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 3 – Hàm khối xác xuất**

.....

Giảng viên HD: **TS. Nguyễn Văn Hậu**

Học viên thực hiện: **Nguyễn Văn Dương**

Lớp: **H01222**

*Hưng Yên, 6/2023*

**LỜI CẢM ƠN**

Trong suốt quá trình học tập, và hoàn thành tiểu luận môn Cơ sở toán cho học máy, chúng em đã nhận được rất nhiều sự hướng dẫn tận tình quý báu của thầy cô. Với lòng biết ơn sâu sắc em xin được bày tỏ lời cảm ơn đến:

• Ban giám hiệu Trường Đại học sư phạm kỹ thuật Hưng Yên, khoa Công nghệ thông tin đã tận tình giúp đỡ và tạo điều kiện cho em trong quá trình học tập.

• Đặc biệt, chúng em xin cảm ơn thầy **Nguyễn Văn Hậu** - người đã trực tiếp giảng dạy, hướng dẫn và tạo mọi điều kiện để em thực hiện bài tiểu luận này bằng tất cả lòng nhiệt tình và sự quan tâm sâu sắc.

• Xin chân thành cảm ơn các thầy cô trong khoa Công nghệ thông tin đã tạo cơ hội cho em được học tập, nghiên cứu và tích lũy kiến thức để thực hiện bài tiểu luận.

Em đã cố gắng vận dụng những kiến thức đã học được và tìm tòi thêm nhiều thông tin để hoàn thành bài tiểu luận này. Tuy nhiên, do kiến thức còn hạn chế và chưa có nhiều kinh nghiệm trên thực tiễn nên khó tránh khỏi những thiếu sót trong bài làm. Rất kinh mong quý thầy, cô cho em thêm những góp ý để bài tiểu luận của em được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

**MỤC LỤC CHƯƠNG 3**

Pmfs .................................................................................................................................... 2

Hình dung khác.................................................................................................................... 6

Lập chỉ mục khung dữ liệu................................................................................................. 11

Bài tập............................................................................................................................... 16

Bảng chú giải................................................................................................................... 18

**CHƯƠNG 3**

**HÀM KHỐI XÁC SUẤT**

Mã code cho chương này trong tập tin probability.py. Để biết thông tin về cái tải và cách làm việc với đoạn mã này hãy xem “Using the Code” ở trang 11.

**Pmfs**

Một cách khác để biểu diễn một phân bổ là hàm khối xác xuất (PMF), ánh xạ từ mỗi giá trị đến xác xuất của nó. Một xác xuất là một tần suất được biểu diễn dưới dạng một phần nhỏ của kích thước mẫu n. Để chuyển từ tần số sang xác xuất, chúng ta chia cho n, được gọi là chuẩn hóa.

Đưa ra một Hist, chúng ta có thể tạo một dictionary ánh xạ từ từng giá trị đến xác suất của nó:

n = hist.Total()

d = {}

for x, freq in hist.Items():

    d[x] = freq / n

Hoặc chúng ta có thể sử dụng lớp Pmf do thinkstats2 cung cấp. Giống như Hist, hàm tạo Pmf có thể lấy một danh sách, pandas series, dictionary, Hist hoặc một đối tượng Pmf khác. Đây là một ví dụ với một danh sách đơn giản:

import thinkstats2

pmf = thinkstats2.Pmf([1, 2, 2, 3, 5])

pmf Pmf({1: 0.2, 2: 0.4, 3: 0.2, 5: 0.2})

Pmf được chuẩn hóa nên tổng xác suất là 1.

Các đối tượng Pmf và Hist giống nhau theo nhiều cách; trên thực tế, chúng kế thừa nhiều phương thức của chúng từ một lớp cha chung. Ví dụ: các phương thức Giá trị và Mục hoạt động theo cùng một cách cho cả hai. Sự khác biệt lớn nhất là Hist ánh xạ từ các giá trị sang bộ đếm số nguyên; một ánh xạ Pmf từ các giá trị sang xác suất dấu phẩy động.

