Homework 10: Linear Regression

Reading:

- [Prediction (https://www.inferentialthinking.com/chapters/15/prediction.html)](https://www.inferentialthinking.com/chapters/15/prediction.html)

Please complete this notebook by filling in the cells provided. Before you begin, execute the following cell to load the provided tests. Each time you start your server, you will need to execute this cell again to load the tests.

Homework 10 is due **Thursday, 4/16 at 11:59pm**.

Start early so that you can come to office hours if you're stuck. Late work will not be accepted as per the course policies.

Directly sharing answers is not okay, but discussing problems with the course staff or with other students is encouraged. Refer to the policies page to learn more about how to learn cooperatively.

In [ ]:

```
# Don't change this cell; just run it.

import numpy as np
from datascience import *

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

1. Triple Jump Distances vs. Vertical Jump Heights

Does skill in one sport imply skill in a related sport? The answer might be different for different activities. Let us find out whether it's true for the triple jump (https://en.wikipedia.org/wiki/Triple_jump) (a horizontal jump similar to a long jump) and the vertical jump. Since we're learning about linear regression, we will look specifically for a *linear* association between skill level in the two sports.

The following data was collected by observing 40 collegiate level soccer players. Each athlete's distances in both jump activities were measured in centimeters. Run the cell below to load the data.

```
In [ ]: # Run this cell to load the data
        jumps = Table.read_table('triple_vertical.csv')
        jumps
```

Question 1.1. Before running a regression, it's important to see what the data look like, because our eyes are good at picking out unusual patterns in data. Draw a scatter plot with the triple jump distances on the horizontal axis and the vertical jump heights on vertical axis **that also shows the regression line**.

See the documentation on `scatter` here (http://data8.org/datascience/_autosummary/datascience.tables.Table.scatter.html#datascience.tables.Table.s for instructions on how to have Python draw the regression line automatically.

```
In [ ]: ...
```

Question 1.2. Does the correlation coefficient `r` look closest to 0, .5, or -.5? Explain.

Write your answer here, replacing this text.

Question 1.3. Create a function called `regression_parameters`. It takes as its argument a table with two columns. The first column is the x-axis, and the second column is the y-axis. It should compute the correlation between the two columns, then compute the slope and intercept of the regression line that predicts the second column from the first, in original units (centimeters). It should return an array with three elements: the correlation coefficient of the two columns, the slope of the regression line, and the intercept of the regression line.

In [ ]:

```
def regression_parameters(t):
    r = ...
    slope = ...
    intercept = ...
    return make_array(r, slope, intercept)

# When your function is finished, the next lines should
# compute the regression line predicting vertical jump
# distances from triple jump distances. Set parameters
# to be the result of calling regression_parameters appropriately.
parameters = ...
print('r:', parameters.item(0), '; slope:', parameters.item(1), '; intercept:', parameters.item(2))
```

Question 1.4. Let's use `parameters` to predict what certain athletes' vertical jump heights would be given their triple jump distances.

The world record for the triple jump distance is 18.29 *meters* by Johnathan Edwards. What's our prediction for what Edwards' vertical jump would be?

Hint: Make sure to convert from meters to centimeters!

In [ ]:

```
triple_record_vert_est = ...
print("Predicted vertical jump distance: {:f} centimeters".format(triple_record_vert_est))
```

Question 1.5. Do you expect this estimate to be accurate within a few centimeters? Why or why not?

Hint: Compare Edwards' triple jump distance to the triple jump distances in `jumps`. Is it relatively similar to the rest of the data?

Write your answer here, replacing this text.

2. Cryptocurrencies

Imagine you're an investor in December 2017. Cryptocurrencies, online currencies backed by secure software, are becoming extremely valuable, and you want in on the action!

The two most valuable cryptocurrencies are Bitcoin (BTC) and Ethereum (ETH). Each one has a dollar price attached to it at any given moment in time. For example, on December 1st, 2017, one BTC costed $10859.56 and one ETH costed $424.64.

You want to predict the price of ETH at some point in time based on the price of BTC. Below, we load (https://www.kaggle.com/jessevent/all-crypto-currencies/data) two tables called `btc` and `eth` . Each has 5 columns:

- `date` , the date
- `open` , the value of the currency at the beginning of the day
- `close` , the value of the currency at the end of the day
- `market` , the market cap or total dollar value invested in the currency
- `day` , the number of days since the start of our data

In [ ]:

```
btc = Table.read_table('btc.csv')
btc
```

In [ ]:

```
eth = Table.read_table('eth.csv')
eth
```

Question 2.1. In the cell below, make one or two plots to investigate the opening prices of BTC and ETH as a function of time. Then comment on whether you think the values roughly move together.

In [ ]:

```
...
```

Write your answer here, replacing this text.

Question 2.2. Now, calculate the correlation coefficient between the opening prices of BTC and ETH.

Hint: It may be helpful to define and use the function `std_units` .

In [ ]:

```
def std_units(arr):
    ...

standard_btc = ...
standard_eth = ...

r = ...
r
```

Question 2.3. Regardless of your conclusions above, write a function `eth_predictor` which takes an opening BTC price and predicts the price of ETH. Again, it will be helpful to use the function `regression_parameters` that you defined earlier in this homework.

Note: Make sure that your `eth_predictor` is using linear regression.

In [ ]:

