BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN**



**TIỂU LUẬN**

**CƠ SỞ TOÁN CHO HỌC MÁY**

**Tên tiểu luận: Chương 4- Hàm phân bố tích lũy**

......................Giảng viên HD: **TS. Nguyễn Văn Hậu**

Học viên thực hiện: Dương Thành Đạo

Ngành: Công nghệ thông tin (H01222)

*Hưng Yên, 7/2023*

**Chương 4: Cumulative Distribution Functions ( Hàm phân bố tích lũy)**

Mã cho chương này nằm trong cumulative.py. Để biết thông tin về tải xuống và làm việc với mã này, xem “Using the Code” trên trang XI.

***The Limits of PMFs ( Sự hạn chế của các PMF)***

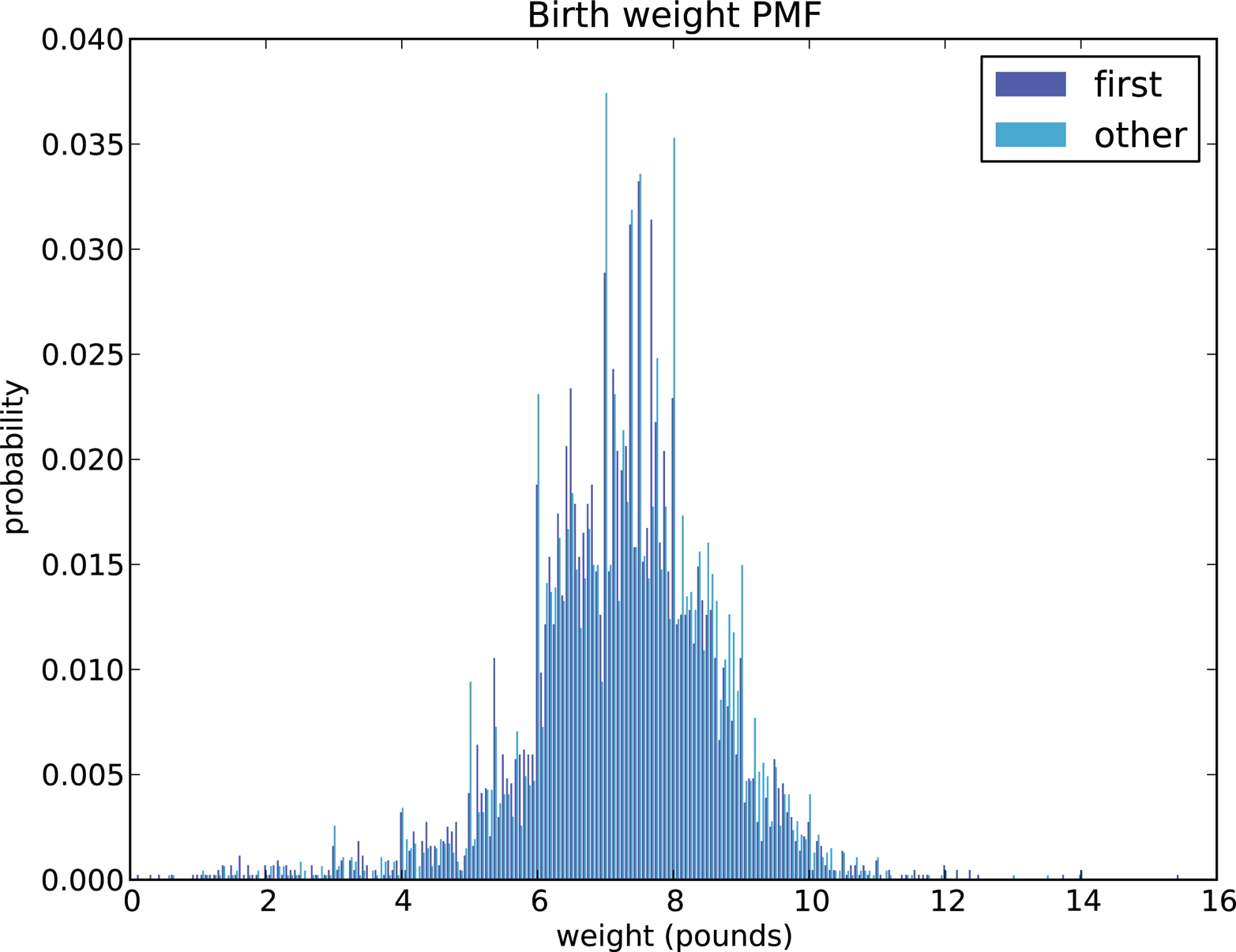
PMF hoạt động tốt nếu số lượng giá trị nhỏ. Nhưng khi số lượng giá trị tăng lên, xác suất liên quan đến từng giá trị trở nên nhỏ hơn và ảnh hưởng của nhiễu ngẫu nhiên tăng.

