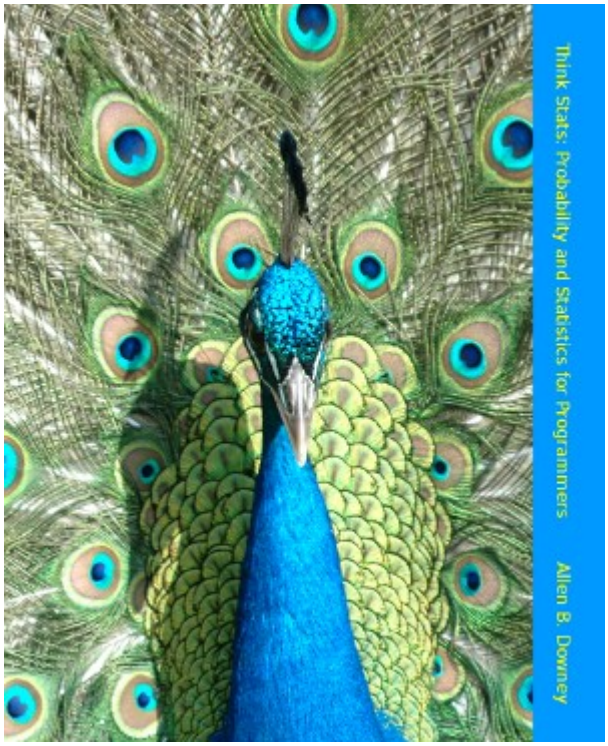
Think Stats: Xác suất thống kê dành cho người lập trình



Nếu biết lập trình, bạn đã có kĩ năng chuyển đổi dữ liệu thành kiến thức bằng cách dùng các công cụ xác suất và thống kê. Cuốn sách hướng dẫn này chỉ cho bạn cách thực hiện tính toán phân tích thống kê trên máy tính, thay vì dùng công thức toán học, với những chương trình viết bằng Python.

Bạn sẽ làm việc với một nghiên cứu cụ thể xuyên suốt cuốn sách, nhằm giúp bạn nắm được toàn bộ quá trình phân tích dữ liệu—từ thu thập dữ liệu và phát sinh các đặc trưng thống kê đến nhận diện các dạng mẫu và kiểm định giả thiết. Đồng thời, bạn cũng làm quen với các dạng phân bố, định luật xác suất, cách hiển thị, cùng nhiều công cụ và khái niệm khác.

Lời đề tựa của tác giả

Tôi viết cuốn sách này cho lớp học mà tôi dạy tại Olin College. Mục đích của lớp học là dạy cho sinh viên biết cách dùng công cụ thống kê để khám phá các bộ số liệu thực sự và trả lời những câu hỏi thú vị. Trang web của lớp học là: sites.google.com/site/thinkstats2011a — trong đó bao gồm bài giảng, bài tập trên lớp, bài về nhà, v.v.

Các ví dụ trong sách được viết bằng Python, nhưng là một phần nhỏ của ngôn ngữ lập trình này. Nếu bạn đã đọc hết 14 chương đầu tiên trong cuốn Think Python, bạn có thể sẵn sàng tiếp thu nội dung sách.

Tác giả, Allen Downey là giáo sư ngành Khoa học máy tính tại Olin College of Engineering. Ông đã dạy khoa học máy tính tại Wellesley College, Colby College và U.C. Berkeley. Ông nhận bằng tiến sĩ khoa học máy tính từ U.C. Berkeley và bằng thạc sĩ từ MIT.

Mục lục

Lời nói đầu

Lí do thôi thúc tôi viết quyển sách này

Think Stats: Xác suất thống kê dành cho người lập trình là một cuốn giáo trình nhập môn theo kiểu mới dành cho khóa học xác suất thống kê. Cuốn sách nhấn mạnh cách dùng thống kê để khảo sát những tập dữ liệu lớn. Cách tiếp cận là dùng máy tính, vốn có một số ưu điểm:

- Các bạn sinh viên viết chương trình như một cách xây dựng và thử nghiệm kiến thức. Chẳng hạn, bạn viết các hàm để tính phép khớp bình phương nhỏ nhất, tính phần dư, và các hệ số của định thức. Việc viết và thử nghiệm mã lệnh đòi hỏi bạn hiểu được những khái niệm và ngầm sửa được những hiểu nhầm về môn học.
- Các bạn chạy những chương trình để thử nghiệm các biểu hiện thống kê. Chẳng hạn, có thể khám phá Định lý giới hạn trung tâm (Central Limit Theorem, CLT) bằng cách phát sinh các mẫu từ một số dạng phân bố khác nhau. Khi thấy được rằng tổng của các giá trị tuân theo phân bố Pareto không hội tụ về phân bố chuẩn, bạn sẽ nhớ lại về những giả thiết cơ sở của CLT.
- Một số ý tưởng rất khó nắm bắt về mặt toán học lại rất dễ hiểu qua mô phỏng. Chẳng hạn, ta xấp xỉ các giá trị p bằng cách chạy mô phỏng Monte Carlo; cách này sẽ củng cố ý nghĩa của giá trị p.
- Việc dùng những dạng phân bố rời rạc và chương trình máy tính có thể minh họa cho các chủ đề như ước tính Bayes, vốn thường không được đưa vào các lớp nhập môn. Chẳng hạn, một bài tập yêu cầu bạn tính phân bố hậu nghiệm của "bài toán xe tăng Đức", vốn rất khó tính toán bằng cách giải tích nhưng sẽ dễ khi dùng máy tính.
- Vì được dùng một ngôn ngữ lập trình đa dụng (Python), bạn sẽ có thể nhập dữ liệu từ hầu như bất kì nguồn nào. Không có sự bó hẹp trong những dữ liệu được làm sạch và định dạng riêng cho một công cụ phần mềm thống kê nào đó.

Cuốn sách này thích hợp cho cách dùng với bài tập lớn. Trong lớp tôi, sinh viên phải hoàn thành một bài tập lớn trong một học kì, trong đó yêu cầu họ đặt ra một câu hỏi thống kê, tìm một tập số liệu phục vụ cho nhận định đó, rồi áp dụng từng kĩ thuật mới học được vào cho chính dữ liệu họ đã chọn.

Để giới thiệu về kiểu phân tích mà tôi muốn sinh viên làm theo, cuốn sách này giới thiệu một nghiên cứu cụ thể xuyên suốt các chương. Nó dùng số liệu từ hai nguồn:

- The National Survey of Family Growth (NSFG) [Chương trình điều tra của Hoa Kỳ về tăng trưởng gia đình], do U.S. Centers for Disease Control and Prevention (CDC) [các Trung tâm phòng chống kiểm soát dịch bệnh] tiến hành để thu thập "thông tin về cuộc sống gia đình, kết hôn và li dị, thai nghén, vô sinh, sử dụng biện pháp tránh thai, và sức khỏe nam nữ." (Xem `http://cdc.gov/nchs/nsfg.htm`.)
- The Behavioral Risk Factor Surveillance System (BRFSS) [Hệ thống theo dõi nhân tố rủi ro về biểu hiện], do National Center for Chronic Disease Prevention and Health Promotion [Trung tâm của Hoa Kỳ về Phòng bệnh mãn tính và phát triển sức khỏe] để "theo dõi tình trạng sức khỏe và các biểu hiện rủi ro trên lãnh thổ Hoa Kỳ." (Xem `http://cdc.gov/BRFSS/`.)

Các ví dụ khác đã dùng số liệu từ IRS, U.S. Census, và Boston Marathon.

Tôi đã viết cuốn sách này thế nào

Khi một tác giả viết cuốn sách mới, họ thường bắt đầu bằng việc đọc một chồng sách cũ. Vì vậy, phần lớn những cuốn sách mới đều có chứa cùng nội dung theo thứ tự gần như giống các cuốn sách trước đó. Thường có những cụm từ, những lỗi sai lan truyền từ quyển cũ sang quyển mới. Stephen Jay Gould đã viết một bài luận với tựa đề "The Case of the Creeping Fox Terrier[1] Clone" có đề cập đến ví dụ kiểu này.

Tôi thì không làm như vậy. Thật ra, khi viết sách này tôi gần như không dùng tài liệu in ấn nào, với một số lí do:

- Mục đích của tôi là khám phá một hướng tiếp cận mới trong tập sách này, vì vậy tôi không muốn đề cập nhiều đến những cách tiếp cận đã có.
- Vì tôi phát hành cuốn sách này theo một giấy phép tự do nên cũng muốn đảm bảo rằng khô có chỗ nào trong sách bị ảnh hưởng bởi những ràng buộc về bản quyền.
- Nhiều độc giả của tôi không có điều kiện tham khảo sách in trong thư viện, vì vậy tôi cố gắng đưa các tài liệu bổ trợ trong sách lên Internet để bạn truy cập miễn phí.
- Những người ủng hộ cho kiểu sách truyền thống thì cho rằng chỉ dùng những tài liệu điện tử là sự lười biếng và thiếu tin cậy. Họ có thể nói đúng vế đầu, như vế sau thì tôi nghĩ họ đã sai, vì vậy tôi muốn kiểm nghiệm giả thuyết của mình.

Tài liệu tôi tham khảo nhiều nhất là Wikipedia, "ông ba bị" đối với mọi thư viện. Nói chung, các bài báo tôi đọc về những chủ đề thống kê đều rất tốt (dù tôi đã đồng thời sửa chữa một số chỗ). Tôi ghi chú suốt quyển sách những tham khảo đến các trang Wikipedia, và khuyên bạn nên theo các đường link đó để tìm hiểu thêm; nhiều khi trang Wikipedia nói tiếp vào đúng những chỗ tôi tạm ngừng lại. Những thuật ngữ và kí hiệu dùng trong sách này nói chung đều thống nhất với Wikipedia, trừ khi tôi có một lý do thỏa đáng để viết khác.

Những nguồn tham khảo khác mà tôi thấy có ích gồm Wolfram MathWorld và (dĩ nhiên là) Google. Tôi cũng dùng hai cuốn sách khác, *Information Theory, Inference, and Learning Algorithms* của David McKay, quyển sách khiến tôi đam mê thống kê Bayes, và *Numerical Recipes in C* của Press và nnk. Nhưng cả hai quyển đều xem được trên mạng miễn phí, vì vậy tôi cũng không quá xấu hổ.

Allen B. Downey
Needham MA

Danh sách độc giả góp ý sửa chữa

Nếu bạn có góp ý sửa chữa cuốn sách, hãy gửi thư điện tử đến {downey@allendowney.com}. Nếu tôi thực hiện thay đổi theo trong thư của bạn, tôi sẽ điền tên bạn vào danh sách dưới đây (trừ khi bạn từ chối). Nếu bạn ghi cả một phần đoạn câu chứa lỗi thì sẽ tiện hơn nhiều để tôi tìm ra câu lỗi đó. Nếu ghi trang sách và số đề mục thì cũng được nhưng không tiện bằng. Cám ơn bạn!

- Lisa Downey và June Downey đọc bản nháp và có nhiều góp ý, sửa chữa.
- Steven Zhang tìm thấy một vài lỗi.

- Andy Pethan và Molly Farison giúp việc gỡ lỗi cho một số lời giải, và Molly phát hiện ra một số lỗi đánh máy.
- Andrew Heine phát hiện một lỗi trong hàm sai số của tôi.
- Dr. Nikolas Akerblom biết rằng loài Hyracotherium to cỡ nào.
- Alex Morrow làm rõ một đoạn ví dụ mã lệnh.
- Jonathan Street tìm thấy một lỗi vào ngay phút cuối cùng.
- Gábor Lipták tìm thấy một lỗi đánh máy trong cuốn sách và lời giải cho bài toán chạy tiếp sức.
- Cám ơn Kevin Smith và Tim Arnold rất nhiều vì đã phát triển plasTeX; tôi dùng công cụ này để chuyển cuốn sách sang dạng DocBook.
- George Caplan gửi một số gợi ý về cách viết để rõ ý hơn.
- Julian Ceipek đã phát hiện được một lỗi.
- Stijn Debrouwere, Leo Marihart III, Jonathan Hammler, và Kent Johnson phát hiện ra các lỗi trong bản in đầu tiên.

---

1. Một loài chó to bằng nửa con thú Hyracotherium (xem http://wikipedia.org/wiki/Hyracotherium). ↩

Chương 1: Tư duy thống kê cần cho người lập trình

> Dịch từ cuốn Think Stats: Probability and Statistics for Programmers của Allen B. Downey, NXB Green Tea Press. Sách điện tử được phát hành miễn phí theo giấy phép: Creative Commons Attribution-NonCommercial 3.0 Unported License. Trong quá trình phân phối bạn nên ghi rõ nguồn gốc cuốn sách.

Cuốn sách này bàn về việc chuyển đổi số liệu thành kiến thức. Số liệu thì rẻ, chí ít là so với kiến thức, vốn là thứ rất khó kiếm được.

Tôi sẽ trình bày ba mảng kiến thức liên quan đến nhau:

Xác suất
: là môn học về các sự kiện ngẫu nhiên. Hầu hết chúng ta ai cũng có hiểu biết trực quan về xác suất ở một mức độ nào đó, vì vậy mà thường nói "dường như" mà chẳng cần ai dạy, nhưng ta sẽ nghiên cứu về cách lượng hóa những khẳng định đó bằng những con số.

Thống kê
: là lĩnh vực trong đó dùng các mẫu số liệu để hỗ trợ cho nhận định về quần thể. Đa phần các phân tích thống kê đều dựa vào xác suất, đó là lý do tại sao hai mảng kiến thức này thường được trình bày cùng nhau.

Tính toán
: là công cụ thích hợp để phân tích định lượng, và máy tính thường được dùng để xử lý bài toán thống kê. Ngoài ra, các thí nghiệm tính toán cũng giúp ích cho việc tìm hiểu các khái niệm trong xác suất và thống kê.

Phương châm của cuốn sách này là nếu bạn đã biết lập trình, bạn có thể dùng kĩ năng đó để tự giúp mình hiểu được xác suất và thống kê. Các chủ đề này thường được trình bày từ góc nhìn toán học, và có thể phù hợp với những người nhất định. Nhưng một số ý tưởng quan trọng trong lĩnh vực này rất khó hiểu theo cách toán học và lại tương đối dễ tiếp cận bằng cách dùng máy tính.

Phần còn lại của chương này trình bày một nghiên cứu cụ thể được khởi nguồn từ câu hỏi mà tôi nghe được khi vợ chồng chúng tôi đón chờ cháu đầu lòng: liệu bé đầu có xu hướng ra chậm hay không?

Liệu trẻ đầu lòng có ra đời chậm hay không?

Nếu bạn tra Google câu hỏi này, bạn sẽ tìm thấy nhiều cuộc thảo luận khác nhau. Có người cho rằng đúng, người khác cho rằng đó chỉ là tưởng tượng, và có người cho rằng phải ngược lại: bé đầu lòng sẽ ra sớm.

Trong nhiều cuộc thảo luận nói trên, có người đưa ra con số để hỗ trợ cho khẳng định của mình. Tôi đã thấy nhiều ví dụ kiểu như:

> "Hai người bạn tôi gần đây vừa sinh cháu đầu, *cả hai* đều sinh cháu chậm 2 tuần trước khi dùng biện pháp ép sinh."

> “Cháu đầu lòng của tôi ra đời chậm 2 tuần và bây giờ đứa thứ hai sẽ ra đời sớm 2 tuần!!”
>
> “Tôi không nghĩ rằng điều đó đúng vì chị của tôi là cả, và được ra đời sớm, và các anh chị em họ của tôi cũng vậy.”

Những thông báo kiểu này được gọi là **dẫn chứng chủ quan** vì chúng dựa trên số liệu chưa được công bố và thường mang tính cá nhân. Khi nói chuyện thông thường, chẳng có gì sai khi kể chuyện cá nhân như vậy, nên tôi cũng không hề có ý trách những người nêu ý kiến nói trên. Nhưng ta có thể muốn các bằng chứng có sức thuyết phục cao hơn và câu trả lời đáng tin cậy hơn. Khi xét theo các thiêu chuẩn đó, dẫn chứng cá nhân thường không đạt, vì:

Số lần quan sát ít:
: Nếu thời gian thai nghén đối với trẻ đầu lòng là lâu hơn, thì mức độ khá biệt có lẽ vẫn nhỏ so với quy luật tự nhiên vì thời gian này luôn biến động. Trong trường hợp này, có thể ta phải so sánh nhiều trường hợp thai nghén để chắc chắn rằng có sự khác biệt.

Sự thiên lệch trong lựa chọn:
: Người tham gia vào cuộc tranh luận này có thể quan tâm đến câu hỏi chỉ vì con đầu lòng của họ ra đời muộn. Trong trường hợp này quá trình chọn lựa số liệu sẽ làm thiên lệch kết quả.

Sự thiên lệch khi khẳng định:
: Người tin vào lời khẳng định có nhiều khả năng đóng góp những ví dụ minh chứng cho điều đó. Những người nghi ngờ về lời khẳng định có nhiều khả năng dẫn ra những phản ví dụ.

Sự không chính xác:
: Các kinh nghiệm thường gắn với chuyện cá nhân, và thường bị nhớ nhầm, truyền đạt nhầm, thuật lại không chính xác, v.v.

Vậy bằng cách nào để ta làm tốt hơn?

Cách tiếp cận thống kê

Để nhận định những hạn chế của kinh nghiệm, chúng ta sẽ dùng các công cụ thống kê, bao gồm:

Thu thập số liệu:
: Ta sẽ dùng số liệu từ một cuộc điều tra lớn mang tầm cỡ quốc gia, được thiết kế rõ ràng nhằm mục tiêu đưa ra những suy luận đúng đắn theo thống kê, về dân số Hoa Kỳ.

Thống kê mô tả:
: Chúng ta sẽ tính ra các đặc trưng thống kê để có thể tóm gọn được số liệu, và đánh giá những cách khác nhau để hiển thị số liệu.

Phân tích khảo sát số liệu:
: Ta sẽ tìm kiếm những dạng mẫu, sự khác biệt, và những đặc điểm khác để giải đáp câu hỏi cần quan tâm. Đồng thời ta cũng kiểm tra sự không thống nhất và phát hiện ra những hạn chế.

Kiểm định giả thiết:
: Khi chúng ta thấy hiệu ứng rõ rệt, như một sự khác biệt giữa hai nhóm, ta sẽ đánh giá xem sự khác biệt đó là thực sự, hay là chỉ tình cờ diễn ra.

Ước tính:

Chúng ta sẽ dùng dữ liệu từ một mẫu để ước tính các đặc trưng của quần thể.

Bằng cách thực hiện những bước này cẩn thận để tránh những lỗi dễ mắc, ta có thể đi đến những kết luận hợp lý hơn và có nhiều khả năng sẽ đúng.

Cuộc điều tra quốc gia Hoa Kỳ về phát triển gia đình

Từ năm 1973, Trung tâm Kiểm soát và phòng chống dịch bệnh của Hoa Kỳ (U.S. Centers for Disease Control and Prevention, CDC) đã tiến hành đợt điều tra trên quy mô quốc gia về sự phát triển gia đình (National Survey of Family Growth, NSFG), vốn nhằm mục đích thu thập "thông tin về cuộc sống gia đình, kết hôn và li dị, thai nghén, vô sinh, sử dụng biện pháp tránh thai, và sức khỏe nam nữ. Kết quả của cuộc điều tra được dùng […] để hoạch định những dịch vụ chăm sóc sức khỏe và chương trình giáo dục sức khỏe, và để tiến hành nghiên cứu thống kê về gia đình, sức khỏe và khả năng sinh sản." [1] Chúng ta sẽ dùng số liệu thu thập từ đợt điều tra này để xem có phải là bé đầu lòng thường hay ra đời muộn không, và cả những câu hỏi khác. Để tận dụng được số liệu này, chúng ta phải hiểu được cách thức tiến hành cuộc điều tra này.

NSFG là công trình nghiên cứu **cắt ngang**, tức là nó chỉ ghi nhận tình hình của hệ thống tại một thời điểm nhất định. Hình thức nghiên cứu thường gặp khác là nghiên cứu **dọc**, trong đó quan sát hệ thống lặp lại nhiều lần trong suốt một khoảng thời gian. Đợt điều tra NSFG đã được tiến hành 7 lần; mỗi lần triển khai được gọi là một **cycle** (chu trình). Ta sẽ dùng dữ liệu từ Cycle 6, với thời gian tiến hành từ tháng 1/2002 đến 3/2003. Mục tiêu của đợt điều tra là rút ra kết luận về một **tổng thể**; đối với NSFG tổng thể là cư dân Mỹ có độ tuổi từ 15 đến 44. Những người tham gia vào cuộc điều tra được gọi là **respondents**; một nhóm các respondent được gọi là **cohort**. Nhìn chung, các nghiên cứu cắt ngang nhằm mục đích **đại biểu**, tức là mỗi thành viên trong tổng thể cần nghiên cứu đều có cơ hội tham gia như nhau. Dĩ nhiên là điều lý tưởng đó rất khó xảy ra trên thực tế, song những người tiến hành điều tra cố gắng thực hiện điều này ở mức tốt nhất có thể. NSFG không có tính đại biểu; nó được cố ý **lặp mẫu**. Các nhà thiết kế cho đợt nghiên cứu này đã tuyển ba nhóm người—Hispanics, Mỹ-Phi và thiếu niên—với tần số cao hơn mức đại biểu của họ trong dân số Hoa Kỳ. Lý do cho việc tuyển lặp mẫu này là cần đảm bảo được số người được điều tra trong mỗi nhóm phải đủ lớn để rút ra suy luận thống kê đúng. Dĩ nhiên, nhược điểm của lặp mẫu là không dễ để rút ra kết luận về tổng thể dựa trên con số thống kê từ cuộc điều tra. Ta sẽ trở lại vấn đề này sau.

> Mặc dù NSFG đã được tiến hành 7 lần nhưng nó không phải là một nghiên cứu dọc. Hãy đọc các trang Wikipedia http://wikipedia.org/wiki/Cross-sectional_study và http://wikipedia.org/wiki/Longitudinal_study để chắc rằng bạn tự hiểu lý do.
>
> Trong bài tập này, bạn sẽ tải số liệu về từ NSFG; chúng ta sẽ dùng số liệu này xuyên suốt cuốn sách.
>
> 1. Đến trang http://thinkstats.com/nsfg.html. Hãy đọc các điều khoản quy định việc sử dụng số liệu này và kích chuột vào đường link "I accept these terms" (coi như là bạn cam kết thực hiện điều khoản).
> 2. Tải về các file có tên là `2002FemResp.dat.gz` và `2002FemPreg.dat.gz`. File đầu có chứa những câu trả lời, trong đó mỗi

dòng là 1 câu trả lời của từng phụ nữ trong số tổng cộng 7643 người. Ở file thứ hai, mỗi dòng ứng với một trường hợp mang thai mà người tham gia báo cáo.

3. Tài liệu trực tuyến về đợt điều tra được đặt ở http://nsfg.icpsr.umich.edu/cocoon/WebDocs/NSFG/public/index.htm. Hãy xem các mục ở thanh bên trái để hình dung được có những số liệu gì. Bạn cũng có thể đọc bảng câu hỏi tại http://cdc.gov/nchs/data/nsfg/nsfg_2002_questionnaires.htm.
4. Trang web cho cuốn sách này cung cấp mã lệnh để xử lý các file số liệu từ NSFG. Hãy tải về http://thinkstats.com/survey.py và chạy nó trong cùng thư mục với các file số liệu. Nó sẽ đọc các file số liệu và in ra số dòng trong mỗi file:

```
Number of respondents 7643
Number of pregnancies 13593
```

5. Hãy xem qua mã lệnh để hình dung ra công dụng của nó. Mục tiếp theo sẽ giải thích cách hoạt động của mã lệnh này.

Bảng và bản ghi

Nhà thơ–triết gia Steve Martin có lần đã nói [tạm dịch]:

> "Oeuf" nghĩa là trứng, "chapeau" nghĩa là mũ. Dường như tiếng Pháp luôn có một từ dành cho mỗi vật.

Cũng như tiếng Pháp, người lập trình cơ sở dữ liệu nói một ngôn ngữ hơi khác, và vì chúng ta sẽ làm việc với cơ sở dữ liệu, ta cần phải học một số từ vựng liên quan. Mỗi dòng trong file ghi ý kiến có chứa thông tin về một ý kiến. Thông tin này được gọi lời **bản ghi**. Các biến tạo nên một bản ghi được gọi là **trường**. Một tập hợp các bản ghi được gọi là **bảng**. Nếu đọc `survey.py` bạn thấy lời định nghĩa lớp cho `Record`, vốn là một đối tượng thể hiện một bản ghi, và `Table`, vốn biểu diễn cho một bảng.

Có hai lớp con của `Record`—`Respondent` và `Pregnancy`—vốn chứa các bản ghi từ các bảng ý kiến và các ca mang thai. Tạm thời lúc này, các lớp này đều rỗng; nói riêng, không có phương thức khởi tạo nào để gán giá trị ban đầu cho các thuộc tính của chúng. Thay vào đó ta sẽ dùng `Table.MakeRecord` để chuyển đổi một dòng chữ thành một đối tượng `Record`. Cũng có hai lớp con của `Table`: `Respondents` và `Pregnancies`. Phương thức khởi tạo trong mỗi lớp chỉ rõ tên mặc định của file dữ liệu và kiểu bản ghi cần tạo ra. Mỗi đối tượng `Table` có một thuộc tính tên là `records`, vốn là một danh sách chứa các đối tượng `Record`.

Với mỗi `Table`, phương thức `GetFields` trả lại một danh sách các bộ để chỉ định các trường từ bản ghi mà sẽ được lưu vào các thuộc tính trong mỗi đối tượng `Record`. (Nếu cần, bạn có thể đọc lại câu trước.)

Chẳng hạn, sau đây là `Pregnancies.GetFields`:

```
    def GetFields(self):
        return [
            ('caseid', 1, 12, int),
            ('prglength', 275, 276, int),
```

```
('outcome', 277, 277, int),
('birthord', 278, 279, int),
('finalwgt', 423, 440, float),
]
```

Bộ thứ nhất cho thấy rằng trường `caseid` nằm ở các cột từ 1 đến 12 và nó là một số nguyên. Mỗi bộ có chứa các thông tin sau đây:

trường:
: Tên của thuộc tính mà trường sẽ được lưu vào. Thường thì tôi sẽ lấy chính tên gọi trong quy định (codebook) của NSFG, rồi chuyển sang toàn bộ chữ thường.

điểm đầu:
: Chỉ số của cột bắt đầu trường này. Chẳng hạn, điểm đầu của `caseid` là 1. Bạn có thể tra các chỉ số này trong NSFG codebook tại http://nsfg.icpsr.umich.edu/cocoon/WebDocs/NSFG/public/index.htm.

điểm cuối:
: Chỉ số của cột cuối trong trường; chẳng hạn, điểm cuối của `caseid` là 12. Khái với quy định trong Python, chỉ số cuối này { cũng tính vào} khoảng.

hàm chuyển đổi:
: Một hàm nhận vào một chuỗi và chuyển nó sang kiểu thích hợp. Bạn có thể dùng các hàm lập sẵn, như `int` và `float`, hoặc các hàm bạn tự định nghĩa. Nếu việc chuyển đổi thất bại, thuộc tính sẽ nhận được giá trị là chuỗi `'NA'`. Nếu bạn không muốn chuyển đổi trường nào, hãy cung cấp một hàm bất biến hoặc dùng `str`.

Với các bản ghi những ca mang thai, chúng ta kết xuất các biến sau:

caseid
: là số nguyên thứ tự của ý kiến.

prglength
: là số nguyên chỉ quãng thời gian mang thai tính theo tuần.

outcome
: là một mã số nguyên chỉ kết quả của thai nghén. Số 1 để chỉ việc sinh nở thành công.

birthord
: là số nguyên chỉ thứ tự của đứa trẻ sinh ra trong trường hợp còn sống; chẳng hạn khi sinh con đầu lòng thì mã số bằng 1. Nếu trường hợp ca đẻ không thành công, thì trường này sẽ được bỏ trống.

finalwgt
: là trọng số thống kê ứng với mỗi ý kiến. Nó là giá trị số chấm động để chỉ số người trong dân số nước Mỹ mà ý kiến này đại diện cho. Các thành viên của những nhóm được lấy mẫu lặp lại thì có trọng số thấp hơn.

Nếu đọc kĩ file ghi chép, bạn sẽ thấy rằng phần nhiều các biến này đều là **số liệu suy diễn**, nghĩa là chúng không phải các số liệu gốc mà được tính ra từ số liệu gốc. Chẳng hạn, với các ca đẻ thành công thì `prglength` bằng biến gốc `wksgest` (số tuần thai nghén) nếu có số liệu; còn nếu không

nó sẽ được tính theo công thức `mosgest * 4.33` (số tháng thai nghén nhân với số tuần có trong một tháng).

Các số liệu suy diễn thường dựa trên logic để kiểm tra sự thống nhất và độ chính xác của số liệu. Nói chung, dùng số liệu suy diễn là tốt trừ phi có một lý do thuyết phục để bạn tự xử lý số liệu gốc.

Bạn có thể cũng nhận thấy được rằng `Pregnancies` có một phương thức mang tên `Recode` để thực hiện kiểm tra thêm và suy diễn số liệu.

Trong bài tập này, bạn sẽ viết một chương trình để khám phá số liệu trong bảng Pregnancies.

1. Trong cùng thư mục mà bạn lưu `survey.py` cùng với các file dữ liệu, hãy tạo ra một file tên là `first.py` và gõ vào, hoặc dán đoạn mã sau vào:

   ```
   import survey
   table = survey.Pregnancies()
   table.ReadRecords()
   print 'Number of pregnancies', len(table.records)
   ```

   Kết quả cần thu được là 13593 trường hợp mang thai.

2. Hãy viết một vòng lặp để lặp qua `table` và đếm số ca đẻ thành công. Hãy tìm tài liệu ghi về `outcome` và kiểm tra để chắc rằng kết quả thu được thống nhất với phần tóm tắt trong tài liệu này.
3. Hãy chỉnh lại vòng lặp để chia những bản ghi có ca đẻ thành công thành hai nhóm, một là cho bé đầu lòng và một cho các trường hợp còn lại. Một lần nữa, hãy đọc tài liệu ghi về `birthord` để xem liệu ràng kết quả tính được có thống nhất không.

   Mỗi khi làm việc với một tập số liệu mới, việc kiểm tra như thế này sẽ giúp ích tìm ra các lỗi và sự không thống nhất có trong số liệu, phát hiện ra các lỗi trong chương trình vừa viết, và kiểm tra xem bạn có hiểu được cách mã hóa các trường hay không.

4. Hãy tính độ dài thời kì mang thai trung bình (theo tuần) đối với các bé đầu lòng và trường hợp khác. Có sự khác biệt gì giữa hai nhóm này không? Khác biệt có lớn không?

Bạn có thể tải về lời giải cho bài tập này từ `http://thinkstats.com/first.py`.

Mức ý nghĩa

Ở bài tập trước, bạn đã so sánh thời kì mang thai bé đầu lòng với các trường hợp khác; nếu giải đúng, bạn sẽ thấy rằng tính trung bình, bé đầu lòng sẽ chào đời muộn hơn khoảng 13 giờ. Một sự khác biệt như vậy được gọi là **hiệu ứng biểu kiến**; nghĩa là, có thể điều gì đó đang diễn ra, nhưng bạn không biết chắc. Vẫn có một vài câu hỏi mà bạn muốn nêu ra:

- Nếu hai nhóm có số trung bình khác nhau, những **đặc trưng thống kê** khác như trung vị và phương sai thì sao? Chúng ta có thể nói một cách chính xác hơn xem hai nhóm khác nhau đến mức nào không?

- Liệu có thể là sự khác biệt ta vừa thấy đã xảy ra tình cờ, ngay cả khi nếu các nhóm được so sánh thực ra là giống nhau? Nếu vậy, ta có thể kết luận rằng hiệu ứng đó không có **ý nghĩa về mặt thống kê**.
- Liệu có thể hiệu ứng biểu kiến kia là do lựa chọn thiên lệch hay một sai sót gì trong khâu thiết lập thử nghiệm? Nếu vậy thì ta có thể kết luận rằng hiệu ứng này là một **dị biệt**; nghĩa là điều được tình cờ tạo ra chứ không phải do bạn phát hiện ra.

Trong phần còn lại của cuốn sách này ta sẽ tập trung giải đáp các câu hỏi trên.

> Cách học thống kê tốt nhất là thực hiện một dự án nghiên cứu về vấn đề bạn quan tâm. Liệu có một câu hỏi như, “Bé đầu lòng có chào đời muộn không,” mà bạn cần tìm hiểu không?
>
> Hãy nghĩ về những câu hỏi mà bản thân bạn thấy thú vị, hoặc những vấn đề thuộc về tri thức phổ quát, các chủ đề gây tranh cãi, hay những câu hỏi có quan điểm chính trị, rồi xem liệu bạn có thể lập nên một câu hỏi thích hợp cho việc điều tra thống kê không.
>
> Hãy tìm dữ liệu để giúp bạn trả lời câu hỏi. Chính phủ thường cung cấp nguồn dữ liệu tốt vì số liệu từ nghiên cứu cộng đồng thường có thể truy cập được miễn phí.[2]
>
> Một cách tìm khác để tìm thông tin là Wolfram Alpha, vốn là một tập hợp thông tin đã qua xử lý, có chất lượng cao tại `http://wolframalpha.com`. Kết quả từ Wolfram Alpha chịu các hạn chế bản quyền; có thể bạn phải xem các điều khoản trước khi tiến hành khai thác từ nguồn này.
>
> Google và các máy tìm kiếm khác cũng có thể giúp bạn tìm số liệu, nhưng có thể khó đánh giá hơn về chất lượng của các tài nguyên trên mạng.
>
> Nếu có vẻ như ai đó đã trả lời câu hỏi bạn đặt ra, hãy xem kĩ rằng câu trả lời đã thỏa đáng chưa. Có thể do số liệu hoặc cách tính bị lỗi mà kết luận đã đưa ra vẫn chưa đáng tin cậy. Với trường hợp như thế, bạn có thể tính cách khác cho cùng số liệu đó, hoặc tìm kiếm một nguồn số liệu tốt hơn.
>
> Nếu đã tìm thấy một bài báo được công bố đã giải đáp câu hỏi đặt ra, bạn có thể lấy được số liệu gốc. Nhiều tác giả đưa dữ liệu công khai lên web, nhưng với các số liệu có tính nhạy cảm bạn phải liên lạc với tác giả, cung cấp các thông tin về việc bạn định dùng số liệu thế nào, hoặc đồng ý những điều khoản sử dụng nhất định. Hãy kiên trì!

Thuật ngữ

chứng cứ kinh nghiệm:
: Chứng cứ, thường thuộc về cá nhân, được thu thập tự phát thay vì theo một kế hoạch được thiết kế cẩn thận.

tổng thể:
: Nhóm được quan tâm nghiên cứu, thường là một nhóm người, nhưng thuật ngữ cũng dùng được với động vật, thực vật và khoáng chất[3].

nghiên cứu cắt ngang:
: Nghiên cứu trong đó thu thập số liệu về một tổng thể tại một thời điểm cụ thể.

nghiên cứu dọc:
: Nghiên cứu dõi theo một tổng thể qua thời gian, thu thập số liệu về tổng thể nhiều lần.

người tham gia:
: Người trả lời câu hỏi điều tra.

nhóm:
: Một nhóm người tham gia.

mẫu:
: Tập hợp con của tổng thể được dùng để thu thập số liệu.

đại biểu:
: Một mẫu có tính đại biểu nếu mỗi thành viên của tổng thể đều có cùng khả năng có ở trong mẫu.

lấy mẫu lặp:
: Kĩ thuật làm tăng tính đại biểu cho một tập con của tổng thể nhằm tránh lỗi gây ra do kích thước mẫu nhỏ.

bản ghi:
: Trong một cơ sở dữ liệu, tập hợp thông tin về một cá thể được nghiên cứu.

trường:
: Trong một cơ sở dữ liệu, một trong số các biến có tên cấu thành bản ghi.

bảng:
: Trong một cơ sở dữ liệu, tập hợp các bản ghi.

dữ liệu gốc:
: Các giá trị được thu thập và ghi lại mà không được hoặc ít được kiểm tra, tính toán hay diễn giải.

dữ liệu suy diễn:
: Giá trị được phát sinh bởi tính toán hoặc dùng phép logic đối với dữ liệu gốc.

đặc trưng thống kê:
: Kết quả của phép tính toán nhằm rút gọn một tập số liệu thành một con số (hay ít ra là một tập số nhỏ hơn) mà vẫn thể hiện được tính chất đặc trưng nào đó của số liệu.

hiệu ứng biểu kiến:
: Kết quả đo hoặc đặc trưng thống kê mà gợi thấy điều đáng quan tâm.

có ý nghĩa thống kê:
: Hiệu ứng biểu kiến được gọi là có ý nghĩa thống kê nếu nó có vẻ như không xuất hiện tình cờ.

dị biệt:
: Hiệu ứng biểu kiến được hình thành do thiên lệch, lỗi đo đạc, hay một dạng lỗi nào khác.

---

1. Xem http://cdc.gov/nchs/nsfg.htm. ↩

2. Khi tôi viết đến đây cũng là ngày có phiên tòa ở Anh xử để đi đến quyết định áp dụng Đạo luật Tự do Thông tin cho số liệu nghiên cứu khoa học. ↩
3. “Animal, Vegetable, Mineral”. Xem cụm từ này ở http://wikipedia.org/wiki/Twenty_Questions#Popular_variants. ↩

Chương 2: Thống kê mô tả

Trở về Muc luc cuốn sách

Trị trung bình và kỳ vọng

Ở chương trước, tôi đã đề cập đến ba đặc trưng thống kê—trị trung bình, phương sai và trung vị—mà không giải thích ý nghĩa của chúng. Vì vậy, trước khi đi tiếp, ta sẽ làm rõ điều này. Nếu bạn có một mẫu gồm n giá trị, x_i, thì giá trị trung bình, μ, là tổng của các giá trị trên chia cho số các giá trị; nói cách khác

$$\mu = (1 / n) \sum_i x_i$$

Các từ "trị trung bình" và "average" đôi khi có thể dùng thay thế được cho nhau, nhưng tôi vẫn muốn phân biệt:

- "Trị trung bình" của một mẫu là đặc trưng thống kê được tính theo công thức trên.
- "average" là một trong nhiều đặc trưng thống kê mà bạn có thể chọn để mô tả giá trị điển hình hay **xu hướng trung tâm** trong một mẫu.

Đôi khi trị trung bình là một cách mô tả tốt một tập hợp các giá trị. Chẳng hạn, các quả táo nói chung đều có kích thước gần bằng nhau (ít ra là táo bày bán ở siêu thị). Vì vậy nếu tôi mua 6 quả táo và khối lượng tổng cộng là 3 pound thì cũng có lý khi nói rằng mỗi quả táo nặng cỡ nửa pound. Nhưng bí đỏ thì đa dạng hơn. Chẳng hạn ở vườn nhà, tôi trồng bí và một ngày kia thua hoạch được 3 quả bí để bày, mỗi quả nặng 1 pound và hai quả bí pie, mỗi quả nặng 3 pound, và một quả bí Atlantic Giant nặng đến 591 pound. Trị trung bình của mẫu này là 100 pound, nhưng nếu tôi nói "Quả bí trung bình ở vườn nhà tôi nặng 100 pound," thì sẽ là sai, hoặc chí ít sẽ gây ngộ nhận. Trong trường hợp này, không có trung bình nào có ý nghĩa vì không có quả bí điển hình nào có khối lượng như vậy.

Phương sai

Nếu không có một con số nào đặc trưng được cho khối lượng quả bí thì tốt hơn là chúng ta dùng hai con số: trị trung bình và **phương sai**.

Cũng như việc trị trung bình được dùng vào mục đích mô tả khuynh hướng trung tâm, phương sai được dùng để mô tả độ **phân tán**. Phương sai của một tập hợp giá trị thì bằng

$$\sigma^2 = (1 / n) \sum_i (x_i - \mu)^2$$

Số hạng $x_i - \mu$ được gọi là "độ lệch so với trung bình," vì vậy phương sai là giá trị trung bình của bình phương độ lệch, đó là lí do tại sao nó được kí hiệu là σ^2. Căn bậc hai của phương sai, σ, được gọi là **độ lệch chuẩn**. Bản thân phương sai thì rất khó được diễn giải. Một vấn đề là đơn vị của nó rất kì lạ. Trong trường hợp này, đại lượng được đo tính theo pound, do đó phương sai được tính theo pound bình phương. Độ lệch chuẩn thì có ý nghĩa hơn, trong trường hợp này đơn vị của nó là pound.

> Để làm các bài tập trong chương này, bạn cần tải về `http://thinkstats.com/thinkstats.py`, vốn có chứa các hàm đa năng mà

ta sẽ sử dụng xuyên suốt cuốn sách. Bạn có thể đọc hướng dẫn sử dụng các hàm này ở http://thinkstats.com/thinkstats.html. Hãy viết một hàm có tên `Pumpkin` trong đó dùng các hàm từ `thinkstats.py` để tính trị trung bình, phương sai và độ lệch chuẩn của khối lượng các quả bí trong mục trước.

Hãy sử dụng lại mã lệnh từ `survey.py` và `first.py` để tính độ lệch chuẩn của thời kỳ mang thai các bé đầu lòng và các bé sinh sau. Liệu có phải là độ lệch trong hai trường hợp này cũng giống nhau không? Sự khác biệt giữa trị trung bình và độ lệch chuẩn tương ứng trong hai trường hợp này là bao nhiêu? Việc so sánh này có gợi ra điều gì về ý nghĩa thống kê của sự khác biệt?

Nếu có kinh nghiệm, bạn có thể đã thấy một công thức tính phương sai với $n - 1$ ở mẫu số, thay vì n. Đặc trưng thống kê này có tên là "phương sai mẫu," và nó được dùng để ước tính phương sai trong một tổng thể bằng cách dùng một mẫu. Ta sẽ quay trở lại vấn đề này trong Chương 8.