```
def eth_predictor(btc_price):
    parameters = ...
    slope = ...
    intercept = ...
    ...
```

Question 2.4. Now, using the `eth_predictor` you defined in the previous question, make a scatter plot with BTC prices along the x-axis and both real and predicted ETH prices along the y-axis. The color of the dots for the real ETH prices should be different from the color for the predicted ETH prices.

Hints:

- An example of such a scatter plot is generated [here. (https://www.inferentialthinking.com/chapters/15/2/regression-line.html)](https://www.inferentialthinking.com/chapters/15/2/regression-line.html)
- Think about the table that must be produced and used to generate this scatter plot. What data should the columns represent? Based on the data that you need, how many columns should be present in this table? Also, what should each row represent? Constructing the table will be the main part of this question; once you have this table, generating the scatter plot should be straightforward as usual.

In [ ]:

```
...
```

Question 2.5. Considering the shape of the scatter plot of the true data, is the model we used reasonable? If so, what features or characteristics make this model reasonable? If not, what features or characteristics make it unreasonable?

Write your answer here, replacing this text.

Question 2.6. Now suppose you want to go the other way: to predict a BTC price given an ETH price. What would the regression parameters of this linear model be? How do they compare to the regression parameters from the model where you were predicting ETH price given a BTC price? Set `regression_changes` to an array of 3 elements, with each element corresponding to whether or not the corresponding item returned by `regression_parameters` changes when switching BTC and ETH as x and y. For example, if r changes, the slope changes, but the intercept wouldn't change, the array would be `[True, True, False]`

```
In [ ]: regression_changes = ...
        regression_changes
```

3. Evaluating NBA Game Predictions

A brief introduction to sports betting

In a basketball game, each team scores some number of points. Conventionally, the team playing at its own arena is called the "home team," and the other team is called the "away team." The winner is the team with more points.

We can summarize what happened in a game by the "**outcome**", defined as the **the away team's score minus the home team's score**:

$$\text{outcome} = \text{points scored by the away team} - \text{points scored by the home team}$$

If this number is positive, the away team won. If it's negative, the home team won.

In order to facilitate betting on games, analysts at casinos try to predict the outcome of the game. This prediction of the outcome is called the **spread.**

```
In [ ]: spreads = Table.read_table("spreads.csv")
        spreads
```

Here's a scatter plot of the outcomes and spreads, with the spreads on the horizontal axis.

```
In [ ]: spreads.scatter("Spread", "Outcome")
```

Question 3.1. Why do you think that the spread and outcome are never 0, aside from 1 case of the spread being 0?

Hint: Read the first paragraph of the Wikipedia article on basketball here (https://en.wikipedia.org/wiki/Basketball) if you're confused!

Write your answer here, replacing this text.

Let's investigate how well the casinos are predicting game outcomes.

One question we can ask is: Is the casino's prediction correct on average? In other words, for every value of the spread, is the average outcome of games assigned that spread equal to the spread? If not, the casino would apparently be making a systematic error in its predictions.

Question 3.2. Among games with a spread between 3.5 and 6.5 (including both 3.5 and 6.5), what was the average outcome?

Hint: Read the documentation for the predicate `are.between_or_equal_to` here (http://data8.org/datascience/predicates.html#datascience.predicates.are.between_or_equal_to).

In [ ]:
```
spreads_around_5 = ...
spread_5_outcome_average = ...
print("Average outcome for spreads around 5:", spread_5_outcome_average)
```

Question 3.3. If the average outcome for games with any given spread turned out to be **exactly** equal to that spread, what would the slope and intercept of the linear regression line be, in original units? Hint: If you're stuck, try drawing a picture!

In [ ]:
```
expected_slope_for_equal_spread = ...
expected_intercept_for_equal_spread = ...
```

Question 3.4. Fix the `standard_units` function below. It should take an array of numbers as its argument and return an array of those numbers in standard units.

In [ ]:

```
def standard_units(nums):
    """Return an array where every value in nums is converted to standard units."""
    return nums/np.std(nums) - np.mean(nums)
```

Question 3.5. Compute the correlation coefficient between outcomes and spreads.

Note: It might be helpful to use the `standard_units` function.

In [ ]:

```
spread_r = ...
spread_r
```

Question 3.6. Compute the slope of the least-squares linear regression line that predicts outcomes from spreads, in original units.

In [ ]:

```
spread_slope = ...
spread_slope
```

Question 3.7. For the "best fit" line that estimates the average outcome from the spread, the slope is less than 1. Does knowing the slope alone tell you whether the average spread was higher than the average outcome? If so, set the variable name below to `True` . If you think you need more information than just the slope of the regression line to answer that question, then respond `False` . Briefly justify your answer below. (HINT: Does the intercept matter?)

In [ ]:

```
slope_implies_average_spread_above_average_outcome = ...
```

Write your answer here, replacing this text.

4. Submission

Once you're finished, submit your assignment as a .ipynb (Jupyter Notebook) and .pdf (download as .html, then print to save as a .pdf) on the class Canvas site.