Ví dụ, chúng ta có thể quan tâm đến sự phân bổ cân nặng khi sinh. trong NSFG dữ liệu, biến totalwgt\_lb ghi lại cân nặng khi sinh tính bằng pound. Hình 4-1 cho thấy PMF của các giá trị này cho trẻ sơ sinh đầu tiên và những trẻ khác.

Nhìn chung, các bản phân phối này giống hình chuông của một bản phân phối bình thường, với nhiều các giá trị gần giá trị trung bình và một vài giá trị cao hơn và thấp hơn nhiều.

Nhưng các phần của con số này rất khó giải thích. Có nhiều gai và thung lũng, và một số khác biệt rõ ràng giữa các bản phân phối. Thật khó để nói cái nào trong số này các tính năng có ý nghĩa. Ngoài ra, rất khó để nhìn thấy các mẫu tổng thể; ví dụ, mà phân phối bạn nghĩ có ý nghĩa cao hơn?

Những vấn đề này có thể được giảm thiểu bằng cách tạo dữ liệu; đó là, chia phạm vi của các giá trị thành các khoảng không chồng chéo và đếm số lượng giá trị trong mỗi ngăn. đóng thùng có thể hữu ích, nhưng thật khó để có được kích thước phù hợp của các thùng. Nếu chúng đủ lớn để làm dịu tiếng ồn, chúng cũng có thể làm mịn thông tin hữu ích.

Một giải pháp thay thế để tránh những vấn đề này là hàm phân phối tích lũy (CDF), đó là chủ đề của chương này. Nhưng trước khi tôi có thể giải thích về CDF, tôi phải giải thích phân vị.

Hình 4-1. PMF của cân nặng khi sinh cho thấy hạn chế của PMF: chúng khi so sánh cắt trực quan

***Percentiles ( Số phần Trăm)***

Nếu bạn đã thực hiện một bài kiểm tra tiêu chuẩn hóa, có thể bạn đã nhận được kết quả của mình ở dạng thô. Điểm số và xếp hạng phần trăm. Trong ngữ cảnh này, thứ hạng phần trăm là phần của những người đạt điểm thấp hơn bạn (hoặc bằng nhau). Vì vậy, nếu bạn “thuộc phân vị thứ 90”, thì bạn đã cũng như bằng hoặc hơn 90% số người đã tham gia kỳ thi.

Đây là cách bạn có thể tính thứ hạng phần trăm của một giá trị, your\_score, so với các giá trị trong chuỗi điểm số:

*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\_*

Ví dụ: nếu điểm trong dãy là 55, 66, 77, 88 và 99, và bạn nhận được 88, thì thứ hạng phần trăm của bạn sẽ là 100 \* 4/5 tức là 80.

Nếu bạn được cho một giá trị, bạn sẽ dễ dàng tìm được thứ hạng phần trăm của nó; đi theo con đường khác là một chút khó hơn. Nếu bạn được xếp hạng phần trăm và bạn muốn tìm giá trị tương ứng, một tùy chọn là sắp xếp các giá trị và tìm kiếm giá trị 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 phần trăm. Ví dụ: phân vị thứ 50 là giá trị với thứ hạng phần trăm 50. Trong phân phối điểm thi, phần trăm thứ 50 là 77.

Việc triển khai Percentile này không hiệu quả. Một cách tiếp cận tốt hơn là sử dụng thứ hạng phần trăm để tính chỉ số của phần trăm tương ứng:

*def Percentile2(scores, percentile\_rank):  
scores.sort()  
index = percentile\_rank \* (len(scores)-1) / 100  
return scores[index]*

Sự khác biệt giữa "phân vị" và "xếp hạng phân vị" có thể gây nhầm lẫn và mọi người không phải lúc nào cũng sử dụng các thuật ngữ một cách chính xác. Tóm lại, PercentileRank lấy một giá trị và tính thứ hạng phần trăm của nó trong một tập hợp các giá trị; Percentile có thứ hạng phần trăm và tính giá trị tương ứng.

***CDFs ( Hàm phân bố tích lũy)***

Bây giờ chúng ta đã hiểu phần trăm và xếp hạng phần trăm, chúng ta đã sẵn sàng giải quyết vấn đềhàm phân phối tích lũy (CDF). CDF là chức năng ánh xạ từ mộtgiá trị theo thứ hạng phần trăm của nó.

CDF là một hàm của x, trong đó x là bất kỳ giá trị nào có thể xuất hiện trong phân phối.

Để đánh giá CDF(x) cho một giá trị cụ thể của x, chúng ta tính phần của các giá trị trong phân phối nhỏ hơn hoặc bằng x.

