

text_causality

December 9, 2015

```
In [1]: # HIDDEN
        from datascience import *
        %matplotlib inline

        import matplotlib.pyplot as plots
        plots.style.use('fivethirtyeight')
        import pylab as pl
        import math
        from scipy import stats
```

0.1 Difference and Causality

In the previous example, we asked whether the amount of radiation a patient received affected his/her 15-month score. Radiation and the 15-month score are distinct physical quantities, measured in different units. The scatter plot and regression are tools that help us measure the relation between such variables. However, if the variables are “like” quantities – that is, measured in the same units, such as the baseline score and 15-month score – then it might be possible to use simpler methods.

In the following examples, we will try to see if we can identify a difference between two groups of “like” measurements; if there is a difference, we will see if we can identify the cause. How we go about doing this, and whether we succeed, will depend very much on the details of the randomization involved in the sampling.

Example 1. Is there a difference between baseline scores and 15-month scores? If so, why?

The first question we will address is whether there is a difference between the scores of the Hodgkin’s disease patients before and after treatment. In order to make an inference about this based on the data, we first note that the baseline and 15-month scores are *paired* by patient – that is, there is one (baseline, 15-month) pair of scores for each patient. The interesting question is not whether the baseline and 15-month scores differ overall on average; it is whether there is a change in score per patient. It makes sense, therefore, to convert the two lists of baseline and 15-month scores to a single list of differences. We will do this by calculating the drop in score from baseline to 15 months after treatment.

```
In [27]: hodgkins = Table.read_table('hodgkins.csv')
        hodgkins['drop'] = hodgkins['base'] - hodgkins['month15']
        hodgkins
```

```
Out[27]: <Table(6 rows): | height | rad | chemo | base | month15 | drop |>
```

It seems apparent that there is indeed a drop: almost all the entries in the `drop` column are positive. One way to formalize this is to construct a bootstrap confidence interval for the mean drop in the population of patients, and see whether that confidence interval contains 0.

