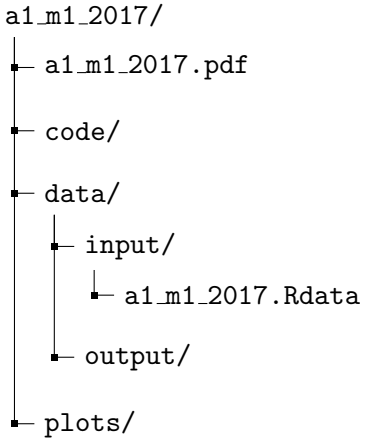
2017 PSS SUMMER SCHOOL
Assignment 4 Data Visualization
Due: 12:30pm on June 30

1 Overview and Submission Instructions

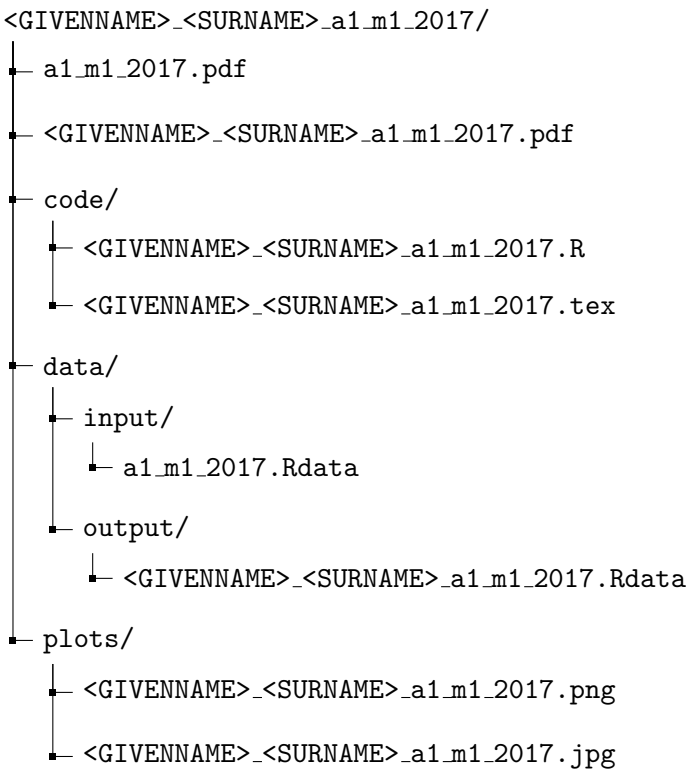
This assignment is centered on visualizations for data. The data manipulation component covered in the last assignment is just one component of the analysis pipeline. After we have the data in the form that we want it is up to our visualizations to tell the story. We will briefly cover various plotting functions that make data visualization simple and elegant. In the end you will acquire experience with all types conventional graphs.

To submit your assignment, email all of your files in a single zip folder to `pss.ss.hw.submit@sp.frb.gov`. Please using the naming convention `<GIVENNAME>_<SURNAME>_a<NUMBER>_m1_2017.<FILEXTENSION>` for any **file that you edit or create**, for files that you simply use without modifying, such as the assignment directions or a data file, leave the name as is, usually something like `a1_m1_2017.pdf`. The below directory comparison shows you an example of the zip directory you will receive and the one you will email submit after completion.

(zip to students)



(zip from students)



Where you make the proper substitutions, e.g. `JUSTIN_SKILLMAN_a3_m1_2017.zip`. Remember that you don't need to change the names of the files you do not modify, such as `a1_m1_2017.Rdata`, only the files that you create or modify. Your code for each assignment example would be saved under `code/` in a `.R` file, named as shown above.