Phân bố

Các đặc trưng thống kê tuy gọn nhưng nguy hiểm ở chỗ chúng che khuất đi số liệu. Một cách làm khác là nhìn vào **phân bố** của số liệu, vốn miêu tả mỗi giá trị xuất hiện bao nhiêu lần.

Cách biểu thị chung nhất cho một phân bố là **biểu đồ tần số**, vốn là một đồ thị cho thấy các tần số hay tần suất của mỗi giá trị. Ở đây, **tần số** nghĩa là số lần xuất hiện của một giá trị trong tập dữ liệu—nó không có gì liên quan đến độ cao thấp của âm thanh hay việc chỉnh núm vặn của ra-đi-ô. Một **tần suất** là một tần số được biểu diễn dưới dạng tỷ lệ so với kích thước mẫu, *n*. Trong Python, một cách hiệu quả để biểu diễn các tần số là dùng từ điển. Với một dãy giá trị đã cho, `t`:

```
hist = {}
for x in t:
    hist[x] = hist.get(x, 0) + 1
```

Kết quả là một từ điển cho tương ứng (ánh xạ) mỗi giá trị với một tần số. Để chuyển từ tần số sang tần suất, ta đem chia cho *n*; cách này được gọi là **chuẩn hóa**:

```
n = float(len(t))
pmf = {}
for x, freq in hist.items():
    pmf[x] = freq / n
```

Biểu đồ tần suất (sau khi chuẩn hóa) được gọi là **PMF**, viết tắt cho "probability mass function" (hàm khối xác suất); tức là một hàm ánh xạ từ giá trị đến tần suất (còn ý nghĩa của "mass" [khối lượng] sẽ được giải thích ở Mục {mật độ xác suất}). Có thể sẽ dễ lẫn khi gọi một từ điển của Python là một hàm. Trong toán học, một hàm là phép ánh xạ từ một tập giá trị tới một tập giá trị khác. Trong Python, ta *thường* biểu diễn hàm toán học với các đối tượng hàm, nhưng trong trường hợp này ta dùng một từ điển (từ điển đôi khi cũng được gọi là "ánh xạ," bạn có thể gặp tên gọi này ở đâu đó).

Thể hiện biểu đồ tần số

Tôi đã viết một module Python có tên `Pmf.py` trong đó chứa định nghĩa hàm cho các đối tượng Hist, vốn để biểu diễn biểu đồ tần số, và các đối tượng Pmf, để biểu diễn các hàm khối xác suất.

Bạn có thể đọc hướng dẫn sử dụng tại `thinkstats.com/Pmf.html` và tải mã lệnh về từ `thinkstats.com/Pmf.py`.

Hàm `MakeHistFromList` nhận vào một danh sách các giá trị và trả lại một đối tượng Hist mới. Bạn có thể kiểm tra nó từ chế độ tương tác của Python:

```
>>> import Pmf
>>> hist = Pmf.MakeHistFromList([1, 2, 2, 3, 5])
>>> print hist
<Pmf.Hist object at 0xb76cf68c>
```

`Pmf.Hist` có nghĩa là đối tượng này là một thành viên của lớp Hist, vốn được định nghĩa trong module Pmf. Nói chung, tôi dùng chữ in để viết tên các lớp và hàm, còn tên biến thì được viết chữ thường toàn bộ.

Các đối tượng Hist cung cấp những phương thức để tra tìm giá trị cùng với tần suất tương ứng. `Freq` nhận vào một giá trị và trả lại tần số của nó:

```
>>> hist.Freq(2)
2
```

Nếu bạn tra tìm một giá trị mà thực tế không xảy ra, tần số sẽ bằng 0.

```
>>> hist.Freq(4)
0
```

`Values` trả lại một danh sách không được sắp xếp, có chứa các giá trị trong Hist:

```
>>> hist.Values()
[1, 5, 3, 2]
```

Để lặp qua các giá trị theo thứ tự, bạn có thể dùng hàm lập sẵn `sorted`:

```
for val in sorted(hist.Values()):
    print val, hist.Freq(val)
```

Nếu bạn dự định tra tìm tất cả các tần số, cách tốt hơn là dùng `Items`, vốn trả lại một danh sách không được sắp xếp gồm các cặp giá trị–tần số:

```
for val, freq in hist.Items():
    print val, freq
```

> Số đông (mode) của một dạng phân bố là giá trị hay xuất hiện nhất (xem http://wikipedia.org/wiki/Mode_(statistics)). Hãy viết một hàm có tên `Mode` nhận vào một đối tượng Hist và trả lại giá trị xuất hiện nhiều nhất.Một nhiệm vụ khó hơn là, hãy viết một hàm có tên `AllModes` nhận vào đối tượng Hist và trả lại một danh sách các cặp giá trị–tần số xếp theo thứ tự tần số giảm dần. Gợi ý: module `operator` có một hàm tên là `itemgetter` mà bạn có thể truyền như một khóa vào cho `sorted`.

Vẽ đồ thị tần số

Có một số gói Python đảm nhiệm việc vẽ hình và biểu đồ. Tôi sẽ trình bày gói `pyplot`, vốn là một phần của gói `matplotlib` tại http://matplotlib.sourceforge.net.

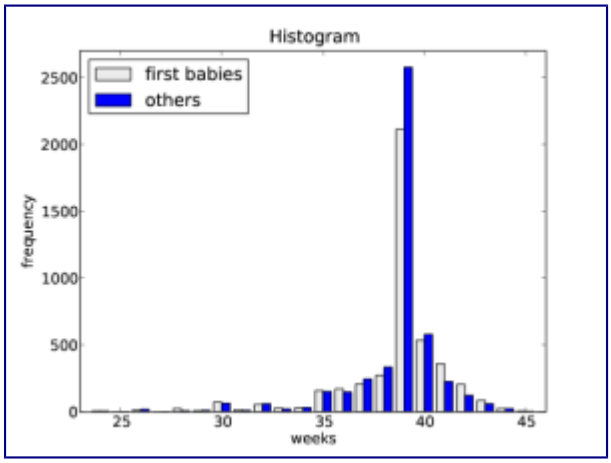
Gói này được kèm trong nhiều bản cài đặt Python. Để xem liệu máy của bạn đã có nó chưa, hãy bật trình thông dịch Python và chạy đoạn chương trình:

```
import matplotlib.pyplot as pyplot
pyplot.pie([1,2,3])
pyplot.show()
```

Nếu đã có `matplotlib` bạn sẽ nhìn thấy một biểu đồ hình quạt; còn nếu không bạn sẽ phải cài nó. Các biểu đồ tần số và PMF thường được vẽ dưới dạng biểu đồ thanh. Hàm `pyplot` để vẽ biểu đồ thanh có tên là `bar`. Các đối tượng Hist có phương thức tên là `Render` để trả lại một danh sách đã sắp xếp gồm các giá trị và một danh sách các tần số tương ứng, vốn được định dạng thích hợp với `bar`:

```
>>> vals, freqs = hist.Render()
>>> rectangles = pyplot.bar(vals, freqs)
>>> pyplot.show()
```

Tôi đã viết một module có tên `myplot.py` trong đó có các hàm vẽ biểu đồ tần số, biểu đồ PMF và các đối tượng khác mà ta sẽ sớm gặp. Bạn có thể đọc hướng dẫn sử dụng tại `thinkstats.com/myplot.html` và tải về mã lệnh từ `thinkstats.com/myplot.py`. Hoặc bạn có thể dùng `pyplot` trực tiếp, nếu muốn. Dù bằng cách nào, bạn đều có thể tìm được hướng dẫn dùng `pyplot` trên mạng. Hình dưới đây cho thấy biểu đồ tần số của các độ dài thời kì mang thai với trẻ đầu lòng và trẻ sinh sau.



Biểu đồ tần số độ dài thời kì mang thai. [nsfg_hist

Biểu đồ tần số rất có ích vì nó khiến cho những đặc điểm sau được rõ ràng ngay:

Số đông:
: Giá trị thường gặp nhất trong một phân bố được gọi là **số đông**. Trên Hình {nsfg_hist} rõ ràng có một số đông là 39 tuần. Ở trường hợp này, số đông là đặc trưng thống kê vì nó mô tả được rõ nhất giá trị điển hình.

Hình dạng:
: Xung quanh số đông, phân bố này có tính bất đối xứng; nó nhanh chóng giảm sút về phía tay phải nhưng chỉ giảm từ từ về phía tay trái. Theo quan điểm y học, điều này có nghĩa. Trẻ sơ sinh thường được ra đời sớm, nhưng hiếm khi muộn hơn 42 tuần. Ngoài ra, phần bên phải của phân bố này bị chặn lại bởi bác sĩ thường can thiệp với những ca thai nghén quá 42 tuần.

Điểm biệt lập:

> Giá trị nằm cách xa số đông được gọi là **điểm biệt lập**. Có điểm biệt lập do trường hợp hy hữu, như trẻ sinh ra vào tuần thứ 30. Nhưng đa số điểm biệt lập có thể do lỗi, trong quá trình báo cáo số liệu hoặc ghi chép số liệu.

Mặc dù biểu đồ tần số đã làm rõ một số đặc điểm, nhưng chúng thường không giúp ích cho việc so sánh hai dạng phân bố. Trong ví dụ này, số trẻ đầu lòng thì ít hơn số trẻ sinh sau, nên một phần biểu hiện khác nhau là do hai mẫu có kích thước khác nhau. Ta có thể giải quyết vấn đề này bằng cách dùng PMF.

Biểu diễn PMF

`Pmf.py` cung cấp một lớp có tên `Pmf` để biểu diễn các PMF. Cách kí hiệu có thể dễ lầm, nhưng nó là thế này: `Pmf` là tên của module đồng thời cũng là tên lớp, vì vậy tên đầy đủ của lớp là `Pmf.Pmf`. Tôi thường dùng `pmf` để đặt tên biến. Sau cùng, trong cuốn sách này, tôi dùng chữ PMF để chỉ khái niệm chung cho hàm khối xác suất, nó độc lập đến cách viết chương trình.

Để tạo ra một đối tượng Pmf, hãy dùng `MakePmfFromList`, vốn nhận vào một danh sách các giá trị:

```
>>> import Pmf
>>> pmf = Pmf.MakePmfFromList([1, 2, 2, 3, 5])
>>> print pmf
<Pmf.Pmf object at 0xb76cf68c>
```

Các đối tượng Pmf và Hist giống nhau về nhiều mặt. Các phương thức `Values` và `Items` có tác dụng giông nhau trong hai kiểu đối tượng trên. Khác biệt lớn nhất là Hist ánh xạ từ giá trị đến số đếm; còn Pmf ánh xạ từ giá trị đến tần suất có giá trị là số dấu phẩy động.

Để tra tìm tần suất ứng với một giá trị, hãy dùng `Prob`:

```
>>> pmf.Prob(2)
0.4
```

Bạn có thể sửa lại một Pmf đã có bằng cách tăng tần suất ứng với một giá trị:

```
>>> pmf.Incr(2, 0.2)
>>> pmf.Prob(2)
0.6
```

Hoặc bạn có thể nhân xác suất với một hệ số:

```
>>> pmf.Mult(2, 0.5)
>>> pmf.Prob(2)
0.3
```

Nếu sửa đổi một Pmf, kết quả có thể sẽ không được chuẩn hóa; nghĩa là các tần suất cộng lại sẽ không bằng 1 nữa. Để kiểm tra, bạn có thể gọi `Total`, vốn trả lại tổng các tần suất:

```
>>> pmf.Total()
0.9
```

Để chuẩn hóa lại, hãy gọi `Normalize`:

```
>>> pmf.Normalize()
>>> pmf.Total()
```

```
1.0
```

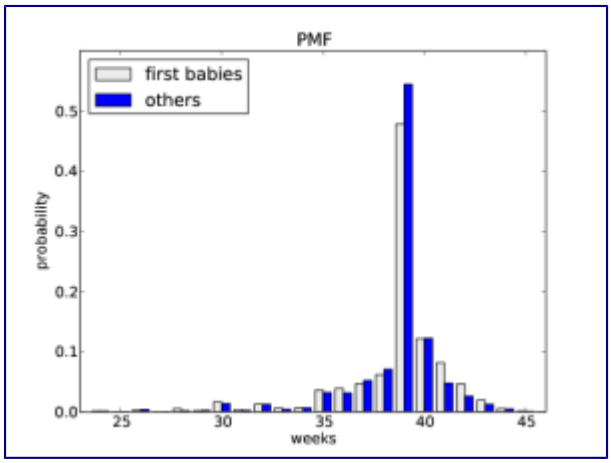
Đối tượng Pmf có một phương thức `Copy` để bạn có thể tạo ra một bản sao và sửa nó mà không làm ảnh hưởng đến bản gốc.

> Theo Wikipedia [tạm dịch], "Phân tích tồn vong là một nhánh của kê học liên quan đến hiện tượng chết của sinh vật và hỏng hóc của hệ cơ học;" xem `http://wikipedia.org/wiki/Survival_analysis`. Trong một phần của phân tích tồn vong, ta thường phải tính quãng đời còn lại của, chẳng hạn, một chi tiết máy. Nếu biết phân bố của quãng đời và tuổi thọ của chi tiết, thì ta có thể tính được phân bố của quãng đời còn lại.
>
> Hãy viết một hàm có tên `RemainingLifetime` nhận vào một Pmf của các quãng đời cùng tuổi hiện tại, rồi trả lại một Pmf mới biểu thị cho phân bố của quãng đời còn lại.
>
> Ở Mục {trị trung bình} ta đã tính được trị trung bình của mẫu bằng cách cộng các phần tử lại rồi chia cho *n*. Nếu đã có một PMF, bạn vẫn có thể tính trị trung bình, nhưng cách làm hơi khác:
> $\mu = \sum_i p_i x_i$
> trong đó các x_i là các giá trị duy nhất trong PMF mà p_i=PMF(x_i). Tương tự, bạn có thể tính phương sai như sau:
> $\sigma^2 = \sum_i p_i (x_i - \mu)^2$
> Hãy viết một hàm có tên `PmfMean` và `PmfVar` nhận vào một đối tượng Pmf rồi tính trị trung bình và phương sai, Để thử nghiệm các phương thức này, hãy kiểm tra xem chúng có thống nhất với các phương thức `Mean` và `Var` trong `Pmf.py` hay không.

Vẽ đồ thị các PMF

Có hai cách thường dùng để vẽ các Pmf:

- Để vẽ Pmf dưới dạng biểu đồ cột, bạn có thể dùng `pyplot.bar` hoặc `myplot.Hist`. Biểu đồ cột có ích nhất khi số các giá trị trong Pmf còn ít.
- Để vẽ Pmf dưới dạng đường, bạn có thể dùng `pyplot.plot` hoặc `myplot.Pmf`. Biểu đồ dạng đường có ích nhất khi có nhiều các giá trị và Pmf là một hàm trơn.



PMF của quãng thời gian mang thai.

Hình trên cho thấy PMF của các quãng thời gian mang thai được biểu diễn dưới dạng biểu đồ cột. Bằng cách dùng PMF, ta có thể thấy rõ hơn được sự khác biệt giữa hai phân bố. Trẻ đầu lòng thường có vẻ ít chào đời đúng tuần 39, mà thường có thiên hướng muộn (các tuần 41 và 42).

Mã lệnh để tạo ra các hình vẽ trong chương này có sẵn ở http://thinkstats.com/descriptive.py. Để chạy nó, bạn sẽ cần các module nhập vào và dữ liệu từ NSFG (xem Mục {nsfg}). Chú ý: `pyplot` có một hàm tên là `hist` nhận vào một dãy các giá trị, tính ra và vẽ biểu đồ tần số. Vì tôi dùng các đối tượng `Hist` nên thường không dùng đến `pyplot.hist`.

Điểm biệt lập

Điểm biệt lập là giá trị nằm cách xa xu hướng trung tâm. Điểm biệt lập có thể gây bởi các lỗi trong quá trình thu thập và xử lý số liệu, hoặc có thể đo đúng nhưng gặp hiện tượng bất thường. Bạn nên luôn kiểm tra các điểm biệt lập, và đôi khi việc loại bỏ nó là cần thiết và xác đáng.

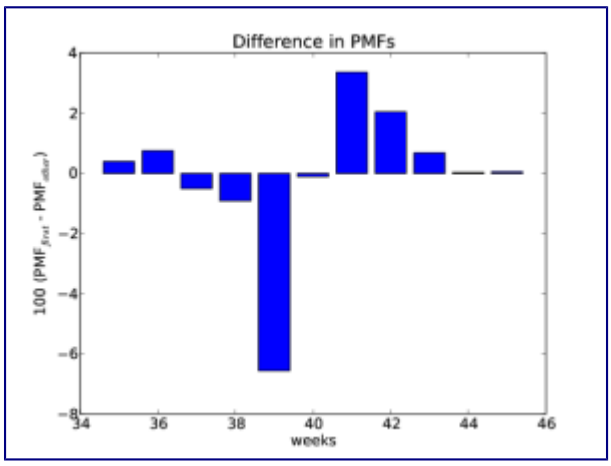
Trong danh sách các khoảng thời gian mang thai các ca đẻ thành công, 10 giá trị thấp nhất là {0, 4, 9, 13, 17, 17, 18, 19, 20, 21}. Các giá trị thấp hơn 20 tuần rõ ràng là có lỗi, và các giá trị cao hơn 30 tuần có lẽ là hợp lệ. Nhưng các giá trị nằm giữa 20 và 30 thì khó phân giải. Ở đầu kia, các giá trị cao nhất là:

```
weeks  count
43     148
44     46
45     10
46     1
47     1
48     7
50     2
```

Một lần nữa, có một số giá trị gần như chắc chắn là có lỗi, nhưng ta rất khó biết được toàn bộ. Một cách lựa chọn là **tỉa bớt** dữ liệu bằng cách bỏ đi một phần gồm các giá trị cao nhất và thấp nhất (xem http://wikipedia.org/wiki/Truncated_mean).

Những cách hiển thị khác

Biểu đồ tần số và PMF rất có ích cho việc phân tích khám phá số liệu; một khi bạn đã có ý tưởng về hiện tượng xảy ra thì thường sẽ cần sắp đặt để hiển thị số liệu nhằm tập trung vào hiệu ứng biểu kiến. Trong bộ số liệu NSFG, khác biệt lớn nhất giữa hai phân bố xảy ra ở gần số đông. Vì vậy sẽ có ý nghĩa khi ta phóng to phần đó của đồ ghị, và thực hiện chuyển đổi số liệu để làm nổi bật sự khác biệt. Hình {nsfg_diffs} cho thấy sự khác biệt giữa hai đường PMF trong tuần từ 35–45. Tôi đã nhân lên 100 lần để biểu thị độ chênh lệch tính theo phần trăm.



Độ chênh lệch theo phần trăm, tính với từng tuần. [nsfg_diffs

Hình vẽ này làm cho kiểu mẫu càng rõ hơn: trẻ đầu lòng thường ít chào đời vào tuần 39, và dễ chào đời vào các tuần 41 và 42 so với trẻ sinh sau.

Rủi ro tương đối

Chúng ta đã bắt đầu với câu hỏi, “Liệu trẻ đầu lòng có chào đời muộn không?” Chuẩn xác hơn, ta hãy nói rằng đứa trẻ được gọi là chào đời sớm nếu nó được sinh ra từ tuần 37 hoặc sớm hơn, ra đời đúng hạn nếu sinh vào các tuần 38, 39, hoặc 40; và ra đời muộn nếu sinh vào tuần 41 hoặc muộn hơn. Các khoảng như vậy, được dùng để nhóm dữ liệu, có tên gọi là **ngăn**.

> Hãy tạo ra một file có tên `risk.py`. Hãy viết các hàm tên là `ProbEarly`, `ProbOnTime` và `ProbLate` nhận vào một PMF rồi tính tỷ lệ ca sinh rơi vào từng ngăn. Gợi ý: hãy viết một hàm tổng quát để những hàm nói trên gọi đến.
>
> Hãy vẽ ba đường PMF, một đường cho trẻ đầu lòng, một đường cho trẻ sinh sau, và một đường chung cho tất cả trẻ được sinh ra. Với mỗi đường PMF, hãy tính xacs suất sinh sớm, sinh đúng hạn, và sinh muộn.
>
> Một cách tóm tắt số liệu như vậy là bằng **rủi ro tương đối**, vốn là tỉ số giữa hai xác suất. Chẳng hạn, xác suất một đứa trẻ đầu lòng ra đời sớm bằng 18,2%. Với các đứa trẻ sinh sau, xác suất này bằng 16,8%, vì vậy rủi ro tương đối bằng 1,08. Điều này nghĩa là trẻ đầu lòng có thêm chừng 8% khả năng chào đời sớm.
>
> Hãy viết mã lệnh để kiểm tra và khẳng định kết quả này, rồi tính các rủi ro tương đối giữa sinh đúng hạn và sinh muộn. Bạn có thể tải về một lời giải từ `http://thinkstats.com/risk.py`.

Xác suất có điều kiện

Hãy tưởng tượng rằng một người mà bạn biết đang mang bầu, bước vào tuần 39. Khả năng đứa trẻ sẽ được sinh ra vào tuần tới là bao nhiêu? Đáp số sẽ thay đổi thế nào nếu đó là trẻ đầu lòng?

Chúng ta có thể trả lời các câu hỏi trên bằng cách tính **xác suất có điều kiện**, vốn (a hèm!) là một xác suất phụ thuộc vào điều kiện nào đó. Trong trường hợp này, điều kiện là ta đã biết rằng đứa trẻ không sinh ra vào các tuần 0–38.

Sau đây là một cách tính:

1. Cho trước một đường PMF, ta phát sinh ra một nhóm giả gồm 1000 ca mang thai. Với mỗi tuần thứ *x*, số ca mang thai với khoảng thời gian *x* là 1000 PMF(*x*).
2. Xóa bỏ khỏi nhóm những ca mang thai với khoảng thời gian dưới 39.
3. Tính ra PMF của các thời kì còn lại; kết quả này là PMF có điều kiện.
4. Tính PMF có điều kiện ứng với *x* = 39 tuần.

Thuật toán này về khái niệm thì rất rõ ràng, nhưng không hiệu quả lắm. Một cách làm khác đơn giản hơn là xóa bỏ các giá trị nhỏ hơn 39 trong phân bố rồi chuẩn hóa lại.

> Hãy viết một hàm thực hiện một trong hai thuật toán trên và tính xác suất để một đứa trẻ được sinh ra trong tuần 39, khi biết rằng nó không chào đời trước tuần 39.
>
> Hãy khái quát hóa hàm trên để tính xác suất để một đứa trẻ sẽ được sinh vào tuần *x*, khi biết rằng nó không ra đời trước tuần *x*, với mọi giá trị có thể của *x*. Hãy vẽ đồ thị của giá trị xác suất này theo *x* đối với trẻ đầu lòng và trẻ sinh sau.
>
> Bạn có thể tải về một lời giải của bài toán này từ `http://thinkstats.com/conditional.py`.

Báo cáo kết quả

Đến lúc này ta đã khảo sát xong số liệu và thấy được một số hiệu ứng biểu kiến. Hiện giờ, hãy tạm giả thiết rằng những hiệu ứng này là thật (nhớ rằng đây mới chỉ là giả thiết). Làm thế nào để ta báo cáo những kết quả này?

Câu trả lời có thể tùy thuộc vào người hỏi. Chẳng hạn, một nhà khoa học có thể quan tâm đến bất kì một hiệu ứng (thật) nào, bất kể nó nhỏ đến bao nhiêu. Một bác sĩ có thể chỉ quan tâm đến các hiệu ứng **có ý nghĩa về y học**; tức là những khác biệt có ảnh hưởng đến quyết định điều trị. Một phụ nữ mang thai có thể quan tâm đến những kết quả có liên quan tới cô ấy, như các xác suất có điều kiện trong mục trước. Cách mà bạn báo cáo kết quả cũng phụ thuộc vào mục tiêu của bạn. Nếu phải biểu diễn ý nghĩa của một hiệu ứng, bạn có thể chọn đặc trưng thống kê, như rủi ro tương đối, để nhấn mạnh sự khác biệt. Nếu bạn cần trấn an bệnh nhân, bạn có thể chọn số thống kê có tính đến sự khác biệt.

> Dựa vào các kết quả từ những bài tập trước, chẳng hạn bạn được yêu cầu phải tóm tắt những gì bạn đã biết về vấn đề liệu trẻ đầu lòng có chào đời muộn hay không.
>
> Bạn sẽ dùng đặc trưng thống kê nào nếu muốn đưa vào chuyện kể trong bản tin buổi tối? Bạn sẽ dùng đặc trưng nào nếu muốn an ủi một bệnh nhân đang lo lắng? Sau cùng, hãy tưởng tượng rằng mình là Cecil Adams, tác giả của *The Straight Dope* (`http://straightdope.com`), và nhiệm vụ của bạn là trả lời câu hỏi, “Liệu trẻ đầu lòng có chào đời muộn không?” Hãy viết bài trả lời dựa trên những kết quả trong chương này để giải đáp câu hỏi một cách rõ ràng, tập trung, và chính xác.

Thuật ngữ

xu thế trung tâm:
: Đặc tính của một mẫu hoặc tổng thể; theo trực giác, đó là giá trị trung bình nhất.

phân tán:
: Đặc tính của một mẫu hoặc tổng thể; theo trực giác, nó mô tả độ biến động là bao nhiêu.
phương sai:
: Đặc trưng thống kê thường được dùng để lượng hóa mức phân tán.
độ lệch chuẩn:
: Căn bậc hai của phương sai, cũng được dùng để đo mức phân tán.
tần số:
: Số lần mà một giá trị xuất hiện trong mẫu.
biểu đồ tần số:
: Một ánh xạ từ giá trị đến tần số, hoặc một biểu đồ thể hiện ánh xạ này.
tần suất:
: Tần số được biểu thị dưới dạng tỉ lệ của kích thước mẫu.
chuẩn hóa:
: Việc chia tần số cho kích thước mẫu để thu được tần suất.
phân bố:
: Dạng tóm tắt các giá trị xuất hiện trong một mẫu cùng với tần số, hay tần suất, của mỗi giá trị.
PMF:
: Hàm khối xác suất (probability mass function): cách biểu diễn một phân bố dưới dạng hàm ánh xạ từ giá trị đến tần suất.
số đông:
: Giá trị hay gặp nhất trong mẫu.
điểm biệt lập:
: Giá trị nằm cách xa xu thế trung tâm.
tỉa bớt:
: Xóa bỏ những điểm biệt lập khỏi tập số liệu.
ngăn:
: Một khoảng dùng để nhóm các giá trị gần bằng nhau.
rủi ro tương đối:
: Tỉ số giữa hai tần suất, thường được dùng để đo độ khác biệt giữa hai phân bố.
xác suất có điều kiện:
: Xác suất được tính theo giả thiết rằng một điều kiện nào đó phải được đảm bảo.
có ý nghĩa về y học:
: Kết quả, chẳng hạn một khác biệt giữa hai nhóm, có liên quan trong thực tế ngành y.

Chương 3: Hàm phân bố lũy tích

Trở về Muc luc cuốn sách

Nghịch lý về số sinh viên trong lớp

Trong nhiều trường đại học Hoa Kỳ, tỉ lệ sinh viên so với giảng viên đều vào khoảng 10:1. Nhưng sinh viên thường ngạc nhiên khi biết rằng lớp học của họ trung bình đều hơn 10 sinh viên. Có hai lý do giải thích sự khác biệt này:

- Sinh viên nói chung đều học 4–5 lớp mỗi kỳ, trong khi giáo sư thường chỉ dạy 1 hoặc 2 lớp.
- Số sinh viên may mắn được học lớp vắng thì ít, còn sinh viên phải học lớp lớn thì (a hèm!) có rất nhiều.

Điều thứ nhất thì rất dễ thấy (nó đã được nhắc đến ít nhất một lần); điều thứ hai thì tinh vi hơn. Vì vậy ta hãy xét một ví dụ. Chẳng hạn một trường có 65 lớp học trong một kì, với phân bố số học viên mỗi lớp như sau:

số SV	số lớp
5- 9	8
10-14	8
15-19	14
20-24	4
25-29	6
30-34	12
35-39	8
40-44	3
45-49	2

Nếu bạn hỏi hiệu trưởng về số sinh viên trung bình trong một lớp, ông ta sẽ dựng một đường PMF, tính trung bình, và báo rằng số sinh viên trung bình trong mỗi lớp bằng 24.

Nhưng nếu bạn điều tra một nhóm sinh viên, hỏi xem có bao nhiêu sinh viên trong lớp họ, rồi tính trung bình, bạn sẽ nghĩ rằng số sinh viên trung bình sẽ cao hơn.

> Hãy dựng đường PMF từ những số liệu trên rồi tính trung bình theo cách nhìn nhận của hiệu trưởng. Vì số liệu đã được chia vào các ngăn, bạn có thể dùng một điểm giữa cho mỗi ngăn.
>
> Bây giờ hãy tính phân bố của số sinh viên từng lớp theo cách nhìn nhận của sinh viên rồi tính trị trung bình.
>
> Giả sử rằng bạn muốn tìm phân bố của số sinh viên từng lớp của một trường, nhưng không thể có số liệu chính xác từ hiệu trưởng. Một cách khác là chọn một mẫu sinh viên ngẫu nhiên rồi hỏi họ về số sinh viên từng lớp mà họ theo học. Sau đó bạn có thể tính PMF từ số liệu thu được qua trả lời của họ.
>
> Kết quả sẽ thiên lệch vì những lớp học lớn sẽ được phản ánh lặp lại trong mẫu, nhưng bạn vẫn có thể ước tính được phân bố thực sự của số sinh viên mỗi lớp bằng cách áp dụng một phép chuyển đổi thích hợp đối với dạng phân bố đã quan sát thấy.

Hãy viết một hàm có tên `UnbiasPmf` để nhận vào PMF của các giá trị quan sát được và trả lại một đối tượng Pmf mới để ước tính phân bố của số sinh viên mỗi lớp.

Bạn có thể tải về một lời giải cho bài này từ http://thinkstats.com/class_size.py.

Trong đa số các cuộc thi chạy, mọi người đều xuất phát cùng lúc. Nếu bạn chạy nhanh, thường bạn sẽ vượt được nhiều người ngay đầu cuộc đua, nhưng rồi sau một vài dặm đường, mọi người xung quanh bạn sẽ chạy với cùng tốc độ.

Khi lần đầu tiên tham gia cuộc chạy tiếp sức đường dài (209 dặm), tôi nhận thấy một hiện tượng lạ: khi mình vượt một đối thủ khác, tôi thường nhanh hơn rất nhiều, và khi một đối thủ khác vượt tôi, người ta cũng thương nhanh hơn tôi rất nhiều.

Ban đầu tôi cứ nghĩ rằng phân bố tốc độ có lẽ sẽ chứa hai số đông; nghĩa là có nhiều người chạy chậm và nhiều người chạy nhanh, còn ít người chạy cùng tốc độ với tôi.

Nhưng sau đó tôi nhận thấy rằng đã bị thiên lệch trong việc lựa chọn. Cuộc đua khác thường ở hai điểm: nó xuất phát không đều, các đội xuất phát vào các thời điểm khác nhau; ngoài ra nhiều đội gồm những người có sức chạy không đều nhau.

Kết quả là, các vận động viên sẽ tản ra trong cuộc đua với ít mối tương quan giữa tốc độ và vị trí. Khi tôi bắt đầu phiên chạy, những người chạy gần tôi đều (khá giống) một mẫu ngẫu nhiên của tổng thể các vận động viên trong cuộc đua.

Vậy thì sự thiên lệch này từ đâu mà có? Trong khi tôi chạy thi, cơ may vượt một đối thủ khác, hoặc bị vượt, đều tỉ lệ với hiệu số tốc độ giữa hai người. Để thấy được tại sao, bạn hãy hình dung về hai giới hạn như sau. Nếu có ai đó chạy nhanh bằng tôi thì chúng tôi không vượt được nhau. Nếu ai đó nhanh đến mức có thể chạy hết chặng đường đua trong khi tôi vẫn đang chạy thì chắc chắn họ vượt được tôi.

Hãy viết một hàm tên là `BiasPmf` để nhận vào một Pmf biểu thị cho phân bố thực sự của tốc độ người chạy, và một tốc độ của một người quan sát đang chạy, rồi trả về một Pmf mới biểu thị cho phân bố tốc độ tương đối của những người chạy so với người quan sát.

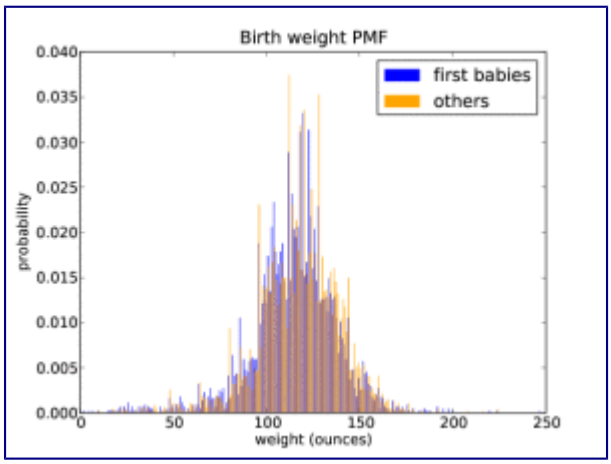
Để kiểm tra hàm vừa viết, hãy lấy một phân bố các tốc độ trong một cuộc đua bình thường (không phải là tiếp sức). Tôi đã viết một chương trình có nhiệm vụ đọc vào kết quả cuộc đua James Joyce Ramble 10K ở Dedham MA rồi chuyển đổi tốc độ từng người chạy sang dặm/giờ. Hãy tải về chương trình này từ http://thinkstats.com/relay.py. Chạy nó rồi xem PMF các tốc độ.

Bây giờ hãy tính phân bố các tốc độ mà bạn sẽ quan sát thấy nếu bạn chạy trong cuộc thi tiếp sức với tốc độ 7,5 dặm/giờ cùng nhóm người chạy nói trên. Bạn có thể tải về một lời giải từ http://thinkstats.com/relay_soln.py

Sự hạn chế của các PMF

PMF sẽ phát huy tác dụng nếu có ít các giá trị. Nhưng khi số các giá trị nhiều lên, xác suất ứng với mỗi giá trị sẽ nhỏ đi và hiệu ứng của nhiễu ngẫu nhiên sẽ tăng lên.

Chẳng hạn, chúng ta có thể quan tâm đến phân bố cân nặng trẻ sơ sinh. Trong số liệu NSFG, biến `totalwgt_oz` ghi lại các cân nặng trẻ sơ sinh tính theo ounce. Hình dưới đây cho thấy PMF của các giá trị này cho trẻ đầu lòng và các trẻ sinh sau.



PMF của cân nặng trẻ sơ sinh. Hình vẽ cho thấy một hạn chế của các PMF: chúng rất khó so sánh với nhau. (nsfg_birthwgt_pm)

Nói chung, các phân bố này đều trông giống hình "quả chuông" quen thuộc, với nhiều giá trị gần số trung bình và ít giá trị quá lớn hay quá nhỏ.

Nhưng có những phần của hình vẽ này rất khó diễn giải. Có nhiều đỉnh nhọn và chỗ trũng, và một số khác biệt trông thấy giữa các phân bố. Thật khó nói được những đặc điểm nào trong số này là có ý nghĩa. Đồng thời, cũng khó nhận thấy những đặc điểm tổng thể; chẳng hạn, bạn nghĩ rằng phân bố nào có trị trung bình cao hơn?

Vấn đề này có thể được giảm nhẹ bằng việc chia số liệu vào các ngăn; nghĩa là chia miền giá trị thành những khoảng không trùng nhau và đếm số giá trị rơi vào mỗi ngăn. Việc chia ngăn có thể có ích, nhưng để chọn kích thước ngăn thích hợp sẽ rất mẹo mực. Nếu ngăn đủ lớn để làm trơn các nhiễu động thì nó cũng có thể làm trơn những thông tin có ích.

Một cách khác để tránh vấn đề trên là dùng **hàm phân bố lũy tích**, hay **CDF** (cumulative distribution function). Nhưng trước khi làm được điều này, ta phải nói về số phần trăm.

Số phần trăm

Nếu bạn đã từng làm một bài thi tiêu chuẩn, có thể bạn sẽ được chấm điểm dưới dạng điểm gốc kèm theo một **hạng phần trăm**. Ở đây, hạng phần trăm là tỉ lệ số người có điểm thấp hơn bạn (hoặc bằng). Vì vậy, nếu bạn đứng "hạng 90 phần trăm", thì bạn đã làm bài tốt bằng hoặc hơn 90% số người đã làm bài thi.

Sau đây là cách để bạn có thể tính hạng phần trăm của một giá trị có tên `your_score`, so với các giá trị trong dãy `scores`:

```
def PercentileRank(scores, your_score):
    count = 0
    for score in scores:
        if score <= your_score:
            count += 1
```

```
    percentile_rank = 100.0 * count / len(scores)
    return percentile_rank
```

Chẳng hạn, nếu các giá trị trong dãy là 55, 66, 77, 88 và 99, bạn đạt điểm 88, thì hạng phần trăm của bạn sẽ là `100 * 4 / 5` tức là bằng 80.

Nếu cho trước một giá trị, thì sẽ tìm được hạng phần trăm của nó; còn làm ngược lại thì hơi khó hơn. Nếu bạn được cho một hạng phần trăm và muốn tìm giá trị tương ứng, một cách làm là sắp xếp các giá trị và tìm lấy giá trị mà bạn muốn:

```
def Percentile(scores, percentile_rank):
    scores.sort()
    for score in scores:
        if PercentileRank(scores, score) >= percentile_rank:
            return score
```

Kết quả của phép tính này là một **số phần trăm**. Chẳng hạn, số phần trăm thứ 50 là giá trị ứng với hạng phần trăm 50. Trong phân bố điểm thi, số phần trăm thứ 50 là 77.

> Đoạn chương trình thực hiện `Percentile` như trên không được hiệu quả. Một cách hay hơn là dùng hạng phần trăm để tính chỉ số của số phần trăm tương ứng. Hãy viết một phiên bản `Percentile` có dùng thuật toán này.
>
> Bạn có thể tải về một lời giải từ http://thinkstats.com/score_example.py.
>
> Tùy chọn: Nếu bạn chỉ muốn tính một số phần trăm, thì cách sắp xếp các điểm thi sẽ không hiệu quả. Một cách làm hay hơn là thuật toán lựa chọn, mà bạn có thể đọc từ http://wikipedia.org/wiki/Selection_algorithm.
>
> Hãy viết (hoặc tìm) một đoạn chương trình thực hiện thuật toán lựa chọn và dùng nó để viết một phiên bản hiệu quả cho `Percentile`.

Hàm phân bố lũy tích

Bây giờ khi đã hiểu được số phần trăm, chúng ta sẵn sàng xét đến hàm phân bố lũy tích (cumulative distribution function, CDF). CDF là một hàm với ánh xạ từ các giá trị đến hạng phần trăm của chúng trong một phân bố.

CDF là một hàm theo *x*, trong đó *x* là một giá trị bất kì có thể xuất hiện trong phân bố. Để tính CDF(*x*) cho một giá trị cụ thể của *x*, ta cần tính tỉ lệ của các giá trị trong mẫu mà nhỏ hơn (hoặc bằng) *x*.

Sau đây là một hàm như vậy, nhận vào một mẫu `t`, và một giá trị, `x`:

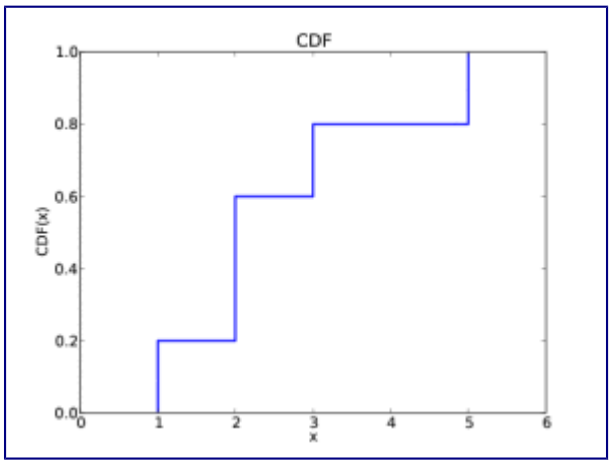
```
def Cdf(t, x):
    count = 0.0
    for value in t:
        if value <= x:
            count += 1.0

    prob = count / len(t)
    return prob
```

Hàm này sẽ trông giống; nó gần như y hệt `PercentileRank`, chỉ khác là kết quả lại là một tần suất nằm trong khoảng từ 0–1 thay vì một hạng phần trăm trong khoảng từ 0–100.

Lấy ví dụ, chẳng hạn có một mẫu gồm các giá trị sau {1, 2, 2, 3, 5}. Dưới đây là một số giá trị lấy từ CDF của nó:

CDF(0) = 0
CDF(1) = 0.2
CDF(2) = 0.6
CDF(3) = 0.8
CDF(4) = 0.8
CDF(5) = 1

Ta có thể lượng giá CDF cho bất kì giá trị *x* bất kì, chứ không chỉ các giá trị có xuất hiện trong mẫu. Nếu *x* nhỏ hơn giá trị nhỏ nhất trong mẫu, CDF(*x*) bằng 0. Nếu *x* lớn hơn giá trị lớn nhất trong mẫu, CDF(*x*) bằng 1.



Ví dụ về một CDF.

Hình trên biểu diễn cho CDF này. CDF của một mẫu là một hàm bậc thang. Trong chương tiếp theo ta sẽ thấy các phân bố có CDF là các hàm liên tục.

Biểu diễn CDF

Tôi đã viết một module có tên `Cdf` trong đó có một lớp tên là `Cdf` để biểu diễn các CDF. Bạn có thể đọc tài liệu hướng dẫn cách dùng module này tại http://thinkstats.com/Cdf.html và tải module về từ http://thinkstats.com/Cdf.py.

Cdf được viết trong chương trình với hai danh sách đã sắp xếp: `xs` để chứa các giá trị, và `ps` để chứa các tần suất. Những phương thức quan trọng nhất mà Cdf cung cấp là:

`Prob(x)`:
: Với giá trị *x* cho trước, tính xác suất *p* = CDF(*x*).