We can do this using code that we developed in an earlier section. The code is provided here for ease of reference.

```
In [ ]: """Bootstrap empirical distribution of random sample means,
        resampling from a sample
```

```

Arguments:
table of original sample data, column label, number of repetitions"""

def bootstrap_mean(samp_table, column_label, repetitions):

    # Set up an empty table to collect all the replicated medians
    means = []

    # Run the bootstrap and place all the medians in the table
    for i in range(repetitions):
        resample = samp_table.select(column_label).sample(with_replacement=True)
        m = np.mean(resample[column_label])
        means.append(m)

    # Display results
    means = Table([means], ['means'])
    means.hist(bins=20, normed=True)
    plots.xlabel('resampled means')
    print("Original sample mean:", np.mean(samp_table[column_label]))
    print("2.5 percent point of resampled means:", means.percentile(2.5).rows[0][0])
    print("97.5 percent point of resampled means:", means.percentile(97.5).rows[0][0])
    """Permutation test for the difference in means
    Category A: 0      Category B: 1"""

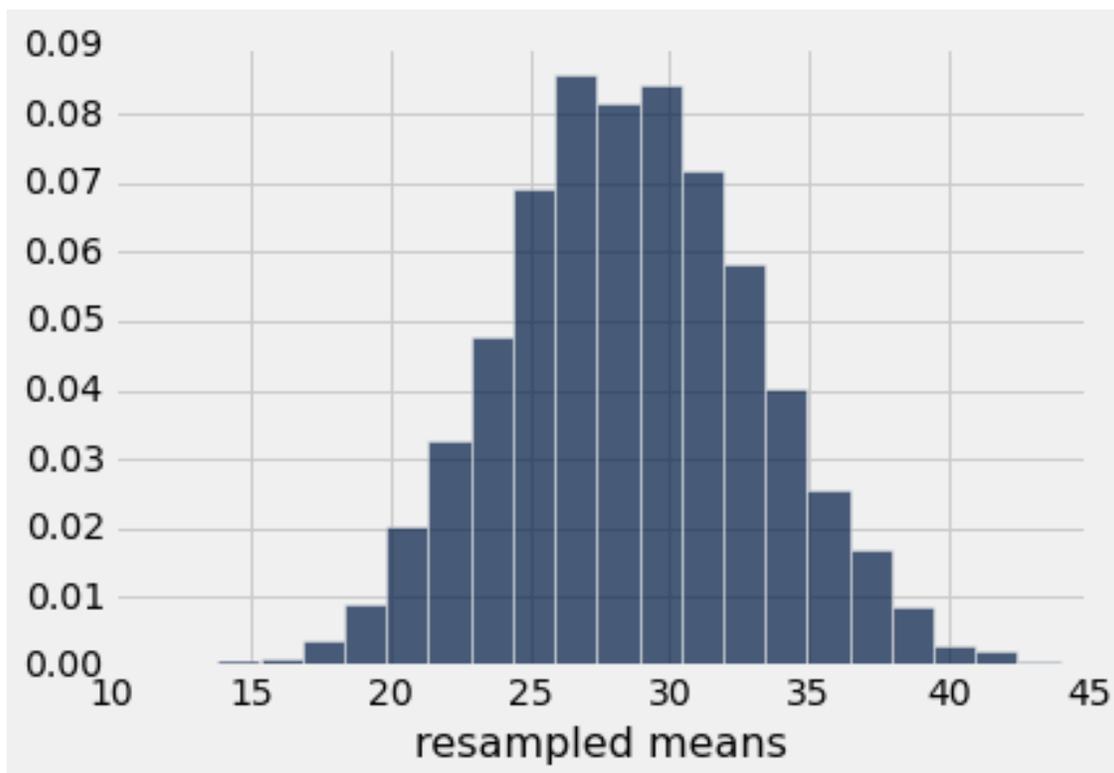
```

```
In [14]: bootstrap_mean(hodgkins, 'drop', 4000)
```

```

Original sample mean: 28.6159090909
2.5 percent point of resampled means: 20.0590909091
97.5 percent point of resampled means: 37.7040909091

```



An approximate 95% bootstrap confidence interval for the mean drop in the population is (20.06, 37.70). The interval does not contain 0, so it is reasonable to conclude that there is a drop in the population. But the confidence interval does more than suggest whether or not there is a drop; it estimates the size of the drop. With about 95% confidence, we are estimating an average drop of between about 20 and 38 in the population.

Why is there a drop? It is tempting to answer, “Because of the treatment.” But in fact the data don’t tell us that. In order to establish causality, a minimum requirement is comparison, as we saw in Chapter 1. We have to compare the treatment group with a control group. In the study in which our data arose, there was indeed a control group and much careful attention to eliminate confounding. That made it possible to point to the treatment as the cause of the observed difference.

In the absence of a control group, researchers sometimes resort to using *historical controls*. These are groups in previous analyses, that are similar to the current sample except for the treatment. It can be quite difficult to justify this similarity. That is why it is necessary to be cautious about making conclusions in settings where historical controls have been used.

Example 2. Is there a difference between the results of the treatment and control groups? If so, why? One setting in which it is possible to establish causality is a randomized controlled experiment.

One such experiment concerned a method for drawing blood intravenously. The standard IV method involved tying a band around the patient’s arm to make it easier to insert a needle into the vein; the new treatment essentially replaced the tied band with something like a rubber band that could be slipped on.

