

STA 380 Exam-II

Parthiv Borgohain, Ruchi Sharma, Saurabh Arora and Soumith Reddy Palreddy

08/14/2022

Link to GitHub Repository <https://github.com/parthiv-borgohain/STA-380-Part-II>

This markdown has been prepared as a submission for STA 380: Introduction to Machine Learning by **Parthiv Borgohain (MSBA, pb25347)**, **Ruchi Sharma (MSBA, rs58898)**, **Saurabh Arora (MSBA,sa55445)** and **Soumith Reddy Palreddy (MSBA, sp52466)**

Probability Practice

Question:

Part A. Visitors to your website are asked to answer a single survey question before they get access to the content on the page. Among all of the users, there are two categories: Random Clicker (RC), and Truthful Clicker (TC). There are two possible answers to the survey: yes and no. Random clickers would click either one with equal probability. You are also giving the information that the expected fraction of random clickers is 0.3. After a trial period, you get the following survey results: 65% said Yes and 35% said No. What fraction of people who are truthful clickers answered yes? Hint: use the rule of total probability.

Part B. Imagine a medical test for a disease with the following two attributes:

The sensitivity is about 0.993. That is, if someone has the disease, there is a probability of 0.993 that

The specificity is about 0.9999. This means that if someone doesn't have the disease, there is probability

In the general population, incidence of the disease is reasonably rare: about 0.0025% of all people have

Suppose someone tests positive. What is the probability that they have the disease?

Part A.

$$P(Y|RC) = P(N|RC) = 0.5$$

$$P(RC) = 0.3$$

$$P(Y) = 0.65 \Rightarrow P(N) = 0.35$$

$$\Rightarrow 0.65 = 0.7 \times P(Y|TC) + 0.3 \times 0.5$$

$$\Rightarrow P(Y|TC) = 0.714$$

Part B.

```
P(P|H) = 0.993  
P(P|H) = 0.9999  
P(H) = 0.000025  
=> P(H|P) = 0.000025 X 0.993 / [(0.000025 x 0.993) + (1 - 0.9999) x  
(1 - 0.000025)]  
=> P(H|P) = 0.198882413
```

Wrangling the Billboard Top 100

Part A

```
library(tidyverse)  
  
## -- Attaching packages ----- tidyverse 1.3.2 --  
## v ggplot2 3.3.6     v purrr   0.3.4  
## v tibble  3.1.7     v dplyr    1.0.9  
## v tidyr   1.2.0     v stringr  1.4.0  
## v readr   2.1.2     vforcats  0.5.1  
## -- Conflicts ----- tidyverse_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag()   masks stats::lag()  
  
library(dplyr)  
library(gridExtra)  
  
##  
## Attaching package: 'gridExtra'  
##  
## The following object is masked from 'package:dplyr':  
##  
##     combine  
  
billboard=read.csv('billboard.csv')  
summary(billboard)  
  
##           X             url            week_id        week_position  
##  Min.   : 1  Length:327895  Length:327895  Min.   : 1.0  
##  1st Qu.: 81975 Class :character  Class :character  1st Qu.: 25.5  
##  Median :163948 Mode  :character  Mode  :character  Median : 50.0  
##  Mean   :163948                  Mode  :character  Mean   : 50.5  
##  3rd Qu.:245922                  Mode  :character  3rd Qu.: 75.0  
##  Max.   :327895                  Mode  :character  Max.   :100.0  
##  
##           song          performer        song_id        instance  
##  Length:327895  Length:327895  Length:327895  Min.   : 1.000  
##  Class :character Class :character  Class :character  1st Qu.: 1.000
```

```

##   Mode :character    Mode :character    Mode :character    Median : 1.000
##                                         Mean   : 1.073
##                                         3rd Qu.: 1.000
##                                         Max.   :10.000
##
##   previous_week_position peak_position    weeks_on_chart      year
##   Min.     : 1.0          Min.     : 1.000  Min.     :1958
##   1st Qu.: 23.0          1st Qu.: 14.00  1st Qu.:1974
##   Median  : 47.0          Median  : 39.00  Median  :1989
##   Mean    : 47.6          Mean    : 41.36  Mean    :1989
##   3rd Qu.: 72.0          3rd Qu.: 66.00  3rd Qu.:2005
##   Max.    :100.0          Max.    :100.00  Max.    :2021
##   NA's    :31954
##
##   week
##   Min.   : 1.00
##   1st Qu.:14.00
##   Median :27.00
##   Mean   :26.59
##   3rd Qu.:40.00
##   Max.   :53.00
##
```

```

df = billboard

head(sort(df$weeks_on_chart, decreasing = TRUE), n = 10)

```

```

## [1] 87 86 85 84 83 82 81 80 79 79

```

```

df2 <- df %>%
  group_by(performer, song) %>%
  summarize(countt = sum(weeks_on_chart))

```

```

## `summarise()` has grouped output by 'performer'. You can override using the
## `.` argument.

```

```

df2

```

```

## # A tibble: 29,389 x 3
## # Groups:   performer [10,061]
##   performer           song       countt
##   <chr>              <chr>      <int>
## 1 'N Sync            (God Must Have Spent) A Little More Time On ~ 253
## 2 'N Sync            Bye Bye Bye  276
## 3 'N Sync            Gone        300
## 4 'N Sync            I Drive Myself Crazy 78
## 5 'N Sync            I Want You Back 300
## 6 'N Sync            It's Gonna Be Me 325
## 7 'N Sync            Pop         120
## 8 'N Sync            Tearin' Up My Heart 1
## 9 'N Sync            This I Promise You 351
## 10 'N Sync & Gloria Estefan Music Of My Heart 210
## # ... with 29,379 more rows
## # i Use `print(n = ...)` to see more rows

```

```

head(sort(df2$countt, decreasing = TRUE), n = 10)

## [1] 3828 3160 2926 2926 2415 2346 2346 2145 2145 2080

df2 <- df2[order(-df2$countt),]

df3 <- df %>%
  group_by(song_id) %>%
  summarize(countt = sum(weeks_on_chart)) %>%
  head

summary(df3)

##      song_id          countt
##  Length:6      Min.   : 3.0
##  Class :character 1st Qu.:15.0
##  Mode  :character Median :30.0
##                  Mean   :94.0
##                  3rd Qu.:168.8
##                  Max.   :276.0

```

Part B.

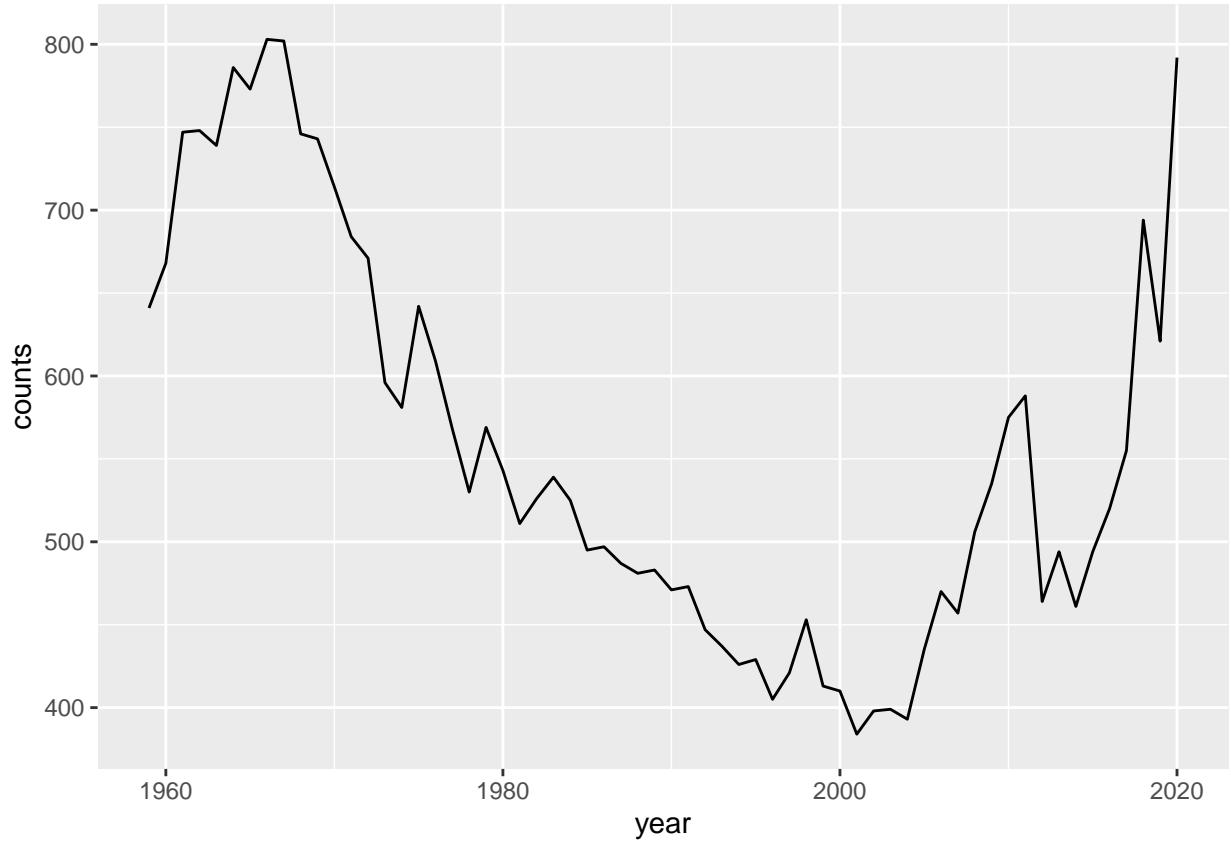
```

df4 <- df %>%
  group_by(year) %>%
  summarize(counts = length(unique(song)))

df4 = df4[-1,]
df4 = df4[-63,]

ggplot(df4) +
  geom_line(aes(x=year, y=counts))

```



Part C.

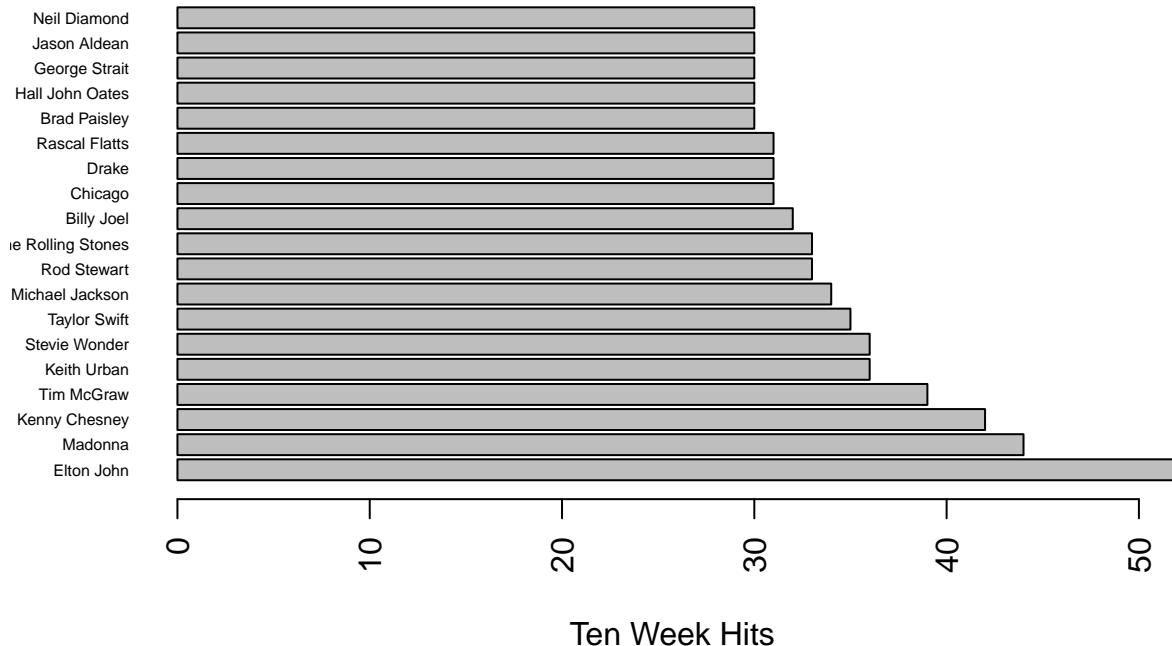
```
df5 <- df[which(df$weeks_on_chart >= 10),]

df6 <- df5 %>%
  group_by(performer) %>%
  summarize(counts = length(unique(song)))

df6 <- df6[which(df6$counts >= 30),]
df6 <- df6[order(-df6$counts),]

barplot(df6$counts, main="Ten Week Hits",
        xlab="Ten Week Hits", names.arg = df6$performer,
        horiz = TRUE, las=2, cex.names=.5)
```

Ten Week Hits



Visual Story Telling Part 1: Green Buildings

```
library(readr)
library(plyr)

## -----
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)

## -----
## 
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':
## 
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarise
```

```

## The following object is masked from 'package:purrr':
##
##     compact

library(reshape2)

##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths

greenbuildings <- read_csv("greenbuildings.csv")

## Rows: 7894 Columns: 23

## -- Column specification -----
## Delimiter: ","
## dbl (23): CS_PropertyID, cluster, size, empl_gr, Rent, leasing_rate, stories...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

df = greenbuildings
summary(df)

##   CS_PropertyID      cluster        size       empl_gr
##   Min. : 1      Min. : 1.0      Min. : 1624      Min. :-24.950
##   1st Qu.: 157452  1st Qu.: 272.0    1st Qu.: 50891     1st Qu.: 1.740
##   Median : 313253  Median : 476.0    Median : 128838    Median : 1.970
##   Mean   : 453003  Mean   : 588.6    Mean   : 234638    Mean   : 3.207
##   3rd Qu.: 441188  3rd Qu.: 1044.0   3rd Qu.: 294212   3rd Qu.: 2.380
##   Max.   : 6208103 Max.   :1230.0    Max.   :3781045   Max.   : 67.780
##                                NA's   :74
##   Rent      leasing_rate      stories      age
##   Min. : 2.98  Min. : 0.00  Min. : 1.00  Min. : 0.00
##   1st Qu.: 19.50 1st Qu.: 77.85 1st Qu.: 4.00  1st Qu.: 23.00
##   Median : 25.16  Median : 89.53  Median : 10.00 Median : 34.00
##   Mean   : 28.42  Mean   : 82.61  Mean   : 13.58 Mean   : 47.24
##   3rd Qu.: 34.18  3rd Qu.: 96.44  3rd Qu.: 19.00 3rd Qu.: 79.00
##   Max.   :250.00  Max.   :100.00  Max.   :110.00 Max.   :187.00
##
##   renovated      class_a      class_b      LEED
##   Min. :0.0000  Min. :0.0000  Min. :0.0000  Min. :0.0000000
##   1st Qu.:0.0000 1st Qu.:0.0000  1st Qu.:0.0000 1st Qu.:0.0000000
##   Median :0.0000  Median :0.0000  Median :0.0000  Median :0.0000000
##   Mean   :0.3795  Mean   :0.3999  Mean   :0.4595  Mean   :0.006841
##   3rd Qu.:1.0000  3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:0.0000000
##   Max.   :1.0000  Max.   :1.0000  Max.   :1.0000  Max.   :1.0000000
##

```

```

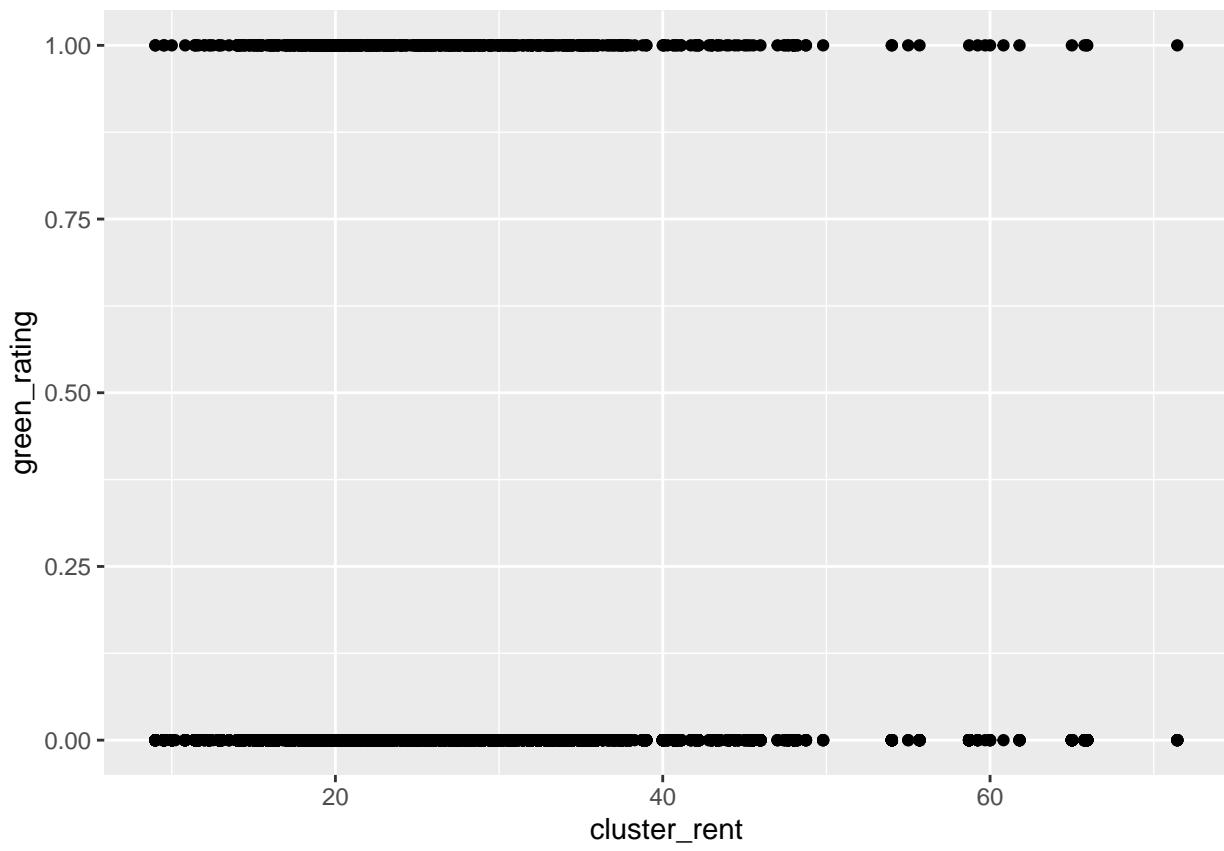
##   Energystar      green_rating       net      amenities
## Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.0000
## 1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.0000
## Median :0.00000   Median :0.00000   Median :0.00000   Median :1.0000
## Mean    :0.08082   Mean    :0.08677   Mean    :0.03471   Mean    :0.5266
## 3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:1.0000
## Max.    :1.00000   Max.    :1.00000   Max.    :1.00000   Max.    :1.0000
##
##   cd_total_07      hd_total07      total_dd_07  Precipitation
## Min.   : 39   Min.   : 0   Min.   :2103   Min.   :10.46
## 1st Qu.: 684  1st Qu.:1419  1st Qu.:2869   1st Qu.:22.71
## Median : 966  Median :2739   Median :4979   Median :23.16
## Mean    :1229  Mean    :3432   Mean    :4661   Mean    :31.08
## 3rd Qu.:1620  3rd Qu.:4796  3rd Qu.:6413   3rd Qu.:43.89
## Max.    :5240  Max.    :7200   Max.    :8244   Max.    :58.02
##
##   Gas_Costs      Electricity_Costs  cluster_rent
## Min.   :0.009487  Min.   :0.01780   Min.   : 9.00
## 1st Qu.:0.010296  1st Qu.:0.02330   1st Qu.:20.00
## Median :0.010296  Median :0.03274   Median :25.14
## Mean    :0.011336  Mean    :0.03096   Mean    :27.50
## 3rd Qu.:0.011816  3rd Qu.:0.03781   3rd Qu.:34.00
## Max.    :0.028914  Max.    :0.06280   Max.    :71.44
##

