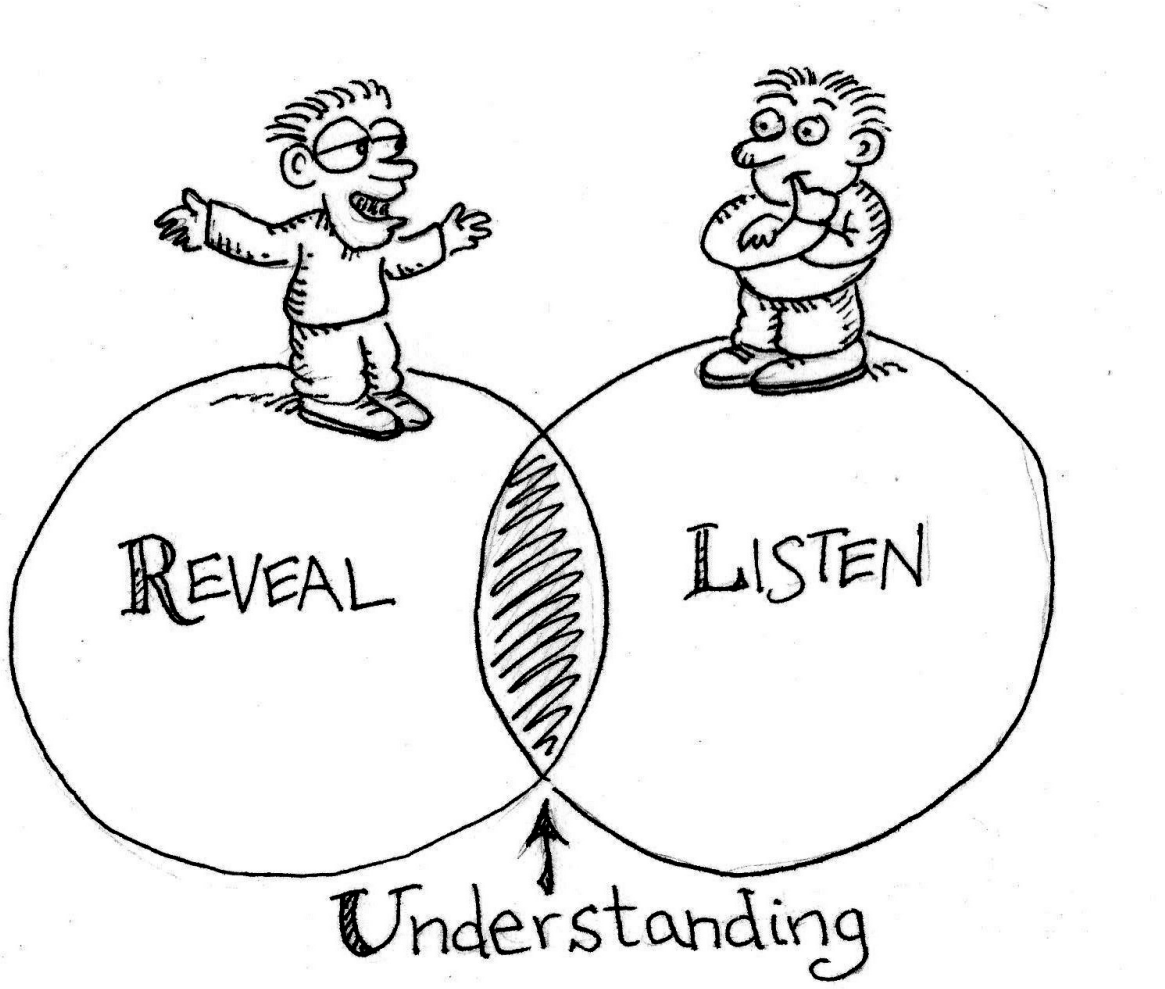
Learning and Understanding, Some Thoughts and Recommendations

Motivation

I have been thinking for a long time what I should choose for a topic of this post- I wanted it to be relatable to my life, courseworks I've taken, and thoughts I had in my mind recently, and combining all these I was able to formulate a concrete subject on what I wanted to talk about.

