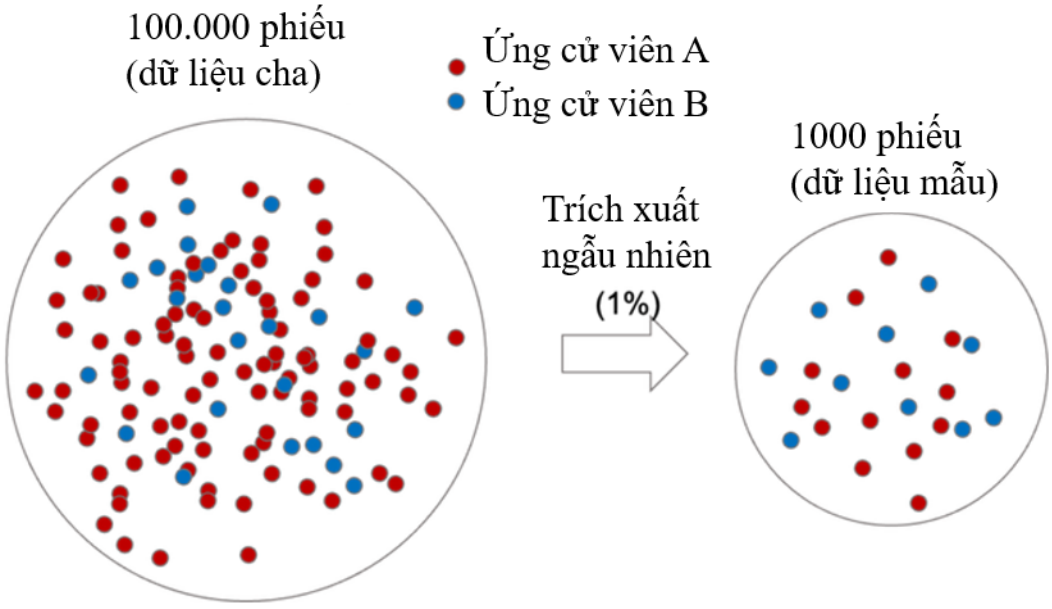
1. Trích xuất ngẫu nhiên dữ liệu từ dữ liệu cha để tạo dữ liệu mẫu. Trong trường hợp đã có dữ liệu mẫu thì thiết định đâu là dữ liệu cha.

Đầu tiên ta cần tạo dữ liệu mẫu. Trong trường hợp đã có dữ liệu mẫu, hãy thiết định dữ liệu cha là gì trong phạm vi mà chúng ta có thể tiến hành. Như trong bài học trước đã nói, **dữ liệu mẫu là lý tưởng nếu nó được trích xuất ngẫu nhiên từ dữ liệu cha**. Tuy nhiên hầu hết các trường hợp không thể nào trích xuất ngẫu nhiên. Trường hợp như vậy hãy cố gắng làm sao để trích xuất càng có tính ngẫu nhiên, hay trông có vẻ ngẫu nhiên thì càng tốt. Ví dụ trong trường hợp này, chúng ta không lấy phiếu bầu từ một điểm bầu cử mà sẽ trích xuất phiếu bầu từ nhiều điểm bầu cử khác nhau. Nếu có thể thì trích xuất phiếu bầu ở các thời điểm khác nhau, khung thời gian khác nhau nữa thì càng tốt.

Trong thống kê suy luận, đây là bước **vô cùng quan trọng**. Tất cả các bước sau này đều sẽ dựa trên mẫu dữ liệu này và dữ liệu cha ở tại bước này.

Tôi thường thấy các trường hợp thiết định dữ liệu cha không rõ ràng, hoặc thiết định dữ liệu cha một cách không hợp lý, vì vậy không thể ước lượng bằng mẫu đó, khi đó việc ước lượng khoảng được tiến hành một cách khiên cưỡng (cưỡng bức). Do đó các bạn hãy chú ý.

Lần này chúng ta sẽ trích xuất ngẫu nhiên 1000 phiếu bầu (= 1%) từ dữ liệu cha có 100.000 phiếu bầu để tạo dữ liệu mẫu.



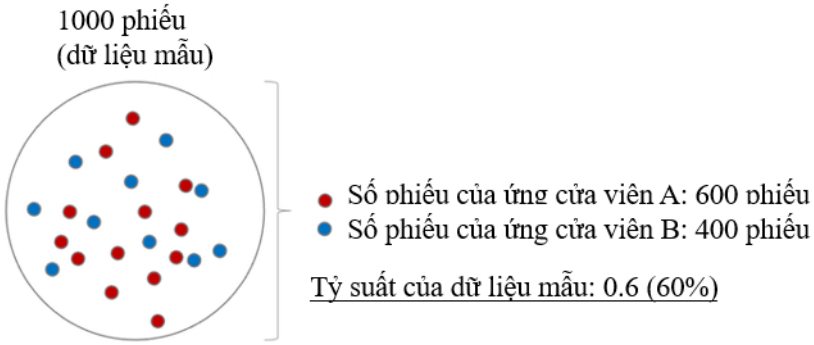
2. Tính công thức ước lượng từ dữ liệu mẫu.

Lần này chúng ta muốn ước lượng đối số của dữ liệu cha đó là tỷ suất. Vậy thì có thể sử dụng giá trị như thế nào của dữ liệu mẫu với tư cách là công thức ước lượng?
Trong nhiều trường hợp, đại lượng thống kê dữ liệu mẫu (tỷ suất) sẽ tương ứng với đối số của dữ liệu cha (là thứ mà ta đang quan tâm - tỷ suất), được sử dụng để làm công cụ ước lượng. Nói tóm lại, trong trường hợp này ta sẽ sử dụng tỷ suất của dữ liệu mẫu làm công cụ ước lượng tỷ suất của dữ liệu cha.
Giả sử đối số của dữ liệu cha là giá trị trung bình, khi đó ta sẽ sử dụng giá trị trung bình của dữ liệu mẫu làm cơ sở ước lượng. Ngoài ra hệ số tương quan của dữ liệu mẫu có thể được sử dụng như một công cụ ước lượng.
Trong trường hợp giá trị kỳ vọng của công thức ước lượng trùng với giá trị của đối số của dữ liệu cha, thì có thể nói rằng công cụ ước lượng đó có tính bất thiên (công bằng, không thiên vị).
Trong Bài học 6, Bài học 7 tôi đã nói về tính bất thiên, tuy nhiên nếu có tính bất thiên (công bằng, không thiên vị), thì nó phù hợp hơn với vai trò là công cụ ước lượng.
Như đã nói trong Bài học 7, giá trị trung bình của dữ liệu mẫu có tính bất thiên, ta biết rằng, tỷ suất cũng có tính bất thiên.

Bổ sung

Nếu bỏ phiếu cho ứng cử viên A thì là 1, bỏ phiếu cho ứng cử viên B thì là 0. Khi suy nghĩ như thế ta thấy rằng, tỷ suất cũng có thể được tính toán giống với giá trị trung bình. Từ đó có thể nói tỷ suất là **trường hợp đặc biệt của trung bình cộng**. Vì vậy có thể nói rằng nếu giá trị trung bình cũng có tính bất thiên thì tỷ suất cũng có tính bất thiên.

Trong ví dụ này ta giả định rằng ứng cử viên có lượng phiếu bầu là 0.6 tức là chiếm 60% phiếu bầu, nói cách khác là được 600 phiếu bầu (= tỷ suất dữ liệu mẫu).



3. Thiết định khoảng tin cậy.

Khi nhìn vào kết quả của 1000 phiếu bầu thấy rằng tỷ suất của dữ liệu mẫu (tỷ lệ phiếu bầu của ứng cử viên A) là 0.6 tức 60%. Thế nhưng tỷ suất của dữ liệu cha (100.000 phiếu bầu) không phải lúc nào cũng là 0.6 đúng không nào?
Vì vậy ta cần thiết định khoảng tin cậy. Ví dụ tỷ suất của dữ liệu cha nằm trong khoảng $0.56 \sim 0.64$ với khoảng tin cậy là 95%. Ta có thể nói rằng tỷ suất dữ liệu cha là 0.6 nhưng không biết độ chính xác là bao nhiêu.

Khoảng tin cậy (confidence interval) thường được viết tắt là **CI**.

Như vậy ta cần thiết định khoảng tin cậy, tức là mức độ chính xác. Ví dụ khi nãy ta nói khoảng tin cậy là 95%. Thế nhưng một ai đó nói, 95% thì chưa được, tôi cần khoảng tin cậy 99%. Khi đó tôi e rằng có thể lúc này sẽ là, tỷ suất của dữ liệu cha nằm trong khoảng $0.5 \sim 0.7$ với khoảng tin cậy 99%. Lúc này phạm vi được nới rộng ra và tin cậy cũng trở nên cao hơn. Xác suất để tỷ suất của dữ liệu cha nằm trong khoảng này ($0.5 \sim 0.7$) có khả năng cao hơn so với khoảng trước đó ($0.56 \sim 0.64$).

Lúc này có thể bạn chưa hiểu khoảng tin cậy 95% hay 99% có ý nghĩa như thế nào, nhưng tôi nghĩ bạn có thể sẽ hiểu thêm sau khi tôi giải thích toàn bộ các bước để thực hiện ước lượng khoảng. Vì vậy xin đừng lo lắng.

Vậy làm thế nào để bạn thiết lập khoảng tin cậy? Nó phụ thuộc vào độ chính xác cần thiết để ước lượng. Tùy vào từng trường hợp mà có sự khác nhau.

Tuy nhiên thông thường **khoảng tin cậy thường là** 95%. Điều này cho chúng ta cảm giác "Trong số 100 lần thì không chắc bạn sẽ mắc lỗi 5 lần". Thỉnh thoảng khoảng tin cậy 99% cũng được sử dụng nhưng trong khóa học này thông thường sẽ áp dụng khoảng tin cậy 95%.

4. Xem xét phân bố dữ liệu mẫu của công thức ước lượng.

Nào, sau khi đã thiết định mẫu, công thức ước lượng cũng đã được tính toán, chúng ta cần xem xét phân bố của dữ liệu mẫu.

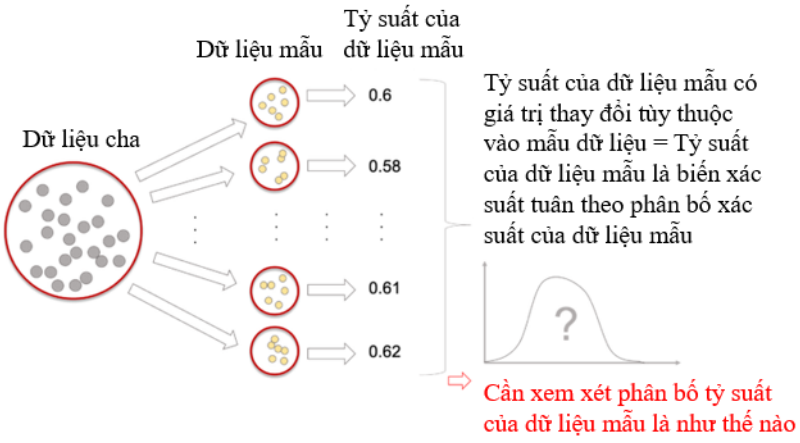
Tại sao phân bố của dữ liệu mẫu lại cần thiết?

Tỷ suất dữ liệu mẫu lần này (công thức ước lượng) chúng ta đã tính toán được là 0.6 (60%). Tuy nhiên nếu như một thời điểm nào đó chúng ta lấy 1000 lá phiếu khác một cách ngẫu nhiên thì có thể sẽ lại ra kết quả khác đúng không nào?

Chính vì thế mà chúng ta cần xem xét tỷ suất của dữ liệu mẫu được phân bố như thế nào, phân bố dữ liệu mẫu trong trường hợp này chính là phân bố của tỷ suất của dữ liệu mẫu là thứ chúng ta cần quan tâm.

Phân bố dữ liệu mẫu là phân bố xác suất tuân theo công thức thống kê của dữ liệu mẫu đã được nêu trong Bài học 18.

Nói tóm lại, tỷ suất của dữ liệu mẫu 0.6 thu được lần này là may mắn hay là giá trị thường thu được, để hiểu được điều này chỉ có thể thông qua phân bố dữ liệu mẫu.



Vậy làm thế nào để lấy được phân bố của dữ liệu mẫu? Nó có thể được tạo ra từ giá trị thực tế của dữ liệu cha.

> Chẳng phải chúng ta cũng không biết giá trị thực tế của dữ liệu cha hay sao?

Đúng là như thế, vì không biết giá trị thực tế của dữ liệu cha, nên chúng ta sẽ tiến hành ước lượng khoảng và để ước lượng khoảng thì chúng ta cần biết giá trị thực của dữ liệu cha.

Đó là lúc tỷ suất của dữ liệu mẫu phát huy tác dụng, nhưng đầu tiên chúng ta giả định tỷ suất của dữ liệu cha là p và chúng ta sử dụng giá trị p này để tìm hiểu phân bố của dữ liệu mẫu sẽ như thế nào.

Vậy phân bố dữ liệu mẫu (ở đây là tỷ suất) sẽ như thế nào? Thực ra đây là phân bố nhị thức.

> Cái gì cơ, tại sao phân bố dữ liệu mẫu (ở đây là tỷ suất) lại là phân bố nhị thức?

Phân bố nhị thức đã được giải thích trong Bài học 19 nhưng phân bố nhị thức là phân bố xác suất như thế nào?