Để tra cứu xác suất liên quan đến một giá trị, hãy sử dụng Prob:

pmf.Prob(2)

0.4

Hoặc có thể sử dụng toán tử dấu ngoặc cũng tương đương:

pmf[2]

0.4

Bạn có thể sửa đổi một Pmf hiện có bằng cách tăng xác suất được liên kết với một giá trị:

pmf.Incr(2, 0.2)

pmf.Prob(2)

0.6

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

pmf.Mult(2, 0.5)

pmf.Prob(2)

0.3

Nếu bạn sửa đổi Pmf, kết quả có thể không được chuẩn hóa; nghĩa là, các xác suất có thể không còn bằng 1 nữa. Để kiểm tra, bạn có thể gọi Total, hàm trả về tổng của các xác suất:

pmf.Total()

0.9

Để chuẩn hóa lại, hãy gọi Normalize():

pmf.Normalize()

pmf.Total()

1.0

Các đối tượng Pmf cung cấp một phương thức Sao chép để bạn có thể tạo và sửa đổi một bản sao mà không ảnh hưởng đến bản gốc.

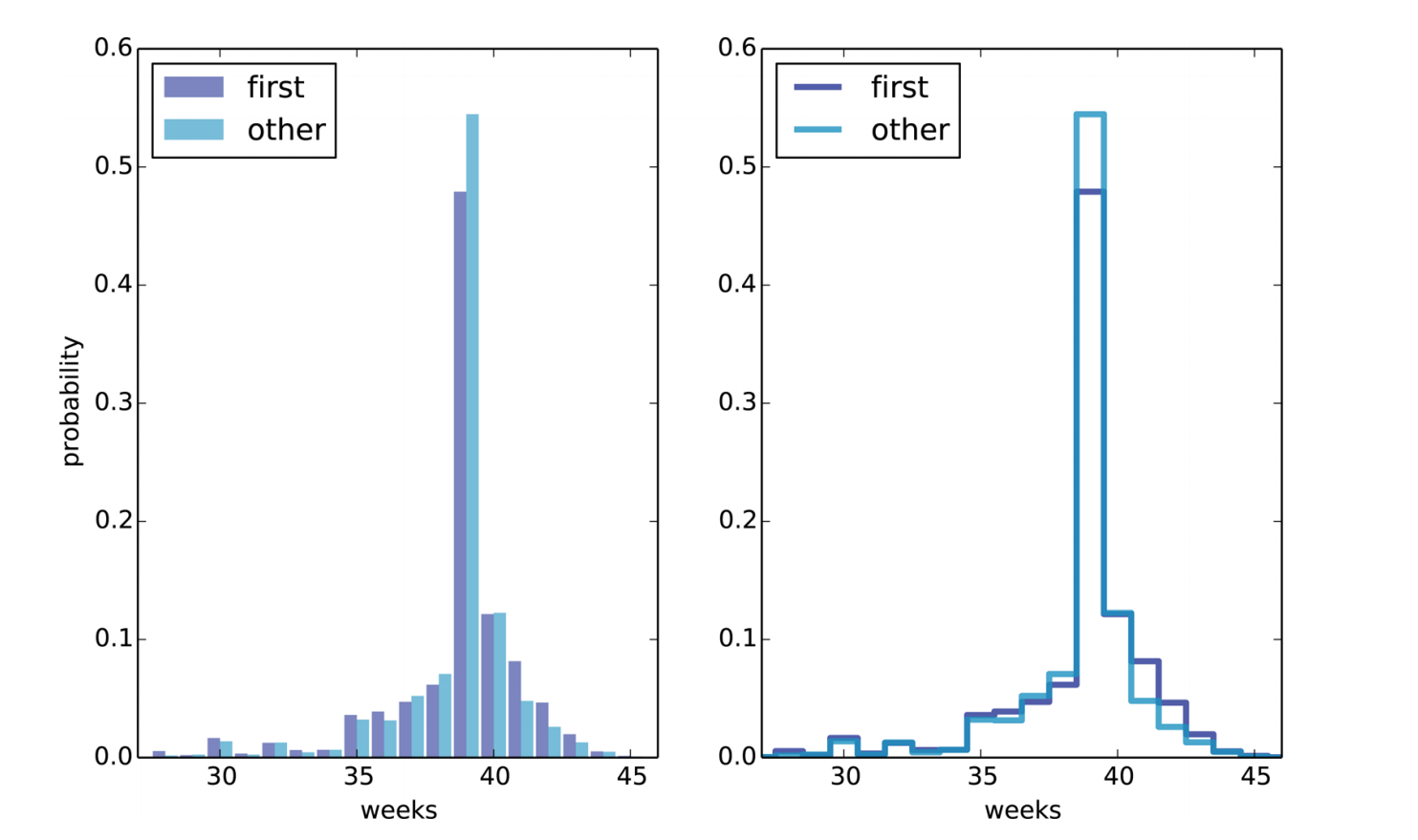
Ký hiệu của tôi trong phần này có vẻ không nhất quán, nhưng có một hệ thống: Tôi sử dụng Pmf cho tên của lớp, pmf cho một thể hiện của lớp và PMF cho khái niệm toán học của hàm khối lượng xác suất.