Background

I've been taking classes for Computer Science, as well as some classes from Data Science and Statistics that also taught coding and alike. Stats 133 is one of these latter classes that I'm taking this semester. One thing I noticed from taking these classes is that the pace of introductory coding classes quite differs. I also noticed that taking a class doesn't necessarily mean that I understood everything, and that the content and pace of class really mattered in me absorbing the material. So I decided to compare the "learning pace, difficulty and understanding" of each introductory coding classes from following three majors - Computer Science, Data Science and Statistics. That is, I will be comparing the content, pace and the amount of understandings I got from CS 61A, DATA C8, and STAT 133.

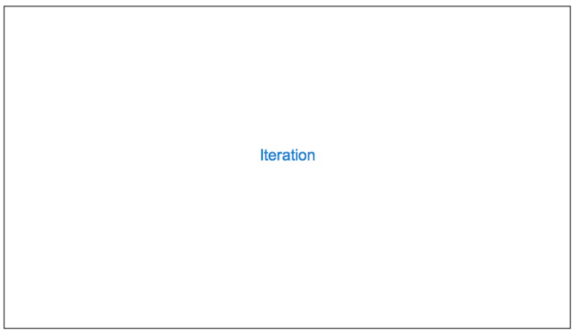


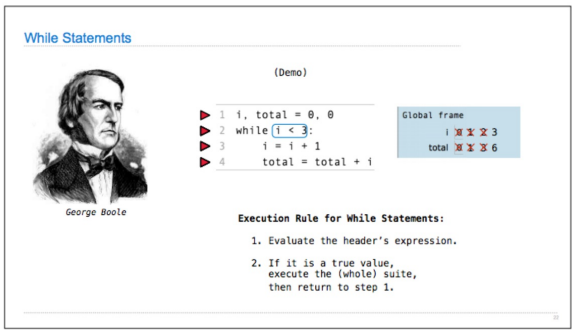
Some Examples

The following are three examples that I thought would show the different learning paces of each classes. I went to official home pages of CS 61A, DATA C8, and STAT 133 of this semester, Fall 2017, for the most accurate and updated comparison of these classes.
The links to references can be found in the bottom of this post.

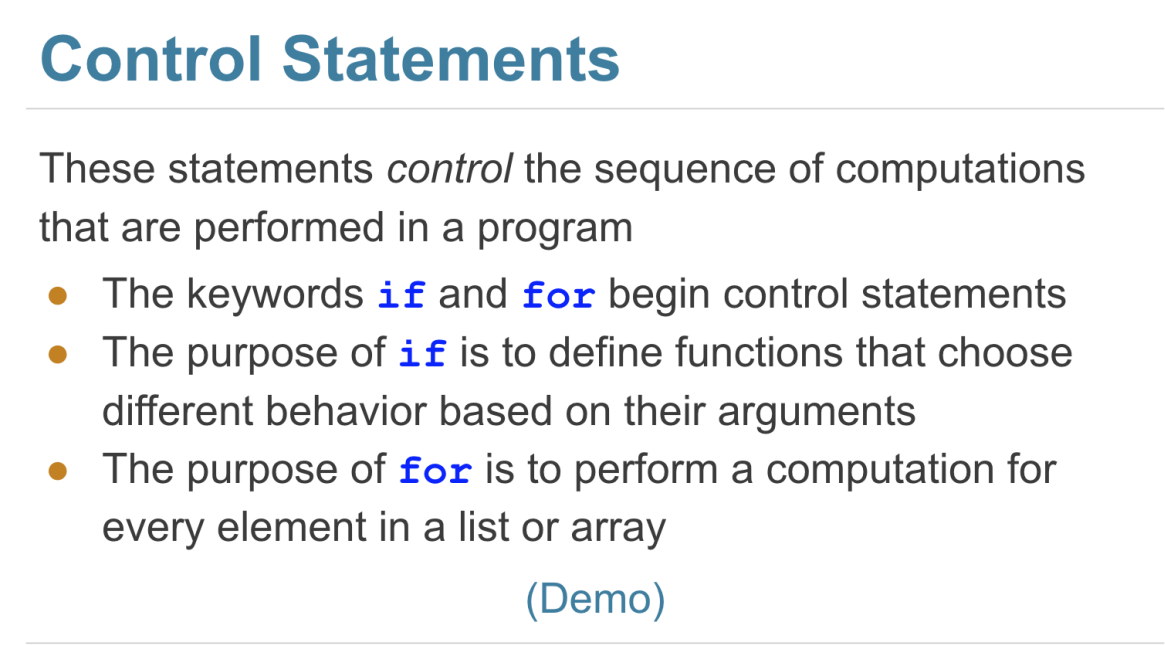
1. Iteration (While Loops and For Loops)

I found out that in **CS 61A**, *while loop* was taught during Lecture 3. <*Reference 1*>





Then I found introduction to *for loop* in Lecture 13 of **DATA C8**. <*Reference 2*>.