Phân bố nhị thức là phân bố xác suất tuân theo "Trong n lần quan sát (thực nghiệm), xác suất để sự kiện đó xảy ra x lần". Biết rằng xác suất để một sự kiện xảy ra là p trong 1 lần quan sát.

Xác suất một sự kiện xảy ra trong 1 lần quan sát giống với ví dụ lần này của chúng ta chính là "tỷ suất". Trong 100.000 phiếu, lấy ngẫu nhiên 1 phiếu, xác suất để lá phiếu này là bầu cho ứng viên A chắc chắn trùng với tỷ suất p của dữ liệu cha.

"Trong n lần quan sát" có nghĩa là trong 100.000 lá phiếu, lấy ra 1000 phiếu ($= n$ phiếu).

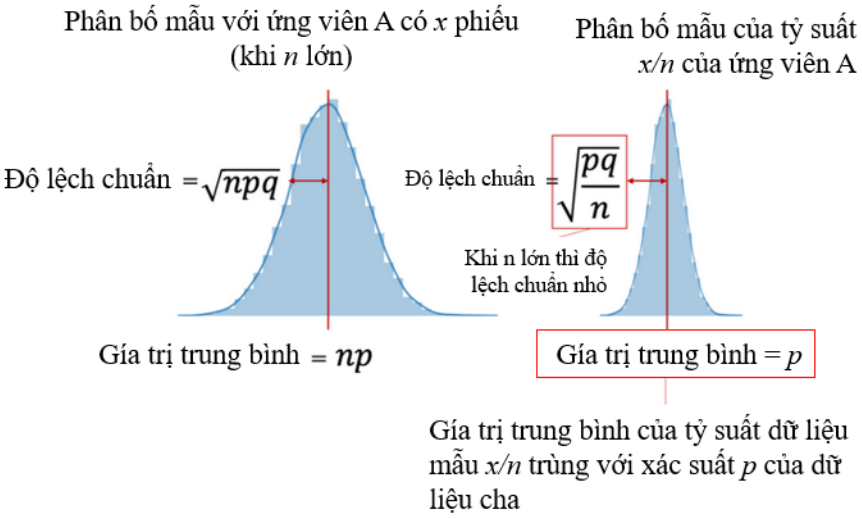
Nói tóm lại, phát biểu về phân bố nhị thức có thể thay đổi bằng cách nói như sau:

Gọi p là tỷ suất phiếu bầu ứng viên A trong dữ liệu cha (100.000 **phiếu). Từ dữ liệu cha lấy ngẫu nhiên n phiếu, phân bố xác suất mà ở đó số phiếu ứng viên A xuất hiện x lần là phân bố nhị thức.**

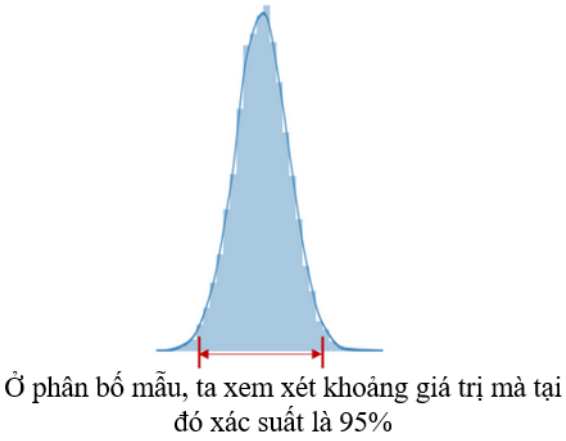
Số phiếu bầu cho ứng viên A là x, nếu biết x thì có thể tính được tỷ suất là $\frac{x}{n}$. Do đó nếu bạn biết phân bố xác suất để phiếu bầu ứng viên A là x (là phân bố nhị thức) thì tất yếu bạn sẽ biết phân bố dữ liệu mẫu.

Trong Bài học 22, chúng ta đã biết giá trị trung bình của phân bố nhị thức là np, phân tán(phương sai) là $npq\,(q = 1 - p)$. Hơn nữa, khi n càng lớn, phân bố nhị thức sẽ càng gần giống với phân bố chuẩn.

Do đó phân bố dữ liệu mẫu của tỷ suất x/n có thể lấy xấp xỉ theo phân bố chuẩn, giá trị trung bình là p và độ lệch chuẩn là $\sqrt{pq/n}$.



Như vậy từ phân bố mẫu, ta xem xét khoảng giá trị mà giá trị lấy từ đây có xác suất 95%. Tóm lại, từ dữ liệu mẫu thông qua phân bố của nó, dựa trên giá trị 0.6, ta tính toán các giá trị cho xác suất trung bình p là 95%.



Ta có thể tính toán dựa trên phân bố chuẩn nhưng nếu chuẩn hóa thì sẽ dễ hiểu hơn. Các bạn hãy xem xét chuẩn hóa bằng cách tính điểm z. Các bạn có thể xem lại Bài học 9.

$$z = \frac{\frac{x}{n} - p}{\sqrt{\frac{pq}{n}}}$$

Cách tính điểm z sẽ đưa dữ liệu có giá trị trung bình là 0 và có độ lệch chuẩn là 1. Chúng ta sẽ sử dụng phân bố này, dựa vào khoảng tin cậy để tính toán phạm vi đối số dữ liệu cha p.

5. Tính toán các giá trị có thể có của khoảng tin cậy (độ chính xác) từ phân bố lấy mẫu. Điểm z có nghĩa là phân bố mẫu của tỷ suất x/n đã được chuẩn hóa. Ví dụ, nếu chúng ta cần tìm khoảng giá trị mà ở đó tập trung 95% dữ liệu, chúng ta có thể tính toán khoảng giá trị của p với xác suất 95%.

Với phân bố chuẩn tắc, ở Bài học 8 chúng ta đã biết khoảng giá trị từ $-1.96 \sim 1.96$ sẽ

có xác suất là 95%. Tóm lại là:

$$-1.96 < \frac{\frac{x}{n} - p}{\sqrt{\frac{pq}{n}}} < 1.96$$

Công thức trên có nghĩa là nếu chúng ta biết p, ta có thể tìm được khoảng giá trị mà tỷ suất của dữ liệu cha p có độ chính xác là 95%.
Trong ví dụ của chúng ta, ta thay $x = 600, n = 1000, q = 1 - p$ và giải bất phương trình trên. Kết quả ta sẽ thu được p_1, p_2 với $p_1 < p_2$, đây chính là khoảng giá trị mà đối số dữ liệu cha (tỷ suất của dữ liệu cha) có thể lấy mà ở đó khoảng tin cậy là 95%.
Trong thực tế **chúng ta không cần phải tự mình giải bất phương trình trên**, các phần mềm thống kê như SPSS, R, Python có thể giúp chúng ta làm việc này.
Trong khóa học này, ngôn ngữ của chúng ta là Python, do đó chúng ta sẽ thử dùng Python để tìm khoảng ước lượng. Thông qua code cụ thể, các bạn sẽ hiểu được bài học này.
Vậy thì trong thực tế từ 100.000 lá phiếu, tỷ suất dữ liệu mẫu (1000 lá phiếu) là 0.6, với khoảng tin cậy là 95% thì đối số của dữ liệu cha (tỷ suất) có khoảng ước lượng là gì?

24.3 Sử dụng Python để tìm khoảng ước lượng

Sử dụng Module `stats` của thư viện `scipy` có thể tính được khoảng ước lượng.
Cho tới thời điểm này, nếu là phân bố nhị thức ta có `scipy.stats.binorm` trong Bài học 19, nếu là phân bố chuẩn thì sử dụng `scipy.stats.norm` nêu trong Bài học 22, ta có thể tạo được phân bố xác suất tùy ý.
Ứng với các phân bố xác suất nói trên, ta có thể sử dụng `.interval()` để tìm ước lượng khoảng.
Trong bài học này ta sẽ sử dụng phân bố nhị thức (`scipy.stats.binorm`) để thử ước lượng khoảng xem sao nhé. (Khi n càng lớn thì càng gần (xấp xỉ) với phân bố chuẩn nhưng thực tế tôi nghĩ nên sử dụng phân bố nhị thức thay vì sử dụng phân bố chuẩn.)
`scipy.stats.binorm.interval()` có 3 đối số mà ta cần truyền giá trị cho chúng.
alpha: Khoảng tin cậy (Nếu là 95% thì truyền vào là 0.95)
n: Độ lớn của dữ liệu mẫu (Số lần thực nghiệm. Lần này là 1000)
p: Tỷ suất của dữ liệu mẫu (Nếu là 60% thì truyền vào là 0.6)

```
stats.binom.interval(0.95, n=1000, p=0.6)
```

Kết quả:

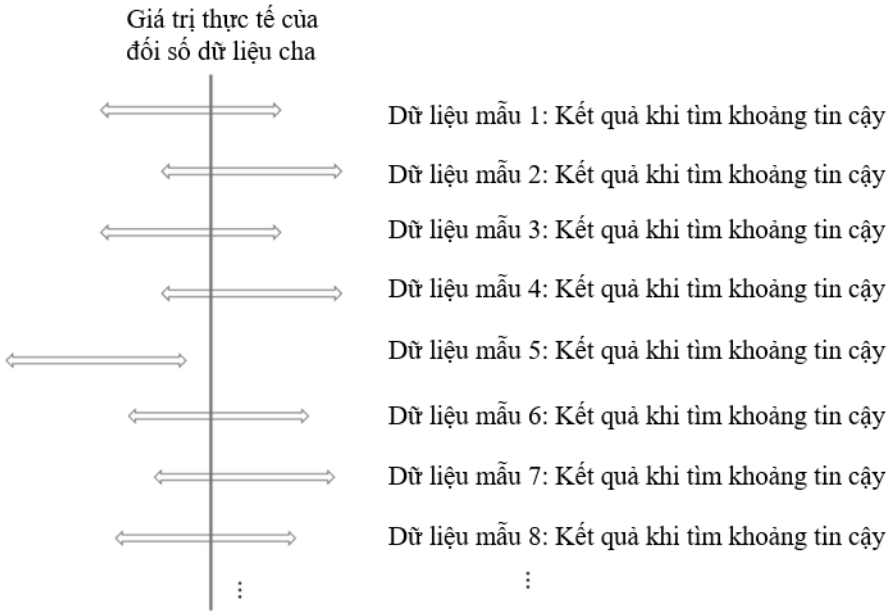
```
(570.0, 630.0)
```

Chỉ có vậy thôi, thật đơn giản phải không nào. Không chỉ là Python, với phần mềm thống kê khác, chúng ta cũng chỉ mất một dòng code để tính toán. Bạn không hiểu gì cũng có thể sử dụng code này để tìm ra kết quả. Tuy nhiên nếu như không hiểu logic, chúng ta có thể sử dụng phân bố xác suất không đúng và không thể sử dụng kết quả tính toán một cách chính xác, do đó xin hãy lưu ý. Kết quả trả về là kiểu **`tuple`** có hai số là 570 và 630.

Điều này có nghĩa là **dù lấy bao nhiêu lần thì cứ hễ trích xuất ngẫu nhiên** 1000 **phiếu,** 95% **phiếu bầu cho ứng cử viên A nằm trong khoảng** 530 **phiếu tới** 630 **phiếu**.
Nói tóm lại điều này có nghĩa là tỷ suất của dữ liệu cha nằm trong khoảng từ 0.57 (57%) ∼ 0.63 (63%) với độ chính xác là 95%.

24.4 Khoảng tin cậy 95% nghĩa là gì (độ chính xác 95% nghĩa là gì)

Khoảng tin cậy 95% nghĩa là khi lặp đi lặp lại công việc ước lượng khoảng thì 95% xác suất đối số của dữ liệu cha (trường hợp ví dụ này là tỷ suất) nằm trong khoảng giá trị đó.
Tóm lại, nếu lấy dữ liệu mẫu một cách ngẫu nhiên 100 lần và tìm ra khoảng ước lượng thì ước chừng có 95 lần khoảng mà chúng ta ước lượng chưa đối số của dữ liệu mẫu.



Trong ví dụ lần này của chúng ta, có thể nói rằng kết quả khi ước lượng khoảng từ dữ liệu mẫu với khoảng tin cậy 95% là: Tỷ suất của dữ liệu cha nằm trong khoảng 0.57 (57%) ∼ 0.63 (63%) với độ chính xác 95%.
Điều đó có nghĩa là nếu chúng ta lại lấy 1000 phiếu bầu khác một cách ngẫu nhiên và nếu ước lượng khoảng, ví dụ được khoảng là 0.56 ∼ 0.62, tỷ suất của dữ liệu mẫu chắc chắn sẽ có biến động một chút. Nếu lại lấy 1000 phiếu bầu khác ⋯ Cứ tiến hành nhiều lần như vậy, lấy dữ liệu mẫu và ước lượng khoảng, 95% có thể ước lượng chính xác. Khoảng tin cậy 95% có ý nghĩa như vậy. Vì vậy kết quả 0.57 ∼ 0.63 có nghĩa là có xác suất 5% sai sót.

