Chương 9

THỬ NGHIỆM GIẢ THUYẾT

Mã cho chương này nằm trong hypothese.py. Để biết thông tin về cách tải xuống và làm việc với mã này, hãy xem "Using the Code" trên trang xi.

Thử nghiệm giả thuyết cổ điển

Khi khám phá dữ liệu từ NSFG, chúng ta đã thấy một số "tác động rõ ràng", bao gồm hiệu số giữa trẻ sơ sinh đầu lòng và những trẻ khác. Cho đến nay, chúng ta đã coi những tác động này theo mệnh giá; trong chương này, chúng ta đưa chúng vào thử nghiệm.

Câu hỏi cơ bản mà chúng ta muốn giải quyết là liệu những tác động mà chúng ta thấy trong một mẫu có khả năng xuất hiện trong dân số lớn hơn hay không. Ví dụ, trong mẫu NSFG, chúng ta thấy hiệu số về thời gian mang thai trung bình của trẻ đầu lòng và những trẻ khác. Chúng ta muốn biết liệu hiệu ứng đó có phản ánh hiệu số thực sự đối với phụ nữ ở Hoa Kỳ hay liệu nó có thể xuất hiện một cách tình cờ trong mẫu hay không.

Có một số cách chúng ta có thể hình thành câu hỏi này, bao gồm thử nghiệm giả thuyết vô hiệu Fisher, lý thuyết quyết định Neyman-Pearson và suy luận Bayes. Những gì tôi trình bày ở đây là một tập hợp con của cả ba thứ tạo nên hầu hết những gì mọi người sử dụng trong thực tế, mà tôi sẽ gọi là thử nghiệm giả thuyết cổ điển.

Mục tiêu của thử nghiệm giả thuyết cổ điển là trả lời câu hỏi, "Cho một mẫu và một hiệu ứng rõ ràng, xác suất tình cờ nhìn thấy một hiệu ứng như vậy là bao nhiêu?". Đây là cách chúng ta trả lời câu hỏi đó:

• Bước đầu tiên là định lượng quy mô của hiệu ứng rõ ràng bằng cách chọn một thống kê kiểm tra. Trong ví dụ NSFG, tác động rõ ràng là hiệu số về thời gian mang thai giữa trẻ đầu lòng và những trẻ khác, do đó, lựa chọn tự nhiên cho thống kê kiểm tra là hiệu số về giá trị trung bình giữa hai nhóm.

• Bước thứ hai là xác định một giả thuyết không, là một mô hình của hệ thống dựa trên giả định rằng hiệu ứng rõ ràng là không có thật. Trong ví dụ về NSFG, giả thuyết vô hiệu là không có hiệu số giữa những đứa trẻ đầu lòng và những đứa trẻ khác; nghĩa là thời gian mang thai của cả hai nhóm có cùng phân phối.

• Bước thứ ba là tính toán giá trị p, là xác suất nhìn thấy tác động rõ ràng nếu giả thuyết không đúng. Trong ví dụ NSFG, chúng ta sẽ tính toán hiệu số thực tế về phương tiện, sau đó tính xác suất nhận thấy hiệu số là lớn hoặc lớn hơn, theo giả thuyết không.

• Bước cuối cùng là diễn giải kết quả. Nếu giá trị p thấp, hiệu ứng được cho là có ý nghĩa thống kê, có nghĩa là nó không có khả năng xảy ra một cách tình cờ. Trong trường hợp đó, chúng ta suy luận rằng hiệu ứng có nhiều khả năng xuất hiện trong dân số lớn hơn.

Logic của quá trình này tương tự như chứng minh bằng mâu thuẫn. Để chứng minh một mệnh đề toán học, A, bạn tạm thời giả sử rằng A sai. Nếu giả định đó dẫn đến mâu thuẫn, bạn kết luận rằng A thực sự phải đúng.

Tương tự như vậy, để kiểm tra một giả thuyết như, "Hiệu ứng này là có thật", chúng ta tạm thời giả định rằng nó không có thật. Đó là giả thuyết vô hiệu. Dựa trên giả định đó,

chúng ta tính toán xác suất của hiệu ứng rõ ràng. Đó là giá trị p. Nếu giá trị p thấp, chúng ta kết luận rằng giả thuyết không có khả năng đúng.

Kiểm tra giả thuyết

thinkstats2 cung cấp HypothesisTest, một lớp đại diện cho cấu trúc của một bài kiểm tra giả thuyết cổ điển. Đây là định nghĩa:

```
class HypothesisTest(object):

    def __init__(self, data):

        self.data = data
        self.MakeModel()
        self.actual = self.TestStatistic(data)

    def PValue(self, iters=1000):

        self.test_stats = [self.TestStatistic(self.RunModel())

                           for _ in range(iters)]
                           count = sum(1 for x in self.test_stats if x >= self.actual)

        return count / iters

    def TestStatistic(self, data):

        raise UnimplementedMethodException()

    def MakeModel(self):

        pass

    def RunModel(self):

        raise UnimplementedMethodException()
```

HypothesisTest là một lớp bố mẹ trừu tượng cung cấp các định nghĩa hoàn chỉnh cho một số phương thức và trình giữ chỗ cho những phương thức khác. Các lớp con dựa trên HypothesisTest kế thừa __init__ và PValue và cung cấp TestStatistic, RunModel và MakeModel tùy chọn.

