# DSO 545 HW#3

Danniel Winarto

March 23, 2016

## CASE 1

This time we are going to explore a modified dataset from Zillow Home Value Index. The dataset records median estimated home value for all homes with two bedrooms in 24 different regions in Los Angeles.

### Questions

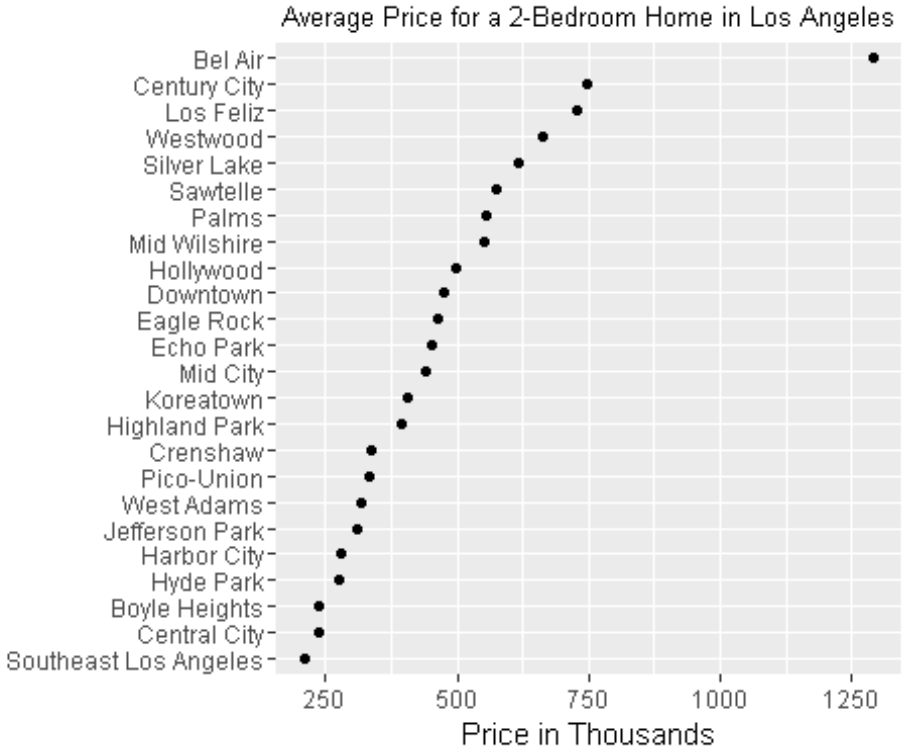
1.(3 points) The following plot shows the mean of the home prices by region. Reproduce an EXACT copy the following graph. (Hint, you might need to aggregate the data!)

Answer: first, we need to set the proper working directory, read the csv files, apply the dplyr and ggplot2 library, and then aggregate using summarise by mean. group by region. Using pipeline method, we can do this using a single line

```
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.2.3

library(dplyr)

## Warning: package 'dplyr' was built under R version 3.2.3

##
## 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

setwd("C:/Users/dan_9/Desktop/DSO 545/HW/HW3")
zillow = read.csv("Zillow2bedroom.csv")
C1Q1_dplyr = zillow %>% group_by(Region) %>% summarise(avgPrice =
mean(Price))
```

This graph is a scatter plot so we use geom_point(). x axis is the average price in thousands so we need to divide x value by 1000, y axis ith the region, with the order from the highest to lowest, so we need to use reorder() function. Lastly, we need to set the x scale from 250 to 1250 with 250 constant increment

```
ggplot(C1Q1_dplyr, aes(x = avgPrice/1000, y = reorder(Region , avgPrice))) +
  geom_point() + xlab("Price in Thousands") + ylab("") +
  ggtitle("Average Price for a 2-Bedroom Home in Los Angeles")+
  theme(plot.title=element_text( size=10))+
  scale_x_continuous( breaks = seq(250,1250,250))
```