```

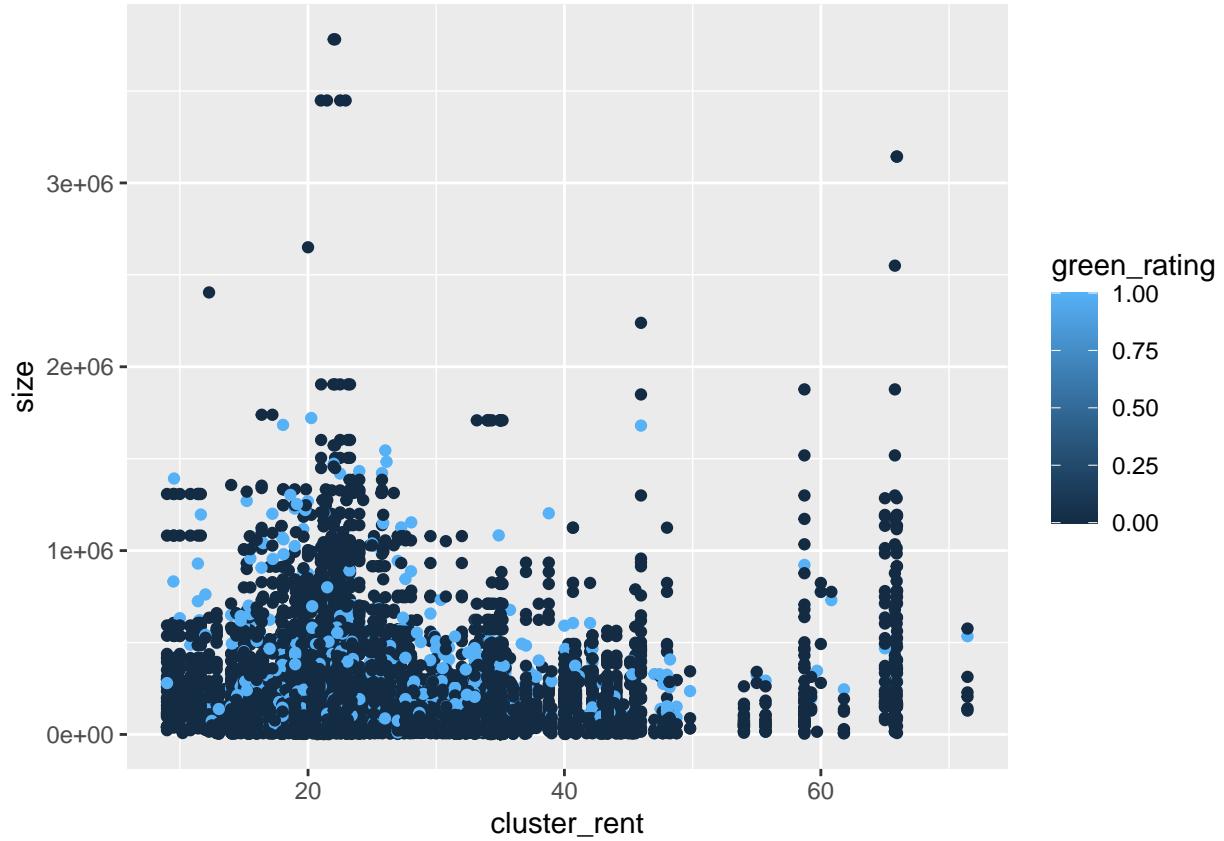
```
# Looking at distributions across some pair combinations
```

```
library(tidyverse)

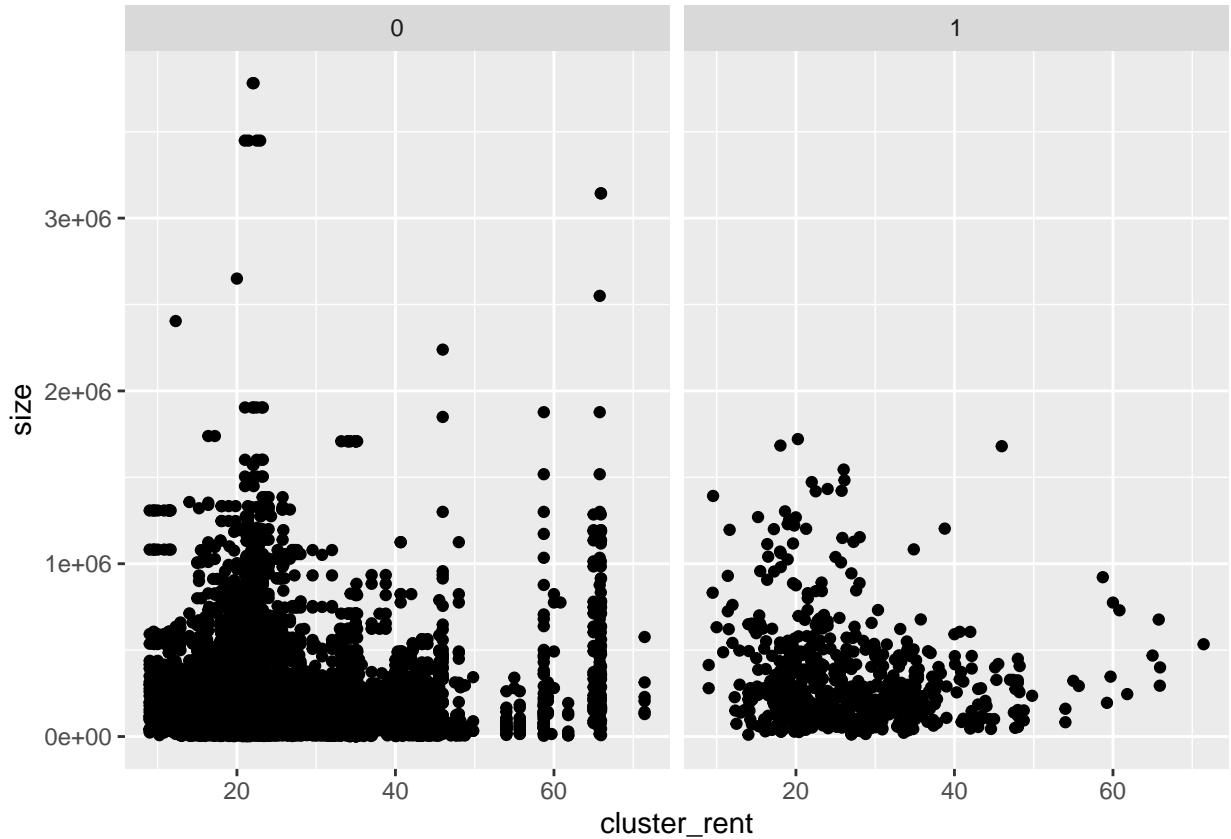
ggplot(df) +
  geom_point(aes(x=cluster_rent, y=green_rating)) # almost similar distribution
```



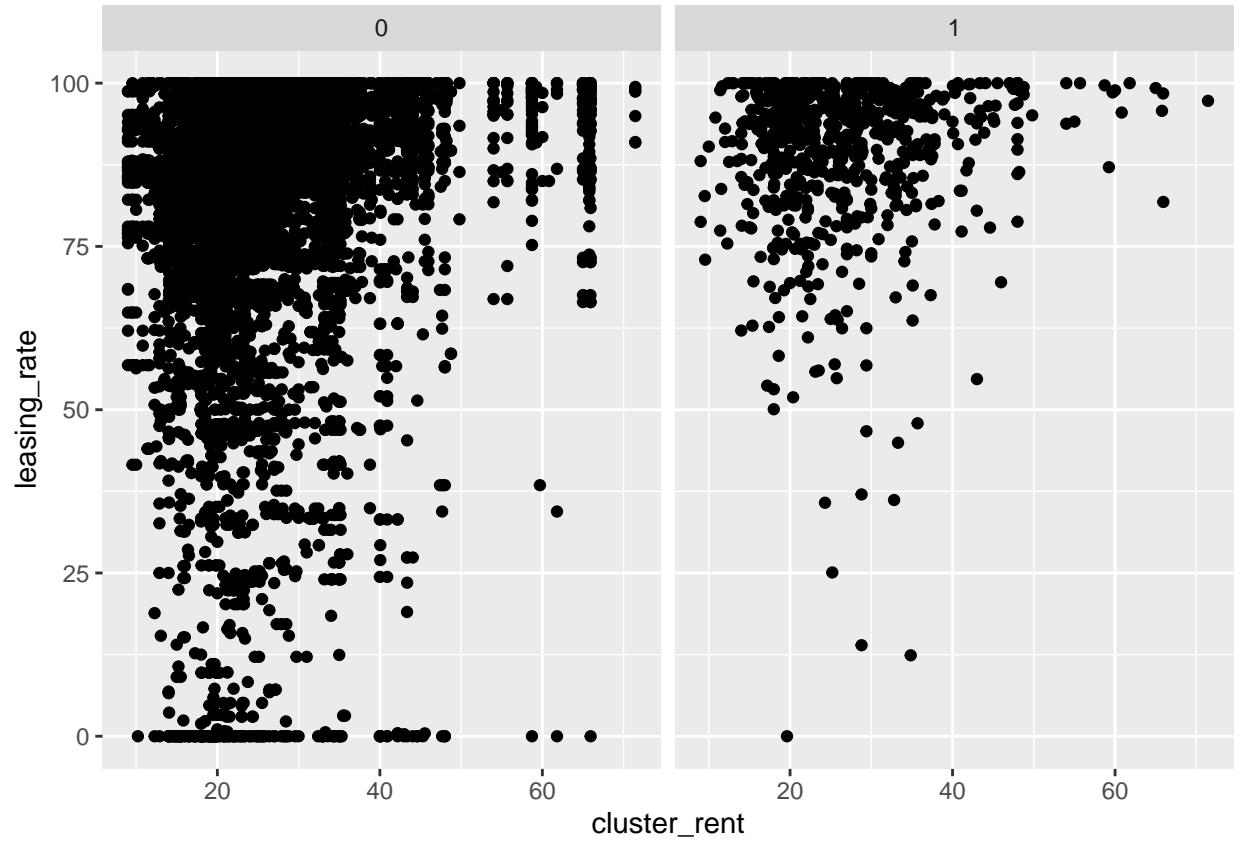
```
ggplot(df) +  
  geom_point(aes(x=cluster_rent, y=size, color=green_rating))
```



```
ggplot(df) +  
  geom_point(aes(x=cluster_rent, y=size)) +  
  facet_wrap(~green_rating)
```

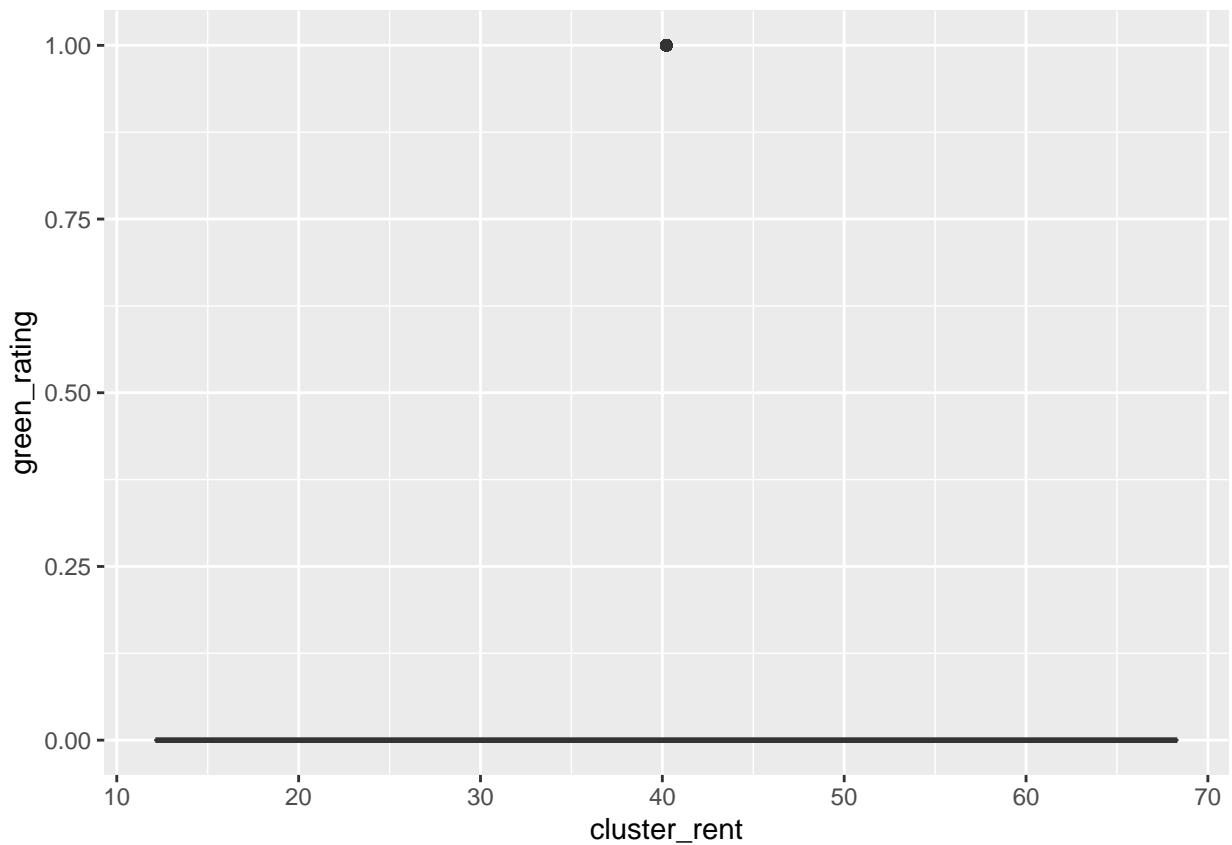


```
ggplot(df) +  
  geom_point(aes(x=cluster_rent, y=leasing_rate)) +  
  facet_wrap(~green_rating)
```



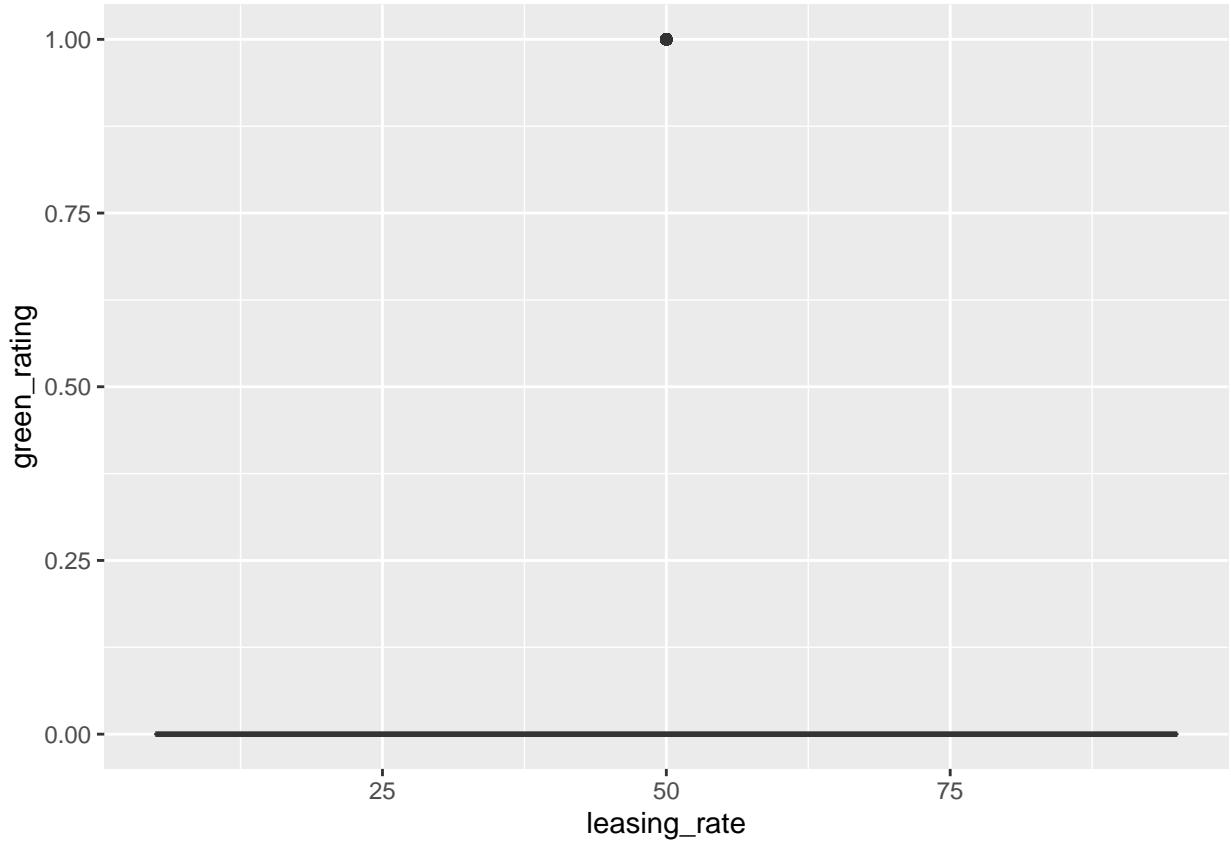
```
ggplot(df) +  
  geom_boxplot(aes(x=cluster_rent, y=green_rating))
```

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```



```
ggplot(df) +  
  geom_boxplot(aes(x=leasing_rate, y=green_rating))
```

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```



```
# Splitting the data frame into green vs non-green

green = subset(df ,df$green_rating == 1)
non_green = subset(df ,df$green_rating!= 1)

median(green$Rent)

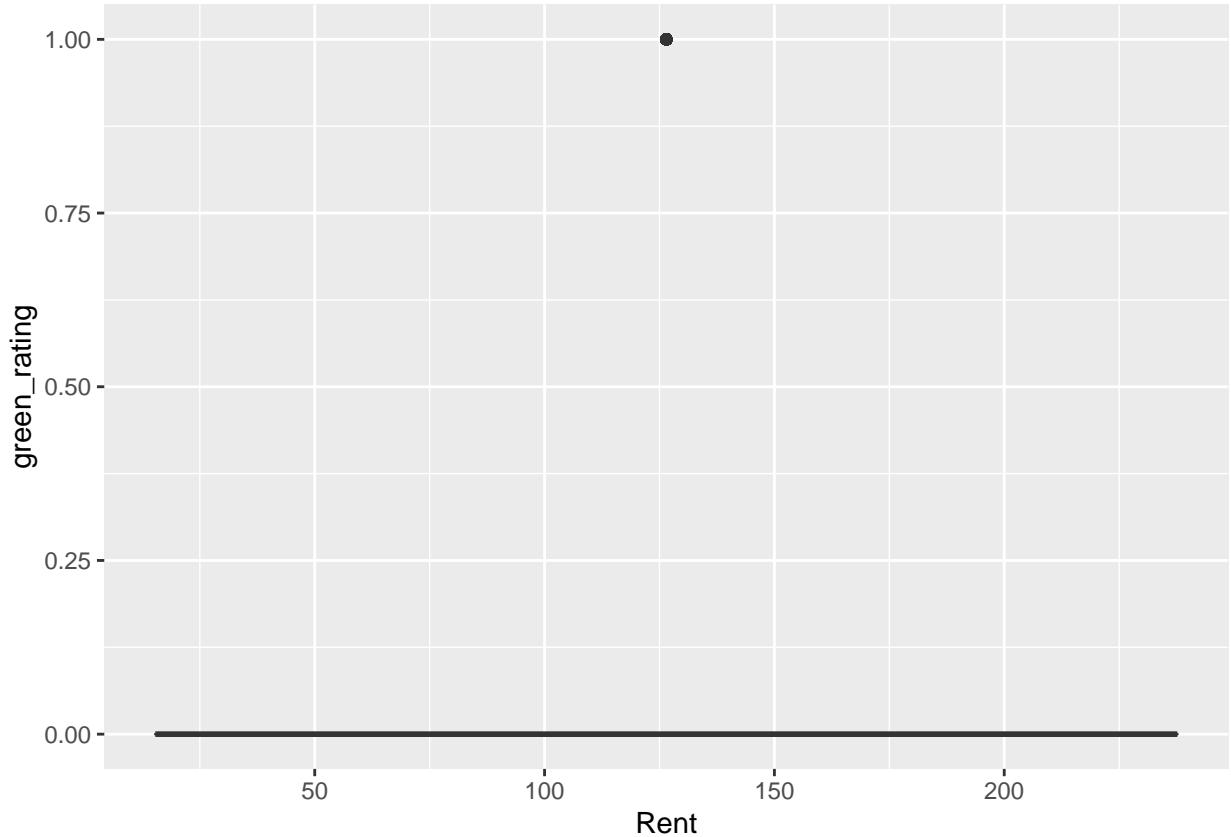
## [1] 27.6

median(non_green$Rent)

## [1] 25

ggplot(df) +
  geom_boxplot(aes(x=Rent, y=green_rating))

## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```



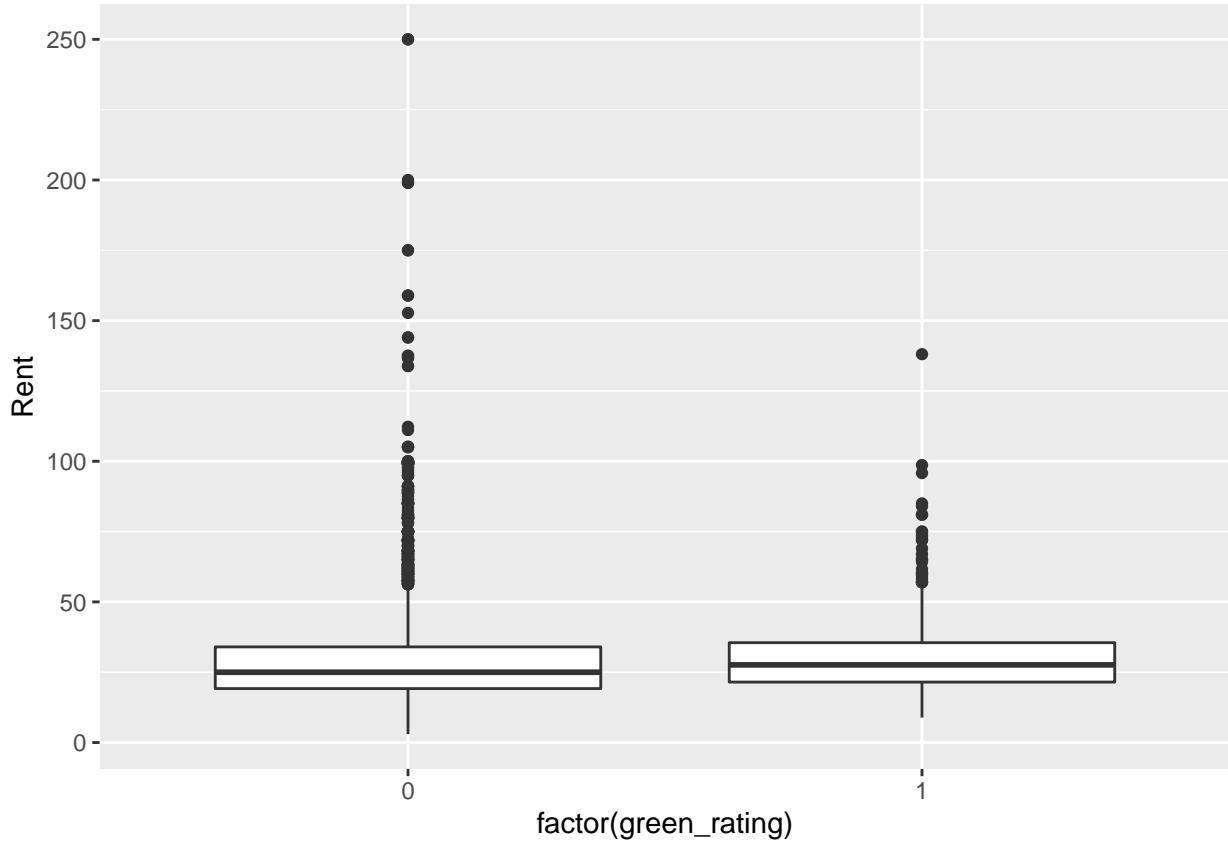
```
# Median Rent for Green Buildings > Median Rent for Non Green Buildings
```