Bổ sung

95% tỷ suất của dữ liệu cha nằm trong khoảng $0.57\,(57\%) \sim 0.63\,(63\%)$, cách nói này có thể gây nhầm lẫn. Chúng ta cần chú ý rằng, tỷ suất thực tế của dữ liệu cha là một giá trị cố định không thay đổi, tuy nhiên đây là giá trị chúng ta không biết chính xác, do đó cách nói trên dễ làm các bạn hiểu nhầm rằng tỷ suất của dữ liệu cha là một biến xác suất, có giá trị thay đổi, điều đó là không đúng. Cái thay đổi ngẫu nhiên lần này không phải là tỷ suất thực tế của dữ liệu cha, mà là khoảng giá trị ước lượng $0.57\,(57\%) \sim 0.63\,(63\%)$. Khoảng giá trị này biến động tùy thuộc vào sự biến động của dữ liệu mẫu. Vì vậy cách nói sau là chính xác, tỷ suất của dữ liệu cha nằm trong khoảng $0.57\,(57\%) \sim 0.63\,(63\%)$ với xác suất 95%, tức là vẫn có 5% khả năng tỷ suất của dữ liệu cha nằm ngoài khoảng trên.

Trong ví dụ này ta trích xuất 1000 phiếu, với khoảng tin cậy 95% và tìm ra tỷ suất nằm trong khoảng $0.57 \sim 0.63$. Các bạn hãy thử làm với code Python trên với khoảng tin cậy là 99.99% bằng cách thay `alpha` là 0.9999, các bạn sẽ thấy khoảng ước lượng lúc này sẽ là $0.539 \sim 0.66$.
Tóm lại, ở thời điểm này tôi nghĩ rằng có thể nói rằng ứng viên A chắc chắn sẽ trúng cử.(Trong thực tế các bản tin về bầu cử có thể xem xét các yếu tố khác, nhưng ý tưởng cơ bản là áp dụng khoảng tin cậy.)

24.5 Tổng kết

Bài học này đã rất dài nhưng có thể tóm tắt nội dung như sau.

- Ước lượng khoảng cho tỷ suất được sử dụng ở xung quanh cuộc sống chúng ta.
- Lý tưởng đó là có thể trích xuất dữ liệu mẫu một cách ngẫu nhiên. Trong trường hợp đã có dữ liệu mẫu, ta cần thiết định dữ liệu cha một cách hợp lý.
- Công thức thống kê được sử dụng trong ước lượng thường trùng với đối số dữ liệu cha mà ta muốn ước lượng.
- Công thức ước lượng tỷ suất của dữ liệu cha là tỷ suất của dữ liệu mẫu.
- Phân bố mẫu của tỷ suất của dữ liệu mẫu là phân bố nhị thức. Khi dữ liệu mẫu đủ lớn thì nó gần với phân bố chuẩn.
- Bằng cách chuẩn hóa, khoảng giá trị chiếm 95% dữ liệu là $-1.96 \sim 1.96$.
- Sử dụng `.interval()` của `.stats`<phân bố xác suất> để ước lượng khoảng.
- Khoảng tin cậy 95% có nghĩa là kết quả ước lượng khoảng có xác suất đúng là 95%. (95% đối số của dữ liệu cha nằm trong khoảng đó là cách nói không đúng. Chú ý đối số của dữ liệu cha là một giá trị cố định không có tính ngẫu nhiên.)

Trong bài học này tôi đã đề cấp đến vấn đề thống kê trong thực tế. Riêng bài viết này, tôi đã sử dụng nhiều kiến thức thống kê khác nhau mà tôi đã học được cho đến nay.
Trong các bài học sau, nhiều nội dung khác nhau trong quá khứ sẽ xuất hiện, vì vậy, hãy tiếp tục xem lại các bài đã học để nắm vững kiến thức.
Cũng giống như tỷ suất, trong bài viết tiếp theo, chúng ta sẽ thử ước lượng khoảng cho giá trị trung bình.

Bài 25

Giải thích dễ hiểu về ước lượng khoảng cho giá trị trung bình

Tôi hi vọng các bạn cảm thấy Bài học 24 dễ hiểu, trong bài viết đó chúng ta đa cùng nhau ước lượng khoảng cho tỷ suất của dữ liệu cha. Trong bài học lần này chúng ta sẽ cùng nhau ước lượng khoảng cho giá trị trung bình của dữ liệu cha.
Cũng giống như tỷ suất, giá trị trung bình của dữ liệu cha là đại lượng thường được ước lượng.
Chúng ta muốn biết chiều cao trung bình của những người nam giới trưởng thành trên toàn quốc. Chúng ta muốn biết sức bền trung bình của các sản phẩm trong một nhà máy. Chúng ta muốn biết giá rau củ trung bình ngoài thị trường .v.v.

25.1 Chìa khóa của ước lượng khoảng là phân bố mẫu

Cũng như bài trước, câu hỏi quan trọng đó là phân bố mẫu trông như thế nào?
Nếu biết về phân bố mẫu, ta có thể thiết định khoảng tin cậy và tính toán giá trị.
Ước lượng khoảng cho giá trị trung bình thì điều đương nhiên chúng ta sẽ **sử dụng giá trị trung bình của dữ liệu mẫu**. Phân bố của giá trị trung bình của dữ liệu mẫu là phân bố như thế nào là điều cần xem xét.

25.2 Giải thích các bước về ước lượng khoảng cho giá trị trung bình

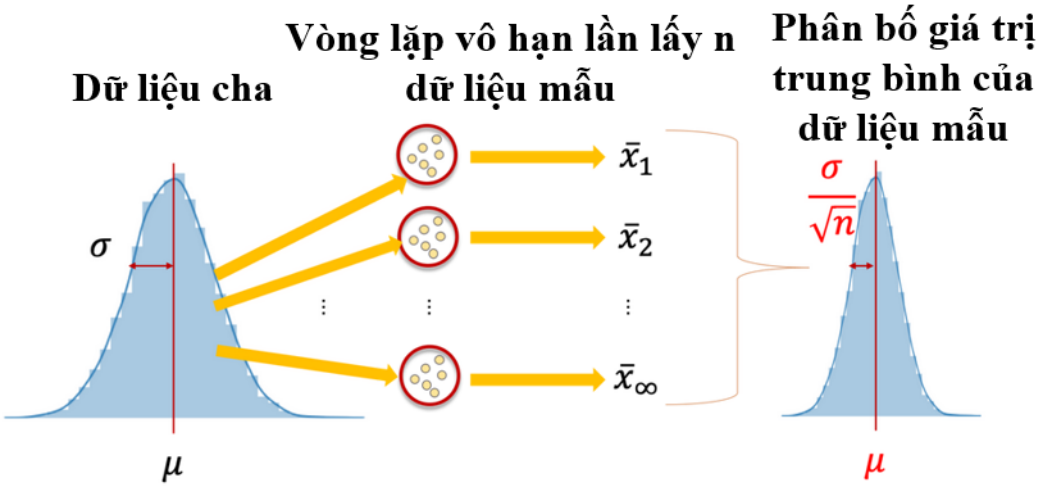
Bước 1: Tạo dữ liệu mẫu bằng cách trích xuất dữ liệu ngẫu nhiên từ dữ liệu cha.
Gọi giá trị trung bình của dữ liệu cha là μ, độ lớn của dữ liệu mẫu là n.
Bước 2: Tính công thức ước lượng từ dữ liệu mẫu.
Như bài trước đã nói, giá trị trung bình của dữ liệu mẫu là công thức ước lượng bất thiên của giá trị trung bình μ của dữ liệu cha.
Ta gọi giá trị trung bình của dữ liệu mẫu là $\bar{x}$.
Bước 3: Quyết định khoảng tin cậy.

Chỉ với giá trị trung bình của dữ liệu mẫu $\bar{x}$ mà ước lượng điểm cho giá trị trung bình của dữ liệu cha μ sẽ hiến chúng ta bất an lo lắng về độ chính xác, do đó chúng ta xem xét thiết định khoảng tin cậy để ước lượng khoảng. Cũng như bài học trước, chúng ta có thể thiết định khoảng tin cậy là 95%.

Bước 4: Xêm xét phân bố mẫu của ước lượng. Đây là điểm **quan trọng**. Giá trị trung bình của dữ liệu mẫu sẽ có phân bố như thế nào?

Như trong Bài Học 7 đã nói, trung bình của phân bố mẫu của giá trị trung bình mẫu $\bar{x}$ trùng khớp với giá trị trung bình của dữ liệu cha μ. Với σ là độ lệch chuẩn của dữ liệu cha, thì phân tán của phân bố mẫu của giá trị trung bình mẫu $\bar{x}$ sẽ là $\frac{\sigma^2}{n}$ (Chú ý độ lệch chuẩn trong biểu đồ dưới là $\frac{\sigma}{\sqrt{n}}$).

Đây là điều rất quan trọng, nếu quên, bạn hãy xem lại Bài Học 7.



Chẳng phải chúng ta đã biết phân bố mẫu của giá trị trung bình của mẫu rồi hay sao?

Đúng là chúng ta đã biết phân tán và giá trị trung bình của phân bố mẫu của giá trị trung bình của mẫu nhưng nó là phân bố như thế nào thì chưa biết.

Dữ liệu cha nếu là phân bố chuẩn, thì phân bố của dữ liệu mẫu cũng là phân bố chuẩn (Tôi nghĩ điều này là dễ hình dung ra được) nhưng **khi dữ liệu cha không phải là phân bố chuẩn thì phân bố dữ liệu mẫu sẽ như thế nào?**

Thực ra khi độ lớn của dữ liệu mẫu n đủ lớn thì phân bố mẫu của giá trị trung bình của dữ liệu mẫu $\bar{x}$ sẽ gần giống (xấp xỉ) với phân bố chuẩn.

Tóm lại, **bất kể phân bố của dữ liệu cha là phân bố gì đi nữa, nếu tăng n thì phân bố mẫu của $\bar{x}$ sẽ trở thành phân bố chuẩn**.

Phân bố chuẩn thật đáng sợ. Nó thực sự xuất hiện ở nhiều nơi. Định lý này được gọi là **định lý cực hạn trung tâm** (các tài liệu khác sử dụng thuật ngữ "định lý giới hạn trung tâm", tên tiếng anh: Central Limit Theorem). Hãy nhớ rằng đây là định lý rất quan trọng trong lý thuyết thống kê.

Nói tóm lại:

- Nếu phân bố dữ liệu cha là phân bố chuẩn thì không cần quan tâm độn lớn của dữ liệu mẫu n, phân bố mẫu của giá trị trung bình của dữ liệu mẫu $\bar{x}$ sẽ là phân bố chuẩn.

- Phân bố dữ liệu cha dù không phải là phân bố chuẩn đi chăng nữa, nếu độ lớn dữ liệu mẫu n đủ lớn, phân bố mẫu sẽ tiệm cận tới phân bố chuẩn.

Trong bài viết này ta giả định n đủ lớn, và hãy xem xét phân bố mẫu như là một phân bố chuẩn.
Khi n đủ lớn thì phân bố mẫu của giá trị trung bình dữ liệu mẫu $\bar{x}$ sẽ được xem như là phân bố chuẩn với giá trị trung bình là μ và phân tán là $\dfrac{\sigma^2}{n}$. Trong phân bố này nếu lấy khoảng giá trị 95% ta sẽ lấy được khoảng giá trị chứa giá trị trung bình của dữ liệu cha có khoảng tin cậy là 95%.
Cũng như bài trước, ta hãy xem xét chuẩn hóa dữ liệu. Khi đó khoảng giá trị 95% là $-1.96 \sim 1.96$. Cách chuẩn hóa tính điểm z là:

$$z = \frac{\bar{x} - \mu}{\dfrac{\sigma}{\sqrt{n}}}$$

Bước 5: Tính toán giá trị có thể có của khoảng tin cậy từ phân bố mẫu.
Vậy thì hãy tính toán giá trị z có thể lấy bằng công thức trên với xác suất 95%. Trong Bài học 8 ta biết rằng 95% dữ liệu nằm trong khoảng $-1.96 \sim 1.96$.

$$-1.96 < \frac{\bar{x} - \mu}{\dfrac{\sigma}{\sqrt{n}}} < 1.96$$

Từ công thức này ta có thể tính toán ước lượng được μ.

Chẳng phải ta vẫn chưa biết độ lệch chuẩn σ của dữ liệu cha hay sao?

Đúng vậy, trong công thức trên có sử dụng tới độ lệch chuẩn của dữ liệu cha σ. Thông thường thì chúng ta không biết độ lệch chuẩn của dữ liệu cha. (Cũng như chúng ta không biết giá trị trung bình của dữ liệu cha μ vì vậy phải ước lượng.)

Ở đây, thay vì sử dụng độ lệch chuẩn của dữ liệu cha σ, chúng ta sử dụng độ lệch chuẩn của dữ liệu mẫu s'. Độ lệch chuẩn của dữ liệu mẫu s', là căn bậc hai của phân tán bất thiên s'^2. Phân tán bất thiên s'^2 đã được nói tới trong Bài học 6. Nó là công thức dưới đây:

$$s'^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

Do đó ta có công thức:

$$-1.96 < \frac{\bar{x} - \mu}{\dfrac{s'}{\sqrt{n}}} < 1.96$$

Biến đổi ta được:

$$\bar{x} - 1.96\frac{s'}{\sqrt{n}} < \mu < \bar{x} + 1.96\frac{s'}{\sqrt{n}}$$

Bổ sung

Thay vì sử dụng độ lệch chuẩn của dữ liệu cha σ, bằng việc sử dụng độ lệch chuẩn của dữ liệu mẫu s', phân bố của dữ liệu mẫu (phân bố mẫu) sau khi được chuẩn hóa không còn là phân bố chuẩn, nó đã trở thành phân bố có hình thù bị biến đổi một chút. Người ta gọi phân bố này là phân bố t. Nhưng khi n đủ lớn thì phân bố t sẽ tiệm cận (xấp xỉ) với phân bố chuẩn. Tôi sẽ giải thích chi tiết trong bài viết tiếp theo.