Finally, I found out that *while loop* and *for loop* were taught during Lecture 22ish (Oct 16-20, but without exact lecture number on reference) of **STAT 133**. The followings are code examples from <*Reference 3*>.

While Loop

```
value <- 2

while (value < 40) {
  value <- value * 2
  print(value)
}
```

```
## [1] 4
## [1] 8
## [1] 16
## [1] 32
## [1] 64
```

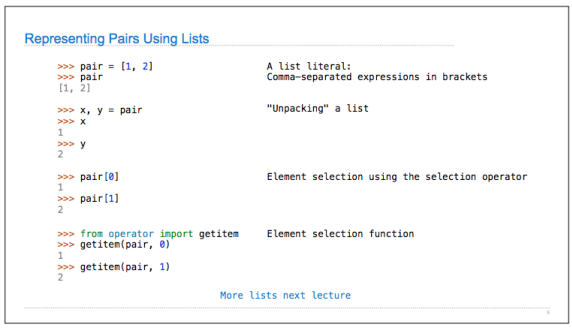
For Loop

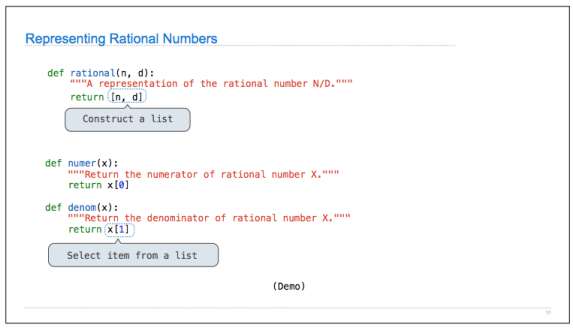
```
value <- 2
times <- c('one', 'two', 'three', 'four')

for (i in times) {
  value <- value * 2
  print(value)
}
```

```
## [1] 4
## [1] 8
## [1] 16
## [1] 32
```

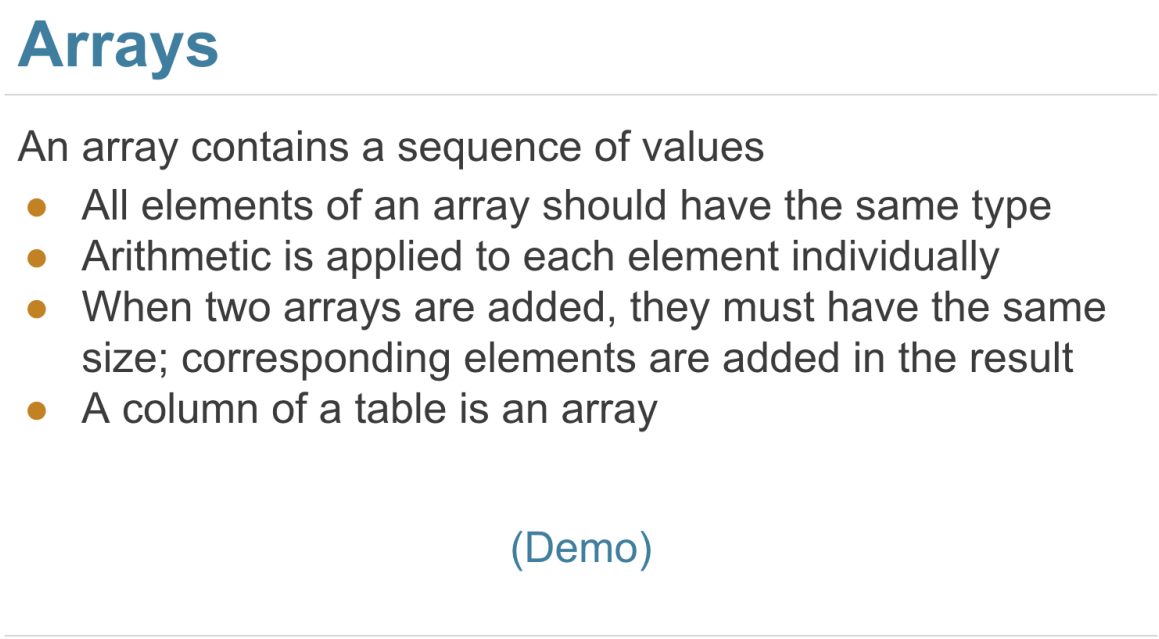
2. Expressions and Data Representations (List and Array)