Đây là những gì trông giống như một hàm nhận một chuỗi, t và một giá trị, x:

*def EvalCdf(t, x):  
count = 0.0  
for value in t:  
if value <= x:  
count += 1  
prob = count / len(t)  
return prob*

Chức năng này gần giống với PercentileRank, ngoại trừ kết quả là một thử nghiệm khả năng trong phạm vi 0–1 thay vì xếp hạng phần trăm trong phạm vi 0–100.

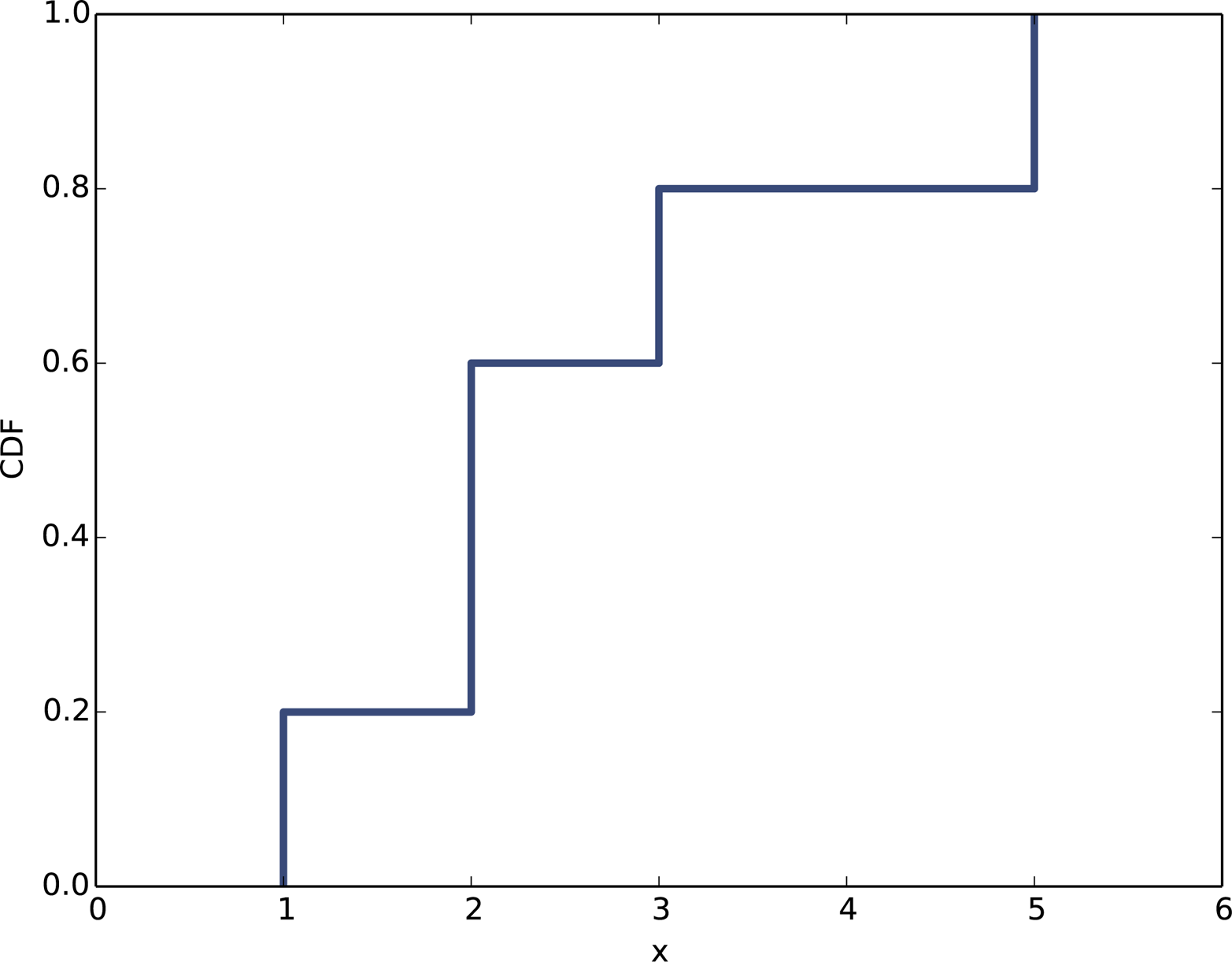
Ví dụ: giả sử chúng tôi thu thập một mẫu có giá trị [1, 2, 2, 3, 5]. Đây là một số giá trị từ CDF của nó:

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

Chúng tôi có thể đánh giá CDF cho bất kỳ giá trị nào của x, không chỉ các giá trị 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, CDF(x) là 1.

Hình 4-2 là một biểu diễn đồ họa của CDF này. CDF của một mẫu là một bước chức năng.