> Tôi chẳng hiểu công thức trên chút nào!

Hãy từ từ theo dõi từng bước, nó không phải là công thức khó đâu. Và tôi nghĩ bạn không nhớ công thức ở trên cũng không sao cả.
Trong thống kê suy luận, điều quan trọng là trình tự. Ví dụ, dữ liệu cha có phân bố không phải là phân bố chuẩn, cho dù n nhỏ nhưng ta vẫn sử dụng phân bố chuẩn, điều đó sẽ dẫn tới ước lượng sai.
Nếu không lý giải đầy đủ, ta không thể ước lượng đúng đắn, kết quả tính ra cũng không diễn giải đúng hiện thực của dữ liệu.
Sau khi đã diễn giải đúng, ta sẽ giao phó công việc tính toán cho các công cụ phần mềm tính toán.

25.3 Thử ước lượng khoảng cho giá trị trung bình trong thực tế

Cũng giống như trong Bài học 24, chúng ta có thể sử dụng `.interval()` đối với các phân bố xác suất trong Module `stats` của `scipy`.
Chúng ta sẽ thử ước lượng khoảng cho giá trị trung bình bằng dữ liệu thực tế.
Bạn có thể sử dụng dataset (dữ liệu trong thống kê) bất kỳ để thực hành. Trong khóa học Nhập Môn Python hướng tới Data Science bài 11, tôi đã giới thiệu Kaggle. Bạn có thể đăng ký tài khoản miễn phí và lấy các dataset ở trang này.
Trong bài học này tôi sẽ sử dụng data set US Household Income Statistics .
Tôi muốn xem dữ liệu về thu nhập trung bình của các khu vực ở Hoa Kỳ. (Lưu ý rằng mỗi bản ghi (mỗi dòng dữ liệu) là một thu nhập trung bình của một hộ gia đình theo từng khu vực.)
Sau khi download file csv về, ta sẽ nạp dữ liệu vào `DataFrame` của `Pandas`.

```
import pandas as pd
import seaborn as sns
%matplotlib inline

df = pd.read_csv('kaggle_income.csv', encoding="ISO-8859-1")
```

Chú ý, nếu không truyền đối số **encoding** thì sẽ xảy ra lỗi như dưới đây:

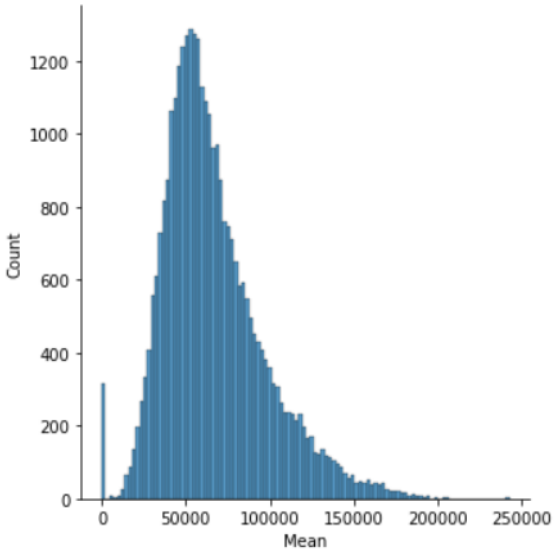
```
'utf-8' codec can't decode byte 0xf1 in position 2: invalid continuation byte
```

Tôi e rằng trong tệp dữ liệu chứa ký tự không thể đọc bằng utf-8.
Sau khi tra google, tôi biết cần phải chỉ định cho **encoding**, trong nhiều trường hợp chúng ta chỉ định **encoding="ISO-8859-1"** thì có thể đọc được dữ liệu.
Tiếp theo, để quan sát dữ liệu ta sử dụng **df.head()**.

	id	State_Code	State_Name	State_ab	County	City	Place	Type	Primary	Zip_Code	Area_Code	ALand	AWater	Lat	Lon	Mean	Median	Stdev	sum_w
0	1011000	1	Alabama	AL	Mobile County	Chickasaw	Chickasaw city	City	place	36611	251	10894952	909156	30.771450	-88.079697	38773	30506	33101	1638.260513
1	1011010	1	Alabama	AL	Barbour County	Louisville	Clio city	City	place	36048	334	26070325	23254	31.708516	-85.611039	37725	19528	43789	258.017685
2	1011020	1	Alabama	AL	Shelby County	Columbiana	Columbiana city	City	place	35051	205	44835274	261034	33.191452	-86.615618	54606	31930	57348	926.031000
3	1011030	1	Alabama	AL	Mobile County	Satsuma	Creola city	City	place	36572	251	36878729	2374530	30.874343	-88.009442	63919	52814	47707	378.114619
4	1011040	1	Alabama	AL	Mobile County	Dauphin Island	Dauphin Island	Town	place	36528	251	16204185	413605152	30.250913	-88.171268	77948	67225	54270	282.320328

Bạn có thể thấy hình thù của dữ liệu được hiện ra trước mắt. Mỗi một bản ghi đại diện cho một khu vực (Ví dụ: Thành phố Chickasaw, Quận Mobile, Alabama) và thu nhập trung bình ứng với khu vực đó nằm trong cột Mean.
Vậy thì hãy xem phân bố của cột Mean.

```
sns.displot(df['Mean'])
```



Về cách sử dụng **seaborn**, các bạn có thể tham khảo Nhập Môn Python hướng tới Data science bài học 24.

Bổ sung

seaborn mới nhất, **.distplot()** đưa ra cảnh báo không được dùng nữa, nghĩa là sẽ biến mất trong tương lai, vì vậy hãy sử dụng **.displot()**.

Phân bố này là phân bố của dữ liệu cha. Đó không phải là phân bố chuẩn. Hầu hết các trường hợp liên quan tới thu nhập hay dữ liệu về tiền tệ, đó không phải là phân bố chuẩn. (Tôi nghĩ là cũng có những giá trị nhiễu-giá trị ngoại lệ, lần này ta sẽ bỏ qua điều này.)
Nào, hãy thử nhìn vào giá trị trung bình của cột dữ liệu Mean. (Chú ý, lần này giá trị

chúng ta muốn ước lượng là giá trị trung bình của dữ liệu cha, thông thường thì chúng ta không biết giá trị này, do đó xin hãy lưu ý.)

```
import numpy as np
np.mean(df['Mean'])
```

Kết quả:

```
66703.98604193568
```

Ước chừng khoảng 66k đô-la. (Khoảng 1.5 tỷ VNĐ theo tỷ giá hiện tại).

Chú ý trung bình của trung bình không phải là trung bình của toàn bộ dữ liệu

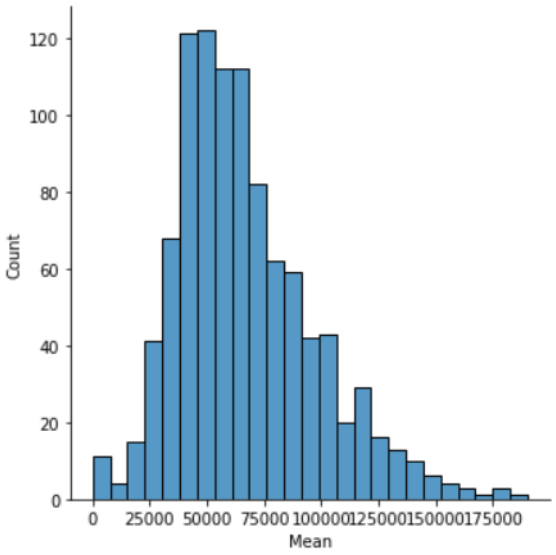
Trong trường hợp ví dụ này chúng ta muốn lấy giá trị trung bình của thu nhập của các vùng toàn nước Mỹ. Nhưng giá trị trung bình này không phải là giá trị trung bình của các hộ giá đình trong các khu vực trên toàn nước Mỹ. Do đó nó không phải là giá trị trung bình thu nhập của một hộ gia đình. Ví dụ trường tiểu học A và trường tiểu học B cùng làm một bài kiểm tra. 100 em học sinh trường tiểu học A có điểm trung bình là 60 điểm. 50 em học sinh trường tiểu học B có điểm trung bình là 70 điểm. Rõ ràng điểm trung bình của 150 em học sinh không phải là $(60 + 50)/2 = 55$ điểm. Giá trị trung bình đúng phải là $(100 \times 60 + 50 \times 70)/150 = 63.3$ điểm.
Theo cách này, giá trị trung bình trong trường hợp các giá trị có trọng số (100 người và 50 người) ta gọi là trung bình gia quyền, hay trung bình có trọng số.
Trong ví dụ này tôi chỉ muốn lấy giá trị trung bình của tất cả các vùng, vì vậy tôi không cần trung bình có trọng số.

Nào, chúng ta sẽ thử ước lượng giá trị trung bình theo từng bước của ước lượng khoảng. (Chú ý đối số dữ liệu cha lần này là 66703 đô-la.)
Bước 1: Tạo dữ liệu mẫu bằng cách trích xuất ngẫu nhiên từ dữ liệu cha. Từ `df` ta trích xuất ngẫu nhiên dữ liệu để tạo dữ liệu mẫu. Dữ liệu cha `df` có 32526 dòng dữ liệu. Vì vậy ta sẽ lấy ngẫu nhiên 1000 dòng dữ liệu để tạo dữ liệu mẫu.
Để lấy dữ liệu ngẫu nhiên trong `DataFrame`, ta sử dụng `.sample(n)` . (Trong đó n là độ lớn của dữ liệu mẫu. Ta có thể chỉ định tỷ lệ bằng cách truyền cho `frac` giá trị từ $0 \sim 1$.)

```
n=1000
sample_df = df.sample(n=n)
```

Chúng ta thử nhìn phân bố dữ liệu mẫu xem sao nhé.

```
sns.displot(sample_df['Mean'])
```



Tôi nghĩ là nó khá giống với dữ liệu cha.

Bước 2: Tính công thức ước lượng từ dữ liệu mẫu.

Lần này đối tượng muốn ước lượng là giá trị trung bình của dữ liệu cha. DO đó đương nhiên công thức ướng lượng sẽ là giá trị trung bình của dữ liệu mẫu.

Hãy thử nhìn giá trị trung bình của dữ liệu mẫu, nó gần giống với dữ liệu cha.

```
np.mean(sample_df['Mean'])
```

Kết quả:

```
65102.071
```

Kết quả là 65102 đô-la. Trước đó ta đã biết giá trị trung bình của dữ liệu cha là 66703 đô-la. Có thể nói giá trị trung bình của dữ liệu mẫu khá gần với giá trị trung bình của dữ liệu cha.

Bước 3: Quyết định khoảng tin cậy.

Lần này ta sẽ tính toán với khoảng tin cậy là 95%.

Bước 4: Lần này ta sẽ xem xét phân bố mẫu của công thức ướng lượng.

Nếu phấn bố của dữ liệu cha là phân bố chuẩn thì bất kể n lớn hay không, phân bố mẫu của giá trị trung bình mẫu cũng sẽ là phân bố chuẩn.

Lần này phân bố của dữ liệu cha không phải là phân bố chuẩn.

Bổ sung

Thông thường, chúng ta không biết phân bố dữ liệu cha là phân bố như thế nào, nhưng cũng có thể dự đoán được ở một mức độ nào đó. Những gì trong thế giới tự nhiên, thường là phân bố chuẩn. Ví dụ, chiều cao nam giới trưởng thành, độ dày thân hoa, hoặc các sản phẩm lỗi ở một nhà máy đều có thể xem là phân bố chuẩn. Thế nhưng, lợi nhuận kinh doanh, thu nhập năm, dữ liệu liên quan tới tiền bạc thường có thể dự đoán đó không phải là dữ liệu có phân bố chuẩn. Lần này là dữ liệu thu nhập trung bình của các hộ gia đình của các vùng, cho nên không thể coi đó là phân bố chuẩn.

Vậy thì khi dữ liệu cha không phải là phân bố chuẩn, phân bố dữ liệu mẫu sẽ như thế nào? Theo định lý cực hạn trung tâm, n là độ lớn của dữ liệu mẫu, n càng lớn thì phân bố dữ liệu mẫu càng gần với phân bố chuẩn.
Vậy thì lần này với $n = 1000$ đã có thể nói là có đủ lớn hay chưa?
Tôi sẽ giải thích chi tiết hơn trong bài viết sau, nhưng bây giờ tôi có thể trả lời rằng **Có**.
Khi $n = 1000$, phân bố mẫu gần với phân bố chuẩn.
Giống như Bài học 24, chúng ta sử dụng phương thức `.interval()` của Module `stats` để ước lượng.