```
# Looking at the statistics for average rent
```

```
df %>%
  summarize(avg_rent = mean(Rent),
            sd_rent = sd(Rent),
            q05_rent = quantile(Rent, 0.05),
            q95_rent = quantile(Rent, 0.95)) %>%
  round(1)
```

```
##   avg_rent  sd_rent  q05_rent  q95_rent
## 1     28.4    15.1     12.5     51.8
```

```
ggplot(df) +
  geom_boxplot(aes(x=factor(green_rating), y=Rent))
```



```

df %>%
  group_by(green_rating) %>%
  summarize(avg_rent = mean(Rent),
            sd_rent = sd(Rent),
            q05_rent = quantile(Rent, 0.05),
            q95_rent = quantile(Rent, 0.95)) %>%
  round(1)

##   avg_rent sd_rent q05_rent q95_rent
## 1     28.4    15.1     12.5     51.8

# Plotting a set of graphs to see what other factors impact the Rent

knitr::opts_chunk$set(fig.width = 20, fig.height = 10)

# Leasing_rate

leasing_rate = ggplot(df, aes(x=leasing_rate, y=Rent))+ 
  theme_classic() + geom_point(colour = "darkolivegreen", size = 1.5, alpha = 0.5) +
  labs(x = "Leasing Rate", y = "Rent", title = "Leasing Rate Vs Rent",
       subtitle = "All buildings") +
  theme(axis.text.x =
        element_text(face = "bold", color = "black", size = 8, angle = 0),
        axis.text.y = element_text(face="bold", color="black",
        size=8, angle=0), plot.title = element_text(hjust = 0.5),

```

```

plot.subtitle = element_text(hjust = 0.5))

# Age

age = ggplot(df, aes(x = age,y = Rent)) + theme_classic() +
  geom_point(colour = "navyblue", size = 1.5,alpha = 0.5)+ 
  labs(x = "Age", y = "Rent",title = "Age Vs Rent",
       subtitle = "All buildings") + theme(axis.text.x = element_text(
       face="bold",color="black", size=8, angle=0),
       axis.text.y = element_text(face="bold", color="black", size=8, angle=0),
       plot.title = element_text(hjust = 0.5),
       plot.subtitle = element_text(hjust = 0.5))

# Renovations

df$renovated = as.factor(df$renovated)
renovation <- ggplot(df, aes(x = renovated, y = Rent)) + 
  geom_boxplot(colour = "grey", fill = "#CC9900") + theme_classic() + 
  labs(x = "Renovation(Yes:1| No:0)", y = "Rent",title = "Renovation Vs Rent",
       subtitle = "All buildings") + theme(axis.text.x = element_text(
       face="bold",color="black", size=8, angle=0),
       axis.text.y = element_text(face="bold", color="black", size=8, angle=0),
       plot.title = element_text(hjust = 0.5),
       plot.subtitle = element_text(hjust = 0.5)) + stat_summary(fun=median,
       geom="point", shape=20, size=3, color="red", fill="red")

# Renovations in buildings older than 30 years

renovation_30_years_more = subset(df, df$age >=30)
renovation_30 <- ggplot(renovation_30_years_more,
  aes(x = renovated, y = Rent)) + geom_boxplot(colour = "darkgrey",
  fill = "#33CC99") + theme_classic() + labs(x = "Renovation(Yes:1| No:0)",
  y = "Rent",title = "Old renovated buildings Vs Rent",
  subtitle = "All buildings") + theme(axis.text.x = element_text(
  face="bold",color="black", size=8, angle=0), axis.text.y = element_text(
  face="bold", color="black", size=8, angle=0),
  plot.title = element_text(hjust = 0.5),
  plot.subtitle = element_text(hjust = 0.5)) + stat_summary(fun=median,
  geom="point", shape=20, size=3, color="#999900", fill="red")

# Amenities

df$amenities = as.factor(df$amenities)
amenities <- ggplot(df, aes(x = amenities, y = Rent)) + 
  geom_boxplot(colour = "lightgrey", fill = "#000066") + theme_classic() + 
  labs(x = "Amenities", y = "Rent",title = "Amenities Vs Rent",
       subtitle = "All buildings") + theme(axis.text.x = element_text(
       face="bold",color="black", size=8, angle=0),
       axis.text.y = element_text(face="bold", color="black", size=8,angle=0),
       plot.title = element_text(hjust = 0.5),
       plot.subtitle = element_text(hjust = 0.5))+ 
  stat_summary(fun=median, geom="point", shape=20, size=3,
              color="#00FF66", fill="red")

```

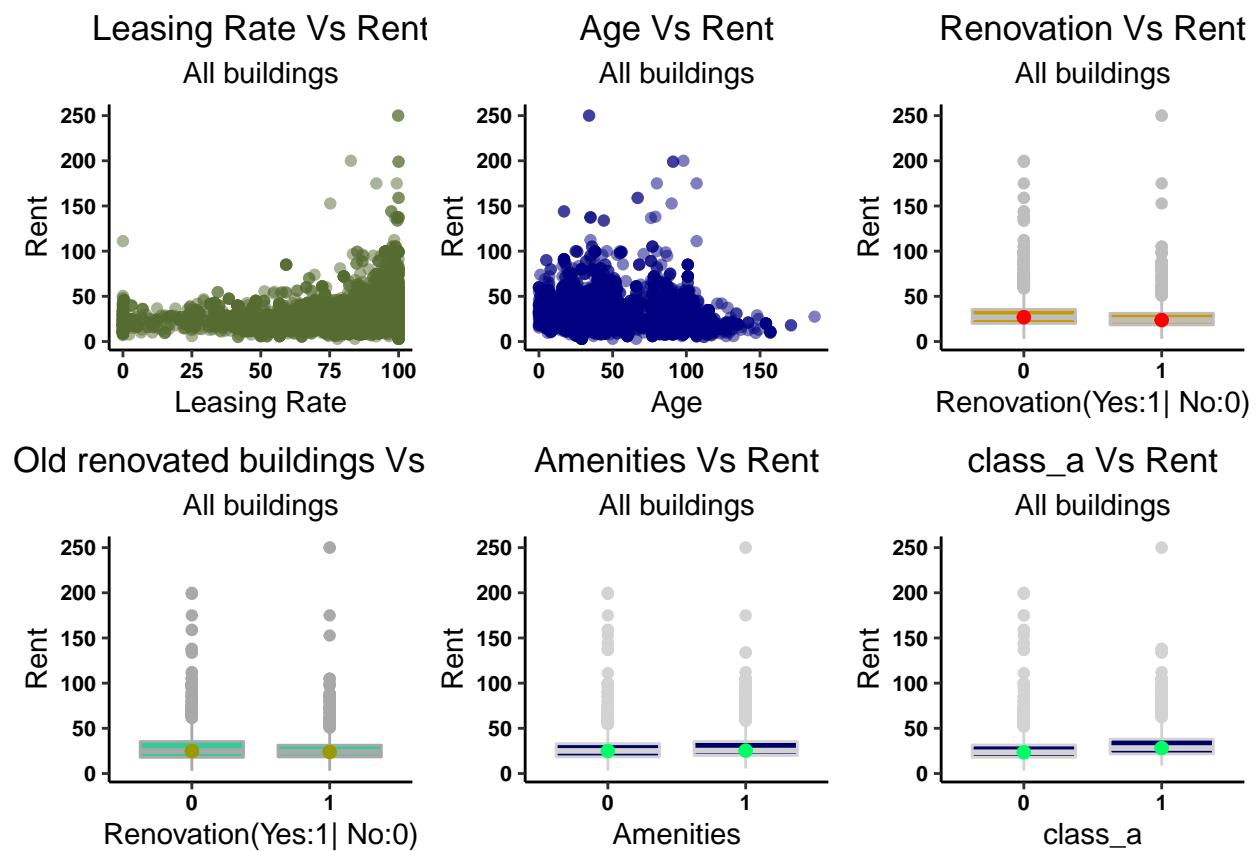
```

# Class_a buildings

df$class_a = as.factor(df$class_a)
class_a <- ggplot(df, aes(x = class_a, y = Rent)) +
  geom_boxplot(colour = "lightgrey", fill = "#000066") + theme_classic() +
  labs(x = "class_a", y = "Rent", title = "class_a Vs Rent",
       subtitle = "All buildings") + theme(axis.text.x = element_text(
         face="bold", color="black", size=8, angle=0),
       axis.text.y = element_text(face="bold", color="black", size=8, angle=0),
       plot.title = element_text(hjust = 0.5),
       plot.subtitle = element_text(hjust = 0.5)) +
  stat_summary(fun=median, geom="point", shape=20, size=3,
               color="#00FF66", fill="red")

grid.arrange(leasing_rate, age, renovation, renovation_30,
             amenities, class_a, ncol = 3)

```



```

# Observations:

# Buildings with amenities and class_a buildings have a slightly higher rent
# Rent and Age do not have a visible relationship
# Leasing Rate and Rent seem to have a slightly positive relationship

# Looking at the green vs non green buildings separately

```

```

knitr::opts_chunk$set(fig.width=12, fig.height=8)

green_buildings = subset(df, df$green_rating == 1)
non_green_buildings = subset(df, df$green_rating!= 1)

# Leasing_rate

# Green

leasing_rate_g = ggplot(green_buildings, aes(x=leasing_rate, y=Rent)) +
  theme_classic() + geom_point(colour = "darkolivegreen", size = 1.5, alpha= 0.5) +
  labs(x = "Leasing Rate", y = "Rent", title = "Leasing Rate Vs Rent",
       subtitle = "Green buildings") + ylim(0,250) + theme(axis.text.x =
    element_text(face="bold", color="black", size=8, angle=0),
    axis.text.y = element_text(face="bold", color="black", size=8, angle=0),
    plot.title = element_text(hjust = 0.5), plot.subtitle =
    element_text(hjust = 0.5))

# Non Green

leasing_rate_ng = ggplot(non_green_buildings, aes(x=leasing_rate, y=Rent)) +
  theme_classic() + geom_point(colour = "darkolivegreen", size = 1.5, alpha= 0.5) +
  labs(x = "Leasing Rate", y = "Rent", title = "Leasing Rate Vs Rent",
       subtitle = "Non Green buildings") + ylim(0,250) + theme(axis.text.x =
    element_text(face="bold", color="black", size=8, angle=0),
    axis.text.y = element_text(face="bold", color="black", size=8, angle=0),
    plot.title = element_text(hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.5))

# Age

age_g = ggplot(green_buildings, aes(x = age, y = Rent)) +
  theme_classic() + geom_point(colour = "navyblue", size = 1.5, alpha = 0.5) +
  labs(x = "Age", y = "Rent", title = "Age Vs Rent",
       subtitle = "Green buildings") + ylim(0,250) + theme(axis.text.x =
    element_text(face="bold", color="black", size=8, angle=0),
    axis.text.y = element_text(face="bold", color="black", size=8, angle=0),
    plot.title = element_text(hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.5))

# Renovations

renovation_g <- ggplot(green_buildings, aes(x = renovated, y = Rent)) +
  geom_boxplot(colour = "grey", fill = "#CC9900") + ylim(0,250) +
  theme_classic() + labs(x = "Renovation(Yes:1| No:0)", y = "Rent",
    title = "Renovation Vs Rent", subtitle = "Green buildings") +
  theme(axis.text.x = element_text(face="bold", color="black",
    size=8, angle=0), axis.text.y = element_text(face="bold",
    color="black", size=8, angle=0), plot.title =
    element_text(hjust = 0.5), plot.subtitle = element_text(hjust = 0.5)) +
  stat_summary(fun=median, geom="point", shape=20, size=3, color="red",
    fill="red")

```

```

# Renovations in buildings older than 30 years

renovation_30_years_more = subset(green_buildings,green_buildings$age >=30)
renovation_30_g <- ggplot(renovation_30_years_more,
  aes(x = renovated, y = Rent)) + geom_boxplot(colour = "darkgrey",
  fill = "#33CC99") + theme_classic() + ylim(0,250) +
  labs(x = "Renovation(Yes:1| No:0)", y = "Rent",
  title = "Old renovated buildings Vs Rent",
  subtitle = "Green buildings") + theme(axis.text.x = element_text(
  face="bold",color="black", size=8, angle=0), axis.text.y = element_text(
  face="bold", color="black", size=8, angle=0),
  plot.title = element_text(hjust = 0.5),
  plot.subtitle = element_text(hjust = 0.5))+ 
  stat_summary(fun=median, geom="point", shape=20,
  size=3, color="#9999900", fill="red")

# Amenities

amenities_g <- ggplot(green_buildings, aes(x = amenities, y = Rent)) +
  geom_boxplot(colour = "lightgrey", fill = "#000066") + theme_classic() +
  ylim(0,250) + labs(x = "Amenities", y = "Rent",title = "Amenities Vs Rent",
  subtitle = "Green buildings") + theme(axis.text.x = element_text(
  face="bold",color="black", size=8, angle=0), axis.text.y = element_text(
  face="bold", color="black", size=8, angle=0),
  plot.title = element_text(hjust = 0.5),
  plot.subtitle = element_text(hjust = 0.5)) +
  stat_summary(fun=median, geom="point",
  shape=20, size=3, color="#00FF66", fill="red")

# class_a buildings

class_a_g <- ggplot(green_buildings, aes(x = class_a, y = Rent)) +
  geom_boxplot(colour = "lightgrey", fill = "#000066") +
  theme_classic() + ylim(0,250) + labs(x = "class_a", y = "Rent",
  title = "class_a Vs Rent",subtitle = "Green buildings") +
  theme(axis.text.x = element_text(face="bold",color="black", size=8,
  angle=0), axis.text.y = element_text(face="bold", color="black", size=8,
  angle=0), plot.title = element_text(hjust = 0.5),
  plot.subtitle = element_text(hjust = 0.5)) + stat_summary(fun=median,
  geom="point", shape=20, size=3, color="#00FF66", fill="red")

# Non-Green buildings

leasing_rate_ng = ggplot(non_green_buildings, aes(x=leasing_rate, y=Rent)) +
  theme_classic()+geom_point(colour = "darkolivegreen", size = 1.5) +
  labs(x = "Leasing Rate", y = "Rent",title = "Leasing Rate Vs Rent",
  subtitle = "Non-Green buildings") + theme(axis.text.x =
  element_text(face="bold",color="black", size=8, angle=0),
  axis.text.y = element_text(face="bold", color="black", size=8, angle=0),
  plot.title = element_text(hjust = 0.5),
  plot.subtitle = element_text(hjust = 0.5))

# Age

```

```

age_ng = ggplot(non_green_buildings,aes(x = age,y = Rent)) + geom_point() +
  theme_classic() + geom_point(colour = "navyblue", size = 1.5) + labs(x = "Age",
  y = "Rent",title = "Age Vs Rent",subtitle = "Non-Green buildings") +
  theme(axis.text.x = element_text(face="bold",color="black", size=8, angle=0),
  axis.text.y = element_text(face="bold", color="black", size=8, angle=0),
  plot.title = element_text(hjust = 0.5),plot.subtitle =
  element_text(hjust = 0.5))

# Renovations

renovation_ng <- ggplot(non_green_buildings, aes(x = renovated, y = Rent)) +
  geom_boxplot(colour = "grey", fill = "#CC9900") + theme_classic() +
  labs(x = "Renovation(Yes:1| No:0)", y = "Rent",title = "Renovation Vs Rent",
  subtitle = "Non-Green buildings") +
  theme(axis.text.x = element_text(face="bold",
  color="black", size=8, angle=0),
  axis.text.y = element_text(face="bold",
  color="black", size=8, angle=0),
  plot.title = element_text(hjust = 0.5),
  plot.subtitle = element_text(hjust = 0.5))+ stat_summary(fun=median,
  geom="point", shape=20, size=3, color="red", fill="red")

# Renovations in buildings older than 30 years

renovation_30_years_more = subset(non_green_buildings,
  non_green_buildings$age >=30)
renovation_30_ng <- ggplot(renovation_30_years_more, aes(x = renovated,
  y = Rent)) + geom_boxplot(colour = "darkgrey", fill = "#33CC99") +
  theme_classic() + labs(x = "Renovation(Yes:1| No:0)", y = "Rent",
  title = "Old renovated buildings Vs Rent", subtitle =
  "Non-Green buildings") + theme(axis.text.x = element_text(
  face="bold",color="black", size=8, angle=0), axis.text.y =
  element_text(face="bold", color="black", size=8, angle=0),plot.title =
  element_text(hjust = 0.5),plot.subtitle = element_text(hjust = 0.5))+
  stat_summary(fun=median, geom="point", shape=20, size=3,
  color="#999900", fill="red")

# Amenities

amenities_ng <- ggplot(non_green_buildings, aes(x = amenities, y = Rent)) +
  geom_boxplot(colour = "lightgrey", fill = "#000066") + theme_classic() +
  labs(x = "Amenities", y = "Rent",title = "Amenities Vs Rent",subtitle =
  "Non-Green buildings") + theme(axis.text.x = element_text(face=
  "bold",color="black", size=8, angle=0), axis.text.y = element_text(
  face="bold", color="black", size=8, angle=0),plot.title =
  element_text(hjust = 0.5),plot.subtitle = element_text(hjust = 0.5))+
  stat_summary(fun=median, geom="point", shape=20, size=3, color="#00FF66",
  fill="red")

