Homework 1

Problem 0

Install R and Python on your computer. (There is nothing to turn in associated with this problem.)

More details: In this class we will use R and Python to perform statistical analyses of data. R is a high-level programming language specifically designed for statistical computing and comes with many statistical tools built in. Python is probably the most popular language for machine learning. It is a general-purpose language and requires installing additional packages for statistics and machine learning features. For many assignments you can choose whichever you prefer, but you will be expected to use both at times throughout the course and should begin by installing both now. Here is how to do that:

- Installing Python via Anaconda (recommended). Anaconda is a package and environment manager for Python with a simple graphical interface. Download and install Anaconda Individual Edition and run the installer. On Windows, this will install Anaconda Navigator (a graphical user interface) and the Anaconda Powershell Prompt (a command line interface).
  - Once installed, launch Anaconda Navigator. You will see a set of tiles for various data science programs (most of which we will not use) in the main screen and tabs for "Home", "Environments", "Learning", and "Community" on the left. "Learning" provides links to some documentation which you may find useful. For now, click on "Environments". You should an environment listed titled "base (root)".
  - A list of all the installed packages is shown on the right. Should you want to install another package at some point, select "Not Installed" from the dropdown menu and then search for the appropriate package name.
- Installing R: your installation of Anaconda may have included R already. If RStudio is among the applications on the Anaconda Navigator homepage, you should be good to go. *Otherwise*, install R manually: Download R and run the installer. You may also want to install RStudio and use it as your R development environment. Familiarize yourself with adding packages (e.g. using the package manager in the lower right of RStudio). Base R can do most things for this course, but you may find that additional packages make certain tasks easier. In particular, you should make sure an R kernel for Jupyter is installed.
- Installing Python manually (not recommended for beginners): If you already use Python, have it installed, and are comfortable managing Python packages, make sure that `numpy`, `pandas`, `matplotlib`, `scipy`, and `statsmodels` are installed. You should also be set up to create Jupyter notebooks.

Generally I expect that you are capable of figuring out how to use different software packages and how to solve problems with them when they arise. Your first resource is official documentation and the internet (stackexchange, et cetera). There is far more expertise there than I can provide!

Submission: If working in python, your homework submission should take the form of a jupyter notebook (`.ipynb` file) or python script (`.py` file). If working in R, either an R notebook/markdown file (`.rmd`) or an R script (`.R` file) is acceptable. Unless otherwise specified, submit only one file per assignment (for this one you will submit two files). I need to be able to run the code you submit and see as output the results asked for in each problem. Thus, before submitting your answers, I recommend clearing all workspace variables and then running the entire notebook/script from start to finish. Make sure that everything executes without error and that exactly the output you wish to submit is produced.

- Creating a Jupyter notebook: In Anaconda Navigator, select the home tab. Look for the "Jupyter Notebook" tile and click "Launch". A web browser will open at your home file directory. Navigate to the directory you would like to work in, then in the upper-right click `New -> Notebook: Python 3`. If you have not worked with Jupyter notebooks before, there is a nice introductory tutorial here.
- Creating an R script: Open RStudio. Go to `File -> New File -> R Script`. Alternatively, create an R markdown document by going to `File -> New File -> R Markdown`.

Problem 1 - Basic Analysis in Python

> Perform this problem in python. You should submit your analysis as a Jupyter notebook or python script. Executing this notebook/script from start to finish should generate the outputs as stated in each sub-problem. Write any explanations in markdown cells (if using a notebook) or in code comment blocks (if using a script).

The heart disease study we have been considering in the video lectures includes two additional datasets: one collected at hostpitals in Hungary and one in Switzerland. The goal of this problem is to repeat the basic analysis on the Hungarian portion of the data. Start by downloading the `datasets.zip` file and extracting its contents. For this problem we are interested in the data file `processed.hungarian.data` and the dataset description `heart-disease.names`.

1A - Preprocessing

Import `processed.hungarian.data` as a dataframe. There are missing values and no column names in the file. Add the column names and convert the "?"s to NaN when importing the CSV.

Convert categorical data columns as needed.

Addtionally, some of the columns are missing many values. Report the number of missing values in each column.

1B - Visualization

Make a histogram, a density plot, a scatter plot, and a box-and-whisker plot for an appropriate variable (or pair of variables) of your choice.

Use the `pairplot` command in Seaborn to plot every pair of variables against each other.

1C - Simple Models

Compute the mean maximum heart rate (`thalach`). Compute the mean among those subjects with heart disease and also among those without. Report the difference between the means.

Fit a simple linear regression model for maximum heart rate as a function of age. Report the slope and intercept from the fitted model.

Make a scatter plot of maximum heart rate versus age with the regression line overlaid.

1D - Two models

Fit two simple linear regression models for maximum heart rate as a function of age. In one, only use the data from patients with heart disease. In the other, only use data from patients without. For each, make a scatter plot with the regression line overlaid (even better - figure out how to create a single plot with both datasets and regression lines, differentiated by color).

Examine the fitted parameter values. How do they differ?

Problem 2 - Basic Analysis in R

> Perform this problem in R. You should submit your analysis as either an R notebook or an R script. Executing this notebook/script from start to finish should generate the outputs as stated in each sub-problem. Write any explanations in markdown (if using a notebook) or in code comment blocks (if using a script).

For this problem we will perform a similar overview analysis of a new data set. Start by reading through the file `wpbc.names`.

2A - Preprocessing

Import `wpbc.data` as a dataframe. Name the columns, handle missing values, and convert any data types as needed.

Report the complete summary statistics of the data frame.

2B

The variables measured with respect to each nucleus measured in this dataset are represented three ways - the mean, the standard error of the mean, and the "worst" of the measurements. Choose a measurement of interest and make histograms of each of the three representations of that measurement.

2C

Make a new data frame with the all of the standard error and "worst" columns removed. Also remove the ID number column. Use the `pairs` command to make a scatterplot of every pair of variables in this smaller data frame. Do any variables appear to have strong relationships? Can you explain this?

2D

Fit a simple linear regression for the "perimeter mean" variable as a function of "radius mean". Make a scatter plot of these two variables with the fitted regression line overlaid. Report the fitted coefficients. Explain these numbers.

Problem 3 - Loops and Conditionals

> Perform this problem in your choice of R or Python. You can include it in the same script/notebook used for an earlier problem.

3A

Use a combination of `for` loop and conditional statements to print every number less than 100 which is evenly divisible by both 3 and 7.

3B

Now use loops and conditionals to find the *sum* of all numbers less than 1,000 which are divisible by either 3 *or* 7. Print this number.