Phân bố xác suất lần này có giá trị trung bình là $\bar{x}$ và phân tán của phân bố chuẩn là $\frac{\sigma^2}{n}$. Với phân bố chuẩn có thể sử dụng `stats.norm` nhưng vì chúng ta không biết phân tán của dữ liệu cha σ^2, nên thay vào đó ta sử dụng phân tán bất thiên ${s'}^2$ như dưới đây. (Truyền khoảng tin cậy 0.95 cho `alpha`, truyền giá trị trung bình của phân bố $\bar{x}$ cho `loc`, truyền độ lệch chuẩn của phân bố $\frac{s'}{\sqrt{n}}$ cho `scale`).

```
from scipy import stats
sample_mean = np.mean(sample_df['Mean'])
sample_var = stats.tvar(sample_df['Mean'])
stats.norm.interval(alpha=0.95, loc=sample_mean, scale=np.sqrt(sample_var/n))
```

Kết quả:

```
(63302.127654258715, 66902.01434574129)
```

Phân tán bất thiên ${s'}^2$ có thể tính theo cách của Bài học 5, sử dụng `stats.tvar()` cũng được.
Kết quả trên có nghĩa là 95% xác suất mà giá trị trung bình của dữ liệu cha năm trong khoảng 63302 USD $\sim$ 66902 USD.
Trong thực tế giá trị trung bình của dữ liệu cha là là 66703 USD, nên có thể nói ước lượng trên là chuẩn xác.
Chúng ta không cần điều tra toàn bộ các khu vực của nước Mỹ, bằng cách lấy ngẫu nhiên một số vùng (cụ thể là 1000 vùng), từ đó đã có thể đưa ra ước lượng có mức chính xác nhất định.

25.4 Thực tế sử dụng phân bố t

Theo định lý cực hạn trung tâm, bất kể phân bố dữ liệu cha như thế nào, trong trường hợp độ lớn dữ liệu mẫu là lớn, phân bố mẫu của giá trị trung bình có thể biểu diễn là phân bố chuẩn.
Tuy nhiên trường hợp này là chúng ta đã biết phân tán của dữ liệu cha. Thông thường chúng ta không biết phân tán của dữ liệu cha, và ở ví dụ trên chúng ta đã thay thế bằng phân tán bất thiên của dữ liệu mẫu.
Tuy nhiên **vì sử dụng phân tán bất thiên (phương sai không chệch) cho nên phân bố sẽ khác một chút so với phân bố chuẩn**.(Trường hợp **kích thước dữ liệu mẫu lớn, quả nhiên có thể biểu diễn gần với phân bố chuẩn**. Vì vậy, trong bài

này chúng ta không bận tâm mà coi đó là phân bố chuẩn.)
Khi ta sử dụng phân tán bất thiên thay cho phân tán của dữ liệu cha, phân bố sẽ khác một chút so với phân bố chuẩn, ta gọi đó là **phân bố t**.

25.5 Tổng kết

Bài học lần này đã dài, tôi xin tóm tắt như dưới đây.
Trong bài học này chúng ta đã cùng nhau ước lượng khoảng cho giá trị trung bình.

- Để ước lượng giá trị trung bình dữ liệu cha, ta sử dụng giá trị trung bình dữ liệu mẫu làm công thức ước lượng. (Giá trị trung bình mẫu là công thức ước lượng bất thiên của giá trị trung bình của dữ liệu cha.)
- Dữ liệu cha không phải là phân bố chuẩn đi chăng nữa thì nếu kích thước dữ liệu mẫu đủ lớn, phân bố mẫu của giá trị trung bình mẫu cũng trở thành phân bố chuẩn với giá trị trung bình là μ, phân tán là $\frac{\sigma^2}{n}$. (Định lý cực hạn trung tâm)
- Độ lệch chuẩn của dữ liệu cha σ là giá trị chúng ta chưa biết, chúng ta sử dụng độ lệch chuẩn của dữ liệu mẫu s' thay cho nó.
- Vì điều này mà hình thù phân bố mẫu khác một chút so với phân bố chuẩn. Ta gọi đó là phân bố t (Sẽ giải thích chi tiết ở bài sau.). Kích thước dữ liệu mẫu càng lớn thì càng gần với phân bố chuẩn.
- Sử dụng `stats.norm.interval()` để ước lượng khoảng cho giá trị trung bình.

Ước lượng khoảng cho giá trị trung bình có ba điểm chính:

- Phân bố của dữ liệu cha là phân bố chuẩn hay không?
- Phân tán cảu dữ liệu cha đã biết hay chưa?
- Kích thước dữ liệu mẫu là bao nhiêu?

Khi phân bố mẫu trở thành phân bố chuẩn, nếu phân tán của dữ liệu cha chưa biết thì có thể sử dụng phân bố t để tính toán. Vì tôi giả định dữ liệu mẫu là lớn cho nên mặc định coi đây là phân bố chuẩn thay vì tiến hành với phân bố t.
Phân bố t là phân bố xác suất vô cùng quan trọng, trong bài sau tôi sẽ giải thích chi tiết.

Bài 26

Giải thích dễ hiểu về phân bố t

Trong Bài học 25 chúng ta đã ước lượng khoảng cho giá trị trung bình nhưng trong bài này tôi sẽ giải thích về phân bố t(student's t distribution) khi ước lượng khoảng cho giá trị trung bình.
Tôi nghĩ có nhiều người biết thuật ngữ "phân bố t" nhưng lảng tránh tìm hiểu về nó. Tuy nhiên phân bố t là phân bố xác suất vô cùng quan trọng trong lý luận của thống kê học.

26.1 Phân bố t là gì?

Chúng ta đã có kết luận rằng, phân bố t là phân bố xác suất của giá trị trung bình $\bar{x}$ của dữ liệu mẫu tuân theo khi mà ta thay độ lệch chuẩn của dữ liệu cha σ bằng độ lệch chuẩn dữ liệu mẫu s', và sau đó thực hiện chuẩn hóa.
Trong Bài học 25, ta có, phân bố mẫu của giá trị trung bình là phân bố chuẩn với giá trị trung bình μ và phân tán $\frac{\sigma^2}{n}$. (Trong đó μ là giá trị trung bình của dữ liệu cha, *sigma* là độ lệch chuẩn của dữ liệu cha).
Trong Bài học 9, ta đã biết công thức chuẩn hóa:

$$z = \frac{\bar{x} - \mu}{\frac{\sigma}{\sqrt{n}}}$$

Vì z tuân theo phân bố chuẩn tắc (tham khảo Bài học 8)

$$-1.96 < \frac{\bar{x} - \mu}{\frac{\sigma}{\sqrt{n}}} < 1.96$$

ta sử dụng nó để ước lượng khoảng.
Tuy nhiên trong công thức trên ta sử dụng độ lệch chuẩn của dữ liệu tổng thể (dữ liệu cha) σ.
Thông thường đây là giá trị chúng ta không biết. (Chúng ta không biết giá trị trung bình μ của dữ liệu tổng thể nên mới đi ước lượng nó, vì vậy chẳng có lý do gì mà chúng ta lại biết độ lệch chuẩn của dữ liệu tổng thể σ.)
Vì lý do trên, chúng ta đã sử dụng độ lệch chuẩn dữ liệu mẫu s' thay cho độ lệch chuẩn

dữ liệu tổng thể σ. Hãy chú ý độ lệch chuẩn dữ liệu mẫu s' là căn bậc hai của phân tán bất thiên s'^2:

$$s'^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

Hãy chú ý không phải chia cho n mà là $n-1$. Lý do chia cho $n-1$ đã được giải thích trong Bài học 7.
Tóm lại khi ước lượng khoảng cho giá trị trung bình, chúng ta xem xét phân bố xác suất theo $\frac{\bar{x}-\mu}{\frac{s'}{\sqrt{n}}}$ chứ không phải $\frac{\bar{x}-\mu}{\frac{\sigma}{\sqrt{n}}}$. Phân bố này gọi là phân bố t.

$$t = \frac{\bar{x}-\mu}{\frac{s'}{\sqrt{n}}}$$

Nói cách khác, phân bố t là một phương án thay thế cho phân phối chuẩn tắc.

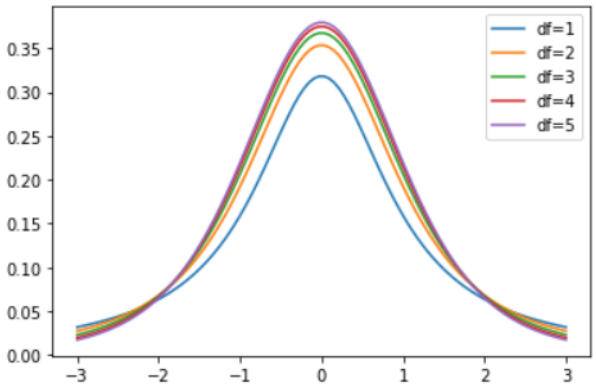
26.2 Thử nhìn vào phân bố t

Nhìn vào công thức tôi thấy mệt quá!

Vâng, vậy thì bây giờ chúng ta sẽ thử nhìn phân bố t trong thực tế coi xem ó là phân bố như thế nào.

Phân bố chuẩn tắc là phân bố chuẩn có giá trị trung bình là 0 và phân tán là 1. Đó là phân bố xác suất được xác định là không có tham số (đối số).
Phân bố t sẽ thay đổi hình thù tùy thuộc vào đối số $n-1$. Chúng ta đã biết phân tán bất thiên chia cho $n-1$. Và $n-1$ là đối số duy nhất của phân bố t.
(Người ta gọi $n-1$ là bậc tự do :degree of freedom.)
Nào, bây giờ chúng ta sẽ sử dụng `stats.t` của Module `stats` để xem phân bố t trong thực tế như thế nào. (Về `.pdf()` chúng ta cũng đã sử dụng trong Bài học 21 và Bài học 22, `.pdf` là từ viết tắt của probability density function nó sẽ trả về kết quả của hàm mật độ xác suất với thông số đầu vào mà ta truyền cho nó.).

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
%matplotlib inline


x = np.linspace(-3, 3, 100)
for df in range(1, 6):
t = stats.t.pdf(x, df)
plt.plot(x, t, label=f"df={df}")
plt.legend()
```

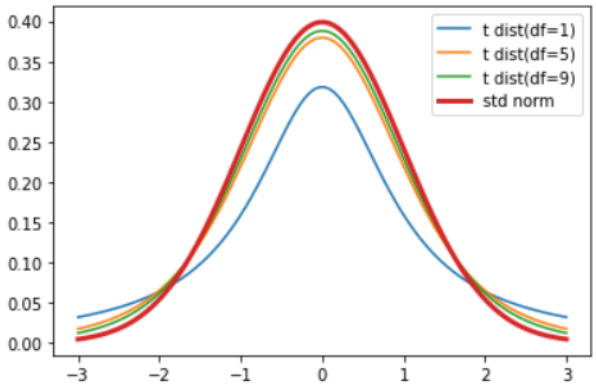


Ở đây đối số của `stats.t.pdf()` là `x` và `df` (bậc tự do). Lần này để xem hình thù biến đổi phụ thuộc vào bậc tự do ra sao, ta đã vẽ biểu đồ với df $= 1 \sim 5$.

Hình thù gần giống với phân bố chuẩn, có dạng hình cái chuông. Có thể thấy khi **bậc tự do nhỏ thì đáy lớn, nhưng xác suất tại trung tâm thì nhỏ. Nngược lại khi bậc tự do lớn thì xác suất tại trung tâm cao hơn nhưng đáy thì thu hẹp lại**.

Chúng ta hãy thử so sánh với phân bố chuẩn. Với phân bố chuẩn có thể sử dụng `stats.norm.pdf(x, loc=0, scale=1)` .

```
x = np.linspace(-3, 3, 100)
z = stats.norm.pdf(x, loc=0, scale=1)
for df in range(1, 10, 4):
t = stats.t.pdf(x, df)
plt.plot(x, t, label=f"t dist(df={df})")
plt.plot(x, z, label='std norm', linewidth=3)
plt.legend()
```



Đường màu đỏ dày là phân bố chuẩn. Chúng ta nhận thấy **khi bậc tự do tăng lên thì nó gần với phân bố chuẩn hơn**.

Nói tóm lại điều đó có nghĩa là nếu dữ liệu mẫu càng lớn thì không sử dụng phân bố t mà sử dụng phân bố chuẩn tắc cũng không sao.

Khi bậc tự do nhỏ thì đáy đồ thị của phân bố t dài hơn nhưng như vậy thì khoảng giá trị 95% cũng trở nên rộng hơn.

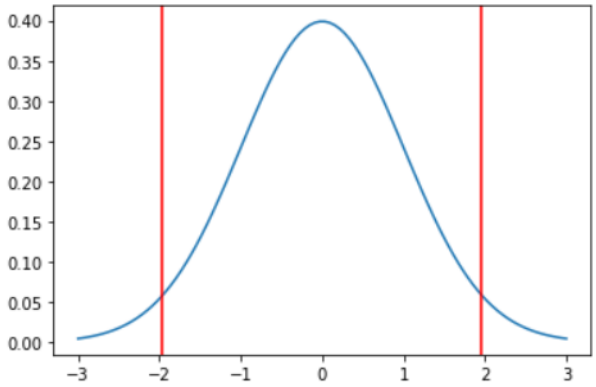
Chúng ta hãy thử xem xét khoảng tin cậy 95% của phân bố chuẩn tắc với khoảng tin cậy 95% của phân bố t với bậc tự do là 1.

Ở đây có thể dùng `.interval()` (về `.interval()` đã nói ở Bài học 25 và Bài học 24).

Phân bố chuẩn tắc như mọi khi, khoảng tin cậy 95% chính là từ $-1.96 \sim 1.96$

