CSE199 Internet and Data Homework 2 Fall 2022

Plotting, Correlation, and Regression

This homework provides a 'get-acquainted' experience with a few elemental data tools and concepts: plotting, correlation, linear regression, and quantitative analysis, using short programs and modules written in Python. No past programming experience in Python is needed and the required portion is shorter than last week's SQL activity. Python can be accessed via a web app in **Chrome** or **Firefox** without having to download and install a Python system (of course, that is fine too). Doing this homework is also useful prep for the recitation activity, in which a different Python program is used to analyze emotional "affect" of web pages.

The homework continues the NFL teams example of last week. What factors are correlated with long playoff win droughts? We will explore • the number of coaching changes since 2010 and • the sizes of the teams' media markets. Is coaching turnover symptomatic or causative of playoff failure? Are small-market teams starved for resources needed to win? Well, this is "small data"—there are only 32 NFL "apples" and we're not considering the "oranges" of other sports—but having just 3 rows of 32 data points allows us to see all of what if anything is going on.

The homework has two parts, aside from the preparation of setting up Python access:

- Play the game guessthecorrelation.com (notice the retro graphics) and take a screenshot confirming a finals score. (33.3%)
- Analyze the project files beginning with `NFL` (besides the official CSE199 repository, they are duplicated at https://www.cse.buffalo.edu/~regan/cse199/) and submit answers to questions given below. (66.7%)

1 Python Setup

The main technical requirement is access to a working Python 3 applet. *The alternative of installing your own Python system is described on the last page.* The "Python 3 Trinket" site https://trinket.io/python3 works best with Chrome or Firefox. For simple Web use, no login or registration is apparently needed:

1. Go to https://trinket.io/python3 ("Python 3 Trinket" free demo level). (Alternate page)
2. Copy and paste the `NFLtest.py` file into the window for "`main.py`"—no need to change the filename or add any new files. (Unlike with SQLizer last week, there is no file upload.)
3. Click the triangle "play" button to run.
4. After it runs, slide the vertical divider bar left to widen the output window.

Troubleshooting: If it works for you, leave it alone—often running it a second time has led to its doing nothing. If it does nothing the first time, *wait* and try again. While waiting, you can try making it load `NFLTeams.xml` directly rather than via HTTP: Hit the '`+`' sign at upper right to add a file, name it `NFLTeams.xml`, and copy and paste the XML file's contents. Then go back into the code, comment-*out* line 95, comment-*in* line 94 to use a simple filename as the location, and try again. *If it keeps twirling at upper right, try my alternate Trinket page*, which is set up for part II but can be used for the NFL part by just pasting that code over `main.py`.

If it really doesn't work (or if the site bogs down from simultaneous use), first try another browser. As a last resort—which you can also use as a first resort—try the alternative of your own installation or running Python on an accessible machine that has it.

2 Other Preparation

Besides playing "Guess the Correlation" to gain some visual context for the NFL data plots, please also read the webpage "Regression with scikit.learn" (which `sklearn` further abbreviates). Be warned that the webpage leaves some Python lines incomplete on purpose and that the loading syntax is a little different from ours.

You're already familiar with the XML format of the supplied data file `NFLTeams.xml` from last week. Download the project code files from the main CSE199 repository or from the `https://www.cse.buffalo.edu/~regan/cse199/` folder. In full they are:

```
NFLtest.py
NFLTeams.xml
```

Please finally skim-read the Python code.

3 Homework

The score is out of 9 pts., which may then be converted to a score out of 3 pts.

First, as stated above, playing "Guess the Correlation" and including a screenshot of your score in what you submit is worth 33%, i.e., 3pts.

The second part actually runs in one shot—nothing more to do. Please, however, note and remark on the following:

1. What factors might *cause*, *affect*, and/or be *symptomatic* of Y = long playoff droughts?

2. One might be Z = the size of the local media market. The Bills are a small-market team. (Unless, that is, we can appeal to Toronto, which in-toto would jump the 2.9 million figure to over 12 million.)

3. Another could be X = high coaching turnover. It might not just be a symptom of losing seasons. Frequent changes could upset the "team chemistry" needed to win.