# class_a buildings

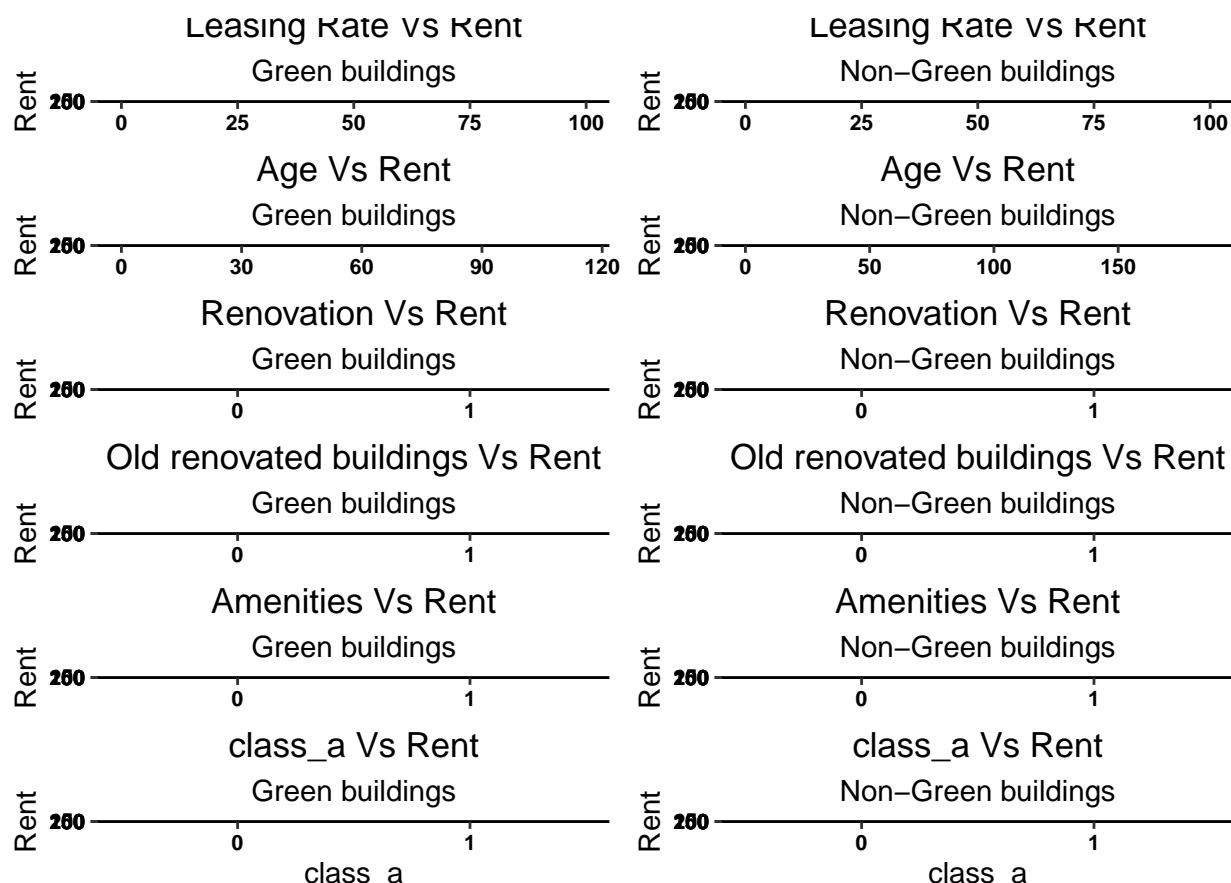
class_a_ng <- ggplot(non_green_buildings, aes(x = class_a, y = Rent)) +
  geom_boxplot(colour = "lightgrey", fill = "#000066") + theme_classic() +

```

```

  labs(x = "class_a", y = "Rent", title = "class_a Vs Rent",
       subtitle = "Non-Green buildings") + theme(axis.text.x = element_text(
         face="bold", color="black", size=8, angle=0), axis.text.y = element_text(
         face="bold", color="black", size=8, angle=0), plot.title = element_text(
         hjust = 0.5), plot.subtitle = element_text(hjust = 0.5)) +
  stat_summary(fun=median, geom="point", shape=20, size=3,
              color="#00FF66", fill="red")
grid.arrange(leasing_rate_g,
             leasing_rate_ng,
             age_g,age_ng,
             renovation_g,
             renovation_ng,
             renovation_30_g,
             renovation_30_ng,
             amenities_g,
             amenities_ng,
             class_a_g,
             class_a_ng,
             ncol = 2)

```



```
# Observations:
```

```

# Older green buildings seem to charge higher rents when renovated
# There is not a clearly visible variable that impacts the distribution of rent
# even after splitting for green vs non-green

```

```

# Diving deep into Age

green_buildings_by_age = subset(df,
                               df$green_rating == 1 & df$age <= 30 & df$net ==0)
non_green_buildings_by_age = subset(df,
                                     df$green_rating!= 1 & df$age <= 30 & df$net ==0)

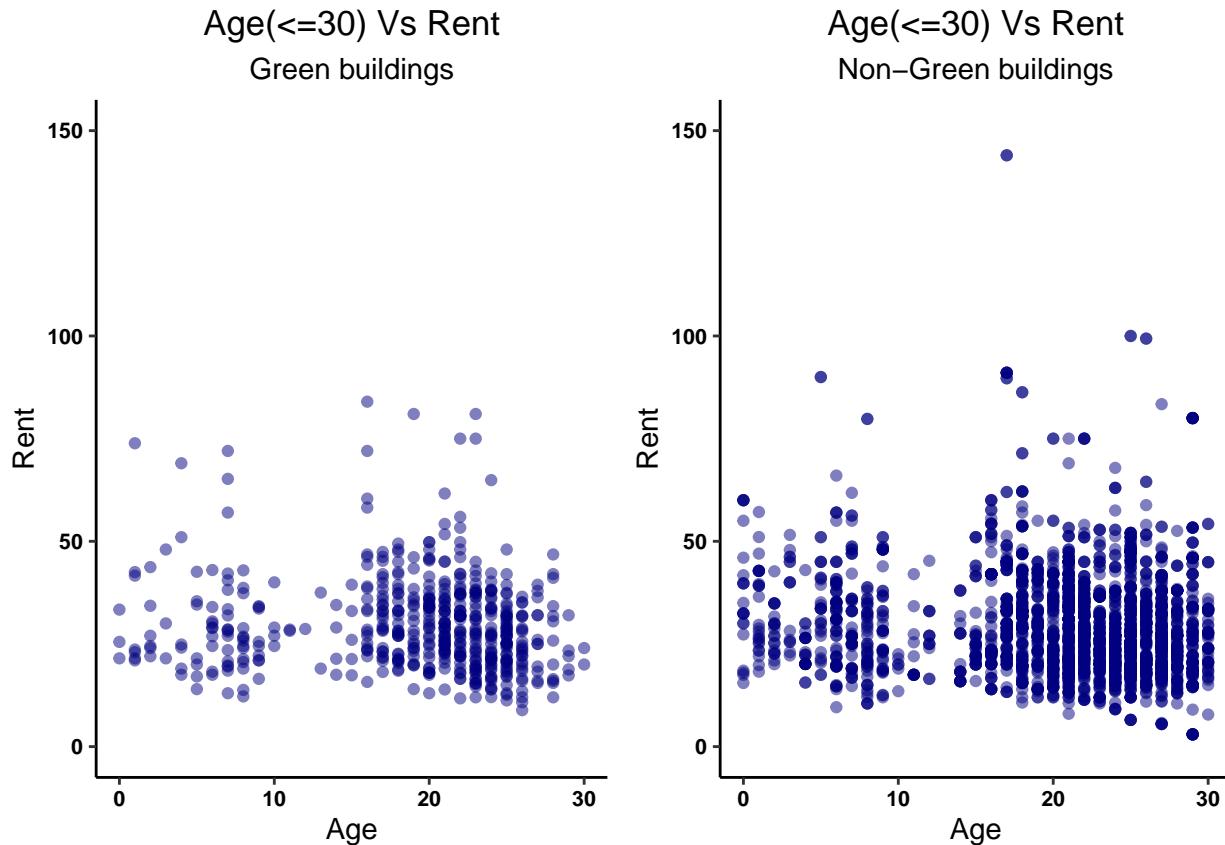
# Age plots and rent

rent_g_by_age = ggplot(green_buildings_by_age,aes(x = age,y = Rent)) +
  ylim(0,150) + theme_classic()+geom_point(colour = "navyblue", size = 1.5,
  alpha = 0.5)+ labs(x = "Age", y = "Rent", title = "Age(<=30) Vs Rent",
  subtitle = "Green buildings") + theme(axis.text.x = element_text(face="bold",
  color="black", size=8, angle=0), axis.text.y = element_text(face="bold",
  color="black", size=8, angle=0),plot.title = element_text(hjust = 0.5),
  plot.subtitle = element_text(hjust = 0.5))

rent_ng_by_age = ggplot(non_green_buildings_by_age,aes(x = age,y = Rent)) +
  ylim(0,150) + theme_classic()+geom_point(colour = "navyblue", size = 1.5,
  alpha = 0.5)+ labs(x = "Age", y = "Rent",title = "Age(<=30) Vs Rent",
  subtitle = "Non-Green buildings") + theme(axis.text.x = element_text(
  face="bold",color="black", size=8, angle=0), axis.text.y = element_text(
  face="bold", color="black", size=8, angle=0),plot.title = element_text(
  hjust = 0.5),plot.subtitle = element_text(hjust = 0.5))

grid.arrange(rent_g_by_age,rent_ng_by_age,ncol = 2)

```



```

print(paste("Median rent of green buildings less than 30 years of age:",
median(green_buildings_by_age$Rent)))

## [1] "Median rent of green buildings less than 30 years of age: 28"

print(paste("Median rent of non - green buildings less than 30 years of age:",
median(non_green_buildings_by_age$Rent)))

## [1] "Median rent of non - green buildings less than 30 years of age: 27"

# Observations:

# We see that even the age is not a primary factor since rent of green buildings
# is just higher across ages as well.

# After diving into all the separate variables, and also considering the green vs
# non- green scenario, it is evident that people are willing to pay more rent
# based on the green building parameter

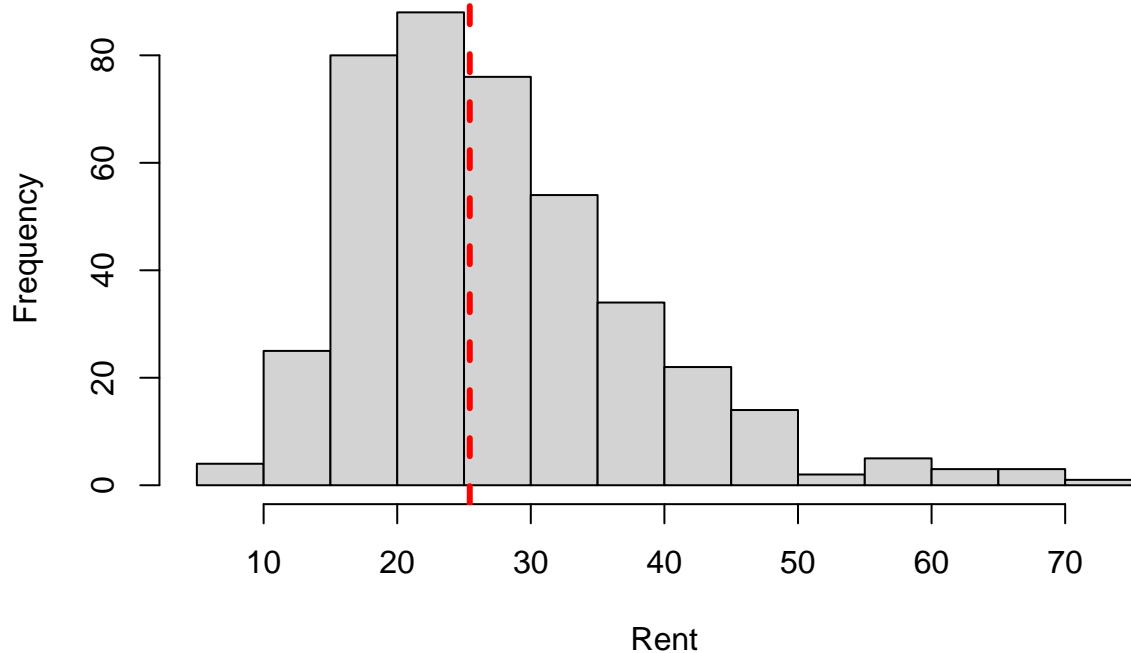
# Estimating Returns

# Consider a local market cluster

hist(unique(df$cluster_rent),main = paste("Histogram of Cluster rent"),xlab = 'Rent')
abline(v = median(unique(df$cluster_rent)), col="red", lwd=3, lty=2)

```

Histogram of Cluster rent



```
# Calculation the no. of local markets where rent for green building is greater  
# than median cluster (Considering median as it is more robust to outliers)  
  
cluster_quants = ddply(df,.(cluster), function(x)quantile(x$Rent))[c('cluster','50%')]  
temp = merge(green_buildings,cluster_quants,by = 'cluster')  
more_rent_green = subset(temp,temp$'Rent' >= temp$'50%')  
less_rent_green = subset(temp,temp$'Rent' < temp$'50%')  
  
# Observations:  
# Green building rent > Median rent -> for more than 75% of local markets  
  
# If we consider the mean of difference bw green and median local market's rent,  
# we see that green buildings get ~$3 more than non-green  
  
# Adjusting the estimates of the stats guru, by 0.4 , we can see that an extra  
# $750,000 revenue can be earned by building a green building.  
  
# Based on the extra revenue, we can recuperate the costs in 6.66 years and  
# even with 90% occupancy as is evident from data, the builder can start earning  
# profits after 7.4 years
```

Visual story telling part 2: Capital Metro data

Portfolio Modeling

```
library(mosaic)
library(quantmod)
library(foreach)
```

Low Risk ETFs : All Fixed Income ETFs - We are first going to create a low risk portfolio. For this purpose, we are going to select three specific type of ETFs:

USMV: Consists of US stocks with less risk. The Expense Ratio for this ETF is extremely low (about 15 cents for a \$100 investment) More than 25% of this fund is exposed to Information Technology stocks which has grown almost similarly to the S&P index. It also has heavy exposure to sectors like Healthcare and Telecommunications which are not traditionally volatile.

EFAV: This ETF enables us to achieve some geographic diversification as it has exposure to Europe and Asia. The Net Expense ratio is 0.20 for this ETF. It has exposure to Healthcare and Consumer Staples, again two non-volatile sectors.

EEMV: We will achieve further geographic diversity as this ETF has holdings from emerging markets like China, India and Taiwan. This has exposure to the Financial Sector and will help us in diversifying away from the Healthcare sector.

```
## Suppress some useleess warnings
defaultW <- getOption("warn")
options(warn = -1)

# Import a few stocks
low_risk_stocks = c("USMV", "EFAV", "EEMV")
quantmod::getSymbols(low_risk_stocks, from = "2016-08-01", to = Sys.Date(),)

## [1] "USMV" "EFAV" "EEMV"

# Adjust for splits and dividends
USMVa = quantmod::adjustOHLC(USMV)
EFAVa = quantmod::adjustOHLC(EFAV)
EEMVa = quantmod::adjustOHLC(EEMV)
```

Let us now try to find out the volatility of these stocks.

```
all_returns = cbind(C1C1(USMVa),
                     C1C1(EFAVa),
                     C1C1(EEMVa))
all_returns = as.matrix(na.omit(all_returns))
head(all_returns)

##          C1C1.USMVa   C1C1.EFAVa   C1C1.EEMVa
## 2016-08-02 -0.004471870  0.0000000000 -0.004709872
## 2016-08-03 -0.007486588 -0.0041408901  0.002649990
## 2016-08-04  0.003017220 -0.0011879122  0.002643005
```

```

## 2016-08-05 0.003008143 0.0001486024 0.007154980
## 2016-08-08 -0.002570673 -0.0031217335 0.005234586
## 2016-08-09 0.002147745 0.0062631228 0.004649433

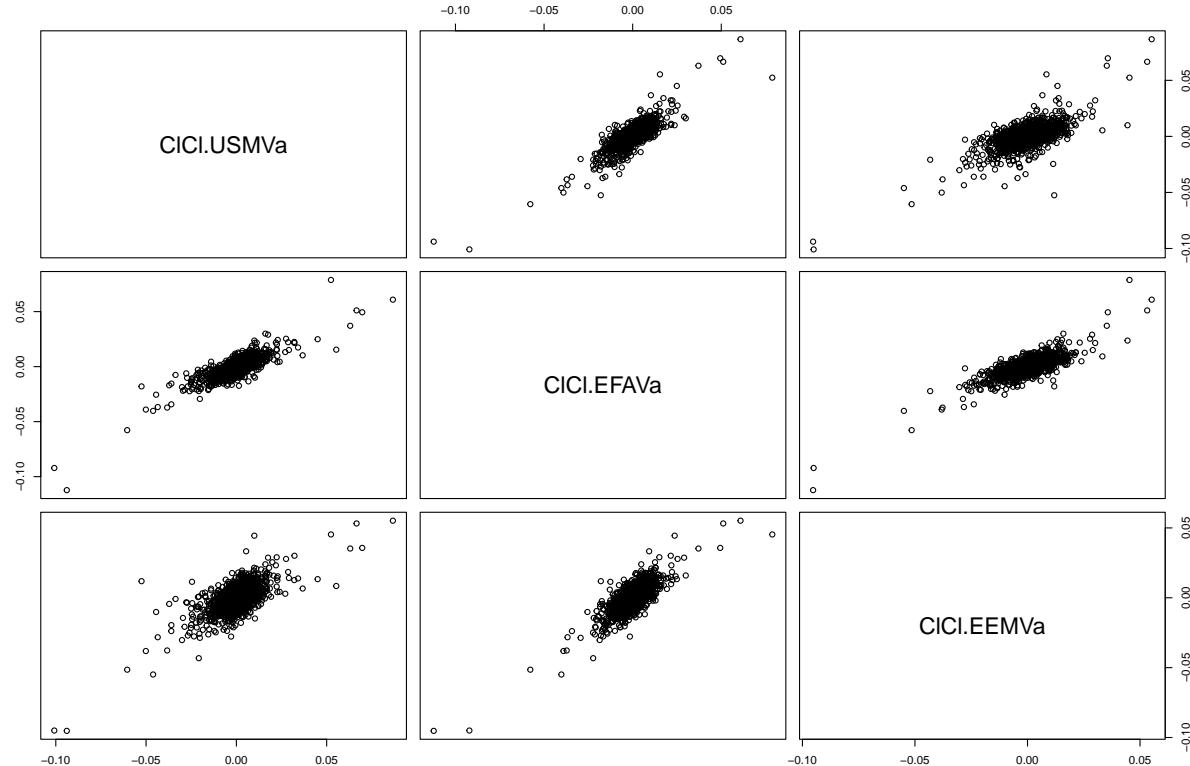
```

We can plot some scatter plots to check if there is a strong relationship between these 3 ETFs.

```

## There is not very significant linear relationship between these ETFs
pairs(all_returns)

```



```

## We will apportion our total wealth in the ratio 25:35:40, putting more weight on the EEMVa ETF that ...
set.seed(0)
# Now simulate many different possible futures
# just repeating the above block thousands of times
initial_wealth = 100000
sim1 = foreach(i=1:5000, .combine='rbind') %do% {
  total_wealth = initial_wealth
  weights = c(0.25, 0.35, 0.40)
  holdings = weights * total_wealth
  n_days = 20
  wealthtracker = rep(0, n_days)
  for(today in 1:n_days) {
    {
      return.today = resample(all_returns, 1, orig.ids=FALSE)
      holdings = holdings + holdings*return.today
      total_wealth = sum(holdings)
      wealthtracker[today] = total_wealth
    }
  }
}

```

```

    holdings = weights * total_wealth
}
wealthtracker
}

```

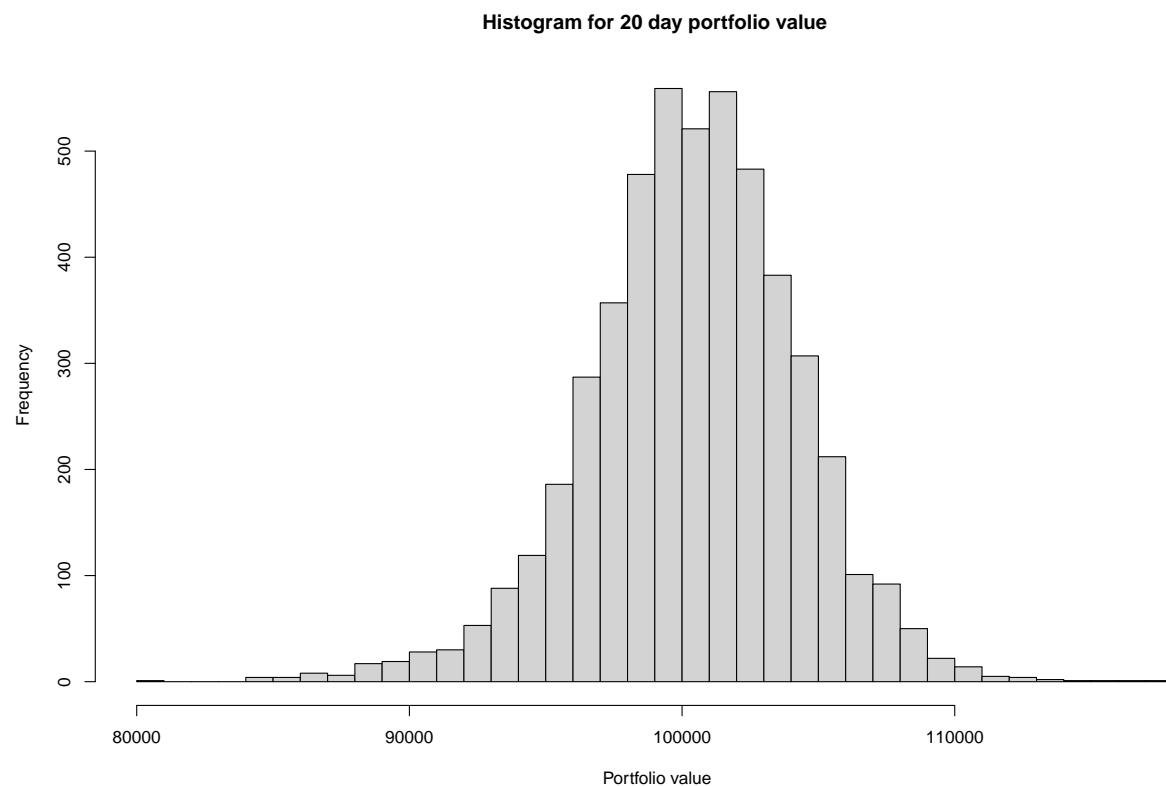
Let's look at the distribution plot by plotting a histogram

```

# Let's look at the distribution of total holdings value at the end of 20 days

hist(sim1[,20], 30, main = "Histogram for 20 day portfolio value", xlab = " Portfolio value")

```



```

# 5% VAR
quantile(sim1[,20] - initial_wealth, prob=0.05)

```

```

##          5%
## -6041.996

```

High Risk ETFs : All Equity ETFs We shall take all Equity ETFs for this purpose.

We will follow the same strategy as above for diversifying our portfolio.

AAXJ: This is an equity ETF with heavy exposure to emerging markets like China, India, Taiwan, Singapore, etc. It has very a diverse sector and industry composition ranging from E-Commerce to Electronics

BBEU: This is an extremely volatile fund. The returns on this fund has been very good over the last couple years. This fund has heavy exposure to Europe and will improve our portfolio diversification as it

has exposure to sectors like Healthcare and Consumer Staples. Moreover, this fund is mostly concentrated into healthcare and consumer staple that will add industry diversity to our portfolio

XLF: This fund focuses on the Financial and Real Estate behemoths of the US.