__init__ lấy dữ liệu ở bất kỳ dạng nào phù hợp. Nó gọi MakeModel, xây dựng một đại diện cho giả thuyết không, sau đó chuyển dữ liệu đến TestStatistic, tính toán kích thước của hiệu ứng trong mẫu.

PValue tính toán xác suất của hiệu ứng rõ ràng theo giả thuyết không. Nó nhận một tham số iters, là số lượng mô phỏng để chạy. Dòng đầu tiên tạo dữ liệu mô phỏng, tính toán số liệu thống kê thử nghiệm và lưu trữ chúng trong test_stats. Kết quả là tỷ lệ phần tử trong test_stats vượt quá hoặc bằng thống kê kiểm tra được quan sát, self.actual.

ví dụ đơn giản, giả sử chúng ta tung đồng xu 250 lần và thấy 140 mặt ngửa và 110 mặt sấp. Dựa trên kết quả này, chúng ta có thể nghi ngờ rằng đồng xu bị sai lệch; nghĩa là, có nhiều khả năng hạ cánh bằng đầu hơn. Để kiểm tra giả thuyết này, chúng ta tính toán xác suất nhìn thấy hiệu số như vậy nếu đồng xu thực sự công bằng:

```
class CoinTest(thinkstats2.HypothesisTest):
```

```
def TestStatistic(self, data):
        heads, tails = data
        test_stat = abs(heads - tails)
        return test_stat
def RunModel(self):
        heads, tails = self.data
        n = heads + tails
        sample = [random.choice('HT') for _ in range(n)]
        hist = thinkstats2.Hist(sample)
        data = hist['H'], hist['T']
        return data
```

Tham số, dữ liệu, là một cặp số nguyên: số đầu và đuôi. Thống kê kiểm tra là hiệu tuyệt đối giữa chúng, vì vậy self.actual là 30.

RunModel mô phỏng việc tung đồng xu với giả định rằng đồng xu thực sự công bằng. Nó tạo ra một mẫu gồm 250 lần tung, sử dụng Hist để đếm số lượng mặt ngửa và mặt sấp, đồng thời trả về một cặp số nguyên.

Bây giờ tất cả những gì chúng ta phải làm là khởi tạo CoinTest và gọi PValue:

```
ct = CoinTest((140, 110))
pvalue = ct.PValue()
```

Kết quả là khoảng 0,07, có nghĩa là nếu đồng xu công bằng, chúng tôi hy vọng sẽ thấy hiệu số lớn tới 30 trong khoảng 7% thời gian.

Chúng ta nên giải thích kết quả này như thế nào? Theo quy ước, 5% là ngưỡng có ý nghĩa thống kê. Nếu giá trị p nhỏ hơn 5%, hiệu ứng được coi là đáng kể; ngược lại thì không.

Nhưng việc lựa chọn 5% là tùy ý và (như chúng ta sẽ thấy sau) giá trị p phụ thuộc vào việc lựa chọn thống kê kiểm định và mô hình của giả thuyết khống. Vì vậy, giá trị p không nên được coi là phép đo chính xác.

Tôi khuyên bạn nên diễn giải các giá trị p theo thứ tự độ lớn của chúng: nếu giá trị p nhỏ hơn 1%, hiệu ứng này không chắc là do ngẫu nhiên; nếu nó lớn hơn 10%, hiệu ứng có thể được giải thích một cách hợp lý là do ngẫu nhiên. Giá trị P trong khoảng từ 1% đến 10% nên được coi là đường biên giới. Vì vậy, trong ví dụ này, tôi kết luận rằng dữ liệu không cung cấp bằng chứng chắc chắn rằng đồng xu có bị sai lệch hay không.

Kiểm tra hiệu số giá trị trung bình

Một trong những hiệu ứng phổ biến nhất để kiểm tra là hiệu số giá trị trung bình giữa hai nhóm. Trong dữ liệu của NSFG, chúng tôi thấy rằng thời gian mang thai trung bình của trẻ sơ sinh dài hơn một chút và cân nặng khi sinh trung bình nhỏ hơn một chút. Bây giờ chúng ta sẽ xem liệu những tác động đó có ý nghĩa thống kê hay không.

Đối với những ví dụ này, giả thuyết không là các bản phân phối cho hai nhóm là như nhau. Một cách để lập mô hình giả thuyết khống là hoán vị; nghĩa là, chúng ta có thể lấy các giá trị cho những đứa trẻ đầu lòng và những đứa trẻ khác rồi xáo trộn chúng, coi hai nhóm là một nhóm lớn:

```
class DiffMeansPermute(thinkstats2.HypothesisTest):

    def TestStatistic(self, data):

        group1, group2 = data

        test_stat = abs(group1.mean() - group2.mean())

        return test_stat

    def MakeModel(self):

        group1, group2 = self.data

        self.n, self.m = len(group1), len(group2)

        self.pool = np.hstack((group1, group2))

    def RunModel(self):

        np.random.shuffle(self.pool)

        data = self.pool[:self.n], self.pool[self.n:]

        return data
```

data là một cặp trình tự, một cho mỗi nhóm. Thống kê kiểm tra là hiệu số tuyệt đối trong phương tiện.