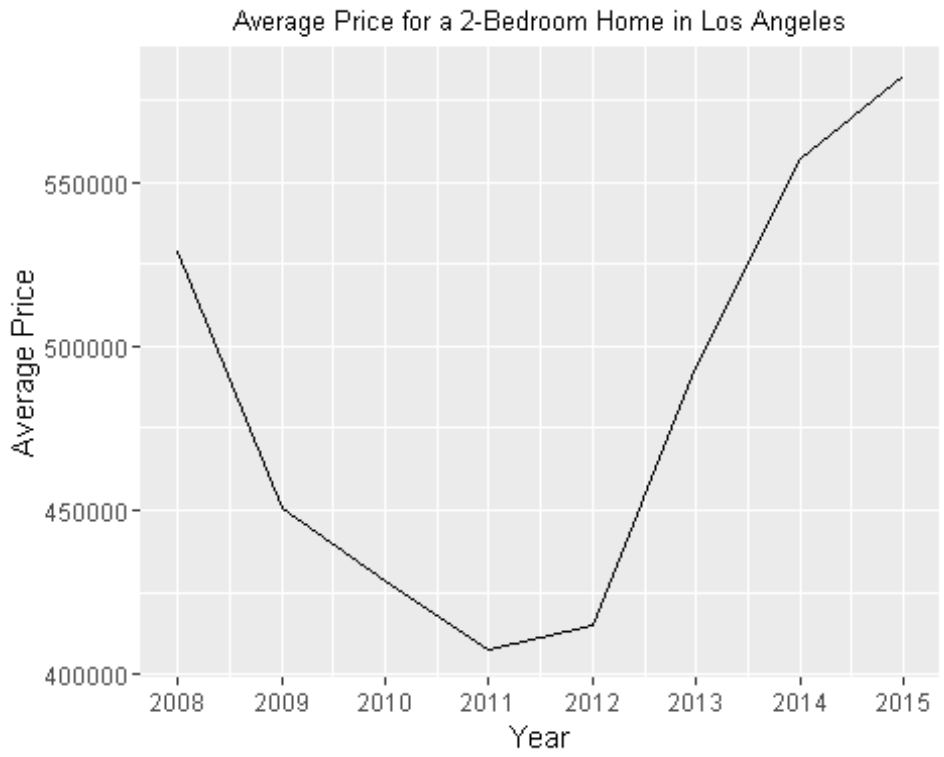

2. (2.5 points) The following plot shows the change in the mean of home prices for the whole city of Los Angeles from 2008 till 2015. Reproduce an EXACT copy the following graph. (Hint, you might need to aggregate the data!)

Answer: first, we need aggregate using summarise by mean, and group by region. Using pipeline method, we can do this using a single line

```
C1Q2_dplyr = zillow %>% group_by(Year) %>% summarise( avgPrice = mean(Price))
```

We use geom line() function since this is a line graph. X-axis is the year and y axis is average peice. then we use scale x continuous() to set the year from 2008 to 2015 with constant increment of 1 year

```
ggplot(C1Q2_dplyr, aes(x = Year , y = avgPrice)) +
  geom_line() + xlab("Year") + ylab("Average Price") +
  ggtitle("Average Price for a 2-Bedroom Home in Los Angeles") +
  theme(plot.title=element_text(size=10))+
  scale_x_continuous( breaks = seq(2008,2015,1))
```