Hình 4-2. Ví dụ về CDF

***Representing CDFs ( Biểu diễn CDF)***

Thinkstats2 cung cấp một lớp có tên là Cdf đại diện cho các CDF. Phương pháp cơ bản-ods Cdf cung cấp là:

*Prob(x)  
 Given a value x, computes the probability p = CDF(x). The bracket operator is  
equivalent to Prob.  
Value(p)  
 Given a probability p, computes the corresponding value, x; that is, the inverse CDF of p.*

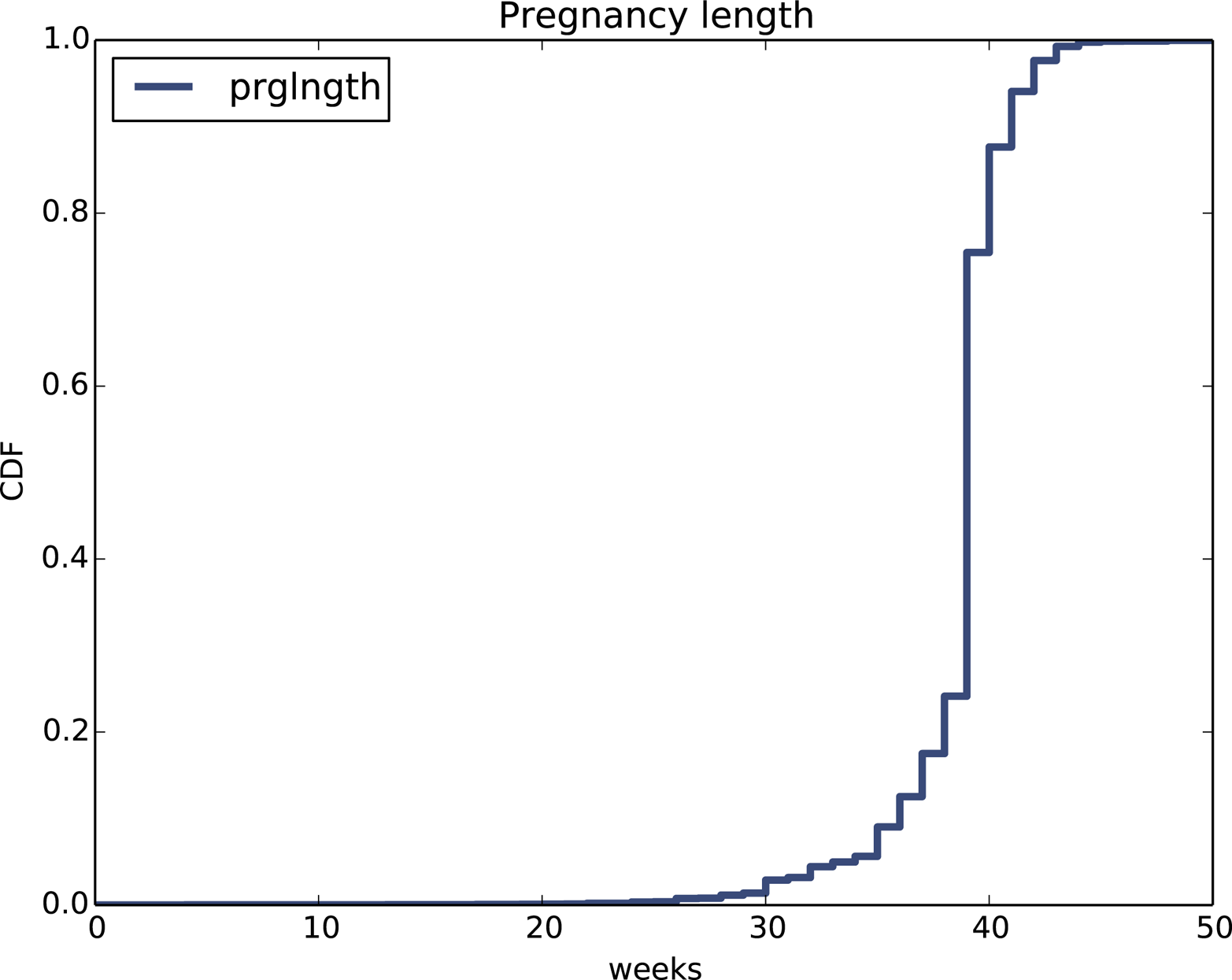
Hàm tạo Cdf có thể lấy làm đối số là danh sách giá trị, Sê-ri gấu trúc, Lịch sử, Pmf, hoặc Cdf khác. Đoạn mã sau tạo một Cdf để phân phối thai kỳ độ dài trong NSFG:

*live, firsts, others = first.MakeFrames()  
cdf = thinkstats2.Cdf(live.prglngth, label='prglngth')*

thinkplot cung cấp một hàm có tên là Cdf để vẽ các Cdf dưới dạng các dòng:

*thinkplot.Cdf(cdf)  
thinkplot.Show(xlabel='weeks', ylabel='CDF')*

Hình 4-3 cho thấy kết quả. Một cách để đọc CDF là tra cứu phần trăm. Ví dụ, có vẻ như khoảng 10% trường hợp mang thai ngắn hơn 36 tuần và khoảng 90% là ngắn hơn 41 tuần. CDF cũng cung cấp một biểu diễn trực quan về hình dạng của sự phân phối. Các giá trị phổ biến xuất hiện dưới dạng các phần dốc hoặc thẳng đứng của CDF; trong này ví dụ, chế độ ở tuần thứ 39 là rõ ràng. Có một vài giá trị dưới 30 tuần, vì vậy CDF trong phạm vi này là bằng phẳng.