Here are the data. There were 505 subjects in the study. Of these, 241 were randomized into the control group and received the standard treatment. The remainder became the treatment group and received the new treatment. The table `iv` contains the data. We will ignore the columns of height and weight and just focus on the other two. The `Group` column contains the group label: 0 stands for control and 1 for treatment. The `success` column contains a 1 if the needle was inserted successfully, and 0 otherwise. There was a clear definition of “success” involving the number of attempts at insertion and the time involved.

```
In [15]: iv = Table.read_table('IV.csv').drop('sbp')
         iv
```

```
Out[15]: <Table(4 rows): | Group | height | weight | success |>
```

It is natural to compare the proportion of successes in the two groups. Somewhat surprisingly, the success rate in the control group is higher: 95% compared to 78% in the treatment group.

```
In [29]: control = iv.where(iv['Group'],0)
         treatment = iv.where(iv['Group'],1)
```

```
In [30]: np.count_nonzero(control['success'])/control.num_rows
```

```
Out[30]: 0.950207468879668
```

```
In [20]: np.count_nonzero(treatment['success'])/treatment.num_rows
```

```
Out[20]: 0.7803030303030303
```

Is the difference due to chance? To answer this question, we must first be clear about exactly what is being tested. Notice that unlike the data we have analyzed in the past, there is no large population from which a sample was drawn at random. Rather, a group of 505 people was split randomly into treatment and control. So what exactly is the question about?

From a practical perspective, the question is motivated by the possibility that it is just easier to insert IV needles into some patients than others, and that perhaps many of these patients were placed in the control group just by chance.

Formally, the question is about an *abstract model*. Before the experiment is conducted, imagine that each person has a score of 1 or 0 that he/she would get if assigned to the control group, and also a score that he/she would get if assigned to the treatment group. For each patient, we only get to see one of these two hypothetical scores. Which one we get to see depends on chance: the chance that determines whether the patient was assigned to the treatment group or to the control group.

There are 505 people in the experiment, so there are 505 of these hypothetical control scores and 505 hypothetical treatment scores. We get to see the control scores of 241 randomly picked patients, and the treatment scores of the *remaining* 264 patients.

The question, then, is about the difference between the proportion of 1's among all hypothetical control scores and the proportion of 1's among all hypothetical treatment scores.

Analysis in a randomized controlled experiment **Null Hypothesis:** The proportion of 1's among all 505 control scores is the same as the proportion of 1's among all 505 treatment scores. The difference in the sample is due to chance.

Alternative Hypothesis: The proportions of 1's in the two groups of 505 scores are different.

As our test statistic, we will use the difference between the proportions of 1's in the control sample and the treatment sample. The observed value of the test statistic is about 0.17:

```
In [31]: np.count_nonzero(control['success'])/control.num_rows - np.count_nonzero(treatment['success'])
```

```
Out[31]: 0.16990443857663773
```

We now have two ways of performing our test. One way is to use a bootstrap A/B test for the difference between means. Here is the code we developed for this in an earlier section, and the application of the code to our current question.

```
In [36]: """Bootstrap A/B test for the difference in the mean response
Assumes A=0, B=1"""
```

```
def bootstrap_AB_test_means(samp_table, response_label, ab_label, repetitions):

    # Sort the sample table according to the A/B column;
    # then select only the column of effects.
    response = samp_table.sort(ab_label).select(response_label)

    # Find the number of entries in Category A.
    n_A = samp_table.where(samp_table[ab_label],0).num_rows

    # Calculate the observed value of the test statistic.
    meanA = np.mean(response[response_label][:n_A])
    meanB = np.mean(response[response_label][n_A:])
    obs_diff = meanA - meanB

    # Run the bootstrap procedure and get a list of resampled differences in means
    diffs = []
    for i in range(repetitions):
        resample = response.sample(with_replacement=True)
        d = np.mean(resample[response_label][:n_A]) - np.mean(resample[response_label][n_A:])
        diffs.append([d])

    # Compute the bootstrap empirical P-value
    diff_array = np.array(diffs)
    p_value = np.count_nonzero(abs(diff_array) >= abs(obs_diff))/repetitions

    # Display results
```