I found introduction to *list* in Lecture 9 of **CS61A**. *<Reference 4>*





Then I found in Lecture 4 of **DATA C8** where it addressed *arrays* and how to index into array values. *<Reference 5>*



Finally, I was able to find *list* in Lecture 5 of **STAT 133**. The following is an example of list from the lecture. *<Reference 6>*

```
# make a list that contains many different types of data containers
lst <- list(
  c(1,2,3),
  matrix(1:9, nrow=3, ncol=3),
  list(1:2, c(TRUE, FALSE), c("a", "b"))
)

lst[2]
```

```
## [[1]]
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
```

3. Visualization (Tables and Graphs)

I found out that in **CS 61A**, they didn't go over *tables* yet, but will go over next month (according to the syllabus). They will be using SQL to select statements, join, aggregate and group data.

On the other hand, *table* is a big topic in **DATA C8**, and I was able to see it from Lecture 3. *<Reference 7>*

Table Structure

- We organize our data in tables
- A Table is a sequence of labeled columns
- Data within a column should be of the same "type"

Label

Name	Code	Area (m2)
California	CA	163696
Nevada	NV	110567

Row

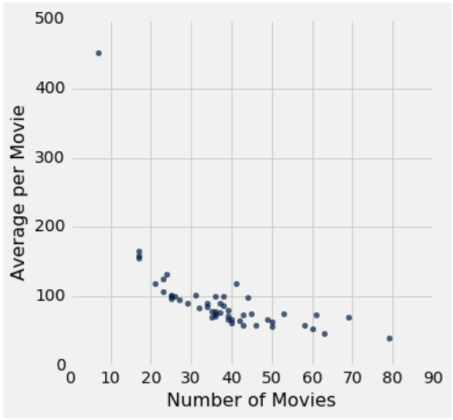
Column

(Demo)

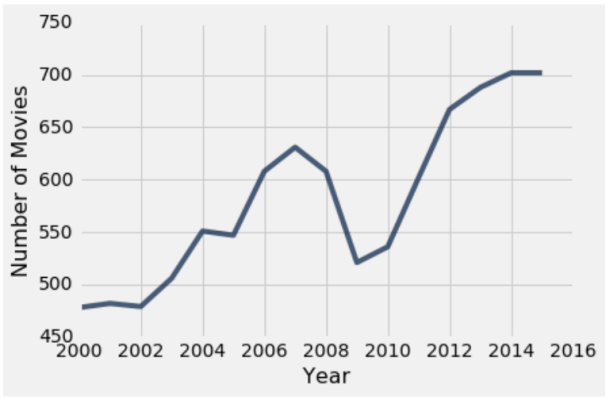
In the same lecture, simple table operations were also taught. Then in Lecture 7, they introduced creating *graphs*. <*Reference 8*>

Plotting Two Numerical Variables

Scatter plot: `scatter`



Line graph: `plot`



Finally, *tables* are introduced in Lecture 9 of **STAT 133**. The example is from lab 4 of the same week as Lecture 9 where we practiced manipulating data frames. <*Reference 9*>
We created table like the following:

```
# make a table with proper column names and valid values
Player = c("Thompson", "Curry", "Green", "Durant", "Pachulia")
Position = c("SG", "PG", "PF", "SF", "C")
Salary = c(16663575, 12112359, 15330435, 26540100, 2898000)
Points = c(1742, 1999, 776, 1555, 426)
PPG = c(22.3, 25.3, 10.2, 25.1, 6.1)
Rookie =  c(FALSE, FALSE, FALSE, FALSE, FALSE)
first_table <- data.frame(Player, Position, Salary, Points, PPG, Rookie)
first_table
```

```
##      Player Position   Salary Points  PPG Rookie
## 1 Thompson       SG 16663575   1742 22.3  FALSE
## 2     Curry       PG 12112359   1999 25.3  FALSE
## 3     Green       PF 15330435    776 10.2  FALSE
## 4    Durant       SF 26540100   1555 25.1  FALSE
## 5 Pachulia        C  2898000    426  6.1  FALSE
```