MakeModel ghi lại kích thước của các nhóm, n và m, đồng thời kết hợp các nhóm thành một mảng NumPy, self.pool.

RunModel mô phỏng giả thuyết không bằng cách xáo trộn các giá trị được gộp lại và chia chúng thành hai nhóm có kích thước n và m. Như mọi khi, giá trị trả về từ RunModel có cùng định dạng với dữ liệu được quan sát.

Để kiểm tra hiệu số về thời gian mang thai, chúng tôi chạy:

```
live, firsts, others = first.MakeFrames()

data = firsts.prglngth.values, others.prglngth.values

ht = DiffMeansPermute(data)

pvalue = ht.PValue()
```

MakeFrames đọc dữ liệu NSFG và trả về DataFrames đại diện cho tất cả các ca sinh sống, trẻ sơ sinh đầu tiên và các trường hợp khác. Chúng tôi trích xuất độ dài thai kỳ dưới dạng mảng NumPy, chuyển chúng dưới dạng dữ liệu tới DiffMeansPermute và tính toán giá trị p. Kết quả là khoảng 0,17, có nghĩa là chúng ta hy vọng sẽ thấy hiệu số lớn bằng hiệu ứng quan sát được trong khoảng 17% thời gian. Vì vậy ảnh hưởng này không có ý nghĩa thống kê.

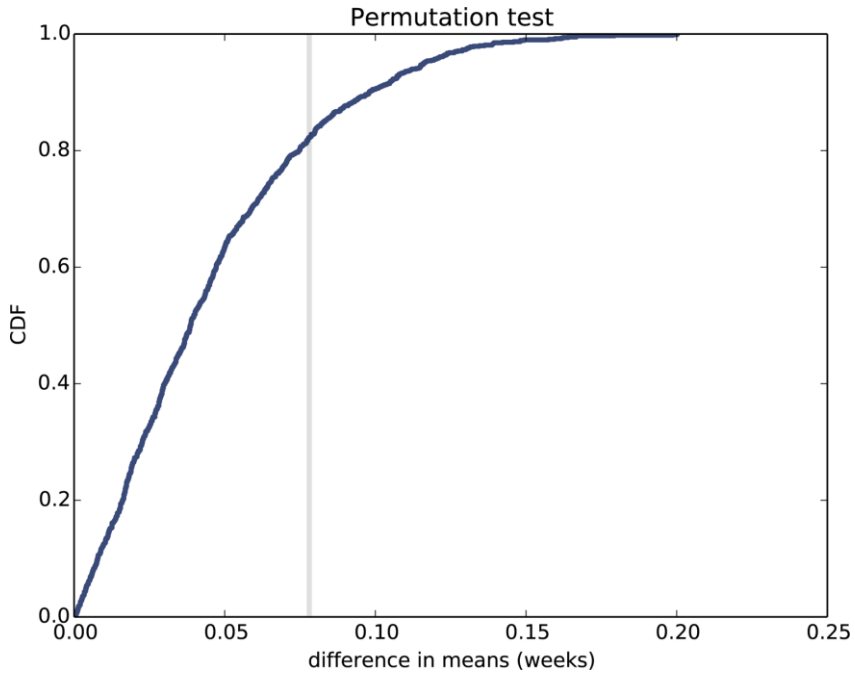
HypothesisTest cung cấp PlotCdf, biểu đồ phân phối thống kê thử nghiệm và một đường màu xám biểu thị kích thước hiệu ứng quan sát được:

```
ht.PlotCdf()

thinkplot.Show(xlabel='test statistic',

                    ylabel='CDF')
```

Hình 9-1 cho thấy kết quả. CDF cắt chênh lệch quan sát được ở 0,83, là phần bù của giá trị p, 0,17.

Nếu chúng ta chạy phân tích tương tự với cân nặng khi sinh, giá trị p được tính là 0; sau 1000 lần thử, mô phỏng không bao giờ mang lại hiệu quả lớn như chênh lệch quan sát được, 0,12 lbs. Vì vậy, chúng tôi sẽ báo cáo p <0,001 và kết luận rằng hiệu số về cân nặng khi sinh là có ý nghĩa thống kê.

Hình 9-1 cho thấy kết quả. CDF cắt chênh lệch quan sát được ở 0,83, là phần bù của giá trị p, 0,17.



Hình 9-1. CDF của hiệu số về thời gian mang thai trung bình theo giả thuyết khống

Nếu chúng ta chạy phân tích tương tự với cân nặng khi sinh, giá trị p được tính là 0; sau 1000 lần thử, mô phỏng không bao giờ mang lại hiệu quả lớn như chênh lệch quan sát được, 0,12 lbs. Vì vậy, chúng tôi sẽ báo cáo p <0,001 và kết luận rằng hiệu số về cân nặng khi sinh là có ý nghĩa thống kê.

Thống kê kiểm tra khác

Việc chọn thống kê thử nghiệm tốt nhất phụ thuộc vào câu hỏi mà bạn đang cố gắng giải quyết. Ví dụ: nếu câu hỏi liên quan là liệu thời gian mang thai có khác nhau đối với những đứa trẻ đầu lòng hay không, thì sẽ hợp lý khi kiểm tra giá trị tuyệt đối của hiệu số trung bình, như chúng ta đã làm trong phần trước.