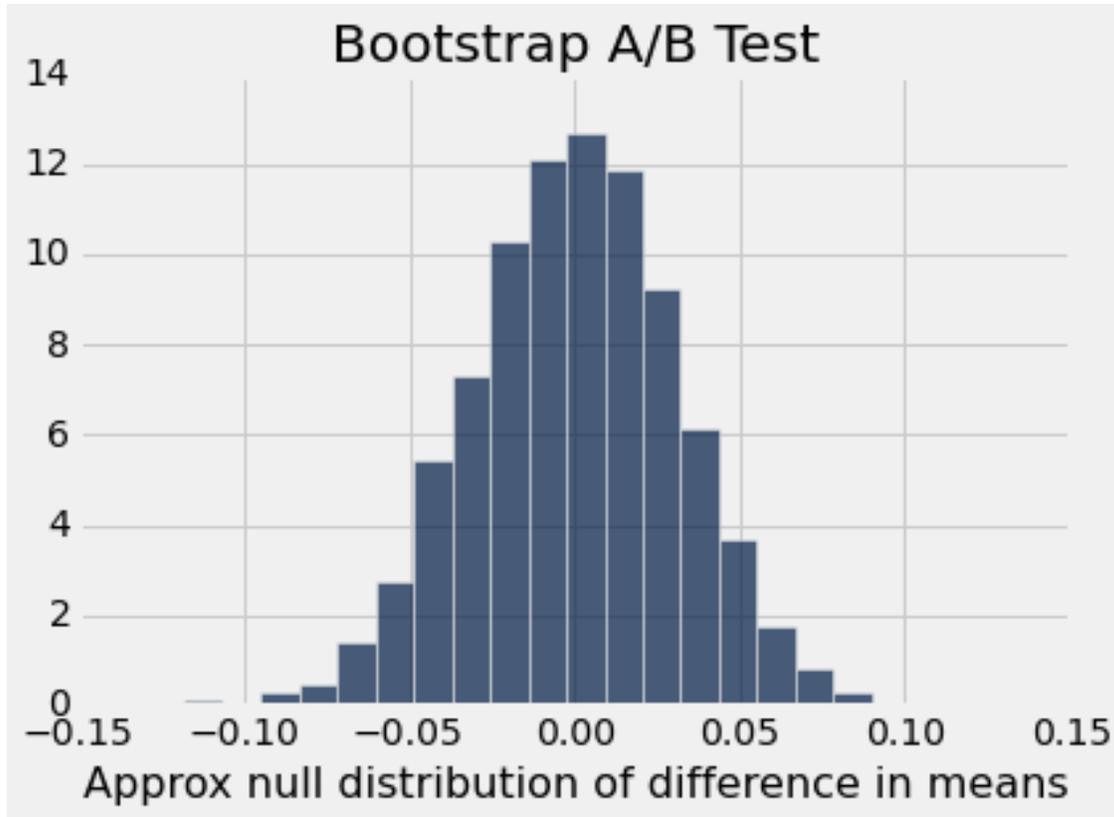
```

diffs = Table([diffs], ['diff_in_means'])
diffs.hist(bins=20, normed=True)
plots.xlabel('Approx null distribution of difference in means')
plots.title('Bootstrap A/B Test')
print("Observed difference in means: ", obs_diff)
print("Bootstrap empirical P-value: ", p_value)

```

```
In [37]: bootstrap_AB_test_means(iv, 'success', 'Group', 5000)
```

```
Observed difference in means: 0.169904438577
Bootstrap empirical P-value: 0.0
```



The empirical P -value is 0; none of the bootstrap replications yielded a difference as big as the observed. The data support the alternative hypothesis: the difference is not due to chance. Because the data are from a randomized controlled experiment, the only difference between the two groups being compared is the treatment. Therefore it is reasonable to conclude that the difference is due to the treatment.