4. Would you expect X to be strongly correlated with Y?

5. The first linear regression tests Y against X. It gives numbers s and i (for *slope* and *intercept*) that can be interpreted as *projecting*

$$y = i + s \cdot x,$$

where x is the number of coaches any one team has had (since 1990) and y is the *predicted* playoff drought.

6. Look at the s and i for the first regression (drought vs. coaches) at the bottom. Do they make sense? Is it weird for i to be negative? (Well, for the Patriots, the whole drought is zero.)

7. Look at the R^2 figure printed at the end for the "number of coaches since 2010" run. Also look at the printed plot (the crude ASCII plot you can scroll-up for is good enough; if your system allows making a proper `matplotlib` plot, all the better) and compare with your "Guess the Correlation" experience. Would you say the correlation is strong?

8. Now examine the second regression, Y versus Z. Does it show a strong effect? any effect?

9. Examine and comment on the second plot too.

10. Finally, the third and last line shows the results of regressing Y against both X and Z. That is, it predicts

$$y = i + s_1 \cdot x + s_2 \cdot z,$$

with two "slope" coefficients. How did the respective slopes and the R^2 score change?

For a quick further experiment, look at the data point for the Detroit Lions. They have not won a playoff game since **1991**, which is two decades outside our since-2010 range. Change the code to load `http://www.cse.buffalo.edu/ regan/cse199/NFLTeamsNoDET.xml` or comment out the DET line in your copy of the `NFLTeams.xml` file. (To do the latter, you need only change the initial `<` to `<!--` and the final `/>` to `/-->` as exemplified by other lines in the file.) Re-run. How much do the s, i, and especially the R^2 figures change?

What to Submit: The "Guess the Correlation" screenshot (3 pts.) and a "lab report" on the Python run (6 pts.) The items and questions above are intended to structure your report and provoke some thought; you do not need to number your answers the same way.

Alternative Option: Setting up your own Python system

It may ultimately be worth your while to download and install a complete Python3 system such as Anaconda (https://www.anaconda.com/distribution/) and/or learn to use `python3` on the CSE undergraduate machines if you have an account for CSE115 or another course.

The libraries `html, re, sys, traceback, urllib, xml` and their child packages are standard in Python 3 (3.4 or later). The ones to check, in order of need, are `numpy`, `sklearn`, `pandas`, `matplotlib`, and `scipy`. The last two already have work-arounds in place so are not required for this activity; `numpy` is virtually standard and `pandas` can be worked around. So the onus falls on `sklearn` for linear regression—the alternatives in `scipy` or the less-common `statsmodels` packages are more complicated so please verify your access to `sklearn` beforehand. The lines invoking the problematic five packages all work in Python 3 Trinket and on the CSE machines:

```
import numpy as np
import scipy
import mathplotlib
import pandas as pd
from sklearn import linear_model
```

We've put them all in the `NFLtest.py` file so you can get a one-shot test of everything you'll probably need for other courses as well. The installation should take care of all the system paths you need so that placing `NFLtest.py` in the base folder for code and entering `python NFLtest.py` in a command window is all you need do.

For the third option, if you are able to get an account on the CSE machines, you can create a folder `CSE199` in your home directory for work. It's simplest if you first go there and enter

```
cp ~regan/cse199/NFLtest.py .
```

to copy the file over. Then enter

```
python3 NFLtest.py
```

Or you may first load the Python 3 environment directly—as is also possible on the CSE machines by typing just `python3`. In that case, you can both load and run the file by entering (at Python's own prompt which might be `>>>`):

```
from NFLtest import *
```

Once loaded, repeating that command has no effect, but you can re-run by re-pasting lines of the top-level code under the comment `# main` beginning at or after the creation of arrays called `X`, `Y`, and `Z`. And/or, you can change those lines to use different formulas...

The CSE machines will not show the `matplotlib` color plots, nor might your home system, but both will save them as PNG pictures `XYplot.png` and `ZYplot.png` for separate viewing. My code also prints crude ASCII plots of the data points, though not the regression lines. The points are enough to get the "point," hehe. Python 3 Trinket *does* show the PNG pictures—a pleasant feature that usually takes a separate "notebook" setup to see.