Vẽ biểu đồ PMFs thinkplot cung cấp hai cách để vẽ biểu đồ Pmfs:

• Để vẽ biểu đồ Pmf dưới dạng biểu đồ thanh, bạn có thể sử dụng thinkplot.Hist. Biểu đồ thanh hữu ích nhất nếu số lượng giá trị trong Pmf nhỏ.

• Để vẽ biểu đồ Pmf dưới dạng hàm bước, bạn có thể sử dụng thinkplot.Pmf. Tùy chọn này hữu ích nhất nếu có một số lượng lớn các giá trị và Pmf trơn tru. Chức năng này cũng hoạt động với các đối tượng Hist.

Ngoài ra, pyplot cung cấp một hàm gọi là hist, hàm này nhận một chuỗi các giá trị, tính toán một biểu đồ tần suất và vẽ đồ thị của nó. Vì tôi sử dụng các đối tượng Hist nên tôi thường không sử dụng pyplot.hist. Hình 3-1 cho thấy các PMF về thời gian mang thai của trẻ sơ sinh đầu lòng và những trẻ khác sử dụng biểu đồ thanh (trái) và hàm bước (phải).



*Hình 3-1. PMF về thời gian mang thai của trẻ sơ sinh đầu lòng và những trẻ khác, sử dụng biểu đồ thanh và hàm bước.*

Đây là mã tạo Hình 3-1:

thinkplot.PrePlot(2, cols=2)

thinkplot.Hist(first\_pmf, align='right', width=width)

thinkplot.Hist(other\_pmf, align='left', width=width)

thinkplot.Config(xlabel='weeks', ylabel='probability', axis=[27, 46, 0, 0.6])

thinkplot.PrePlot(2) thinkplot.SubPlot(2)

thinkplot.Pmfs([first\_pmf, other\_pmf])

thinkplot.Show(xlabel='weeks', axis=[27, 46, 0, 0.6])

Bằng cách vẽ biểu đồ PMF thay vì biểu đồ, chúng ta có thể so sánh hai bản phân phối mà không bị nhầm lẫn bởi sự khác biệt về kích thước mẫu. Dựa trên con số này, những đứa trẻ đầu tiên Vẽ sơ đồ PMF | 29 dường như ít đến đúng giờ hơn những người khác (tuần 39) và có nhiều khả năng đến muộn hơn (tuần 41 và 42).

PrePlot lấy tham số tùy chọn các hàng và cột để tạo một lưới các hình, trong trường hợp này là một hàng gồm hai hình. Hình bên trái hiển thị các Pmf sử dụng thinkplot.Hist, như chúng ta đã thấy trước đây.

Cuộc gọi thứ hai tới PrePlot đặt lại bộ tạo màu. Sau đó, SubPlot chuyển sang hình thứ hai (ở bên phải) và hiển thị Pmfs bằng cách sử dụng thinkplot.Pmfs. Tôi đã sử dụng tùy chọn trục để đảm bảo rằng hai hình nằm trên cùng một trục, đây thường là một ý tưởng hay nếu bạn định so sánh hai hình.

**Hình dung khác**

Biểu đồ và PMF rất hữu ích khi bạn đang khám phá dữ liệu và cố gắng xác định các mẫu và mối quan hệ. Khi bạn có ý tưởng về những gì đang diễn ra, bước tiếp theo tốt là thiết kế một hình ảnh trực quan hóa làm cho các mẫu bạn đã xác định rõ ràng nhất có thể.

Trong dữ liệu NSFG, sự khác biệt lớn nhất trong các bản phân phối là gần chế độ. Vì vậy, thật hợp lý khi phóng to phần đó của biểu đồ và chuyển đổi dữ liệu để nhấn mạnh sự khác biệt:

weeks = range(35, 46)

diffs = [] for week in weeks:

p1 = first\_pmf.Prob(week)

p2= other\_pmf.Prob(week)

diff = 100 \* (p1 - p2)

diffs.append(diff)