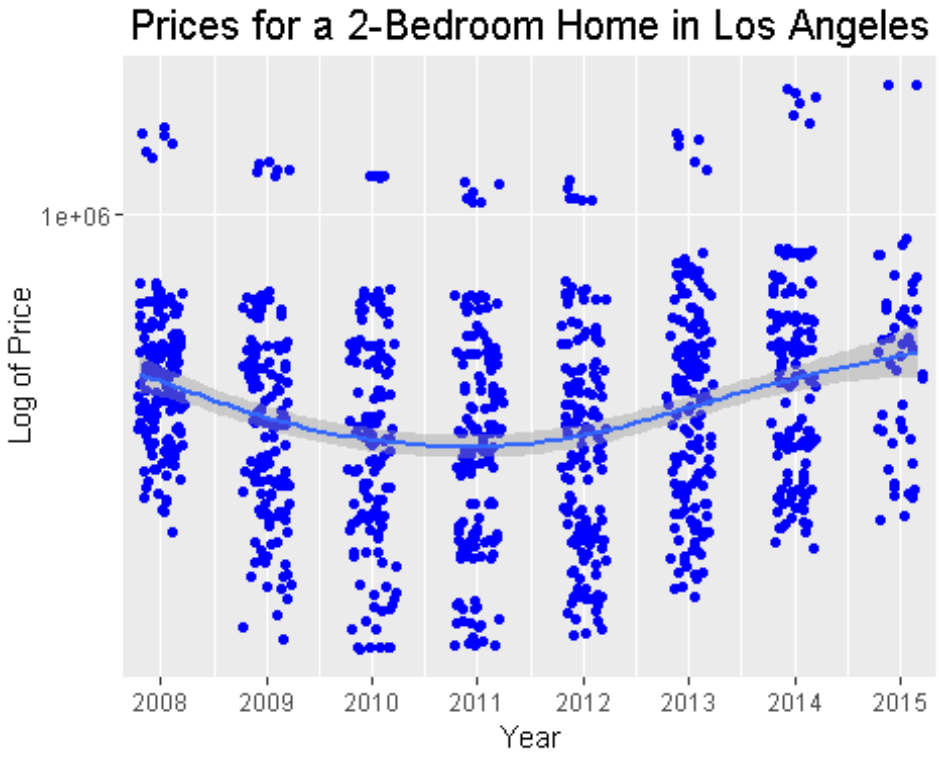

3.(3 points) The following plot shows the home prices for the whole city of Los Angeles from 2008 till 2015. Reproduce an EXACT copy the following graph. (Hint, you might need to add some jitter)

Answer: This is a jitter plot, so we need to put geom_jitter(). x-axis represent year, we need to use jitter(as.numeric(Year)) otherwise it wont create the smooth line. we put the color blue and width jitter = 0.1. We need we set scale x continuous from 2008 to 2015 with increment 1. then set the cartesian limit from 2008 to 20115. Lastly, we add geom smooth for adding the smooth line

```
ggplot(zillow, aes(x = jitter(as.numeric(Year)), y = Price)) +
  xlab("Year") + ylab("Log of Price") + ggtitle("Prices for a 2-Bedroom Home
in Los Angeles") +
   theme(plot.title=element_text(family="Times", size=15))+

  scale_y_log10() + geom_jitter(color = "blue", width = .1) +
  scale_x_continuous( breaks = seq(2008,2015,1))+
  coord_cartesian(xlim = c(2008,2015)) + geom_smooth()
```

```
## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## font family not found in Windows font database

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## font family not found in Windows font database
```

```
## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## font family not found in Windows font database
```