`Value(p)`:
: Với xác suất *p* cho trước, tính giá trị *x* tương ứng; nghĩa là CDF nghịch đảo của *p*.

Vì `xs` và `ps` đều đã được sắp xếp nên các phương thức trên có thể dùng thuật toán phân đôi, vốn rất hiệu quả. Thời gian chạy tỉ lệ với loga của số các giá trị; xem http://wikipedia.org/wiki/Time_complexity.

Cdf cũng cung cấp `Render`, vốn trả lại 2 danh sách, `xs` và `ps`, phù hợp để vẽ CDF. Vì CDF là một hàm bậc thang nên các danh sách này có hai phần tử ứng với mỗi giá trị duy nhất trong phân bố.

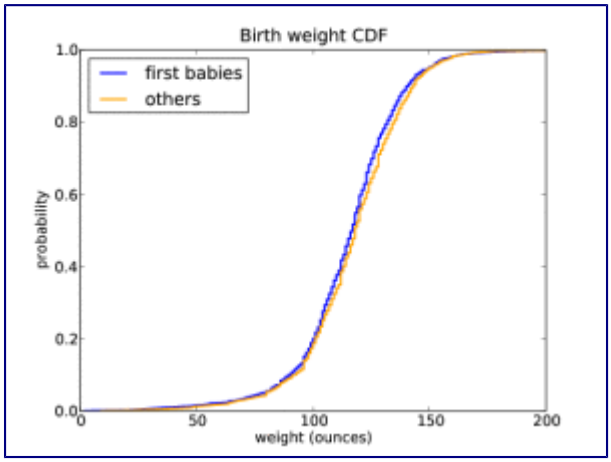
Module Cdf cung cấp một số hàm để tạo ra các Cdf, bao gồm `MakeCdfFromList`, vốn nhận vào một dãy các giá trị rồi trả lại Cdf của chúng.

Sau cùng, `myplot.py` có các hàm tên là `Cdf` và `Cdfs` để vẽ đồ thị các Cdf dưới dạng đường.

> Hãy tải về `Cdf.py` và `relay.py` (xem Bài tập relay) rồi phát sinh một đồ thị biểu diễn CDF của các tốc độ chạy. Cách nào cho bạn hình dung tốt hơn hình dạng của phân bố, PMF hay CDF? Bạn có thể tải về một lời giải từ http://thinkstats.com/relay_cdf.py.

Quay trở về số liệu điều tra

Hình dưới đây biểu diễn các CDF cân nặng trẻ sơ sinh đối với trẻ đầu lòng và trẻ sinh sau trong bộ số liệu NSFG.



CDF của cân nặng trẻ sơ sinh. (nsfg_birthwgt_cdf)

Hình vẽ này làm cho hình dạng của các phân bố, và sự khác biệt giữa chúng, trở nên rõ ràng hơn nhiều. Ta có thể thấy được trên suốt phân bố, trẻ đầu lòng luôn nhẹ hơn một chút, sự khác biệt lớn hơn ở trên giá trị trung bình.

> Lúc mới sinh bạn nặng bao nhiêu? Nếu bạn không biết, hãy hỏi mẹ hoặc một người khác biết thông tin này. Dùng kho số liệu (tất cả ca sinh thành công) để tính phân bố cân nặng trẻ sơ sinh rồi từ đó tính hạng phần trăm của bạn. Nếu bạn là con đầu, hãy tính hạng phần trăm trong phân bố dành cho trẻ đầu lòng. Còn nếu không thì dùng dạng phân bố của trẻ sinh sau. Sự khác biệt giữa hai hạng phần trăm của cân nặng bạn giữa hai phân bố là bao nhiêu?
>
> Giả sử là bạn và những người cùng lớp đều tính hạng phần trăm của cân nặng sơ sinh mỗi người rồi sau đó tính CDF của các hạng phần trăm này. Bạn sẽ chờ đợi kết quả CDF có dạng như thế nào? Gợi ý: bạn nghĩ rằng tỉ lệ bao nhiêu trong lớp sẽ cao hơn số trung vị?

Phân bố theo điều kiện

Phân bố theo điều kiện là phân bố của một tập con số liệu được chọn lựa theo một điều kiện nào đó.

Chẳng hạn, nếu bạn nặng hơn mức trung bình, nhưng cao hơn nhiều so với mức trung bình, thì bạn sẽ tương đối nhẹ so với chiều cao hiện có. Sau đây là cách làm cho lời khẳng định đó chuẩn xác hơn.

1. Chọn ra một nhóm người có cùng chiều cao với bạn (hiểu theo nghĩa xấp xỉ trong khoảng nào đó).
2. Tìm CDF về cân nặng của những người đó.
3. Tính hạng phần trăm cân nặng của bạn trong phân bố này.

Hạng phần trăm rất có ích trong việc so sánh những kết quả đo từ các phép thử khác nhau, hoặc những phép thử đối với các nhóm khác nhau.

Chẳng hạn, những người thi chạy thường được nhóm theo tuổi tác và giới tính. Để so sánh những người trong các nhóm khác nhau, bạn có thể chuyển đổi thời gian chạy về hạng phần trăm.

> Gần đây tôi có thi chạy giải James Joyce Ramble 10K ở Dedham MA. Kết quả được đăng trên `http://coolrunning.com/results/10/ma/Apr25_27thAn_set1.shtml`. Hãy đến trang đó và tìm thời gian chạy của tôi. Tôi xếp thứ 97 trong tất cả là 1633 người tham gia, vậy hạng phần trăm của tôi là bao nhiêu?
>
> Ở nhóm độ tuổi của mình (M4049 có nghĩa là "nam giới từ 40 đến 49 tuổi"), tôi xếp thứ 26 trong tổng số 256. Hạng phần trăm của tôi là bao nhiêu trong nhóm này?
>
> Nếu tôi vẫn chạy trong 10 năm tới (và tôi hi vọng là vẫn chạy được), tôi sẽ được xếp vào nhóm M5059. Giả dụ rằng hạng phần trăm của tôi trong nhóm vẫn không đổi, thì tôi sẽ chạy chậm đi bao nhiêu?
>
> Tôi duy trì ganh đua lành mạnh với một cô sinh viên trong nhóm F2039. Cô ấy sẽ phải chạy nhanh mức nào trong cuộc thi 10K sắp tới để vượt tôi về hạng phần trăm?

Số ngẫu nhiên

CDF rất có ích trong việc phát sinh ra các số ngẫu nhiên từ một phân bố cho trước. Sau đây là cách làm:

- Chọn một xác suất ngẫu nhiên trong khoảng từ 0–1.
- Dùng `Cdf.Value` để tìm trong phân bố giá trị tương ứng với xác suất đã chọn.

Có thể không hiển nhiên là tại sao cách này lại cho kết quả, nhưng vì sẽ dễ thực hiện hơn là giải thích, nên ta hãy làm thử.

> Hãy viết một hàm có tên `Sample`, để nhận vào một Cdf và một số nguyên, , rồi trả lại một danh sách giá trị chọn ngẫu nhiên từ Cdf đó. Gợi ý: hãy dùng `random.random`. Bạn sẽ tìm thấy một lời giải bài này ở `Cdf.py`.

> Dùng phân bố cân nặng trẻ sơ sinh từ NSFG, hãy phát sinh một mẫu ngẫu nhiên gồm 1000 phần tử. Tính CDF của mẫu đó. Vẽ biểu đồ cho thấy cả CDF ban đầu và CDF của mẫu ngẫu nhiên. Với các giá lớn, hai phân bố sẽ phải giống nhau.

Quá trình này, phát sinh một mẫu ngẫu nhiên dựa trên một mẫu đo đạc, được gọi là **lấy mẫu lại**.

Có hai cách lấy mẫu từ một tổng thể: có thay thế và không thay thế. Nếu bạn hình dung lấy những viên bi từ một lọ[1], "có thay thế" nghĩa là sau khi chọn bi lại bỏ vào lọ (rồi trộn lẫn), để cho tổng thể không thay đổi sau mỗi lần lấy bi. "Không thay thế" nghĩa là mỗi viên bi chỉ có thể lấy ra một lần, vì vậy tổng thể còn lại sẽ khác đi sau mỗi lần lấy bi.

Trong Python, việc lấy mẫu có thay thế được thực hiện bằng `random.random` để chọn một hạng phần trăm, hoặc `random.choice` để chọn một phần tử từ một dãy. Việc lấy mẫu không thay thế được thực hiện với `random.sample`.

> Những số được phát sinh bởi `random.random` được thiết kế để phân bố đều trong khoảng từ 0 đến 1; nghĩa là mỗi giá trị trong khoảng cần phải có cùng xác suất.
>
> Hãy phát sinh 1000 số bằng `random.random` rồi vẽ các đồ thị PMF và CDF của các số này. Bạn có thể nói rằng chúng phân bố đều không?
>
> Bạn có thể tìm hiểu về phân bố đều ở `http://wikipedia.org/wiki/Uniform_distribution_(discrete)`.

Quay về đặc trưng thống kê

Một khi bạn đã tính được CDF, sẽ dễ dàng tính được các đặc trưng thống kê khác. Số trung vị chính là số phần trăm thứ 50[2]. Các số phần trăm thứ 25 và 75 thường được dùng để kiểm tra xem liệu một phân bố có đối xứng không, và hiệu số giữa chúng, vốn được gọi là **khoảng tứ phân vị**, để đo độ phân tán.

> Hãy viết một hàm có tên `Median` nhận vào một Cdf rồi tính số trung vị, và một hàm có tên `Interquartile` để tính khoảng tứ phân vị.
>
> Tính các số phần trăm thứ 25, 50, và 75 từ CDF cân nặng trẻ sơ sinh. Những giá trị này có dấu hiệu cho thấy rằng phân bố có tính đối xứng không?

Thuật ngữ

hạng phần trăm:
: Số phần trăm của những giá trị trong một phân bố nhỏ hơn hoặc bằng một giá trị cho trước.

CDF:
: Hàm phân bố lũy tích, một hàm ánh xạ từ các giá trị đến hạng phần trăm của chúng.

số phần trăm:
: Giá trị ứng với một hạng phần trăm cho trước.

phân bố theo điều kiện:
: Phân bố được tính dựa trên giải thiết rằng một điều kiện nào đó được thỏa mãn.

lấy mẫu lại:
: Quá trình phát sinh một mẫu ngẫu nhiên từ một phân bố đã được tính từ một mẫu.

thay thế:

Trong quá trình lấy mẫu, “có thay thế” để chỉ rằng tổng thể vẫn giữ nguyên giữa các lần lấy mẫu. “Không thay thế” để chỉ rằng mỗi phần tử chỉ có thể được chọn một lần.

khoảng tứ phân vị:
Độ đo của sự phân tán, tính bằng hiệu số giữa các số phần trăm thứ 75 và 25.

---

1. “Bi trong lọ” là một mô hình chuẩn cho các quá trình lấy mẫu ngẫu nhiên (xem http://wikipedia.org/wiki/Urn_problem). ↩
2. Bạn có thể đã thấy các định nghĩa khác về số trung vị. Đặc biệt, có tài liệu nói rằng nếu bạn có mẫu chứa một số chẵn các phần tử thì trung vị là trung bình cộng của hai phần tử trung tâm. Trường hợp đặc biệt này thật không cần thiết, và nó có một hiệu ứng kì quặc là phát sinh ra một giá trị không nằm trong mẫu. Đến giờ tôi chỉ cần biết rằng trung vị là số phần trăm thứ 50. Chấm hết. ↩

Chương 4: Phân bố liên tục

Trở về Muc luc quyển sách

Những phân bố mà chúng ta đã gặp cho đến giờ được gọi là **phân bố kinh nghiệm** vì chúng được dựa trên những quan sát kinh nghiệm, vốn là các mẫu có kích thước giới hạn.

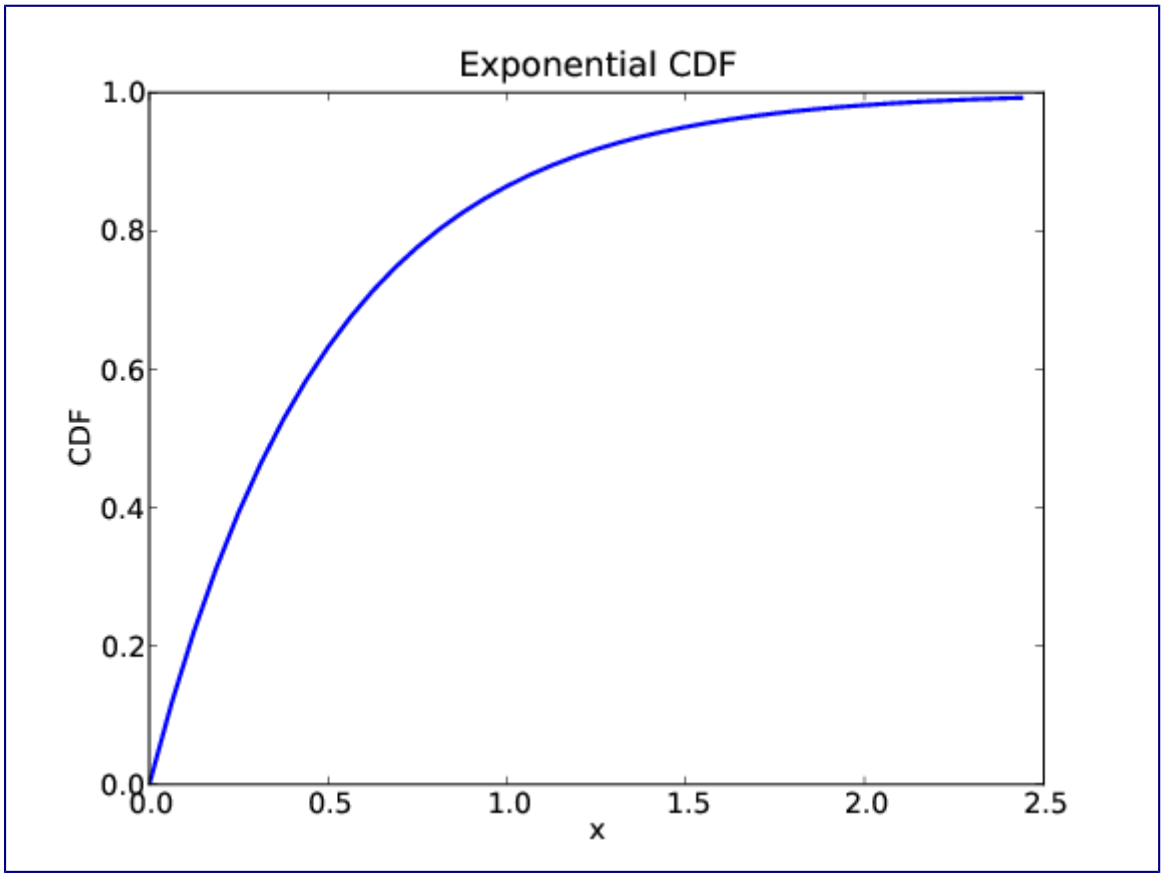
Một phân bố khác là **phân bố liên tục**, vốn được đặc trưng bởi một CDF dưới dạng hàm liên tục (thay vì hàm bậc thang). Nhiều hiện tượng thực tế có thể được xấp xỉ bằng những phân bố liên tục.

Phân bố lũy thừa

Tôi sẽ bắt đầu với phân bố lũy thừa vì giải thích nó rất dễ. Trong thực tế, các phân bố lũy thừa thường được bắt gặp khi ta quan sát một chuỗi các hiện tượng và đo khoảng thời gian giữa hai hiện tượng kế tiếp, mà chúng ta gọi là **thời gian giữa hai sự kiện**. Nếu các sự kiện có vẻ như xảy ra được bất kì lúc nào thì phân bố giữa khoảng thời gian liên tiếp này sẽ có xu hướng tuân theo một phân bố lũy thừa. CDF của một phân bố lũy thừa là:

$\mathrm{CDF}(x) = 1 - e^{-\lambda x}$

Thông số λ quyết định hình dạng của phân bố. Hình dưới đây cho thấy CDF này trông ra sao với λ = 2.



CDF của phân bố lũy thừa. (expo_cdf)

Nói chung, giá trị trung bình của phân bố lũy thừa bằng 1/ λ, nên trị trung bình của phân bố này bằng 0,5. Số trung vị thì bằng log(2)/λ, vốn xấp xỉ bằng 0,35. Để xem ví dụ về một phân bố có dạng

xấp xỉ lũy thừa, ta sẽ xét khoảng thời gian giữa hai đứa trẻ chào đời liên tiếp. Vào ngày 18/12/1997, có 44 trẻ được sinh ra ở một bệnh viện tại Brisbane, Úc.[1] Thời điểm chào đời của cả 44 đứa bé được đăng trên một tờ báo địa phương; bạn có thể tải dữ liệu về từ `http://thinkstats.com/babyboom.dat`. Hình {interarrival_cdf} cho thấy CDF của khoảng thời gian, tính theo phút, giữa các ca sinh. Dường như nó có hình dạng tựa như một phân bố lũy thừa, nhưng làm sao ta cho thấy được điều này? Một cách làm và vẽ đồ thị của hàm bù CDF, 1 - CDF(x), theo trục tỉ lệ log-y. Với số liệu của một hàm lũy thừa, kết quả sẽ là một đường thẳng. Hãy xem bằng cách nào mà ta có điều đó.

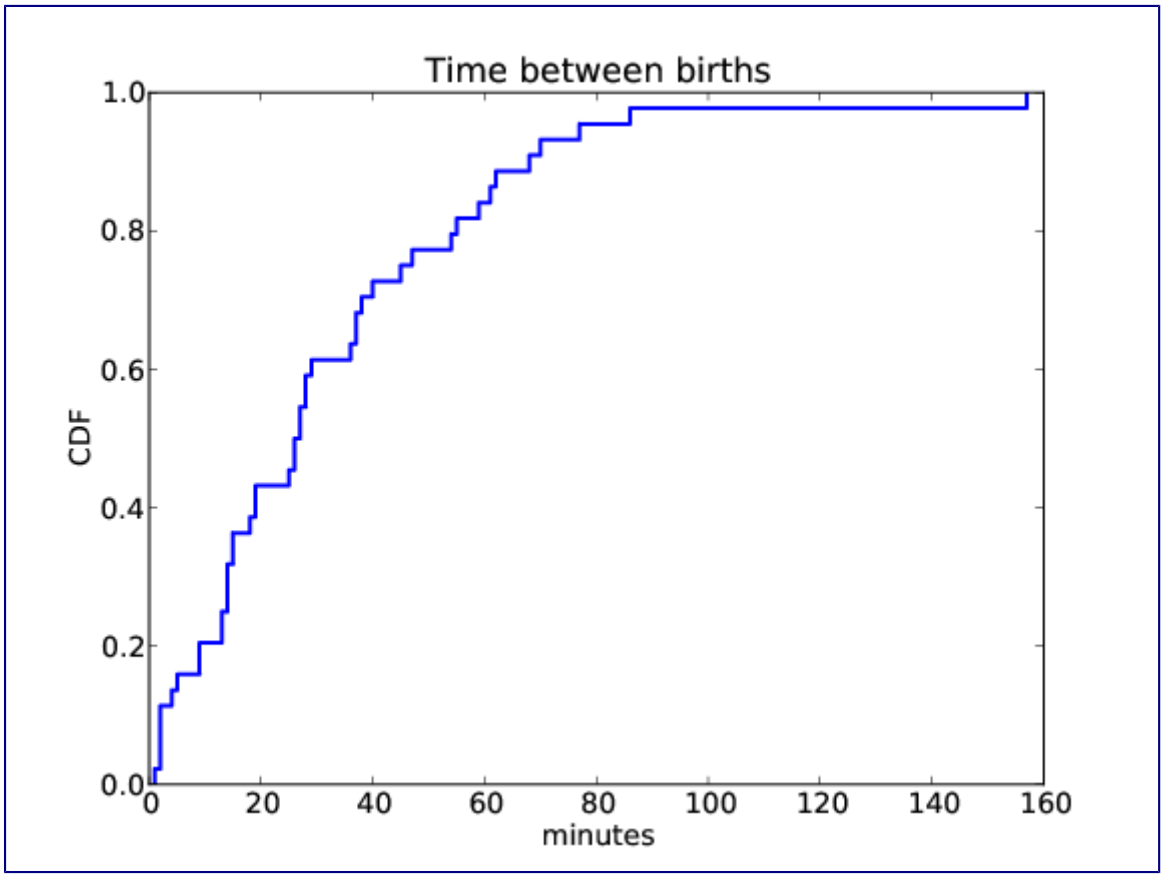
Nếu bạn vẽ hàm bù CDF (tức CCDF) của một bộ số liệu mà bạn cho rằng có phân bố lũy thừa, bạn sẽ trông đợi một hàm như:

$y \approx e^{-\lambda x}$

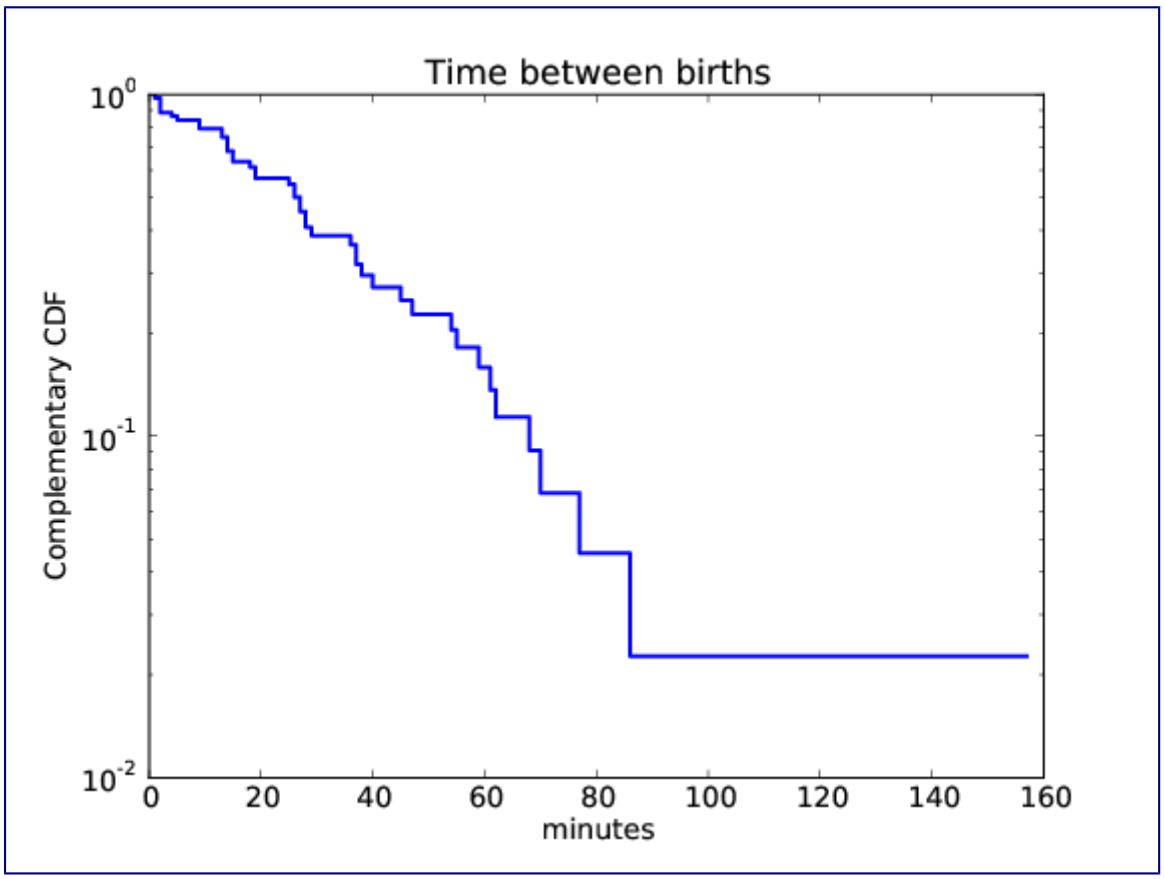
Lấy loga cả 2 vế ta được:

$$\log y \approx -\lambda x$$

Vì vậy theo trục log-y hàm CCDF là một đường thẳng với độ dốc $-\lambda$.



CDF của các thời gian giữa hai sự kiện (interarrival_cdf)



CCDF của các thời gian giữa hai sự kiện (interarrival_ccdf)

Hình {interarrival_ccdf} cho thấy CCDF của thời gian giữa hai ca sinh nở theo trục log-*y*. Nó không thẳng hoàn toàn, nghĩa là phân bố lũy thừa chỉ là một cách xấp xỉ. Trong phần lớn trường hợp thì giả thiết nền tảng ở đây—việc sinh nở có khả năng xảy ra như nhau bất kể giờ trong ngày—là không hoàn toàn đúng.

Với những giá trị *n* nhỏ, ta không trông đợi một phân bố kinh nghiệm khớp đúng với phân bố liên tục. Một cách đánh giá chất lượng khớp là phát sinh một mẫu từ phân bố liên tục và xem nó hợp với số liệu đến mức nào. Hàm `expovariate` trong module `random` sẽ phát sinh các giá trị ngẫu nhiên từ một phân bố lũy thừa, khi cho trước giá trị của λ. Hãy dùng nó để phát sinh 44 giá trị từ một phân bố lũy thừa với trị trung bình bằng 32,6. Vẽ CCDF lên trục tỉ lệ log-*y* và so sánh nó với Hình {interarrival_ccdf}. Gợi ý: Bạn có thể dùng hàm `pyplot.yscale` để dựng trục *y* theo thang loga.

Hoặc, nếu bạn dùng `myplot`, hàm `Cdf` sẽ nhận một tùy chọn kiểu boolean có tên `complement`, để quy định xem cần phải vẽ đồ thị CDF hay CCDF, và các tùy chọn kiểu chuỗi, `xscale` và `yscale`, để chuyển đổi các trục; từ đó có thể vẽ CCDF theo trục tỉ lệ log-*y*:

```
myplot.Cdf(cdf, complement=True, xscale='linear', yscale='log')
```

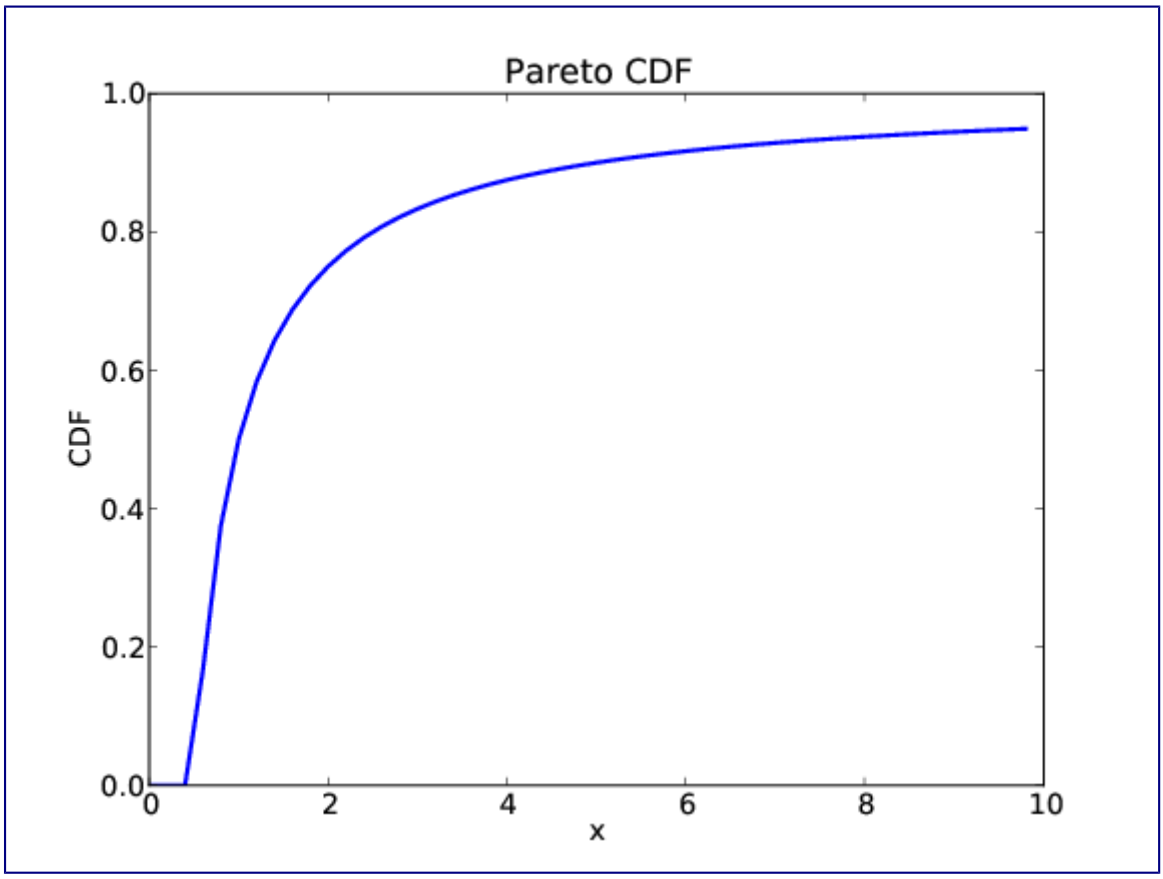
Hãy thu thập ngày sinh của các sinh viên cùng lớp, sắp xếp và tính khoảng thời gian theo ngày giữa các ngày sinh. Vẽ đồ thị CDF của các thời gian giữa này, cùng CCDF theo trục tỉ lệ log-*y*. Liệu nó có trông giống một phân bố lũy thừa không?

Phân bố Pareto

Phân bố Pareto được đặt tên theo nhà kinh tế học Vilfredo Pareto, người đã dùng nó để mô tả sự phân bố mức giàu nghèo (xem http://wikipedia.org/wiki/Pareto_distribution). Từ đó, nó được dùng để mô tả những hiện tượng trong khoa học tự nhiên và xã hội bao gồm quy mô các thành phố, kích cỡ các hạt cát hay thiên thạch, quy mô cháy rừng và động đất. CDF của phân bố Pareto là:

$\text{CDF}(x) = 1 - (x / x_m)^{-\alpha}$

Các tham số x_m và α quyết định vị trí và hình dạng của phân bố. x_m là giá trị khả dĩ nhỏ nhất. Hình {pareto_cdf} cho thấy CDF của một phân bố Pareto với các tham số $x_m = 0{,}5$ và $\alpha = 1$.



CDF của một phân bố Pareto. (pareto_cdf)

Số trung vị của phân bố này là $x_m 2^{1/\alpha}$, tức là bằng 1, nhưng số phần trăm thứ 95 lại bằng 10. Thật khác với phân bố lũy thừa, với số trung vị bằng 1, thì số phần trăm thứ 95 chỉ bằng 1,5. Có một cách kiểm tra đơn giản bằng mắt thường để phát hiện rằng liệu một phân bố kinh nghiệm có khớp với một phân bố Pareto hay không: trên thang log-log, hàm CCDF sẽ trông như một đường thẳng. Nếu bạn vẽ đồ thị CCDF của một mẫu tuân theo phân bố Pareto lên một thang tuyến tính thì bạn sẽ trông đợi một hàm có dạng:

$y \approx (x / x_m)^{-\alpha}$

Lấy loga cả hai vế ta được:

$\log y \approx -\alpha (\log x - \log x_m)$

Vì vậy nếu bạn vẽ log *y* theo log *x*, nó sẽ có hình dạng như một đường thẳng với độ dốc - α và giao điểm - α log x_m với trục tung.

Module `random` có phương thức `paretovariate` để phát sinh các giá trị ngẫu nhiên từ một phân bố Pareto. Nó nhận vào một tham số α, nhưng không phải x_m. Giá trị mặc định cho x_m là 1; bạn có thể phát sinh một phân bố với tham số khác bằng cách nhân phân bố này với x_m. Hãy viết một hàm bọc có tên `paretovariate` nhận vào các tham số α và x_m rồi dùng `random.paretovariate` để phát sinh các giá trị từ một phân bố Pareto hai tham số. Hãy dùng hàm vừa viết để phát sinh một mẫu từ phân bố Pareto. Tính CCDF rồi vẽ đồ thị của nó lên thang log-log. Liệu đồ thị này có là đường thẳng không? Độ dốc của nó bằng bao nhiêu?

Để hình dung được phân bố Pareto, hãy tưởng tượng thế giới sẽ ra sao nếu như cân nặng của con người tuân theo phân bố Pareto. Chọn các tham số x_m = 100 cm và α = 1,7, ta thu được một phân bố với chiều cao tối thiểu hợp lý là 100 cm, và số trung vị 150 cm. Hãy phát sinh 6 tỉ giá trị ngẫu nhiên từ phân bố này. Giá trị trung bình của mẫu này bằng bao nhiêu. Có mấy phần của tổng thể với chiều cao dưới trị trung bình? Người cao nhất trong thế giới Pareto này sẽ cao bao nhiêu?

Định luật Zipf được là kết quả quan sát xem mức độ thường xuyên mà các từ khác nhau được dùng là bao nhiêu. Những từ thường dùng nhất thì có tần số rất cao, nhưng cũng có nhiều từ kì lạ, như “hapaxlegomenon,” chỉ xuất hiện một số ít lần. Định luật Zipf dự đoán rằng trong một văn bản, hay tác phẩm (“corpus”), sự phân bố các tần số từ vựng thì có dạng xấp xỉ Pareto. Hãy tìm một tác phẩm lớn dưới dạng điện tử, bằng bất kì ngôn ngữ nào. Hãy đếm xem mỗi từ xuất hiện bao nhiêu lần. Tính CCDF của số từ đếm được rồi vẽ đồ thị của nó theo thang tỉ lệ log-log. Liệu định luật Zipf có đúng trong trường hợp này không? Giá trị α xấp xỉ bằng bao nhiêu?

Phân bố Weibull là một dạng tổng quát của phân bố lũy thừa, xuất hiện trong phân tích rủi ro (xem http://wikipedia.org/wiki/Weibull_distribution). CDF của nó là

CDF(x) = 1 - $e^{-(x / \lambda)k}$

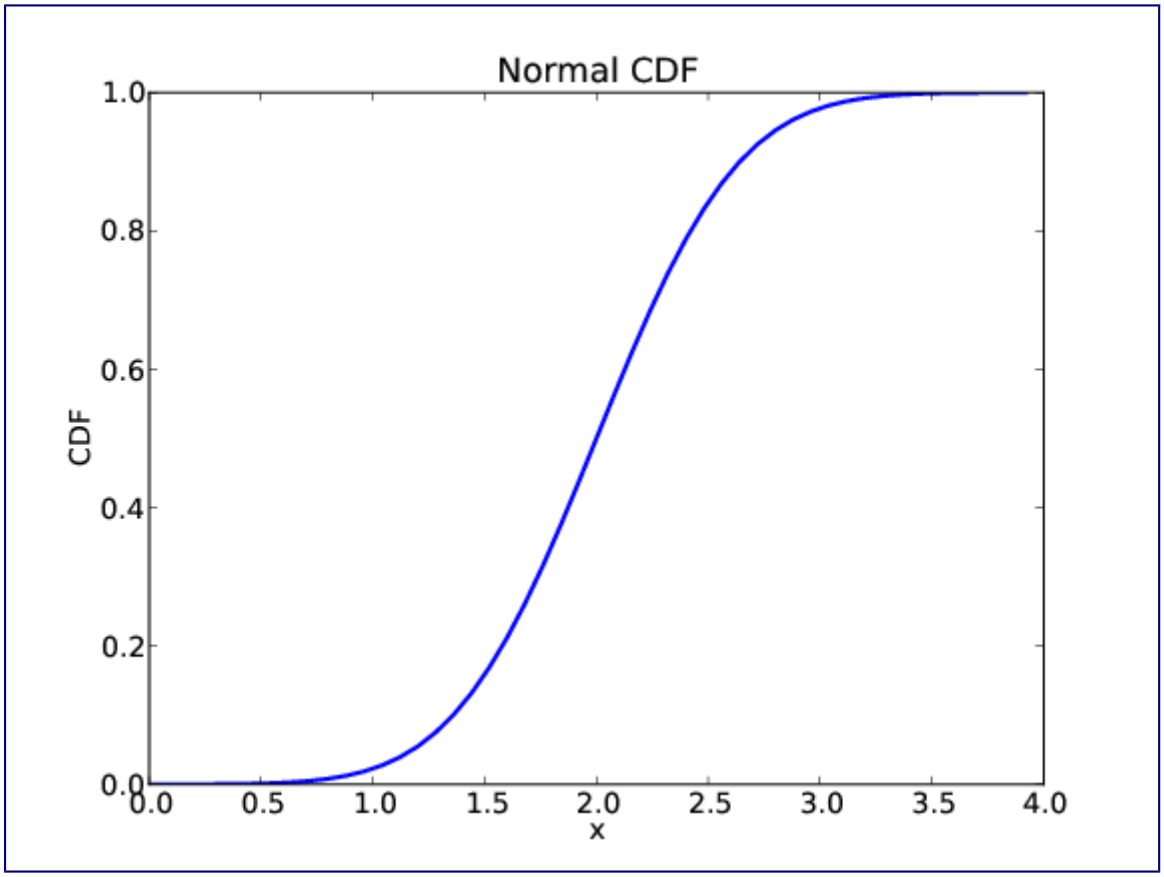
Bạn có thể tìm được một phép biến đổi nào khiến cho phân bố Weibull trở nên giống đường thẳng không? Khi đó độ dốc và tung độ giao điểm sẽ biểu thị điều gì? Hãy dùng `random.weibullvariate` để phát sinh một mẫu từ phân bố Weibull rồi dùng nó để thử nghiệm phép biến đổi của bạn.

Phân bố chuẩn

Phân bố chuẩn, còn gọi là phân bố Gauss, là loại thường được dùng nhất vì nó mô tả rất nhiều hiện tượng, chí ít là gần đúng. Hóa ra còn một lý do giải thích được tính đa năng của phân bố này, mà ta sẽ xét đến trong Mục {Định lý giới hạn trung tâm}. Phân bố chuẩn có nhiều thuộc tính khiến nó thích hợp với việc dùng để phân tích, nhưng CDF không phải là thuộc tính như vậy. Khác với những kiểu phân bố khác mà ta đã xét đến, với phân bố chuẩn CDF không có dạng biểu thức chính xác nào; cách làm thay thế thông dụng nhất cho CDF là dưới dạng **hàm sai số**, vốn là hàm đặc biệt ký hiệu bởi erf(x):

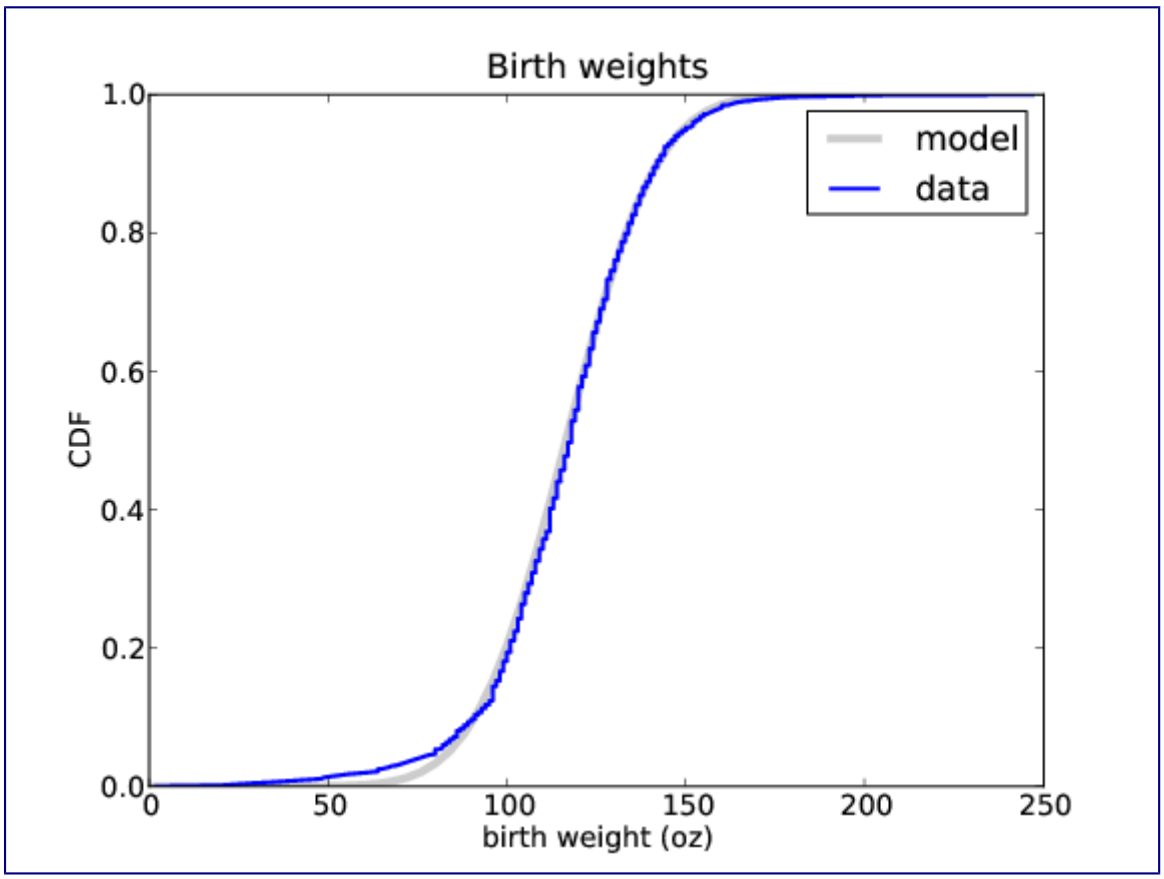
Các tham số μ và σ quy định trị trung bình và độ lệch chuẩn của phân bố này. Nếu các công thức trên làm bạn thấy đau đầu thì đừng lo; chúng rất dễ viết trong Python[2]. Có nhiều cách xấp xỉ erf(*x*) nhanh chóng và chính xác. Bạn có thể tải về một cách như vậy từ `http://thinkstats.com/erf.py`, vốn có các hàm tên là `erf` và `NormalCdf`.