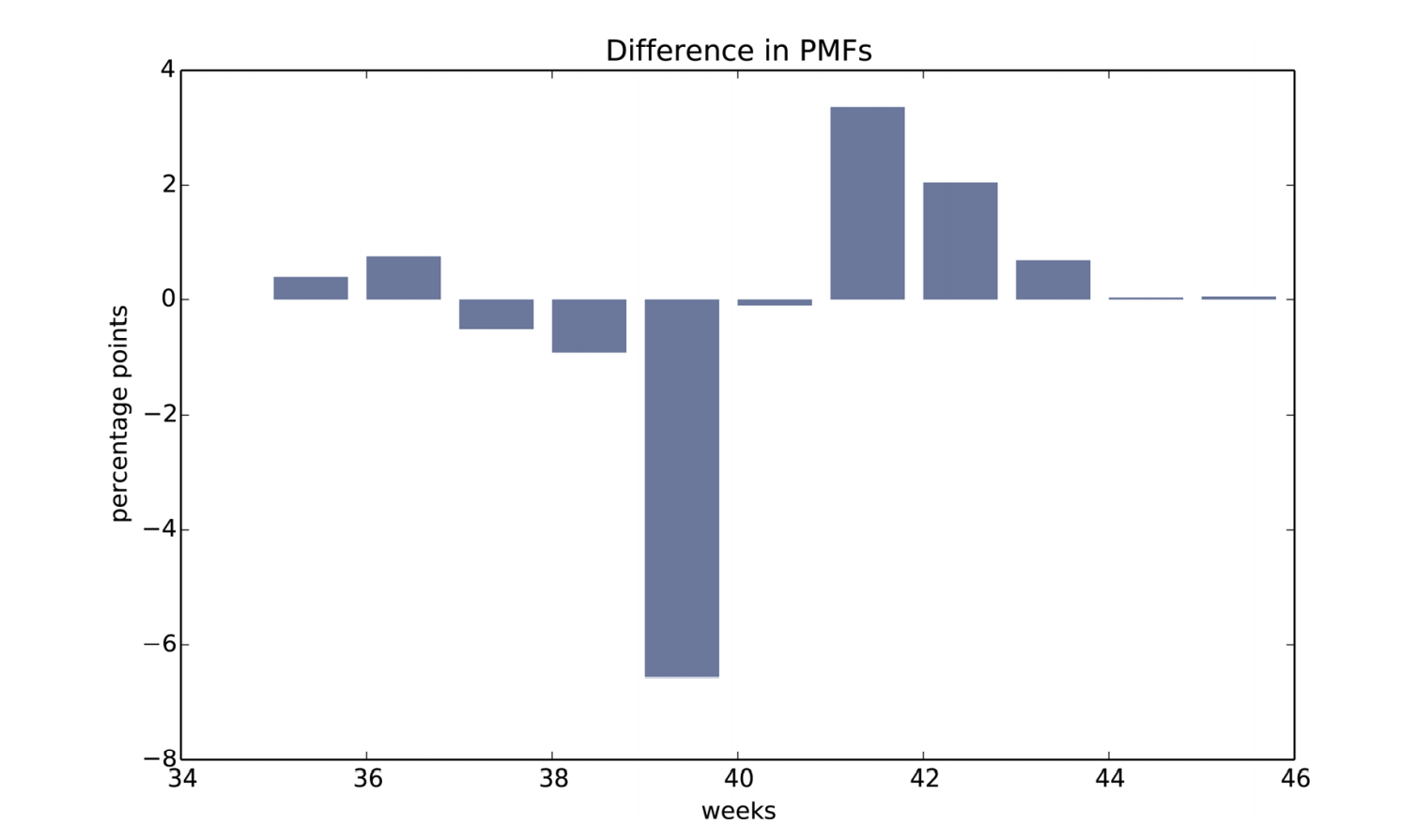
thinkplot.Bar(weeks, diffs)

Trong mã này, weeks là phạm vi tuần; diffs là sự khác biệt giữa hai PMF tính theo điểm phần trăm. Hình 3-2 hiển thị kết quả dưới dạng biểu đồ thanh. Con số này làm cho mô hình rõ ràng hơn: những đứa trẻ đầu tiên ít có khả năng được sinh ra vào tuần 39 và có nhiều khả năng được sinh ra vào tuần 41 và 42.

Hiện tại chúng ta chỉ nên giữ kết luận này một cách tạm thời. Chúng tôi đã sử dụng cùng một bộ dữ liệu để xác định sự khác biệt rõ ràng và sau đó chọn một hình ảnh trực quan hóa làm cho sự khác biệt trở nên rõ ràng. Chúng tôi không thể chắc chắn hiệu ứng này là có thật; nó có thể là do sự thay đổi ngẫu nhiên. Chúng tôi sẽ giải quyết mối quan tâm này sau.