Nếu chúng ta có lý do nào đó để nghĩ rằng những đứa trẻ đầu lòng có khả năng sinh muộn, thì chúng ta sẽ không lấy giá trị tuyệt đối của hiệu số; thay vào đó, chúng tôi sẽ sử dụng thống kê thử nghiệm này:

```
class DiffMeansOneSided(DiffMeansPermute):

    def TestStatistic(self, data):

        group1, group2 = data

        test_stat = group1.mean() - group2.mean()

        return test_stat
```

DiffMeansOneSided kế thừa MakeModel và RunModel từ DiffMeansPermute; hiệu số duy nhất là TestStatistic không lấy giá trị tuyệt đối của hiệu số. Loại bài kiểm tra này được gọi là kiểm tra một phía vì nó chỉ tính một phía của sự phân bổ hiệu số. Bài kiểm tra trước, sử dụng cả hai phía, là hai phía.

Đối với phiên bản thử nghiệm này, giá trị p là 0,09. Nói chung, giá trị p của phép thử một phía bằng khoảng một nửa giá trị p của phép thử hai phía, tùy thuộc vào hình dạng của phân phối.

Giả thuyết một phía, rằng những đứa trẻ đầu tiên được sinh ra muộn, cụ thể hơn giả thuyết hai phía, vì vậy giá trị p nhỏ hơn. Nhưng ngay cả đối với giả thuyết mạnh hơn, hiệu số không có ý nghĩa thống kê.

Chúng ta có thể sử dụng cùng một khuôn khổ để kiểm tra hiệu số về độ lệch chuẩn. Trong phần "Những hình dung khác" ở trang 30, chúng tôi đã thấy một số bằng chứng cho thấy những đứa trẻ đầu lòng có nhiều khả năng sinh sớm hoặc muộn hơn và ít có khả năng sinh đúng giờ hơn. Vì vậy, chúng tôi có thể đưa ra giả thuyết rằng độ lệch chuẩn cao hơn. Đây là cách chúng tôi có thể kiểm tra điều đó:

```
class DiffStdPermute(DiffMeansPermute):

    def TestStatistic(self, data):

        group1, group2 = data

        test_stat = group1.std() - group2.std()

        return test_stat
```

Đây là phép thử một phía vì giả thuyết cho rằng độ lệch chuẩn của những đứa trẻ đầu lòng cao hơn chứ không chỉ khác biệt. Giá trị p là 0,09, không có ý nghĩa thống kê.

Kiểm tra mối tương quan

Khung này cũng có thể kiểm tra các mối tương quan. Ví dụ: trong bộ dữ liệu NSFG, mối tương quan giữa cân nặng khi sinh và tuổi của mẹ là khoảng 0,07. Có vẻ như những bà mẹ lớn tuổi hơn có những đứa con nặng cân hơn. Nhưng hiệu ứng này có thể là do cơ hội?

Đối với thống kê kiểm tra, tôi sử dụng tương quan của Pearson, nhưng tương quan của Spearman cũng sẽ hoạt động tốt. Nếu chúng tôi có lý do để mong đợi mối tương quan tích cực, chúng tôi sẽ thực hiện kiểm tra một phía. Nhưng vì chúng tôi không có lý do như vậy, tôi sẽ thực hiện một bài kiểm tra hai phía bằng cách sử dụng giá trị tương quan tuyệt đối.

Giả thuyết khống là không có mối tương quan giữa tuổi của mẹ và cân nặng khi sinh. Bằng cách xáo trộn các giá trị được quan sát, chúng ta có thể mô phỏng một thế giới nơi phân bố tuổi và cân nặng khi sinh giống nhau, nhưng ở đó các biến số không liên quan:

```
class CorrelationPermute(thinkstats2.HypothesisTest):

    def TestStatistic(self, data):

        xs, ys = data

        test_stat = abs(thinkstats2.Corr(xs, ys))

        return test_stat

    def RunModel(self):

        xs, ys = self.data

        xs = np.random.permutation(xs)

        return xs, ys
```

data là một cặp trình tự. TestStatistic tính giá trị tuyệt đối của tương quan Pearson. RunModel xáo trộn xs và trả về dữ liệu mô phỏng.

Đây là mã đọc dữ liệu và chạy thử nghiệm:

```
live, firsts, others = first.MakeFrames()

live = live.dropna(subset=['agepreg', 'totalwgt_lb'])

data = live.agepreg.values, live.totalwgt_lb.values

ht = CorrelationPermute(data)

pvalue = ht.PValue()
```

Tôi sử dụng dropna với đối số tập hợp con để loại bỏ các hàng thiếu một trong các biến mà chúng tôi cần.

Mối tương quan thực tế là 0,07. Giá trị p được tính là 0; sau 1000 lần lặp, tương quan mô phỏng lớn nhất là 0,04. Vì vậy, mặc dù mối tương quan quan sát được là nhỏ, nhưng nó có ý nghĩa thống kê.

Ví dụ này là một lời nhắc nhở rằng "có ý nghĩa thống kê" không phải lúc nào cũng có nghĩa là một tác động là quan trọng hoặc có ý nghĩa trong thực tế. Nó chỉ có nghĩa là nó không thể xảy ra một cách tình cờ.