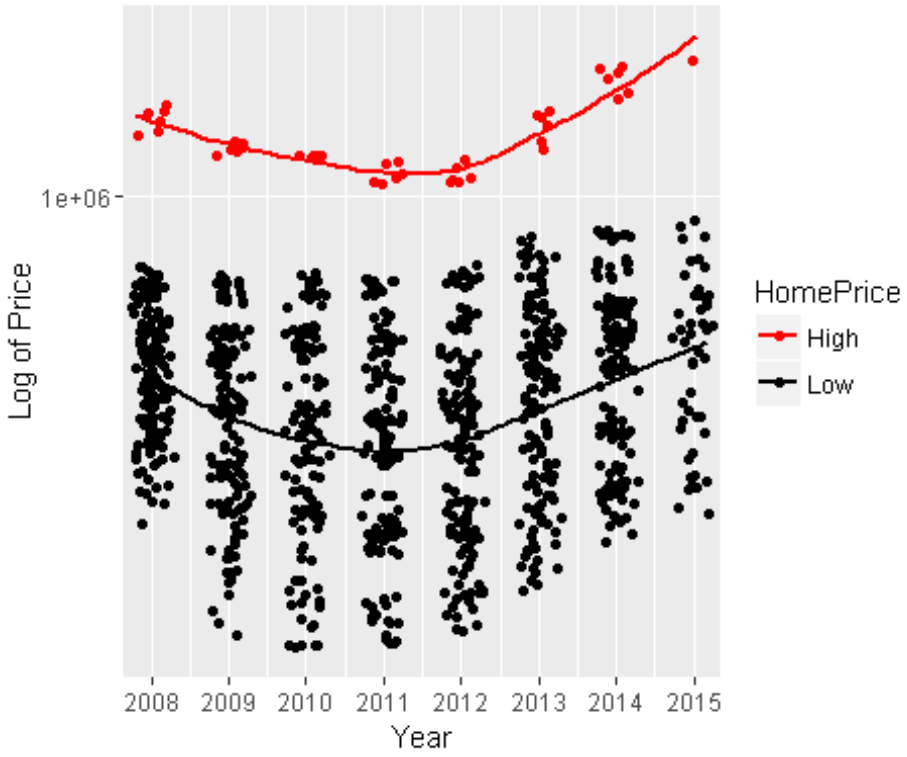



4. (3 points) You can see that there is clear seperation in our plot. For this reason, we will split our data into two catgories: houses with high prices (price >= $1,000,000), and houses with low prices (price < $1,000,000). Reproduce an EXACT copy the following graph. (Hint, you might need to add an extra variable to your dataset!)

Answer: We need to add another variable which is High and Low, using the mutate and pipeline method, we can add this in a single line. We use ifelse function for this variable(HomePrice Variable) For creating the graph, we use very similar method as question 3, but we set the color in aes() to differentiate which one is high and low. then we set the color red and black using function scale color manual. Lastly, clear up the confident interval and smoothing the smooth line using command geom_smooth(se = FALSE, method = "loess")

```
C1Q4_dplyr = zillow %>% mutate(HomePrice = ifelse(Price >= 10^6, "High",
"Low"))

ggplot(C1Q4_dplyr, aes(x = jitter(as.numeric(Year)), y = Price, colour =
HomePrice)) +
  xlab("Year") + ylab("Log of Price") +
  scale_y_log10() + geom_jitter(width = 0.25) +
  scale_x_continuous( breaks = seq(2008,2015,1))+
  scale_color_manual(values = c("red","black"))+
```

```
  coord_cartesian(xlim = c(2008,2015))+
  geom_smooth(se = FALSE,  method = "loess")
```




## CASE 2

It's almost summer time!Let's enjoy some beer together. This dataset contains ratings of 400 different craft beers from various breweries. The dataset contains 7 variables:

- Name : name of the beer
- Brewer : name of the brewer
- Style : style of the beer
- Abv : % of alcohol by volume of the beer
- Ratings : rating of the beer
- ScoreOverall: The overall score of the beer

### Questions

1. (3 points) Create a subset of only IPAs, i.e. include both Imperial/Double IPA and India Pale Ale (IPA). Show the following histagram of Abv. What does the histogram tell you?