Hình {normal_cdf} cho thấy CDF của một phân bố chuẩn với các tham số μ = 2,0 và σ = 0,5. Hàm sigmoid của đường cong này là một đặc trưng dễ nhận biết của phân bố chuẩn.



CDF của một phân bố chuẩn. (normal_cdf)

Trong chương trước ta đã xét đến phân bố của cân nặng trẻ sơ sinh từ NSFG. Hình {nsfg_birthwgt_model} cho thấy CDF kinh nghiệm của cân nặng tất cả trẻ được sinh ra và CDF của một phân bố chuẩn có cùng trị trung bình và độ lệch chuẩn.



CDF của cân nặng trẻ sơ sinh theo mô hình phân bố chuẩn. (nsfg_birthwgt_model)

Phân bố chuẩn là một mô hình hợp lý cho bộ số liệu này. Một **mô hình** là một sự giản hóa có ích. Trong trường hợp này, vì ta có thể tóm gọn dạng phân bố chỉ với hai số, μ = 116,5 và σ = 19,9; và sai số thu được (hiệu số giữa mô hình và số liệu) là nhỏ. Phía dưới số phần trăm thứ 10, đã có sự khác biệt giữa số liệu và mô hình; số liệu cho thấy nhiều trẻ nhẹ cân hơn ta mong đợi từ phân bố chuẩn. Nếu ta cần nghiên cứu những ca sinh sớm thì rất cần phải mô phỏng đúng phần này của phân bố, vì vậy có lẽ sẽ không tốt nếu dùng mô hình phân bố chuẩn.

Thang Weschler đo mức thông minh của người lớn là phép kiểm tra để đo trí thông minh[3]. Kết quả được chuyển đổi sao cho phân bố của điểm số trong tổng thể nói chung có dạng chuẩn với μ = 100 và σ = 15. Hãy dùng `erf.NormalCdf` để khảo sát tần số của các hiện tượng hiếm trong một phân bố chuẩn. Có mấy phần dân số có IQ cao hơn trung bình? Có mấy phần vượt trên 115? 130? 145?

Một hiện tượng "6-sigma" là giá trị vượt mức trung bình một khoảng cách bằng 6 lần độ lệch chuẩn. Vì vậy IQ của người 6-sigma thì bằng 190. Trên thế giới 6 tỉ người, có bao nhiêu người có IQ từ 190 trở lên?[4]

Hãy vẽ đồ thị CDF của thời gian mang thai cho tất cả những ca sinh thành công. Liệu nó có dạng giống như phân bố chuẩn không? Hãy tính trị trung bình và phương sai của mẫu rồi vẽ đồ thị phân bố chuẩn với cùng các tham số đó. Liệu phân bố chuẩn có phải là mô hình tốt cho số liệu này không? Nếu bạn phải tóm tắt dạng phân bố này chỉ với hai đặc trưng thống kê, thì bạn sẽ chọn những đặc trưng nào?

Đồ thị xác suất chuẩn

Đối với các phân bố lũy thừa, Pareto và Weibull, có những phép biến đổi đơn giản mà ta cso thể dùng để kiểm xem liệu một phân bố liên tục có phải là mô hình tốt cho bộ số liệu hay không. Đối với phân bố chuẩn thì không có phép chuyển đổi nào như vậy cả, nhưng có một cách làm thay thế là **đồ thị xác suất chuẩn**. Nó được dựa theo **rankit**: nếu bạn phát sinh *n* giá trị từ một phân bố chuẩn và sắp xếp chúng lại, thì rankit thứ sẽ bằng trị trung bình của phân bố đối với giá trị thứ .

> Hãy viết một hàm có tên `Sample` để phát sinh ra 6 mẫu từ một phân bố chuẩn với μ = 0 và σ = 1. Thực hiện sắp xếp và trả lại các giá trị.
>
> Hãy viết một hàm có tên `Samples` để gọi `Sample` 1000 lần và trả lại một danh sách gồm 1000 danh sách.
>
> Nếu bạn áp dụng `zip` cho danh sách chứa các danh sách nói trên, thì kết quả sẽ là 6 danh sách với mỗi danh sách chứa 1000 giá trị. Hãy tính trị trung bình của mỗi danh sách này rồi in kết quả ra. Tôi dự đoán rằng bạn sẽ nhận được đáp số tựa như sau:
>
> { - 1.2672, - 0.6418, - 0.2016, 0.2016, 0.6418, 1.2672}
>
> Nếu bạn tăng số lần gọi `Sample` lên, kết quả sẽ hội tụ về những con số trên.

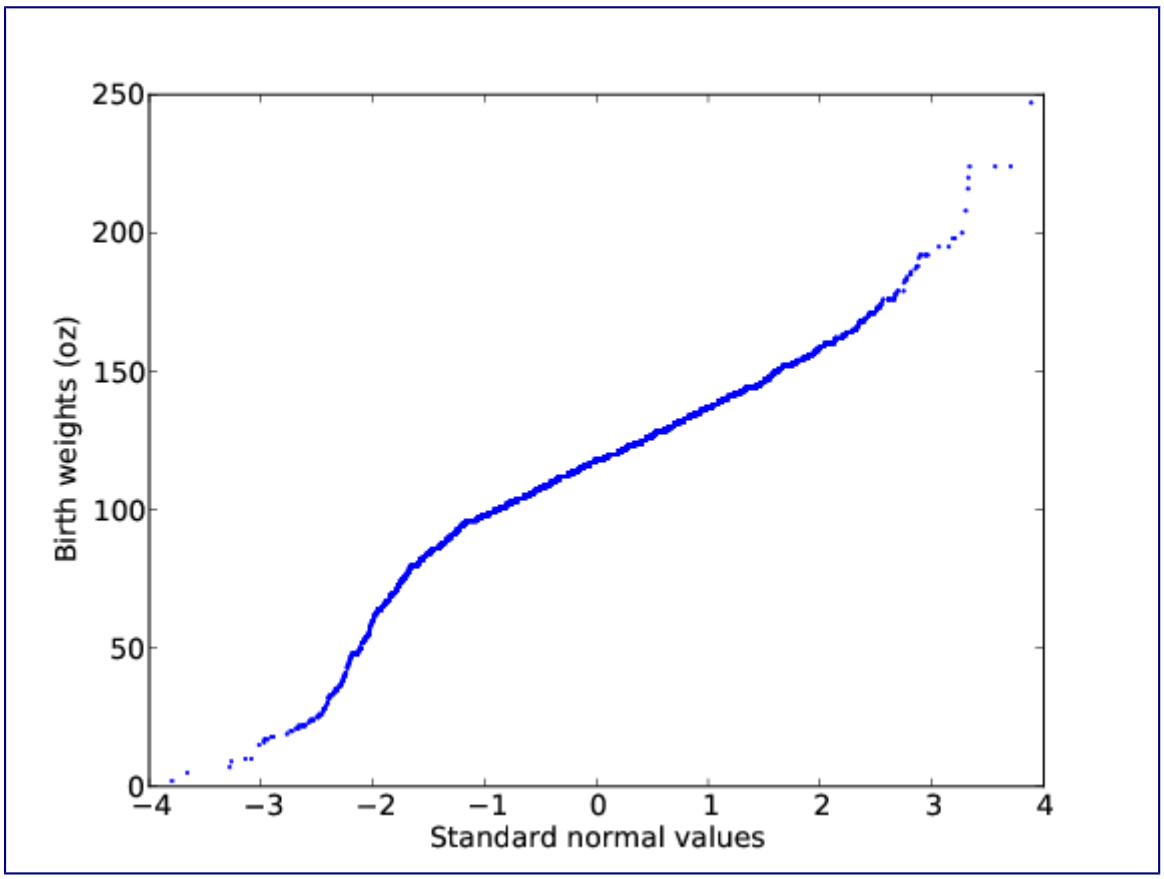
Việc tính chính xác rankit thì tương đối khó, nhưng có những phương pháp số để xấp xỉ chúng. Và có một cách tính mẹo dễ thực hiện hơn:

1. Từ phân bố chuẩn với μ = 0 và σ = 1, hãy phát sinh một mẫu có cùng kích thước với bộ số liệu hiện có, rồi sắp xếp mẫu này.
2. Sắp xếp các giá trị trong bộ số liệu.
3. Chấm các điểm số liệu ban đầu theo các điểm ngẫu nhiên được phát sinh.

Với các bộ dữ liệu lớn, phương pháp này hoạt động tốt. Với bộ số liệu nhỏ hơn, bạn có thể cải thiện nó bằng cách phát sinh *m* (*n*+1) - 1 giá trị từ một phân bố chuẩn, trong đó *n* là kích thước bộ số liệu còn *m* là một thừa số. Sau đó bắt đầu từ giá trị thứ *m*, cứ cách *m* giá trị lại chọn một.

Phương thức này cũng hoạt động với các phân bố khác, chỉ cần bạn biết cách phát sinh ra mẫu ngẫu nhiên.

Hình {nsfg_birthwgt_normal} là cách mẹo để vẽ được đồ thị xác suất chuẩn của số liệu cân nặng trẻ sơ sinh.



Đồ thị xác suất chuẩn của cân nặng trẻ sơ sinh. (nsfg_birthwgt_normal)

Độ cong của đường này cho thấy rằng có sự lệch khỏi một phân bố chuẩn; tuy vậy, đó vẫn là một mô hình dùng được cho nhiều mục đích.

> Hãy viết một hàm có tên `NormalPlot` nhận vào một dãy giá trị rồi phát sinh một đồ thị xác suất chuẩn. Bạn có thể tải về một lời giải từ `http://thinkstats.com/rankit.py`. Hãy dùng tốc độ chạy từ `relay.py` để phát sinh một đồ thị xác suất chuẩn. Liệu rằng phân bố chuẩn có phải là một mô hình tốt cho số liệu này không? Bạn có thể tải về một lời giải từ `http://thinkstats.com/relay_normal.py`.

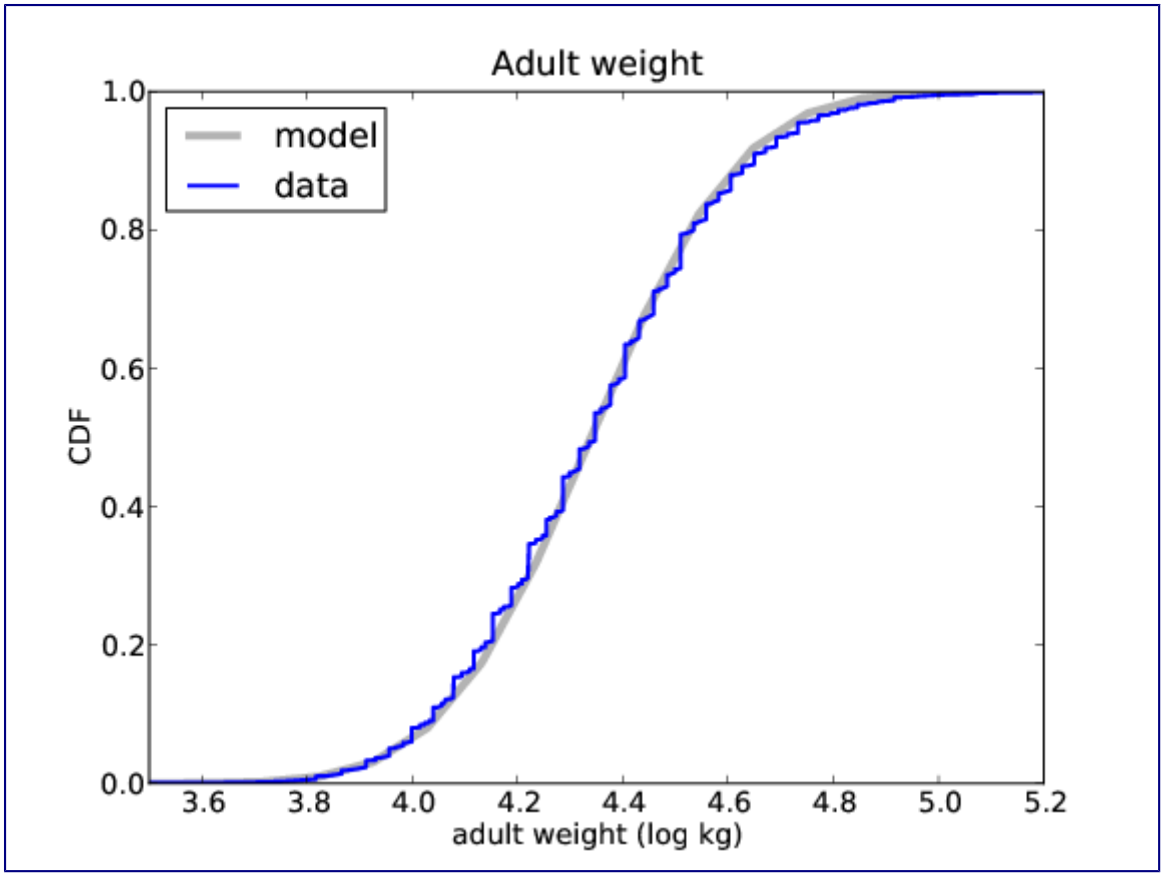
Phân bố loga chuẩn

Nếu logarit của một bộ các giá trị hợp thành phân bố chuẩn, thì bản thân các giá trị này sẽ có phân bố **loga chuẩn**. CDF của phân bố loga chuẩn cũng giống như CDF của phân bố chuẩn khi thay log *x* cho *x*.

$\text{CDF}_{\text{loga chuẩn}}(x) = \text{CDF}_{\text{chuẩn}}(\log x)$

Các tham số của phân bố loga chuẩn thường được viết là μ và σ. Song cần nhớ rằng những tham số này *không* phải là trị trung bình và độ lệch chuẩn; trị trung bình của phân bố loga chuẩn là $\exp(\mu + \sigma^2/2)$ và độ lệch chuẩn thì lằng nhằng hơn [5]. Hóa ra là phân bố cân nặng của người lớn có dạng xấp xỉ loga chuẩn [6].

“National Center for Chronic Disease Prevention and Health Promotion” đã tiến hành cuộc điều tra thường niên như một phần của “Behavioral Risk Factor Surveillance System” (BRFSS)[7]. Vào năm 2008, họ đã phỏng vấn 414.509 người và lấy các thông tin về cấu trúc dân số, sức khỏe và rủi ro liên quan đến sức khỏe.



CDF của cân nặng người lớn (chuyển đổi loga). (brfss_weight_log)

Trong số dữ liệu thu thập được có cân nặng tính theo ki-lô của 398.484 người. Hình {brfss_weight_log} trên cho thấy phân bố của log *w*, trong đó *w* là cân nặng theo ki-lô, cùng với một mô hình phân bố chuẩn. Mô hình phân bố chuẩn khớp với số liệu, mặc dù các số cân nặng lớn nhất ngay cả khi lấy loga vẫn vượt quá mô hình chuẩn. Vì phân bố của log *w* khớp với một phân bố chuẩn, ta kết luận được rằng *w* khớp với một phân bố loga chuẩn.

Hãy tải số liệu BRFSS về từ `http://thinkstats.com/CDBRFS08.ASC.gz`, và mã lệnh do tôi viết để đọc số liệu đó từ `http://thinkstats.com/brfss.py`. Hãy chạy `brfss.py` và khẳng định rằng nó in ra các đặc trưng thống kê cho một vài biến. Hãy viết một chương trình để đọc vào cân nặng của người lớn từ BRFSS và phát sinh các đồ thị xác suất chuẩn cho *w* và log *w*. Bạn có thể tải về một lời giải từ `http://thinkstats.com/brfss_figs.py`.

Phân bố của số dân thành phố được đề xuất lấy làm ví dụ cho một hiện tượng thực tế mà có thể được mô tả bằng một phân bố Pareto. Cục thống kê dân số Hoa Kỳ (U.S. Census Bureau) đã xuất bản số liệu về dân số của từng thành phố / thị trấn trên lãnh thổ nước Mỹ. Tôi đã viết một chương trình nhỏ để tải về số liệu này và lưu nó vào một file. Bạn có thể tải chương trình về từ `http://thinkstats.com/populations.py`.

1. Hãy đọc qua chương trình để nắm vững mục đích của nó; rồi chạy nó để tải về và xử lý số liệu.
2. Viết một chương trình tính rồi vẽ đồ thị phân bố của dân số cho 14593 thành phố và thị trấn trong bộ số liệu.
3. Vẽ đồ thị CDF lần lượt trên trục tỉ lệ tuyến tính và log-*x*, từ đó hình dung được hình dạng của phân bố này. Sau đó vẽ CCDF lên trục tỉ lệ log-log để xem liệu nó có hình dạng đặc trưng của một phân bố Pareto không.
4. Hãy thử các phép biến đổi và kiểu đồ thị khác trong chương này để xem liệu có mô hình nào tốt hơn cho bộ số liệu này không.

Bạn rút ra được kết luận gì về phân bố quy mô (số dân) các thành phố và thị trấn? Bạn có thể tải về một lời giải từ `http://thinkstats.com/populations_cdf.py`.

Ở Hoa Kỳ, tổ chức Internal Revenue Service (IRS) cung cấp số liệu về thuế thu nhập tại `http://irs.gov/taxstats`. Một trong số các file của họ, bao gồm thông tin về các khoản thu nhập cá nhân trong năm 2008, được đăng tại `http://thinkstats.com/08in11si.csv`. Tôi đã chuyển nó sang dạng file chữ CSV ("comma-separated values"); bạn có thể đọc file này bằng module `csv`.

Từ bộ số liệu này, hãy kết xuất phân bố của thu nhập. Liệu có dạng phân bố liên tục nào trong chương này là mô hình phù hợp cho số liệu này không? Bạn có thể tải về lời giải từ `http://thinkstats.com/irs.py`.

Tại sao cần dùng mô hình?

Ở đầu chương này, tôi đã nói rằng nhiều hiện tượng thực tế có thể được mô hình hóa bởi phân bố liên tục. Bạn có thể hỏi, "Vậy thì sao?"

Cũng như tất cả mô hình, các phân bố liên tục đều trừu tượng, theo nghĩa chúng lược bỏ tất cả những chi tiết nào được coi là thừa. Chẳng hạn, một phân bố được quan sát có thể chứa những sai số đo đạc hay nhiễu đặc thù của mẫu đó; các mô hình liên tục thì làm trơn tất cả những biến động này. Mô hình liên tục cũng là một hình thức nén dữ liệu. Khi một mô hình khớp phù hợp với bộ số liệu thì một tập hợp ít các tham số có thể tóm tắt được cả một lượng số liệu rất lớn. Đôi khi thật ngạc nhiên khi số liệu từ một hiện tượng tự nhiên lại khớp một phân bố liên tục, nhưng các quan sát này có thể dẫn tới kiến thức sâu sắc về hệ vật lý. Đôi khi chúng ta có thể giải thích tại sao một phân bố quan sát được lại có dạng riêng nào đó. Chẳng hạn, phân bố Pareto thường là kết quả của các quá trình phát sinh với phản hồi tích cực (thường gọi là quá trình gắn kết theo ý thích: hãy xem `http://wikipedia.org/wiki/Preferential_attachment`.).

Các phân bố liên tục rất thích hợp cho việc phân tích toán học, như ta sẽ được thấy ở Chương {tính toán}.

Phát sinh số ngẫu nhiên

Các CDF liên tục rất có ích trong việc phát sinh ra số ngẫu nhiên. Nếu có một cách làm hiệu quả để tính được CDF ngược, ICDF(p), thì ta sẽ có thể phát sinh ra những giá trị ngẫu nhiên với dạng phân bố thích hợp bằng cách chọn một phân bố đều từ 0 đến 1, rồi chọn

x = ICDF(p)

Chẳng hạn, CDF của phân bố lũy thừa là

$p = 1 - e^{-\lambda x}$

Giải theo x ta được:

$x = -\log(1 - p) / \lambda$

Vì vậy trong Python, ta có thể viết

```
def expovariate(lam):
    p = random.random()
    x = -math.log(1-p) / lam
    return x
```

Tôi gọi tham số là `lam` vì `lambda` trùng với một từ khóa của Python. Hầu hết phương thức `random.random` có thể trả lại giá trị 0 nhưng không thể trả lại 1, vì vậy $1 - p$ có thể bằng 1 nhưng không bằng 0; điều này tốt vì log 0 là vô định.

> Hãy viết một hàm có tên `weibullvariate` nhận vào `lam` và `k` rồi trả lại một giá trị ngẫu nhiên từ phân bố Weibull với các tham số đó.

Thuật ngữ

phân bố kinh nghiệm:
: Phân bố của các giá trị trong một mẫu.

phân bố liên tục:
: Phân bố được mô tả bởi một hàm liên tục.

khoảng thời gian giữa:
: Khoảng thời gian trôi qua giữa hai sự kiện.

hàm sai số:
: Hàm toán học đặc biệt, nó có tên như vậy vì được tìm ra trong quá trình nghiên cứu sai số của phép đo đạc.

đồ thị xác suất chuẩn:
: Đồ thị biểu diễn các giá trị đã sắp xếp trong một mẫu, theo các giá trị được trông đợi của chúng nếu phân bố có dạng chuẩn.

rankit:
: Giá trị kì vọng của một phần tử trong danh sách đã sắp xếp gồm các giá trị từ một phân bố chuẩn.

mô hình:
: Một cách giản hóa có ích. Các phân bố liên tục thường là mô hình tốt cho những phân bố kinh nghiệm phức tạp hơn.

tác phẩm:
: Chỉnh thể văn bản được dùng làm mẫu phân tích ngôn ngữ.

hapaxlegomenon:
: Từ xuất hiện chỉ một lần trong tác phẩm. Trong quyển sách này, đến giờ thì nó xuất hiện hai lần.

---

1. Ví dụ này được dựa theo thông tin và số liệu từ Dunn, “A Simple Dataset for Demonstrating Common Distributions,” Journal of Statistics Education v.7, n.3 (1999). ↩
2. Từ Python 3.2, thậm chí còn dễ hơn; `erf` đã sẵn có trong module `math`. ↩

3. Về việc liệu đây có phải là một chủ đề gây tranh cãi thú vị hay không thì bạn có thể tự tìm hiểu lúc rảnh rỗi. ↩
4. Về chủ đề này, bạn đọc quan tâm có thể xem http://wikipedia.org/wiki/Christopher_Langan. ↩
5. Xem http://wikipedia.org/wiki/Log-normal_distribution. ↩
6. Tôi được khuyến cáo điều này, có lẽ qua một lời bình (không ghi chú thích) ở http://mathworld.wolfram.com/LogNormalDistribution.html. Sau đó tôi đã tìm thấy một bài báo đề xuất lý do và cách thực hiện phép chuyển đổi loga; đó là Penman and Johnson, “The Changing Shape of the Body Mass Index Distribution Curve in the Population,” Preventing Chronic Disease, 2006 July; 3(3): A74. Bản trực tuyến tại http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1636707. ↩
7. Centers for Disease Control and Prevention (CDC). Behavioral Risk Factor Surveillance System Survey Data. Atlanta, Georgia: U.S. Department of Health and Human Services, Centers for Disease Control and Prevention, 2008. ↩

Chương 5: Xác suất

Trở về Muc luc cuốn sách

Trong Chương 2, tôi đã đề cập rằng một xác suất có thể coi như tần suất, tức là tần số biểu diễn theo tỉ lệ so với kích thước mẫu. Đó cũng là một định nghĩa của xác suất, nhưng không phải duy nhất. Trên thực tế, định nghĩa của xác suất là một chủ đề gây tranh cãi.

Chúng ta sẽ bắt đầu với các phần không gây tranh cãi trước khi đi xa hơn. Nói chung mọi ý kiến đều thống nhất là xác suất là một giá trị số thực trong khoảng từ 0 đến 1, vốn được định trước là một độ đo ứng với khả năng mà một hiện tượng nào đó dễ xảy ra hơn so với hiện tượng khác. Những "hiện tượng" mà ta gán cho xác suất như trên được gọi là **sự kiện**. Nếu E biểu diễn một sự kiện, thì P(E) biểu diễn xác suất mà E sẽ xảy ra. Tình huống mà E có thể hoặc không xảy ra được gọi là một **phép thử**. Lấy ví dụ, chẳng hạn bạn có một con xúc sắc 6 mặt và muốn biết xác suất để gieo được mặt "lục". Mỗi lần gieo là một phép thử. Mỗi lần mặt "lục" xuất hiện được coi là **thành công**; còn những phép thử khác đều được coi là **thất bại**. Những thuật ngữ này được dùng ngay cả trong tình huống ở đó "thành công" có nghĩa xấu còn "thất bại" có nghĩa tốt.

Nếu ta có một mẫu hữu hạn gồm *n* phép thử và quan sát được *s* thành công, thì xác suất thành công là *s*/*n*. Nếu không gian phép thử là vô hạn thì việc định nghĩa xác suất sẽ phải khéo hơn một chút, nhưng đa số mọi người đều sẵn lòng chấp nhận những khẳng định xác suất về một chuỗi các phép thử giống hệt nhau (điều này có tính giả tưởng), như việc tung đồng xu hoặc gieo xúc sắc. Chúng ta bắt đầu gặp rắc rối khi nói về xác suất của những sự kiện duy nhất. Chẳng hạn, có thể ta muốn biết xác suất một ứng cử viên sẽ thắng cuộc bầu cử. Nhưng vì mỗi cuộc bầu cử là duy nhất, nên không có một chuỗi các phép thử giống nhau để xét. Trong những trường hợp như thế này có người sẽ nói rằng khái niệm về xác suất không dùng được. Điều này đôi khi được gọi là **định nghĩa theo tần suất** vì theo đó xác suất được ước tính từ tần suất. Vì không có một loạt những phép thử giống nào nên sẽ không có xác suất. Định nghĩa theo tần suất thì an toàn về triết lý, nhưng rất bó buộc vì nó hạn chế phạm vi của xác suất chỉ trong những hệ vật lý hoặc là ngẫu nhiên (như phân rã nguyên tử) hoặc là không thể dự đoán được mà ta buộc phải mô phỏng như là ngẫu nhiên (chẳng hạn con xúc sắc khi gieo). Bất cứ điều gì liên quan đến con người thì miễn bàn đến.

Một cách làm khác là **thuyết Bayes**, trong đó định nghĩa xác suất như là mức độ tin cậy rằng một sự kiện sẽ xảy ra. Theo định nghĩa này, khái niệm xác suất có thể được áp dụng trong hầu hết mọi trường hợp. Một khó khăn đối với xác suất Bayes là nó phụ thuộc vào trạng thái kiến thức của mỗi người; những người với thông tin khác nhau có thể có những cấp độ tin cậy khác nhau về cùng một sự kiện. Vì lý do này, nhiều người cho rằng xác suất Bayes mang tính chủ quan nhiều hơn so với xác suất tính theo tần suất. Lấy ví dụ, xác suất để ông Thaksin Shinawatra là thủ tướng Thái Lan bằng bao nhiêu? Một người ủng hộ quan điểm tần suất sẽ nói rằng không có xác suất cho sự kiện này vì không có loạt phép thử nào. Thaksin hoặc là thủ tướng, hoặc không phải; điều này không can hệ gì đến xác suất.

Trái lại, người ủng hộ quan điểm Bayes sẽ sẵn lòng ấn định một xác suất cho sự kiện này dựa trên hiện trạng kiến thức của người đó. Chẳng hạn, nếu bạn nhớ rằng có một cuộc bạo loạn giành chính quyền ở Thái Lan vào năm 2006, và bạn chắc rằng Thaksin chính là ông thủ tướng bị lật đổ, thì bạn

sẽ có thể gán một xác suất bằng 0,1; tức là vẫn dành chỗ cho một khả năng nhỏ là bạn nhớ nhầm, hoặc Thaksin tái đắc cử.

Nếu bạn tra Wikipedia, bạn có thể thấy rằng Thaksin không phải là thủ tướng Thái Lan (tại thời điểm cuốn sách này được viết). Dựa trên thông tin này, bạn có thể sửa lại ước tính cho xác suất chỉ còn 0,01; tức là có xét đến khả năng Wikipedia bị nhầm lẫn.

Các quy tắc với xác suất

Với các xác suất theo tần suất, chúng ta có thể rút ra những quy tắc gắn với xác suất của những sự kiện khác nhau. Có lẽ trong số những quy tắc kiểu này thì thường được biết đến rõ nhất là:

P(A và B) = P(A) P(B) Lưu ý: không phải lúc nào cũng đúng!

trong đó P(A và B) là xác suất mà cả hai hiện tượng A và B đều xảy ra. Công thức này dễ nhớ; song vấn đề duy nhất là *không phải* lúc nào nó cũng đúng. Công thức này chỉ dùng được khi A và B là **độc lập**, có nghĩa là nếu tôi biết rằng A đã xảy ra, thì điều đó chẳng làm thay đổi xác suất của B, và ngược lại. Chẳng hạn, nếu A là việc tung đồng xu để được mặt ngửa, và B là việc gieo xúc sắc và được mặt "nhất", thì A và B là độc lập nhau, vì tung đồng xu chẳng cho tôi biết thêm gì về việc gieo xúc sắc. Nhưng nếu tôi gieo hai con xúc sắc, và A là việc được ít nhất một mặt "lục", và B là được hai mặt lục, thì A và B không độc lập, vì nếu tôi đã biết rằng A xảy ra rồi, thì xác suất xảy ra B cao hơn, và nếu tôi biết B đã xảy ra rồi, thì xác suất của A bằng 1. Khi A và B không độc lập nhau, thường ta sẽ cần tính xác suất có điều kiện, P(A|B), vốn là xác suất của A khi biết rằng B đã xảy ra:
$P(A|B) = P(A \text{ và } B) / P(B)$
Từ đó ta có thể rút ra hệ thức tổng quát

P(A và B) = P(A) P(B|A)

Công thức này có thể không dễ nhớ bằng, nhưng nếu bạn dịch ra ngôn ngữ nói thì nó sẽ có nghĩa: "Khả năng xảy ra cả hai sự kiện bằng khả năng xảy ra sự kiện thứ nhất, và sau đó là sự kiện thứ hai khi sự kiện thứ nhất đã xảy ra."

Không có gì đặc biệt về thứ tự các sự kiện, do đó ta có thể viết

P(A và B) = P(B) P(A|B)

Các hệ thức này đúng bất kể A và B có độc lập hay không. Nếu chúng độc lập nhau, thì P(A|B) = P(A), và ta quay về điểm xuất phát.

Vì tất cả các xác suất đều có khoảng từ 0 đến 1, nên dễ thấy được

P(A và B) ≤ P(A)

Để hình dung được điều này, hãy tưởng tượng một câu lạc bộ chỉ kết nạp những người có được tiêu chuẩn nào đó, A. Bây giờ, giả sử họ bổ sung một tiêu chuẩn nữa, B. Dường như hiển nhiên là câu lạc bộ sẽ bị thu nhỏ đi, hoặc sẽ giữ nguyên nếu mọi thành viên ai cũng thỏa mãn B. Nhưng có một số tình huống mà mọi người lại phân tích kiểu bài toán này rất kém. Các bạn có thể tham khảo những ví dụ và bình luận về điều này ở
http://wikipedia.org/wiki/Conjunction_fallacy.

Nếu tôi gieo hai con xúc sắc và thu được tổng là 8, thì có bao nhiêu phần khả năng một con xúc sắc có mặt lục?

Nếu tôi gieo 100 con xúc sắc, có bao nhiêu phần khả năng thu được toàn mặt lục? Bao nhiêu khả năng không thu được mặt lục nào?

Những câu hỏi sau đây được chỉnh lại từ nguồn Mlodinow, *The Drunkard's Walk*.

1. Nếu một gia đình có hai con, thì khả năng có hai con gái là bao nhiêu?
2. Nếu một gia đình có hai con và ta đã biết rằng có ít nhất một con gái, thì khả năng có hai con gái là bao nhiêu?
3. Nếu một gia đình có hai con và ta đã biết rằng con lớn là một con gái, thì khả năng có hai con gái là bao nhiêu?
4. Nếu một gia đình có hai con và ta đã biết rằng có ít nhất một đứa con gái có tên là Florida, thì khả năng có hai con gái là bao nhiêu?

Bạn có thể giả sử rằng xác suất để một đứa bé bất kì là gái thì bằng 1/2, và các đứa trẻ trong một gia đình là các phép thử độc lập (theo mọi nghĩa). Bạn cũng có thể giả sử rằng tỉ lệ phần trăm của những đứa bé gái có tên là Florida thì rất nhỏ.

Monty Hall

Bài toán Monty Hall có thể không phải là câu hỏi dễ gây tranh cãi nhất trong lịch sử môn học xác suất. Tình huống rất đơn giản, nhưng câu trả lời đúng lại thật phản trực giác đến nỗi nhiều người không thể chấp nhận được nó, và nhiều người thông minh đã tự làm khó mình không chỉ vì đã đoán sai, mà còn lý sự để bênh vực cái sai, ngay trước công chúng.

Monty Hall có thời là chủ biên của sô diễn *Let's Make a Deal*. Bài toán Monty Hall được dựa trên một trò chơi thường xuyên của chương trình này. Nếu bạn được nhập vai người chơi thì mọi việc sẽ xảy ra như sau:

- Monty chỉ cho bạn thấy ba cánh cửa đóng kín và nói rằng có một phần thưởng sau mỗi cánh cửa: một giải là chiếc xe hơi, hai giải thưởng còn lại thì kém giá trị hơn, chẳng hạn kem lạc và móng tay giả. Giải thưởng được sắp xếp ngẫu nhiên.
- Mục đích của trò chơi là đoán cánh cửa nào đang che chiếc xe hơi. Nếu bạn đoán đúng, bạn sẽ được thưởng chiếc xe đó.
- Vì vậy bạn chọn một cánh cửa, mà ta sẽ gọi là cửa A. Ta gọi các cánh cửa kia là B và C.
- Trước khi mở cánh cửa mà bạn đã chọn, Monty muốn tăng mức độ gay cấn bằng cách mở cánh cửa B hoặc C, mà không có xe ở sau. (Ngay cả nếu có xe sau cửa A, Monty vẫn có thể mở cửa B hoặc C được, và họ sẽ chọn mở cửa ngẫu nhiên).
- Sau đó Monty cho bạn lựa chọn hoặc là giữ nguyên cánh cửa đã chọn, hoặc là chuyển sang cánh cửa còn lại vẫn đang đóng.

Câu hỏi là, liệu bạn nên "giữ nguyên" hay "chuyển" hoặc hai điều này chẳng khác gì? Phần lớn chúng ta đều có trực giác mạnh rằng có chuyển hay giữ cũng chẳng khác gì; với suy luận là, còn hai cánh cửa thì khả năng xe ở sau cửa A là 50%.

Nhưng điều này sai. Thật ra, khả năng đoán đúng nếu bạn cố giữ cửa A chỉ còn 1/3; nếu bạn chuyển thì sẽ có khả năng 2/3. Tôi sẽ giải thích tại sao, nhưng cũng không trông đợi là bạn tin tôi ngay.

Vấn đề mâu chốt để nhận thấy điều này là có ba tình huống có thể: xe ở sau cánh cửa A, B, hoặc C. Vì giải thưởng được sắp xếp ngẫu nhiên nên xác suất của mỗi tình huống này đều bằng 1/3.

Nếu chiến thuật của bạn là giữ nguyên cửa A, thì bạn chỉ giành giải trong tình huống A, tức là với xác suất bằng 1/3.

Nếu chiến thuật của bạn là chuyển, thì bạn sẽ giành giải trong cả tình huống B hoặc C, như vậy xác suất tổng hợp để giành giải sẽ là 2/3.

Nếu bạn vẫn không chịu thuyết phục bởi lập luận này, thì bạn có vấn đề cũng như nhiều người khác. Khi một người bạn trình bày lời giải này cho Paul Erdos, ông đã trả lời, "Không, không thể nào. Điều này là không thể. Đúng ra là không có khác biệt nào.[1]"

Mọi lý luận đều không thể thuyết phục được ông. Cuối cùng phải nhờ đến mô phỏng bằng máy tính thì ông mới chịu tin.

> Hãy viết một chương trình để mô phỏng bài toán Monty Python rồi dùng nó để ước tính xác suất giành giải nếu bạn giữ nguyên và nếu bạn chuyển.
>
> Sau đó hãy đọc lời bàn luận xoay quanh bài toán này tại `http://wikipedia.org/wiki/Monty_Hall_problem`.
>
> Bạn thấy cách nào thuyết phục hơn, chương trình mô phỏng hay lý lẽ, và tại sao?
>
> Để hiểu được bài toán Monty Hall, điều quan trọng là nhận thấy được bằng cách chọn cửa để mở, Monty đã cho bạn thông tin. Để thấy được tại sao điều này lại có ích, hãy hình dung trường hợp mà Monty không biết xe nằm sau cửa nào, và họ chọn ngẫu nhiên cửa B hoặc C. Nếu họ mở được trúng vào cửa có xe, thì trò chơi kết thúc, bạn thua, mà còn không được quyền chọn giữ nguyên hoặc chuyển nữa.
>
> Nếu không như vậy, bạn còn thấy nên giữ hay chuyển?

Poincare

Henri Poincare từng là một nhà toán học người Pháp dạy ở trường Sorbonne vào khoảng những năm 1900. Lời đồn sau đây về ông có thể đã được thêu dệt, nhưng nó trở thành một bài toán xác suất thú vị. Giả sử rằng Poincare nghi ngờ là xưởng sản xuất bánh mì ở làng ông đang bán ổ bánh mì nhẹ hơn khối lượng niêm yết là 1 kg, vì vậy hằng ngày ông đều mua một ổ bánh, mang về nhà và đem cân lên. Đến cuối năm, ông vẽ đồ thị phân bố kết quả cân rồi cho thấy kết quả này khớp với một phân bố chuẩn có trị trung bình 950 g và độ lệch chuẩn 50 g. Ông mang bằng chứng này đến cảnh sát, và họ cảnh cáo chủ xưởng bánh. Sang năm tới, Poincare tiếp tục thói quen cân bánh mì mỗi ngày. Và đến cuối năm, ông thấy rằng khối lượng bằng 1000 g đúng như yêu cầu, nhưng một lần nữa ông có ý kiến với cảnh sát, và lần này họ lại phạt xưởng bánh. Vì sao? Vì hình dạng của phân bố bất đối xứng. Không như phân bố chuẩn, nó bị lệch sang phải, tức là thống nhất với giả thuyết rằng thợ làm bánh vẫn sản xuất ổ bánh 950 g, nhưng cố ý bán cho Poincare những ổ nặng hơn.

> Hãy viết một chương trình để mô phỏng người thợ bánh khi chọn *n* ổ từ một phân bố với trị trung bình 950 g và độ lệch chuẩn 50 g, rồi bán cho Poincare những chiếc nặng

nhất. Giá trị nào của *n* cho ta phân bố có trị trung bình bằng 1000 g? Độ lệch chuẩn bằng bao nhiêu?

Hãy so sánh dạng phân bố này với một phân bố chuẩn có cùng trị trung bình và độ lệch chuẩn. Liệu sự khác biệt về hình dạng phân bố có đủ rõ ràng để thuyết phục được người cảnh sát không?

Nếu bạn đi khiêu vụ ở đó các cặp được chọn ngẫu nhiên thì có bao nhiêu phần trăm những cặp khác phái có người phụ nữ cao hơn nam giới? Trong BRFSS (xem Mục [phân bố loga chuẩn]), phân bố của chiều cao con người có dạng xấp xỉ chuẩn với các tham số $\mu = 178$ cm và $\sigma^2 = 59{,}4$ cm ở nam giới, và $\mu = 163$ cm và $\sigma^2 = 52{,}8$ cm ở nữ giới. [* Lưu ý: Mặc dù chỗ này tôi viết giống sách gốc dùng đơn vị cm cho σ^2 nhưng chính ra đơn vị phải là cm^2. *] Cũng phải nói thêm, bạn có thể nhận thấy rằng độ lệch chuẩn của nam giới thì lớn hơn và tự hỏi rằng liệu chiều cao của nam giới có biến động nhiều hơn không. Để so sánh độ biến động giữa các nhóm, cần phải tính **hệ số biến động**, vốn là tỉ lệ giữa độ lệch chuẩn với trị trung bình, σ/μ. Theo độ đo này, chiều cao của nữ giới có biến động cao hơn một chút.

Một quy tắc khác về xác suất

Hai sự kiện được gọi là **xung khắc tương hỗ** nếu chỉ một trong số chúng có thể xảy ra, vì vậy các xác suất điều kiện đều bằng 0:

$P(A|B) = P(B|A) = 0$

Trong trường hợp này, dễ tính được xác suất của từng sự kiện:

$P(A \text{ hoặc } B) = P(A) + P(B)$ Lưu ý: không phải lúc nào cũng đúng.

Nhưng nhớ rằng điều này chỉ áp dụng được khi các sự kiện xung khắc tương hỗ. Nói chung, xác suất của A hoặc B hoặc cả hai thì bằng:

$P(A \text{ hoặc } B) = P(A) + P(B) - P(A \text{ và } B)$

Lý do ta phải trừ đi P(A và B) là vì nếu không thì nó sẽ được tính hai lần. Chẳng hạn, nếu tôi tung 2 đồng xu thì khả năng ít nhất một mặt sấp sẽ bằng $1/2 + 1/2 - 1/4$. Tôi phải trừ đi 1/4 vì nếu không tôi sẽ tính trường hợp cả 2 đều ngửa là hai lần. Vấn đề sẽ còn rõ hơn nêu tôi tung 3 đồng xu.

Nếu tôi gieo hai con xúc sắc thì khả năng gieo được ít nhất một mặt lục sẽ bằng bao nhiêu?

Công thức chung để tính xác suất của A hoặc B nhưng không phải cả hai là gì?

Phân bố nhị thức

Nếu tôi gieo 100 con xúc sắc thì khả năng thu được toàn bộ mặt lục là $(1 / 6)^{100}$. Và khả năng không được mặt lục nào thì bằng $(5 / 6)^{100}$.