2 Stock Market Data

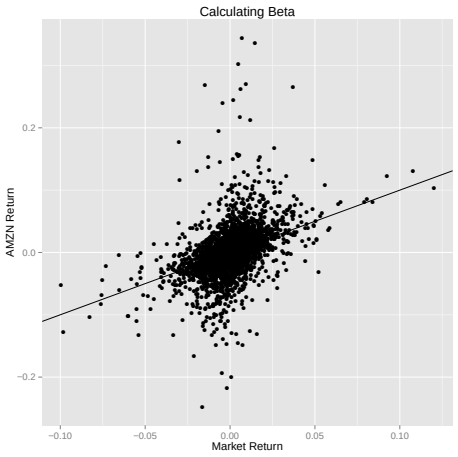
We will be looking at daily stock market trading data for individual stocks in the S&P 500 Index from 2000 to 2015. The stock price was compiled from Yahoo Finance and the sector classification was from wikipedia.

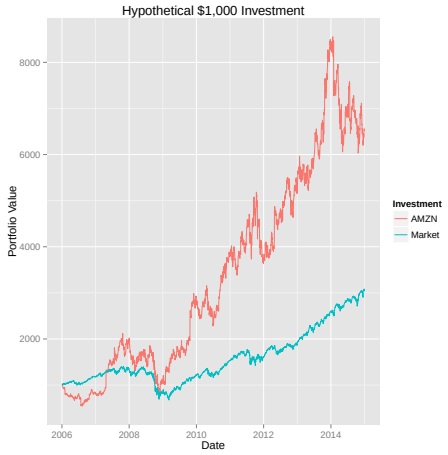
- **trades** - data frame of all trades from 2000 to 2015 for each given stock in the index.
  - *date* - day of the year for closing price
  - *ticker* - symbol for selected stock
  - *price* - closing price of given security on given date
- **desc** - description of each stock by ticker, including full name and industry
  - *ticker* - Symbol for security
  - *fullname* - Full name of company
  - *sector* - Sector of security provided by GICS
  - *subsector* - Sub Industry of security provided by GICS
  - *headquarters* - City which the company headquarter is located
  - *cik* - CIK of security

Problem 2.1. In this exercise we will first be generating a few high level graphs to look at the S&P 500 index as a whole. We will gradually narrow the scope of our analysis to compare an individual stock performance to that of the overall index. In each task your goal is to write code to replicate the shown visualization. Note that, to simplify the exercise, we will not be weighing the stocks by their market capitalizations (which is how the actual index is constructed). Instead we will be assuming that all stocks carry the same weight in the index. Hence the return of the overall index would just be the simple average of the returns of all stocks in the index.

Graph 2.1.1. Market Beta *(Scatter Plot)*

A common measure of how well correlated a given stock's return is to the market is beta. The measure is usually used to analyse the risk a given portfolio has to the market that cannot easily be diversified through the additional of other equities. Here we will calculate beta by fitting a linear trend line on a scatter plot of returns of a given stock to returns of the market, which in this case is the index fund. Save this in your `plot/` directory as `calc_beta.pdf`.



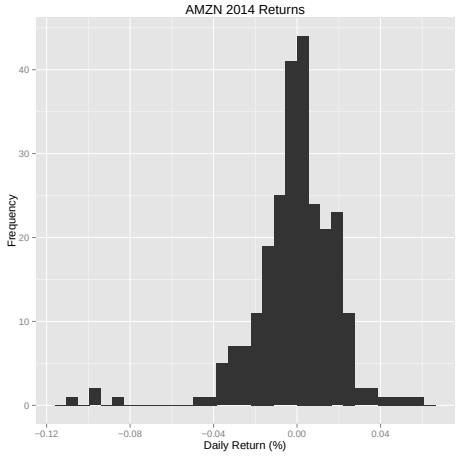


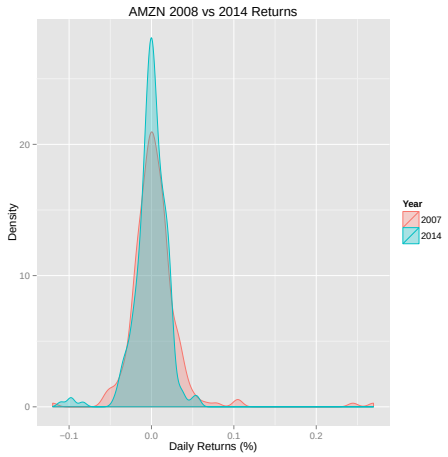
Graph 2.1.2. Hypothetical Portfolio *(Line Graph)*

