Looking for a difference between two systems

Welcome to lab 5.5! In this lab we are going to practice using simulation to determine whether or not two sets of data came from similar or different systems.

In [1]:
```
# Run this cell to set up the notebook, but please don't change it.

# These lines import the Numpy and Datascience modules.
import numpy as np
from datascience import *

# These lines do some fancy plotting magic.
import matplotlib
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
%matplotlib inline
import warnings
warnings.simplefilter('ignore', FutureWarning)
```

1. Basketball shooting styles

In this section, we are going to look at data comparing two different styles of shooting a basket in basketball. One is the style you are familiar with if you watch basketball. The second is the "granny shot" which can be seen in the following video.

In [2]:
```
from IPython.display import YouTubeVideo
# The original URL is:
#   https://www.youtube.com/watch?v=B52aqaOUH-c
YouTubeVideo("B52aqaOUH-c")
```

Out[2]:

To explore whether the granny shot improves shooting, we are going to start by looking at the "null" situation; in other words, we will start by looking at data where we know nothing is different between the two shooting systems.

Life under the null hypothesis

Suppose someone is normally a 70% free throw shooter. We can represent their shooting percentage after taking 20 shots with the following code:

In [3]:
```
shooting_percentage = sample_proportions(20, make_array(0.7,0.3))
shooting_percentage
```

Out[3]:
```
array([ 0.85,  0.15])
```

Question 1

Imagine two players who are both 70% free throw shooters go out 10 times each and shoot 20 free throws. Make a table with one column for each player that reports their results.

In [4]:
```
## player A
player_A = make_array()
for i in np.arange(10):
    shooting_percentage_1time = sample_proportions(20, make_array(0.7,0.3))
    player_A = np.append(player_A, shooting_percentage_1time.item(0))

## player B
player_B = make_array()
for i in np.arange(10):
    shooting_percentage_1time = sample_proportions(20, make_array(0.7,0.3))
    player_B = np.append(player_B, shooting_percentage_1time.item(0))

results_table = Table().with_columns("Player A", player_A,
                                     "Player B", player_B)
results_table
```

Out[4]:

Player A	Player B
0.6	0.55
0.65	0.75
0.6	0.7
0.75	0.9
0.75	0.8
0.8	0.55
0.5	0.55
0.8	0.8
0.75	0.9
0.45	0.7

Question 2

Does one player have a higher overall average than the other? Is this surprising?

In [5]:
```
## Write your code here
avg_playerA = np.average(results_table["Player A"])
```

```
avg_playerB = np.average(results_table["Player B"])
difference = avg_playerA - avg_playerB
difference
```

Out[5]: -0.055000000000000049

One player have a higher overall average than the other player, but this is not surprising. Since the shooting percentage is based on randomness, we can have different average between the two players even when they are both 70% free throw shooters.

Question 3

Please write a function that does the following:

1. Takes the 20 free-throw percentages in the table above and randomly assigns them between the two players.
2. Calculates the difference in overall average for the randomized data.

In [24]:
```
def random_diff(table):
    append_array = np.concatenate([table[0], table[1]])
    shuffled = np.random.choice(append_array, 20, replace=False)
    player_A = shuffled[0:int(len(shuffled)/2)]
    avg_playerA = np.average(player_A)
    player_B = shuffled[int(len(shuffled)/2):]
    avg_playerB = np.average(player_B)
    return avg_playerA - avg_playerB

random_diff(results_table)
```

Out[24]: -0.025000000000000022

Question 4

Based on what you know about how these data were generated, what do you expect about the differences in overall average? Should they be smaller, larger,or similar to what you saw in Question 2?