Những trường hợp như vậy rất dễ, nhưng tổng quá hơn, chúng ta có thể muốn biết khả năng thu được *k* mặt lục, với mọi giá trị của *k* từ 0 đến 100. Câu trả lời là **phân bố nhị thức**, vốn có PMF sau:

$$\text{PMF}(k) = C_n^k p^k (1 - p)^{n - k}$$

trong đó n là số phép thử, p là xác suất thành công, và k là số lần thành công. **Hệ số nhị thức** được đọc là "n chọn k", và nó có thể được tính trực tiếp như sau:

$$C_n^k = n! / k!(n - k)!$$

hoặc theo cách truy hồi sau:

$$C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$$

với hai trường hợp cơ bản: nếu $n = 0$ thì kết quả sẽ bằng 0; nếu $k = 0$ thì kết quả sẽ bằng 1. Nếu tải về `http://thinkstats.com/thinkstats.py`, bạn sẽ thấy một hàm có tên `binom` để tính hệ số nhị thức khá hiệu quả.

> Nếu bạn tung đồng xu 100 lần, bạn sẽ trông đợi 50 mặt ngửa, nhưng xác suất để có được chính xác 50 mặt ngửa thì bằng bao nhiêu?

Chuỗi thắng lợi và những bàn tay vàng

Chúng ta thường không có trực giác tốt về những quá trình ngẫu nhiên. Nếu bạn yêu cầu người khác viết ra các số "ngẫu nhiên", họ sẽ có xu hướng tạo ra các dãy số trông có vẻ ngẫu nhiên, nhưng thực ra là trật tự hơn so với các dãy ngẫu nhiên theo đúng nghĩa. Ngược lại, khi bạn cho họ xem một chuỗi số ngẫu nhiên thực sự, thông thường họ sẽ có xu hướng tìm ra những quy luật mà thực ra không có ở chuỗi số đó.

Một ví dụ cho hiện tượng thứ hai nói trên là có nhiều người tin vào "chuỗi thắng" trong thể thao: một người chơi thành công gần đây được gọi là "bàn tay vàng;" một người chơi khác không thành công bằng thì gọi là "hết thời." Các nhà thống kê đã kiểm tra những giả thiết này trong một loạt các môn thể thao, và kết quả thống nhất là không có thứ gì gọi là chuỗi thắng.[2] Nếu bạn giả sử rằng mỗi cố gắng có tính độc lập với những cố gắng trước đó thì sẽ thấy được rằng hiếm khi xảy ra chuỗi chiến thắng hoặc thất bại liên tiếp. Những chuỗi thắng hiện hữu đó không đủ để chứng tỏ có mối liên hệ gì giữa những nỗ lực kế tiếp. Một hiện tượng có liên quan là ảo tượng cụm, vốn là xu thế nhìn thấy những cụm trong các mẫu hình không gian hoàn toàn ngẫu nhiên (xem `http://wikipedia.org/wiki/Clustering_illusion`). Để kiểm tra xem rằng liệu sự hiện hữu của một cụm thì có ý nghĩa hay không, chúng ta có thể mô phỏng động thái của một hệ ngẫu nhiên để xem liệu nó có thể tạo ra một cụm tương tự không. Quá trình này được gọi là mô phỏng **Monte Carlo** vì việc phát sinh các số ngẫu nhiên gợi cho ta nhớ về các sòng bạc (và Monte Carlo nổi tiếng về sòng bạc).

> Nếu có 10 người chơi một trận bóng rổ và trong trận mỗi người ném 15 quả, mỗi quả có xác suất trúng bằng 50% thì xác suất mà bạn sẽ thấy trong một trận đấu có ít nhất một người ném được liền 10 quả là bao nhiêu? Nếu bạn xem cả mùa giải gồm 82 trận thì khả năng bạn sẽ thấy được ít nhất một cầu thủ ném được trúng liền 10 quả hay trượt liền 10 quả là bao nhiêu? Bài toán này cho thấy mặt mạnh và mặt yếu của mô phỏng Monte Carlo. Một điểm mạnh là viết được chương trình mô phỏng thường dễ đàng và nhanh chóng, mà không cần phải có kiến thức cao siêu về xác suất. Còn điểm yếu là ở chỗ ước tính xác suất của một sự kiện hiếm có thể sẽ mất nhiều thời gian! Việc phân tích một chút có thể sẽ giúp bớt được nhiều tính toán.

Vào năm 1941 Joe DiMaggio đã có chuỗi 56 trận liền mạch mà mỗi trận đầu ít nhất 1 lần đánh trúng.[3] Nhiều cổ động viên bóng chày cho rằng đây là chuỗi thành tích lớn nhất trong bất kì môn thể thao nào có trong lịch sử, vì nó quá hiếm khi xảy ra. Hãy dùng phép mô phỏng Monte Carlo để ước tính xác suất mà một cầu thủ trong giải bóng chày liên đoàn [Major League] sẽ đánh được chuỗi liên tiếp 57 trận hoặc cao hơn trong khoảng 100 năm tới.

Một cụm ung thư được Centers for Disease Control (CDC) định nghĩa là "số ca ung thư cao hơn dự tính xảy ra trong một nhóm người ở một khu vực địa lý trong suốt một khoảng thời gian.[4]" Nhiều người diễn giải cụm ung thư như là một bằng chứng cho hiểm họa môi trường, nhưng nhiều nhà khoa học và thống kê học nghĩ rằng việc nghiên cứu cụm ung thư chỉ phí thời gian.[5] Tại sao? Một trong số các lý do là vì nhận diện cụm ung thư là một trường hợp kinh điển thuộc về Sự ngộ nhận của xạ thủ (xem http://wikipedia.org/wiki/Texas_sharpshooter_fallacy). Mặc dù vậy, khi ai đó báo cáo về một cụm ung thư, CDC buộc phải tiến hành điều tra. Theo trang web của họ: [tạm dịch]

> "Các điều tra viên xây dựng một định nghĩa về 'ca', một khung thời gian nghiên cứu, và tổng thể đang chịu rủi ro. Sau đó họ sẽ tính số ca được trông đợi rồi so sánh với số ca quan sát được. Một nhóm sẽ được xác nhận là có tồn tại nếu tỉ lệ giữa số quan sát được / số trông đợi lớn hơn 1.0, và sự khác biệt là có ý nghĩa thống kê."

1. Giả sử rằng một căn bệnh ung thư nào đó có quan sát thấy 1 ca trong số 1000 người mỗi năm. Nếu bạn theo một nhóm 100 người nhất định trong suốt 10 năm, thì bạn trông đợi sẽ gặp 1 ca. Nếu bạn thấy 2 ca, thì điều đó cũng không quá ngạc nhiên, nhưng hơn 2 ca sẽ rất hiếm.Hãy viết một chương trình để mô phỏng một số đông các nhóm theo quãng thời gian 10 năm rồi ước tính phân bố của tổng số các ca.
2. Một quan sát được coi là có ý nghĩa về thống kê nếu xác suất của riêng nó, gọi là giá trị p, nhỏ hơn 5%. Trong một nhóm 100 người suốt 10 năm, bạn sẽ gặp bao nhiêu ca có tiêu chuẩn trên?
3. Bây giờ hãy tưởng tượng như bạn chia tổng thể 10000 người thành 100 nhóm rồi theo dõi họ suốt 10 năm. Có khả năng bao nhiêu là ít nhất một nhóm trong đó có cụm "có ý nghĩa thống kê" xuất hiện? Bao nhiêu nếu ta yêu cầu giá trị p bằng 1%.?
4. Bây giờ, tưởng tượng là bạn sắp xếp 10000 người vào một lưới gồm 100 × 100 ô rồi theo dõi họ trong vòng 10 năm. Có bao nhiêu phần khả năng có ít nhất một khối gồm 10 × 10 ô trong lưới ban đầu có một cụm với ý nghĩa thống kê?
5. Cuối cùng, hãy tưởng tượng là bạn dõi theo một lưới gồm 10000 người trong suốt 30 năm. Có bao nhiêu phần khả năng sẽ có một khoảng thời gian 10 năm trong đó tồn tại ít nhất một khối 10 × 10 ô trong lưới có cụm với ý nghĩa thống kê?

Định lý Bayes

Định lý Bayes phát biểu về mối liên hệ giữa các xác suất điều kiện của hai biến cố. Một xác suất điều kiện, thường được viết P(A|B) là xác suất để Biến cố A sẽ xảy ra khi ta biết rằng Biến cố B đã xảy ra rồi. Định lý Bayes phát biểu rằng:
$P(A|B) = P(B|A)\, P(A) / P(B)$
Để thấy được điều này là đúng, ta cần viết P(A và B), vốn là xác suất để cả A và B đều xuất hiện

P(A và B) = P(A) P(B|A)

Nhưng cũng đúng nếu viết

P(A và B) = P(B) P(A|B)

Vì vậy

P(B) P(A|B) = P(A) P(B|A)

Chia cả hai vế cho P(B) ta được định lý Bayes.[6] Định lý Bayes thường được diễn đạt như một khẳng định về cách thức mà một bằng chứng, E, làm ảnh hưởng đến xác suất của một giả thiết, H:
$P(H|E) = P(H)\, P(E|H) / P(E)$
Bằng lời nói, phương trình này phát biểu rằng xác suất của H sau khi bạn đã thấy E bằng tích của P(H), vốn bằng xác suất của H trước khi bạn thấy bằng chứng này, và tỉ số giữa P(E|H), xác suất của việc thấy bằng chứng với giả định rằng H đúng, với P(E), xác suất của việc thấy bằng chứng trong bất kì trường hợp nào (H đúng hoặc sai). Cách phát biểu này của định lý Bayes được gọi là diễn giải tính "thay đổi theo thời gian" vì nó mô tả xác suất của một giả thiết được **cập nhật** theo thời gian như thế nào, thường là từ quan điểm của bằng chứng mới. Ở đây, P(H) được gọi là xác suất **tiên nghiệm** còn P(H|E) được gọi là xác suất **hậu nghiệm**. P(E|H) là **độ đo khả năng** của bằng chứng, còn P(E) là **hằng số chuẩn hóa**. Một cách sử dụng kinh điển của định lý Bayes là việc diễn giải kết quả khám y khoa. Chẳng hạn, việc khám định kỳ để phát hiện trường hợp dùng thuốc bị cấm đang ngày càng phổ biến trong các công sở và trường học (Xem http://aclu.org/drugpolicy/testing.). Các công ty thực hiện những đợt kiểm tra này khẳng định rằng: phép kiểm tra rất nhạy, có nghĩa là rất dễ thu được kết quả dương tính nếu có thuốc (hay metabolite) trong mẫu; đồng thời cũng rất đặc hiệu, nghĩa là rất dễ nhận được kết quả âm tính nếu không có thuốc. Nghiên cứu tiến hành bởi Journal of the American Medical Association[7] ước tính được rằng các phép kiểm tra thuốc nói chung độ nhạy vào khoảng 60% và độ đặc hiệu khoảng 99%.

Bây giờ giả sử như các phép thử này được áp dụng cho một đoàn nhân viên trong đó tỉ lệ thật của việc dùng thuốc là 5%. Với những nhân viên có kết quả dương tính, sẽ có bao nhiêu người thực sự dùng thuốc?

Theo cách làm của Bayes, ta muốn tính xác suất của việc dùng thuốc khi biết trước phép thử dương tính, P(D|E). Định luật Bayes cho ta:
$P(D|E) = P(D)\, P(E|D) / P(E)$
Xác suất tiên nghiệm, P(D) là xác suất dùng thuốc trước khi ta thấy kết quả của kiểm tra, vốn bằng 5%. Độ đo khả năng, P(E|D), là xác suất của một kết quả dương tính khi có dùng thuốc, vốn chính là độ nhạy.

Hằng số chuẩn hóa, P(E) hơi khó lượng giá hơn một chút. Ta phải xét hai trường hợp, P(E|D) và P(E|*n*), trong đó *n* là giả thiết rằng người được kiểm tra không dùng thuốc:

P(E) = P(D) P(E|D) + P(*n*) P(E|*n*)

Xác suất của một kết quả dương tính nhầm, P(E|*n*), bằng phần bù của độ đặc hiệu, hay 1%. Ghép chúng lại với nhau, ta có
$P(D|E) = P(D)\, P(E|D) / [P(D)\, P(E|D) + P(N)\, P(E|N)]$

Thay các giá trị đã cho vào biểu thức này ta được P(D|E) = 0,76; điều đó có nghĩa là trong số những người dương tính sau khi kiểm tra, cứ khoảng 4 người thì có 1 người thật ra không dùng thuốc.

> Hãy viết một chương trình nhận vào tỉ lệ dùng thuốc thực sự, các độ nhạy và độ đặc hiệu của phép thử, rồi dùng định lý Bayes để tính P(D|E).
>
> Giả sử rằng phép thử này được áp dụng cho quần thể trong đó tỉ lệ dùng thuốc thực tế bằng 1%. Xác suất để một người nào đó với kết quả dương tính đúng là người đã dùng thuốc bằng bao nhiêu?
>
> Bài tập này được lấy từ `http://wikipedia.org/wiki/Bayesian_inference`.
>
> > "Giả sử rằng có hai bát đựng đầy bánh. Bát 1 có 10 bánh sô-cô-la và 30 chiếc bánh thường, trong khi Bát 2 có 20 chiếc bánh mỗi loại. Anh bạn Fred chọn một bát ngẫu nhiên, rồi chọn ngẫu nhiên một chiếc bánh. Hóa ra đây là một chiếc bánh thường. Hỏi khả năng Fred đã chọn bánh từ Bát 1 là bao nhiêu?"
>
> Các viên sô-cô-la M&M màu xanh lam được ra mắt vào năm 1995. Trước đó, tỉ lệ trộn lẫn sô-cô-la trong mỗi gói M&M là (30% Nâu, 20% Vàng, 20% Đỏ, 10% Xanh lục, 10% Cam, 10% Màu da). Sau này được đổi thành (24% Xanh lam, 20% Xanh lục, 16% Da cam, 14% Vàng, 13% Đỏ, 13% Nâu).
>
> Một cậu bạn tôi có hai gói M&M, và anh cho tôi biết rằng một gói từ năm 1994 và gói kia từ năm 1996. Anh không bảo tôi biết gói nào, mà đưa tôi một viên M&M từ mỗi túi. Một viên màu vàng và viên kia màu xanh lục. Hỏi xác suất để viên M&M màu vàng thuộc về túi năm 1994 bằng bao nhiêu?
>
> Bài tập này được chỉnh lại từ nguồn MacKay, *Information Theory, Inference, và Learning Algorithms*: Elvis Presley có người anh sinh đôi nhưng đã mất lúc vừa chào đời. Theo Wikipedia thì thông tin về các cặp sinh đôi như sau:
>
> > "Các cặp sinh đôi chiếm khoảng 1,9% số dân trên thế giới, trong đó sinh đôi cùng trứng chiếm khoảng 0,2% tổng số dân—hay 8% các cặp sinh đôi."
>
> Vậy xác suất để Elvis có người anh sinh đôi cùng trứng là bao nhiêu?

Thuật ngữ

biến cố:
: Điều có thể xảy ra hoặc không, với một xác suất nào đó.

phép thử:
: Một trong chuỗi các dịp mà một biến cố có thể xảy ra.

thành công:
: Phép thử trong đó một biến cố đã xảy ra.

thất bại:
: Phép thử trong đó không có biến cố nào xảy ra.

thuyết tần suất:
: Cách diễn giải chặt chẽ về xác suất chỉ áp dụng được với một dãy các phép thử đồng nhất.

thuyết Bayes:

Cách diễn giải tổng quát hơn có dùng đến xác suất để biểu diễn một mức độ tin cậy mang tính chủ quan.

độc lập:
: Hai biến cố được gọi là độc lập nếu sự xuất hiện của biến cố này không làm ảnh hưởng đến xác suất của biến cố kia.

hệ số biến thiên:
: Một đặc trưng thống kê nhằm tóm tắt độ phân tán, được chuẩn hóa theo xu thế trung tâm, để so sánh giữa các phân bố có trị trung bình khác nhau.

mô phỏng Monte Carlo:
: Phương pháp tính xác suất bằng cách mô phỏng những quá trình ngẫu nhiên (xem http://wikipedia.org/wiki/Monte_Carlo_method).

cập nhật:
: Quá trình dùng dữ liệu để tính lại một xác suất.

tiên nghiệm:
: Xác suất trước khi được cập nhật bằng định lý Bayes.

hậu nghiệm:
: Xác suất được tính theo định lý Bayes.

độ đo khả năng của bằng chứng:
: Một thuật ngữ trong định lý Bayes, đó là xác suất của bằng chứng đóng vai trò là điều kiện cho một giả thiết.

hằng số chuẩn hóa:
: Mẫu số của Định lý Bayes, được dùng để chuẩn hóa kết quả thành một xác suất.

---

1. Xem Hoffman, *The Man Who Loved Only Numbers*, page 83. ↩
2. Chẳng hạn, xem nguồn Gilovich, Vallone và Tversky, “The hot hvà in basketball: On the misperception of random sequences,” 1985. ↩
3. Xem http://wikipedia.org/wiki/Hitting_streak. ↩
4. Nguồn: http://cdc.gov/nceh/clusters/about.htm. ↩
5. Xem Gawvafe, “The Cancer Cluster Myth,” *New Yorker*, Feb 8, 1997. ↩
6. Xem http://wikipedia.org/wiki/Q.E.D.! ↩
7. Tôi lấy những con số này từ Gleason và Barnum, “Predictive Probabilities In Employee Drug-Testing,” tại http://piercelaw.edu/risk/vol2/winter/gleason.htm. ↩

Chương 6: Các phép toán đối với phân bố

Trở về Muc luc cuốn sách

Độ bất đối xứng

Độ bất đối xứng là một đặc trưng thống kê để đo mức bất đối xứng của một phân bố. Với một chuỗi giá trị cho trước, x_i, độ bất đối xứng của mẫu là:

$g_1 = m_3/m_2^{3/2}$

$m_2 = 1/n \sum_i (x_i - \mu)^2$

$m_3 = 1/n \sum_i (x_i - \mu)^3$

Bạn có thể nhận ra rằng m_2 là độ lệch bình phương trung bình (còn được gọi là phương sai); m_3 là độ lệch lập phương trung bình.

Độ bất đối xứng âm cho thấy một phân bố "lệch qua trái;" có nghĩa là nó trải dài sang phía trái nhiều hơn là sang phải. Độ bất đối xứng dương cho thấy một phân bố lệch qua phải.

Trên thực tế, việc tính toán độ bất đối xứng của một mẫu thường không phải là cách làm hay. Một điểm biệt lập, nếu có mặt, sẽ làm ảnh hưởng đến g_1 không theo tỉ lệ thông thường.

Một cách khác để đánh giá độ bất đối xứng của một phân bố là nhìn vào tương quan giữa trị trung bình và số trung vị. Các giá trị cực hạn sẽ có ảnh hưởng nhiều đến trị trung bình hơn là đến số trung vị, vì vậy nếu một phân bố bị lệch trái thì trị trung bình sẽ nhỏ hơn số trung vị.

Hệ số bất đối xứng trung vị của Pearson là một cách đo khác đối với độ bất đối xứng, trong đó thể hiện rõ quan hệ giữa trị trung bình, μ, và số trung vị, $\mu_{1/2}$:

$g_p = 3(\mu - \mu_{1/2})/\sigma$

Đặc trưng thống kê này **vững**, theo nghĩa nó ít bị ảnh hưởng bởi các điểm biệt lập.

> Hãy viết một hàm có tên `Skewness` để tính g_1 của một mẫu.
>
> Hãy tính độ bất đối xứng của các phân bố thời gian mang thai và cân nặng trẻ sơ sinh. Những kết quả này có thống nhất với hình dạng của các phân bố không?
>
> Hãy viết một hàm có tên `PearsonSkewness` để tính g_p cho những phân bố này. Hãy so sánh g_p với g_1.
>
> "Hiệu ứng Hồ Wobegon" là một tên đặt hài hước[1] cho **thói tự huyễn**, vốn là xu hướng của người đánh giá quá cao khả năng của bản thân so với người xung quanh. Chẳng hạn, một số cuộc khảo sát cho thấy trên 80% người được hỏi tin rằng họ lái xe tốt hơn các tài xế bình thường (xem http://wikipedia.org/wiki/Illusory_superiority).

Nếu ta diễn giải “bình thường” [chỉ khả năng lái xe] là trung vị, thì kết quả này về logic sẽ không thể tồn tại, nhưng nếu “bình thường” là trị trung bình, thì kết quả này vẫn có thể, mặc dù không dễ xảy ra.

Có mấy phần trăm dân số có nhiều hơn số chân trung bình?

Tổ chức Internal Revenue Service (IRS) của Hoa Kỳ cung cấp số liệu về thuế thu nhập và các con số thống kê khác, tại `http://irs.gov/taxstats`. Nếu đã làm Bài tập irs thì bạn đã từng làm việc với số liệu này. Nếu chưa, bạn có thể làm theo những hướng dẫn trong bài đó để lấy được các phân bố về thu thập từ bộ số liệu.

Có bao nhiêu phần dân số đã báo cáo thu nhập chịu thuế dưới mức trung bình?

Hãy tính số trung vị, trị trung bình, độ bất đối xứng và thông số này theo cách tính của Pearson, đối với số liệu thu nhập. Vì số liệu đã được chia ngăn, bạn sẽ phải tính gần đúng.

Hệ số Gini là một độ đo mức bất bình đẳng về thu nhập. Bạn có thể tìm hiểu ở `http://wikipedia.org/wiki/Gini_coefficient` rồi viết một hàm có tên `Gini` để tính hệ số này cho phân bố thu nhập.

Gợi ý: dùng PMF để tính độ khác biệt trung bình tương đối (xem `http://wikipedia.org/wiki/Mean_difference`).

Bạn có thể tải về một lời giải cho bài tập này từ `http://thinkstats.com/gini.py`.

Biến ngẫu nhiên

Biến ngẫu nhiên biểu diễn một quá trình phát sinh ra một số ngẫu nhiên. Biến ngẫu nhiên thường được kí hiệu bằng chữ in, như *X*. Khi bạn thấy một biến như vậy, bạn có thể hình dung là “một giá trị được lựa chọn từ một phân bố.”

Chẳng hạn, định nghĩa chính thức của hàm phân bố lũy tích là:

$\text{CDF}_X(X) = \Pr(X \le X)$

Tôi đã tránh cách viết này suốt đến giờ vì nó rất xấu, nhưng cách hiểu của nó là thế này: CDF của một biến ngẫu nhiên *X*, được tính tại một giá trị *X* cụ thể, thì được định nghĩa bằng xác suất mà một giá trị được phát sinh bởi quá trình ngẫu nhiên *X* nhỏ hơn hoặc bằng *X*.

Là một nhà khoa học máy tính, tôi đã thấy rất có ích khi hình dung một biến ngẫu nhiên như một đối tượng có một phương thức, tạm gọi là `generate`; phương thức này dùng một quá trình ngẫu nhiên để phát sinh ra giá trị.

Chẳng hạn, sau đây là một định nghĩa lớp biểu thị các biến ngẫu nhiên:

```
class RandomVariable(object):
    """Parent class for all random variables."""
```

Và sau đây là một biến ngẫu nhiên với một phân bố lũy thừa:

```
class Exponential(RandomVariable):
```

```
    def __init__(self, lam):
        self.lam = lam

    def generate(self):
        return random.expovariate(self.lam)
```

Phương thức khởi tạo đã nhận vào tham biến, λ, rồi lưu nó dưới dạng thuộc tính. Phương thức `generate` trả lại một giá trị ngẫu nhiên từ phân bố lũy thừa với tham số đó.

Mỗi lần gọi `generate`, bạn sẽ nhận được một giá trị khác trước. Giá trị mà bạn nhận được gọi là **giá trị ngẫu nhiên**, điều đó giải thích tại sao nhiều tên hàm trong module `random` lại có chữ "variate" [biến].

Nếu chỉ cần phát sinh các biến ngẫu nhiên theo phân bố lũy thừa, tôi sẽ không bận tâm định nghĩa một lớp khác, mà dùng ngay `random.expovariate`. Nhưng với các phân bố khác, có thể sẽ cần dùng các đối tượng RandomVariable. Chẳng hạn, phân bố Erlang là một phân bố liên tục với các tham số là λ và *k* (xem http://wikipedia.org/wiki/Erlang_distribution).

Một cách phát sinh các giá trị từ phân bố Erlang là thêm vào *k* giá trị từ mộ phân bố lũy thừa có cùng λ. Sau đây là một cách viết:

```
class Erlang(RandomVariable):
    def __init__(self, lam, k):
        self.lam = lam
        self.k = k
        self.expo = Exponential(lam)

    def generate(self):
        total = 0
        for i in range(self.k):
            total += self.expo.generate()
        return total
```

Phương thức khởi tạo đã lập nên một đối tượng Exponential với tham số đã cho; sau đó phương thức `generate` sử dụng nó. Nói chung, phương thức khởi tạo có thể nhận một bộ tham số bất kì và hàm `generate` có thể thực hiện bất kì quá trình ngẫu nhiên nào.

> Hãy viết định nghĩa một lớp để biểu thị một biến ngẫu nhiên theo phân bố Gumbel (xem http://wikipedia.org/wiki/Gumbel_distribution).

Hàm mật độ xác suất (PDF)

Đạo hàm của một CDF được gọi là **hàm mật độ xác suất**, hay PDF (probability density function). Chẳng hạn, PDF của một phân bố lũy thừa là

$$PDF_{LT}(x) = \lambda e^{-\lambda x}$$

PDF của phân bố chuẩn là

Việc lượng giá PDF cho một giá trị cụ thể của *X* thường không mấy hữu ích. Kết quả không phải là một xác suất mà là một *mật độ* xác suất.

Trong vật lý, mật độ là khối lượng có trong một đơn vị thể tích. Để có được khối lượng, bạn bải đem nhân mật độ với thể tích, hay nếu mật độ không phải hằng số, thì cần lấy tích phân trên thể tích.

Tương tự, mật độ xác suất là số đo xác suất trên mỗi đơn vị của *X*. Để có được khối xác suất,[2] bạn phải lấy tích phân theo *X*. Chẳng hạn, nếu *X* là một biến ngẫu nhiên có PDF là PDF $_X$, ta có thể tính được xác suất để một giá trị từ *X* rơi vào khoảng giữa –0,5 và 0,5:

$$P(-0{,}5 \le X < 0{,}5) = \int_{-0{,}5}^{0{,}5} PDF_X(x)dx$$

Hoặc, vì CDF chính là tích phân của PDF, ta có thể viết

$$P(-0{,}5 \le X < 0{,}5) = CDF_X(0{,}5) - CDF_X(-0{,}5)$$

Với một số dạng phân bố, ta có thể tính trực tiếp CDF và do vậy sẽ dùng cách thứ hai. Nếu không, ta thường phải lấy tích phân PDF bằng cách số trị.

> Tính xác suất để một giá trị chọn sẵn từ phân bố lũy thừa với tham số λ rơi vào giữa 1 và 20? Hãy biểu diễn đáp số dưới dạng một hàm số phụ thuộc λ. Hãy ghi lại kết quả này; ta sẽ còn dùng nó ở Mục số liệu kiểm duyệt.
>
> Trong BRFSS (xem Mục lognormal), phân bố chiều cao chỉ gần như dạng chuẩn với các tham số μ = 178 cm và σ^2 = 59,4 cm đối với nam giới; μ = 163 cm và σ^2 = 52,8 cm đối với nữ giới.
>
> Để gia nhập được Nhóm Blue Man, bạn phải là nam giới có chiều cao giữa 5’10’’ và 6’1’’ (xem http://bluemancasting.com). Có bao nhiêu phần trăm nam giới Mỹ có chiều cao trong khoảng này? Gợi ý: xem Mục [phân bố chuẩn].

Tích chập

Giả sử ta có hai biến ngẫu nhiên, *X* và *Y*, với các phân bố CDF $_X$ và CDF $_Y$. Khi đó phân bố của tổng *Z* = *X* + *Y* sẽ như thế nào?

Một cách làm là viết một đối tượng RandomVariable để phát sinh ra tổng:

```
class Sum(RandomVariable):
    def __init__(X, Y):
        self.X = X
        self.Y = Y

    def generate():
        return X.generate() + Y.generate()
```

Cho trước các RandomVariable bất kì, *X* và *Y*, ta có thể tạo ra đối tượng Sum để biểu diễn cho *Z*. Sau đó ta có thể dùng một mẫu từ *Z* để tính xấp xỉ CDF $_Z$.

Cách làm này đơn giản và linh hoạt, nhưng không hiệu quả lắm; ta phải phát sinh một mẫu lớn để có thể ước tính chính xác được CDF $_Z$, và ngay cả khi đó cũng không nhận được giá trị đúng.

Nếu CDF $_X$ và CDF $_Y$ được biểu diễn dưới dạng các hàm, thì đôi khi ta có thể tìm được đúng CDF $_Z$. Sau đây là cách làm:

1. Để bắt đầu, hãy giả sử rằng giá trị cụ thể của *X* là *X*. Khi đó CDF $_Z$(*Z*) là$P(Z \le z|X=x) = P(Y \le z–x)$

   Ta hãy đọc lại biểu thức này. Vế trái là "xác suất để tổng nhỏ hơn *Z*, khi cho trước số hạng đầu là *X*." À, nếu số hạng đầu là *X* và tổng phải nhỏ hơn *Z*, thì số hạng thứ hai phải nhỏ hơn *Z* – *X*.

2. Muốn có được xác suất để *Y* nhỏ hơn *Z* – *X*, ta đi tính CDF $_Y$.$P(Y \le z–x) = CDF_Y(z–x)$

   Điều này được rút ra từ định nghĩa của CDF.

3. Vẫn ổn chứ? Ta hãy tiếp tục nhé. Vì thực ra chúng ta vẫn chưa biết giá trị của *X*, nên ta cần phải xét tất cả các giá trị có thể nhận và lấy tích phân trên khắp khoảng giá trị đó:$P(Z \le z) = \int_{-\infty}^{\infty} P(Z \le z|X=x) PDF_X(x) dx$

   Biểu thức lấy tích phân là "xác suất để *Z* nhỏ hơn hoặc bằng *Z*, khi đã cho *X* = *X*, nhân với xác suất để *X* = *X*."

   Thay thế kết quả từ các bước tính trước ta có

   $P(Z \le z) = \int_{-\infty}^{\infty} CDF_Y(z–x) PDF_X(x) dx$

   Vế trái là định nghĩa của CDF $_Z$, vì vậy ta đi đến kết luận:

   $CDF_Z(z) = \int_{-\infty}^{\infty} CDF_Y(z–x) PDF_X(x) dx$

4. Để có được PDF $_Z$, hãy lấy đạo hàm của cả hai vế theo *Z*. Kết quả là$PDF_Z(z) = \int_{-\infty}^{\infty} PDF_Y(z–x) PDF_X(x) dx$

   Nếu bạn đã học môn tín hiệu và hệ thống, bạn có thể nhận ra tích phân này. Đó là **tích chập** của PDF $_Y$ và PDF $_X$, được biểu thị bởi toán tử *.

   PDF $_Z$ = PDF $_Y$ * PDF $_X$

   Như vậy phân bố của tổng là tích chập của các phân bố. Hãy xem http://wiktionary.org/wiki/booyah!

Lấy ví dụ, giả sử *X* và *Y* là các biến ngẫu nhiên tuân theo phân bố lũy thừa với tham số λ. Phân bố của *Z* = *X* + *Y* là:

$PDF_Z(z) = \int_{-\infty}^{\infty} PDF_X(x) PDF_Y(z–x) dx = \int_{-\infty}^{\infty} \lambda e^{-\lambda x} \lambda e^{\lambda(z–x)}$

Bây giờ ta phải nhớ rằng PDF_{LT} bằng 0 với mọi giá trị âm, nhưng ta có thể xử lý điều này bằng cách điều chỉnh các giới hạn của tích phân:

$PDF_Z(z) = \int_0^z \lambda e^{-\lambda x} \lambda e^{-\lambda(z–x)} dx$

Bây giờ ta có thể kết hợp các số hạng và chuyển hằng số ra ngoài dấu tích phân:

$PDF_Z(z) = \lambda^2 e^{-\lambda z} \int_0^z dx = \lambda^2 z e^{-\lambda z}$

Hóa ra đây là PDF của một phân bố Erlang với tham số $k = 2$ (xem http://wikipedia.org/wiki/Erlang_distribution). Như vậy tích chập của hai phân bố lũy thừa (với cùng giá trị tham số) là một phân bố Erlang.

> Nếu X có phân bố lũy thừa với tham số λ, còn Y có phân bố Erlang với các tham số k và λ, thì phân bố của tổng $Z = X + Y$ sẽ thế nào?
>
> Giả sử tôi lấy hai giá trị từ cùng một phân bố thì phân bố của giá trị lớn hơn là gì? Hãy biêu diễn đáp số của bạn dưới dạng PDF hoặc CDF của phân bố tìm được.
>
> Khi số giá trị tăng lên thì phân bố của giá trị lớn nhất sẽ hội tụ về một trong số các phân bố cực trị; xem http://wikipedia.org/wiki/Gumbel_distribution.
>
> Nếu có sẵn các đối tượng Pmf, bạn có thể tính được phân bố của tổng bằng cách liệt kê hết các cặp giá trị:
>
> ```
> for x in pmf_x.Values():
>     for y in pmf_y.Values():
>         z = x + y
> ```
>
> Hãy viết một hàm nhận vào PMF $_X$ và PMF $_Y$ rồi trả về một Pmf mới biểu thị cho phân bố của tổng $Z = X + Y$.
>
> Hãy viết một hàm tương tự để tính PMF của $Z = \max(X, Y)$.

Tại sao cần có phân bố chuẩn?

Như tôi đã nói trước đây, phân bố chuẩn rất tiện cho việc phân tích nhưng chưa giải thích tại sao. Một lý do là phân bố này có dạng khép kín dưới phép biến đổi tuyến tính và tích chập. Để nói rõ hơn, trước hết tôi sẽ giới thiệu một số kí hiệu.

Nếu phân bố của một biến ngẫu nhiên, X, có dạng chuẩn với các tham số μ và σ, thì bạn có thể viết

$X \sim N(\mu, \sigma)$

trong đó ký hiệu ~ nghĩa là "tuân theo phân bố" và chữ cái N là viết tắt cho "normal" (chuẩn).

Một phép biến đổi tuyến tính của X có dạng kiểu như $X' = a X + b$, trong đó a và b là các số thực.

Một họ phân bố được gọi là khép kín trong phép biến đổi nếu như X' có cùng họ với X. Phân bố chuẩn thỏa mãn tính chất này; nếu $X \sim N(\mu, \sigma^2)$, thì

$X' \sim N(a\,\mu + b, a^2\sigma)$

Các phân bố chuẩn cũng khép kín trong phép tích chập. Nếu $Z = X + Y$, $X \sim N(\mu_X, \sigma_X^2)$ và $Y \sim N(\mu_Y, \sigma_Y^2)$ thì

$Z \sim N(\mu_X + \mu_Y, \sigma_X^2 + \sigma_Y^2)$

Các dạng phân bố khác mà ta đã xét đến không có các tính chất như vậy.

> Nếu $X \sim N(\mu_X, \sigma_X^2)$ và $Y \sim N(\mu_Y, \sigma_Y^2)$, thì phân bố của $Z = aX + bY$ sẽ là gì?
>
> Ta hãy xem điều gì sẽ xảy ra khi thêm các giá trị vào từ những phân bố khác. Chọn một cặp phân bố (bất kì hai phân bố nào trong các dạng lũy thừa, chuẩn, loga chuẩn, và Pareto) rồi chọn các tham số để cho chúng có trị trung bình như nhau và phương sai như nhau.
>
> Hãy phát sinh các số ngẫu nhiên từ các phân bố này rồi tính phân bố của tổng của chúng. Hãy dùng các phép thử ở Chương continuous để xem rằng liệu tổng này có thể được mô phỏng bởi một phân bố liên tục hay không.

Định lý giới hạn trung tâm

Cho đến giờ ta đã thấy được:

- Nếu ta cộng các giá trị được lấy từ các dạng phân bố khác nhau, thì phân bố của giá trị tổng có dạng chuẩn.
- Nếu ta cộng các giá trị được lấy từ các dạng phân bố khác thì tổng thu được nói chung sẽ không giống như dạng phân bố mà ta đã từng thấy.

Nhưng hóa ra là nếu ta cộng lại một rất nhiều các giá trì từ bất kì dạng phân phối nào, thì tổng thu được cũng sẽ hội tụ về phân bố chuẩn.

Cụ thể hơn, nếu phân bố của các giá trị số hạng có trung bình và độ lệch chuẩn lần lượt là μ và σ, thì phân bố của tổng sẽ xấp xỉ là $N(n\,\mu, n\,\sigma^2)$.

Đây được gọi là **Định lý giới hạn trung tâm**. Nó là một trong những công cụ có ích nhất cho việc phân tích thống kê, nhưng cũng cần lưu ý một số điểm sau:

- Các giá trị rút ra phải độc lập.
- Các giá trị này phải được rút từ cùng một dạng thống kê (Cho dù điều này có thể được châm trước).
- Các giá trị phải được rút từ một phân bố có trị trung bì và phương sai hữu hạn, vì thế phần lớn các phân bố Pareto đều bị loại trừ.
- Số các giá trị cần đến thiết để đạt được sự hội tụ thì phụ thuộc vào độ bất đối xứng của phân bố. Tổng của các phần tử từ phân bố lũy thừa sẽ hội tụ ngay cả với mẫu kích thước nhỏ. Còn tổng từ phân bố loga chuẩn thì không.

Định lý giới hạn trung tâm giải thích rằng, ít nhất có phần đúng là, phân bố chuẩn chiếm chỗ quan trọng trong giới tự nhiên. Phần lớn các đặc điểm của động vật và những sinh vật khác đều bị ảnh hưởng bởi nhiều yếu tố gien và môi trường mà hiệu ứng của chúng có tính cộng. Các đặc điểm mà ta đo là tổng của rất nhiều những hiệu ứng nhỏ, vì vậy sự phân bố của chúng có xu thế giống như dạng chuẩn.

> Nếu tôi rút ra một mẫu, $x_1 .. x_n$, độc lập từ một phân bố với trị trung bình μ và phương sai σ^2 đều hữu hạn, thì phân bố của trị trung bình của mẫu,

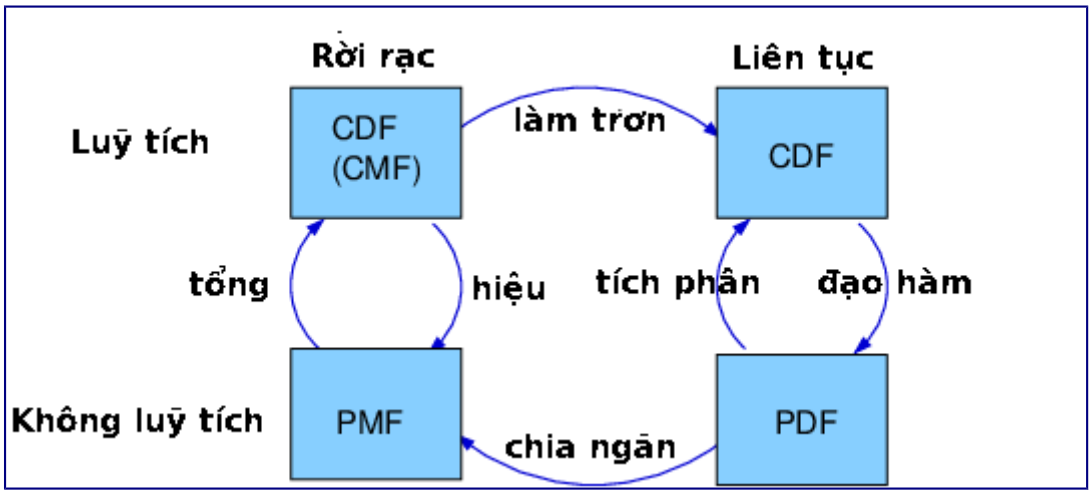
$= (1/n)\sum x_i$

sẽ như thế nào? Khi *n* tăng lên, điều gì sẽ xảy ra với phương sai của trị trung bình mẫu? Gợi ý: hãy xem lại Mục [tại sao cần phân bố chuẩn].

Hãy chọn một phân bố (trong số các phân bố lũy thừa, loga chuẩn, hoặc Pareto) và chọn ra các giá trị cho (các) tham số. Hãy phát sinh các mẫu với kích cỡ 2, 4, 8, v.v., rồi xác định phân bố của tổng các mẫu. Hãy dùng một đồ thị xác suất chuẩn để xem rằng liệu có phải phân bố xấp xỉ với dạng chuẩn hay không. Bạn phải thêm vào bao nhiêu số hạng để thấy được sự hội tụ?

Thay vì phân bố của tổng, hãy xác định các dạng phân bố của tích; điều gì sẽ xảy ra khi tăng số thừa số lên? Gợi ý: hãy xét đến phân bố các loga của tích.

Sơ đồ phân bố



Một sơ đồ liên hệ giữa các dạng biểu diễn của hàm phân bố.

Đến lúc này ta đã gặp các PMF, CDF và PDF; ta hãy cùng dành một phút để ôn lại. Hình dist_framework cho thấy cách liên hệ giữa các hàm này.

Chúng ta bắt đầu từ PMF, vốn biểu diễn cho tần suất trong một tập hợp các giá trị rời rạc. Để thu được CDF từ PMF, ta đi tính tổng lũy tích. Lẽ ra để cho thống nhất, một CDF rời rạc phải được gọi là hàm khối lũy tích (CMF), nhưng theo những gì tôi được biết đến thì không ai dùng thuật ngữ như vậy.

Để tính được PMF từ CDF, bạn phải tính các hiệu số giữa các xác suất lũy tích.

Một các tương tự, PDF là đạo hàm của CDF liên tục; hoặc theo cách nói tương đương, CDF là tích phân của PDF. Nhưng cần nhớ rằng một PDF cho ánh xạ từ giá trị đến mật độ xác suất; để thu được xác suất, bạn phải tính tính phân.

Để đi từ phân bố rời rạc đến liên tục, bạn có thể thực hiện một số phương pháp làm trơn. Một cách làm trơn là giả sử rằng số liệu thu được từ một phân bố liên tục có công thức giải tích (như phân bố lũy thừa hoặc chuẩn) và ước tính các tham số của phân bố đó. Và đây chính là điều mà Chương ước lượng tham số đề cập đến.