Kiểm tra tỷ lệ

Giả sử bạn điều hành một sòng bạc và bạn nghi ngờ rằng một khách hàng đang sử dụng một con súc sắc bị vẹo; nghĩa là, một khuôn mặt đã được sửa đổi để làm cho một trong các khuôn mặt có nhiều khả năng hơn những khuôn mặt khác. Bạn bắt được kẻ bị cáo buộc gian lận và tịch thu súc sắc, nhưng bây giờ bạn phải chứng minh rằng đó là gian lận. Bạn tung xúc xắc 60 lần và nhận được kết quả như sau:

Giá trị	1	2	3	4	5	6
Tần suất	8	9	19	5	8	11

Trung bình bạn mong đợi mỗi giá trị xuất hiện 10 lần. Trong tập dữ liệu này, giá trị 3 xuất hiện thường xuyên hơn dự kiến và giá trị 4 xuất hiện ít hơn. Nhưng những khác biệt này có ý nghĩa thống kê không?

Để kiểm tra giả thuyết này, chúng ta có thể tính toán tần suất dự kiến cho từng giá trị, chênh lệch giữa tần suất dự kiến và tần suất quan sát được và tổng chênh lệch tuyệt đối. Trong ví dụ này, chúng tôi hy vọng mỗi bên sẽ tăng 10 lần trong số 60; độ lệch so với kỳ vọng này là -2, -1, 9, -5, -2 và 1; vậy tổng chênh lệch tuyệt đối là 20. Chúng ta có thường tình cờ thấy hiệu số như vậy không?

Đây là phiên bản của HypothesisTest trả lời câu hỏi đó:

```
class DiceTest(thinkstats2.HypothesisTest):

    def TestStatistic(self, data):

        observed = data

        n = sum(observed)

        expected = np.ones(6) * n / 6

        test_stat = sum(abs(observed - expected))

        return test_stat

    def RunModel(self):

        n = sum(self.data)

        values = [1, 2, 3, 4, 5, 6]

        rolls = np.random.choice(values, n, replace=True)

        hist = thinkstats2.Hist(rolls)

        freqs = hist.Freqs(values)

        return freqs
```

Dữ liệu được biểu diễn dưới dạng danh sách các tần số: các giá trị được quan sát là [8, 9, 19, 5, 8, 11]; tần suất dự kiến đều là 10. Thống kê kiểm tra là tổng của các chênh lệch tuyệt đối.

Giả thuyết vô hiệu là con súc sắc công bằng, vì vậy chúng tôi mô phỏng điều đó bằng cách rút các mẫu ngẫu nhiên từ các giá trị. RunModel sử dụng Hist để tính toán và trả về danh sách tần số.

Giá trị p cho dữ liệu này là 0,13, có nghĩa là nếu súc sắc công bằng, chúng tôi hy vọng sẽ thấy tổng độ lệch quan sát được, hoặc hơn, khoảng 13% thời gian. Vì vậy, hiệu quả rõ ràng là không có ý nghĩa thống kê.

Kiểm tra Chi-bình phương

Trong phần trước, chúng tôi đã sử dụng tổng độ lệch làm thống kê kiểm tra. Nhưng để kiểm tra tỷ lệ, thông thường hơn là sử dụng thống kê Chi-bình phương:

$$\chi^2 = \sum_i \frac{(O_i - E_i)^2}{E_i}$$

Trong đó O_i là tần số quan sát được và E_i là tần số mong đợi. Đây là mã Python:

```
class DiceChiTest(DiceTest):

    def TestStatistic(self, data):

        observed = data

        n = sum(observed)

        expected = np.ones(6) * n / 6

        test_stat = sum((observed - expected)**2 / expected)

        return test_stat
```

Bình phương độ lệch (thay vì lấy giá trị tuyệt đối) mang lại nhiều trọng lượng hơn cho độ lệch lớn. Việc chia cho kỳ vọng tiêu chuẩn hóa độ lệch, mặc dù trong trường hợp này nó không có tác dụng vì các tần số kỳ vọng đều bằng nhau.

Giá trị p sử dụng thống kê Chi-bình phương là 0,04, nhỏ hơn đáng kể so với giá trị chúng tôi nhận được khi sử dụng tổng độ lệch, 0,13. Nếu chúng tôi coi trọng ngưỡng 5%, chúng tôi sẽ coi tác động này có ý nghĩa thống kê. Nhưng khi xem xét cả hai bài kiểm tra cùng nhau, tôi có thể nói rằng kết quả là gần như hoàn hảo. Tôi không loại trừ khả năng con súc sắc bị vẹo, nhưng tôi sẽ không kết tội kẻ gian lận bị buộc tội.

Ví dụ này cho thấy một điểm quan trọng: giá trị p phụ thuộc vào việc lựa chọn thống kê kiểm định và mô hình của giả thuyết không, và đôi khi những lựa chọn này xác định liệu một tác động có ý nghĩa thống kê hay không.