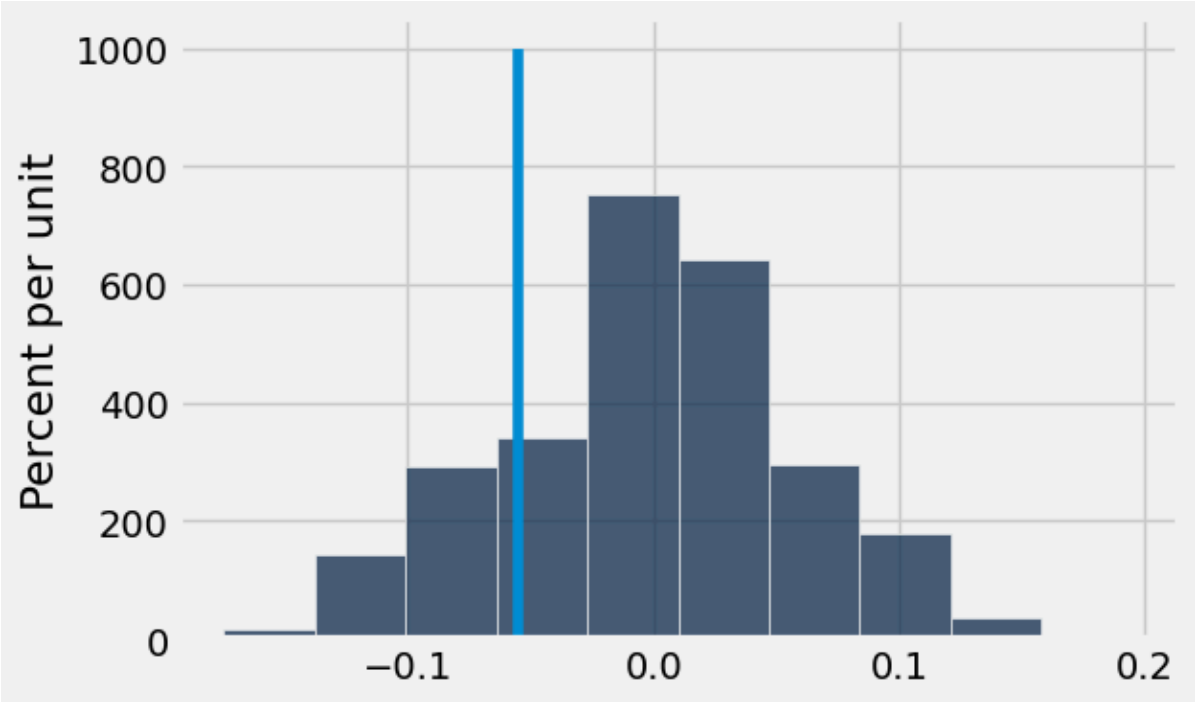
The difference would change every time we simulate the function. They should have similar differeces in overall average to the overall average in question 2.

Question 5

Use the function from Question 3 to generate a 1000 randomized differences. Create a histogram of these differences and compare them to the difference you found in Question 2. Is this consistent with what you know about the source of these data?

In [7]:
```
diff_array = make_array()

for i in np.arange(1000):
    differences = random_diff(results_table)
    diff_array = np.append(diff_array, differences)
```

```
Table().with_column("", diff_array).hist()
_ = plt.plot([difference, difference], [0, 10])
```



Real data

The data below represent a player going out ten times and taking 20 regular shots and 20 granny shots. The total number of shots made are reported each time.

In [8]:
```
shooting_data = Table.read_table('/srv/data/DS_113_S23/Labs/Lab_5.5/Free_thr
shooting_data
```

Out[8]:

Regular	"Granny Shot"
14	14
14	11
13	13
14	18
11	18
15	15
10	16
15	18
13	17
11	16

Question 6

Produce a new table that reports the free-throw percentage for each trial and calculate the difference in overall average between the two shooting styles.

In [9]:
```
regular_array = make_array()
granny_array = make_array()
```

```
for i in np.arange(10):
    regular_percent = (shooting_data[0].item(i))/20
    regular_array = np.append(regular_array, regular_percent)
    granny_percent = (shooting_data[1].item(i))/20
    granny_array = np.append(granny_array, granny_percent)

FT_table = Table().with_columns("Regular", regular_array,
                                "Granny", granny_array)
FT_table.show()

## calculate the difference in overall average
avg_regular = np.average(FT_table[0])
avg_granny = np.average(FT_table[1])
difference = avg_regular - avg_granny
difference
```

Regular	Granny
0.7	0.7
0.7	0.55
0.65	0.65
0.7	0.9
0.55	0.9
0.75	0.75
0.5	0.8
0.75	0.9
0.65	0.85
0.55	0.8