Nếu bạn chia một PDF vào một dãy các ngăn, bạn có thể phát sinh ra một PMF ít nhất xấp xỉ được PDF đó. Chúng ta có thể dùng kĩ thuật này trong Chương estimation để ước tính Bayes.

> Hãy viết một hàm có tên `MakePmfFromCdf` nhận vào một đối tượng Cdf, rồi trả về đối tượng Pmf tương ứng.
>
> Bạn có thể tìm thấy một lời giải bài tập này trong `thinkstats.com/Pmf.py`.

Thuật ngữ

độ bất đối xứng:
: Đặc tính của một phân bố; theo trực giác, đó là độ đo mất cân đối của hình dạng phân bố.

vững:
: Một đặc trưng thống kê được gọi là vững khi nó ít chịu ảnh hưởng từ sự có mặt của điểm biệt lập.

thói tự huyễn:
: Xu thế trong đó con người tưởng tượng rằng mình tốt hơn người bình thường xung quanh.

biến ngẫu nhiên:
: Đối tượng biểu thị cho một quá trình ngẫu nhiên.

trị ngẫu nhiên:
: Giá trị được phát sinh từ một quá trình ngẫu nhiên.

PDF:
: (probability density function) Hàm mật độ xác suất, đạo hàm của một CDF liên tục.

tích chập:
: Phép toán để tính ra phân bố của tổng các giá trị từ hai dạng phân bố khác.

Định lý giới hạn trung tâm:
: "Định luật tối thượng của sự lười suy nghĩ," theo Sir Francis Galton, một nhà thống kê học thời trước.

---

1. Nếu bạn vẫn chưa hiểu, hãy xem http://wikipedia.org/wiki/Lake_Wobegon. ↩
2. Để mở rộng sự tương tự này, cần lưu ý rằng trị trung bình của một phân bố là trọng tâm của nó, còn phương sai là mô-men quán tính. ↩

Chương 7: Kiểm định giả thiết

Trở về Muc luc cuốn sách

Khi khám phá số liệu của NSFG, chúng ta đã thấy một vài "hiệu ứng biểu kiến", gồm một số khác biệt giữa trẻ đầu lòng và trẻ sinh sau. Đến giờ ta đã mới chỉ xét qua những hiệu ứng đó; trong chương này, rốt cuộc chúng ta sẽ kiểm tra chúng.

Câu hỏi cơ bản mà chúng ta muốn đặt ra là liệu những hiệu ứng đó có thật hay không. Chẳng hạn, nếu ta thấy có sự chênh lệch giữa giá trị trung bình của thời gian mang thai những đứa trẻ đầu lòng với trẻ sinh sau, ta muốn biết rằng liệu sự chênh lệch đó có thực không, hay chúng chỉ tình cờ xảy ra.

Câu hỏi này hóa ra lại rất khó trả lời trực tiếp, vì vậy ta sẽ tiến hành theo hai bước. Trước tiên, ta sẽ kiểm tra xem liệu rằng hiệu ứng **có ý nghĩa** hay không, sau đó ta sẽ cố gắng diễn giải kết quả để trả lời câu hỏi đặt ra ban đầu.

Trong thống kê học, "có ý nghĩa" mang một định nghĩa riêng khác với ý nghĩa thông trong câu nói hằng ngày. Như ta đã định nghĩa trước đây, một hiệu ứng biểu kiến được gọi là có ý nghĩa thống kê nếu như nó không có vẻ xuất hiện tình cờ.

Để làm rõ điều này, ta sẽ phải trả lời ba câu hỏi sau:

1. Thế nào là "tình cờ"?
2. Thế nào là "không có vẻ"?
3. Thế nào là "hiệu ứng"?

Cả ba câu hỏi này trông thì dễ như trả lời được lại khó. Dù sao, vẫn có một cấu trúc chung mà mọi người dùng để kiểm định ý nghĩa thống kê:

Giả thiết không:
: **Giả thiết không** là một mô hình của một hệ thống được dựa trên giả thiết rằng hiệu ứng biểu kiến thực ra chỉ tình cờ xảy đến.

Giá trị p:
: **Giá trị p** là xác suất của một hiệu ứng biểu kiến theo giả thiết không.

Diễn giải:
: Dựa trên giá trị p, chúng ta kết luận rằng hiệu ứng hoặc là có ý nghĩa thống kê, hoặc không.

Quy trình này được gọi là **kiểm định giả thiết**. Logic bên trong nó cũng giống như việc chứng minh bằng phản bác. Để chứng minh một phát biểu toán học, A, bạn tạm thời giả sử rằng A sai. Nếu như giả sử đó dẫn đến một nghịch lý thì bạn sẽ kết luận rằng thực ra A phải đúng.

Tương tự, để kiểm định một giả thiết như "Hiệu ứng này là thật," chúng ta tạm giả sử rằng không phải vậy. Đó là giả thiết không. Dựa theo giả thiết đó, ta đi tính xác suất của hiệu ứng biểu kiến. Đó là giá trị p. Nếu giá trị p đủ thấp thì ta sẽ kết luận rằng giả thiết không có vẻ như không đúng.

Kiểm định sự chênh lệch các trị trung bình

Một trong những giả thiết dễ nhất để kiểm tra là có sự khác biệt trông thấy giữa hai trị trung bình của hai nhóm. Từ số liệu NSFG, ta thấy rằng trị trung bình của thời gian mang thai đứa trẻ đầu lòng

thấp hơn một chút, và trị trung bình cân nặng trẻ sơ sinh cũng thấp hơn một chút. Bây giờ ta sẽ xét xem liệu rằng những hiệu ứng đó có ý nghĩa không.

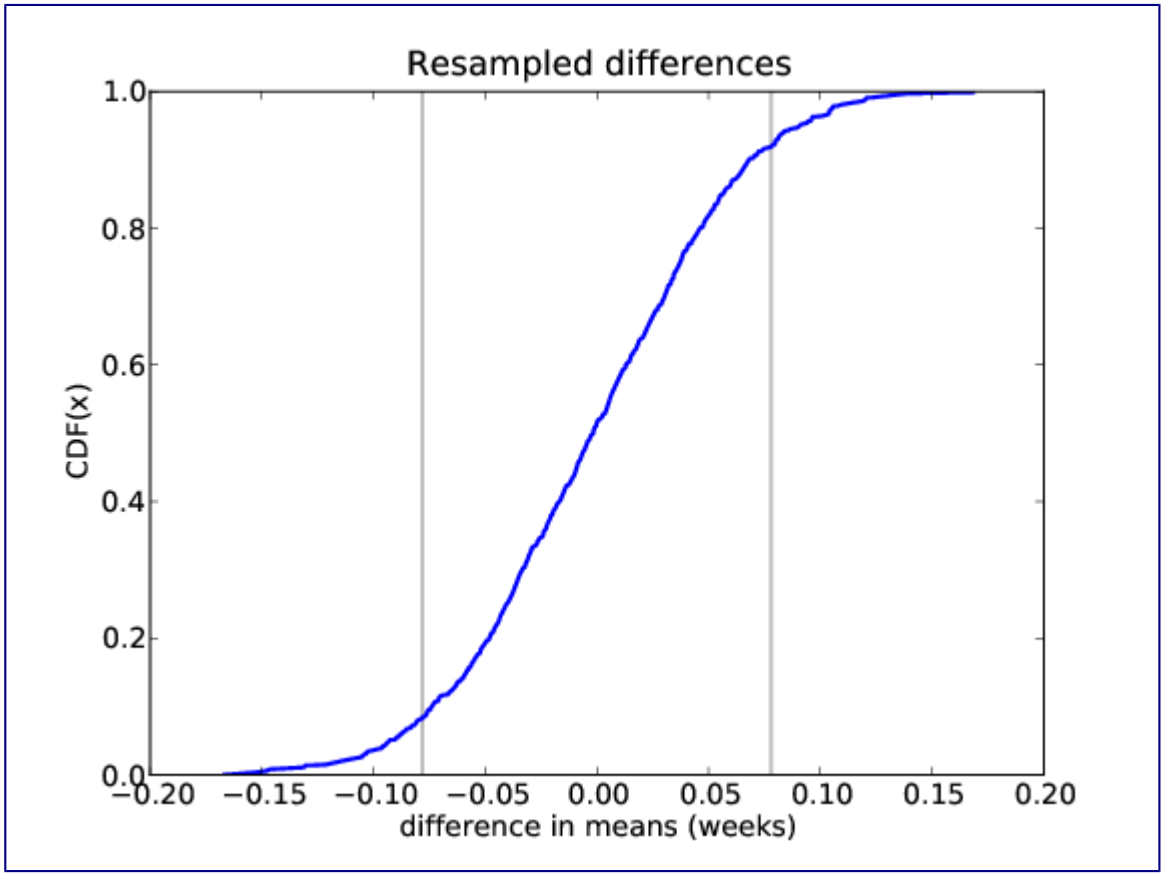
Ở những ví dụ này, giả thiết không là các phân bố của hai nhóm là giống nhau, và sự khác biệt thấy được chỉ là tình cờ.

Để tính các giá trị p, ta tìm phân bố gộp chung của tất cả các ca sinh thành công (kể cả trẻ đầu lòng lẫn sinh sau), phát sinh các mẫu ngẫu nhiên có cùng kích thước với mẫu quan sát được, rồi tính hiệu số giữa các trị trung bình theo giả thiết không.

Nếu phát sinh được rất nhiều mẫu, ta có thể tìm được tần suất của việc hiệu số giữa các trị trung bình (chỉ do tình cờ) lớn bằng hoặc lớn hơn hiệu số mà ta thực sự quan sát được. Tỉ lệ này được gọi là giá trị p.

Đối với thời kì mang thai, chúng ta đếm được n = 4413 trẻ đầu lòng và m = 4735 trẻ sinh sau, cùng với hiệu số giữa các trị trung bình của thời gian mang thai là δ = 0,078 tuần. Để tính xấp xỉ giá trị p cho hiệu ứng này, tôi đã gộp chung các phân bố, phát sinh ra các mẫu có kích thước n và m rồi tính hiệu số giữa hai trị trung bình.

Đây là một ví dụ khác cho việc tái lập mẫu, vì ta đang rút một mẫu ngẫu nhiên từ bộ số liệu mà bản thân nó là một mẫu của tổng thể chung. Tôi đã tính hiệu số giữa 1000 cặp mẫu; Hình [length_deltas_cdf] cho thấy phân bố của hiệu số này.



CDF của các hiệu số về trị trung bình của các mẫu đã tái lập. [length_deltas_cdf]

Hiệu số về trị trung bình gần bằng 0, cũng như bạn trông đợi với các mẫu rút từ cùng một phân bố. Hai đường thẳng đứng biểu thị cho các giới hạn cắt đứt nơi $X = \delta$ hoặc $X = \delta$.

Trong số 1000 cặp mẫu, có 166 cặp mà sự khác biệt về trị trung bình (cả âm lẫn dương) lớn hơn hoặc bằng δ, vì vậy giá trị p xấp xỉ bằng 0,166. Nói cách khác, chúng ta trông đợi thấy được hiệu ứng khác biệt ở mức độ δ vào khoảng 17% trong số các trường hợp xảy ra, ngay cả nếu các phân bố thực sự của hai nhóm là giống nhau.

Như vậy hiệu ứng biểu kiến dường như không rõ, nhưng liệu nó có đủ để khẳng định rằng không rõ? Ta sẽ giải đáp điều này trong mục sau.

> Trong bộ số liệu NSFG, sự khác biệt về cân nặng trung bình của trẻ đầu lòng là 2,0 ounce. Hãy tính giá trị p của hiệu số này.
>
> Gợi ý: với kiểu tái lập mẫu này, điều quan trọng là phải lấy mẫu có thay thế, vì vậy bạn nên dùng `random.choice` thay vì `random.sample` (xem Mục [ngẫu nhiên]).
>
> Bạn có thể bắt tay với mã lệnh mà tôi đã dùng để phát sinh ra kết quả trong mục này, mà có thể được tải về từ `http://thinkstats.com/hypothesis.py`.

Chọn một ngưỡng giá trị

Trong kiểm định thống kê có hai loại lỗi mà ta cần phải lưu tâm.

- Lỗi Loại I, còn được gọi là **dương tính sai** (false positive), xảy ra khi chúng ta chấp nhận một giả thiết mà thực ra nó sai; nghĩa là chúng ta coi một hiệu ứng là có ý nghĩa trong khi thực tế nó chỉ xảy ra tình cờ.
- Lỗi Loại II, còn được gọi là **âm tính sai** (false negative), xảy ra khi chúng ta bác bỏ một giả thiết mà thực ra nó đúng; nghĩa là chúng ta quy kết một hiệu ứng là tình cờ trong khi nó có thật.

Cách tiếp cận thông dụng nhất tới kiểm định thống kê là chọn một ngưỡng,[1], α, cho giá trị p và chấp nhận rằng bất kì hiệu ứng nào có giá trị p nhỏ hơn α đều là có ý nghĩa. Một giá trị của α thường được chọn là 5%. Theo tiêu chí này, sự khác biệt biểu kiến ở thời gian sinh trẻ đầu lòng là không có ý nghĩa, nhưng khác biệt về cân nặng thì có.

Với kiểu kiểm định giả thiết như thế này, ta có thể tính ngay được xác suất của một false positive: nó đúng bằng α.

Để thấy được tại sao, hãy hình dung định nghĩa của false positive— khả năng chấp nhận một giả thiết bị sai—và định nghĩa của giá trị p—khả năng của việc tạo ra hiệu ứng đo được nếu như giả thiết là sai.

Kết hợp hai điều này lại, ta có thể hỏi: nếu như giả thiết là sai thì có bao nhiêu khả năng phát sinh ra một hiệu ứng đo được có thể xem như là ý nghĩa với ngưỡng α? Câu trả lời chính là α.

Ta có thể làm giảm khả năng xuất hiện false positive bằng cách giảm ngưỡng đi. Chẳng hạn, nếu ngưỡng bằng 1%, thì chỉ còn 1% khả năng có false positive.

Nhưng có một cái giá phải trả: việc hạ thấp ngưỡng sẽ làm tăng tiêu chuẩn của bằng chứng, tức là làm tăng khả năng bác bỏ một giả thiết đúng. Nhìn chung luôn có một sự đánh đổi giữa các sai số Loại I và Loại II. Cách duy nhất để giảm cả hai loại sai số này cùng lúc là tăng kích thước mẫu (hoặc, trong một số trường hợp, là giảm sai số đo đạc).

Để tìm hiểu ảnh hưởng của kích thước mẫu đến giá trị p, ta hãy xem điều gì sẽ xảy ra nếu bỏ đi một nửa các số liệu NSFG. Gợi ý: dùng `random.sample`. Điều gì sẽ xảy ra nếu bỏ đi 3 phần tư số liệu, và cứ như vậy?

Kích thước mẫu nhỏ nhất phải là bao nhiêu để sự khác biệt giữa các trị trung bình của cân nặng trẻ sơ sinh vẫn còn ý nghĩa với α = 5%? Kích thước mẫu phải tăng gấp mấy lần để có α = 1%?

Bạn có thể bắt đầu làm với mã lệnh mà tôi đã dùng để phát sinh các kết quả trong mục này; nó có thể được tải về từ `http://thinkstats.com/hypothesis.py`.

Định nghĩa về hiệu ứng

Khi một điều bất thường nào đó xảy ra, mọi người thường nói kiểu như, “Ồ! Sao có thể *như thế* được nhỉ?” Câu hỏi này có lý vì chúng ta có trực giác rằng những việc khác nhau sẽ có khả năng xảy ra không như nhau. Nhưng trực giác này không phải luôn đúng khi ta phân tích nó kĩ lưỡng.

Chẳng hạn, giả sử tôi tung đồng xu 10 lần, và sau mỗi lần tung tôi ghi lại N để kí hiệu đồng xu rơi ngửa và S để kí hiệu sấp. Nếu kết quả là một chuỗi như SNNSNSSSNN, bạn sẽ chẳng lấy làm ngạc nhiên. Nhưng nếu kết quả mà là NNNNNNNNNN, bạn sẽ thốt lên lời nói đại loại như, “Ồ! Sao có thể *như thế* được nhỉ?”

Nhưng ở ví dụ này, xác suất của hai dãy kết quả là như nhau: 1 trên 1024. Và điều này cũng đúng với một chuỗi bất kì khác. Vì vậy khi ta hỏi, “Sao có thể *như thế* được nhỉ?”, ta phải cẩn thận với việc dùng “như thế” với hàm ý gì.

Với số liệu NSFG, tôi định nghĩa hiệu ứng là “một hiệu số về trị trung bình (kể cả dương lẫn âm) lớn hơn hoặc bằng δ.” Bằng lựa chọn này, tôi đã quyết định tính độ lớn của hiệu số, mà bỏ qua không xét đến dấu.

Một cách kiểm định như vậy được gọi là **hai phía**, vì ta xét đến cả hai phía (theo chiều dương và âm) trong phân bố ở Hình length_deltas_cdf. Bằng cách dùng phép kiểm định hai mặt, ta thực hiện kiểm tra giả thiết về sự khác biệt, hay hiệu số, đáng kể giữa các phân bố, mà không nói về dấu của hiệu số này.

Một cách làm khác là dùng kiểm định **một phía**, trong đó câu hỏi đặt ra là liệu trị trung bình đối với trẻ đầu lòng có *cao hơn* đáng kể so với trị trung bình của các trẻ sinh sau không. Vì giả thiết này đã cụ thể hơn, nên giá trị p sẽ thấp hơn—ở trường hợp này chỉ bằng khoảng một nửa so với trước.

Diễn giải kết quả

Lúc bắt đầu chương này, tôi đã nói rằng điều mà ta cần giải đáp là liệu một hiệu ứng biểu kiến có phải là thật không. Chúng ta đã bắt đầu bằng việc định nghĩa giả thiết không, kí hiệu là {0}, vốn là giả thiết rằng hiệu ứng không phải thật. Sau đó chúng ta đã định nghĩa giá trị p, vốn là Pr(*E*|{0}), trong đó *E* là một hiệu ứng lớn bằng hoặc lớn hơn hiệu ứng biểu kiến. Sau đó chúng ta tính các giá trị p rồi so sánh chúng với một ngưỡng, α.

Đây là một bước cần thiết, nhưng chưa giải đáp được câu hỏi ban đầu, đó là liệu rằng hiệu ứng có thật không. Có một số cách diễn giải kết quả của một phép kiểm định giả thiết:

Cổ điển:

Trong kiểm định giả thiết theo cách cổ điển, nếu một giá trị p nhỏ hơn α, thì bạn có thể nói rằng hiệu ứng là có ý nghĩa về mặt thống kê, nhưng không thể kết luận rằng hiệu ứng đó có thực. Công thức nói trên đã cẩn thận tránh được việc nhảy đến kết luận, nhưng nó không xác đáng chút nào.

Thực dụng:

Trên thực tế, mọi người không quá thực hiện theo quy củ như vậy. Trong phần lớn các tạp chí khoa học, những nhà nghiên cứu đều công bố các giá trị p mà chẳng có biện luận gì, và độc giả diễn giải chúng như những bằng chứng cho thấy các hiệu ứng biểu kiến là có thật. Giá trị p càng thấp, thì họ càng tin tưởng vào kết luận này.

Phương pháp Bayes:

Thứ mà chúng ta cần biết là Pr(H_A|E), trong đó H_A là giả thiết rằng hiệu ứng có thật. Theo định lý Bayes:$P(H_A|E)=P(E|H_A)\ P(H_A)\ /\ P(E)$

trong đó Pr(H_A) là xác suất tiên nghiệm của H_A trước khi ta thấy được hiệu ứng, Pr(E|H_A) là xác suất thấy được E, với giả thiết rằng hiệu ứng này có thật, còn Pr(E) là xác suất thấy được E dưới bất kì giả thiết nào. Vì hiệu ứng thì hoặc là có thật, hoặc là không nên

$$\Pr(E) = \Pr(E|H_A)\ \Pr(H_A) + \Pr(E|H_0)\ \Pr(H_0)$$

Lấy ví dụ, tôi sẽ đi tính Pr(H_A|E) đối với thời gian mang thai theo số liệu của NSFG. Chúng ta đã tính được Pr(E|{0}) = 0.166, vì vậy tất cả những gì ta cần làm là tính được Pr(E|H_A) rồi chọn một giá trị cho xác suất tiên nghiệm.

Để tính được, Pr(E|H_A), chúng ta giả sử rằng hiệu ứng là có thật, —nghĩa là khác biệt trong thời gian mang thai trung bình, δ, chính là giá trị mà ta quan sát được, 0,078. (Cách xác lập H_A như thế này đôi chút giả dối. Tôi sẽ giải thích và giải quyết vấn đề này trong mục tiếp theo.)

Bằng cách phát sinh ra 1000 cặp mẫu, mỗi cặp có chứa hai giá trị từ hai phân bố, tôi ước tính được Pr(E|H_A) = 0,494. Với xác suất tiên nghiệm Pr(H_A) = 0,5, thì xác suất hậu nghiệm của H_A là 0,748.

Vì vậy nếu xác suất tiên nghiệm của H_A là 50%, thì xác suất được cập nhật, tức là đã tính đến bằng chứng thu được từ bộ số liệu này, đạt gần 75%. Việc xác suất hậu nghiệm cao hơn tiên nghiệm là hợp lý, vì số liệu đã phần nào củng cố cho giả thiết. Nhưng dường như có sự bất ngờ vì chênh lệch này quá lớn, đặc biệt là khi ta đã tính ra được sự khác biệt về trị trung bình không có ý nghĩa thống kê.

Trên thực tế, phương pháp mà tôi đã dùng ở mục này không hoàn toàn đúng, và nó có khuynh hướng phóng đại tầm ảnh hưởng của bằng chứng. Ở mục kế tiếp, ta sẽ điều chỉnh lại xu hướng này.

Hãy dùng số liệu từ NSFG để tính xem xác suất hậu nghiệm để các dạng phân bố cân nặng trẻ sơ sinh đối với trẻ đầu lòng và trẻ sinh sau khác biệt là bao nhiêu?

Bạn có thể bắt đầu với mã lệnh mà tôi đã dùng để phát sinh kết quả trong mục này, vốn tải được về từ `http://thinkstats.com/hypothesis.py`.

Thẩm định chéo

Ở ví dụ trước, ta đã sử dụng bộ số liệu để lập nên giả thiết H_A, và rồi dùng chính bộ số liệu đó để kiểm định. Điều này là không hay và rất dễ dẫn đến những kết quả gây lạc hướng.

Vấn đề là ở chỗ ngay cả khi giả thiết không là đúng thì dường như vẫn có một khác biệt δ nào đó giữa hai nhóm, chỉ do tình cờ. Nếu ta dùng giá trị quan sát được, δ, để lập nên giả thiết thì Pr(H_A|E) sẽ có khả năng cao ngay cả khi H_A là sai.

Ta có thể giải quyết vấn đề này bằng cách **thẩm định chéo**, trong đó dùng một bộ số liệu để tính δ và một bộ số liệu *khác* để tính H_A. Bộ số liệu thứ nhất được gọi là **bộ huấn luyện**; còn cái thứ hai được gọi là **bộ kiểm tra**.

Trong một nghiên cứu kiểu như NSFG, vốn phân tích các nhóm khác nhau đối với từng chu kì thì ta có thể dùng một chu kì để huấn luyện và một chu kì khác để kiểm tra. Hoặc ta có thể phân chia số liệu thành hai bộ số liệu con (một cách ngẫu nhiên), rồi dùng một bộ để huấn luyện và một bộ để kiểm tra.

Tôi thực hiện cách làm thứ hai, phân chia số liệu của Chu kì 6 thành hai nửa gần bằng nhau. Tôi tiến hành chạy chương trình kiểm định vài lần với các nhóm khác nhau ngẫu nhiên. Giá trị trung bình của xác suất hậu nghiệm bằng Pr(H_A|E) = 0,621. Như được mong đợi, ảnh hưởng của bằng chứng đã nhỏ hơn, phần là vì kích cỡ mẫu kiểm định nhỏ hơn, và cuxgn vì chúng ta không còn dùng chung số liệu để huấn luyện và kiểm thử.

Thông báo kết quả tính xác suất Bayes

Ở mục trước ta đã chọn xác suất tiên nghiệm Pr(H_A) = 0,5. Nếu ta có một tập hợp cá giả thiết và không có lý do nào để coi rằng một giả thiết có khả năng xảy ra nhiều hơn giả thiết khác, thì cách thông thường là gán chúng các xác suất bằng nhau.

Có người phản đối xác suất Bayes vì họ dựa vào các xác suất tiên nghiệm, và người ta có thể không thống nhất cùng một xác suất tiên nghiệm đúng. Với những người luôn mong đợi kết quả khoa học phải khách quan và phổ quát thì đặc tính nói trên thật không ổn thỏa.

Câu trả lời đối với ý kiến phản bác trên là, trong thực tế, các bằng chứng vững vàng thường có xu thế áp đảo ảnh hưởng của điều kiện tiên nghiệm, vì vậy dù ta có xuất phát từ những điều kiện tiên nghiệm khác nhau thì cuối cùng vẫn sẽ hội tụ về cùng một xác suất hậu nghiệm.

Một lựa chọn khác là chỉ thông báo mỗi **tỉ số khả năng**, Pr(E | H_A) / Pr(E|H_0), thay vì xác suất hậu nghiệm. Bằng cách này bạn đọc có thể đưa vào bất kì điều kiện tiên nghiệm nào tùy ý và tự tính ra xác suất hậu nghiệm (nói nghiêm túc). Tỉ số khả năng đối khi còn được gọi là hệ số Bayes (xem http://wikipedia.org/wiki/Bayes_factor).

> Nếu xác suất tiên nghiệm trong giải thiết H_A của bạn bằng 0,3 và một bằng chứng mới xuất hiện cho ra tỉ số khả năng bằng 3 so với giả thiết không, H_0, thì xác suất hậu nghiệm của H_A sẽ bằng bao nhiêu?

Bài tập này được điều chỉnh từ nguồn MacKay, *Information Theory, Inference, and Learning Algorithms*:

> Hai người vừa để lại vết máu ở hiện trường một vụ phạm tội. Nghi phạm Oliver được lấy mẫu máu và xác định là thuộc nhóm máu O. Các nhóm máu của hai vết được phát hiện lần lượt thuộc nhóm O (một nhóm máu thông dụng trong tổng thể dân cư của địa phương, với tần số 60%) và nhóm AB (nhóm máu hiếm với tần số chỉ 1%). Liệu số liệu này (kiểu nhóm máu tìm được ở hiện trường) có là bằng chứng ủng hộ việc quy kết Oliver là một trong hai người để lại vết máu ở hiện trường hay không?

Gợi ý: Hãy tính tỉ lệ khả năng cho bằng chứng này; nếu nó lớn hơn 1, thì bằng chứng có tính ủng hộ cho việc quy kết. Bạn có thể xem những phân tích và lời giải ở trang 55 trong cuốn sách của MacKay.

Kiểm định khi-bình phương

Ở Mục threshold chúng ta đã kết luận rằng khác biệt biểu kiến về thời gian mang thai trung bình với trẻ đầu lòng và trẻ sinh sau là không có ý nghĩa. Nhưng đến Mục relative.risk, khi tính rủi ro tương đối, ta đã thấy rằng các trẻ đầu lòng có xu hướng được chào đời sớm hơn, ít khi sinh đúng lịch, và cũng có xu hướng chào đời muộn so với trẻ sinh sau.

Vì vậy có lẽ các phân bố sẽ có cùng trị trung bình nhưng với phương sai khác nhau. Ta có thể kiểm tra mức ý nghĩa của khác biệt về phương sai, tuy nhiên phương sai thì không vững bằng trị trung bình, và các kiểm định giả thiết về phương sai thường có động thái không tốt.

Một cách làm khác là kiểm định một giả thiết phản ánh trực tiếp hơn về hiệu ứng biểu hiện; đó là giả thiết rằng trẻ đầu lòng có nhiều khả năng chào đời xớm hơn, ít chào đời đúng lịch, và cũng nhiều khả năng chào đời muộn hơn.

Chúng ta tiến hành theo năm bước đơn giản sau:

1. Chúng ta định nghĩa một tập hợp các hạng mục, gọi là **ô**, để phân loại cho từng đứa trẻ. Ở ví dụ này, có 6 ô vì trẻ có hai nhóm (đầu lòng và sinh sau) cùng ba ngăn (thời gian sinh sớm, đúng lịch, và sinh muộn).Tôi sẽ dùng các định nghĩa ở Mục rủi ro tương đối: một đứa trẻ được coi là sinh sớm nếu chào đời vào Tuần 37 hoặc trước dó, sinh đúng lịch nếu chào đời vào một trong các Tuần 38, 39 hoặc 40, và muộn nếu chào đời vào Tuần 41 hoặc chậm hơn.
2. Chúng ta đi tính số trẻ được trông đợi trong từng ô. Theo giả thiết không, ta giả sử rằng hai nhóm trẻ em này thuộc hai nhóm có cùng dạng phân bố, vì vậy ta có thể tính xác suất chung: Pr(sớm), Pr(đúng lịch) and Pr(muộn).Với trẻ đầu lòng, ta có n = 4413 mẫu, vì vậy theo giả thiết không chúng ta trông đợi rằng có n Pr(sớm) đứa trẻ đầu lòn sẽ chào đời sớm, n Pr(đúng lịch) chào đời đúng lịch, v.v. Tương tự, ta có m = 4735 đứa trẻ sinh sau, vì vậy sẽ trông đợi có m Pr(sớm) trẻ sinh sau sẽ chào đời sớm, v.v.
3. Với từng ô, ta đi tính độ lệch, tức là hiệu số giữa giá trị quan sát được, O_i, với giá trị được trông đợi (kì vọng), E_i.

4. Chúng ta đi tính một độ đo nào đó cho tổng độ lệch; đại lượng này được gọi là **đặc trưng thống kê kiểm định**. Lựa chọn thông dụng nhất là đặc trưng thống kê khi-bình phương:$\chi^2 = \sum_i (O_i - E_i)^2 / E_i$
5. Có thể sử dụng mô phỏng Monte Carlo để tính giá trị p, vốn là xác suất thấy được đặc trưng thống kê khi-bình phương cao bằng giá trị quan sát được theo giả thiết không.

Khi dùng đến đặc trưng thống kê khi-bình phương, quá trình này được gọi là **kiểm định khi-bình phương**. Một đặc điểm của kiểm định khi-bình phương là phân bố của đặc trưng được kiểm định có thể tính được theo công thức chính xác.

Dựa vào số liệu từ NSFG tôi tính được $\chi^2 = 91{,}64$; vốn sẽ xảy ra ngẫu nhiên chừng một lần trong số 1 vạn lần. Tôi kết luận rằng kết quả này có ý nghĩa về mặt thống kê, chỉ với một lưu ý: một lần nữa ta đã dùng cùng bộ dữ liệu cho việc khám phá và kiểm thử. Tốt hơn hết là kiểm tra lại kết quả này với một bộ số liệu khác.

Bạn có thể tải về mã lệnh mà tôi dùng cho mục này từ `http://thinkstats.com/chi.py`.

> Giả dụ rằng bạn điều hành một sòng bạc và nghi ngờ rằng một khách chơi đã đánh tráo một quân xúc sắc sẵn có của sòng bạc bằng một quân xúc sắc "lệch"; nghĩa là nó có xu hướng gieo được một mặt nhiều hơn hẳn những mặt khác. Bạn đã bắt giữ người khách gian lận và tịch thu quân xúc sắc, nhưng giờ đây bạn phải chứng tỏ rằng quân xúc sắc này bị lệch.
>
> Bạn gieo xúc sắc 60 lần và nhận được kết quả sau:
>
> ```
> Giá trị 1 2 3  4 5 6
> Tần số  8 9 19 6 8 10
> ```
>
> Đặc trưng thống kê khi-bình phương cho các giá trị này bằng bao nhiêu? Xác suất để thấy được một giá trị khi-bình phương lớn như vậy một cách ngẫu nhiên là bao nhiêu?

Lấy mẫu lại một cách hiệu quả

Bất cứ ai trước khi đọc sách này đã được học xác suất có thể bật cười khi thấy Hình [length_deltas_cdf], bởi vì tôi đã phải chạy máy tính rất nhiều mới mô phỏng được điều mà lẽ ra đã hình dung được bằng cách giải tích.

Rõ ràng là phân tích toán học không phải là trọng tâm của cuốn sách này. Tôi sẵn lòng dùng máy tính để làm theo cách "đần độn" này, vì tôi nghĩ rằng người mới học thì sẽ dễ hiểu các kết quả mô phỏng bởi máy tính hơn, và dễ thấy hơn rằng chúng đúng đắn. Miễn là chương trình mô phỏng không chạy quá lâu, thì tôi cũng không bận tâm gì việc bỏ qua bước phân tích theo công thức.

Dù vậy, vẫn có những lúc mà việc phân tích một chút có thể tiết kiệm được những công sức tính toán, và Hình length_deltas_cdf là một trong những trường hợp đó.

Hãy nhớ rằng chúng ta đang kiểm định hiệu số quan sát được giữa thơi gian mang thai với $n = 4413$ trẻ đầu lòng và $m = 4735$ trẻ sinh sau. Ta đã thiết lập được phân bố tổng hợp cho tất cả những đứa trẻ, lấy ra các mẫu với các kích thước n và m, rồi đi tính hiệu số giữa các trị trung bình mẫu.

Tay vào đó, ta có thể tính trực tiếp phân bố của hiệu số giữa các trị trung bình mẫu. Để bắt đầu, ta hãy hình dung trị trung bình của một mẫu là gì: ta rút ra n mẫu từ một phân bố, cộng chúng lại, rồi chia cho n. Nếu phân bố có trị trung bình μ và phương sai σ^2, thì theo Định lý giới hạn trung tâm, ta biết rằng tổng của các mẫu tuân theo N ($n\ \mu$, $n\ \sigma^2$).

Để hình dung ra phân bố của các trị trung bình mẫu, ta phải dùng đến một trong số các thuộc tính của phân bố chuẩn: nếu X tuân theo N (μ, σ^2),

$aX + b \sim \mathrm{N}(a\ \mu + b, a^2\sigma^2)$

Khi chia cho n, $a = 1/n$ và $b = 0$, vì vậy

$X/n \sim \mathrm{N}(\mu\ /n, \sigma^2/\ n^2)$

Do đó phân bố của trị trung bình mẫu là N (μ, σ^2/n).

Để thu được phân bố của hiệu số giữa hai trị trung bình mẫu, ta dùng đến một thuộc tính khác của phân bố chuẩn: nếu X_1 tuân theo N (μ_1, ${\sigma_1}^2$) và X_2 tuân theo N (μ_2, ${\sigma_2}^2$),

$aX_1 + bX_2 \sim N(a\mu_1 + b\mu_2, a^2{\sigma_1}^2 + b^2{\sigma_2}^2)$

Vì vậy, như một trường hợp đặc biệt:

$X_1 - X_2 \sim N(\mu_1 - \mu_2\ , {\sigma_1}^2 + {\sigma_2}^2)$

Kết hợp lại, ta rút ra kết luận rằng mẫu trong Hình length_deltas_cdf được rút ra từ N (0, $f\ \sigma^2$), trong đó $f = 1/n + 1/m$. Thay $n = 4413$ và $m = 4735$ vào, ta trông đợi hiệu số các trị trung bình mẫu thuân theo N (0; 0,0032).

Chúng ta có thể dùng `erf.NormalCdf` để tính được giá trị p của hiệu số các trị trung bình quan sát được:

```
delta = 0.078
sigma = math.sqrt(0.0032)
left = erf.NormalCdf(-delta, 0.0, sigma)
right = 1 - erf.NormalCdf(delta, 0.0, sigma)
```

Tổng của các đuôi trái và phải cho ta giá trị p, bằng 0,168; vốn khá sát với giá trị mà ta ước tính từ việc tái lập mẫu; 0,166. Bạn có thể tải về đoạn mã lệnh mà tôi dùng trong mục này từ http://thinkstats.com/hypothesis_analytic.py

Độ mạnh

Khi kết quả của một kiểm định thống kê là âm tính (nghĩa là hiệu ứng không có ý nghĩa thống kê) thì liệu ta có kết luận rằng hiệu ứng đó không phải là thật không? Điều này tùy thuộc vào độ mạnh của phép kiểm định.

Độ mạnh thống kê là xác suất mà phép kiểm định sẽ dương tính khi giải thiết không bị sai. Nói chung, độ mạnh của một phép kiểm định thì phụ thuộc vào kích thước mẫu, độ lớn của hiệu ứng, và ngưỡng α.

Độ mạnh của phép kiểm định ở Mục threshold bằng bao nhiêu, với α = 0,05 và giả thiết rằng hiệu số đúng giữa hai trị trung bình bằng 0,078 tuần?

Bạn có thể ước tính độ mạnh bằng cách phát sinh ra những mẫu ngẫu nhiên từ các phân bố với hiệu số trị trung bình co trước, kiểm định hiệu số trị trung bình quan sát được, và đếm số lần dương tính.

Độ mạnh của phép kiểm định với α = 0,10 là bao nhiêu?

Một cách báo cáo độ mạnh của một phép kiểm định, cùng với một kết quả âm tính, là phát biểu kiểu như, “Nếu hiệu ứng biểu kiến mà lớn bằng *X*, thì phép kiểm định này sẽ bác bỏ giả thiết không với xác suất *p*.”

Thuật ngữ

ý nghĩa:
: Một hiệu ứng được gọi là có ý nghĩa thống kê nếu nó dường như không xảy ra một cách tình cờ.

giả thiết không:
: Mô hình của một hệ thống dựa trên giả thiết rằng hiệu ứng biểu kiến chỉ là tình cờ.

giá trị p:
: Xác suất để một hiệu ứng xảy ra tình cờ.

kiểm định giá thiết:
: Quá trình xác định xem liệu một hiệu ứng biểu kiến có ý nghĩa về mặt thống kê không.

dương tính sai:
: Kết luận rằng một hiệu ứng là có thật trong khi không phải như vậy.

âm tính sai:
: Kết luận rằng một hiệu ứng là tình cờ trong khi không phải như vậy.

phép thử hai phía:
: Phép thử trong đó đặt câu hỏi, “Khả năng của một hiệu ứng lớn bằng hiệu ứng quan sát được, bất kể âm hoặc dương, là bao nhiêu?”

phép thử một phía:
: Phép thử trong đó đặt câu hỏi, “Khả năng của một hiệu ứng lớn bằng hiệu ứng quan sát được, với cùng dấu, là bao nhiêu?”

thẩm định chéo:
: Quá trình kiểm định giả thiết trong đó dùng một bộ số liệu để phân tích số liệu khám phá và một bộ kia để dùng cho việc kiểm định.

bộ huấn luyện:
: Bộ số liệu dùng để xác lập một giả thiết cho việc kiểm thử.

bộ kiểm tra:
: Bộ số liệu dùng để kiểm thử.

đặc trưng thống kê kiểm định:
: Một đặc trưng thống kê được dùng để đo mức độ khác biệt giữa một hiệu ứng biểu kiến so với điều được trông đợi xảy ra tình cờ.

kiểm định khi-bình phương:
: Phép kiểm định có dùng đặc trưng khi-bình phương làm đặc trưng thống kê kiểm định.

tỉ số khả năng:
: Tỉ số giữa Pr(*E*|*A*) và Pr(*E*|*B*) của hai giả thiết *A* và *B*, là một cách để báo cáo kết quả của phân tích Bayes mà không phụ thuộc vào xác suất tiên nghiệm.

ô:
: Trong kiểm định khi-bình phương, những hạng mục để phân chia các kết quả quan sát vào.

độ mạnh:

Xác suất mà một phép thử sẽ bác bỏ giả thiết không nếu giả thiết này là sai.

---

1. Còn được biết đến với tên “Mức [tiêu chuẩn] ý nghĩa.” ↩

Chương 8: Ước lượng

Trở về Muc luc cuốn sách

Trò chơi ước lượng

Ta hãy cùng tham gia một trò chơi. Tôi sẽ nghĩ trong đầu một dạng phân bố, còn bạn phải đoán xem đó là phân bố gì. Chúng ta sẽ bắt đầu một cách dễ dàng rồi sau đó sẽ khó dần lên.

Tôi đang nghĩ đến một dạng phân bố. Sau đây là hai lời gợi ý; đó là một phân bố chuẩn, và sau đây là một mẫu ngẫu nhiên được rút từ nó:

{–0,441 ; 1,774 ; –0,101 ; –1,138 ; 2,975 ; –2,138}

Bạn thử đoán xem tham số trị trung bình, μ, của phân bố này bằng bao nhiêu?

Một cách lựa chọn là dùng trị trung bình mẫu để ước tính μ. Cho đến giờ ta dùng kí hiệu μ chung cho cả trị trung bình mẫu và tham số trung bình, nhưng bây giờ để phân biệt chúng, tôi sẽ dùng  cho trị trung bình mẫu. Ở ví dụ này,  bằng 0,155; vì vậy sẽ có lý khi dự đoán μ = 0,155.

Quá trình này được gọi là **ước lượng**, và đặc trưng thống kê mà ta xét đến (trị trung bình mẫu) được gọi là **tham số ước lượng**.

Việc dùng trị trung bình mẫu để ước lượng μ là hiển nhiên đến nỗi thật khó tưởng tượng nổi một cách làm khác. Nhưng giả dụ rằng chúng ta thay đổi trò chơi bằng cách đưa vào các điểm biệt lập.

Tôi đang nghĩ về một phân bố. Đó là một phân bố chuẩn, và sau đây là một mẫu thu thập được bởi người điều tra viên không đáng tin cậy, đôi khi lại mắc lỗi đặt nhầm dấu phẩy của phần thập phân.

{–0.441, 1.774, –0.101, –1.138, 2.975, –213.8}

Bây giờ ước đoán của bạn về μ bằng bao nhiêu? Nếu bạn dùng trị trung bình mẫu thì ước đoán sẽ bằng –35,12. Liệu đó có phải là lựa chọn tốt nhất không? Còn có những cách nào khác?

Có một cách khác là nhận biết và loại bỏ các điểm biệt lập, rồi tính trị trung bình mẫu của những số còn lại. Một cách nữa là dùng số trung vị để ước lượng.