Những đứa trẻ đầu lòng một lần nữa

Ở đầu chương này, chúng ta đã xem xét thời gian mang thai của trẻ đầu lòng và những trẻ khác, và kết luận rằng hiệu số rõ ràng về giá trị trung bình và độ lệch chuẩn không có ý nghĩa thống kê. Nhưng trong phần "Hình dung khác" ở trang 30, chúng tôi đã thấy một số khác biệt rõ ràng trong sự phân bổ độ dài của thai kỳ, đặc biệt là trong khoảng từ 35 đến 43 tuần. Để xem liệu những khác biệt đó có ý nghĩa thống kê hay không, chúng ta có thể sử dụng phép kiểm tra dựa trên thống kê Chi-bình phương.

Mã này kết hợp các yếu tố từ các ví dụ trước:

```
class PregLengthTest(thinkstats2.HypothesisTest):

    def MakeModel(self):

        firsts, others = self.data

        self.n = len(firsts)

        self.pool = np.hstack((firsts, others))

        pmf = thinkstats2.Pmf(self.pool)

        self.values = range(35, 44)

        self.expected_probs = np.array(pmf.Probs(self.values))

    def RunModel(self):

        np.random.shuffle(self.pool)
```

```
                data = self.pool[:self.n], self.pool[self.n:]

                return data
```

Dữ liệu được biểu diễn dưới dạng hai danh sách về thời gian mang thai. Giả thuyết không là cả hai mẫu được rút ra từ cùng một phân phối. MakeModel lập mô hình phân phối bằng cách gộp hai mẫu bằng hstack. Sau đó, RunModel tạo dữ liệu mô phỏng bằng cách xáo trộn mẫu gộp và chia thành hai phần.

MakeModel cũng xác định các giá trị, đó là phạm vi số tuần chúng tôi sẽ sử dụng vàexpected_probs, là xác suất của từng giá trị trong phân phối gộp.

Đây là mã tính toán thống kê thử nghiệm:

```
# class PregLengthTest:

        def TestStatistic(self, data):

                firsts, others = data

                stat = self.ChiSquared(firsts) + self.ChiSquared(others)

                return stat

        def ChiSquared(self, lengths):

                hist = thinkstats2.Hist(lengths)

                observed = np.array(hist.Freqs(self.values))

                expected = self.expected_probs * len(lengths)

                stat = sum((observed - expected)**2 / expected)

                return stat
```

TestStatistic tính toán thống kê Chi-bình phương cho trẻ sơ sinh đầu lòng và những trẻ khác, rồi cộng chúng lại.

ChiSquared lấy một chuỗi thời gian mang thai, tính toán biểu đồ của nó và tính toán quan sát được, đây là danh sách các tần số tương ứng với giá trị bản thân. Để tính toán danh sách các tần suất dự kiến, nó nhân các xác suất được tính toán trước, expected_probs, với kích thước mẫu. Nó trả về thống kê Chi-bình phương.

Đối với dữ liệu NSFG, tổng thống kê Chi-bình phương là 102, bản thân điều này không có nhiều ý nghĩa. Nhưng sau 1.000 lần lặp lại, thống kê thử nghiệm lớn nhất được tạo ra theo giả thuyết không là 32. Chúng tôi kết luận rằng thống kê Chi-bình phương được quan sát là không thể theo giả thuyết không, vì vậy tác động rõ ràng là có ý nghĩa thống kê.

Ví dụ này cho thấy một hạn chế của các bài kiểm tra Chi-bình phương: chúng chỉ ra rằng có hiệu số giữa hai nhóm, nhưng chúng không nói bất cứ điều gì cụ thể về hiệu số đó là gì.

Sai số

Trong thử nghiệm giả thuyết cổ điển, một hiệu ứng được coi là có ý nghĩa thống kê nếu giá trị p nằm dưới ngưỡng nào đó, thường là 5%. Thủ tục này đặt ra hai câu hỏi:

• Nếu hiệu quả thực sự là do ngẫu nhiên, xác suất mà chúng ta sẽ coi nó là quan trọng là bao nhiêu? Xác suất này là **tỷ lệ dương tính giả**.

• Nếu hiệu quả là có thật, khả năng kiểm tra giả thuyết sẽ thất bại là bao nhiêu? Xác suất này là **tỷ lệ âm tính giả**.

Tỷ lệ dương tính giả tương đối dễ tính: nếu ngưỡng là 5%, tỷ lệ dương tính giả là 5%. Đây là lý do tại sao:

• Nếu không có hiệu ứng thực sự, thì giả thuyết không là đúng, vì vậy chúng ta có thể tính toán phân phối của thống kê kiểm định bằng cách mô phỏng giả thuyết không. Gọi CDF_T phân phối này.

• Mỗi khi chúng tôi chạy một thử nghiệm, chúng tôi nhận được một thống kê thử nghiệm, *t*, được rút ra từ CDF_T. Sau đó, chúng tôi tính toán giá trị p, là xác suất mà một giá trị ngẫu nhiên từ CDF_T vượt quá *t*, do đó, đó là $1 - CDF_T(t)$.

• Giá trị p nhỏ hơn 5% nếu $CDF_T(t)$ lớn hơn 95%; nghĩa là, nếu $_t$ vượt quá phân vị thứ 95. Và giá trị được chọn từ CDF_T có thường xuyên vượt quá phân vị thứ 95 không? 5% thời gian.