We could also have answered the question using a permutation test for the difference between means. Here is the code we developed earlier for that test, and the result in our current setting. As you can see, the result is the same as before, except that the permutation-based empirical distribution of the statistic is less smooth than its bootstrap counterpart.

```

In [32]: def perm_test_means(samp_table, response_label, ab_label, repetitions):
          n_A = samp_table.where(samp_table[ab_label], 0).num_rows
          t_sorted = samp_table.select([response_label, ab_label]).sort(ab_label)

          # calculate the observed difference in means

```

```

meanA = np.mean(t_sorted[response_label][:n_A])
meanB = np.mean(t_sorted[response_label][n_A:])
obs_diff = meanA - meanB

diffs = []

for i in range(repetitions):

    # sample WITHOUT replacement, same number as original sample size
    resample = t_sorted.sample()

    # Compute the difference of the means of the resampled values, between Categories A and B
    dd = np.mean(resample[response_label][:n_A]) - np.mean(resample[response_label][n_A:])
    diffs.append([dd])

# Compute the empirical P-value
# Compute the bootstrap empirical P-value
diff_array = np.array(diffs)
p_value = np.count_nonzero(abs(diff_array) >= abs(obs_diff))/repetitions

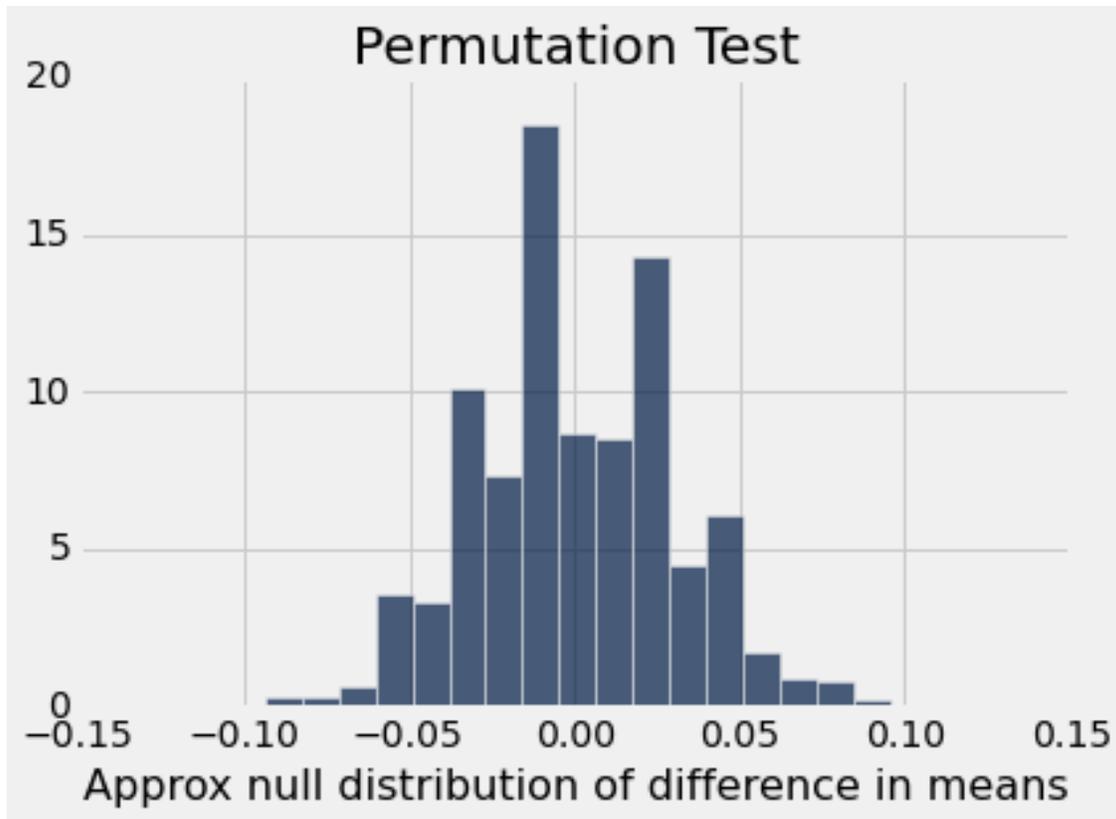
# Display results
diffs = Table([diffs], ['diff_in_means'])
diffs.hist(bins=20, normed=True)
plots.xlabel('Approx null distribution of difference in means')
plots.title('Permutation Test')
print("Observed difference in means: ", obs_diff)
print("Bootstrap empirical P-value: ", p_value)

```

```
In [35]: perm_test_means(iv, 'success', 'Group', 5000 )
```

```
Observed difference in means: 0.169904438577
```

```
Bootstrap empirical P-value: 0.0
```



Example 3. Was there a difference between the Patriots’ footballs and the Colts’? If so, why?