Then we learned how to plot a *graph* in the following week. The example is from lab 5 <*Reference 10*>.

```
# load these functions to make a graph
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

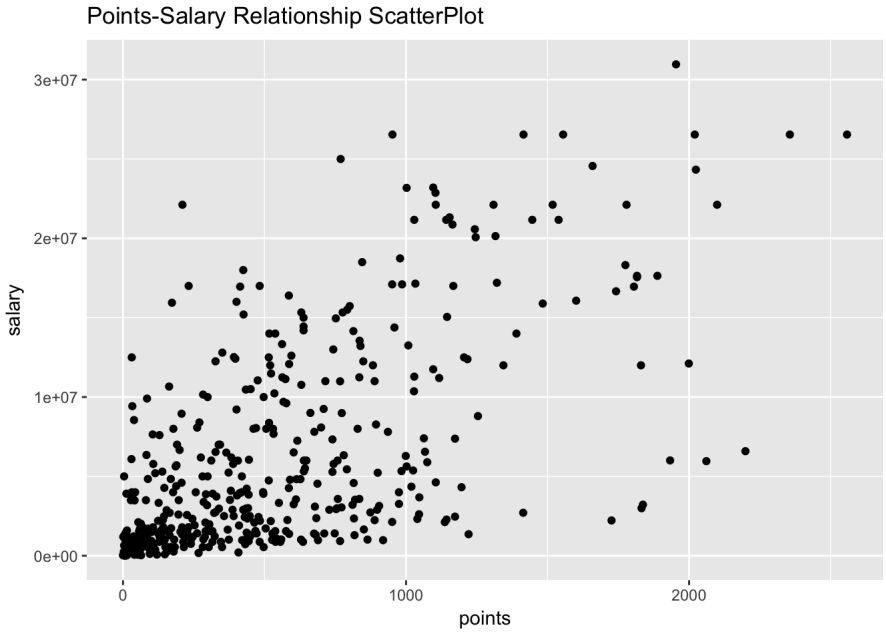
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked _by_ '.GlobalEnv':
##
##     Position
```

```
# load the data
dat <- read.csv('nba2017-players.csv', stringsAsFactors = FALSE)

# create a scatterplot using ggplot
ggplot(data = dat) +
  geom_point(aes(x = points, y = salary)) +
  ggtitle("Points-Salary Relationship ScatterPlot")
```



The above graph shows the relationship between salary and the points made by the player in 2017 NBA.

Content Differences

We can look at the difference in content of each class by looking at the syllabus.
Looking at the syllabus, I found out that each class focused on:

- CS 61A : Functions, Environmental Diagram, Tree, OOP, Scheme, Interpreters, Tail Calls, Exceptions, Iterators, Interpreters, Streams, Tables, Recursive Select, Natural Language
- DATA C8 : Causality and Experience, Programming in Python, Data Types, Tables, Visualization, Functions, Randomness, Empirical Distribution, Testing Hypothesis, Prediction, Inference for Regression, Classification
- STAT 133 : R skills, Data Types, Filesystem, Tabular Data, Tables, Graphics, Programming Basics, Random Numbers, Manipulating String, Regular Expressions, XML, Data Visualization

My Understandings and Evaluations on These Courses

One thing that I noticed while taking STAT 133 is that even though we started coding from the first class, more CS-ish materials like defining function and using iterations were taught later in the course. We used more already-defined functions given by R, and I think that is why I suddenly noticed that there was a big difference between classes. I also figured out that differences came from different learning objectives of the classes, because for CS, the main goal is to learn how to come up with your own code, basically using your own brain and imagination to create new from nothing (or a bit of code chunk to help build onto the project), while STAT and DATA is more like using R or other data analysis programs (such as pandas, SQL) to help computation and visualization of given data, putting more emphasis on analysis of data than coding.
CS 61A is a very fast-paced class; with all new concepts and several different programs such as Python, Scheme, SQL to work with, I just couldn't follow that well. Even though I finished the course, I didn't know half of what I did at the end. I didn't have any CS background before taking CS 61A in my freshman year, and I still do think that if one doesn't have good enough CS background, CS 61A is not a good class to start with. The instructors assume that we understand how coding works, and what we learned in lecture was nothing compared to actual project, and I was very lost.
DATA C8 was far lot better to me than CS 61A. I loved the reading material the class gave out, and it did help me through most of the course works. I felt that I was able to learn a lot in this class, so I would say the pace was nice and the material was also fitting. Also, this class is the reason why I chose to major in Data Science. STAT 133 is what I'm taking right now. I think the pace is great, and I also love how things are gradually getting harder. But out of all, I think homework and especially labs help you understand the material really well. I would also say that I'm understanding most of what is going on right now, and I'm enjoying learning R!