```
## Suppress some useless warnings
defaultW <- getOption("warn")
options(warn = -1)

# Import a few stocks
high_risk_stocks = c("AAXJ", "BBEU", "XLF")
quantmod::getSymbols(high_risk_stocks, from = "2016-08-01", to = Sys.Date(),)

## [1] "AAXJ" "BBEU" "XLF"

# Adjust for splits and dividends
AAXJa = quantmod::adjustOHLC(AAXJ)
BBEUa = quantmod::adjustOHLC(BBEU)
XLFa = quantmod::adjustOHLC(XLF)
```

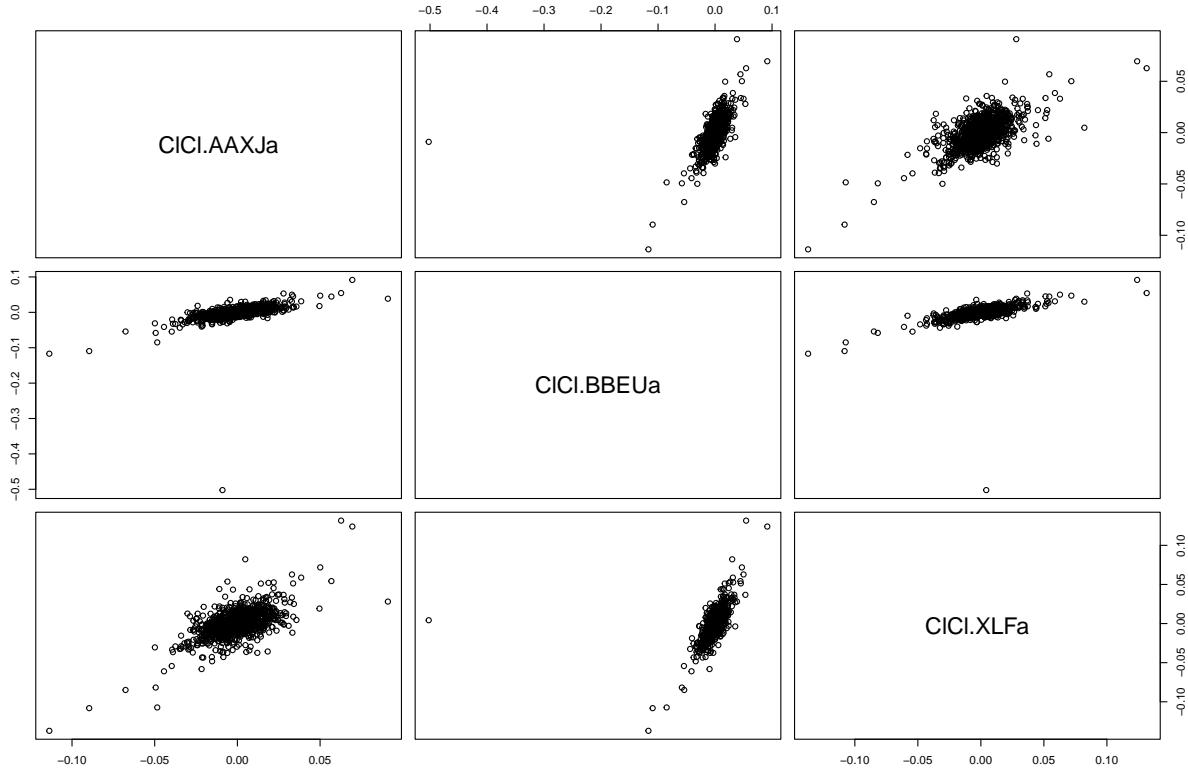
Let's now try to look at the volatility of these stocks

```
all_returns = cbind(C1C1(AAXJa),
                     C1C1(BBEUa),
                     C1C1(XLFa))
all_returns = as.matrix(na.omit(all_returns))
head(all_returns)

##           C1C1.AAXJa   C1C1.BBEUa   C1C1.XLFa
## 2018-06-19 -0.013058860  0.000000000 -0.002552881
## 2018-06-20  0.003403213 -0.004446281 -0.002559378
## 2018-06-21 -0.017094044 -0.008526147 -0.002932551
## 2018-06-22  0.009247840  0.000000000 -0.004779448
## 2018-06-25 -0.015043845  0.001760831 -0.010712929
## 2018-06-26 -0.002221550 -0.004210399 -0.003360717
```

Let's see if there is a strong relationship between these ETFs.

```
## There is not very significant linear relationship between these ETFs
pairs(all_returns)
```



```

## We will apportion our total wealth in the ratio 25:35:40, putting more weight on the AAXJ ETF that is
set.seed(0)
# Now simulate many different possible futures
# just repeating the above block thousands of times
initial_wealth = 100000
sim2 = foreach(i=1:5000, .combine='rbind') %do% {
  total_wealth = initial_wealth
  weights = c(0.25, 0.35, 0.40)
  holdings = weights * total_wealth
  n_days = 20
  wealthtracker = rep(0, n_days)
  for(today in 1:n_days) {
    {
      return.today = resample(all_returns, 1, orig.ids=FALSE)
      holdings = holdings + holdings*return.today
      total_wealth = sum(holdings)
      wealthtracker[today] = total_wealth
      holdings = weights * total_wealth
    }
  }
  wealthtracker
}

```

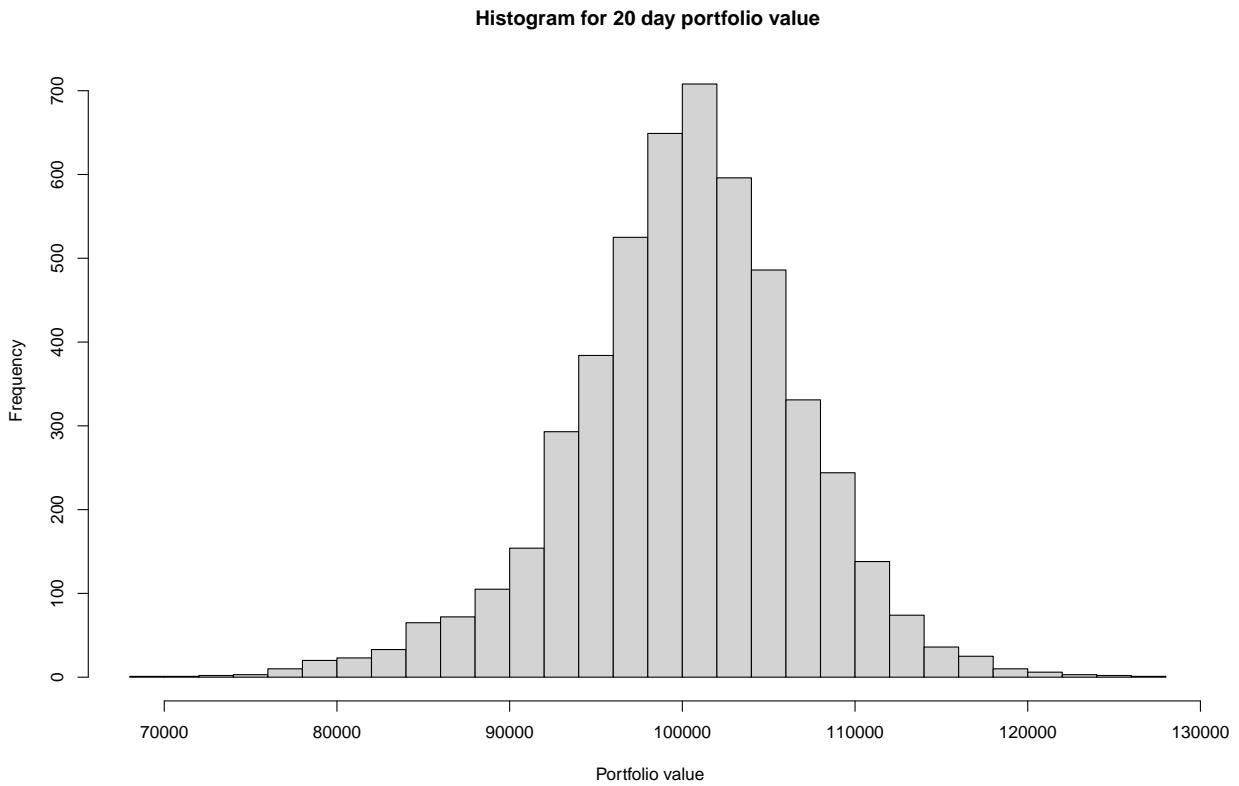
Let's look at the distribution plot by plotting a histogram.

```

# Let's look at the distribution of total holdings value at the end of 20 days

hist(sim2[,20], 30, main = "Histogram for 20 day portfolio value", xlab = " Portfolio value")

```



```
# 5% VAR
quantile(sim2[,20] - initial_wealth, prob=0.05)
```

```
##      5%
## -11476.36
```

Balanced ETFs : Mix of Equity and Fixed Income our strategy for this will be taking 3 funds:

AOR : This fund has a good mix of bonds and global stocks.

AOM: This fund is pretty similar to AOM

IYLD: This fund consists of 60% bonds, 20% equity. It is a good addition to a balanced portfolio.

```
## Suppress some useleess warnings
defaultW <- getOption("warn")
options(warn = -1)

# Import a few stocks
balanced_efts = c("AOR", "AOM", "IYLD")
quantmod::getSymbols(balanced_efts, from = "2016-08-01", to = Sys.Date(), )
```

```
## [1] "AOR"   "AOM"   "IYLD"
```

```
# Adjust for splits and dividends
AORa = quantmod::adjustOHLC(AOR)
AOMa = quantmod::adjustOHLC(AOM)
IYLDa = quantmod::adjustOHLC(IYLD)
```

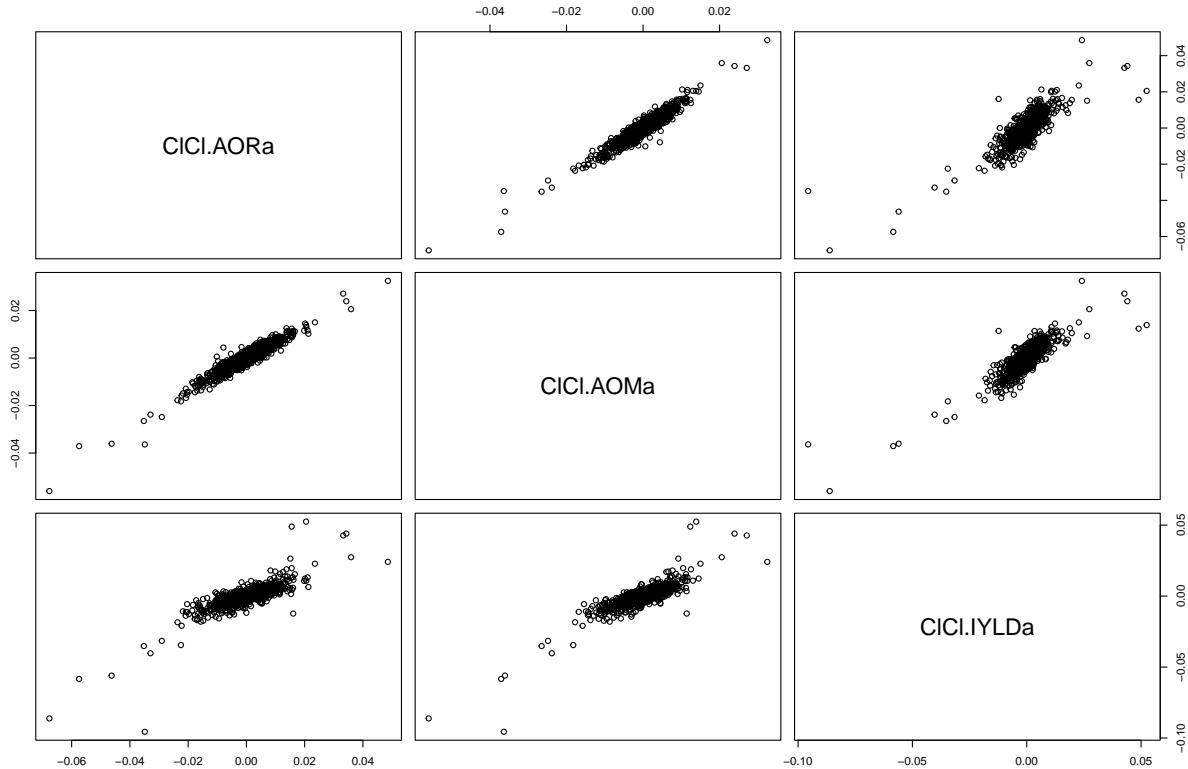
Let us now try to look at the volatility of these stocks.

```
all_returns = cbind(C1C1(AORa),
                     C1C1(AOMa),
                     C1C1(IYLDa))
all_returns = as.matrix(na.omit(all_returns))
head(all_returns)

##          C1C1.AORa    C1C1.AOMa    C1C1.IYLDa
## 2016-08-02 -0.0056469433 -0.006419313 -0.003134121
## 2016-08-03  0.0000000000  0.002528118  0.003224506
## 2016-08-04  0.0034567654  0.000000000  0.004017718
## 2016-08-05  0.0036909942  0.003922752  0.001200480
## 2016-08-08  0.0007354498 -0.001116411  0.001998401
## 2016-08-09  0.0034296668  0.004191059  0.005584324
```

Let us see if there is a strong relationship between these ETFs by plotting a few scatter plots.

```
## There is not very significant linear relationship between these ETFs
pairs(all_returns)
```



```
## We will apportion our total wealth in the ratio 25:35:40, putting more weight on the AAXJ ETF that is
set.seed(0)
# Now simulate many different possible futures
# just repeating the above block thousands of times
initial_wealth = 100000
```

```

sim3 = foreach(i=1:5000, .combine='rbind') %do% {
  total_wealth = initial_wealth
  weights = c(0.25, 0.35, 0.40)
  holdings = weights * total_wealth
  n_days = 20
  wealthtracker = rep(0, n_days)
  for(today in 1:n_days)
  {
    return.today = resample(all_returns, 1, orig.ids=FALSE)
    holdings = holdings + holdings*return.today
    total_wealth = sum(holdings)
    wealthtracker[today] = total_wealth
    holdings = weights * total_wealth
  }
  wealthtracker
}

```

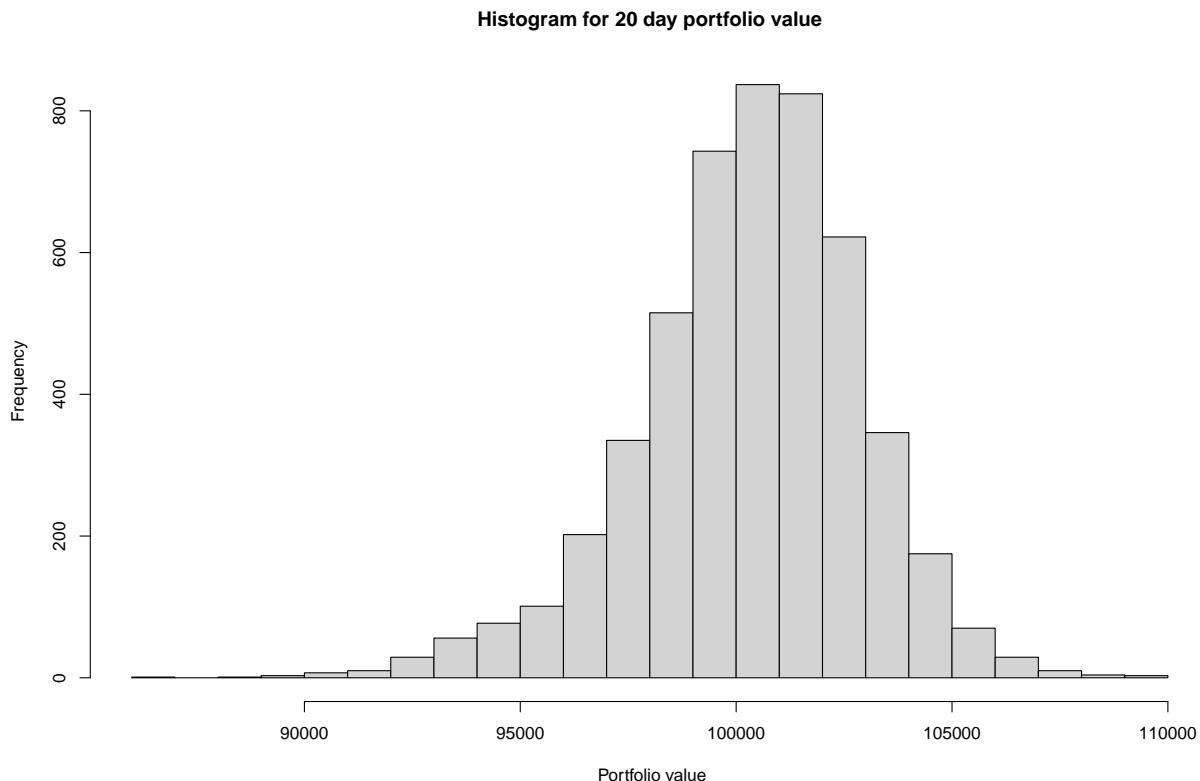
Let us look at the distribution plot by plotting a histogram.

