Lab 10: Regression

Part 0: Submission/Reference Materials

Remember to read (and write!) good documentation and use the internet to find code examples. Finding and using appropriate/accurate reference materials is hard to teach directly, but tends to be what separates bad scientific programmers from good ones!

If you get stuck:

- Review the lecture material
- Check the docstrings of functions you are trying to use (hint: use ?function in jupyter)
- The pandas official documentation includes all of the commands you need to complete this week's practical https://pandas.pydata.org/docs/user_guide/index.html#user-guide
- Use online resources like stackoverflow, w3schools, realpython etc. (but make sure you don't blindly copy code without working out HOW it works).
- Looking up materials is totally fine but remember if you copy code (or autocomplete it) directly from any source, you MUST cite where you got it from in a comment next to the code.

Submit this assignment as a formatted notebook - include an explanation of your answers and make sure every function has a clear docstring that explains what it does, its arguments, and what it returns. This will be part of the grading of each of your answers.

Part 0: Parse the coffee cooling dataset

We will use the numpy function loadtxt to load our text file into a numpy array. The documentation for `np.loadtxt` can be found here.

Assuming the data is in a file called "my_data.txt", which has two columns of floats, separated by whitespace, then we can load the data into and x and y numpy array like this:

```
data = np.loadtxt("my_data.txt")
x = data[:,0]     # take all the rows of column 0
y = data[:,1]
```

Download the data file coffee_temps.txt from the course website using this link:

https://maguire-lab.github.io/scientific_computing/static_files/practicals/coffee_temps.txt

The file coffee_temps.txt holds the time and temperature data for a cup of coffee cooling.

Q1 [2 points]: Write a function which takes in a filepath to a 2-column whitespaced separated text file and using `np.loadtxt` returns an `x` and `y` numpy array. Show this function works by using it to read the the downloaded coffee_temps.txt file

Q2 [4 points]: Using matplotlib.pyplot (`import matplotlib.pyplot as plt`) create a scatter plot of the x and y arrays you just created with red dots representing each data point and appropriate x and y-axis labels. By inspecting the plot manually try to approximate the intercept and slope of a best fitting straight line.

Part 1: Linear Regression

A model of coffee cooling in a room at 22°C is

$$\Delta T = T_{n+1} - T_n = -k(T_n - T_r)$$

Where T_r is the room temperature (i.e., 22°C), T_n is the temperature at time point n and T_{n+1} is the temperature at the n+1 time point.

Q3 [4 points]: Using both scipy's `linregress` function and numpy's `polyfit` function fit a straight line to the coffee temp data using y as the temperature and x as the time interval.

Compare the intercept and slope parameter inferred using these methods with your estimate from Q2.

```
import scipy.stats as scs
import numpy as np
scs.linregress
np.polyfit
```

Your code should print out the intercept and slope for each method (visual estimation, scipy and numpy).

Q4 [2 points]: Using `matplotlib.pyplot` make a plot of the 3 regression lines. Use title, a different colour/style and label for each line and use the legend function to create a legend explaining what each line represents.

Q5 [4 points]: Using 1 of these regression models - calculate the residuals (distance of actual y values from those predicted using the regression model for a given x-value). Create a scatterplot of these residuals for different values of x. Do these residuals look randomly distributed or is there a pattern in them? You can distinguish the residuals on the plot using plt.axhline

Q6 [6 points]: Using a bootstrapping approach and `scipy.linregress` calculate the 95% confidence intervals for the slope and intercept coefficients for a linear regression of the coffee data. (*Hint: look at last week's practical and lecture if you are unsure how to do this*).

Part 2: Polynomial Regression

Q7 [4 points]: Implement a function for polynomial regression that fits polynomials of different degrees (2, 3, and 4) to the coffee cooling data using `np.polyfit`. Compare the residual sum of squares for each of these models to 1 of the linear models.