Vì vậy, nếu bạn thực hiện một thử nghiệm giả thuyết với ngưỡng 5%, bạn sẽ có 1 lần dương tính giả trong 20.

Sức mạnh

Tỷ lệ âm tính giả khó tính toán hơn vì nó phụ thuộc vào kích thước hiệu ứng thực tế và thông thường chúng tôi không biết điều đó. Một lựa chọn là tính toán tỷ lệ có điều kiện dựa trên kích thước hiệu ứng giả định.

Ví dụ: nếu chúng ta cho rằng hiệu số quan sát được giữa các nhóm là chính xác, thì chúng ta có thể sử dụng các mẫu được quan sát làm mô hình dân số và chạy thử nghiệm giả thuyết với dữ liệu mô phỏng:

```
def FalseNegRate(data, num_runs=100):
    group1, group2 = data
    count = 0
    for i in range(num_runs):
        sample1 = thinkstats2.Resample(group1)
        sample2 = thinkstats2.Resample(group2)
        ht = DiffMeansPermute((sample1, sample2))
        pvalue = ht.PValue(iters=101)
        if pvalue > 0.05:
            count += 1
    return count / num_runs
```

FalseNegRate lấy dữ liệu ở dạng hai chuỗi, một chuỗi cho mỗi nhóm. Mỗi lần qua vòng lặp, nó mô phỏng một thử nghiệm bằng cách lấy một mẫu ngẫu nhiên từ mỗi nhóm và chạy thử nghiệm giả thuyết. Sau đó, nó kiểm tra kết quả và đếm số lượng âm tính giả.

Resample lấy một chuỗi và vẽ một mẫu có cùng độ dài, với sự thay thế:

```
def Resample(xs):
    return np.random.choice(xs, len(xs), replace=True)
```

Đây là mã kiểm tra thời gian mang thai:

```
live, firsts, others = first.MakeFrames()
data = firsts.prglngth.values, others.prglngth.values
neg_rate = FalseNegRate(data)
```

Kết quả là khoảng 70%, có nghĩa là nếu chênh lệch thực tế về thời gian mang thai trung bình là 0,78 tuần, thì chúng tôi hy vọng một thử nghiệm với cỡ mẫu này sẽ cho kết quả âm tính trong 70% thời gian.

Kết quả này thường được trình bày theo cách khác: nếu chênh lệch thực tế là 0,78 tuần, chúng ta chỉ nên mong đợi kết quả xét nghiệm dương tính trong 30% thời gian. "Tỷ lệ dương tính chính xác" này được gọi là sức mạnh của xét nghiệm, hoặc đôi khi là "độ nhạy". Nó phản ánh khả năng của thử nghiệm để phát hiện ảnh hưởng của một kích thước nhất định.

Trong ví dụ này, thử nghiệm chỉ có 30% cơ hội cho kết quả dương tính (một lần nữa, giả sử rằng hiệu số là 0,78 tuần). Theo nguyên tắc thông thường, công suất 80% được coi là chấp nhận được, vì vậy chúng tôi có thể nói rằng thử nghiệm này "không đủ sức mạnh".

Nói chung, một thử nghiệm giả thuyết tiêu cực không ngụ ý rằng không có hiệu số giữa các nhóm; thay vào đó, nó gợi ý rằng nếu có hiệu số, thì nó quá nhỏ để phát hiện với cỡ mẫu này.

Nhân rộng

Quá trình kiểm tra giả thuyết mà tôi đã trình bày trong chương này, nói đúng ra, không phải là một thực hành tốt.

Đầu tiên, tôi thực hiện nhiều bài kiểm tra. Nếu bạn chạy một thử nghiệm giả thuyết, xác suất dương tính giả là khoảng 1 trên 20, điều này có thể chấp nhận được. Nhưng nếu bạn chạy 20 bài kiểm tra, thì hầu hết thời gian bạn sẽ có ít nhất một kết quả dương tính giả.

Thứ hai, tôi đã sử dụng cùng một bộ dữ liệu để khám phá và thử nghiệm. Nếu bạn khám phá một tập dữ liệu lớn, tìm thấy một hiệu ứng đáng ngạc nhiên và sau đó kiểm tra xem nó có đáng kể hay không, bạn có nhiều khả năng tạo ra kết quả dương tính giả.

Để bù cho nhiều thử nghiệm, bạn có thể điều chỉnh ngưỡng giá trị p (xem trang Wikipedia này). Hoặc bạn có thể giải quyết cả hai vấn đề bằng cách phân vùng dữ liệu, sử dụng một bộ để khám phá và bộ kia để thử nghiệm.

Trong một số lĩnh vực, những thực hành này là bắt buộc hoặc ít nhất là được khuyến khích. Nhưng người ta cũng thường giải quyết những vấn đề này một cách ngầm định bằng cách sao chép các kết quả đã công bố. Thông thường, bài báo đầu tiên báo cáo một kết quả mới được coi là thăm dò. Các bài báo tiếp theo sao chép kết quả với dữ liệu mới được coi là xác nhận.