It is often useful to generate an analysis of the hypothetical. Please choose a stock either or randomly or that you find interesting. In this situation we look at our return at the end of 2014 if we had invested $1,000 in the stock, sector, or index back in 2006. To visualize this we generate a line graph where each line is a different hypothetical situation. This plot should have *three* lines, even though the example here shows only the stock and index. Save this in your `plot/` directory as `hyp_line.pdf`.

Graph 2.1.3. Returns *(Histogram)*

Lets look at the distribution of returns for our stock last year. Plot a histogram of the daily returns for our stock in 2014. Save this in your `plot/` directory as `ret_hist.pdf`.



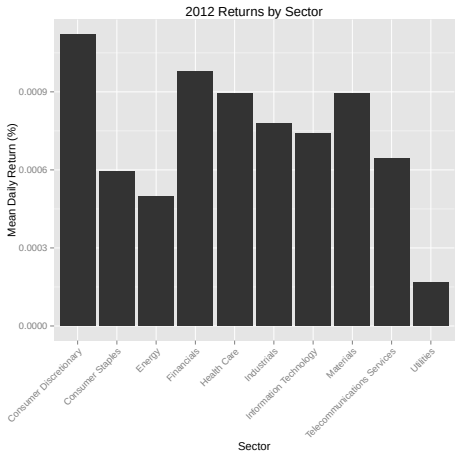


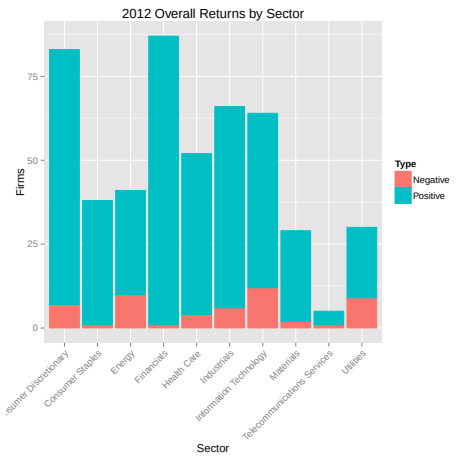
Graph 2.1.4. Distributions *(Kernel Density)*

Let us look at how the distribution of returns differed last year compared to the peak of the recession in 2008. Compare the returns in the years 2008 and 2014. This time we will not create bins for a histogram but instead will be looking at the distribution through kernel density estimates. Plot overlapping kernel density distributions, each representing the distribution of returns for the given year, with enough transparency in the fill such that they are both visible. Do this task for both the stock you have selected but also create a set of kernel density estimate plots that shows one plot for each industry. There are *10* sectors so there will be ten sector graphs. Save each one of these plots in your `plot/` directory as `sector_kern<number>.pdf` where `<number>` is 1:10 respectively.

Graph 2.1.5. Returns by Subsector *(Bar Chart)*

Let us look at how different subsectors compare to each other in terms of their returns. Plot a bar graph that compares the mean daily returns of subsectors to each other for 2014 (2012 shown). Produce a set of plots, one for each sector that compares it's subsectors. Please note that the plot shows sector level returns, you will be doing *subsector*, though they should look similar in form. Again, save successively using numbers appended to `sub_sector_bar<number>.pdf` in your `plot/` directory.





Graph 2.1.6. Positive vs. Negative Year *(Stacked Bar Chart)*

It would be interesting to see the number of stocks in a sector that had a net positive performance (defined as a positive simple mean of daily returns) for the year compared to the proportion that had a net negative. Plot a stacked bar graph with year year as a bar broken up into the proportion of net positive and net negative return stocks. Save this in your `plot/` directory as `sector_posnegs.pdf`.