**Nghịch lý quy mô lớp học**

Trước khi chúng ta tiếp tục, tôi muốn chứng minh một kiểu tính toán mà bạn có thể thực hiện với các đối tượng Pmf; Tôi gọi ví dụ này là “nghịch lý quy mô lớp học”.



*Hình 3-2. Sự khác biệt, tính bằng điểm phần trăm, theo tuần.*

Tại nhiều trường cao đẳng và đại học Mỹ, tỷ lệ sinh viên-giảng viên là khoảng 10:1. Nhưng sinh viên thường ngạc nhiên khi phát hiện ra rằng quy mô lớp học trung bình của họ lớn hơn 10. Có hai lý do cho sự khác biệt:

• Học sinh thường học 4–5 lớp mỗi học kỳ, nhưng các giáo sư thường dạy 1 hoặc 2 lớp.

• Số học sinh thích lớp nhỏ thì ít, nhưng số học sinh lớp đông thì (ahem!) lớn.

Hiệu ứng đầu tiên là rõ ràng, ít nhất là một khi nó được chỉ ra; thứ hai là tinh tế hơn. Hãy xem một ví dụ. Giả sử rằng một trường đại học cung cấp 65 lớp học trong một học kỳ nhất định, với sự phân bổ sĩ số như sau:

size        count

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 sĩ số trung bình của lớp học, ông ấy sẽ xây dựng PMF, tính toán giá trị trung bình và báo cáo rằng sĩ số trung bình của lớp học là 23,7.

Đây là mã:

d = { 7: 8, 12: 8, 17: 14, 22: 4, 27: 6, 32: 12, 37: 8, 42: 3, 47: 2 }

pmf = thinkstats2.Pmf(d, label='actual')

print('mean', pmf.Mean())

Nhưng nếu bạn khảo sát một nhóm sinh viên, hỏi họ có bao nhiêu sinh viên trong lớp học của họ và tính giá trị trung bình, bạn sẽ nghĩ rằng sĩ số trung bình của lớp học lớn hơn. Hãy xem lớn hơn bao nhiêu.

Đầu tiên, tôi tính toán phân phối theo quan sát của sinh viên, trong đó xác suất liên quan đến quy mô mỗi lớp học bị “sai lệch” bởi số lượng sinh viên trong lớp.

def BiasPmf(pmf, label):

new\_pmf = pmf.Copy(label=label)

for x, p in pmf.Items():

    new\_pmf.Mult(x, x)

    new\_pmf.Normalize()

return new\_pmf

Đối với mỗi quy mô lớp học, x, chúng tôi nhân xác suất với x, số học sinh quan sát quy mô lớp học đó. Kết quả là một Pmf mới đại diện cho phân phối sai lệch.