 Hình 4-3. CDF của thời gian mang thai

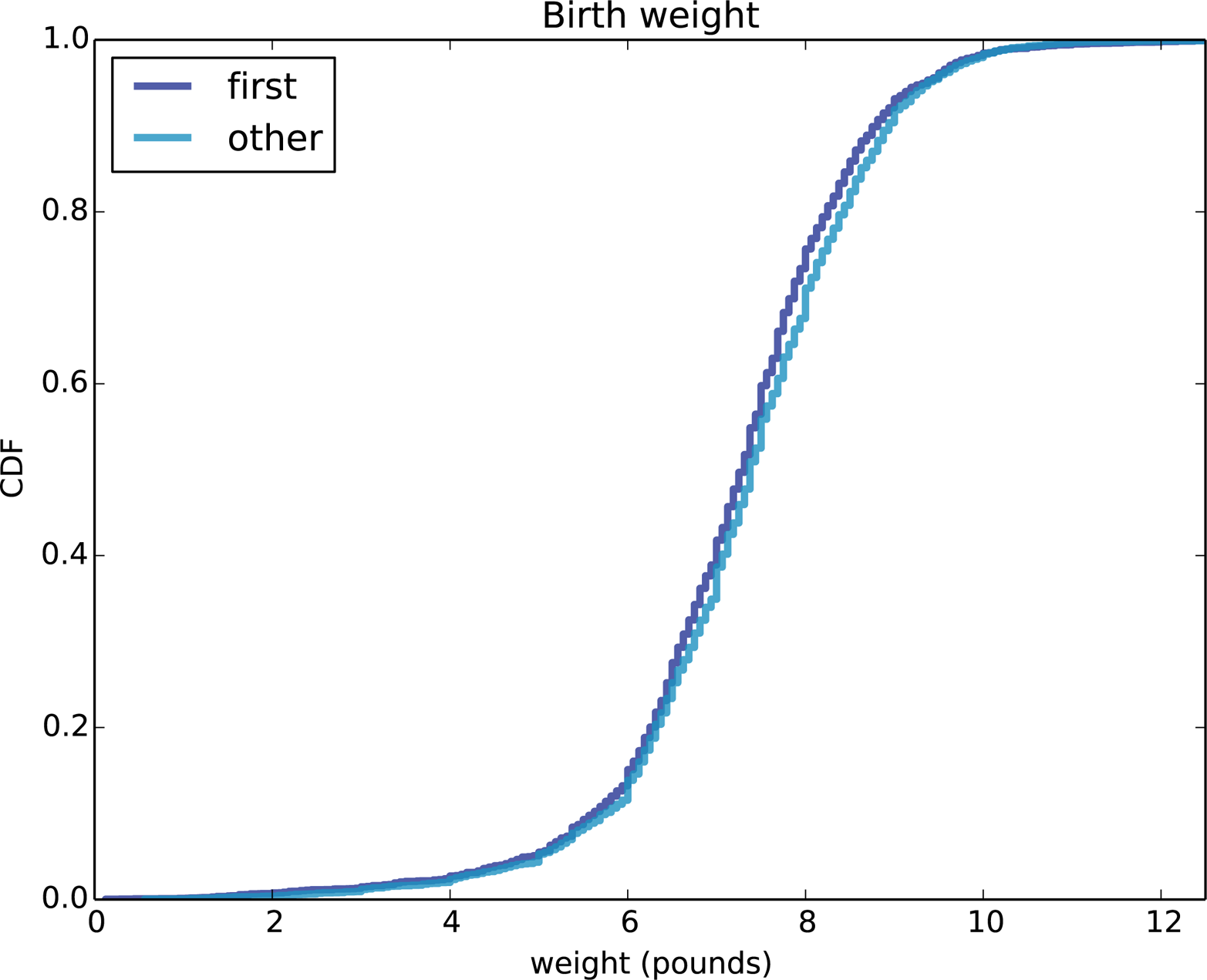
Phải mất một thời gian để làm quen với CDF, nhưng một khi bạn đã làm, tôi nghĩ bạn sẽ thấy rằng chúng hiển thị nhiều thông tin hơn, rõ ràng hơn PMF.

***Comparing CDFs ( So sánh CDF)***

CDF đặc biệt hữu ích để so sánh các bản phân phối. Ví dụ, đây là mã biểu đồ CDF về cân nặng khi sinh của trẻ sơ sinh và những trẻ khác.

