Read Chapter 4 (Linear Algebra) and Chapter 5 (Statistics). Download the sample code. Review the alternative and additional code presented in class.

A zip archive containing Google Analytics User, Session and Event data by day and by hour between January 1st, 2016 and September 30th, 2017 is included in the Files section in Canvas. You will use the files and data in this zip archive for several of the problems below. As usual, please submit all files and responses for individual questions within a single .zip file.

1. While useful for understanding, computing summary statistics via individual functions can be inefficient if more than one statistic is desired. For instance, we can compute summary statistics for the num_friends data set via individual calls to the respective functions we have defined:

- Count = len(num_friends)
- Sum = sum(num_friends)
- Min = min(num_friends)
- Max = max(num_friends)
- Median = median(num_friends)
- Mode = mode(num_friends)
- Mean = mean(num_friends)
- First Quartile (Q1) = quantile(num_friends,0.25)
- Third Quartile (Q3) = quantile (num_friends,0.75)
- Variance = variance(num_friends)
- Standard Deviation = standard_deviation(num_friends)
- Range = data_range (num_friends)
- InterQuartile Range (IQR) = interquartile_range(num_friends)

Computationally, this results in redundant effort – sorting, for instance. Explain where and why the calls above result in redundant sorting.

2. Write a new stand-alone Python function, SummaryStatistics(v) that takes a numeric list (a vector) of arbitrary length, v, as an argument, and returns a dictionary having keys and values corresponding to the summary statistics above. Use the strings Count, Sum, Min, Max, Median, Mode, Mean, Q1, Q3, Variance, StdDev, Range and IQR as the keys. Do not simply call the functions that we have defined. Include the code for the computations within your new function, and remove the redundant effort described in Exercise 1. Include the source code for your new function as your answer.

3. Using the Google Analytics Data and your function from Exercise 2, compute the summary statistics for:

a. Hourly Users

b. Hourly Sessions

c. Hourly Events

d. Daily Users

e. Daily Sessions

f. Daily Events

Present your results in a table.

4. Look at the Sum data for hourly and daily users in Exercise 3. What is strange about these values? Do you see a similar oddity for the session and event data? Explain why you think this happened.

5. Using the Google Analytics Daily User Data **grouped by month** and your function from Exercise 2, compute the summary statistics for:

a. Daily Users

b. Daily Sessions

c. Daily Events

Present your results in a table/tables. There should be summary statistics for all 21 months for users, sessions and events. I.e., there should be summaries for each of January 2016, February 2016, …, August 2017, September 2017.

6. Describe two interesting observations (patterns) that you see in the summary statistics from Exercise 5.

7. Using the Google Analytics Hourly User Data **grouped by hour of day** and your function from Exercise 2, compute the summary statistics for:

a. Hourly Users

b. Hourly Sessions

c. Hourly Events

Present your results in a table/tables. There should be summary statistics for all 24 hours for users, sessions and events. I.e., there should be summaries for hours 0, 1, 2, …, 23. (Hour 0 represents midnight through 1 AM, etc.).

8. Describe two interesting observations (patterns) that you see in the summary statistics from Exercise 7.

9. A boxplot (box and whisker diagram) shows quartiles and extremes, sometimes including identification of outliers, for sets of data. See http://www.itl.nist.gov/div898/handbook/eda/section3/boxplot.htm (Links to an external site.)Links to an external site.. The plotting libary *matplotlib* that we have used includes capability to create boxplots, including showing multiple boxplots with a common axis. Use this capability to create side-by-side boxplots showing **daily session data** grouped by month, the same data you used for Exercise 5b. Your result should include box and whisker representations of each of the 21 months, shown side-by-side with a common axis so they can be compared. Use the default boxplot option, which should show the quartiles (including the median) as well as outliers for each grouping.

10. Describe two interesting observations (patterns) that you see in the boxplots from Exercise 9.

11. At some point in your career, you may need to create a plot from scratch. We don't have time to approach such a task for a detailed, graphical plot, so in this exercise, you will write code to create a text-based, “Stem and Leaf Plot”. See https://en.wikipedia.org/wiki/Stem-and-leaf_display (Links to an external site.)Links to an external site. for further detail. A stem and leaf plot can be very useful for visualizing certain types of data because it shows a histogram of the data while preserving the individual data elements. Of course, it is of limited use depending on the amount of data and the values of the data. Write a Python function StemAndLeafPlot(v) that takes a numeric list (a vector) of arbitrary length, v, as an argument, and prints a correspond stem and leaf plot to standard output, formatted in a manner similar to that shown in the link above. For this exercise, assume the stems are tens and the leaves are units. Include the source code for your new function as your answer.

12. Using the function you wrote for Exercise 11, create a stem and leaf plot for each of the following:

a. Hourly Session Data for August 6th – August 8th, 2017.

b. Hourly Session Data for August 13th – August 15th, 2017.

c. Hourly Session Data for August 20th – August 22nd, 2017.

d. Hourly Session Data for August 27th – August 29th, 2017.

Include your stem and leaf plots as your answer.

13. Compare your plots from Exercise 12. Does the time period corresponding to the eclipse stand out from the other time periods in these plots? Explain.