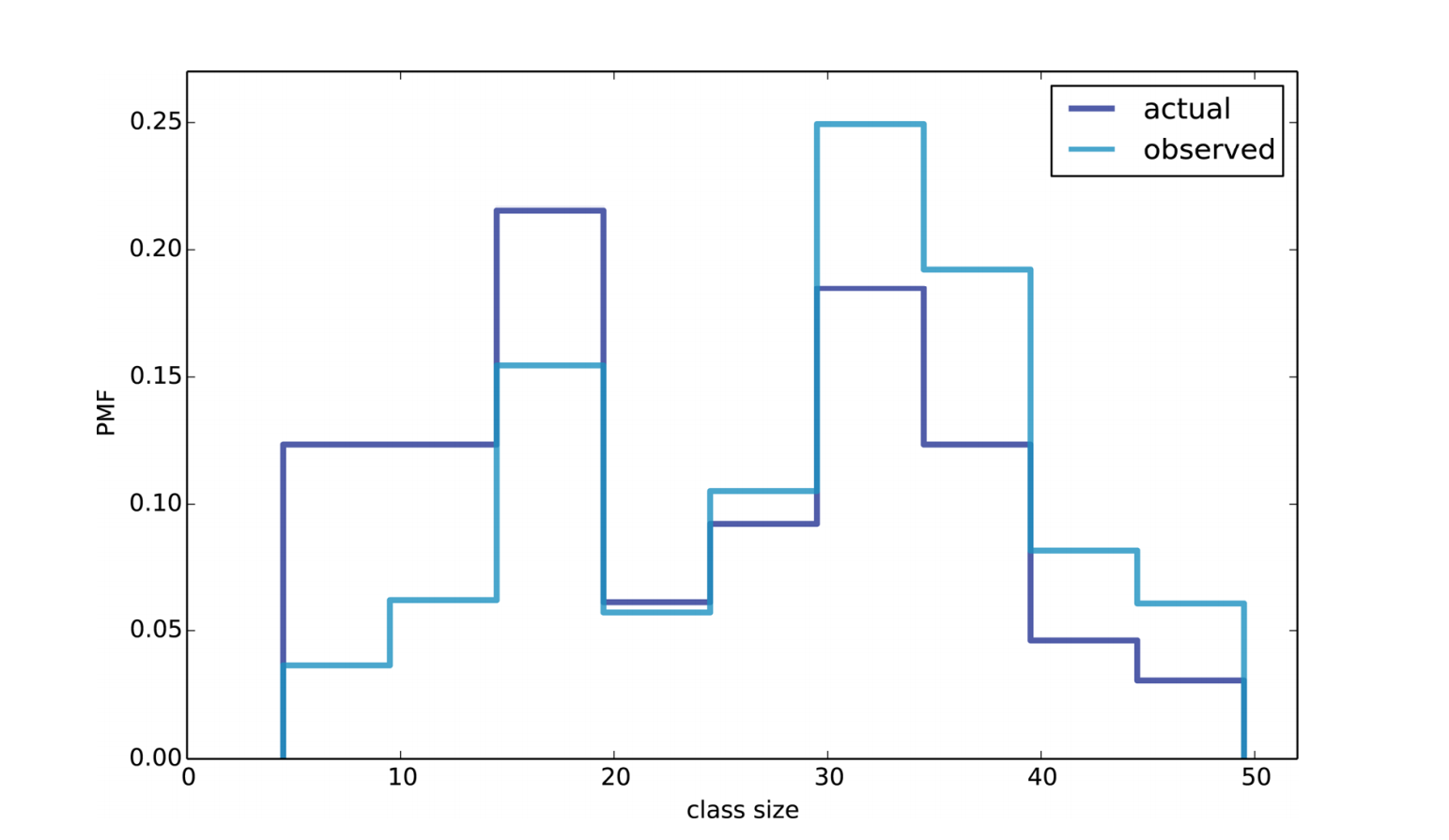
Bây giờ chúng ta có thể vẽ biểu đồ phân phối thực tế và được quan sát:

biased\_pmf = BiasPmf(pmf, label='observed')

thinkplot.PrePlot(2) thinkplot.Pmfs([pmf, biased\_pmf])

thinkplot.Show(xlabel='class size', ylabel='PMF')

Hình 3-3 cho thấy kết quả. Trong phân phối thiên vị, có ít lớp nhỏ hơn và nhiều lớp lớn hơn. Giá trị trung bình của phân phối sai lệch là 29,1, cao hơn gần 25% so với giá trị trung bình thực tế.



*Hình 3-3. Phân bố sĩ số lớp học, thực tế và theo quan sát của học sinh.*

Cũng có thể đảo ngược hoạt động này. Giả sử bạn muốn tìm sự phân bố quy mô lớp học tại một trường đại học, nhưng bạn không thể lấy dữ liệu đáng tin cậy từ Trưởng khoa. Một cách khác là chọn một mẫu học sinh ngẫu nhiên và hỏi có bao nhiêu học sinh trong lớp của họ.

Kết quả sẽ bị sai lệch vì những lý do mà chúng ta vừa thấy, nhưng bạn có thể sử dụng nó để ước tính phân phối thực tế. Đây là chức năng không thiên vị một Pmf:

def UnbiasPmf(pmf, label):

new\_pmf = pmf.Copy(label=label)

for x, p in pmf.Items():

    new\_pmf.Mult(x, 1.0/x)

    new\_pmf.Normalize()

return new\_pmf

Nó tương tự như BiasPmf; sự khác biệt duy nhất là nó chia từng xác suất cho x thay vì nhân.