Ước lượng nào là tốt nhất còn phụ thuộc vào từng trường hợp cụ thể (chẳng hạn, liệu có điểm biệt lập hay không) và vào mục tiêu của ước lượng. Bạn nhằm mục đích giảm thiểu sai số, hay nâng cao tối đa khả năng tìm được đáp số đúng?

Nếu như không có điểm biệt lập nào thì trị trung bình mẫu sẽ làm giảm thiểu **sai số quân phương** (mean squared error, MSE). Nếu bạn chơi trò đoán số này nhiều lần, và mỗi lần tính sai số  – μ, thì trị trung bình mẫu sẽ làm giảm thiểu đại lượng

$MSE = (1/m) \sum(\square - \mu)^2$

trong đó *m* là số lần đoán (khác với *n*, là kích thước mẫu dùng để tính).

Giảm thiểu MSE là một điều tốt, nhưng nó không phải luôn là chiến lược tối ưu. Chẳng hạn, giả sử ta cần ước tính dạng phân bố tốc độ gió tại một công trường. Nếu ta đoán gió quá mạnh thì có thể sẽ phải xây dựng với chi phí quá đắt. Nhưng nếu ta đoán gió quá yếu thì công trình có thể sẽ sụp đổ. Bởi vì chi phí xây dựng phụ thuộc vào sai số theo một dạng bất đối xứng nên việc giảm thiểu MSE không phải là cách tốt nhất.

Lấy một ví dụ khác, chẳng hạn khi tôi gieo xúc sắc 3 lần và đề nghị bạn ước tính tổng số chấm. Nếu đoán đúng, bạn sẽ nhận một phần thưởng; còn nếu không thì sẽ chẳng được gì. Trong trường hợp này giá trị làm giảm thiểu MSE là 10,5; nhưng nó sẽ là một dự đoán hoàn toàn sai. Trong trò chơi này, bạn muốn có một ước đoán có khả năng đúng nhiều nhất, tức là một **ước đoán khả năng cao nhất** (Maximum Likelihood Estimator, MLE). Nếu bạn chọn 10 hoặc 11, bạn sẽ có khả năng đoán trúng bằng 1 phần 8, và đó là khả năng tốt nhất bạn có thể đạt được.

> Hãy viết một hàm để rút ngẫu nhiên 6 giá trị từ một phân bố chuẩn với μ = 0 và σ = 1.
>
> Hãy dùng trị trung bình mẫu để ước tính μ rồi tính sai số – μ. Gọi hàm này 1000 lần và tính MSE.
>
> Bây giờ hãy sửa lại chương trình để lấy số trung vị làm giá trị ước đoán. Tính lại MSE rồi so sánh nó với MSE trong trường hợp dùng .

Ước lượng phương sai

Tôi đang nghĩ về một phân bố. Đó là phân bố chuẩn, và sau đây là một mẫu (quen thuộc):

{–0,441 ; 1,774 ; –0,101 ; –1,138 ; 2,975 ; –2,138}

Theo bạn thì phương sai, σ^2, của phân bố này bằng bao nhiêu? Một lần nữa, lựa chọn hiển nhiên nhất là lấy phương sai mẫu làm giá trị ước lượng. Tôi sẽ kí hiệu phương sai mẫu là S^2, để phân biệt với tham số chưa biết là σ^2.

$$S^2 = \frac{1}{n}\sum(x_i - \bar{x})^2$$

Với những mẫu lớn, S^2 là một ước lượng thích hợp, nhưng với mẫu bé thì giá trị nó thiên nhỏ. Vì đặc điểm không may này mà nó được gọi là một ước lượng **chệch**.

Một ước lượng được gọi là **không chệch** nếu như sai số tổng cộng (hoặc trung bình) được trông đợi sau nhiều lượt chơi bằng 0. Thật may là còn có một đại lượng thống kê đơn giản khác làm ước lượng không chệch cho σ^2:

$$S_{n-1}^2 = \frac{1}{n-1}\sum(x_i - \bar{x})^2$$

Vấn đề nan giải nhất đối với ước lượng này là cách dùng tên gọi và kí hiệu của nó không thống nhất. Cái tên “phương sai mẫu” có thể được dùng cho cả S^2 và ${S_{n-1}}^2$, còn kí hiệu S^2 được dùng để chỉ cả hai đại lượng này.

Để hiểu được tại sao S^2 là ước lượng chệch, và chứng minh rằng S_{n-1}^2 không chệch, bạn có thể tham khảo http://wikipedia.org/wiki/Bias_of_an_estimator.

> Hãy viết một hàm để rút ngẫu nhiên 6 giá trị từ một phân bố chuẩn với μ = 0 và σ = 1. Hãy dùng phương sai mẫu để ước tính σ^2 rồi tính sai số $S^2 - \sigma^2$. Gọi hàm này 1000 lần và tính sai số trung bình (không bình phương).
>
> Bây giờ hãy sửa lại chương trình để dùng ước lượng không chệch S_{n-1}^2. Tính lại sai số trung bình rồi xem xét liệu nó có hội tụ về 0 khi bạn tăng số lượt chơi hay không.

Hiểu được các sai số

Trước khi tiếp tục, ta hay gỡ một chỗ rắc rỗi dễ gây hiểu nhầm. Các thuộc tính như MSE và độ lệch là những giá trị được trông đợi lâu dựa trên nhiều lượt chơi trò ước đoán.

Trong bạn đang chơi, bạn không biết sai số bằng bao nhiêu. Nghĩa là, nếu tôi giao cho bạn một mẫu và yêu cầu ước tính tham số, bạn có thể tính giá trị của một ước lượng, nhưng không thể tính giá trị sai số. Nếu tính được sai số, bạn đã chẳng phải ước lượng làm gì nữa!

Nguyên do mà ta bàn về việc ước tính sai số là để miêu tả động thái của các ước lượng khác nhau về lâu dài. Trong chương này ta sẽ chạy các thử nghiệm mô phỏng để kiểm tra những động thái như vậy; các thí nghiệm này là "nhân tạo" vì ta đã biết giá trị thật của các tham số, vì vậy mà tính được ra sai số. Nhưng khi bạn làm việc với số liệu trên thực tế thì bạn không biết giá trị tham số nên không thể tính được sai số.

Bây giờ ta hãy tiếp tục trò chơi.

Phân bố lũy thừa

Tôi đang nghĩ về một dạng phân bố. Đó là phân bố lũy thừa, và sau đây là một mẫu:

{5,384 ; 4,493 ; 19,198 ; 2,790 ; 6,122 ; 12,844}

Bạn thử đoán xem tham số, λ, của phân bố này bằng bao nhiêu?

Nói chung, trị trung bình của một phân bố lũy thừa bằng 1/λ, vì vậy tính ngược lại, bạn có thể lấy

$\square = 1/\square$

Thông thường ta dùng dấu mũ để kí hiệu cho các ước lượng; ở đây $\square$ là ước lượng của λ. Và không chỉ với một ước lượng bất kì, nó cũng là ước lượng khả năng cao nhất[1]. Vì vậy nếu bạn muốn có cơ hội nhiều nhất để đoán trúng λ, thì $\square$ chính là cách làm.

Nhưng ta biết rằng $\square$ không còn vững nếu xuất hiện các điểm biệt lập, vì vậy ta đoán rằng $\square$ cũng gặp vấn đề tương tự.

Có thể ta sẽ tìm được một cách tính khác dựa trên số trung vị mẫu. Nhớ lại rằng số trung vị của một phân bố lũy thừa bằng log(2) / λ, vì vậy lại tính ngược, ta có định nghĩa cho ước lượng sau

$$\hat{\lambda}_{1/2} = \log(2)\mu_{1/2}$$

trong đó $\mu_{1/2}$ là số trung vị mẫu.

> Hãy chạy một thử nghiệm để xem giữa [ ] và $\hat{\lambda}_{1/2}$, cái nào cho MSE thấp hơn. Kiểm tra xem cái nào là ước lượng chệch, nếu có.

Khoảng tin cậy

Đến giờ ta mới thấy những ước lượng cho ra giá trị đơn lẻ; chúng được gọi là **ước lượng điểm**. Với nhiều bài toán, ta có thể cần một khoảng giá trị, có giới hạn trên và dưới, cho một tham số chưa biết.

Hoặc, tổng quát hơn, ta có thể muốn cả một dạng phân bố; nghĩa là khoảng các giá trị mà tham số có thể nhận, và với mỗi giá trị trong khoảng, có một con số chỉ khả năng xuất hiện là bao nhiêu.

Đầu tiên, ta hãy xét đến **khoảng tin cậy**.

Tôi đang nghĩ về một dạng phân bố. Đó là phân bố lũy thừa, và sau đây là một mẫu:

{5,384 ; 4,493 ; 19,198 ; 2,790 ; 6,122 ; 12,844}

Tôi muốn bạn đưa ra một khoảng giá trị mà bạn nghĩ rằng có nhiều khả năng là tham số chưa biết, λ, sẽ rơi vào. Cụ thể hơn, tôi muốn một khoảng tin cậy 90%, nghĩa là nếu tôi cứ chơi đi chơi lại trò này thì λ sẽ có 90% số lần rơi vào trong khoảng đó.

Hóa ra kiểu trò chơi này cũng khó, cho nên tôi sẽ nói cho bạn biết lời giải, và việc bạn cần làm chỉ là kiểm tra lại nó.

Khoảng tin cậy thường được mô tả dưới dạng tỉ lệ trượt, α, vì vậy khoảng tin cậy 90% có một tỉ lệ trượt là α = 0,1. Khoảng tin cậy của tham số λ đối với phân bố lũy thừa thì bằng

[ ]

trong đó *n* là kích thước mẫu, [ ] là ước lượng dựa trên trị trung bình, như ở mục trước, còn $\chi^2(k,x)$ là CDF của một phân bố khi-bình phương với *k* bậc tự do, được tính tại *x* (xem http://wikipedia.org/wiki/Chi-square_distribution).

Nói chung, các khoảng tin cậy rất khó tính được theo cách giải tích, nhưng khá dễ tính được bằng cách mô phỏng. Song trước hết ta cần nói về ước lượng Bayes.

Ước lượng Bayes

Nếu bạn thu thập một mẫu và tính khoảng tin cậy 90%, thì bạn có thể sẽ muốn phát biểu rằng giá trị thật của tham số có 90% cơ hội rơi vào trong khảng này. Nhưng từ quan điểm tần suất, điều đó là

không đúng vì tham số tuy chưa biết nhưng lại là một giá trị cố định. Nó chỉ có thể hoặc là rơi vào khoảng tính được, hoặc là không, vì vậy cách định nghĩa theo tần suất sẽ không áp dụng được.

Vì vậy ta hãy thử một phiên bản khác của trò chơi.

Tôi đang nghĩ về một dạng phân bố. Đó là phân bố lũy thừa, và tôi chọn λ từ một phân bố đều, nằm giữa 0,5 và 1,5. Sau đây là một mẫu, mà tôi sẽ gọi là *x*:

{2,675 ; 0,198 ; 1,152 ; 0,787 ; 2,717 ; 4,269}

Căn cứ vào mẫu này, theo bạn thì tôi đã chọn giá trị λ bằng bao nhiêu?

Trong phiên bản này của trò chơi, λ *là* một đại lượng ngẫu nhiên, vì vậy việc ta nói về phân bố của nó là hợp lý, và ta có thể dễ dàng tính được nó bằng định lý Bayes.

Sau đây là các bước thực hiện:

1. Chia khoảng (0,5 ; 1,5) thành một tập hợp các ngăn có kích thước bằng nhau. Với mỗi ngăn, chúng ta định nghĩa H_i, vốn là giả thiết rằng giá trị thật của λ rơi vào ngăn thứ *i*. Vì λ được rút ra từ một phân bố đều, nên xác suất tiên nghiệm, $P(H_i)$, là như nhau với tất cả mọi *i*.
2. Với mỗi giả thiết, ta đi tính khả năng, $P(x|H_i)$, vốn là cơ hội rút được mẫu *x* khi cho trước H_i.$P(X|H_i) = \prod_j \text{expo}(\lambda_i, x_j)$trong đó expo(λ, *x*) là một hàm để tính PDF của phân bố lũy thừa với tham số λ, tại điểm giá trị *x*.

   $PDF_{expo}(\lambda,x) = \lambda e^{-\lambda x}$

   Kí hiệu ∏ biểu thị cho tích của một dãy số (xem http://wikipedia.org/wiki/Multiplication#Capital_Pi_notation).
3. Từ đó bằng định lý Bayes, phân bố hậu nghiệm là$P(H_i|x) = P(H_i)\ P(x|H_i) / f$trong đó *f* là thừa số chuẩn hóa

   $f = \sum_i P(H_i)\ P(X|H_i)$

Cho trước một phân bố hậu nghiệm, ta dễ dàng tính được khoảng tin cậy. Chẳng hạn, để tinh được khoảng tin cậy 90%, bạn có thể dùng các số phần trăm thứ 5 và 95 của phân bố hậu nghiệm.

Tiến hành ước lượng Bayes

Để biểu thị phân bố tiên nghiệm, ta có thể dùng Pmf, Cdf, hoặc một cách biểu diễn bất kì nào khác cho một phân bố, nhưng vì ta muốn có sự tương ứng giữa mỗi giả thiết với một xác suất, nên Pmf là lựa chọn tự nhiên.

Mỗi giá trị trong Pmf biểu diễn cho một giả thiết; chẳng hạn, giá trị 0,5 biểu diễn cho giả thiết rằng λ bằng 0,5. Trong phân bố xác suất tiên nghiệm, tất cả những giả thiết đều có cùng xác suất. Vì vậy ta có thể thiết lập điều kiện tiên nghiệm như sau:

```
def MakeUniformSuite(low, high, steps):
    hypos = [low + (high-low) * i / (steps-1.0) for i in range(steps)]
    pmf = Pmf.MakePmfFromList(hypos)
    return pmf
```

Hàm này tạo ra và trả lại một Pmf biểu thị cho một tập hợp các giả thiết có liên quan, được gọi là một **bộ giả thiết**. Mỗi giả thiết có cùng xác suất, vì vậy phân bố có dạng **đều**.

Các đối số `low` và `high` để chỉ định khoảng các giá trị; `steps` là số các giả thiết.

Để thực hiện việc cập nhật, ta lấy một bộ các giả thiết và một bằng chứng:

```
def Update(suite, evidence):
    for hypo in suite.Values():
        likelihood = Likelihood(evidence, hypo)
        suite.Mult(hypo, likelihood)
    suite.Normalize()
```

Với mỗi giả thiết trong bộ, ta đem nhân xác suất tiên nghiệm với khả năng xuất hiện bằng chứng. Sau đó ta chuẩn hóa bộ này.

Trong hàm này, `suite` phải là một Pmf, nhưng `evidence` có thể là một kiểu bất kì, miễn sao `Likelihood` có thể nhận biết được kiểu đó.

Sau đây là hàm tính khả năng:

```
def Likelihood(evidence, hypo):
    param = hypo
    likelihood = 1
    for x in evidence:
        likelihood *= ExpoPdf(x, param)

    return likelihood
```

Trong `Likelihood`, chúng ta giả thiết rằng `evidence` là một mẫu từ phân bố lũy thừa rồi tính tích số đã đề cập ở mục trước.

`ExpoPdf` có nhiệm vụ tính PDF của phân bố lũy thừa tại `x`:

```
def ExpoPdf(x, param):
    p = param * math.exp(-param * x)
    return p
```

Đem gộp tất cả lại, ta được đoạn mã sau đây để tạo ra điều kiện tiên nghiệm và tính xác suất hậu nghiệm:

```
evidence = [2.675, 0.198, 1.152, 0.787, 2.717, 4.269]
prior = MakeUniformSuite(0.5, 1.5, 100)
posterior = prior.Copy()
Update(posterior, evidence)
```

Bạn có thể tải về đoạn mã trong mục này từ http://thinkstats.com/estimate.py.

Khi tôi nghĩ về ước lượng Bayes, tôi hình dung đến một phòng khách chật kín người, trong đó mỗi người có một phỏng đoán khác nhau về con số mà bạn hiện đang ước lượng. Như vậy, ở ví dụ này mỗi người khách sẽ dự đoán về giá trị đúng của λ.

Ban đầu, mỗi người đều có một độ đo sự tự tin riêng về giả thiết của mình. Sau khi biết được bằng chứng, từng người sẽ cập nhật độ tự tin này dựa trên P(E|H), hay khả năng của bằng chứng khi cho trước giả thiết của họ.

Thường thì hàm khả năng có nhiệm vụ tính một xác suất, vốn lớn nhất là bằng 1, vì vậy bạn đầu niềm tin của mọi người giảm xuống (hoặc không đổi). Nhưng sau đó ta thực hiện chuẩn hóa, và điều này làm tăng niềm tin của mọi người lên.

Vì vậy hiệu ứng tổng cộng là có người vững tin hơn, và có người giảm độ tin tưởng, tùy theo khả năng tương đối của giải thuyết mà họ xác lập.

Số liệu bị kiểm duyệt

Bài toán sau đây xuất hiện trong Chương 3 của cuốn *Information Theory, Inference and Learning Algorithms* (David McKay), mà bạn có thể tải về từ `http://www.inference.phy.cam.ac.uk/mackay/itprnn/ps/`.

> Các hạt không bền được phát ra từ nguồn và phân rã ở một khoảng cách x, một số thực có phân bố xác suất lũy thừa với [tham số] λ. Sự phân rã chỉ có thể quan sát được nếu nó xảy ra trong một khung nhìn có phạm vi từ x = 1 cm đến x = 20 cm. Người ta đã quan sát được n sự phân rã tại các vị trí $\{x_1, \ldots, x_N\}$. Hỏi λ bằng bao nhiêu?

Đây là một ví dụ về bài toán ước lượng với **số liệu bị kiểm duyệt**; nghĩa là ta biết rằng có một phần dữ liệu bị loại trừ một cách có hệ thống.

Một trong những điểm mạnh của ước lượng Bayes là nó có thể khá dễ dàng xử lý các số liệu bị kiểm duyệt. Ta có thể dùng phương pháp đã biết ở mục trước, và chỉ cần thay đổi một chỗ: ta phải thay thế PDF_{expo} với xác suất có điều kiện:

$$PDF_{cond}(\lambda,x) = \lambda e^{-\lambda x} / Z(\lambda)$$

với $1 < x < 20$, và 0 với trường hợp còn lại, trong đó

$$Z(\lambda) = \int_1^{20} \lambda e^{-\lambda x} dx = e^{-\lambda} - e^{-20\lambda}$$

Bạn có thể nhớ lại Z(λ) từ Bài tập [expo_pdf]. Tôi có nhắc bạn rằng hãy ghi lại một bản giá trị để có lúc cần dùng đến.

> Hãy tải về `http://thinkstats.com/estimate.py`, trong đó có chứa mã lệnh từ mục trước, và lưu tại một bản sao có tên `decay.py`.
>
> Sửa lại `decay.py` để tính được phân bố hậu nghiệm của λ đối với mẫu x = {1.5, 2, 3, 4, 5, 12}. Còn về xác suất tiên nghiệm, bạn có thể dùng dạng phân bố đều giữa 0 và 1,5 (không kể đến 0).
>
> Bạn có thể tải về một lời giải cho bài toán này từ `http://thinkstats.com/decay.py`.
>
> Vào năm 2008 cuộc tranh cử Minnesota Senate kết thúc với số phiếu 1.212.629 ủng hộ cho Al Franken và 1.212.317 cho Norm Coleman. Franken đã được công nhận thắng cử, nhưng tác giả Charles Seife đã chỉ ra trong *Proofiness*, rằng ranh giới trong việc quyết định thắng cử quá nhỏ so với ranh giới sai số trong quá trình kiểm phiếu, vì vậy kết quả nên được coi như bất phân thắng bại.

> Giả sử rằng có khả năng là một lá phiếu bất kì bị thất lạc và khả năng cho lá phiếu bị đếm trùng 2 lần thì xác suất để Coleman mới thật là người chiếm đa số phiếu sẽ bằng bao nhiêu?
>
> Gợi ý: bạn phải điền thêm các chi tiết vào mới mô phỏng được quá trình sai số.

Bài toán tầu hỏa

Bài toán tầu hỏa là loại bài toán ước lượng kinh điển, mà cũng được biết đến với tên gọi "bài toán xe tăng Đức". Đây là đề bài xuất hiện trong cuốn sách của Mosteller, *Fifty Challenging Problems in Probability*:

> "Một tuyến đường sắt có các đầu tàu được đánh số từ 1..N. Một ngày kia bạn thấy đầu tàu có đánh số 60. Hãy ước tính xem có bao nhiêu tàu trên tuyến đường."

Trước khi đọc phần còn lại của mục này, bạn hãy thử trả lời các câu hỏi sau:

1. Với một ước lượng cho trước, $\hat{n}$, thì khả năng của chứng cứ, P($E|\hat{n}$), sẽ bằng bao nhiêu? Giá trị ước lượng khả năng cao nhất bằng bao nhiêu?
2. Nếu ta nhìn thấy con tàu đánh số *i* thì có vẻ hợp lý để dự đoán con số bằng bội số của *i* vì vậy hãy giả sử rằng $\hat{n} = a\,i$. Giá trị nào của *a* sẽ làm giảm thiểu sai số quân phương?
3. Vẫn với giả thiết rằng $\hat{n} = a\,i$, bạn có thể tìm được một giá trị của *a* để cho  là một ước lượng không chệch?
4. Với giá trị nào của N thì 60 sẽ là giá trị trung bình?
5. Phân bố hậu nghiệm tính theo phương pháp Bayes sẽ bằng bao nhiêu, nếu giả sử rằng phân bố tiên nghiệm là đều trong khoảng từ 1 đến 200?

Tốt nhất là bạn dành chút thời gian cố gắng giải đáp các câu hỏi trên trước khi đọc tiếp.

Với một phân bố cho trước, $\hat{n}$, khả năng thấy được tàu thứ *i* là $1/\hat{n}$ nếu $i \leq \hat{n}$, và bằng 0 trong trường hợp còn lại. Vì vậy ước lượng khả năng cao nhất (MLE) bằng $\hat{n} = i$. Nói cách khác, nếu bạn thấy tàu số 60 và muốn làm tối đa khả năng đoán trúng thì bạn nên đoán rằng có 60 con tàu.

Nhưng ước lượng này không được tốt lắm nếu xét về sai số quân phương (MSE). Ta có thể làm tốt hơn bằng cách chọn $\hat{n} = ai$; tất cả những gì cần làm là tìm một giá trị tốt cho *a*.

Giả sử rằng trên thực tế có *n* tàu. Mỗi lần chơi trò dự đoán này, ta nhìn thấy tàu số *i* và đoán *ai*, vì vậy sai số quân phương là $(ai - n)^2$.

Nếu ta chơi *n* lần và mỗi lần nhìn thấy một đoàn tàu thì sai số quân phương bằng

$$MSE = 1/N \sum_{i=1}^{N} (ai - N)^2$$

Để giảm thiểu MSE, ta cần lấy đạo hàm theo *a*:

$$dMSE/da = 1/N \sum_{i=1}^{N} 2i(ai - N) = 0$$

Và giải ra *a*.

$$a = 3N/(2N + 1)$$

Thoạt nhìn, điều này dường như không có ích gì, vì *n* xuất hiện ở vế phải, tức là ta cần phải biết *n* mới chọn được *a*, mà đã biết *n* thì ngay từ đầu ta đã chẳng ước lượng làm gì.

Tuy nhiên, với các giá trị *n* lớn thì giá trị tối ưu của *a* hội tụ về 3/2, vì vậy ta có thể chọn $\hat{N} = 3i/2$.

Để tìm một ước lượng không chệch, ta có thể tính ra sai số trung bình (ME, mean error):

$ME = 1/N \sum_{i=1}^{N}(ai - N)$

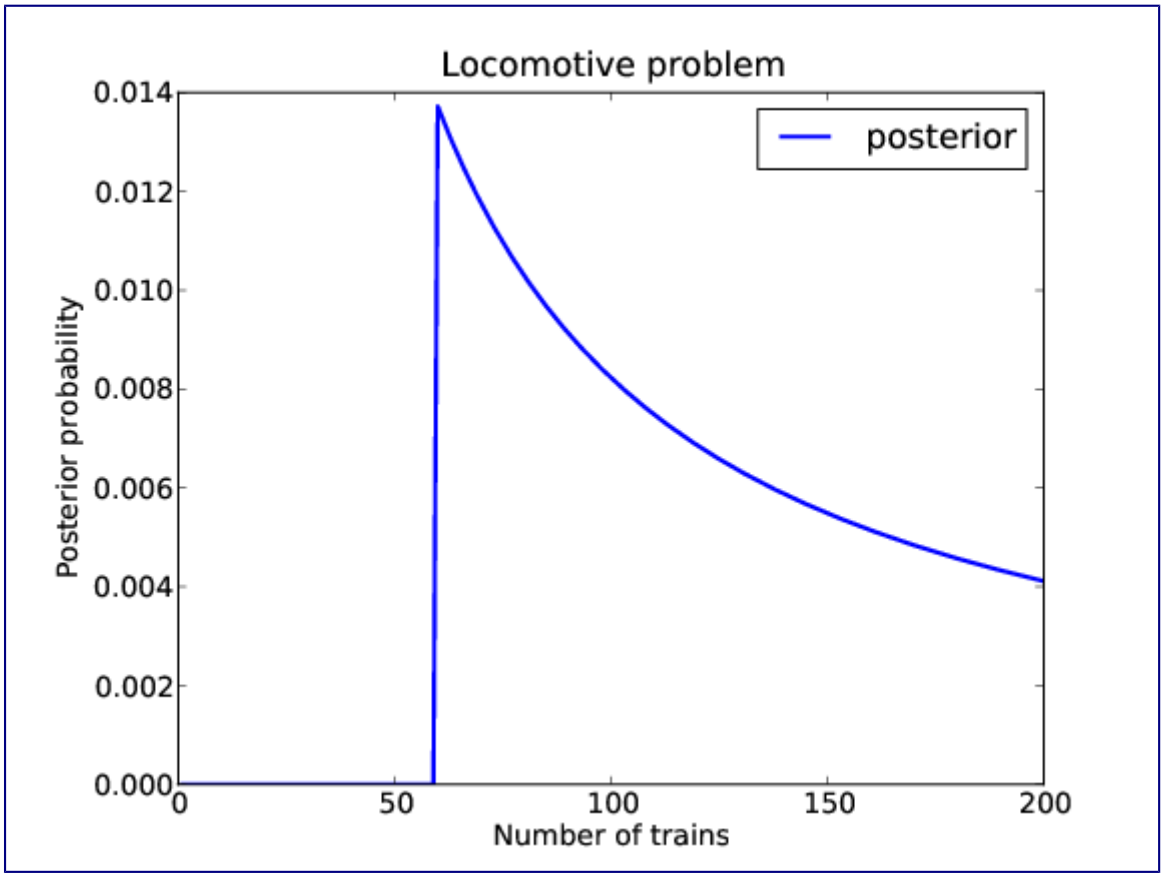
Và tìm giá trị của *a* sao cho ME = 0, vốn bằng

$a = 2N/(N - 1)$

Với các giá trị *n* lớn, *a* hội tụ về 2, vì vậy ta có thể chọn $\hat{N} = 2i$.

Đến đây ta đã phát sinh được ba ước lượng, *i*, 3*i*/2, và 2*i*, với các đặc tính lần lượt là làm tối đa khả năng, làm tối thiểu sai số quân phương, và không chệch.

Song còn một cách phát sinh ước lượng nữa là chọn giá trị để làm cho trị trung bình tổng thể bằng trị trung bình mẫu. Nếu ta thấy tàu số *i*, thì trị trung bình mẫu chính là *i*; tổng thể tàu có cùng trị trung bình sẽ là $\hat{N} = 2i - 1$.



Phân bố hậu nghiệm của số tàu hỏa.

Cuối cùng, để tìm phân bố hậu nghiệm Bayes, ta đi tính

$P(H_n|i) = P(i|H_n)\, P(H_n) / P(i)$

Trong đó H_n là giả thiết rằng có n con tàu, và i là chứng cứ: ta đã nhìn thấy tàu số i. Một lần nữa, $P(i|H_n)$ bằng $1/n$ nếu $i < n$, và bằng 0 trong trường hợp còn lại. Hằng số chuẩn hóa, $P(i)$, chính là tổng của các tử số trong mỗi giả thiết.

Nếu phân bố tiên nghiệm có dạng đều từ 1 đến 200, ta sẽ bắt đầu với 200 giả thiết và tính khả năng cho từng giả thiết một. Bạn có thể tải về một cách làm từ `http://thinkstats.com/locomotive.py`. Hình locomotive cho thấy kết quả sẽ như thế nào.

Khoảng tin cậy 90% cho xác suất hậu nghiệm này là [63; 189], vốn vẫn còn khá rộng. Việc thấy được một con tàu chưa đủ là bằng chứng mạnh để ủng hộ một giả thiết bất kì nào (dù rằng nó đã loại bỏ một loạt các giả thiết có $n < i$).

Nếu ta khởi đầu với một điều kiện tiên nghiệm khác thì xác suất hậu nghiệm sẽ khác hẳn; điều này góp phần giải thích tại sao các ước lượng khác lại phân tán đến mức vậy.

Một cách nghĩ về các ước lượng khác nhau là chúng được ngầm dựa theo các điều kiện tiên nghiệm khác nhau. Nếu như có đủ chứng cứ để xóa nhòa các điều kiện tiên nghiệm, thì tất cả các ước lượng sẽ có xu hướng hội tụ lại; còn nếu không, như trong trường hợp này, thì chẳng có một ước lượng nào mang đủ những tính chất mà ta mong muốn.

> Hãy khái quát hóa `locomotive.py` để xử lí trường hợp khi bạn nhìn thấy nhiều tàu. Bạn chỉ cần thay đổi vài dòng lệnh.
>
> Thử xem bạn có thể trả lời được các câu hỏi khác trong trường hợp khi bạn nhìn thấy nhiều tàu. Bạn có thể xem thêm thông tin về bài toán này cùng một số lời giải ở `http://wikipedia.org/wiki/German_tank_problem`.

Glossary

ước lượng:
: Quá trình suy luận ra các tham số của một dạng phân bố từ một mẫu.

ước lượng (danh từ):
: Đặc trưng thống kê được dùng để ước lượng cho một tham số.

sai số quân phương:
: Một độ đo cho sai số của ước lượng.

ước lượng khả năng cao nhất:
: Ước lượng nhằm tính ra được ước lượng điểm với khả năng xảy ra lớn nhất.

chệch:
: Xu hướng của ước lượng nằm trên hoặc dưới giá trị tham số thực tế, sau khi lấy trung bình qua các mẫu được lặp đi lặp lại.

ước lượng điểm:
: Ước lượng được biểu diễn dưới dạng một điểm.

khoảng tin cậy:
: Ước lượng được biểu diễn dưới dạng một khoảng với xác suất cho trước về việc sẽ chứa trong đó giá trị thực của tham số.

số liệu bị kiểm duyệt:
: Bộ số liệu được lấy mẫu theo cách hệ thống để loại bớt một phần số liệu.

1. Xem http://wikipedia.org/wiki/Exponential_distribution\#Maximum_likelihood. ↩

Chương 9: Tương quan

Trở về Muc luc cuốn sách

Điểm chuẩn

Trong chương này ta sẽ xem xét quan hệ giữa các biến. Chẳng hạn, ta có thể cảm thấy rằng chiều cao có liên quan đến cân nặng; những người cao thì cũng có khuynh hướng nặng hơn. **Tương quan** được dùng để mô tả những mối quan hệ kiểu này.

Một thử thách trong việc đo đạc mối tương quan là việc các biến cần so sánh lại không được biểu diễn theo cùng đơn vị. Chẳng hạn, chiều cao có thể tính bằng cm và khối lượng theo kg. Và ngay cả khi chúng có cùng đơn vị, có thể chúng lại xuất phát từ những dạng phân bố khác nhau.

Có hai giải pháp thường dùng để giải quyết các vấn đề nêu trên:

1. Chuyển đổi tất cả các giá trị về **điểm tiêu chuẩn**. Điều này dẫn đến hệ số tương quan Pearson.
2. Chuyển đổi tất cả các giá trị về hạng phần trăm của chúng. Điều này dẫn đến hệ số tương quan Spearman.

Nếu X là một dãy các giá trị, X_i, thì ta có thể chuyển đổi về điểm tiêu chuẩn bằng cách trừ đi trị trung bình và chia cho độ lệch chuẩn: $z_i = (x_i - \mu) / \sigma$.

Tử số là một độ lệch: khoảng cách từ giá trị đến trị trung bình. Bằng cách chia cho σ ta đã **chuẩn hóa** độ lệch, bởi vì các giá trị của Z là không thứ nguyên (không có đơn vị) và phân bố của chúng có trị trung bình bằng 0 và phương sai bằng 1.

Nếu X tuân theo phân bố chuẩn, thì Z cũng vậy; nhưng nếu X có phân bố lệch hoặc bao gồm điểm biệt lập, thì Z cũng vậy. Trong các trường hợp đó, cách dùng hạng phần trăm sẽ vững hơn. Nếu R là các hạng phần trăm của các giá trị tương ứng với X, thì R sẽ có phân bố đều từ 0 đến 100, bất kể dạng phân bố của X là gì.

Hiệp phương sai

Hiệp phương sai là một độ đo của xu hướng mà hai biến có thay đổi cùng nhau. Nếu ta có hai dãy, X và Y, thì độ lệch của chúng so với các trị trung bình sẽ là

$$dX_i = X_i - \mu_X$$

$$dY_i = Y_i - \mu_Y$$

trong đó μ_X là trị trung bình của X và μ_Y là trị trung bình của Y. Nếu X và Y biến đổi cùng nhau thì các độ lệch của chúng có xu hướng cùng dấu nhau.

Nếu ta nhân chúng lại thì tích số sẽ dương nếu các độ lệch có cùng dấu và âm nếu chúng ngược dấu. Vì vậy việc cộng các tích số sẽ cho ta một độ đo xu hướng biến đổi cùng nhau.

Hiệp phương sai là trị trung bình của các tích số này:

$$\text{Cov}(X,Y) = (1/n)\sum dx_i dy_i$$

trong đó n là độ dài của hai dãy số (chúng phải có cùng độ dài).

Hiệp phương sai có ích trong một số tính toán, nhưng ít khi nó được công bố như một đặc trưng thống kê vì khó diễn giải được. Một trong những lí do là việc đơn vị của hiệp phương sai là tích số các đơn vị của X và Y. Vì vậy hiệp phương sai của khối lượng và chiều cao có khi lại mang đơn vị kilogam-mét, điều này chẳng có nhiều ý nghĩa.

> Hãy viết một hàm có tên là `Cov` nhận vào hai danh sách rồi tính hiệp phương sai của chúng. Để kiểm tra hàm viết được, hãy tính hiệp phương sai của một danh sách với chính nó rồi kiểm tra để khẳng định rằng Cov(X, X) = Var(X).
>
> Bạn có thể tải về một lời giải từ http://thinkstats.com/correlation.py.

Tương quan

Một cách giải quyết vấn đề này là đem chia các độ lệch cho σ, từ đó thu được điểm tiêu chuẩn, rồi tính tích số của các điểm tiêu chuẩn này:

$$p_i = \{(x_i - \mu_X)/\sigma_X\}\ \{(y_i - \mu_Y)/\sigma_Y\}$$

Giá trị trung bình của các tích này là

$$\rho = (1/n)\sum p_i$$

Giá trị này được gọi là **hệ số tương quan Pearson** đặt tên theo Karl Pearson, một "cây đại thụ" trong ngành thống kê học. Giá trị có thể dễ dàng tính toán và diễn giải. Vì các điểm chuẩn không có thứ nguyên nên ρ cũng vậy.

Ngoài ra, kết quả tìm được phải nằm trong khoảng từ –1 đến +1. Để hiểu được nguyên nhân, ta cần viết lại ρ bằng cách phân tích thành các thừa số σ_X và σ_Y:

$$\rho = \{\text{Cov}(X,Y)\} / \{\sigma_X \sigma_Y\}$$

Diễn giải dưới dạng các độ lệch, ta có

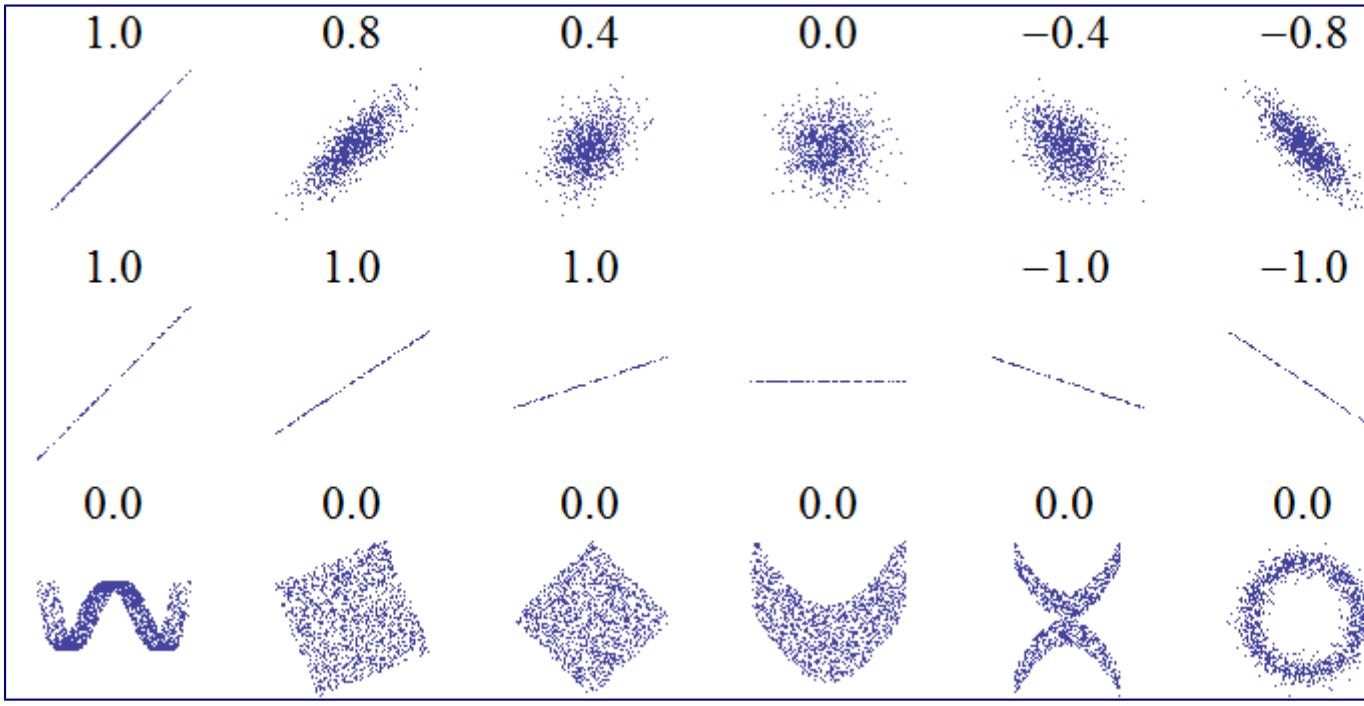
$$\rho = \{\sum dx_i dy_x\} / \{\sum dx_i \sum dy_i\}$$

Sau đó theo bất đẳng thức Cô-si – Svac (Cauchy-Schwarz) kinh điển[1], có thể cho thấy $\rho^2 \le 1$, do vậy $-1 \le \rho \le 1$.

Độ lớn của ρ cho thấy độ mạnh của tương quan. Nếu ρ = 1 thì các biến số có tương quan hoàn hảo, nghĩa là nếu bạn đã biết một biến thì bạn sẽ đoán đúng được biến còn lại. Cũng vậy nếu ρ = –1. Nó có nghĩa rằng nếu các biến có tương quan âm, nhưng để phục vụ cho mục đích dự đoán, thì tương quan âm cũng có ích như tương quan dương.

Trên thực tế hầu hết các tương quan đều không hoàn hảo, nhưng vẫn rất có ích. Chẳng hạn, nếu bạn biết cân nặng của một người thì bạn có thể đoán được chiều cao của người đó. Bạn có thể không đoán đúng, nhưng vẫn còn tốt hơn trong trường hợp bạn đoán không dựa vào số chiều cao. Số tương quan Pearson là một độ đo sự tốt hơn này.

Vì vậy nếu ρ = 0, liệu điều này có nghĩa rằng không có quan hệ giữa các biến này không? Không may là điều này không đúng. Tương quan Pearson chỉ đo các quan hệ *tuyến tính*. Nếu tồn tại một quan hệ phi tuyến thì ρ sẽ xem nhẹ mức độ phụ thuộc giữa hai biến.



Ví dụ về các bộ số liệu với một loạt các tương quan. [corr_examples

Hình trên được lấy từ http://wikipedia.org/wiki/Correlation_and_dependence. Nó cho thấy các biểu đồ điểm chấm và hệ số tương quan của một vài bộ số liệu được thiết kế riêng.