On January 18, 2015, the Indianapolis Colts and the New England Patriots played the American Football Conference (AFC) championship game to determine which of those teams would play in the Super Bowl. After the game, there were allegations that the Patriots’ footballs had not been inflated as much as the regulations required; they were softer. This could be an advantage, as softer balls might be easier to catch.

For several weeks, the world of American football was consumed by accusations, denials, theories, and suspicions: the press labeled the topic Deflategate, after the Watergate political scandal of the 1970’s. The National Football League (NFL) commissioned an independent analysis. In this example, we will perform our own analysis of the data.

Pressure is often measured in pounds per square inch (psi). NFL rules stipulate that game balls must be inflated to have pressures in the range 12.5 psi and 13.5 psi. Each team plays with 12 balls. Teams have the responsibility of maintaining the pressure in their own footballs, but game officials inspect the balls. Before the start of the AFC game, all the Patriots’ balls were at about 12.5 psi. Most of the Colts’ balls were at about 13.0 psi. However, these pre-game data were not recorded.

During the second quarter, the Colts intercepted a Patriots ball. On the sidelines, they measured the pressure of the ball and determined that it was below the 12.5 psi threshold. Promptly, they informed officials.

At half-time, all the game balls were collected for inspection. Two officials, Clete Blakeman and Dyrrol Prioleau, measured the pressure in each of the balls. Here are the data; pressure is measured in psi. The Patriots ball that had been intercepted by the Colts was not inspected at half-time. Nor were most of the Colts’ balls – the officials simply ran out of time and had to relinquish the balls for the start of second half play.

```
In [22]: football = Table.read_table('football.csv')
        football.show()
```

<IPython.core.display.HTML object>

For each of the 15 balls that were inspected, the two officials got different results. It is not uncommon that repeated measurements on the same object yield different results, especially when the measurements are performed by different people. So we will assign to each the ball the average of the two measurements made on that ball.

```
In [23]: football['Combined'] = (football['Blakeman']+football['Prioleau'])/2
        football.show()
```

<IPython.core.display.HTML object>

At a glance, it seems apparent that the Patriots' footballs were at a lower pressure than the Colts' balls. Because some deflation is normal during the course of a game, the independent analysts decided to calculate the drop in pressure from the start of the game. Recall that the Patriots' balls had all started out at about 12.5 psi, and the Colts' balls at about 13.0 psi. Therefore the drop in pressure for the Patriots' balls was computed as 12.5 minus the pressure at half-time, and the drop in pressure for the Colts' balls was 13.0 minus the pressure at half-time.

```
In [24]: football['Drop'] = np.array([12.5]*11 + [13.0]*4) - football['Combined']
        football.show()
```

<IPython.core.display.HTML object>

At a glance, it is apparent that the drop was larger, on average, for the Patriots' footballs. Could the difference be just due to chance?