```
x = np.linspace(-3, 3, 100)
left, right = stats.norm.interval(0.95, loc=0, scale=1)
z = stats.norm.pdf(x, loc=0, scale=1)
plt.plot(x, z)
plt.axvline(left, c='r')
plt.axvline(right, c='r')
print(left, right)
```

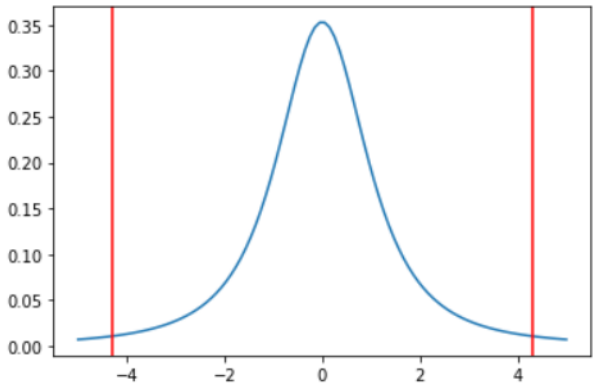
```
-1.959963984540054 1.959963984540054
```



So sánh với phân bố t với bậc tự do là 2:

```
x = np.linspace(-5, 5, 100)
left, right = stats.t.interval(0.95, df=2)
t = stats.t.pdf(x, df=2)
plt.plot(x, t)
plt.axvline(left, c='r')
plt.axvline(right, c='r')
print(left, right)
```

```
-4.302652729911275 4.302652729911275
```

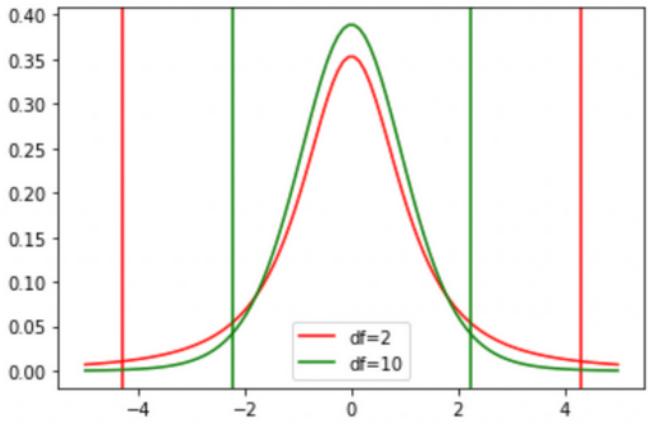


Phạm vi 95% đã trở nên rộng hơn.

Khi bậc tự do là 2, tức là $n = 3$, đây là trường hợp dữ liệu mẫu có kích thước rất nhỏ, vì vậy **khoảng tin cậy** 95% **có độ rộng rất lớn**. Chú ý rằng, khi bậc tự do lớn thì đáy đồ thị nhỏ, xác suất tại trung tâm tăng lên, diện tích bao giờ cũng là 1 do đó khi chiều cao tăng lên thì đáy sẽ nhỏ đi, có thể thấy khi đó khoảng 95% cũng sẽ trở nên hẹp hơn.

Đến đây, tôi nghĩ chúng ta đều dễ hình dung được rằng, khi dữ liệu mẫu có kích thước

càng lớn thì khoảng ước lượng càng hẹp.
Ở hình vẽ dưới đây là hình vẽ so sánh khoảng 95% của phân bố t với bậc tự do 2 là đường màu đỏ, bậc tự do 10 là màu xanh lục.



Nói tóm lại điều này có nghĩa là **khi độ lệch chuẩn mẫu s' được sử dụng thay thế cho độ lệch chuẩn tổng thể (độ lệch chuẩn của dữ liệu cha) σ, khi ước lượng giá trị trung bình trong phấn bố mẫu của $\bar{x}$ thì cần phải lấy một khoảng ước lượng rộng**.
Tuy nhiên nếu kích thước dữ liệu mẫu n lớn thì độ lệch chuẩn mẫu s' gần bằng với độ lệch chuẩn tổng thể σ. Do đó mà sự ảnh hưởng (sai số) cũng ít đi khi thực hiện ước lượng.

26.3 Sử dụng phân bố t để ước lượng khoảng cho giá trị trung bình

Giống như bài trước, nhưng lần này ta sẽ sử dụng phân bố t thay cho phân bố chuẩn để ước lượng khoảng cho giá trị trung bình.
Các bạn có thể download data set US Household Income Statistics.
Đầu tiên chúng ta sẽ đọc dữ liệu file csv và thử nhìn giá trị trung bình của dữ liệu tổng thể (dữ liệu cha).

```
import numpy as np
import pandas as pd

df = pd.read_csv('kaggle_income.csv', encoding="ISO-8859-1")
print(df['Mean'].mean())
```

```
66703.98604193568
```

Giá trị trung bình ước chừng khoảng 66704 đô-la. (Lần này tôi không sử dụng `np.mean()` mà đã dùng `.mean()` của `Series` để tính giá trị trung bình.)
Tiếp theo chúng ta cần tạo dữ liệu mẫu. Lần này tôi cũng trích xuất ngẫu nhiên 1000 dòng dữ liệu từ dữ liệu cha và tính giá trị trung bình cho mẫu.

```
n=1000
sample_df = df.sample(n=n)
print(sample_df['Mean'].mean())
```

```
67102.57
```

Ước chừng khoảng 67103 đô-la. Chú ý mẫu được lấy ngẫu nhiên cho nên giá trị này không phải là cố định, nó có thể khác nhau sau mỗi lần chạy code trên. Nhưng chúng ta cũng có thể thấy rằng giá trị trung bình này khác với giá trị trung bình của dữ liệu tổng thể.
Nào, bây giờ hãy thử sử dụng phân bố t và tính khoảng tin cậy 95%. Vẫn như cách làm trước đây, ta sẽ sử dụng `stats.t.interval()`. Ta truyền đối số cho phương thức này là `alpha=0.95`, loc $= \bar{x}$, scale $= \frac{s'}{\sqrt{n}}$, `df=n-1`. (So với lần trước, lần này có thêm đối số **df**.)

```
from scipy import stats
sample_mean = sample_df['Mean'].mean()
sample_std = stats.tstd(sample_df['Mean'])
stats.t.interval(alpha=0.95, loc=sample_mean, scale=sample_std/np.sqrt(n), df=n-1)
```

```
(65189.2895147609, 69015.8504852391)
```

Bằng phân bố t ta đã ước lượng giá trị trung bình của dữ liệu tổng thể với khoảng tin cậy 95% là 65189 USD $\sim$ 69106 USD, giá trị trung bình của dữ liệu tổng thể là 67103 USD nên có thể nói ước lượng này là chính xác. (Lần này ta không sử dụng phân tán bất thiên s'^2 mà sử dụng `stats.std()` để tính độ lệch chuẩn s'. Chú ý, chúng ta cần sử dụng độ lệch chuẩn là căn bậc hai của phân tán bất thiên, cái mà được chia cho $n-1$. Nếu chúng ta sử dụng `np.std` thì không phải chia cho $n-1$ mà là chia cho n khi tính toán độ lệch chuẩn, do đó nếu muốn sử dụng `np.std()` thì hãy dùng `np.std(ddof=1)`.).
Nào, vậy nếu không sử dụng phân bố t, lần trước chúng ta đã sử dụng phân bố chuẩn, và kết quả khoảng ước lượng như thế nào nhỉ? Ước lượng khoảng trong phân bố chuẩn chúng ta dùng:

```
stats.norm.interval(alpha=0.95, loc=sample_mean, scale=sample_std/np.sqrt(n))
```