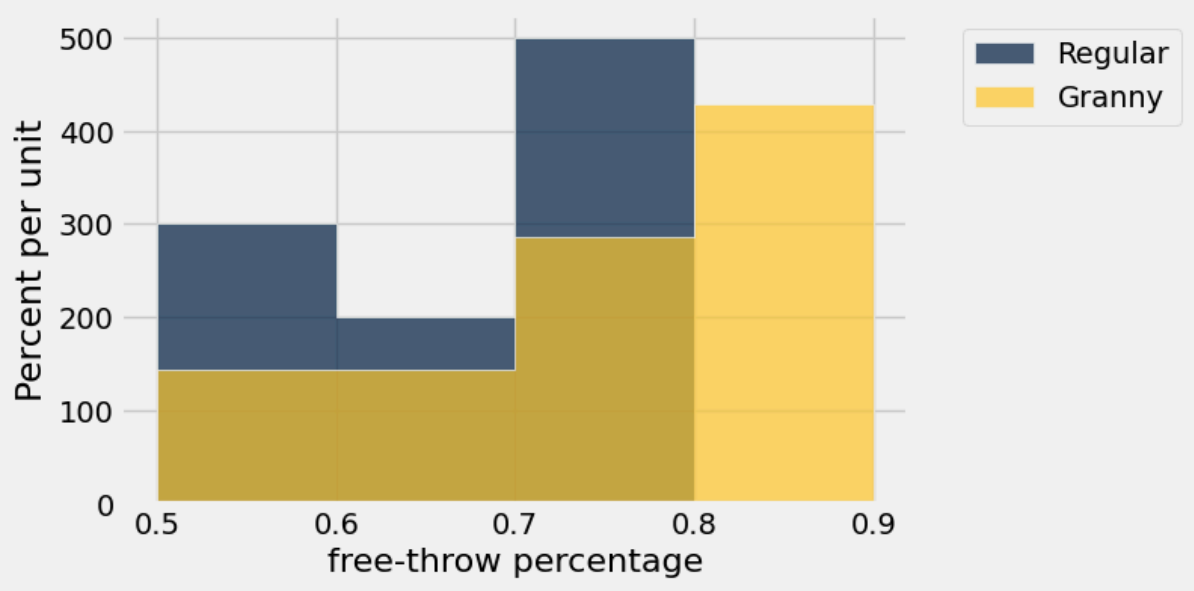
Out[9]:
```
-0.13
```

Question 7

Make two overlaid histograms comparing the two shooting styles. I have provided the bins. Based on the histogram,do you think one style is better than the other?

In [10]:
```
FT_table.hist(bins=np.arange(.5,1,.1))
plt.xlabel('free-throw percentage')
```

Out[10]:
```
Text(0.5, 0, 'free-throw percentage')
```

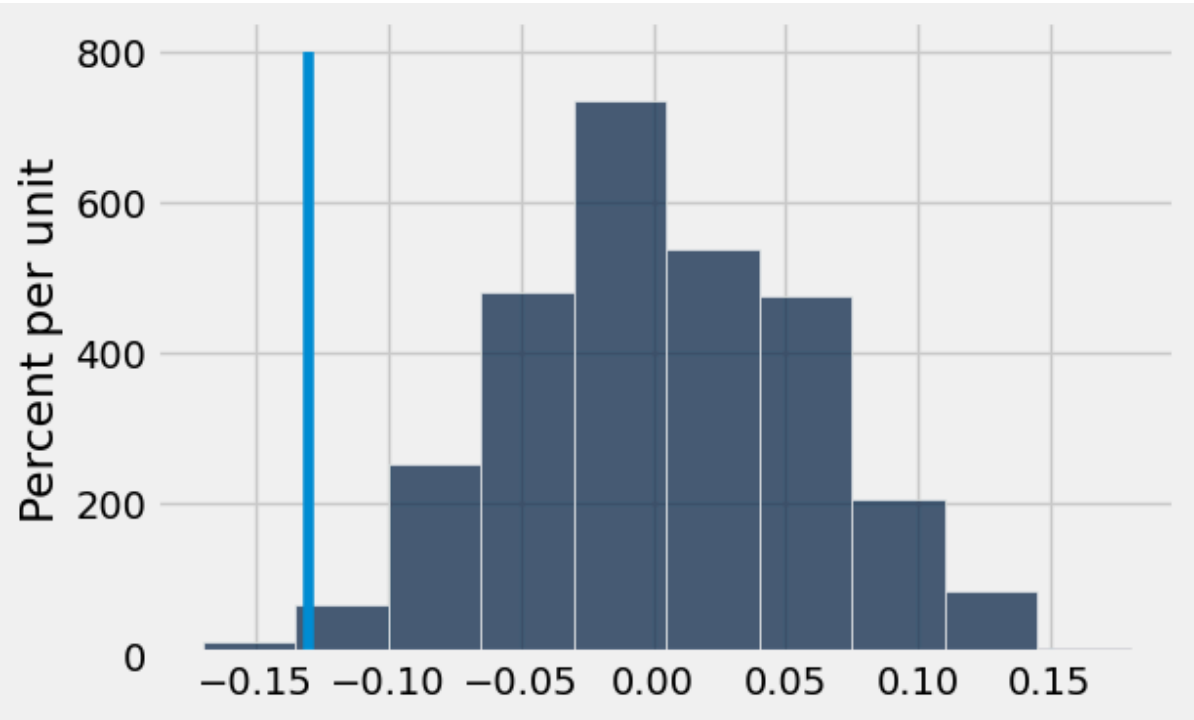


I think the Granny looks better than the regular shoots, because the higher free throw percentage means the player have higher chance to success in throwing the balls.