```

# Let's look at the distribution of total holdings value at the end of 20 days

hist(sim3[,20], 30, main = "Histogram for 20 day portfolio value", xlab = " Portfolio value")

```



```

# 5% VAR
quantile(sim3[,20]- initial_wealth, prob=0.05)

```

```

##           5%
## -4223.054

library(ggplot2)
library(dplyr)

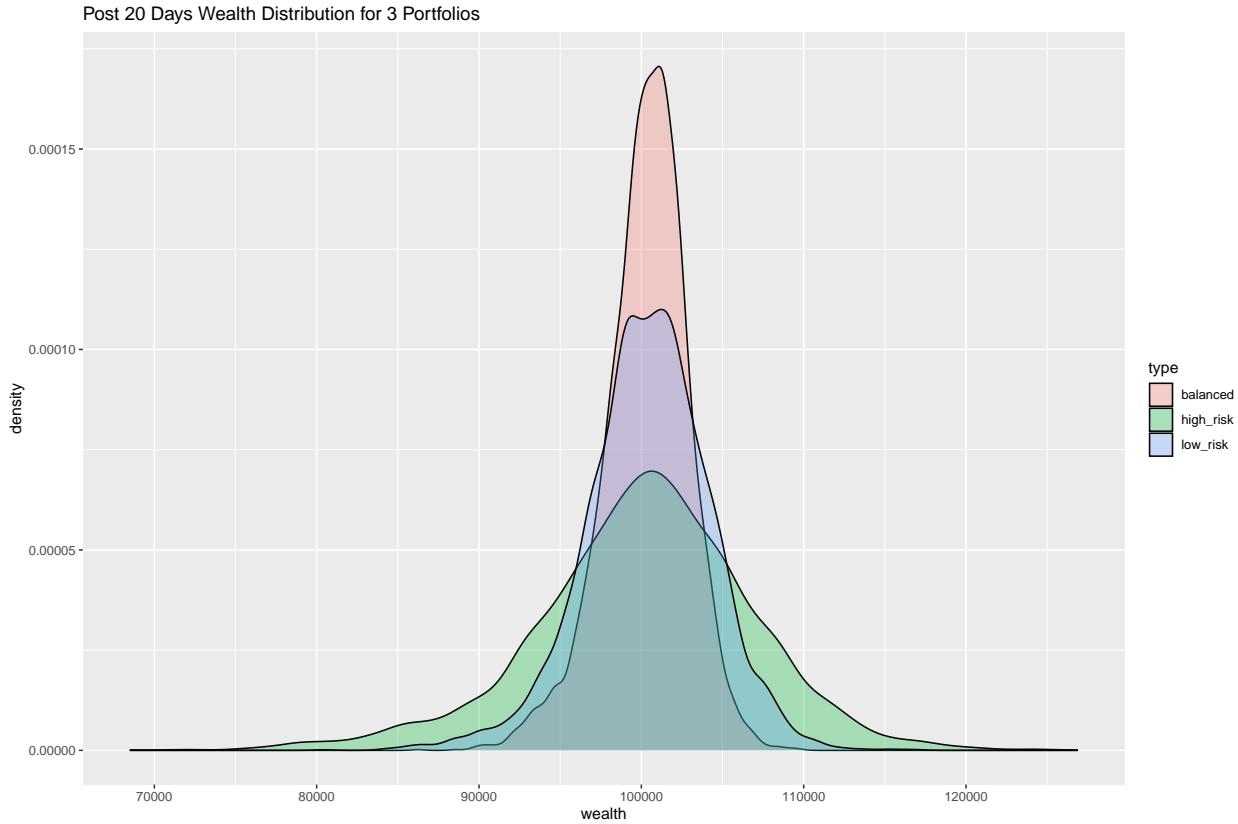
#Sample data

dat1 <- data.frame(wealth = sim1[,20],
                    type = 'low_risk')
dat2 <- data.frame(wealth = sim2[,20],
                    type = 'high_risk')
dat3 <- data.frame(wealth = sim3[,20],
                    type = 'balanced')

hist_df = rbind(dat1,dat2,dat3)

a <- ggplot(hist_df, aes(x = wealth))
a + geom_density(aes(fill = type), alpha=0.3) + ggtitle('Post 20 Days Wealth Distribution for 3 Portfolios')

```



The density plot for wealth after 20 days is mostly in line with our expectations. The high risk portfolio has more variance as expected and has highest VAR at about 5%. Ideally, the balanced portfolio's variance should have lied between the high and low risk portfolios but here we see some deviation from expected behavior.

This could be due to the EFT selections that we have made or building our portfolios.

Visual story telling part 2: Capital Metro data

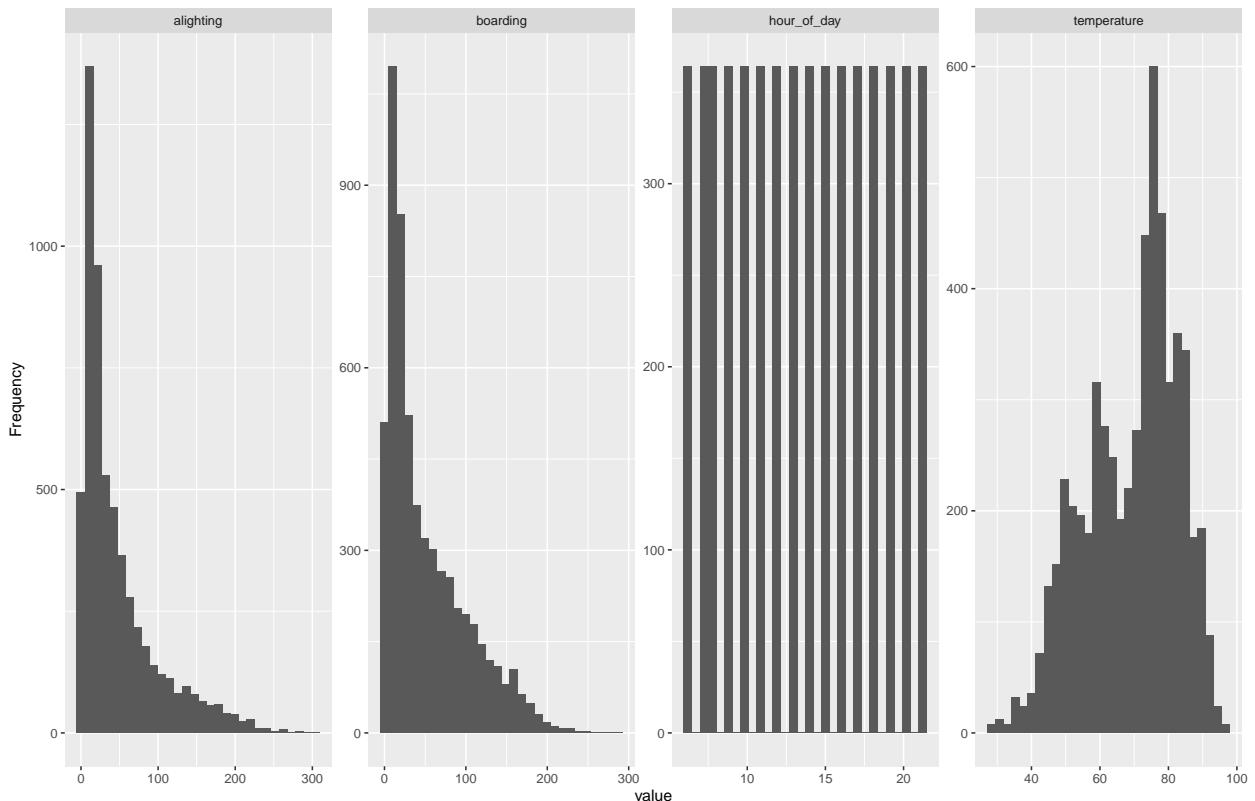
```
library(gridExtra)
library(ggplot2)
# library(dplyr)
library(tidyverse)
library(DataExplorer)

#detach(package:plyr, unload=TRUE)

metro <- read.csv("capmetro_UT.csv")

# Looking at some distributions

ncols <- dplyr::select_if(metro, is.numeric)
plot_histogram(ncols)
```



```
# Diving into riders analysis

library(tidyverse)

r1 = metro %>%
  group_by(hour_of_day) %>%
  dplyr::summarize(riders = mean(boarding))
```

```

r2 = metro %>%
      group_by(hour_of_day) %>%
      dplyr::summarize(riders = mean(alighting))

plot1 = ggplot(r1) + geom_line(aes(x=hour_of_day, y=riders)) +
      ggtitle("Number of people boarding")

plot2 = ggplot(r2) + geom_line(aes(x=hour_of_day, y=riders)) +
      ggtitle("Number of people alighting")

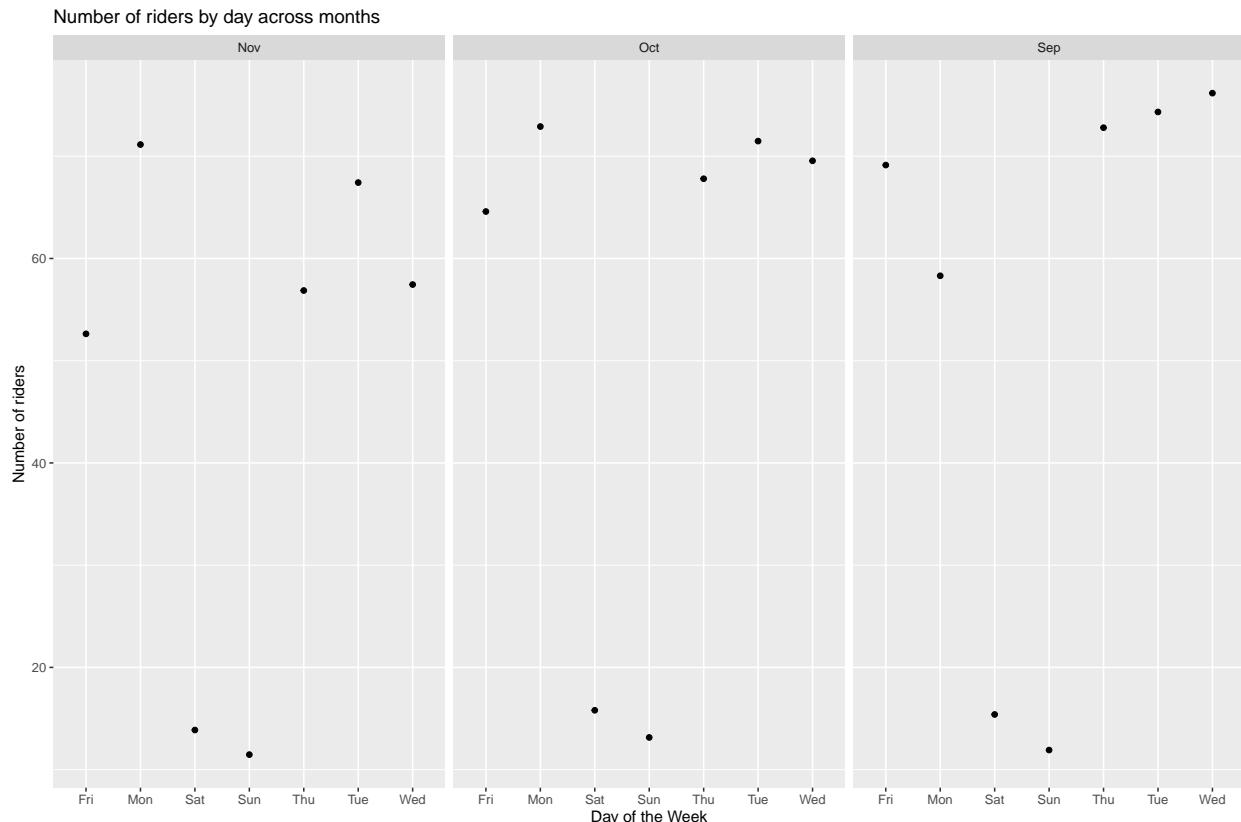
# grid.arrange(r1, r2, nrow = 1)

r3 = metro %>%
      group_by(day_of_week,month) %>%
      dplyr::summarize(riders = mean(boarding))

## `summarise()` has grouped output by 'day_of_week'. You can override using the
## '.groups' argument.

ggplot(r3) + geom_point(aes(x=day_of_week, y=riders)) + facet_wrap(~month) +
      labs(x="Day of the Week", y="Number of riders",
           title="Number of riders by day across months")

```



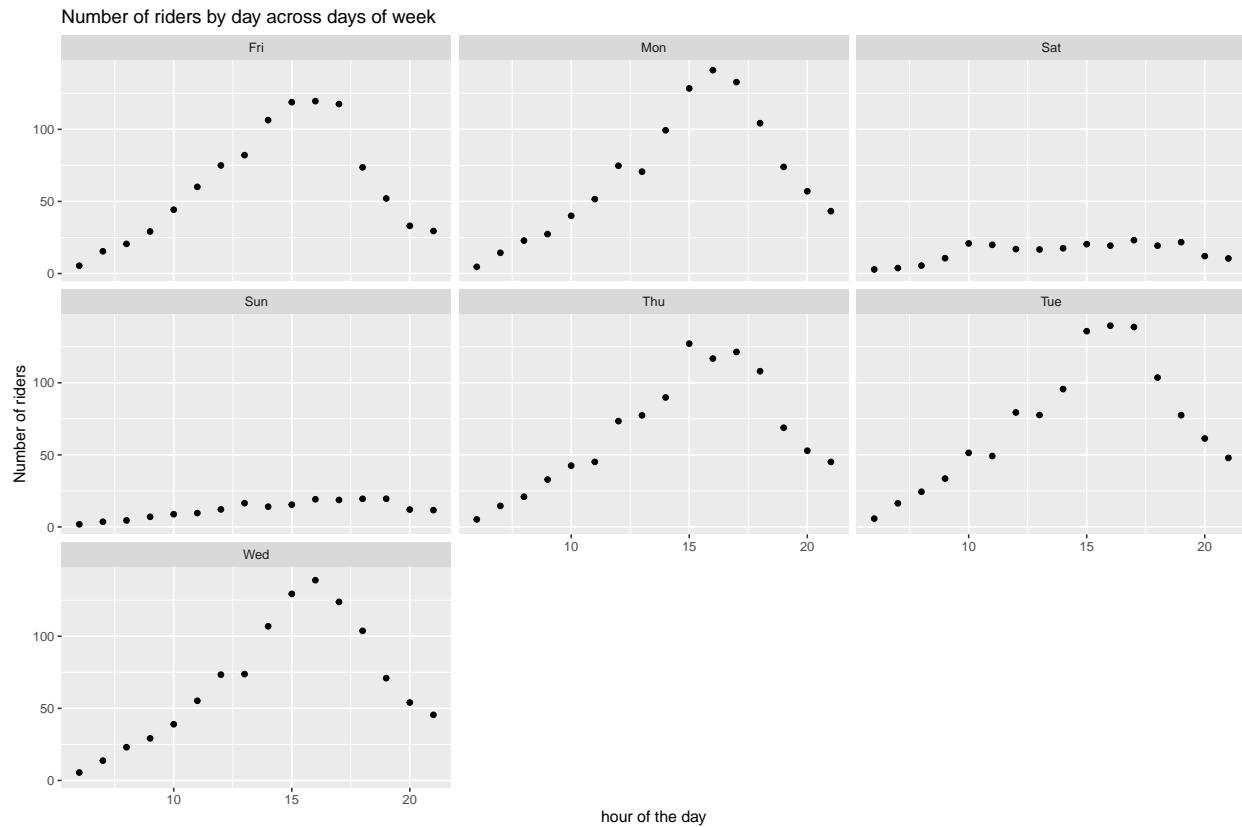
```

r4 = metro %>%
      group_by(hour_of_day, day_of_week) %>%
      dplyr::summarize(riders = mean(boarding))

```

```
## `summarise()` has grouped output by 'hour_of_day'. You can override using the
## `.groups` argument.
```

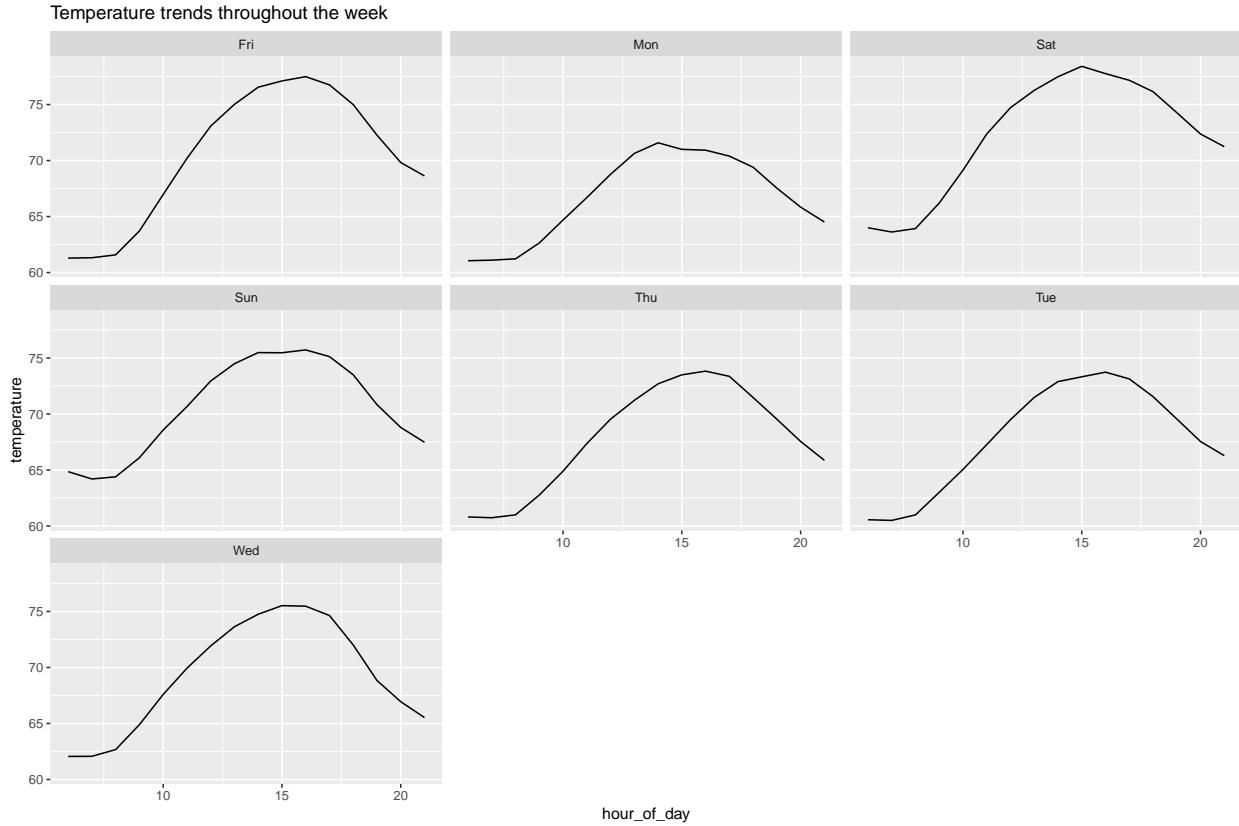
```
ggplot(r4) + geom_point(aes(x=hour_of_day, y=riders)) +
  facet_wrap(~day_of_week) + labs(x="hour of the day",
  y="Number of riders", title="Number of riders by day across days of week")
```



```
r5 = metro %>%
  group_by(day_of_week, hour_of_day) %>%
  dplyr::summarize(temperature = mean(temperature))
```

```
## `summarise()` has grouped output by 'day_of_week'. You can override using the
## `.groups` argument.
```

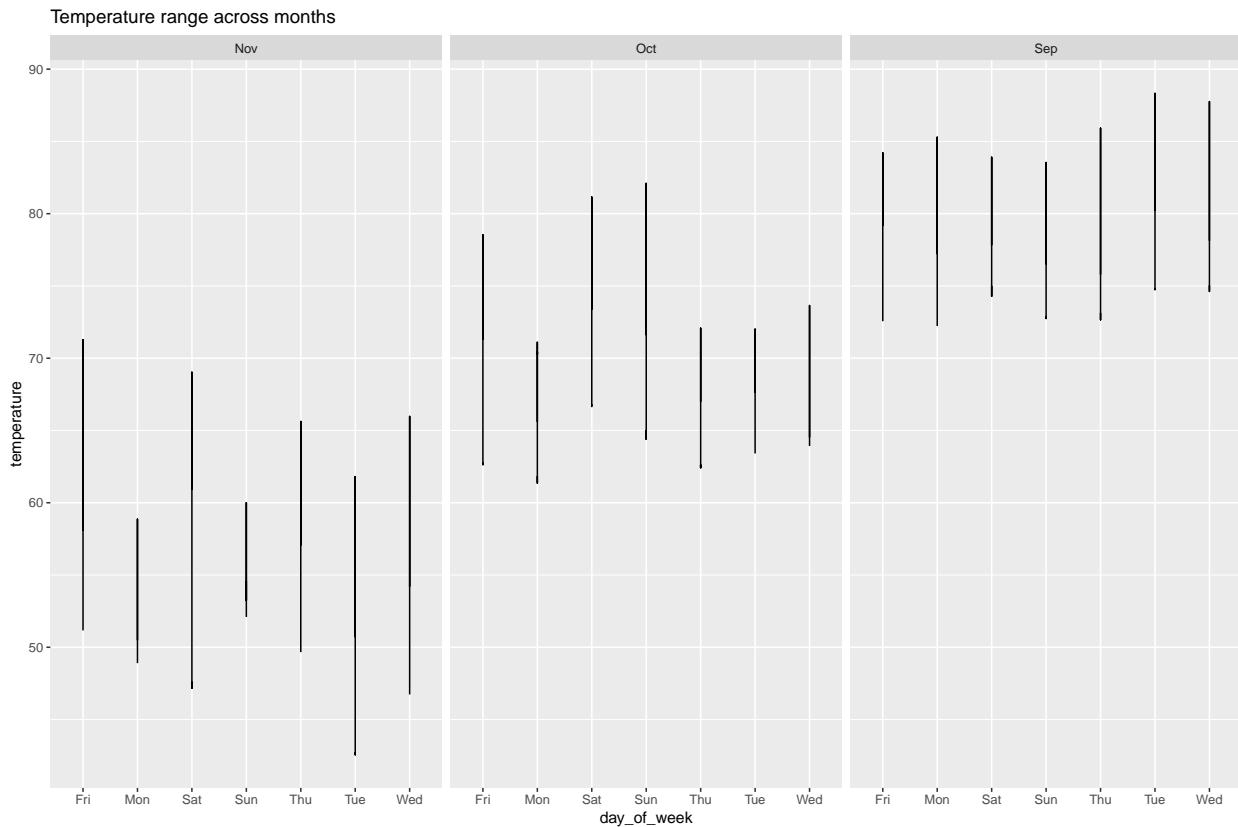
```
ggplot(r5) + geom_line(aes(x=hour_of_day, y=temperature)) +
  ggtitle("Temperature trends throughout the week") + facet_wrap(~day_of_week)
```



```
r6 = metro %>%
  group_by(day_of_week, hour_of_day, month) %>%
  dplyr::summarize(temperature = mean(temperature))
```

```
## `summarise()` has grouped output by 'day_of_week', 'hour_of_day'. You can
## override using the '.groups' argument.
```