Recommendations

As a junior who first intended to major in Computer Science, who then changed her mind to major in Cogsci, and then decided to major in Data Science, I get asked a lot from fellow classmates and younger ones to recommend good tech classes to take. I also get asked a lot about differences in each majors - about what the learning content is like, how hard each classes are, etc, and that's partially why I started thinking about pros and cons of each classes. In my opinion, CS is usually hard for students not having experience in coding. CS 61A, although Berkeley says it's an introduction class, is more like an intermediate class in other schools; it's hard to follow for many students who even have experience in coding. I've seen one friend who never had experience in coding do ok in this class, but most of my friends who never coded before had hard time throughout the class. Many also dropped out after the first midterm. I would recommend this class to someone who has a good grasp of how coding works or took AP CS classes, but not to new learners. Instead, I would recommend Data C8 or Stat 133 to begin with. After learning step by step how coding works from these classes, working on conceptual ideas and big projects will become less stressful. Although these classes are not CS dedicated classes, the essence of coding is the same - once you get it you can use any program or language to code what you want. So if you're not ready to invest all your time in CS61A (it also doesn't mean that you'll get a good grade) it's always good to step back and start slowly but concretely.

Some Lessons I Learned

I learned that to get most out of a class, choosing a class with right pace and content is more important than just taking it because *I want to major in this and earn six figs*. You need to be prepared! They teach you a lot in any classes at Berkeley- looking at the content of the above mentioned classes, we were also able to figure this out- but if the pace and difficulty is also in the way, you'll proably not get to understand a lot of them. Try to process well on the information you're getting than just trying to pass a class, and I feel like that's what college is for.
I wish this post gave you some thoughts on the difference between just taking a class and understanding the material and making it into your own knowledge, and I wish people don't push themselves through difficult, can't-follow classes, because school should enrich you not depress you.

References

1 - CS 61A Fall 2017 Lecture 3 Slides : https://cs61a.org/assets/slides/03-Control_8pp.pdf
2 - DATA C8 Fall 2017 Lecture 13 Slides:
https://docs.google.com/presentation/d/1duruEKu96sqbzBkeu5PmrSfIiiYRnaa_7IFSDNGLuW8/edit#slide=id.p
3 - STAT 133 Fall 2017 Oct 16-20 Tutorial : https://github.com/ucb-stat133/stat133-fall-2017/blob/master/tutorials/08-intro-to-loops.md
4 - CS 61A Fall 2017 Lecture 9 Slides : https://cs61a.org/assets/slides/09-Data_Abstraction_8pp.pdf
5 - DATA C8 Fall 2017 Lecture 4 Slides :
https://docs.google.com/presentation/d/1mO8Vzo9b1RdOadif8lvoGMthwUV1yVBS6D3dvGhWMiE/edit#slide=id.g247d83b95e_0_181
6 - STAT 133 Fall 2017 Lecture 5 Slides : https://docs.google.com/presentation/d/e/2PACX-1vRQsS5XHV4wyzMgTRXgtYXm7GoF6XLYPB-hFWvT_IvsMEBM1S_zsaPMEKd5wy2b9Uk7oARM29YxdsK7/pub?start=false&loop=false&delayms=3000&slide=id.g26807eb427_0_132
7 - DATA C8 Fall 2017 Lecture 3 Slides :
https://docs.google.com/presentation/d/1jOMen2nHup7wUJQZ8mMMQ_IPkiuM5FZCcaNGyHBUgxg/edit#slide=id.g250d41f98b_1_4
8 - DATA C8 Fall 2017 Lecture 7 Slides : https://docs.google.com/presentation/d/12jcaJkO8msZVLq0fdIG1M-fxTNG4X5TECdWi4srVwqk/edit#slide=id.p
9 - STAT C8 Fall 2017 Lab 4 : https://github.com/ucb-stat133/stat133-fall-2017/blob/master/labs/lab04-reading-tables.md
10 - STAT C8 Fall 2017 Lab 5 : https://github.com/ucb-stat133/stat133-fall-2017/blob/master/labs/lab05-dplyr-ggplot-basics.md