3 Health Care Data

Problem 3.2. Synthetically generate a dataset on the health of populations in five countries over the period 1900 through 2015. Your data should include variables that have reasonable values, (e.g. age should be within 1:100). You should have observations for a given person throughout their lifetime. For example, if someone was 35 in the 2015, we should observe them for the last 35 years. Do not make everyone exist in all years of the survey, as realistically not everyone in the world would be 100 with no other ages being represented. Think about how each variable might be distributed and choose to sample it accordingly. We also need to pay attention to correlations, those that have genetic pre-dispositions are more likely to be sick. For example, for the *sick* variable (which is binary) you could model it as a Bernoulli r.v. where p is 0.1 for people who are not pre-disposed but 0.5 for those that are. There is no "correct" way to create this data and the relationships among the variables, creativity and logical thinking is valued above anything else here!

- *year* - year survey was taken
- *name* - name of citizen
- *age* - age of citizen
- *gender* - gender of citizen
- *country* - nationality of citizen
- *occupation* - occupation of citizen
- *children* - number of children
- *genetic* - pre-disposed to disease (binary variable)
- *sick* - were they sick at the time of the survey
- *degree* - sickness based on a scale of 1:5
- *haircolor* - hair color of citizen
- *active* - how active a person is 1:10

Once you have a dataset you will need to produce graphics using the following `geom_` objects in ggplot2. The goal is for you to come up with some interesting plots that give a narrative of these populations health, points won't be awarded for nonsensical plots that don't inform the reader. Note that it is often interesting to see the same slice of the data plotted in different ways so don't feel like you can't reuse a few variables here.

1. `geom_density` multiple on the same plot
2. `geom_boxplot`
3. `geom_smooth`
4. `geom_violin` (plot sickness as a function of age)
5. `geom_tile`
6. `geom_ploygon`

Add these plots to a tex document and write a sentence or two about each, describing what it is telling you. Save the plots in your `plot/` directory and name them as you choose, save the `.tex` in your code directory, while the `.pdf` output should be saved in the parent directory.

4 Bonus

This section is intended to challenge those who feel they are up to it. We suggest everyone at the very least attempts these problems as they push you to think hard about complex programming topics that are common in the current research environment. For those that successfully complete these problems, bonus points will be awarded on the assignment. For incomplete, yet worthy attempts at the problems some bonus may be awarded. The answers to these questions need not follow the same structure as above when turning in your solutions, don't append your solutions onto the *answers* list. Merely save your code as a .R file using the naming convention we went over and append `_bonus` at the end of the filename.

Problem 4.3. You are given a matrix of size $n \times n$, each element being an integer from 1:1000 which are randomly sampled. A valid set of choices are elements that are the only ones chosen in their row or column, e.g. you could not choose 2 elements in row five, or 3 elements in column one. Once we have a valid set of choices we take the sum. Your goal is to find which set of choices is the maximum sum. For example, below we have a matrix where the sum of 3325 is the maximum. We can see that $3325 = 863 + 383 + 343 + 969 + 767$. Devise a function that takes in a matrix of of size $n \times n$, where n cannot be > 8 (your function returns an error if so), and solves the maximization problem.

$$\begin{bmatrix} 7 & 53 & 183 & 439 & 863 \\ 497 & 383 & 563 & 79 & 973 \\ 287 & 63 & 343 & 169 & 583 \\ 627 & 343 & 773 & 969 & 943 \\ 767 & 473 & 103 & 699 & 303 \end{bmatrix}$$

The plotting portion of this bonus problems is that you should graph a line plot that shows runtime on the y axis and n on the x axis. Fit a curve to this line, determine if it is linear, exponential, etc. Plot both lines together and save the image as a .pdf. You will turn in both the .pdf and the .R script that include the function.