```
(65191.60755148405, 69013.53244851597)
```

Tôi nghĩ là khoảng ước lượng nhỏ hơn so với khoảng ước lượng của phân bố t. Ở đây, bậc tự do là 999 ($= 1000 - 1$), độ rộng của đồ thị phân bố rộng hơn so với phân bố chuẩn, vì vậy mà khoảng 95% của phân bố t cũng sẽ rộng hơn so với phân bố chuẩn là điều có thể hiểu được.

26.4 Sử dụng phân bố t hay sử dụng phân bố chuẩn?

> Vậy thì kết quả ước lượng của phân bố t với kết quả ước lượng bằng phân bố chuẩn thì cái nào mới là đúng?

> Tất nhiên không sử dụng gần đúng bằng phân bố chuẩn mà sử dụng trực tiếp phân bố t thì là đúng.

Vì vậy các phần mềm thống kê thường sử dụng phân bố t để ước lượng.
Tuy nhiên trong thực tế có rất nhiều người sử dụng phân bố chuẩn để ước lượng. Khi dữ

liệu mẫu có kích thước lớn, phân bố t gần với phân bố chuẩn, nên sử dụng phân bố chuẩn để ước lượng thì đơn giản và tiện lợi.
Đây chỉ là ý kiến cá nhân của riêng tôi, ở một mức độ nào đó nếu $n > 30$ thì có thể sử dụng phân bố chuẩn để ước lượng cũng không sao đâu. (Tất nhiên nếu bạn muốn hoạt động giống như Python, thì có thể sử dụng phân bố t để ước lượng. Tôi nghĩ đây không phải là vấn đề, nên chúng ta không cần bận tâm.)

26.5 Tổng kết

Trong bài này tôi đã giải thích về phân bố t cũng như đã cùng các bạn sử dụng phân bố t để ước lượng giá trị trung bình.
Phân bố t là lý luận rất quan trọng trong thống kê học, không chỉ trong Machine Learning, cho nên các bạn hãy ghi nhớ những hình ảnh về nó.

- Phân bố t là phân bố xác suất mà giá trị trung bình $\bar{x}$ của dữ liệu mẫu tuân theo, khi độ lệch chuẩn mẫu s' được sử dụng thay thế độ lệch chuẩn tổng thể σ.
- Phân bố t có đối số duy nhất là bậc tự do $n - 1$.
- So với phân bố chuẩn tắc, đáy của phân bố t rộng hơn, tức là cùng khoảng chả hạn là 95% thì xác suất tại khoảng này của phân bố t thấp hơn (đỉnh đồ thị thấp hơn) so với phân bố chuẩn. Nói cách khác, cùng khoảng tin cậy thì khoảng ước lượng khi sử dụng phân bố t sẽ rộng hơn.
- Khi ước lượng khoảng sử dụng phân bố t ta truyền cho `stats.interval()` các đối số `alpha` (hệ số tin cậy), `loc`(giá trị trung bình của dữ liệu mẫu), `scale`(độ lệch chuẩn của giá trị trung bình mẫu), `df` (bậc tự do) để tính toán.
- Nếu kích thước dữ liệu mẫu lớn thì phân bố t hay phân bố chuẩn đều cho kết quả tương đồng, không có nhiều khác biệt.

Từ bài học tới, chúng ta sẽ đi vào phần chính của thống kê học, đó là "Kiểm Định".

Bài 27

Giải thích dễ hiểu về cách kiểm định giả thuyết

Trong thống kê suy luận, chúng ta đã nói với nhau về "ước lượng", và từ bài này chúng ta sẽ nói với nhau về **"kiểm định"**. Khái niệm về ước lượng và kiểm định các bạn có thể xem lại trong Bài học 23.
"Kiểm định" là động lực số một để nghiên cứu thống kê và là lĩnh vực đóng vai trò quan trọng nhất trong lý thuyết thống kê.
Có thể nói cho tới bây giờ, tất cả những gì chúng ta đã học là bước chuẩn bị để học về "Kiểm Định". (Điều đó có phải là quá lời hay không? Nhưng nó thực sự là một lĩnh vực rất quan trọng.)
Vậy thì "Kiểm Định" có thể làm được những công việc như thế nào? Tôi sẽ đi vào giải thích chi tiết.(Trong bài học này sẽ chỉ toàn câu chữ mang tính lý luận, từ buổi sau sẽ sử dụng Python để thực hiện kiểm định thực tế.)

27.1 Kiểm định là gì?

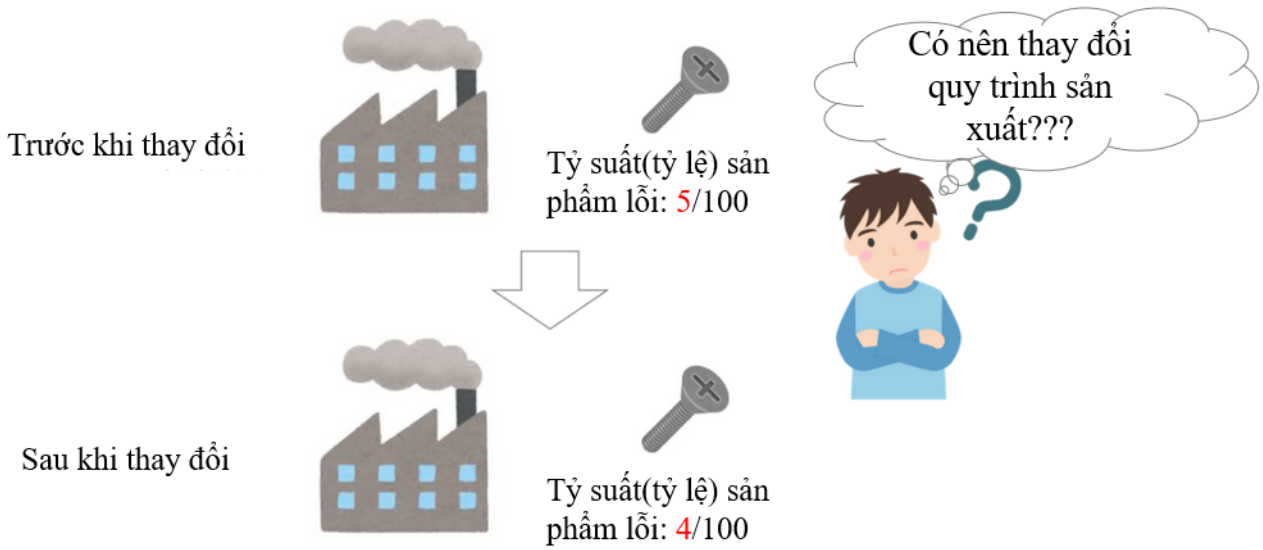
Nói một cách đơn giản, đó là công việc mà đối với một sự vật sự việc giả định, thông qua quan sát thực tế dữ liệu mẫu, ta điều tra xem giả định đó có mâu thuẫn gì hay không.

Hoàn toàn không đơn giản chút nào, tôi vẫn chưa hiểu gì!

Vậy thì hãy cùng nhau xem ví dụ cụ thể để hiểu hơn nhé.

Ví dụ trong một nhà máy sản xuất ra sản phẩm, có một hằng số xác suất mà các sản phẩm lỗi được tạo ra.
Người ta thay đổi quá trình chế tạo tại nhà máy để xác suất tạo ra sản phẩm lỗi giảm xuống.
Công việc của "Kiểm Định" đó là đánh giá cái thay đổi nói trên có hiệu quả hay không, có ích hay không.
Ta sẽ lấy mẫu sản phẩm trước và sau khi thay đổi quy trình sản xuất. Giả sử như sau khi thay đổi quy trình sản xuất, xác suất tạo ra sản phẩm lỗi giảm, khi đó có thể nói rằng

sự thay đổi này là đúng đắn, chính xác. Khi đó có thể tiến hành thay đổi quy trình sản xuất tại nhà máy một cách chính thức.
Giả sử rằng chúng ta lấy lần lượt 100 sản phẩm (lấy mẫu), trước khi thực hiện thay đổi quy trình sản xuất, trong 100 sản phẩm có 5 sản phẩm lỗi, sau khi thay đổi quy trình sản xuất, số sản phẩm lỗi là 4.
Quả nhiên là chúng ta đã nhìn thấy chất lượng đã được cải thiện khi xem xét các sản hẩm lấy mẫu. Nhìn vào kết quả này, liệu rằng chúng ta có thể nghĩ rằng có thể thay đổi quy trình sản xuất hay không?



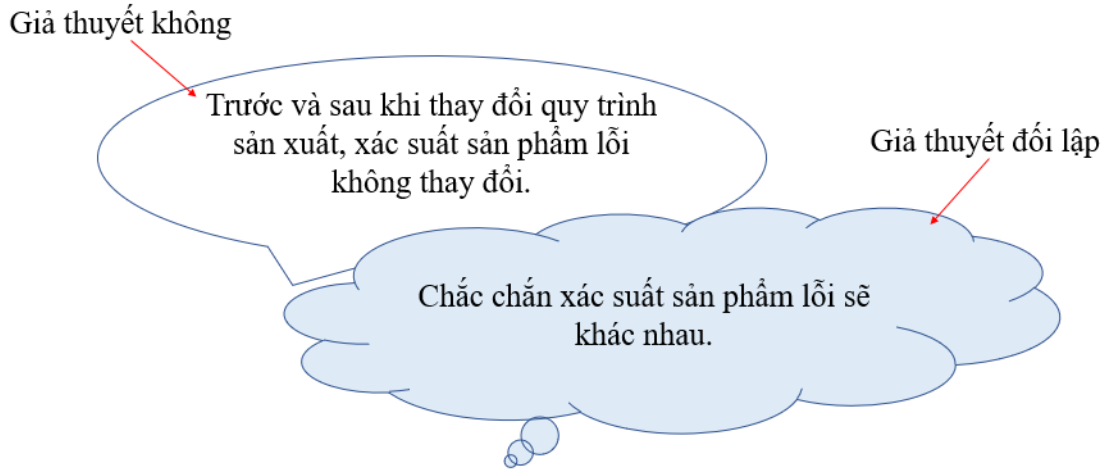
Vậy thì cái thay đổi này nếu sau khi thay đổi, số sản phẩm lỗi là 3 thì sao? Nếu là 2 thì sao? Nếu là 2 thì có vẻ thay đổi đổi quy trình cũng tốt nhỉ!
Công việc phán đoán như thế là việc cần thiết của "Kiểm Định". Nói tóm lại kiểm định đóng một vai trò rất quan trọng trong việc ra quyết định.
Câu hỏi đặt ra là chúng ta sẽ sử dụng kiểm định như thế nào?
Đầu tiên, chúng ta sẽ giả định rằng, trước và sau khi thay đổi quy trình sản xuất, thì xác suất sản phẩm lỗi không thay đổi.
Dựa trên giả định này, ta tìm trong dữ liệu mẫu xem tỷ suất (tỷ lệ) sản phẩm lỗi, nếu tỷ suất sẽ khác nhau, trong trường hợp đó ta có thể nói giả định là sai. Trong trường hợp như vậy, ta có thể nói rằng, trước và sau khi thay đổi quy trình thì xác suất sản phẩm lỗi lác khác nhau. Từ đó ta có thể phán đoán có nên thay đổi quy trình sản xuất hay không.
Giả định như vậy gọi là **giả thuyết (hypothesis)**, việc sử dụng giả thuyết trong kiểm định, ta gọi công việc kiểm định này là **kiểm định giả thuyết thống kê**.

27.2 Các bước kiểm định giả thuyết trong thống kê (Giả thuyết đối lập và giả thuyết không)

Kiểm định giả thuyết thống kê có trình tự cơ bản như sau:

- Thành lập giả thuyết.
- Dựa trên giả thuyết và quan sát dữ liệu mẫu (thống kê mẫu).
- Từ kết quả quan sát dữ liệu mẫu, đưa ra phán đoán giả thuyết là đúng hay không.

Điểm nhấn trong kiểm định giả thuyết thống kê là, **giả thuyết được thành lập ban đầu bị phủ định**.
Nói tóm lại, chúng ta vừa nghĩ rằng "tôi e rằng giả thuyết này sẽ sai" vừa thành lập giả thuyết. Trong khi quan sát dữ liệu mẫu cố gắng thu thập chứng cớ làm sao để có thể nói rằng giả thuyết này chẳng phải là sai hay sao?
Ví dụ, trong ví dụ ban nãy, ta thành lập giả thuyết rằng "trước và sau khi thay đổi quy trình, xác suất sản phẩm lỗi là không thay đổi", thế nhưng trong thâm tâm ta nghĩ rằng "chắc chắn không có chuyện xác suất trước và sau khi thay đổi sẽ giống nhau".
Ngay từ đầu thành lập giả thuyết nhưng sau đó lại phải định giả thuyết, ta gọi đây là **giả thuyết không (null hypothesis)**. Các bạn hãy nhớ thuật ngữ quan trọng này. Cắt nghĩa chính xác phải nói là giả tuyết quay trở về không. Tức là ban đầu thành lập giả thuyết, nhưng sau đó lại quay trở lại phủ định, không công nhận giả thuyết là đúng.
Ngược lại, trường hợp "giả thuyết không" bị phủ định, chúng ta thành lập **"giả thuyết nghịch"** hay còn gọi là **"giả thuyết đối lập" (alternative hypothesis)**.
Ví dụ, "giả thuyết không" được thành lập với nội dung "trước và sau khi thay đổi quy trình sản xuất, xác suất sản phẩm lỗi là không thay đổi". Trong khi quan sát dữ liệu mẫu, nếu kết quả dẫn tới phủ định giả thuyết này, khi đó ta thành lập giả thuyết mới với nội dung "trước và sau khi thay đổi quy trình sản xuất, xác suất sản phẩm lỗi là khác nhau". Đây là "giả thuyết đối lập". Tóm lại **giả thuyết đối lập là giả thuyết mà trong thâm tâm ta nghĩ rằng nó đúng**.



Bổ sung

Một số sách Việt Nam gọi "giả thuyết không" là "giả thuyết rỗng".

Vì vậy, trình tự ở trên ta sẽ thay đổi một chút trong đó sử dụng các thuật ngữ chuyên môn.

- Thành lập *giả thuyết không.*
- Dựa trên *giả thuyết không*, và quan sát dữ liệu mẫu (thống kê mẫu).
- Từ kết quả quan sát dữ liệu mẫu, đưa ra phán đoán *giả thuyết không* là đúng hay sai, trường hợp phủ định *giả thuyết không* thì thành lập *giả thuyết đối lập.*

Tại sao phải phiền hà rắc rối như vậy làm gì? Tại sao ngay từ đầu không thành lập giả thuyết đối lập và sau đó có thể nói giả thuyết này là đúng!

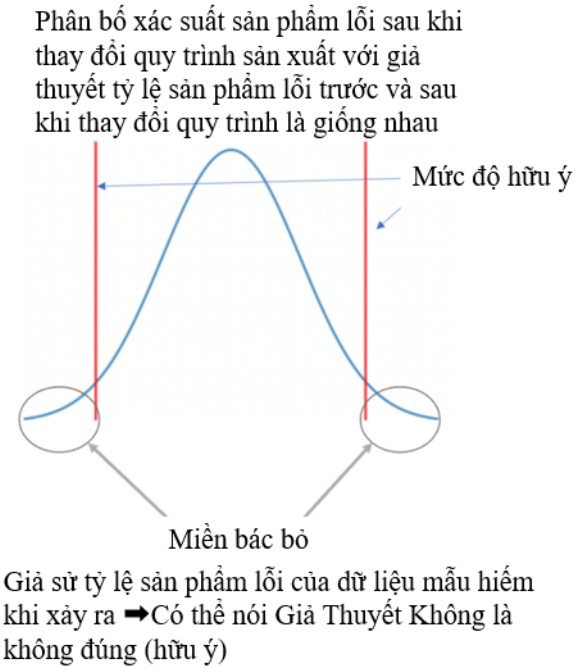
Tôi nghĩ rất nhiều người nghĩ như vậy, nhưng **ngay từ đầu thành lập giả thuyết đối lập và khẳng định giả thuyết đó là đúng là việc rất khó**.

Quay trở lại ví dụ trong bài này, chúng ta rất mốn nói rằng "trước và sau khi thay đổi quy trình sản xuất thì xác suất sản phẩm lỗi là khác nhau", thế nhưng nếu như thế thì sẽ có rất nhiều trường hợp.
Chẳng hạn "xác suất sản phẩm lỗi của trước và sau khi thay đổi quy trình sản xuất thì khác nhau 1%" hoặc "xác suất sản phẩm lỗi của trước và sau khi thay đổi quy trình sản xuất thì khác nhau 1.3%".
Nói chung là sẽ có vô số các trường hợp tồn tại.
Chính vì vậy nếu chỉ phủ định một giả thuyết "trước và sau khi thay đổi quy trình sản xuất thì xác suất sản phẩm lỗi là giống nhau" thì dễ dàng hơn.

27.3 Miền bác bỏ và miền chấp nhận

Vậy thì làm thế nào để phủ định giả thuyết không?
Giả Thuyết Không được thành lập, sau đó dựa vào dữ liệu mẫu tiến hành tính toán thống kê mẫu, tiến hành xác nhận giá trị thống kê mẫu đó, **giá trị thống kê mẫu đó nếu mà khó có thể nói rằng Giả Thuyết Không là đúng (tức là Giả Thuyết Không mặc dù đúng nhưng từ giá trị thống kê mẫu thấy rằng xác suất xảy ra là rất thấp), giả sử như giá trị thống kê mẫu rơi vào cái miền như vậy** thì có thể phủ định Giả Thuyết Không.
Miền như vậy gọi là **miền bác bỏ (regection region)**, ngược lại ta có **miền chấp nhận (acceptance region)**. Nếu thống kê mẫu cho giá trị rơi vào miền chấp nhận, tức là ta không thể phủ định Giả Thuyết Không.
Như vậy phủ định Giả Thuyết Không, ta gọi là Bác Bỏ.
Vậy thì làm thế nào để quyết định đường ranh giới giữa miền bác bỏ và miền chấp nhận?
Khi tính toán giá trị thống kê mẫu, hãy xem xét khả năng Giả Thuyết Không tồn tại như thế nào, xác suất xảy ra bao nhiêu thì chấp nhận được.