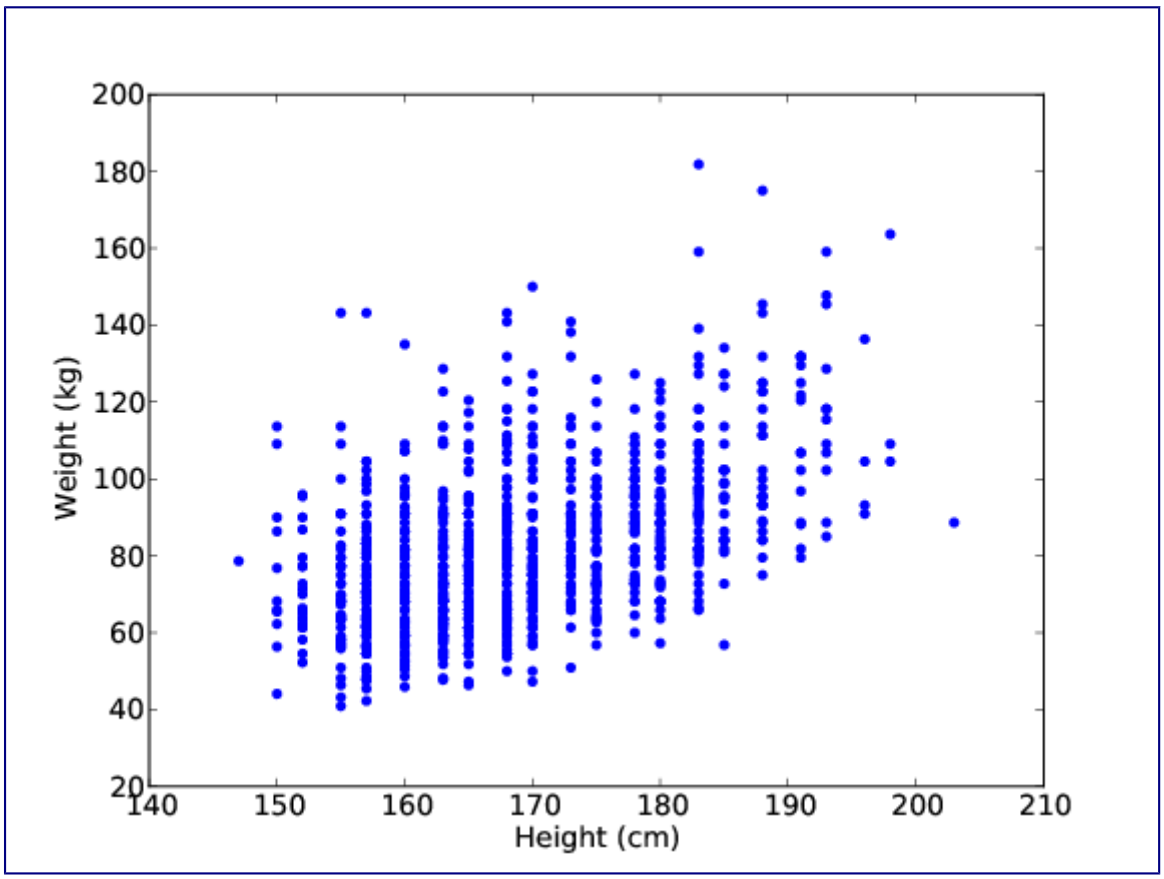
Các biểu đồ trên hàng đầu cho thấy quan hệ tuyến tính với một loạt các sự tương quan; bạn có thể dựa trên hàng này để hình dung được xem ρ bằng bao nhiêu. Hàng thứ hai biểu thị các tương quan hoàn hảo với một loạt các độ dốc, cũng cho thấy rằng tương quan thì không có can hệ gì đến độ dốc (ta sẽ sớm nói về cách ước tính độ dốc). Hàng thứ ba cho thấy các biến rõ ràng là có quan hệ với nhau, nhưng vì mối quan hệ có tính phi tuyến, nên hệ số tương quan bằng 0.

Ý nghĩa của câu chuyện này là bạn luôn nên nhìn vào một biểu đồ điểm chấm các điểm số liệu trước khi tính toán hệ số tương quan một cách mù quáng.

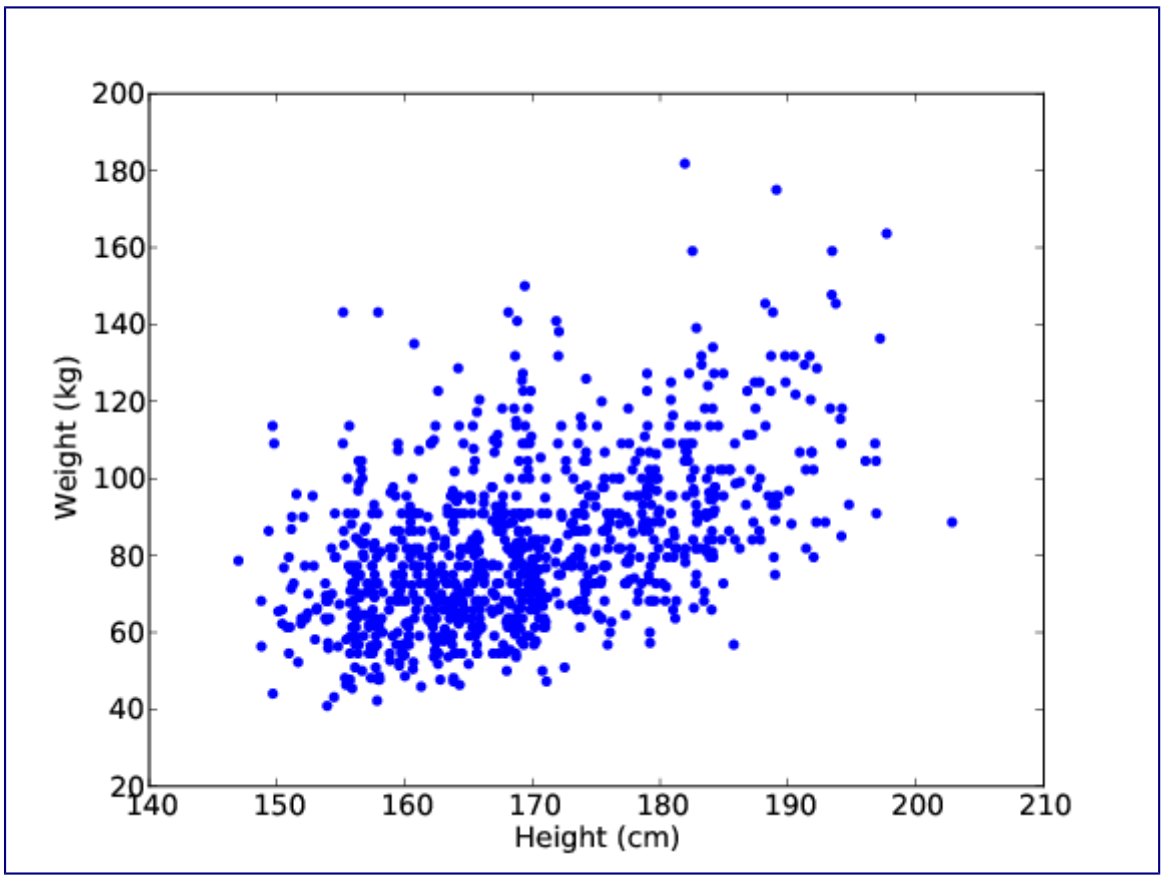
Hãy viết một hàm có tên `Corr` nhận vào hai biến và tính tương quan của chúng. Gợi ý: hãy dùng `thinkstats.Var` và hàm `Cov` đã viết ở bài tập trước.

Để thử nghiệm hàm vừa viết được, hãy dùng nó để tính hiệp phương sai của một danh sách đối với chính nó và kiểm tra rằng Corr(*X*, *X*) bằng 1. Bạn có thể tải về một lời giải từ http://thinkstats.com/correlation.py.

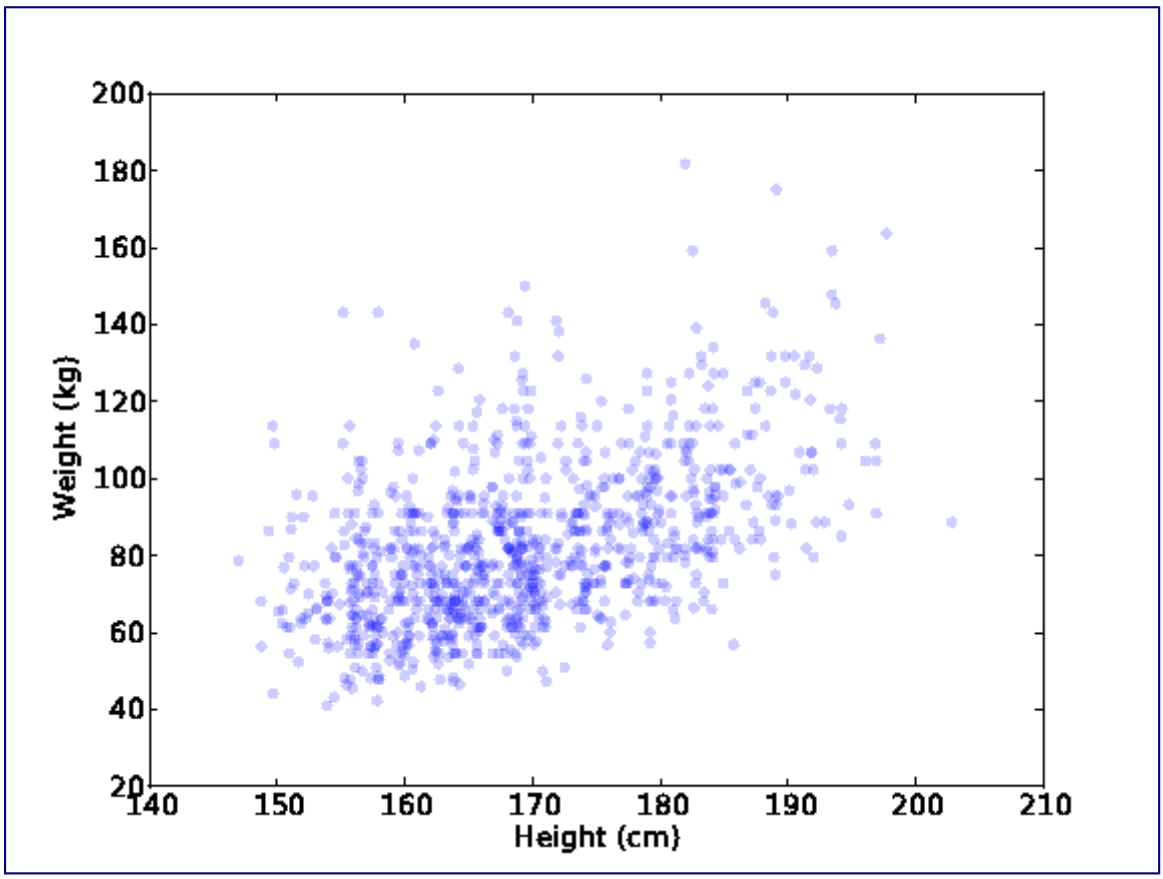
Vẽ đồ thị điểm chấm bằng pyplot



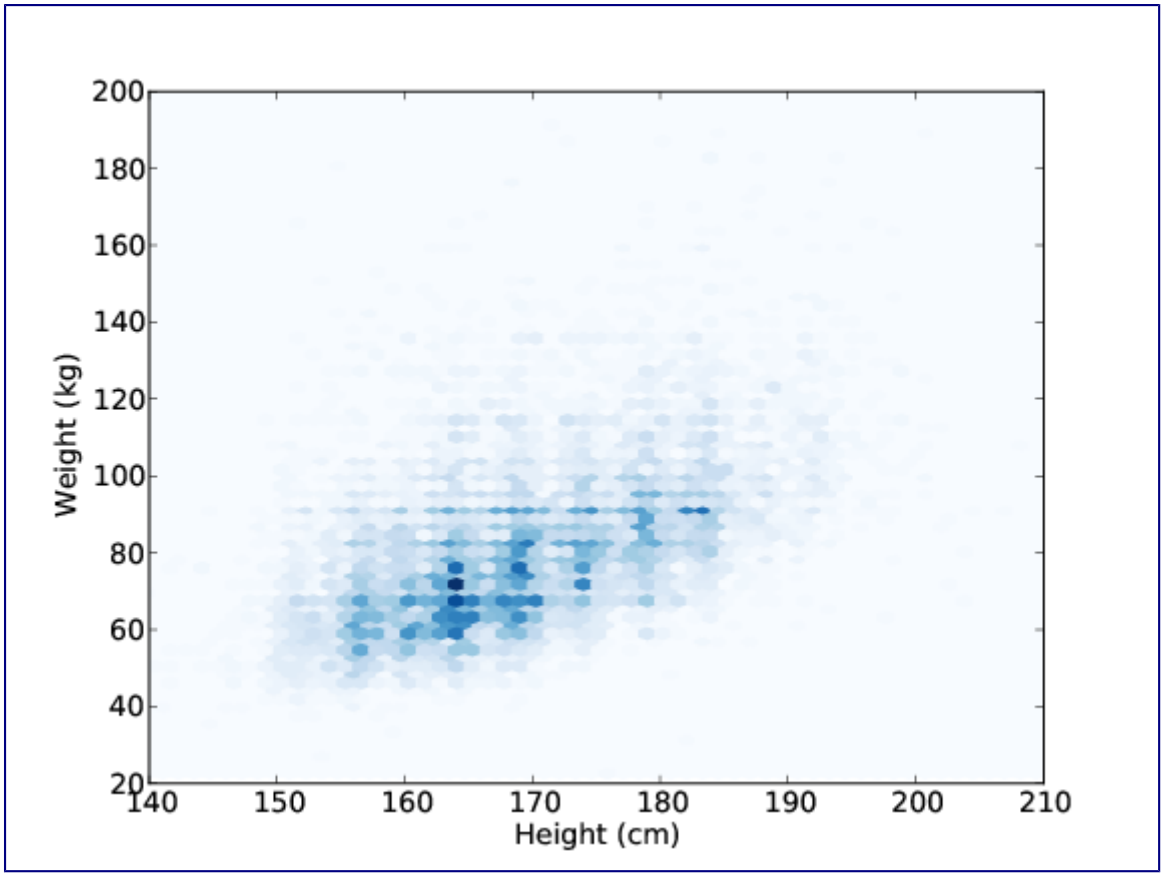
Đồ thị điểm chấm đơn giản giữa cân nặng và chiều cao của những người được điều tra trong BRFSS. [scatterplot1



Đồ thị điểm chấm với số liệu dao động. [scatterplot2



Đồ thị điểm chấm có độ đậm nhạt với số liệu dao động. [scatterplot3



Đồ thị điểm chấm với số liệu được chia ngăn vẽ bằng pyplot.hexbin. [scatterplot4

Cách dễ nhất để kiểm tra quan hệ giữa hai biến là một biểu đồ điểm chấm, nhưng để vẽ đẹp biểu đồ này thì không phải luôn dễ dàng. Chẳng hạn, tôi sẽ chấm điểm cân nặng theo chiều cao của những người tham gia vào BRFSS (xem Mục [phân bố loga chuẩn]). `pyplot` cho ta hàm có tên `scatter` để vẽ các đồ thị kiểu này:

```
import matplotlib.pyplot as pyplot
pyplot.scatter(heights, weights)
```

Hình scatterplot1 biểu thị kết quả. Không ngạc nhiên là nó trông như một xu hướng tương quan dương: những người cao thường có xu hướng nặng hơn. Nhưng đây không phải là cách biểu thị số liệu tốt nhất, vì số liệu được gộp vào các cột. Vấn đề là chiều cao được làm tròn đến số inch gần nhất, được đổi sang centimet, và sau đó lại được làm tròn tiếp. Có thông tin đã bị mất đi trong quá trình chuyển đổi.

Ta không thể lấy lại thông tin đã mất, nhưng có thể giảm thiểu hiệu ứng này đối với biểu đồ bằng cách làm **dao động** số liệu, nghĩa là sẽ thêm vào các nhiễu động ngẫu nhiên để cân bằng với hiệu ứng gây ra bởi làm tròn. Vì những số đo này được làm tròn đến inch gần nhất, nên chúng có thể sai lệch tới 0,5 inch hay 1,3 cm. Vì vậy tôi đem cộng vào các lượng nhiễu động trong khoảng từ –1,3 đến 1,3:

```
jitter = 1.3
heights = [h + random.uniform(-jitter, jitter) for h in heights]
```

Hình scatterplot2 cho thấy kết quả. Việc làm dao động số liệu khiến cho hình dáng mối quan hệ trở nên rõ hơn. Nói chung bạn chỉ nên làm dao động số liệu để phục vụ cho việc hiển thị và tránh dùng cách dao động này để phân tích tính toán.

Ngay cả khi đã dao động thì đó cũng không phải là cách biểu diễn số liệu tốt nhất. Còn có nhiều điểm trùng lặp, vốn làm ẩn đi các số liệu trong những phần dày đặc trên hình vẽ và nhấn mạnh quá mức các điểm biệt lập.

Ta có thể giải quyết vấn đề này bằng tham số `alpha`, có tác dụng làm các điểm chấm trở nên phần nào trong suốt:

```
pyplot.scatter(heights, weights, alpha=0.2)
```

Hình scatterplot3 biểu diễn kết quả. Các điểm số liệu chồng lên nhau sẽ trông đậm hơn, vì vậy độ đậm nhạt sẽ tỉ lệ với mật độ. Trong cách vẽ này ta có thể phát hiện một hiệu ứng tồn đọng: một đường nằm ngang ở mức 90 kg hoặc 200 pound. Vì số liệu này được dựa trên các báo cáo tự giác, con số tính theo pound nên nhiều khả năng ta sẽ hiểu chúng được làm tròn (có thể là làm tròn xuống).

Bằng cách dùng độ trong sẽ phù hợp với các bộ số liệu cỡ trung bình, nhưng hình vẽ này chỉ cho thấy 1000 điểm số liệu đầu tiên trong BRFSS, trong tổng số 414509.

Để xử lý các bộ số liệu lớn hơn, một cách lựa chọn là dùng biểu đồ hexbin, theo đó biểu đồ sẽ được chia thành những ngăn lục giác và được tô màu căn cứ theo số điểm chấm rơi vào trong phạm vi từng ngăn. `pyplot` có một hàm tên là `hexbin`:

```
pyplot.scatter(heights, weights, cmap=matplotlib.cm.Blues)
```

Hình scatterplot4 cho thấy kết quả với dải màu xanh lam. Một ưu điểm khi dùng hexbin là nó cho thấy rõ hình dạng của mối quan hệ, và hiệu quả hơn khi xử lý các bộ số liệu lớn. Một nhược điểm là nó làm biến đi các điểm biệt lập.

Ý nghĩa của câu chuyện này muốn nói là không dễ tạo ra một biểu đồ điểm chấm mà không tiềm ẩn sự đánh lạc hướng người xem. Bạn có thể tải về đoạn mã phát sinh các biểu đồ này từ http://thinkstats.com/brfss_scatter.py.

Tương quan hạng Spearman

Tương quan Pearson phát huy tác dụng khi quan hệ giữa các biến là tuyến tính và các biến tương đối chuẩn hóa. Nhưng nó sẽ không vững vàng trong trường hợp có điểm biệt lập.

Bộ tứ Anscombe minh họa rõ hiệu ứng này; nó bao gồm bốn bộ số liệu với cùng độ tương quan. Một có tương quan tuyến tính với nhiễu ngẫu nhiên, một là quan hệ phi tuyến, một là quan hệ hoàn hảo với một điểm biệt lập, và một thì không có quan hệ nào chỉ trừ một hiệu ứng tồn đọng gây ra bởi một điểm biệt lập. Bạn có thể đọc thêm tại http://wikipedia.org/wiki/Anscombe's_quartet.

Tương quan hạng Spearman là một cách làm khác trong đó giảm thiểu hiệu ứng của các điểm biệt lập và các phân bố lệch. Để tính ra sự điều chỉnh của Spearman, ta phải tính **hạng** của từng giá trị, vốn là thứ tự của nó trong mẫu đã được sắp xếp. Chẳng hạn, trong mẫu {7, 1, 2, 5}, hạng của giá trị 5 là 3, vì nó sẽ xuất hiện ở vị trí thứ ba khi ta sắp xếp các phần tử. Sau đó ta tính tương quan Pearson cho các hạng này.

Một cách khác với cách của Spearman là áp dụng một phép chuyển đổi để khiến cho dữ liệu trở nên gần như được chuẩn hóa, sau đó mới tính tương quan Pearson cho số liệu được chuyển đổi. Ví dụ, nếu số liệu có dạng xấp xỉ với loga-chuẩn thì bạn có thể tính loga của mỗi giá trị rồi đi tính tương quan của các giá trị loga đó.

> Hãy viết một hàm nhận vào một dãy rồi trả lại một danh sách chứa hạng của từng phần tử. Chẳng hạn, nếu dãy là {7, 1, 2, 5}, thì kết quả sẽ phải là {4, 1, 2, 3}.
>
> Nếu một giá trị được lặp lại nhiều lần thì cách giải chặt chẽ nhất là gán mỗi giá trị với trị trung bình giữa các hạng của chúng. Nhưng nếu bạn bỏ qua điều này và gán thứ hạng tùy ý thì sai số cũng thường chỉ nhỏ thôi.
>
> Hãy viết một hàm nhận vào hai dãy (với cùng độ dài) rồi tính tương quan hạng Spearman. Bạn có thể tải về một lời giải từ http://thinkstats.com/correlation.py.
>
> Hãy tải về các file http://thinkstats.com/brfss.py và http://thinkstats.com/brfss_scatter.py. Chạy chúng để khẳng định chắc rằng bạn đọc được số liệu BFRSS và vẽ được các biểu đồ điểm chấm.
>
> So sánh các biểu đồ chấm với Hình corr_examples, giá trị tương quan Pearson mà bạn trông đợi là bao nhiêu? Giá trị nhận được là bao nhiêu?

Bởi vì dạng phân bố của cân nặng người lớn thuộc loại loga chuẩn nên có những điểm biệt lập ảnh hưởng đến tương quan. Hãy thử chấm điểm loga(cân nặng) theo chiều cao, rồi tính hệ số tương quan Pearson cho biến mới được chuyển đổi này.

Sau cùng, hãy tính tương quan hạng Spearman cho cân nặng và chiều cao. Theo bạn thì hệ số nào là đại diện tốt hơn cho độ chặt chẽ của mối quan hệ? Bạn có thể tải về một lời giải từ `http://thinkstats.com/brfss_corr.py`.

Phép khớp bình phương nhỏ nhất

Hệ số tương quan cho thấy độ chặt chẽ của mối quan hệ cùng với dấu của nó, nhưng không cho ta biết độ dốc của đường quan hệ. Có một vài cách giúp ước tính độ dốc: cách thông dụng nhất là **phép khớp bình phương nhỏ nhất**. Một "phép khớp tuyến tính" là một đường thẳng được dùng với ý định mô phỏng mối quan hệ giữa các biến. Một phép khớp "bình phương nhỏ nhất" sẽ làm cực tiểu sai số quân phương (MSE) giữa đường thẳng và các điểm số liệu[2].

Chẳng hạn chúng ta có một dãy các điểm, *Y*, mà ta muốn biểu thị như một hàm của một dãy khác, *X*. Nếu giữa *X* và *Y* có mối quan hệ tuyến tính rõ rệt là đường thẳng với giao điểm α với trục tung, và độ dốc β, thì ta sẽ trông đợi rằng mỗi Y_i sẽ xấp xỉ $\alpha + \beta X_i$.

Nhưng trừ phi tương quan là hoàn hảo, ước đoán này chỉ có tính gần đúng. Độ sai lệch, hay **số dư**, sẽ bằng

$$\varepsilon_i = (\alpha + \beta X_i) - Y_i$$

Số dư có thể là do các yếu tố ngẫu nhiên như sai số đo đạc, hoặc do các nhân tố không ngẫu nhiên mà ta chưa biết. Chẳng hạn, nếu ta thử đoán cân nặng như một hàm theo chiều cao, thì các yếu tố chưa biết sẽ có thể bao gồm chế độ ăn kiêng, luyện tập thể dục, và tạng người.

Nếu ta tính sai các tham số α và β, thì sô dư sẽ lớn hơn, vì vậy theo trực giác thì các giá trị tham số ta cần chính là những giá trị làm cực tiểu những số dư.

Như thường lệ, ta có thể làm cực tiểu giá trị tuyệt đối của các số dư, bình phương, hoặc lập phương của chúng, v.v. Cách thông dụng nhất là làm cực tiểu tổng các bình phương của số dư

$$\min_{\alpha,\beta} \sum \varepsilon_i^2$$

Vì sao? Có ba điều hay và một điều dở trong cách làm nêu trên:

- Cách lấy bình phương có đặc điểm hiển nhiên là nó coi các số dư dương và âm với vai trò như nhau, đây thường là điều mà ta mong muốn.
- Bình phương đặt thêm trọng lượng đối với các số dư lớn, nhưng không nặng quá khiến cho số dư lớn nhất luôn áp đảo.
- Nếu các số dư đều độc lập đối với *X*, ngẫu nhiên, và tuân theo phân bố chuẩn với μ = 0 và σ là hằng số (nhưng chưa biết), thì cách khớp bình phương cũng là ước lượng khả năng cao nhất của α và β.[3]
- Các giá trị và $\hat{\beta}$ làm cực tiểu các số dư bình phương có thể được tính ra một cách hiệu quả.

Chính đặc điểm cuối sẽ có ý nghĩa khi hiệu năng tính toán quan trọng hơn là chọn được phương pháp phù hợp với bài toán hiện tại. Đến nay, điều này không còn đúng nữa, vì vậy mà ta cũng cần phải xét xem liệu rằng các số dư bình phương có phải là đại lượng đúng cần được giảm thiểu hay không.

Chẳng hạn, nếu bạn dùng các giá trị của *X* để dự đoán các giá trị của *Y*, việc đoán quá cao có thể sẽ tốt hơn (hoặc dở hơn) so với ước đoán quá thấp. Trong trường hợp này, bạn có thể muốn tính một hàm giá thành nào đó, cost(ε_i), rồi làm cực tiểu giá thành.

Tuy nhiên, việc tính khớp bình phương nhỏ nhất thì rất nhanh, dễ dàng, và thường là đủ đáp ứng yêu cầu; và sau đây là cách tính:

1. Tính các trung bình mẫu,  và  , phương sai của *X*, và hiệp phương sai của *X* và *Y*.
2. Độ dốc của ước lượng là $\hat{\beta}$ = {Cov(*X*,*Y*)} / {Var(*X*)}
3. Và giao điểm với trục tung là $\hat{\alpha} = \bar{Y} - \hat{\beta}\bar{X}$

Để thấy được cách biến đổi để thu được các công thức trên, bạn có thể xem http://wikipedia.org/wiki/Numerical_methods_for_linear_least_squares.

> Hãy viết một hàm có tên `LeastSquares` nhận vào *X* và *Y* rồi tính  và $\hat{\beta}$. Bạn có thể tải về một lời giải từ http://thinkstats.com/correlation.py.
>
> Lại dùng số liệu của BRFSS để tính khớp bình phương nhỏ nhất cho log(cân nặng) theo chiều cao. Bạn có thể tải về một lời giải từ http://thinkstats.com/brfss_corr.py.
>
> Sự phân bố của tốc độ gió tại một địa điểm quyết định mật độ công suất gió, vốn là một giới hạn trên cho công suất trung bình mà tua-bin gió phát huy được nếu đặt ở địa điểm này. Dựa theo một số nguồn tài liệu, phân bố tốc độ gió đo được có thể mô phỏng rất phù hợp bằng một phân bố Weibull (xem http://wikipedia.org/wiki/Wind_power#Distribution_of_wind_speed).
>
> Để ước lượng được xem liệu một địa điểm có thể thích hợp để đặt tua-bin gió hay không, bạn có thể lập một trạm đo tốc độ gió trong một khoảng thời gian. Nhưng sẽ rất khó để đo được chính xác phần đuôi của dải phân bố tốc độ gió, vì theo định nghĩa, những biến cố ứng với phần đuôi này không mấy khi xảy ra.
>
> Một cách giải quyết vấn đề này là dùng các số liệu thực đo để ước lượng các tham số của một phân bố Weibull, rồi lấy tích phân trên PDF liên tục để tính mật độ công suất gió.
>
> Để ước tính các tham số của phân bố Weibull, ta có thể dùng phép chuyển đổi từ Bài tập weibull rồi dùng một phép khớp tuyến tính để tìm ra độ dốc và tung độ giao điểm của số liệu được chuyển đổi.

> Hãy viết một hàm nhận vào một mẫu từ phân bố Weibull rồi ước lượng các tham số.
>
> Sau đó, viết một hàm nhận vào các tham số của một phân bố Weibull cho tốc độ gió rồi tính mật độ công suất gió trung bình (bạn có thể phải tìm đọc thêm tài liệu để làm được phần này).

Độ phù hợp của phép khớp

Sau khi đã khớp một mô hình tuyến tính cho số liệu, ta có thể muốn biết xem nó tốt bao nhiêu. Điều này còn tùy vào mục tiêu của bản thân việc khớp. Một cách đánh giá mô hình là nằm ở khả năng dự báo của nó.

Xét về mặt dự đoán thì đại lượng mà ta đang cố gắng đoán được gọi là **biến phụ thuộc** và đại lượng mà ta dùng để dự đoán được gọi là **biến độc lập**.

Để đo đạc khả năng dự báo của một mô hình, ta có thể tính **hệ số xác định**, thường được biết đến với tên gọi "R-bình phương":

$R^2 = 1 - \{\mathrm{Var}(\varepsilon)\} / \{\mathrm{Var}(Y)\}$

Để hiểu được R^2 nghĩa là gì, hãy (một lần nữa) giả sử rằng bạn đang thử đoán cân nặng của một ai đó. Nếu bạn không biết gì về họ thì cách tốt nhất có thể sẽ là đoán $\square$; trong trường hợp này sai số quân phương của con số mà bạn dự đoán sẽ là Var(Y):

$MSE = (1/n)\sum(\square - y_i)^2 = \mathrm{Var}(Y)$

Nhưng nếu tôi bảo cho bạn biết chiều cao của họ, bạn sẽ có thể đoán $\square + \hat{\beta} X_i$; trong trường hợp đó sai số quân phương sẽ là Var(ε).

$MSE = (1/n)\sum(\square + \hat{\beta} x_i - y_i)^2 = \mathrm{Var}(\varepsilon)$

Vì vậy đại lượng Var(ε)/Var(Y), tỉ số giữa các sai số quân phương khi có và không có biến độc lập, vốn là tỉ lệ của sự biến đổi mà mô hình vẫn còn bỏ ngỏ. Phần bù của đại lượng đó, R^2, là tỉ lệ của sự biến đổi được mô hình giải thích.

Nếu một mô hình có $R^2 = 0{,}64$, bạn có thể nói rằng mô hình giải thích được 64% sự biến đổi, hay có lẽ chính xác hơn là nó làm giảm sai số quân phương của các con số dự đoán đi 64%.

Trong trường hợp mô hình bình phương nhỏ nhất tuyến tính, hóa ra là có một mối quan hệ đơn giản giữa hệ số xác định và hệ số tương quan Pearson, ρ:

$R^2 = \rho^2$

Xem http://wikipedia.org/wiki/Howzzat!

Thang đo trí thông minh người lớn Wechsler (Wechsler Adult Intelligence Scale, WAIS) được dùng để cho điểm trí thông minh; các điểm được bố trí sao cho trị trung bình và độ lệch chuẩn của dân số nói chung sẽ là 100 và 15.

Giả sử bạn muốn dự đoán điểm WAIS của ai đó dựa trên điểm thi SAT của họ. Căn cứ vào một kết quả nghiên cứu, có một tương quan với hệ số Pearson bằng 0,72 giữa điểm thi SAT và điểm WAIS.

Nếu bạn áp dụng công thức dự báo cho một mẫu lớn thì bạn sẽ trông đợi giá trị sai số quân phương (MSE) của việc dự báo này bằng bao nhiêu?

Gợi ý: MSE sẽ bằng bao nhiêu nếu bạn luôn luôn đoán số 100?

Hãy viết một hàm có tên `Residuals` nhận vào *X*, *Y*, $\hat{\beta}$ và $\hat{\beta}$ rồi trả lại một danh sách chứa ε_2.

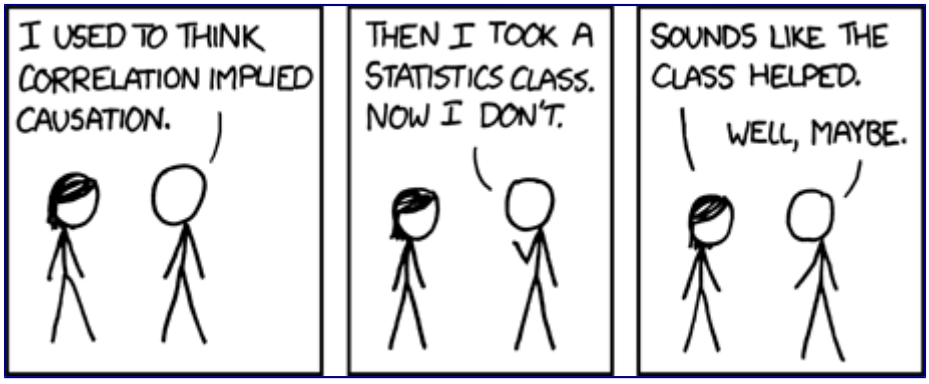
Hãy viết một hàm có tên `CoefDetermination` nhận vào các ε_i và *Y* rồi trả lại R^2.

Để kiểm tra các hàm vừa viết, hãy cho thấy $R^2 = \rho^2$. Bạn có thể tải về một lời giải từ http://thinkstats.com/correlation.py.

Hãy dùng số liệu về chiều cao và cân nặng của BRFSS (một lần nữa), tính $\hat{\beta}$, $\hat{\beta}$ và R^2. Nếu bạn muốn đoán cân nặng của ai đó, thì việc biết trước cân nặng của họ sẽ giúp bạn thêm bao nhiêu? Bạn có thể tải về một lời giải từ http://thinkstats.com/brfss_corr.py.

Tương quan và nhân-quả

Truyện tranh trào phúng trên mạng, `xkcd`, cho ta thấy sự khó khăn khi suy diễn luật nhân-quả:



Lấy từ xkcd.com, vẽ bởi Randall Munroe.

Nhìn chung, mối quan hệ giữa hai biến không cho bạn biết một sự việc này gây ra sự việc kia, hoặc ngược lại, hoặc cả hai phía, hoặc liệu chúng đồng thời được gây ra bởi một điều khác.

Quy luật này có thể được tóm tắt lại bởi câu nói "Tương quan không đồng nghĩa với quan hệ nhân-quả," cô đọng đến nỗi có có riêng một trang Wikipedia: http://wikipedia.org/wiki/Correlation_does_not_imply_causation.

Vậy thì bạn có thể làm gì để chứng tỏ được sự tồn tại của quan hệ nhân-quả?

1. Dùng thời gian. Nếu A xảy ra trước B, thì A có thể gây ra B nhưng không thể ngược lại (ít nhất là theo hiểu biết hiện tại của chúng ta về luật nhân quả). Thứ tự của các sự kiện có thể giúp ta suy diễn ra hướng của nhân-quả, nhưng nó không loại trừ khả năng một điều nào đó khác gây ra cả A và B.
2. Dùng sự ngẫu nhiên. Nếu bạn chia một tổng thể lớn ra thành hai nhóm một cách ngẫu nhiên, và tính các trị trung bình của hầu hết tất cả các biến, thì bạn sẽ trông đợi rằng sự khác biệt là rất nhỏ. Đây là hệ quả của Định lý giới hạn trung tâm (vì vậy nó cũng phải tuân theo cùng các yêu cầu).Nếu hai nhóm gần như tương đồng về tất cả các biến nhưng chỉ trừ một biến thì bạn có thể loại trừ các quan hệ ngụy tạo.Điều này có tác dụng ngay cả khi bạn không biết các biến có liên quan là gì, nhưng nó sẽ có tác dụng hơn khi bạn biết thông tin đó, vì khi ấy bạn có thể kiểm tra để khẳng định là hai nhóm giống hệt nhau.

Các ý tưởng này là cơ sở hình thành **phép thử ngẫu nhiên có kiểm soát**, trong đó chủ thể được gán ngẫu nhiên cho hai (hoặc nhiều) nhóm; một nhóm **điều trị** nhận được sự can thiệp nhất định, như một loại thuốc mới, và một **nhóm kiểm soát** không nhận được can thiệp nào, hoặc điều trị theo một cách mà ta biết trước được những hiệu ứng.

Phép thử ngẫu nhiên có kiểm soát là cách làm tin cậy nhất để chứng minh một quan hệ nhân-quả, và cũng là nền móng cho y thuật dựa trên khoa học (xem http://wikipedia.org/wiki/Randomized_controlled_trial).

Không may là những phép thử có kiểm soát chỉ dùng được trong phòng thí nghiệm, trong y học và một số ít các lĩnh vự khác. Trong khoa học xã hội, những thí nghiệm có kiểm soát là rất hiếm, thường là vì chúng không thể thực hiện được hoặc phi đạo đức.

Một cách làm khác là đi tìm **thí nghiệm tự nhiên**, trong đó các phép "điều trị" khác nhau được áp dụng cho các nhóm người giống nhau. Một sự nguy hiểm của thí nghiệm tự nhiên là các nhóm có thể khác nhau theo cách không dễ nhận ra. Bạn có thể đọc thêm về chủ đề này ở http://wikipedia.org/wiki/Natural_experiment.

Trong một số trường hợp, ta có thể suy diễn quan hệ nhân-quả bằng **phân tích hồi quy**. Một phép khớp bình phương nhỏ nhất tuyến tính là một dạng đơn giản của hồi quy trong đó xác định biến phụ thuộc bằng một biến độc lập. Còn có những kĩ thuật tương tự áp dụng cho bao nhiêu biến độc lập cũng được.

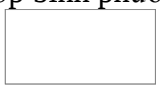
Tôi không đề cập đến các kĩ thuật đó ở đây, nhưng cũng có những cách đơn giản để kiểm soát những mối quan hệ ngụy tạo. Chẳng hạn, trong số liệu NSFG, ta thấy được rằng trẻ đầu lòng thường có xu hướng nhẹ hơn trẻ sinh sau (xem Mục [cân nặng trẻ sơ sinh]). Nhưng cân nặng trẻ sơ sinh cũng tương quan với tuổi của người mẹ, và mẹ của bé đầu lòng thường cũng trẻ hơn mẹ của các bé sinh sau.

Vì vậy có thể trẻ đầu lòng nhẹ hơn vì mẹ của chúng trẻ hơn. Để kiểm soát hiệu ứng của tuổi tác, ta có thể chia các bà mẹ theo những nhóm tuổi và so sánh cân nặng trẻ sơ sinh của các bé đầu lòng và sinh sau đối với từng nhóm tuổi.

Nếu sự khác biệt giữa các trẻ đầu lòng và sinh sau cũng tương đồng trong các nhóm tuổi khác n hau cũng như trong pooled data (số liệu tổng hợp), thì ta kết luận rằng khác biệt này không liên quan

đến tuổi tác. Nếu không có sự khác biệt nào, thì ta kết luận rằng hiệu ứng này hoàn toàn là do tuổi tác. Hoặc, nếu khác biệt là nhỏ hơn thì ta có thể định lượng xem bao nhiêu phần của hiệu ứng là do tuổi tác.

Số liệu NSFG bao gồm một biến có tên `agepreg` để ghi lại tuổi của bà mẹ tại thời điểm sinh bé. Hãy vẽ biểu đồ điểm chấm cho tuổi của mẹ và cân nặng của bé cho các ca sinh thành công. Bạn có thể thấy mối quan hệ nào không?

Hãy tính cách khớp bình phương nhỏ nhất cho các biến này. Những đơn vì của các tham số được ước tính,  và $\hat{\beta}$ là gì? Bạn có thể tóm tắt kết quả tìm được bằng 1 hoặc 2 câu ngắn gọn được không?

Hãy tính tuổi trung bình của mẹ các đứa trẻ đầu lòng và của mẹ các đứa trẻ sinh sau. Dựa vào hiệu số tuổi giữa hai nhóm này, bạn sẽ trông đợi hiệu số giữa cân nặng trẻ sơ sinh trung bình của hai nhóm bằng bao nhiêu? Một phần bao nhiêu của hiệu số thực sự giữa cân nặng hai nhóm được giải thích qua hiệu số tuổi hai nhóm các bà mẹ?

Bạn có thể tải về một lời giải của bài này từ `http://thinkstats.com/agemodel.py`. Nếu bạn muốn biết về hồi quy nhiều biến, bạn có thể chạy `http://thinkstats.com/age_lm.py`, file này giới thiệu cách dùng gói tính toán thống kê của R trong Python. Nhưng đó sẽ là nội dung của cả một cuốn sách khác.

Thuật ngữ

tương quan:
: Việc mô tả sự phụ thuộc giữa các biến.

chuẩn hóa:
: Chuyển đổi một tập hợp các giá trị sao cho tập giá trị mới có trị trung bình bằng 0 và phương sai bằng 1.

điểm chuẩn:
: Giá trị sau khi được chuẩn hóa.

hiệp phương sai:
: Độ đo của xu hướng biến đổi cùng nhau giữa hai biến.

hạng:
: Chỉ số của một phần tử đứng trong danh sách đã được sắp xếp.

phép khớp bình phương nhỏ nhất:
: Mô hình cho một bộ số liệu nhằm làm cực tiểu tổng các bình phương của các số dư.

số dư:
: Độ lệch của một giá trị thực tế so với giá trị trong mô hình.

biến phụ thuộc:
: Biến mà ta đang cố gắng dự đoán hoặc diễn giải.

biến độc lập:
: Biến mà ta dùng để dự đoán một biến phụ thuộc, còn được gọi là biến giải thích.

hệ số xác định:
: Độ đo mức phù hợp của một mô hình tuyến tính.

phép thử ngẫu nhiên có kiểm soát:
: Cách thiết kế thí nghiệm theo đó chủ thể được ngẫu nhiên chia thành nhóm, và các nhóm khác nhau được điều trị theo cách riêng.

điều trị:
: Sự thay đổi hoặc can thiệp áp dụng cho một nhóm trong phép thử có kiểm soát.

nhóm kiểm soát:
: Nhóm trong phép thử có kiểm soát mà không được điều trị, hoặc được điều trị theo cách mà hiệu ứng của nó đã biết trước.

thí nghiệm tự nhiên:
: Cách thiết kế thí nghiệm trong đó tận dụng ưu điểm của sự phân chia tự nhiên các chủ thể vào các nhóm theo hình thức ít nhất là gần như ngẫu nhiên.

---

1. Còn gọi là bất đẳng thức Bu-nhia-cốp-xki. Xem http://wikipedia.org/wiki/Cauchy-Schwarz_inequality. ↩
2. Xem http://wikipedia.org/wiki/Simple_linear_regression. ↩
3. Xem Press và nnk., {Numerical Recipes in C}, Chương 15 tại http://www.nrbook.com/a/bookcpdf/c15-1.pdf. ↩