Khi điều đó xảy ra, chúng ta có cơ hội lặp lại các kết quả trong chương này. Ấn bản đầu tiên của cuốn sách này dựa trên Chu kỳ 6 của NSFG, được phát hành vào năm 2002. Vào tháng 10 năm 2011, CDC đã công bố dữ liệu bổ sung dựa trên các cuộc phỏng vấn được thực hiện từ năm 2006–2010. nsfg2.py chứa mã để đọc và xóa dữ liệu này. Trong tập dữ liệu mới:

• Hiệu số về thời gian mang thai trung bình là 0,16 tuần và có ý nghĩa thống kê với $p < 0,001$ (so với 0,078 tuần trong tập dữ liệu gốc).

• Hiệu số về cân nặng khi sinh là 0,17 pound với $p < 0,001$ (so với 0,12 pound trong tập dữ liệu gốc).

• Mối tương quan giữa cân nặng khi sinh và tuổi mẹ là 0,08 với $p < 0,001$ (so với 0,07).

• Kiểm định Chi-bình phương có ý nghĩa thống kê với $p < 0,001$ (như trong bản gốc).

Tóm lại, tất cả các tác động có ý nghĩa thống kê trong tập dữ liệu gốc đã được sao chép trong tập dữ liệu mới và hiệu số về thời gian mang thai, vốn không có ý nghĩa trong tập dữ liệu gốc, lớn hơn và có ý nghĩa trong tập dữ liệu mới.

Bài tập

Lời giải cho những bài tập này có trong chap09soln.py.

Bài tập 9-1.

Khi kích thước mẫu tăng lên, sức mạnh của phép thử giả thuyết tăng lên, điều đó có nghĩa là nó có nhiều khả năng dương tính hơn nếu hiệu ứng là có thật. Ngược lại, khi kích thước mẫu giảm, thử nghiệm ít có khả năng dương tính ngay cả khi hiệu ứng là có thật.

Để điều tra hành vi này, hãy chạy thử nghiệm trong chương này với các tập hợp con khác nhau của dữ liệu NSFG. Bạn có thể sử dụng thinkstats2.SampleRows để chọn một tập hợp con ngẫu nhiên của các hàng trong DataFrame.

Điều gì xảy ra với giá trị p của các thử nghiệm này khi cỡ mẫu giảm? Kích thước mẫu nhỏ nhất mang lại kết quả xét nghiệm dương tính là bao nhiêu?

Bài tập 9-2.

Trong "Kiểm tra hiệu số về phương tiện" ở trang 104, chúng tôi đã mô phỏng giả thuyết không bằng phép hoán vị; nghĩa là, chúng tôi xử lý các giá trị được quan sát như thể chúng đại diện cho toàn bộ dân số và chỉ định ngẫu nhiên các thành viên của dân số vào hai nhóm.

Một cách khác là sử dụng mẫu để ước tính phân phối cho dân số, sau đó rút một mẫu ngẫu nhiên từ phân phối đó. Quá trình này được gọi là lấy mẫu lại. Có một số cách để thực

hiện lấy mẫu lại, nhưng một trong những cách đơn giản nhất là lấy một mẫu với sự thay thế từ các giá trị được quan sát, như trong "Sức mạnh" trên trang 112.

Viết một lớp có tên DiffMeansResample kế thừa từ DiffMeansPermute và ghi đè RunModel để triển khai lấy mẫu lại, thay vì hoán vị. Sử dụng mô hình này để kiểm tra hiệu số về chiều dài thai kỳ và cân nặng khi sinh. Mô hình ảnh hưởng bao nhiêu đến kết quả?

Chú giải

Thử nghiệm giả thuyết

Quá trình xác định liệu một tác động rõ ràng có ý nghĩa thống kê hay không.

Thử nghiệm thống kê

Một thống kê được sử dụng để định lượng kích thước hiệu ứng.

Giả thuyết vô hiệu

Một mô hình của một hệ thống dựa trên giả định rằng một hiệu ứng rõ ràng là do ngẫu nhiên.

Giá trị p

Xác suất mà một hiệu ứng có thể xảy ra một cách tình cờ.

Ý nghĩa thống kê

Một tác động có ý nghĩa thống kê nếu nó không xảy ra một cách tình cờ.

Kiểm tra hoán vị

Một cách để tính giá trị p bằng cách tạo các hoán vị của tập dữ liệu được quan sát.

Kiểm tra lấy mẫu lại

Một cách để tính giá trị p bằng cách tạo mẫu, có thay thế, từ tập dữ liệu được quan sát.

Kiểm tra hai phía

Một bài kiểm tra đặt câu hỏi, "Khả năng xảy ra một tác động lớn như tác động quan sát được, tích cực hay tiêu cực là bao nhiêu?"

Kiểm tra một phía

Một bài kiểm tra hỏi, "Khả năng xảy ra một hiệu ứng lớn như hiệu ứng quan sát được và có cùng dấu là bao nhiêu?"

Kiểm tra Chi-bình phương

Một bài kiểm tra sử dụng thống kê chi bình phương làm thống kê kiểm tra.

Dương tính giả

Kết luận rằng một hiệu ứng là có thật khi nó không phải vậy.

Âm tính giả

Kết luận rằng một hiệu ứng là do ngẫu nhiên khi nó không phải như vậy.

Sức mạnh

Xác suất của một thử nghiệm tích cực nếu giả thuyết không là sai.