*first\_cdf = thinkstats2.Cdf(firsts.totalwgt\_lb, label='first')  
other\_cdf = thinkstats2.Cdf(others.totalwgt\_lb, label='other')  
thinkplot.PrePlot(2)  
thinkplot.Cdfs([first\_cdf, other\_cdf])  
thinkplot.Show(xlabel='weight (pounds)', ylabel='CDF')*

Hình 4-4 cho thấy kết quả. So với Hình 4-1, hình này tạo nên hình dạng của phân phối, và sự khác biệt giữa chúng, rõ ràng hơn nhiều. Chúng ta có thể thấy rằng đầu tiên em bé nhẹ hơn một chút trong suốt quá trình phân phối, với sự khác biệt lớn hơn ở trên nghĩa.



Hình 4-4. CDF về cân nặng khi sinh của trẻ sơ sinh và những trẻ khác

***Percentile-Based Statistics (Thống kê dựa trên phần trăm)***

Khi bạn đã tính toán CDF, thật dễ dàng để tính toán phần trăm và xếp hạng phần trăm.

Lớp Cdf cung cấp hai phương thức sau:

*PercentileRank(x)  
Given a value x, computes its percentile rank, 100 · CDF(x).  
Percentile(p)  
Given a percentile rank rank, computes the corresponding value, x. Equivalent to  
Value(p/100).*

Phần trăm có thể được sử dụng để tính toán thống kê tóm tắt dựa trên phần trăm. Ví dụ, phân vị thứ 50 là giá trị chia đôi phân phối, còn được gọi là trung bình. Giống như giá trị trung bình, trung vị là thước đo xu hướng trung tâm của phân phối.

Trên thực tế, có một số định nghĩa về "trung vị", mỗi định nghĩa có các thuộc tính khác nhau. Nhưng Percentile(50) rất đơn giản và hiệu quả để tính toán.

Một thống kê dựa trên phần trăm khác là khoảng tứ phân vị (IQR), là thước đo của sự lan rộng của một phân phối. IQR là sự khác biệt giữa 75 và 25 phân vị.

Tổng quát hơn, phần trăm thường được sử dụng để tóm tắt hình dạng của phân phối. Vì ví dụ, phân phối thu nhập thường được báo cáo theo “ngũ phân vị”; nghĩa là, nó được chia tại phần trăm thứ 20, 40, 60 và 80. Các bản phân phối khác được chia thành 10 “dec‐iles”. Các thống kê như thế này đại diện cho các điểm cách đều nhau trong CDF được gọi là quan‐gạch.

***Random Numbers ( Số ngẫu nhiên)***

Giả sử chúng ta chọn một mẫu ngẫu nhiên từ dân số trẻ sinh sống và tra cứu thứ hạng phần trăm của trọng lượng sơ sinh của họ. Bây giờ giả sử chúng ta tính CDF của per-cấp bách phân vị. Bạn nghĩ phân phối sẽ như thế nào?

Đây là cách chúng ta có thể tính toán nó. Đầu tiên, chúng tôi tạo Cdf về cân nặng khi sinh:

*weights = live.totalwgt\_lb  
cdf = thinkstats2.Cdf(weights, label='totalwgt\_lb')*

*Sau đó, chúng tôi tạo một mẫu và tính toán xếp hạng phần trăm của từng giá trị trong mẫu.*