**Lập chỉ mục khung dữ liệu**

Trong “DataFrames” ở trang 4, chúng tôi đọc một DataFrame của pandas và sử dụng nó để chọn và sửa đổi các cột dữ liệu. Bây giờ hãy xem xét lựa chọn hàng. Để bắt đầu, tôi tạo một mảng NumPy gồm các số ngẫu nhiên và sử dụng nó để khởi tạo DataFrame:

import numpy as np

import pandas >>>

array = np.random.randn(4, 2)

df = pandas.DataFrame(array)

df

Kết quả:

    0               1

0   -0.143510       0.616050

1   -1.489647       0.300774

2   -0.074350       0.039621

3   -1.369968       0.545897

Theo mặc định, các hàng và cột được đánh số bắt đầu từ 0, nhưng bạn có thể cung cấp tên cột:

columns = ['A', 'B']

df = pandas.DataFrame(array, columns=columns)

df

Kết quả :

    A           B

0   -0.143510   0.616050

1   -1.489647   0.300774

2   -0.074350   0.039621

3   -1.369968   0.545897

Bạn cũng có thể cung cấp tên hàng. Tập hợp các tên hàng được gọi là chỉ số; bản thân các tên hàng được gọi là nhãn.

index = ['a', 'b', 'c', 'd']

df = pandas.DataFrame(array, columns=columns, index=index)

df

Kết quả :

    A           B

a   -0.143510   0.616050

b   -1.489647   0.300774

c   -0.074350   0.039621

d   -1.369968   0.545897

Như chúng ta đã thấy trong chương trước, lập chỉ mục đơn giản sẽ chọn một cột, trả về một Chuỗi:

df['A']

Kết quả:

a -0.143510

b -1.489647

c -0.074350

d -1.369968

Name: A, dtype: float64

Để chọn một hàng theo nhãn, bạn có thể sử dụng thuộc tính loc, thuộc tính này trả về một Sê-ri:

df.loc['a']

Kết quả :

A -0.14351

B 0.61605

Name: a, dtype: float64

Nếu bạn biết vị trí số nguyên của một hàng, chứ không phải nhãn của nó, bạn có thể sử dụng thuộc tính iloc, thuộc tính này cũng trả về một Sêri.

df.iloc[0]

Kết quả :

A -0.14351

B 0.61605

Name: a, dtype: float64

loc cũng có thể lấy một danh sách các nhãn; trong trường hợp đó, kết quả là một DataFrame.

indices = ['a', 'c']

df.loc[indices]

Kết quả :

    A           B

a   -0.14351    0.616050

c   -0.07435    0.039621

Cuối cùng, bạn có thể sử dụng một lát cắt để chọn một loạt các hàng theo nhãn:

df['a':'c']

Kết quả :

    A           B

a   -0.143510   0.616050

b   -1.489647   0.300774

c   -0.074350   0.039621

Hoặc theo vị trí số nguyên:

df[0:2]

Kế quả:

    A           B

a   -0.143510   0.616050

b   -1.489647   0.300774

Kết quả trong cả hai trường hợp là một DataFrame, nhưng lưu ý rằng kết quả đầu tiên bao gồm phần cuối của lát cắt; cái thứ hai thì không. Lời khuyên của tôi: nếu các hàng của bạn có nhãn không phải là số nguyên đơn giản, hãy sử dụng nhãn một cách nhất quán và tránh sử dụng vị trí số nguyên.

**Bài tập**

Lời giải cho các bài tập này có trong chap03soln.ipynb và chap03soln.py

Bài tập 3-1.

Điều gì đó giống như nghịch lý về quy mô lớp học sẽ xuất hiện nếu bạn khảo sát trẻ em và hỏi gia đình chúng có bao nhiêu trẻ em. Các gia đình có nhiều con có nhiều khả năng xuất hiện trong mẫu của bạn hơn và các gia đình không có con sẽ không có cơ hội xuất hiện trong mẫu.

Sử dụng biến số người trả lời NSFG NUMKDHH để xây dựng phân phối thực tế cho số trẻ em dưới 18 tuổi trong hộ gia đình.

Bây giờ hãy tính phân phối sai lệch mà chúng ta sẽ thấy nếu chúng ta khảo sát trẻ em và hỏi chúng có bao nhiêu trẻ em dưới 18 tuổi (bao gồm cả chúng) trong gia đình chúng.

Vẽ sơ đồ phân phối thực tế và sai lệch, và tính toán phương tiện của chúng. Để bắt đầu, bạn có thể sử dụng chap03ex.ipynb.

Bài tập 3-2.

Trong “Tóm tắt phân phối” trên trang 22, chúng tôi đã tính giá trị trung bình của một mẫu bằng cách cộng các phần tử và chia cho n. Nếu bạn được cung cấp PMF, bạn vẫn có thể tính giá trị trung bình, nhưng quá trình này hơi khác một chút:

x¯ = ∑ i pi xi

trong đó xi là các giá trị duy nhất trong PMF và pi = PMF (xi). Tương tự, bạn có thể tính phương sai như sau:

S 2 = ∑ i pi (xi - x¯ ) 2

Viết các hàm có tên là PmfMean và PmfVar lấy một đối tượng Pmf và tính giá trị trung bình và phương sai. Để kiểm tra các phương thức này, hãy kiểm tra xem chúng có phù hợp với các phương thức Mean và Var do Pmf cung cấp hay không.