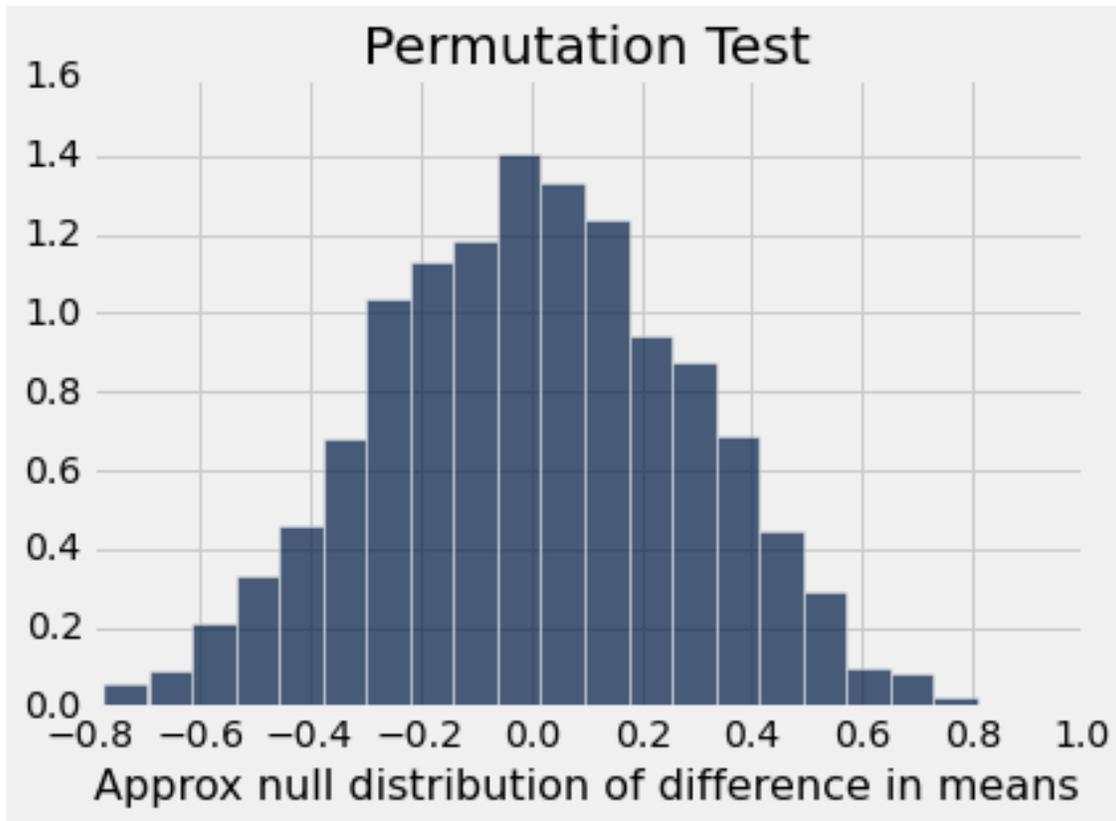
To answer this, we must first examine how chance might enter the analysis. This is not a situation in which there is a random sample of data from a large population. It is also not clear how to create a justifiable abstract chance model, as the balls were all different, inflated by different people, and maintained under different conditions.

One way to introduce chances is to ask whether the pressures of the 11 Patriots balls and the 4 Colts balls resemble a random permutation of the 15 pressures. For example, if we took a random permutation of the 15 pressures, how likely is it that the difference in the means of the first 11 and the last 4 would be as large as the difference observed by the officials?

This can be answered easily by a permutation test for the difference in means.

```
In [25]: perm_test_means(football, 'Drop', 'Team', 4000)
```

```
Observed difference in means: 0.733522727273
Bootstrap empirical P-value: 0.00375
```



The observed difference was roughly 0.7335 psi. According to the empirical distribution above, there is a very small chance that a random permutation would yield a difference that large. So the data support the conclusion that the two groups of pressures were not like a random permutation of all 15 pressures.

The independent investigative team analyzed the data in several different ways, taking into account the laws of physics. The final report said,

“[T]he average pressure drop of the Patriots game balls exceeded the average pressure drop of the Colts balls by 0.45 to 1.02 psi, depending on various possible assumptions regarding the gauges used, and assuming an initial pressure of 12.5 psi for the Patriots balls and 13.0 for the Colts balls.”

– *Investigative report commissioned by the NFL regarding the AFC Championship game on January 18, 2015*

Our analysis shows an average pressure drop of about 0.73 psi, which is consistent with the official analysis.

The all-important question in the football world was whether the excess drop of pressure in the Patriots’ footballs was deliberate. To that question, the data have no answer. If you are curious about the answer given by the investigators, here is the [full report](#).

In []: