Module 11: Additional Exercise(s)

JHU EP 606.206 - Introduction to Programming Using Python

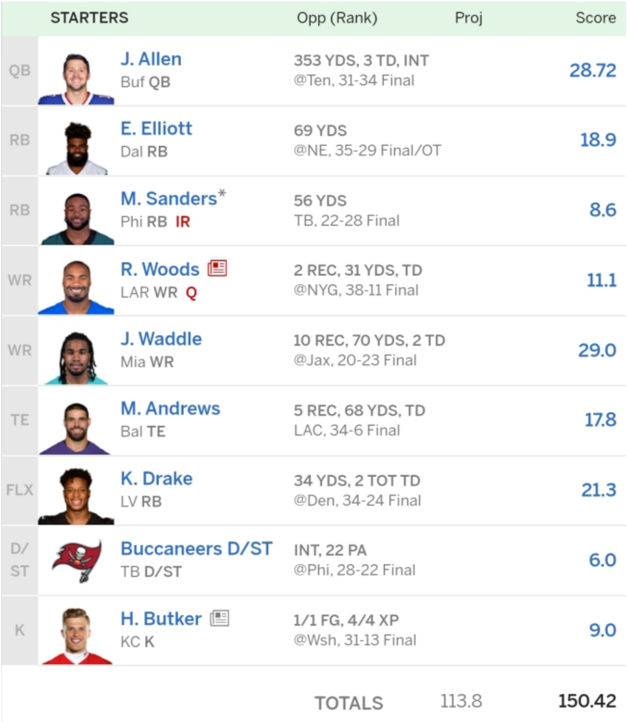
Introduction

Data science and machine learning can be applied to a large number of different types of problems. We can use the frameworks we learned about this week to ask specific questions about a dataset or to make inferences/predictions about future outcomes. This Additional Exercise(s) was conceived during the NFL season (go **Eagles!**, boo **Cowboys/49ers**), so we'll start by using DataFrames to get specific answers from our dataset. Then, we'll use a popular machine learning library, scikit-learn, to make predictions that will allow us to choose a starting QB for our fantasy football line-up.

What is Fantasy Football?

Fantasy Football is a game in which your create fictitious teams based on real National Football League (NFL) players. Each week those players are awarded points for how well they perform in real-life. The sum of the point totals of all members on your team represents your team's score for that week. If your total is greater than your opponent's, you win; if not, you lose. A new game is played against a different opponent each week.

Here is an example of my team's result on ESPN from Week 6 of the 2021 football season:



	STARTERS	Opp (Rank)	Proj	Score
QB	J. Allen Buf QB	353 YDS, 3 TD, INT @Ten, 31-34 Final		28.72
RB	E. Elliott Dal RB	69 YDS @NE, 35-29 Final/OT		18.9
RB	M. Sanders* Phi RB IR	56 YDS TB, 22-28 Final		8.6
WR	R. Woods LAR WR Q	2 REC, 31 YDS, TD @NYG, 38-11 Final		11.1
WR	J. Waddle Mia WR	10 REC, 70 YDS, 2 TD @Jax, 20-23 Final		29.0
TE	M. Andrews Bal TE	5 REC, 68 YDS, TD LAC, 34-6 Final		17.8
FLX	K. Drake LV RB	34 YDS, 2 TOT TD @Den, 34-24 Final		21.3
D/ST	Buccaneers D/ST TB D/ST	INT, 22 PA @Phi, 28-22 Final		6.0
K	H. Butker KC K	1/1 FG, 4/4 XP @Wsh, 31-13 Final		9.0
		TOTALS	113.8	150.42

Additional Exercise #1: DataFrames

For this exercise we'll read in a comma-separated value (CSV) formatted dataset from a file called box_scores.csv which can be found on Blackboard. box_scores.csv contains game data from 2000-2016. From what I can tell the data is somewhat incomplete, but there's no harm in assuming it is comprehensive for our purposes. Your task is to read the data into a DataFrame using **`read_csv()`** and answer the following questions (the roughly estimated difficulty and my answers are provided for reference):

1. **How many total games were played?** (1/5)
   a. `There was a total of 4328 games played.`
2. **What percentage of games were won by the home/away team?** (3/5)
   a. `The home team wins 57.28% of its games and the away team wins 42.72% of its games.`
3. **How many games did the Philadelphia Eagles play?** (2/5)
   a. `The Philadelphia Eagles played 278 games.`
4. **How many wins/losses did the Philadelphia Eagles have?** (4/5)
   a. `The Philadelphia Eagles' record was: 160-118`

Here is a link to our solution.

Additional Exercise #2: scikit-learn

For the second Additional Exercise we'll do some "data wrangling" (manipulating) using DataFrames, convert those DataFrames into NumPy arrays, and use those NumPy arrays to create a machine learning model that makes predictions using a method called linear regression using scikit-learn.

Scenario: we want to understand how the average number of yards per pass by a QB impacts the number of total passing yards they accumulate. QB's are typically awarded 0.1 points for every yard they throw for, so the more yards a QB gains the better.

Approach: we will use the following data fields to build and train our machine learning model:

1. `home_net_yards_passing`: total net number of yards a home team QB throws for
2. `home_yards_per_pass`: average number of yards of each home team pass attempt
3. `away_net_yards_passing`: total net number of yards an away team QB throws for
4. `away_yards_per_pass`: average number of yards of each away team pass attempt

Assumptions:

1. Net yardages belong to a single QB (no QB's were injured/replaced)
2. All yardages per attempt belong to a single QB (no QB's were injured/replaced)

Part #1 – Data Wrangling

1. Create a DataFrame that contains only `home_net_yards_passing` and `home_yards_per_pass` for the home team.
2. Create a DataFrame that contains only `away_net_yards_passing` and `away_yards_per_pass` for the away team.
3. Create a DataFrame that concatenates the DataFrames from Step 1 and Step 1.

Hint: Consider renaming the fields to `yards` and `yards_per_pass` in the merged DataFrame.

Part #2 – Build and Train a Linear Regression Model with scikit-learn

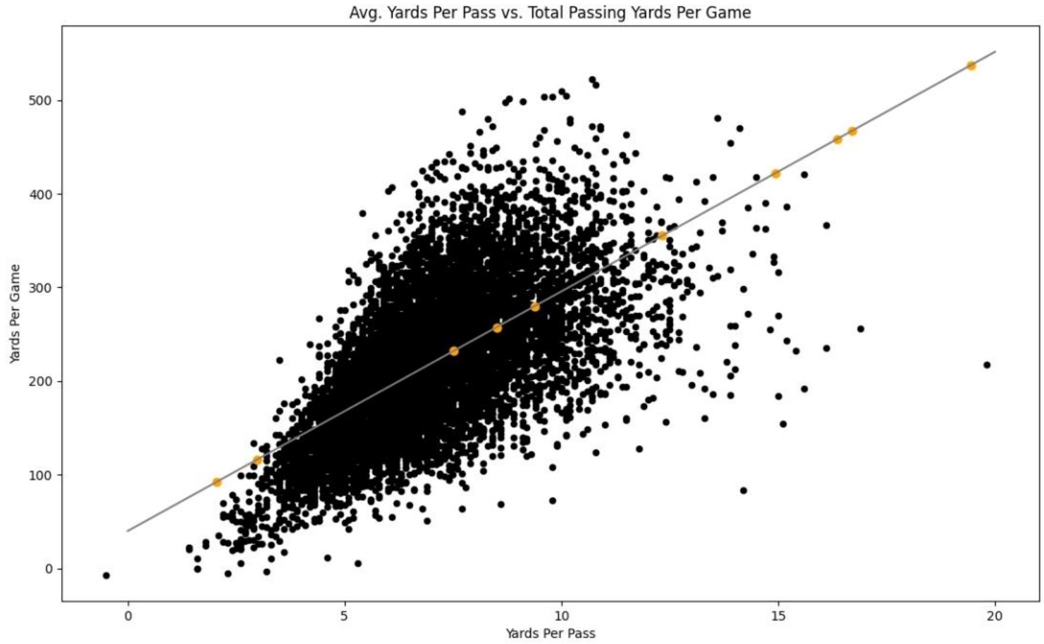
1. Convert the `yards` DataFrame into a NumPy array using `np.array()` and `resize()`
2. Convert the `yards_per_pass` DataFrame into a NumPy array using `np.array()` and `resize()`
3. Create a `LinearRegression()` object using scikit-learn
4. Perform the regression calculation using the `fit()` method
5. Retrieve the y-intercept (alpha) value of the best-fit line from the LinearRegression() object
    a. The `intercept_` instance variable will be useful here
6. Retrieve the slope (beta) value of the best-fit line from the LinearRegression() object
    a. The `coef_` instance variable will be useful here
7. Print the equation of the best-fit line

Part #3 – Use the Trained Model to Make Predictions

1. Create a NumPy array of 10 random QBs (yards/attempt values) using `random.uniform()`
2. Call the `predict()` method belonging to your LinearRegression() object using the resulting NumPy array from Step 1 as the input.
3. Convert the result to a list. This list contains the predictions for how many yards your QB will throw for based on the randomly generated values from Step 1!

Part #4 – Visualize (Plot) the Results

1. Use matplotlib to create a plt object to make a scatterplot of the data:
    a. Call plt.title to give your visualization a title
    b. Call plt.xlabel to give you x-axis a label
    c. Call plt.ylabel to give you y-axis a label
    d. Call plt.scatter using the NumPy array outputs from Part #2 Steps 2, 1 as your inputs
    e. Call plt.scatter using the NumPy array output from Part #3 Step 1 as your input
    f. Call plt.show to display the scatterplot to the console
2. Use your matplotlib plt object to draw the best-fit (regression) line using plt.plot()



Based on the predictions made by our machine learning model, it appears that the QB's who attempt longer passes on average are more likely to throw for more yards in a game and, therefore, most likely to score the most points for our Fantasy Football team. Here is a link to the full solution.