Answer: first, we need to read the csv files, apply the dplyr and ggplot2 library, and then we filter the IPA only. Using pipeline method, we can do this using a single line of code

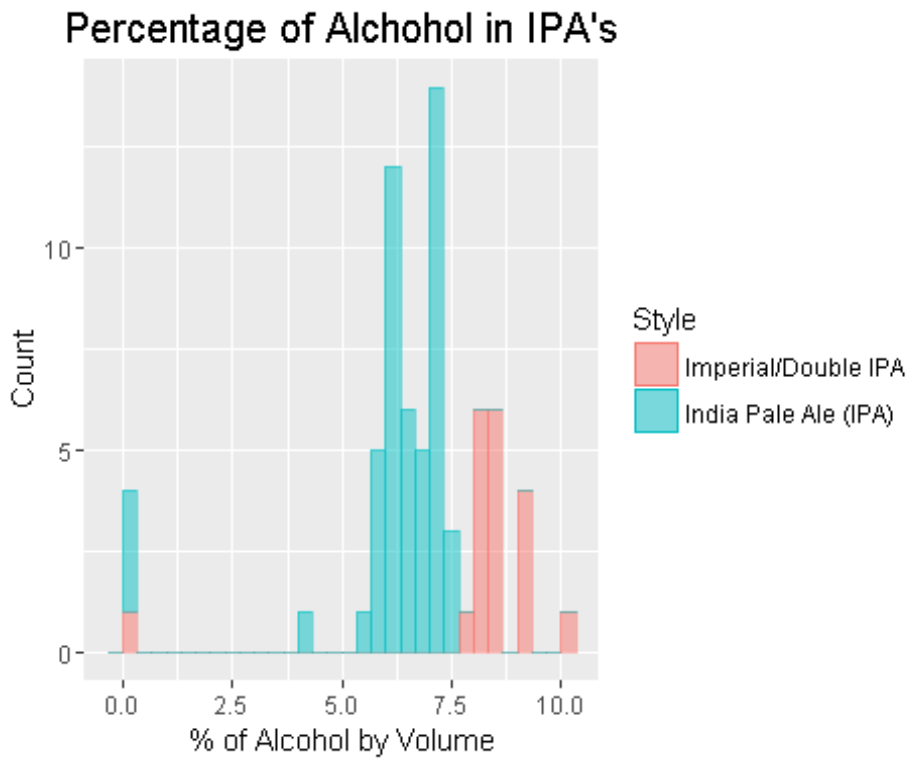
for the plot, we use geom histogram because this is a histogram plot, we set color by Style of the beer and reduca the alpha into 0.5 What does the histogram tell you? the histogram

tell us that majority of imperial/Double IPA beers have more alcohol content than Indian Pale Ale IPA beers

```
beer = read.csv("ratebeer.csv")
case2 = tbl_df(beer)
C2Q1 = case2 %>% filter(grepl("IPA", case2$Style)) %>% filter(Style != "Black IPA")

ggplot(C2Q1, aes(x = Abv, fill = Style, color = Style )) +
geom_histogram(alpha = 0.5)+
  xlab("% of Alcohol by Volume") + ylab("Count") + ggtitle("Percentage of Alchohol in IPA's")+
  theme(plot.title=element_text(family="Times", size=15))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## font family not found in Windows font database

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## font family not found in Windows font database

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## font family not found in Windows font database
```