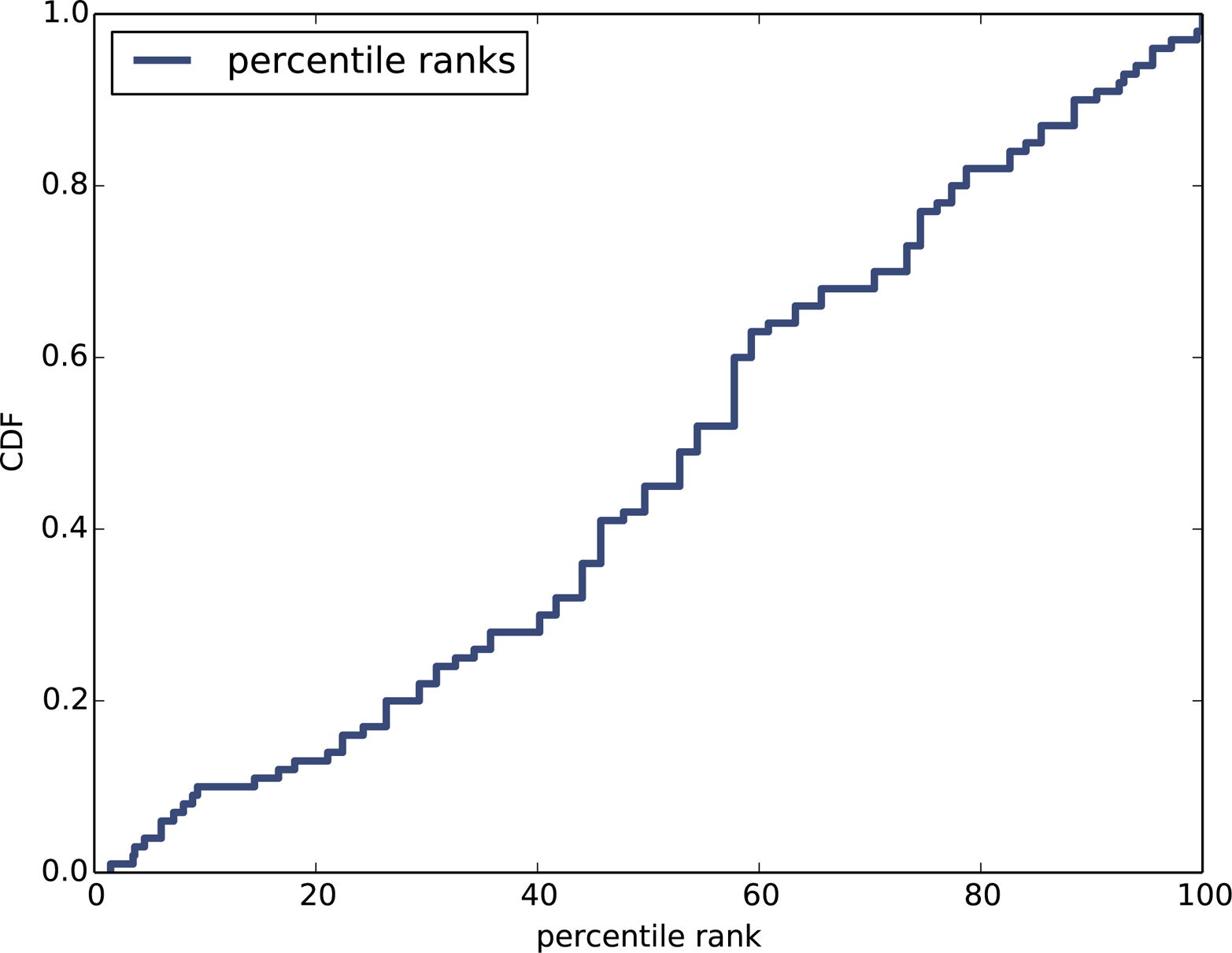
*sample = np.random.choice(weights, 100, replace=True)  
ranks = [cdf.PercentileRank(x) for x in sample]*

mẫu là mẫu ngẫu nhiên gồm 100 cân nặng sơ sinh, được chọn thay thế; đó là, các cùng một giá trị có thể được chọn nhiều lần. xếp hạng là một danh sách các xếp hạng phần trăm.

Cuối cùng, chúng tôi tạo và vẽ đồ thị Cdf của xếp hạng phần trăm.

*rank\_cdf = thinkstats2.Cdf(ranks)  
thinkplot.Cdf(rank\_cdf)  
thinkplot.Show(xlabel='percentile rank', ylabel='CDF')*

Hình 4-5 cho thấy kết quả. CDF xấp xỉ là một đường thẳng, có nghĩa là sự phân bố là thống nhất.



Hình 4-5. CDF của xếp hạng phần trăm cho một mẫu cân nặng khi sinh ngẫu nhiên

Kết quả đó có thể không rõ ràng, nhưng đó là hệ quả của cách CDF xác định. Con số này cho thấy 10% mẫu nằm dưới phân vị thứ 10, 20% nằm dưới phân vị thứ 20, v.v., chính xác như những gì chúng ta mong đợi.

Vì vậy, bất kể hình dạng của CDF như thế nào, sự phân bố của các cấp phân vị là đồng nhất.

Thuộc tính này rất hữu ích vì nó là cơ sở của một thuật toán đơn giản và hiệu quả cho tạo các số ngẫu nhiên với một CDF nhất định. Đây là cách:

• Chọn một thứ hạng phần trăm thống nhất trong phạm vi 0–100.

• Sử dụng Cdf.Percentile để tìm giá trị trong phân phối tương ứng với thứ hạng phần trăm bạn đã chọn.

Cdf cung cấp việc triển khai thuật toán này, được gọi là Ngẫu nhiên:

*# class Cdf:  
def Random(self):  
return self.Percentile(random.uniform(0, 100))*

Cdf cũng cung cấp Mẫu lấy một số nguyên n và trả về một danh sách gồm n giá trị được chọn ngẫu nhiên từ Cdf.

***Comparing Percentile Ranks ( So sánh thứ hạng phần trăm)***

Xếp hạng phần trăm rất hữu ích để so sánh các phép đo giữa các nhóm khác nhau. Vì ví dụ, những người thi đấu trong các cuộc đua chân thường được phân nhóm theo độ tuổi và giới tính. So sánh những người ở các nhóm tuổi khác nhau, bạn có thể chuyển đổi thời gian cuộc đua thành thứ hạng phần trăm.