```
ggplot(r6) +
  geom_line(aes(x=day_of_week, y=temperature)) +
  ggtitle("Temperature range across months") +
  facet_wrap(~month)
```



Clustering and PCA

```

library(ggplot2)
library(ClusterR)

## Loading required package: gtools

##
## Attaching package: 'gtools'

## The following object is masked from 'package:mosaic':
##      logit

library(foreach)
library(mosaic)

wine = read.csv('wine.csv', header=TRUE)

summary(wine)

##   fixed.acidity      volatile.acidity    citric.acid      residual.sugar

```

```

## Min.    : 3.800  Min.    :0.0800  Min.    :0.0000  Min.    : 0.600
## 1st Qu.: 6.400  1st Qu.:0.2300  1st Qu.:0.2500  1st Qu.: 1.800
## Median  : 7.000  Median  :0.2900  Median  :0.3100  Median  : 3.000
## Mean    : 7.215  Mean    :0.3397  Mean    :0.3186  Mean    : 5.443
## 3rd Qu.: 7.700  3rd Qu.:0.4000  3rd Qu.:0.3900  3rd Qu.: 8.100
## Max.    :15.900  Max.    :1.5800  Max.    :1.6600  Max.    :65.800
##      chlorides      free.sulfur.dioxide total.sulfur.dioxide      density
## Min.    :0.00900  Min.    : 1.00      Min.    : 6.0      Min.    :0.9871
## 1st Qu.:0.03800  1st Qu.:17.00     1st Qu.: 77.0     1st Qu.:0.9923
## Median  :0.04700  Median  :29.00     Median  :118.0     Median  :0.9949
## Mean    :0.05603  Mean    :30.53     Mean    :115.7     Mean    :0.9947
## 3rd Qu.:0.06500  3rd Qu.:41.00     3rd Qu.:156.0     3rd Qu.:0.9970
## Max.    :0.61100  Max.    :289.00    Max.    :440.0     Max.    :1.0390
##      pH          sulphates      alcohol      quality
## Min.    :2.720    Min.    :0.2200    Min.    : 8.00    Min.    :3.000
## 1st Qu.:3.110    1st Qu.:0.4300    1st Qu.: 9.50    1st Qu.:5.000
## Median  :3.210    Median  :0.5100    Median  :10.30    Median  :6.000
## Mean    :3.219    Mean    :0.5313    Mean    :10.49    Mean    :5.818
## 3rd Qu.:3.320    3rd Qu.:0.6000    3rd Qu.:11.30    3rd Qu.:6.000
## Max.    :4.010    Max.    :2.0000    Max.    :14.90    Max.    :9.000
##      color
## Length:6497
## Class :character
## Mode   :character
##
##
##
```

```
# Center and scale the data
```

```
X = wine[,-(12:13)]
X = scale(X, center=TRUE, scale=TRUE)
```

```
# trying 2 Prinicipal Components
```

```
wine$Good_Quality_Wine<-ifelse(wine$quality>5,"Good Quality","Bad Quality")
dim(wine)
```

```
## [1] 6497 14
```

```
pca1 = prcomp(wine[,1:12], scale. = TRUE, rank = 2)
summary(pca1)
```

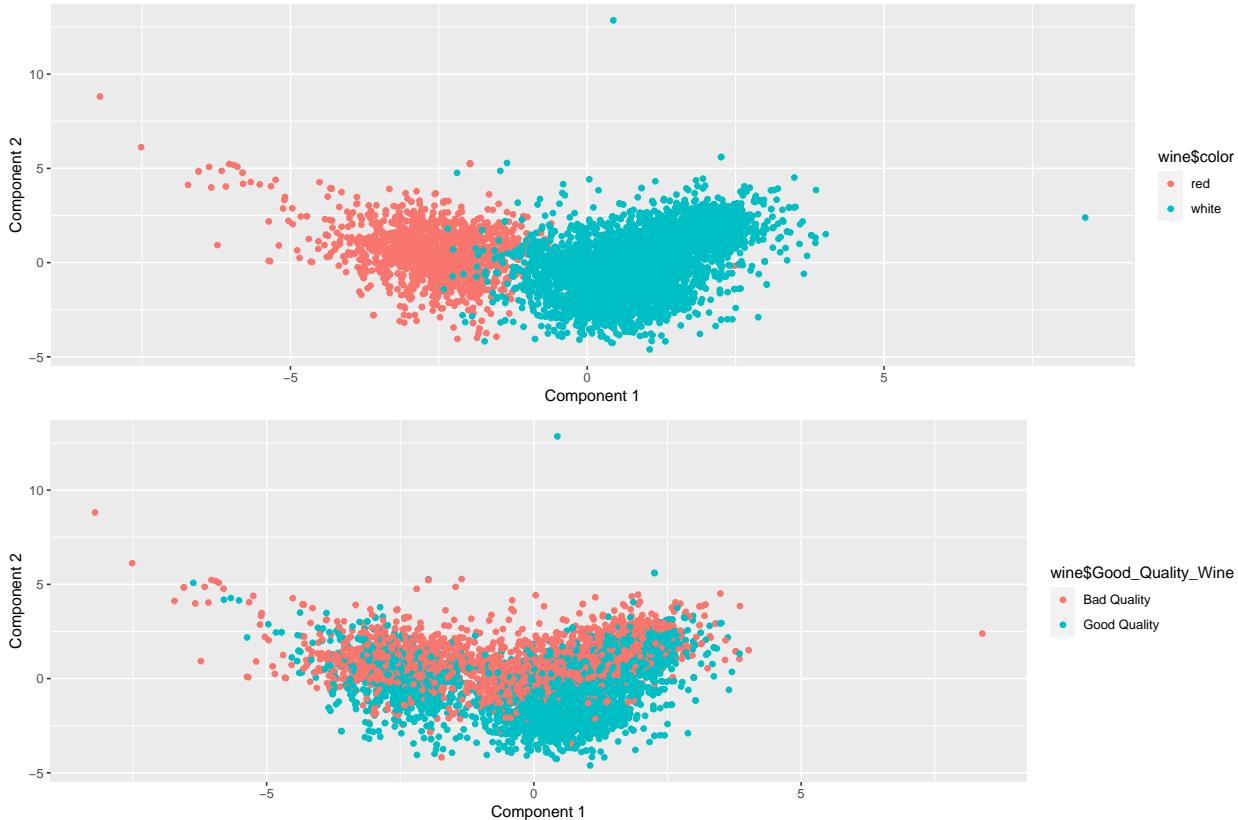
```
## Importance of first k=2 (out of 12) components:
##                  PC1      PC2
## Standard deviation 1.7440 1.6278
## Proportion of Variance 0.2535 0.2208
## Cumulative Proportion 0.2535 0.4743
```

```
loadings = pca1$rotation
scores = pca1$x
```

```

plot1<-qplot(scores[,1], scores[,2], color=wine$color,
              xlab='Component 1', ylab='Component 2')
plot2<-qplot(scores[,1], scores[,2], color=wine$Good_Quality_Wine,
              xlab='Component 1', ylab='Component 2')
grid.arrange(plot1, plot2, nrow = 2)

```



```

# trying 2 Clusters

wine$good<-ifelse(wine$quality>5,"blue","red")
wine$color2<-ifelse(wine$color=="red","red","blue")

X = wine[,1:12]
X = scale(X, center=TRUE, scale=TRUE)

mu = attr(X,"scaled:center")
sigma = attr(X, "scaled:scale")

clust1 = kmeans(X, 2, nstart=25)
clust1$center

##   fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1    -0.2833598      -0.4002457   0.1163397     0.2036791 -0.3133076
## 2     0.8203503       1.1587448  -0.3368131    -0.5896682  0.9070517
##   free.sulfur.dioxide total.sulfur.dioxide density      pH sulphates
## 1      0.2875184        0.4055994 -0.2316456 -0.1918762 -0.2847052
## 2     -0.8323898       -1.1742444  0.6706333  0.5554976  0.8242456

```

```

##      alcohol      quality
## 1  0.02984993  0.09700235
## 2 -0.08641804 -0.28082994

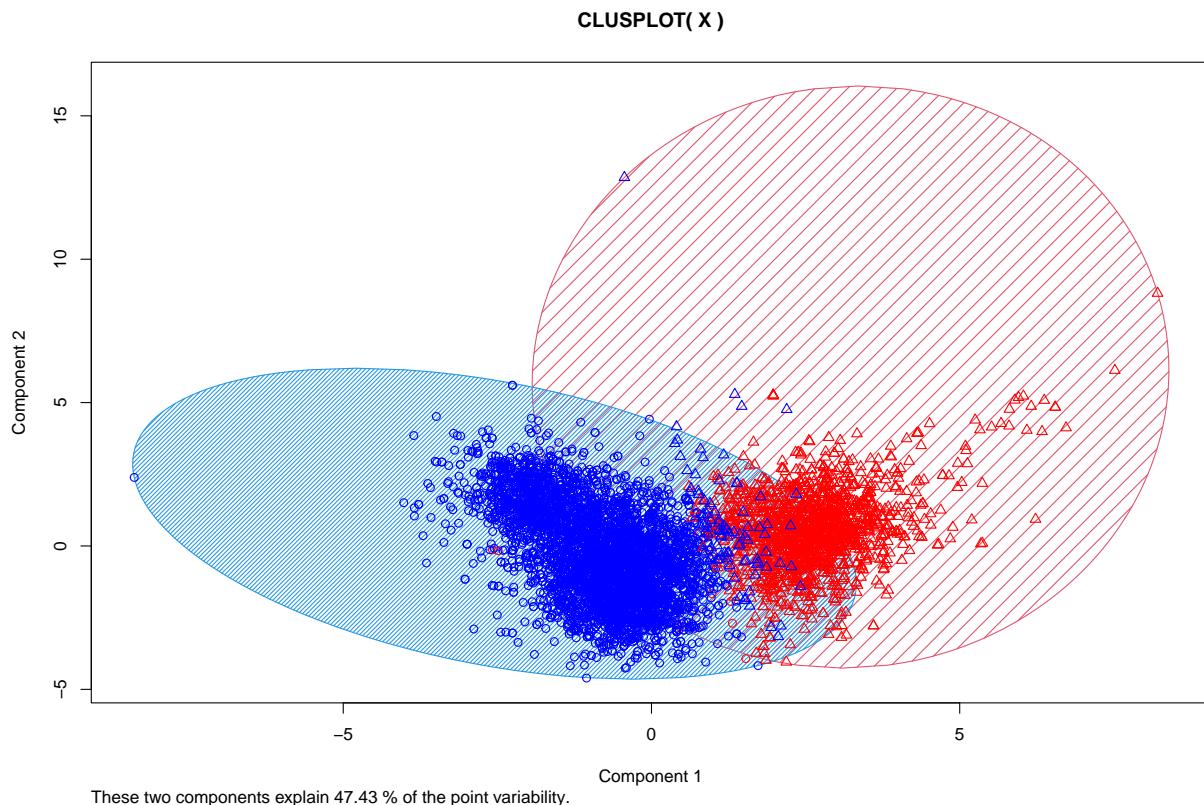
library(cluster)
library(fpc)
library(ppclust)

##
## Attaching package: 'ppclust'

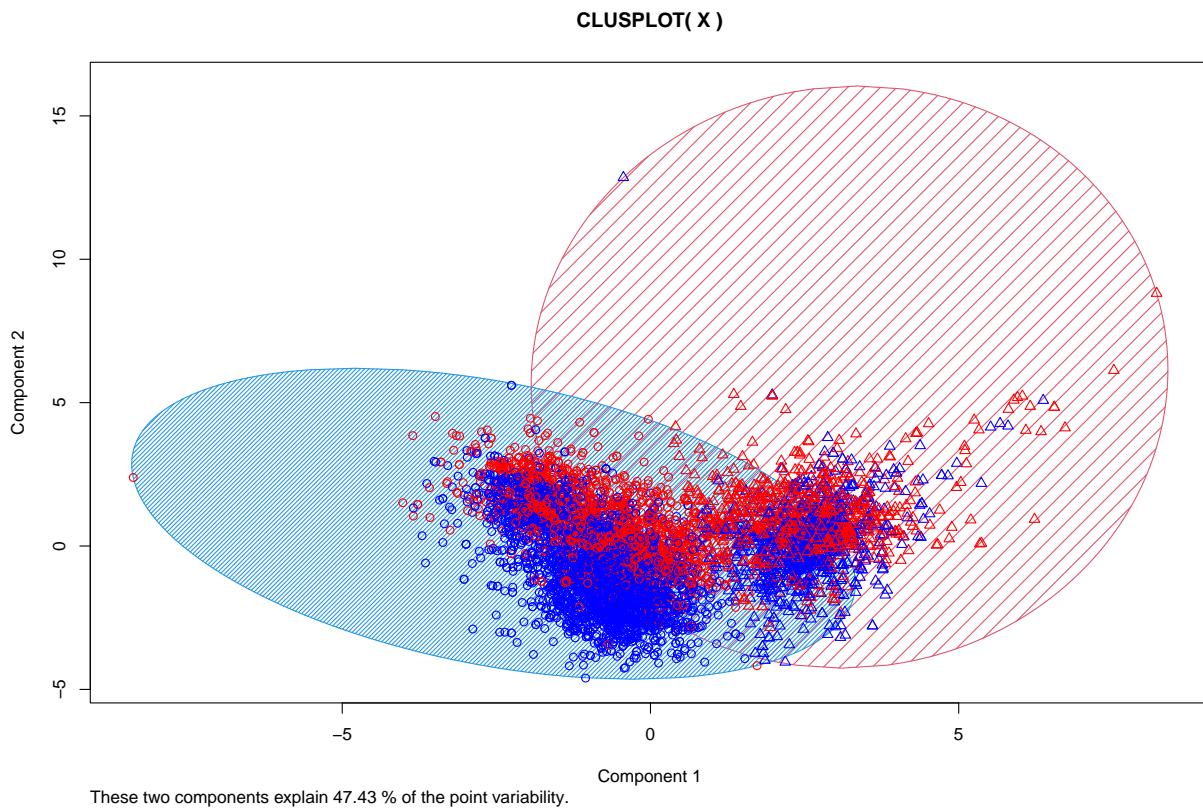
## The following object is masked from 'package:fpc':
## 
##     plotcluster

clusplot(X, clust1$cluster, color=TRUE, shade=TRUE, plotchar=TRUE, col.p=wine$color2)

```



```
clusplot(X, clust1$cluster, color=TRUE, shade=TRUE, plotchar=TRUE, col.p=wine$good)
```



Market Segmentation

Data Loading and Processing

```

rm(list=ls())
library(dplyr)
library(RCurl)

## 
## Attaching package: 'RCurl'

## The following object is masked from 'package:tidyverse':
## 
##     complete

df = read.csv("social_marketing.csv")
column_names <- colnames(df)
column_names[1] <- "user_id"
colnames(df) <- column_names
row.names(df) = df$user_id
rawdf = df %>% select(-c(user_id))
df = rawdf

```

Data Exploration Phase - Let us first take a cursory look at the different tweet categories to figure out what tags are being used in our dataset:

```
c(colnames(df))

## [1] "chatter"          "current_events"    "travel"           "photo_sharing"
## [5] "uncategorized"   "tv_film"          "sports_fandom"    "politics"
## [9] "food"             "family"           "home_and_garden" "music"
## [13] "news"             "online_gaming"    "shopping"         "health_nutrition"
## [17] "college_uni"     "sports_playing"   "cooking"          "eco"
## [21] "computers"        "business"         "outdoors"         "crafts"
## [25] "automotive"       "art"               "religion"         "beauty"
## [29] "parenting"        "dating"            "school"           "personal_fitness"
## [33] "fashion"          "small_business"   "spam"              "adult"
```

We know that the variables: **chatter**, **spam**, **uncategorized**, and **adult** are non useful tweets.

As a first step, we can drop users whose tweets have been tagged under these non useful categories. Let us now check how much data we are left with post this removal.

We will call these categories as “**suspicious categories**”

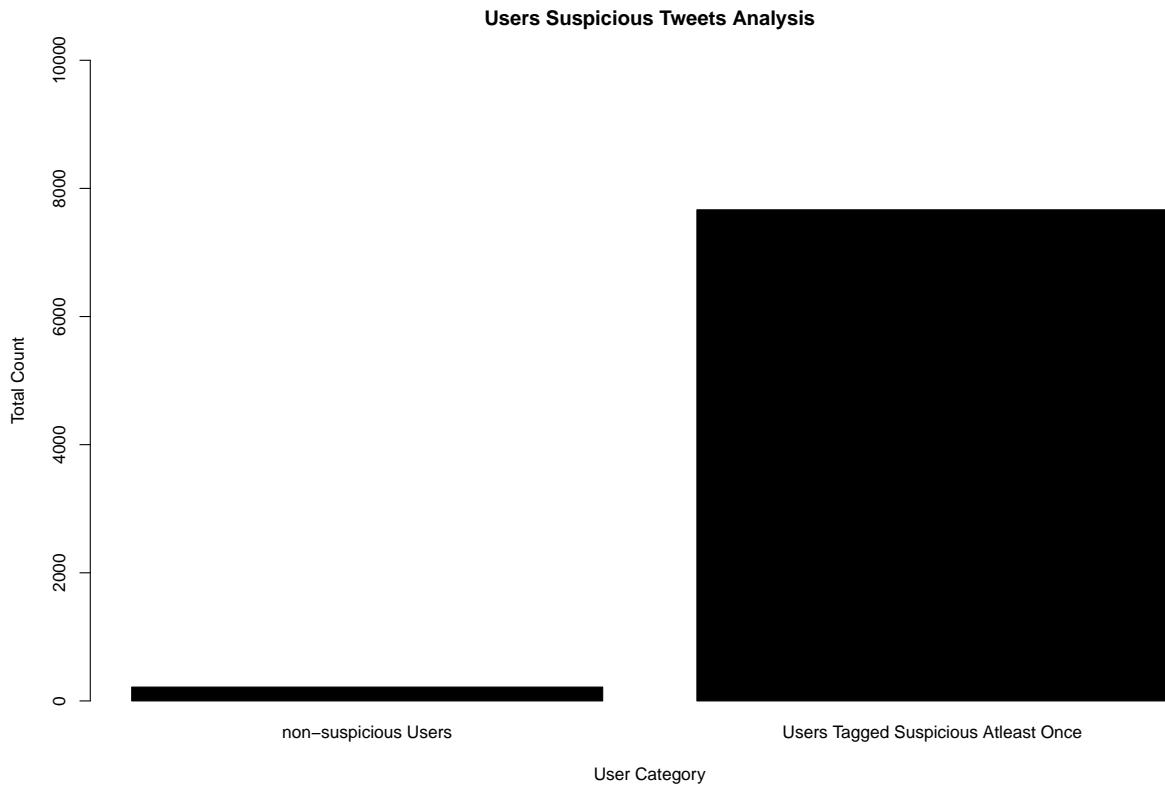
```
unwanted_cat <- c("spam", "chatter", "adult", "uncategorized")
suspicious_cat_flag = ifelse(rowSums(df %>% select(unwanted_cat)) > 0, 1, 0)
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use 'all_of(unwanted_cat)' instead of 'unwanted_cat' to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
df = cbind(df, suspicious_cat_flag)

intdf = df %>% group_by(suspicious_cat_flag) %>% dplyr::summarise(total_records = n())

bp = barplot(intdf$total_records, names.arg = c("non-suspicious Users", "Users Tagged Suspicious Atleast
                                               main = "Users Suspicious Tweets Analysis",
                                               col= 'Black',
                                               xlab = 'User Category',
                                               ylab = 'Total Count',
                                               ylim = c(0,10000))
```



So, there are only approx **2.7%** users whose tweets have never been categorized under the **suspicious category** as defined by us.

Clearly, we cannot simply delete such a huge number of users (~97%).

We'll define a metric - **suspicious %** as "the no. of Times user's tweet is tagged as suspicious / total tags". This will give the proportion of tweets labeled as suspicious for a particular user.

We can then use this metric to filter out users.