Thông thường người ta lựa chọn đường ranh giới là 1% hay 5%. Nói tóm lại **nếu một giá trị đạt được với xác suất là** 1% **hay dưới** 5% **thì Giả Thuyết Không bị bác bỏ**.

Nói cách khác, nếu giá trị thống kê mẫu đi vào miền bác bỏ mà không phải là tình cờ, thì có thể bị bác bỏ ngẫu nhiên.
Tương tự như vậy, có thể bác bỏ Giả Thuyết Không một cách không ngẫu nhiên, mà là có chứa ý nghĩa (hữu ý). Đường ranh rới này ta nói là mức độ hữu ý, được thể hiện bằng α.
Trường hợp đã thiết định mức độ hữu ý (có ý nghĩa) cho 1% hay 5%, khi đó Giả Thuyết Không thỉnh thoảng đúng với xác suất 1% hay 5%, như vậy dù nó có đi vào miền bác bỏ, nhưng tùy thuộc vào nguyên nhân hữu ý mà chúng ta sẽ bác bỏ, chúng ta nói điều này là **hữu ý (significant)**.



Ở chỗ này có khá nhiều từ chuyên môn nên các bạn cảm thấy khó hiểu thì cũng không sao đâu, sau này chúng ta sẽ tiến hành kiểm định bằng ví dụ cụ thể, chắc chắn các bạn sẽ hiểu.

27.4 Kiểm định cái gì?

Các bước kiểm định tôi đã hiểu nhưng rút cuộc là kiểm định cái gì nhỉ?

Kiểm định có rất nhiều loại nhưng trong khuôn khổ khóa học này tôi sẽ chỉ đi vào những thứ nổi tiếng, được biết tới rộng rãi.

27.4.1 Kiểm định sự sai khác của tỷ suất

Ví dụ ban nãy thuộc vào kiểu này. Trong một nhà máy tiến hành cải tiến quy trình sản xuất, tiền hành điều tra thông qua kiểm định xem sự sai khác của tỷ lệ(tỷ suất) sản

phẩm lỗi trước và sau khi cải tiến quy trình sản xuất.
Một ví dụ khác:

- Các biện pháp tiếp thị có làm thay đổi tỷ lệ lượt truy cập website công ty của quý bạn thông qua đường link được gửi từ mail hay không?
- Có sự khác biệt nào giữa tỷ lệ kết hôn của nam giới độ tuổi 30 của Việt Nam so với Hoa Kỳ hay không?

Nói chung có thể suy nghĩ ra rất nhiều ví dụ thuộc kiểu này.

27.4.2 Kiểm định liên quan

Mối tương quan của các biến phân loại được gọi là sự liên quan (liên kết) (association) (Về mối tương quan đã được giải thích trong Bài học 11).
Ví dụ, mọi người đang học Python thì có đang học R không?
Cho dù bạn đang học Python hay không thì đó cũng là một biến phân loại nhị phân, Có(1) hoặc Không(0). Tương tự cho dù bạn đang học R hay không thì đó cũng là một biến phân loại nhị phân.
Trường hợp biến phân loại thì không thể tính toán bằng Tương Quan được nói đến trong Bài học 11. Không phải là Tương Quan, ta gọi là Liên Quan, có phương pháp để tính toán Liên Quan. (Ta sẽ đi sâu vào nó trong các bài học trong tương lai.)
Chúng ta có thể tìm Có hay Không của Liên Quan bằng Kiểm Định. Đó là **kiểu Kiểm Định thường được sử dụng ngay cả trong Kiểm Định Giả Thuyết**. Có rất nhiều trường hợp đại lượng thống kê mà chúng ta sử dụng lấy đại lượng thống kê chi bình phương χ^2 làm cơ sở nên nó được gọi là **kiểm định chi bình phương**.

27.4.3 Kiểm định sự sai khác của giá trị trung bình

Nếu như chúng ta kiểm định sự sai khác của tỷ suất (tỷ lệ) thì với giá trị trung bình cộng cũng có công việc tương tự.
Ví dụ:

- Sai số trung bình của sản phẩm của nhà máy A và nhà máy B có bằng nhau hay không?
- Trung bình thời gian học một ngày của học sinh tiểu học giữa Phú Thọ và Hà Nội có bằng nhau hay không?

Vâng, chúng ta có rất nhiều các ví dụ.
Trong kiểm định sự sai khác của giá trị trung bình, người ta sử dụng phân bố t, gọi là kiểm định t (Student's t-test). Đây cũng là kiểu kiểm định thường được sử dụng trong lĩnh vực kinh doanh và khoa học. (Các bạn có thể xem lại phân bố t trong Bài học 26.)
(Tôi sẽ giải thích chi tiết trong khóa học nhưng các bạn lưu ý rằng, kiểm định t giả định phân tán của mỗi nhóm là đúng, đây là điều kiện tiên quyết khi tiến hành kiểm định t.)
Vì vậy tùy vào từng trường hợp mà có thể cần kiểm định trước "Phân tán có đúng

không?".(Nếu phân tán khác nhau thì cần phải có phương pháp kiểm định nâng cao nhưng trong khóa học này tôi sẽ không đề cập đến.)

27.4.4 Kiểm định phân tán(phương sai)

Kiểm định phân tán(phương sai) của hai dữ liệu cha có khác nhau hay không?

Kiểm định phân tán được sử dụng khi nào?

Trong lý luận của thống kê học, **có nhiều trường hợp giả định độ phân tán của hai dữ liệu cha là bằng nhau**. Kiểm định t vừa khi nãy nhắc tới cũng là một trong số đó. Nó có thể không thường xuất hiện như kiểm định tỷ suất, kiểm định liên quan, kiểm định sự sai khác giá trị trung bình, nhưng nó là một trong những kiểu kiểm định cơ bản nên tôi xin được giải quyết nó trong khóa học này.

27.5 Tổng kết

Trong bài học này, tôi đã giải thích trình tự kiểm định giả thuyết thống kê cùng với các thuật ngữ, các kiểu kiểm định.

- Kiểm định được sử dụng để đưa ra phán đoán chính xác về mặt thống kê. Kiểm định là một trong những lĩnh vực **quan trọng nhất** trong thống kê học.
- Trong kiểm định giả thuyết thống kê, một giả thuyết được đưa ra, một thống kê mẫu được tính toán với giả thiết rằng giả thuyết đó đúng, và việc giả thuyết đó đúng hay không sẽ đuwọc đánh giá về mặt thống kê.
- Điều kiện tiên quyết là **giả định ban đầu sẽ được phủ định**. Ta gọi đó là *giả thuyết không*. Ngược lại, một giả thuyết được thành lập bằng cách phủ nhận *giả thuyết không*, ta gọi đó là giả thuyết đối lập.
- Nếu một thống kê được tính toán, và nó chỉ có thể xảy ra với xác suất 1% hoặc 5% (mức độ có ý nghĩa) theo *giả thuyết không*, khi đó *giả thuyết không* sẽ bị bác bỏ (có ý nghĩa-hữu ý).
- Có rất nhiều kiểu kieẻm định nhưng nổi tiếng phải nói tới kiểm định sự sai khác tỷ suất, kiểm định liên quan, kiểm định sự sai khác của giá trị trung bình, kiểm định phân tán.

Kiểm định giống như đỉnh núi trong thống kê học.
Cho đến bây giờ, có thể nói không quá lời rằng toàn bộ lý thuyết của thống kê học là vì "kiểm định giả thuyết thống kê".

Phần II

Thuật Ngữ

STT	Tiếng Nhật	Cách đọc	Tiếng Anh	Tiếng Việt
1	分散	ブンサン	Variance	Phân tán
2	分布	ブンプ	Distribution	Phân bố
3	範囲	ハンイ	Range	Phạm vi
4	四分位数	シブンイスウ	Quartile points	Vị trí phần tư
5	相関係数	ソウカンケイスウ	Correlation coefficient	Hệ số tương quan
6	四分位範囲	シブンイハンイ	Interquartile Range: IQR	Phạm vi phần tư
7	四分位偏差	シブンイヘンサ	Quartile Deviation: QD	Độ lệch phần tư
8	平均偏差	ヘイキンヘンサ	Mean Deviation	Độ lệch trung bình
9	平均絶対偏差	ヘイキンゼッタイヘンサ	Mean Absolute Deviation	Độ lệch trung bình tuyệt đối
10	標準偏差	ヒョウジュンヘンサ	Standard Deviation	Độ lệch chuẩn
11	不偏分散	フヘンブンサン	Unbiased Variance	Phân tán bất thiên
12	不偏	フヘン	Unbiased	Bất thiên
13	不偏推定量	フヘンスイテイリョウ	Unbiased Estimator	Ước lượng bất thiên
14	期待値	キタイチ	Expected Value	Giá trị kỳ vọng
15	統計的記述(記述統計)	トウケイテキキジュツトウケイ	Descriptive Statistics	Thống kê mô tả
16	統計的推論(推測統計)	トウケイテキスイロンスイソクトウケイ	Inferential Statistics	Thống kê suy luận
17	母集団	ボシュウダン	Population	Dữ liệu cha
18	標本	ヒョウホン	Sample	Dữ liệu tiêu bản, dữ liệu mẫu
19	算術平均	サンジュツヘイキン	Arithmetic Mean	Trung bình cộng

STT	Tiếng Nhật	Cách đọc	Tiếng Anh	Tiếng Việt
20	幾何平均	キカヘイキン	Geometric Mean	Trung bình hình học
21	調和平均	チョウワ平均	Harmonic Mean	Trung bình điều hòa
22	偏差	ヘンサ	Deviation	Thiên sai, độ lệch
23	中央値	チュウオウチ	Median	Trung vị
24	外れ値	ハズレチ	Outlier	Giá trị ngoại lệ
25	最頻値	サイヒンチ	Mode	Tối tần trị
26	平方根	ヘイホウコン	-	Căn bậc hai
27	感覚値	カンカクチ	-	Giá trị cảm quan
28	境目	サカイメ	-	Đường ranh giới
29	偏差値	ヘンサチ	-	Trị số lệch
30	共分散	キョウブンサン	Covariance	Phân tán chung, hiệp phương sai
31	共分散行列	キョウブンサンギョウレツ	Variance Covariance Matrix	Ma trận phân tán chung, ma trận hiệp phương sai
32	分散共分散行列	ブンサンキョウブンサンギョウレツ	↑	↑
33	不偏共分散	フヘンキョウブンサン	Unbiased Covariance	Phân tán bất thiên chung
34	相関係数	ソウカンケイスウ	Correlation Coefficient	Hệ số tương quan
35	相関行列	ソウカンギョウレツ	Correlation Matrix	Ma trận tương quan
36	因果関係	インガカンケイ	-	Quan hệ nhân quả

STT	Tiếng Nhật	Cách đọc	Tiếng Anh	Tiếng Việt
37	回帰	カイキ	Regression	Hồi quy
38	密接	ミッセツ	-	Gần gũi
39	回帰直線	カイキチョクセン	Regression Line	Đường hồi quy
40	回帰直線を線形回帰	-	Linear Regression	Hồi quy tuyến tính
41	回帰変数	カイキヘンスウ	Regressor	Biến số hồi quy
42	被回帰変数	ヒカイキヘンスウ	Regressand	Biến số bị hồi quy
43	独立変数	ドクリツヘンスウ	Independent Variable	Biến số độc lập
44	従属変数	ジュウゾクヘンスウ	Dependent Variable	Biến số phụ thuộc
45	回帰分析	カイキブンセキ	Regression Analysis	Phân tích hồi quy
46	最小二乗法	-	Least Squares Method	Phương pháp bình phương tối thiểu
47	残差	-	Residual Error	Chênh lệch sai sót
48	回帰係数	カイキケイスウ	Regression Coefficient	Hệ số hồi quy
49	決定係数	ケッテイケイスウ	Coefficient Of Determination	Hệ số quyết định
50	確率変数	カクリツヘンスウ	Random Variable	Biến số xác suất
51	確率分布	カクリツブンプ	Probability Distribution	Phân bố xác suất
52	一様分布	イチヨウブンプ	Uniform Distribution	Phân bố đồng đều
53	確率密度関数	カクリツミツドカンスウ	Probability Density Function	Hàm mật độ xác suất

STT	Tiếng Nhật	Cách đọc	Tiếng Anh	Tiếng Việt
54	標本分布	ヒョウホンブンプ	Sampling Distribution	Phân bố mẫu
55	二項分布	ニコウブンプ	Binomial Distribution	Phân bố nhị thức
56	確率質量関数	カクリツシツリョウカンスウ	Probability Mass Function	Hàm chất lượng xác suất
57	幾何分布	キカブンプ	Geometric Distribution	Phân bố hình học
58	指数分布	シスウブンプ	Exponential Distribution	Phân bố mũ
59	標準正規分布	-	Standard Normal Distribution	Phân bố chuẩn tắc
60	正規分布	-	Normal Distribution	Phân bố chuẩn
61	無作為抽出	ムサクイチュウシュツ	Random Sampling	Lấy mẫu ngẫu nhiên
62	推定	スイテイ	Estimation	Ước lượng
63	検定	ケンテイ	Statistical Test	Kiểm định
64	推定量	スイテイリョウ	estimator	Công thức ước lượng
65	母数	ボスウ	Parameter	Đối số dữ liệu cha
66	想定	ソウテイ	Hypothesis	Giả thiết
67	仮説	カセツ	↑	↑
68	点推定	テンスイテイ	Point Estimation	Ước lượng điểm
69	区間推定	クカンスイテイ	Interval Estimation	Ước lượng khoảng
70	比率	ヒリツ	Ratio	Tỷ suất
71	信頼区間	シンライクカン	Confidence Interval	Khoảng tin cậy

STT	Tiếng Nhật	Cách đọc	Tiếng Anh	Tiếng Việt
72	中心極限定理	チュウシンキョクゲンテイリ	Central Limit Theorem	Định lý cực hạn trung tâm
73	-	-	Sampling Distributions	Phân bố mẫu
74	-	-	Population Distributions	Phân bố tổng thể
75	t分布	-	Student's t Distribution	Phân bố t
76	自由度	-	Degree Of Freedom	Bậc tự do
77	帰無仮説	キムカセツ	Null Hypothesis	Giả thuyết không
78	仮説	-	Hypothesis	Giả thuyết
79	標本統計量	-	Sample Statistic	Thống kê mẫu
80	対立仮説	-	Alternative Hypothesis	Giả thuyết đối lập
81	棄却域	キキャクイキ	Regection Region	Miền bác bỏ
82	採択域	サイタクイキ	Acceptance Region	Miền chấp nhận
83	有意	ユウイ	Significant	Hữu ý
84	連関	レンカン	Association	Liên quan
85	カイ二乗検定	-	Chi-squared test	Kiểm định chi bình phương
86	t検定	-	Student's t-test	Kiểm định t