Vài năm trước, tôi đã chạy James Joyce Ramble 10K ở Dedham MA; Tôi đã hoàn thành trong 42:44, đứng thứ 97 trong lĩnh vực năm 1633. Tôi đã đánh bại hoặc hòa 1537 vận động viên chạy trong năm 1633, vì vậy tỷ lệ phần trăm của tôi xếp hạng trong lĩnh vực này là 94%.

Tổng quát hơn, với vị trí và kích thước trường đã cho, chúng ta có thể tính thứ hạng phần trăm:

*def PositionToPercentile(position, field\_size):  
beat = field\_size - position + 1  
percentile = 100.0 \* beat / field\_size  
return percentile*

Trong nhóm tuổi của tôi, được ký hiệu là M4049 cho “nam giới từ 40 đến 49 tuổi”, tôi đã đến thứ 26 trên 256. Vì vậy, thứ hạng phần trăm của tôi trong nhóm tuổi của tôi là 90%.

Nếu tôi vẫn tiếp tục tranh cử trong 10 năm nữa (và tôi hy vọng là như vậy), thì tôi sẽ thuộc phân đội M5059.

Giả sử rằng thứ hạng phần trăm của tôi trong bộ phận của tôi là như nhau, thì nên chậm hơn bao nhiêu. Tôi mong đợi được?

Tôi có thể trả lời câu hỏi đó bằng cách chuyển đổi thứ hạng phần trăm của mình trong M4049 thành một vị trí trong M5059. Đây là mã:

*def PercentileToPosition(percentile, field\_size):  
beat = percentile \* field\_size / 100.0  
position = field\_size - beat + 1  
return position*

Có 171 người trong M5059, vì vậy tôi sẽ phải đến trong khoảng từ ngày 17 đến ngày 18 nơi có cùng thứ hạng phần trăm. Thời gian về đích của người chạy thứ 17 tại M5059 là 46:05, vì vậy đó là thời gian tôi sẽ phải đánh bại để duy trì thứ hạng phần trăm của mình.

**Exercises (Bài tập)**

Đối với các bài tập sau, bạn có thể bắt đầu với chap04ex.ipynb. giải pháp của tôi là trong chap04soln.ipynb.

Bài tập 4-1.

Bạn nặng bao nhiêu khi sinh? Nếu bạn không biết, hãy gọi cho mẹ hoặc người khác ai biết. Sử dụng dữ liệu NSFG (tất cả các ca sinh sống), tính phân bố sinh trọng số và sử dụng nó để tìm thứ hạng phần trăm của bạn. Nếu bạn là đứa con đầu lòng, hãy tìm thứ hạng centile trong phân phối cho trẻ sơ sinh đầu tiên. Nếu không thì sử dụng phân phối cho người khác. Nếu bạn thuộc phân vị thứ 90 trở lên, hãy gọi lại cho mẹ bạn và xin lỗi.

Bài tập 4-2.

Các số được tạo bởi random.random được coi là đồng nhất giữa 0 và 1; nghĩa là, mọi giá trị trong phạm vi phải có cùng xác suất.

Tạo 1000 số từ random.random và vẽ biểu đồ PMF và CDF của chúng. là dis‐ đồng phục cống nạp?

***Glossary ( Thuật ngữ)***

- Thứ hạng phần trăm

Tỷ lệ phần trăm của các giá trị trong một phân phối nhỏ hơn hoặc bằng một giá trị nhất định.

- Phân vị

Giá trị được liên kết với một thứ hạng phần trăm nhất định.

- Hàm phân phối tích lũy (CDF)

Một hàm ánh xạ từ các giá trị đến xác suất tích lũy của chúng. CDF(x) là

phần của mẫu nhỏ hơn hoặc bằng x.

- Nghịch đảo CDF

Một hàm ánh xạ từ xác suất tích lũy, p, tới giá trị tương ứng.

- Trung bình

Phân vị thứ 50, thường được sử dụng như thước đo xu hướng trung tâm.

- Phạm vi liên vùng

Sự khác biệt giữa phần trăm thứ 75 và thứ 25, được sử dụng làm thước đo mức độ lây lan.

- Lượng tử

Một chuỗi các giá trị tương ứng với các cấp phân vị cách đều nhau; cho người cũ rộng rãi, các phần tư phân phối là phần trăm thứ 25, 50 và 75.

- Thay thế

Một thuộc tính của quá trình lấy mẫu. “Với sự thay thế” có nghĩa là cùng một giá trị có thể được chọn nhiều lần; "không thay thế" có nghĩa là một khi giá trị được chọn, nó sẽ bị loại bỏ khỏi quần thể.s