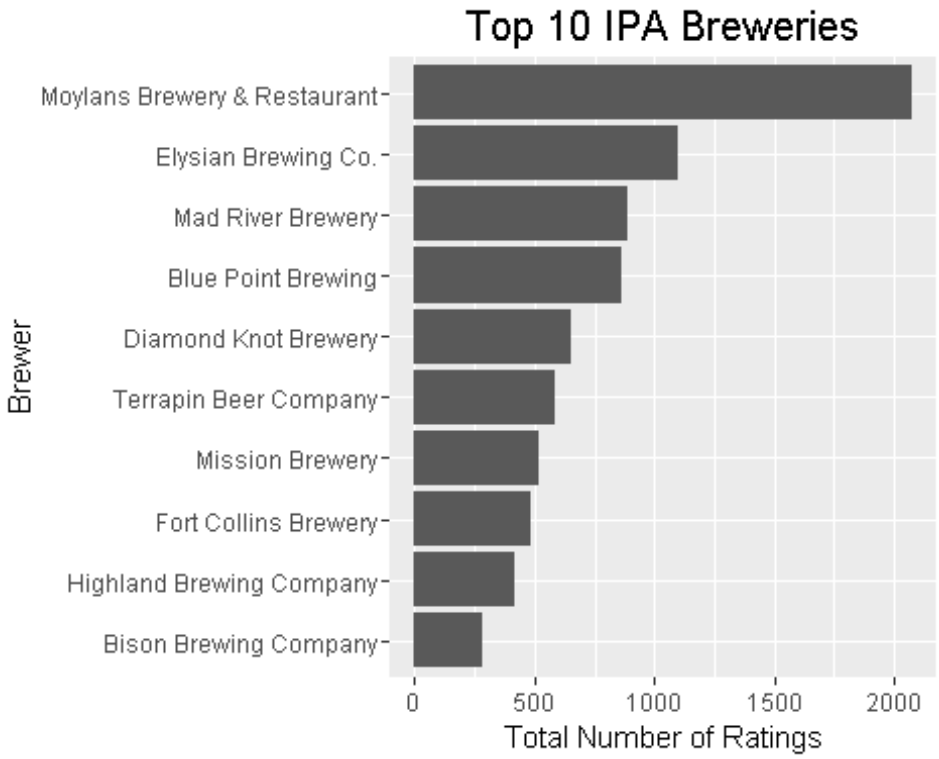
2. (3 points) Based on the subset you created in the previous question, calculate the total number of ratings for each brewer, then reproduce an EXACT graph as the following for the top 10 brewers which has received the highest ratings.

Answer: We need to use group by method from dplyr to group the calculation by brewer, then we use sumarise and sum the ratings by the brewer.

for the plot we use coord flip to flip the barplot, and x is the brewer which ordered from highest to lowest by total ratngs, y axis is the total ratings. lastly, we need to put stats = identity to prevent the program do the count frequency

```
C2Q2 = C2Q1 %>% group_by(Brewer) %>% summarise(totalRatings = sum(Ratings))
%>% arrange(desc(totalRatings))

ggplot(C2Q2[1:10,], aes( y = totalRatings, x = reorder(Brewer, totalRatings
))) + geom_bar(stat = "identity") + coord_flip() +
  ylab("Total Number of Ratings") + xlab("Brewer") + ggtitle("Top 10 IPA
Breweries")+
  theme(plot.title=element_text(family="Times", size=15))
```

```
## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## font family not found in Windows font database

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## font family not found in Windows font database

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## font family not found in Windows font database
```




3. (2.5 points) We're all classy, so we only drink from the top 30 overall rated beers, but we CANNOT drink beers with Abv higher than 8% just because we don't want to be wasted! Randomly choose one beer for yourself from the top 30 rated beers! Output only the name of beer. (Hint: you might need to use sample_n() function from dplyr R package)

Answer: We need to use filter with equal or below specific value (8 percent) of Abv, then we sample random beer using sample n function from dplyr function, while arrange the score from highest to lowest

```
C2Q3 = case2 %>% filter(Abv <= 8.0) %>% arrange(desc(ScoreOverall))

RandomBeer = sample_n(C2Q3[1:10,],1)
RandomBeer
```

```
## Source: local data frame [1 x 7]
##
##                   Name                  Brewer                Style   Abv
##                 (fctr)                  (fctr)               (fctr) (dbl)
## 1 Heretic Evil Cousin Heretic Brewing Company Imperial/Double IPA     8
## Variables not shown: Ratings (int), ScoreOverall (int), ScoreByStyle (int)
```

Random Beer:

```
glimpse(RandomBeer)
```

```
## Observations: 1
## Variables: 7
## $ Name         (fctr) Heretic Evil Cousin
## $ Brewer       (fctr) Heretic Brewing Company
## $ Style        (fctr) Imperial/Double IPA
## $ Abv          (dbl) 8
## $ Ratings      (int) 159
## $ ScoreOverall (int) 97
## $ ScoreByStyle (int) 80
```