Bài tập 3-3.

Tôi bắt đầu với câu hỏi, “Có phải những đứa trẻ đầu lòng thường sinh muộn hơn không?” Để giải quyết vấn đề này, tôi đã tính toán sự khác biệt về phương tiện giữa các nhóm trẻ sơ sinh, nhưng tôi đã bỏ qua khả năng có thể có sự khác biệt giữa những đứa trẻ đầu lòng và những đứa trẻ khác đối với cùng một phụ nữ.

Để giải quyết phiên bản câu hỏi này, hãy chọn những người trả lời có ít nhất hai con và tính toán sự khác biệt theo cặp. Công thức này của câu hỏi có mang lại một kết quả khác không?

Gợi ý: sử dụng nsfg.MakePregMap.

Bài tập 3-4.

Trong hầu hết các cuộc đua chân, mọi người bắt đầu cùng một lúc. Nếu bạn là một người chạy nhanh, bạn thường vượt qua rất nhiều người khi bắt đầu cuộc đua, nhưng sau một vài dặm, mọi người xung quanh bạn đều chạy với tốc độ như nhau.

Khi tôi chạy một cuộc chạy tiếp sức đường dài (209 dặm) lần đầu tiên, tôi nhận thấy một hiện tượng kỳ lạ: khi tôi vượt qua một vận động viên khác, tôi thường nhanh hơn nhiều, và khi một vận động viên khác vượt qua tôi, anh ta thường nhanh hơn nhiều.

Hàm khối lượng xác suất Lúc đầu, tôi nghĩ rằng sự phân bố tốc độ có thể là hai hàm; nghĩa là có nhiều người chạy chậm và nhiều người chạy nhanh, nhưng ít người chạy bằng tốc độ của tôi.

Sau đó, tôi nhận ra rằng mình là nạn nhân của một thành kiến tương tự như tác động của quy mô lớp học. Cuộc đua không bình thường theo hai cách: nó sử dụng xuất phát so le, vì vậy các đội xuất phát vào những thời điểm khác nhau; Ngoài ra, nhiều đội bao gồm các vận động viên chạy ở các cấp độ khả năng khác nhau.

Kết quả là, các vận động viên được dàn trải dọc theo đường chạy với rất ít mối quan hệ giữa tốc độ và vị trí. Khi tôi tham gia cuộc đua, những người chạy gần tôi (khá nhiều) là mẫu ngẫu nhiên của những người chạy trong cuộc đua.

Vậy sự thiên vị đến từ đâu? Trong thời gian tôi tham gia cuộc đua, khả năng vượt qua người chạy hoặc bị vượt qua tỷ lệ thuận với sự khác biệt về tốc độ của chúng tôi. Tôi có nhiều khả năng bị bắt bởi một người chạy chậm, và nhiều khả năng bị bắt bởi một người chạy nhanh. Nhưng những người chạy cùng tốc độ khó có thể nhìn thấy nhau.

Viết một hàm có tên là ObservedPmf lấy một Pmf biểu thị sự phân bố tốc độ thực tế của người chạy và tốc độ của một người quan sát đang chạy, đồng thời trả về một Pmf mới biểu thị sự phân bố tốc độ của người chạy mà người quan sát nhìn thấy.

Để kiểm tra chức năng của mình, bạn có thể sử dụng relay.py, đọc kết quả từ James Joyce Ramble 10K ở Dedham MA và chuyển đổi tốc độ của mỗi người chạy thành mph.

Tính toán sự phân bố tốc độ mà bạn sẽ quan sát được nếu bạn chạy một cuộc chạy tiếp sức với vận tốc 17,5 km/h với nhóm vận động viên này. Giải pháp cho bài tập này có trong relay\_soln.py.

**Bảng chú giải**

*Probability mass function (PMF)*

một biểu diễn của một phân phối dưới dạng một hàm ánh xạ từ các giá trị đến xác suất.

*probability*

Một tần suất được biểu thị dưới dạng một phần nhỏ của kích thước mẫu.

*normalization*

Quá trình chia tần số cho một cỡ mẫu để có xác suất.

*index*

Trong DataFrame của pandas, chỉ mục là một cột đặc biệt chứa các nhãn hàng.