Question 8

Repeat what you did with the previous data--producing 1000 random shufflings of the data and finding the difference in overall averages. When you are finished, compare to your observed difference from Question 6. What conclusions can you reach?

In [11]:
```
## Code here
diff_array = make_array()

for i in np.arange(1000):
    differences = random_diff(FT_table)
    diff_array = np.append(diff_array, differences)

Table().with_column("", diff_array).hist()
_ = plt.plot([difference, difference], [0, 8])
```



The distribution is centered roughly around 0. This makes sense, because under the null hypothesis the two groups should have roughly the same average. Therefore the difference between the group averages should be around 0.

Question 9

Calcuate a p-value for this analysis. What does this mean?

In [15]:
```
p_value = np.count_nonzero(diff_array >= difference)/1000
p_value
```

Out[15]:
```
0.994
```

The empirical p-value is 0.994, meaning that a lot of the 1,000 permuted samples resulted in a difference of -0.13 or lower. This is only an approximation. According to our simulation, we do not have evidence to reject the null hypothesis.

Different sport, same game

Researchers wishing to test a theory that filling a football with helium would make them fly farther conducted 39 trials ofthe same player kicking one ball with helium and one with regular air. Distances (in yards) are reported in the following file.

In [16]:
```
FB_data = Table.read_table('/srv/data/DS_113_S23/Labs/Lab_5.5/helium.csv')
FB_data
```

Out[16]:

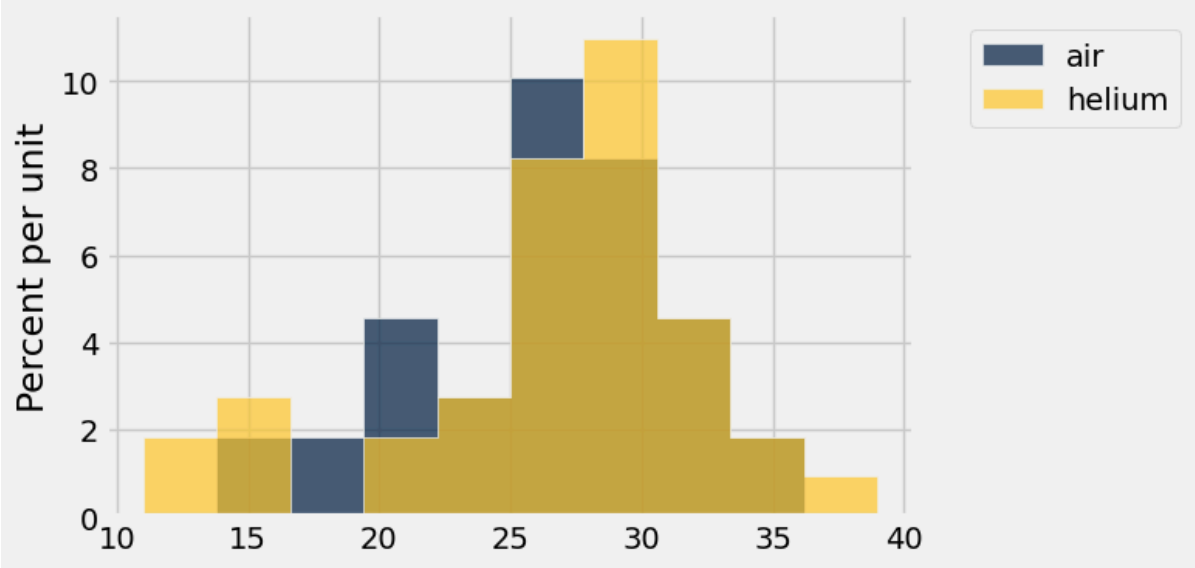
trial	air	helium
1	25	25
2	23	16
3	18	25
4	16	14
5	35	23
6	15	29
7	26	25
8	24	26
9	24	22
10	28	26

... (29 rows omitted)

Question 10

Create an overlaid histogram to compare the two sets of distances and discuss whether you believe there is a difference.

In [18]:
```
FB_data.select(1,2).hist()
```



Although the distances of the two sets are slightly different, I think there is no big difference in the graph as the whole.

Question 11

Write a null and an alternative hypothesis for this situation.

Null hypothesis: The distribution of distances of footballs with air is the same for footballs filled with helium. The difference in the sample is due to chance.

Alternative hypothesis: The footballs filled with helium have a longer distance, on average, than the footballs filled with air.

Question 12

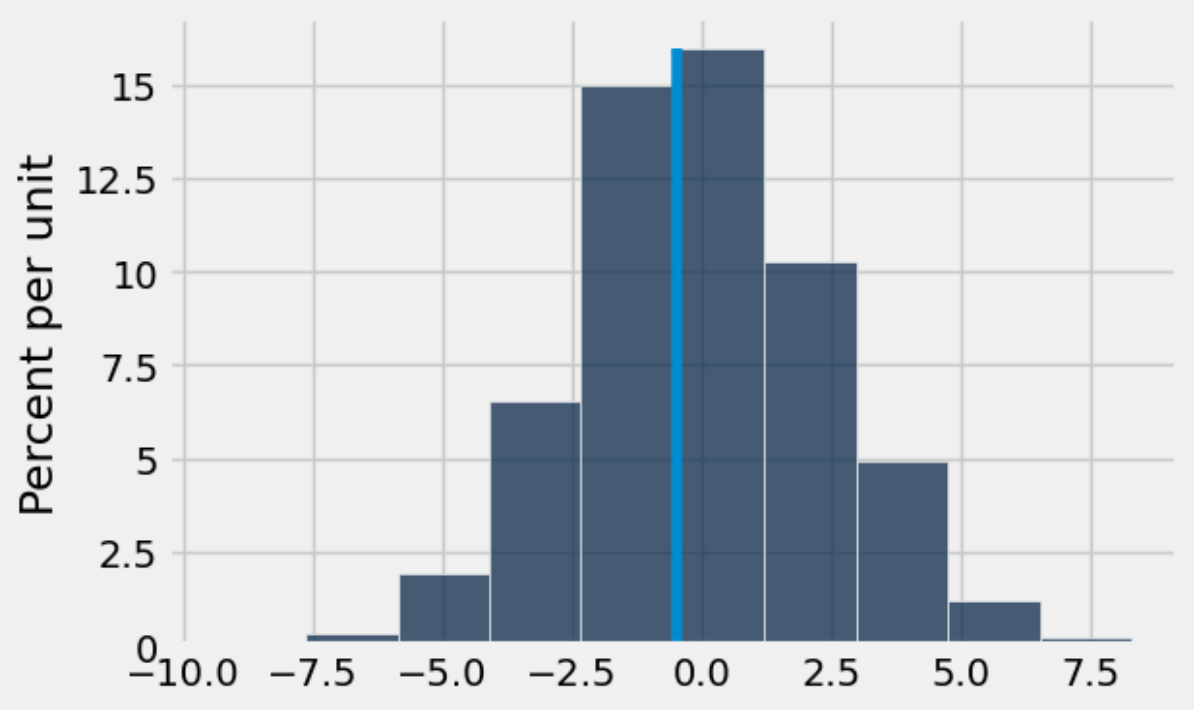
Use a permutation test to decide whether or not the null hypothesis is likely to be true. As part of your work, you should generate a histogram of randomized differences as well as a p-value.

In [37]:
```
## Write your code here.

## calculate the difference in overall average
avg_air = np.average(FB_data[1])
avg_helium = np.average(FB_data[2])
difference = avg_air - avg_helium
print(difference)

## use permutation test
diff_array = make_array()

for i in np.arange(1000):
    differences = random_diff(FB_data.drop(0))
    diff_array = np.append(diff_array, differences)

Table().with_column("", diff_array).hist()
_ = plt.plot([difference, difference], [0, 0.16])

## p-value
```

```
p_value = np.count_nonzero(diff_array >= difference)/1000
p_value
```

```
-0.461538461538
```

Out[37]:
```
0.563
```



Summarize your findings here replacing this text.

The distribution of the histogram is centered roughly around 0. Under the null hypothesis the two groups should have roughly the same average. Therefore the difference between the group averages should be around 0. The p-value is 0.562, meaning that a lot of the 1,000 permuted samples resulted in a difference of -0.462 or lower. This is only an approximation. According to our simulation, we do not have evidence to reject the null hypothesis.