```
## Remove previously created flag
df = df %>% select(- c(suspicious_cat_flag))
suspicious_counts = rowSums(df %>% select(unwanted_cat))
total_counts = rowSums(df)
suspicious_pct = suspicious_counts/total_counts

df = df %>% mutate(suspicious_pct = suspicious_pct)

s1 = seq(0,1,0.0001)
s2 = c()
for (x in s1)
{
  s2 = append(s2, nrow(df %>% filter(suspicious_pct < x))/nrow(rawdf))
}

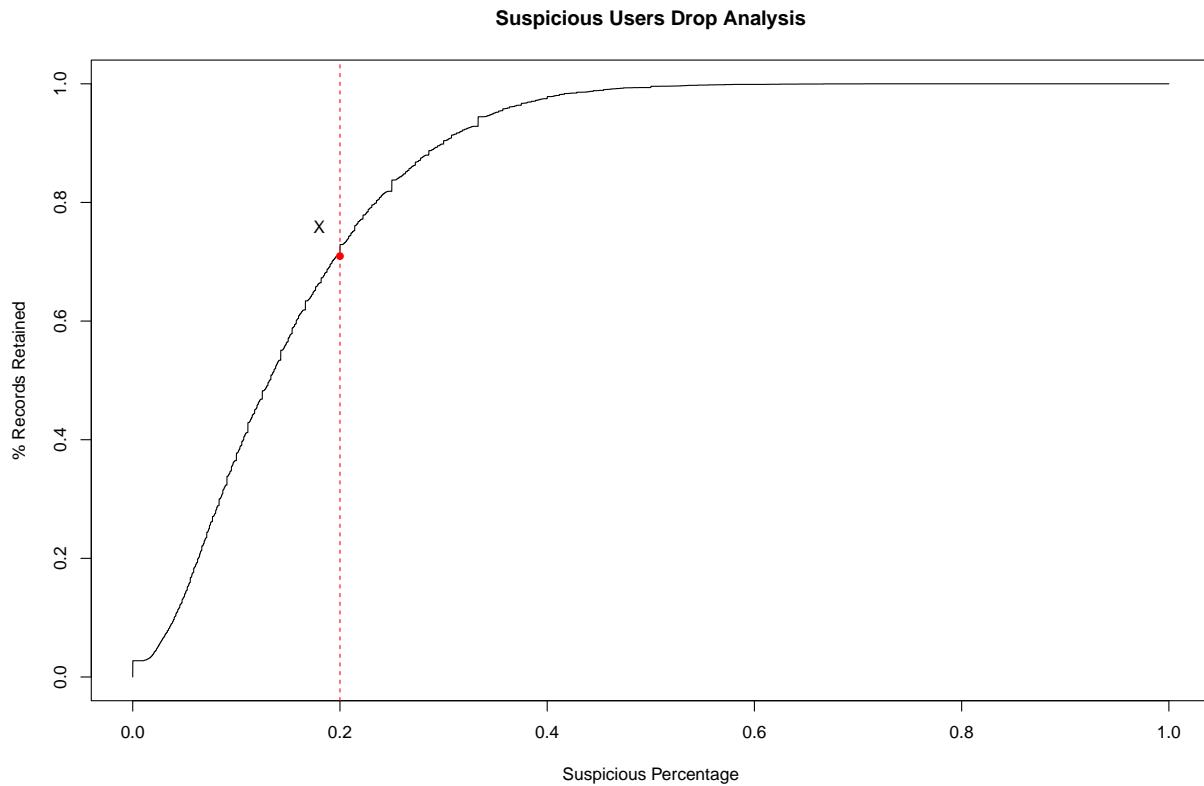
plot(x = s1,
      y = s2,
```

```

    type = 'l',
    xlab = 'Suspicious Percentage',
    ylab = '% Records Retained',
    main = 'Suspicious Users Drop Analysis'

)
abline(v = c(0.2), col = 'red', lty = 2)
points(x = 0.2, y = s2[2000], col = 'red', pch = 16)
text(x = 0.18, y = s2[2000]+0.05, "X")

```



We will set the filter threshold at 20%, meaning the users whose suspicious tweets is more than 20% of their total tweets will be filtered out from the dataset.

Threshold is marked by X in above graph. Let us filter the data and remove these unwanted categories

```

analysis_df = df %>% filter(suspicious_pct<0.2) %>% select(- append(unwanted_cat, 'suspicious_pct'))
print(paste0("We are now left with ", nrow(analysis_df), " rows"))

```

```
## [1] "We are now left with 5592 rows"
```

Analysis

Visualization

Let us try to plot a correlation matrix.

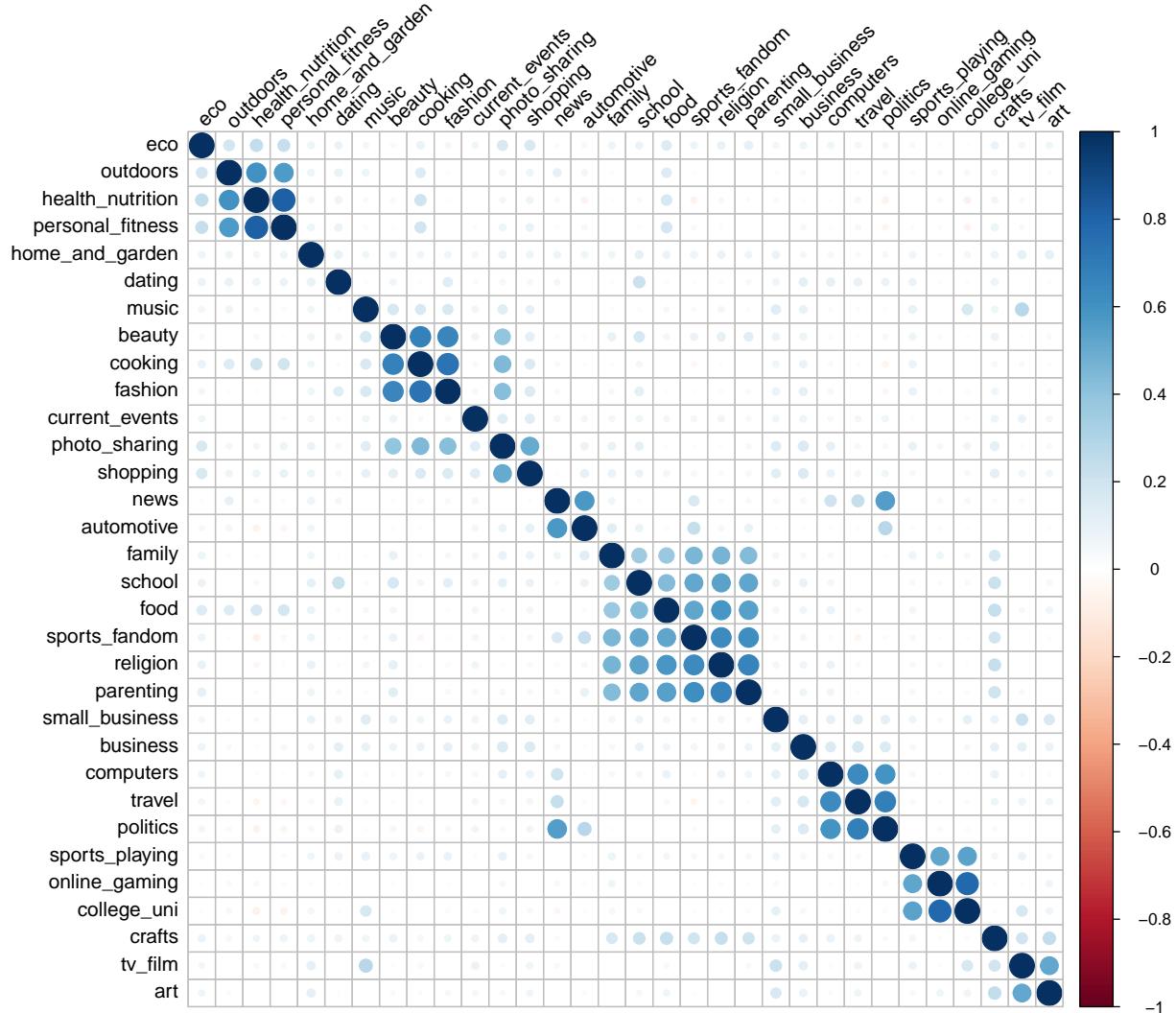
```

library(corrplot)

## corrplot 0.92 loaded

res <- cor(analysis_df)
corrplot(res, order = "hclust", tl.col = "black", tl.srt = 45)

```



Immediately on a cursory glance, we can see 6 clusters in the plot above:

- *Fitness Focused:* Eco, Outdoor, Health_Nutrition, Personal_Fitness
- *Social Media Influencers:* Beauty, Cooking, Fashion

- *Family People* : Family, School, Food, Sports Fandom, Religion, Parenting
- *Geeky traveller*: Computer, Travel, Politics - *Student*: Sports Playing, Online Gaming, College University
- There is a ‘not so highly correlated cluster of’ art, crafts, and TV film as well

Let us try to reduce the dimensions using Principal Component Analysis

```
Z = scale(analysis_df, center=TRUE, scale=FALSE)
pc_Z = prcomp(Z, rank=10)

summary(pc_Z)

## Importance of first k=10 (out of 32) components:
##          PC1      PC2      PC3      PC4      PC5      PC6      PC7
## Standard deviation 5.7539 4.4774 4.3170 4.1992 3.62528 2.44737 2.40030
## Proportion of Variance 0.2217 0.1343 0.1248 0.1181 0.08802 0.04011 0.03858
## Cumulative Proportion 0.2217 0.3560 0.4808 0.5989 0.68688 0.72699 0.76557
##          PC8      PC9      PC10
## Standard deviation 2.24587 1.79320 1.59931
## Proportion of Variance 0.03378 0.02153 0.01713
## Cumulative Proportion 0.79935 0.82089 0.83801
```

Clearly, we can observe that the proportion of variance explained increases very little after the addition of the 6th component. Thus, we will go ahead with 5 components and analyze them one by one.

```
pc_Z = prcomp(Z, rank=5)
summary(pc_Z)

## Importance of first k=5 (out of 32) components:
##          PC1      PC2      PC3      PC4      PC5
## Standard deviation 5.7539 4.4774 4.3170 4.1992 3.62528
## Proportion of Variance 0.2217 0.1343 0.1248 0.1181 0.08802
## Cumulative Proportion 0.2217 0.3560 0.4808 0.5989 0.68688

# Question 2: how are the individual PCs loaded on the original variables?
loadings = pc_Z$rotation
o1 = order(loadings[,1], decreasing=TRUE)

colnames(Z)[head(o1,10)]
```

```
## [1] "health_nutrition" "personal_fitness" "cooking"           "outdoors"
## [5] "photo_sharing"    "fashion"          "food"              "shopping"
## [9] "beauty"           "eco"
```

```
colnames(Z)[tail(o1,10)]
```

```
## [1] "computers"     "religion"       "tv_film"        "automotive"
## [5] "sports_fandom" "news"          "online_gaming"  "travel"
## [9] "college_uni"   "politics"
```

PC1 helps outline health and personal care categories

Let us now analyze PC2

```
# Question 2: how are the individual PCs loaded on the original variables?
loadings = pc_Z$rotation
o1 = order(loadings[,2], decreasing=TRUE)

colnames(Z)[head(o1,10)]

## [1] "health_nutrition" "politics"           "personal_fitness" "news"
## [5] "travel"          "outdoors"          "food"             "computers"
## [9] "automotive"      "sports_fandom"

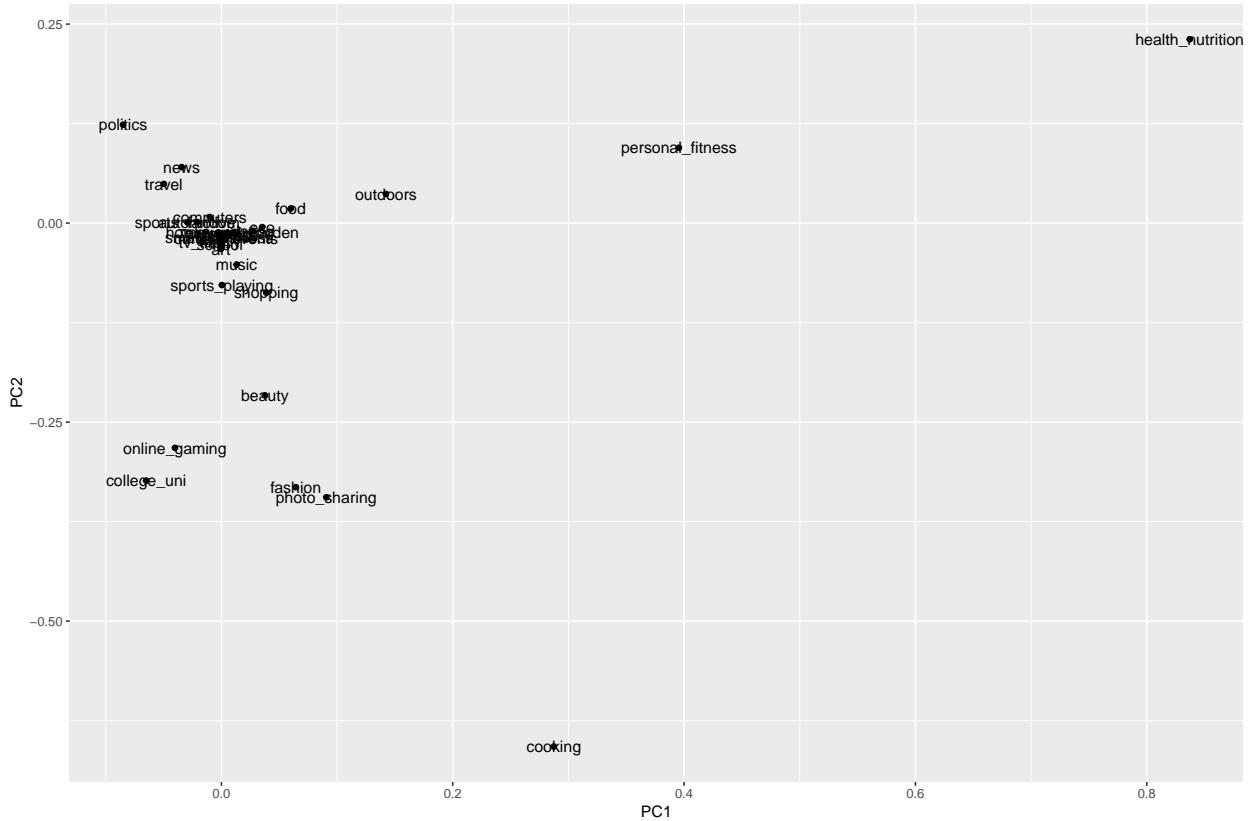
colnames(Z)[tail(o1,10)]

## [1] "art"            "music"           "sports_playing" "shopping"
## [5] "beauty"         "online_gaming"  "college_uni"    "fashion"
## [9] "photo_sharing"  "cooking"
```

PC2 does not seem to be clearly defined

Let us now try to visualize these Principal Components together in a bi-variate plot

```
library(ggplot2)
loadings_df <- as.data.frame(loadings)
ggplot(data = loadings_df , aes(x = PC1, y = PC2))+
  geom_point()+
  geom_text(label = rownames(loadings_df))
```

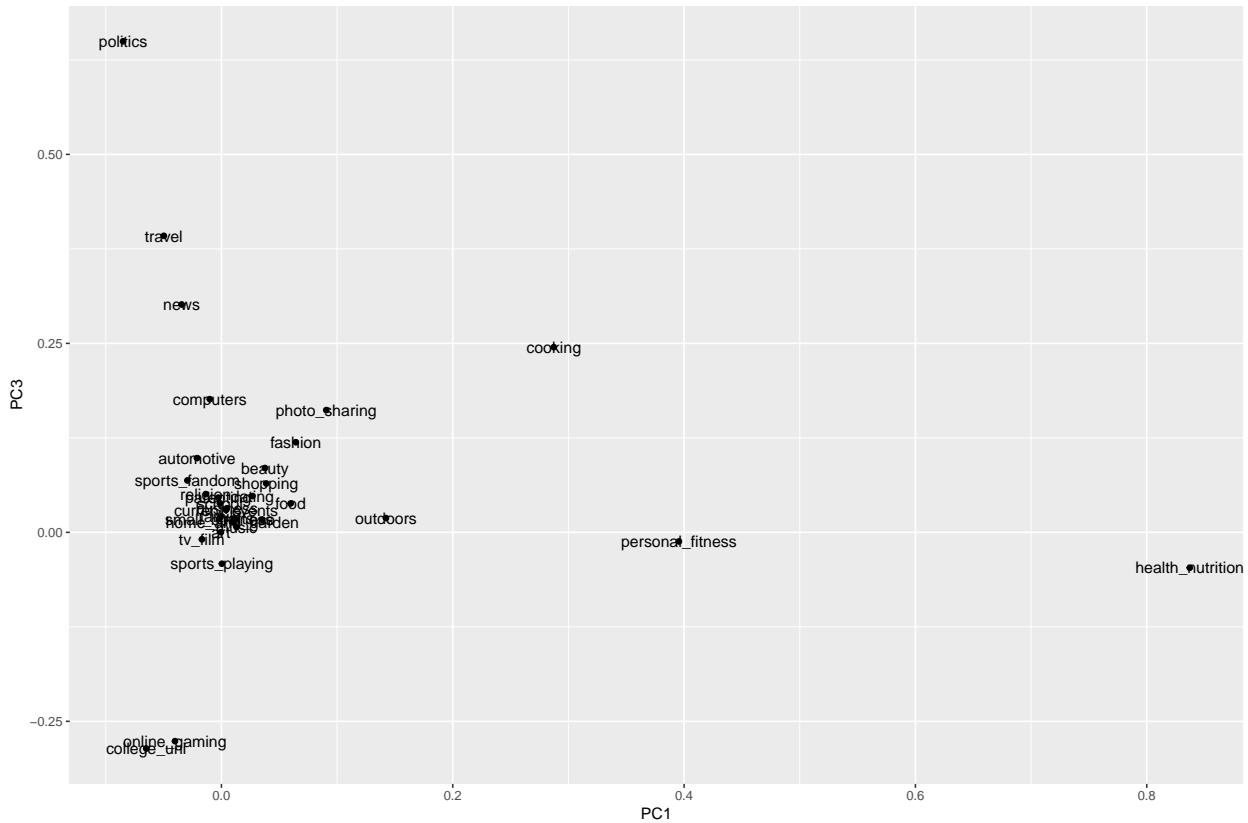


```

loadings_df <- as.data.frame(loadings)
ggplot(data = loadings_df , aes(x = PC1, y = PC3))+  

  geom_point() +  

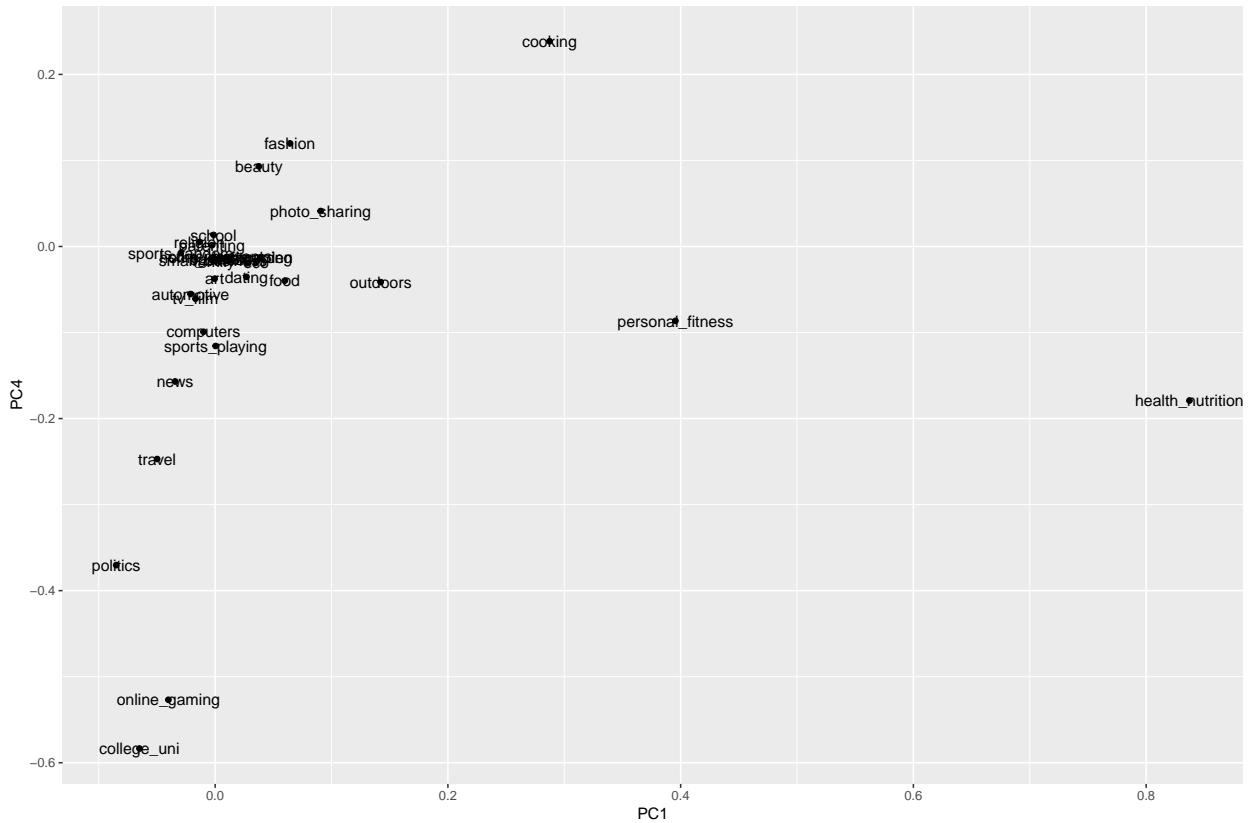
  geom_text(label = rownames(loadings_df))
  
```



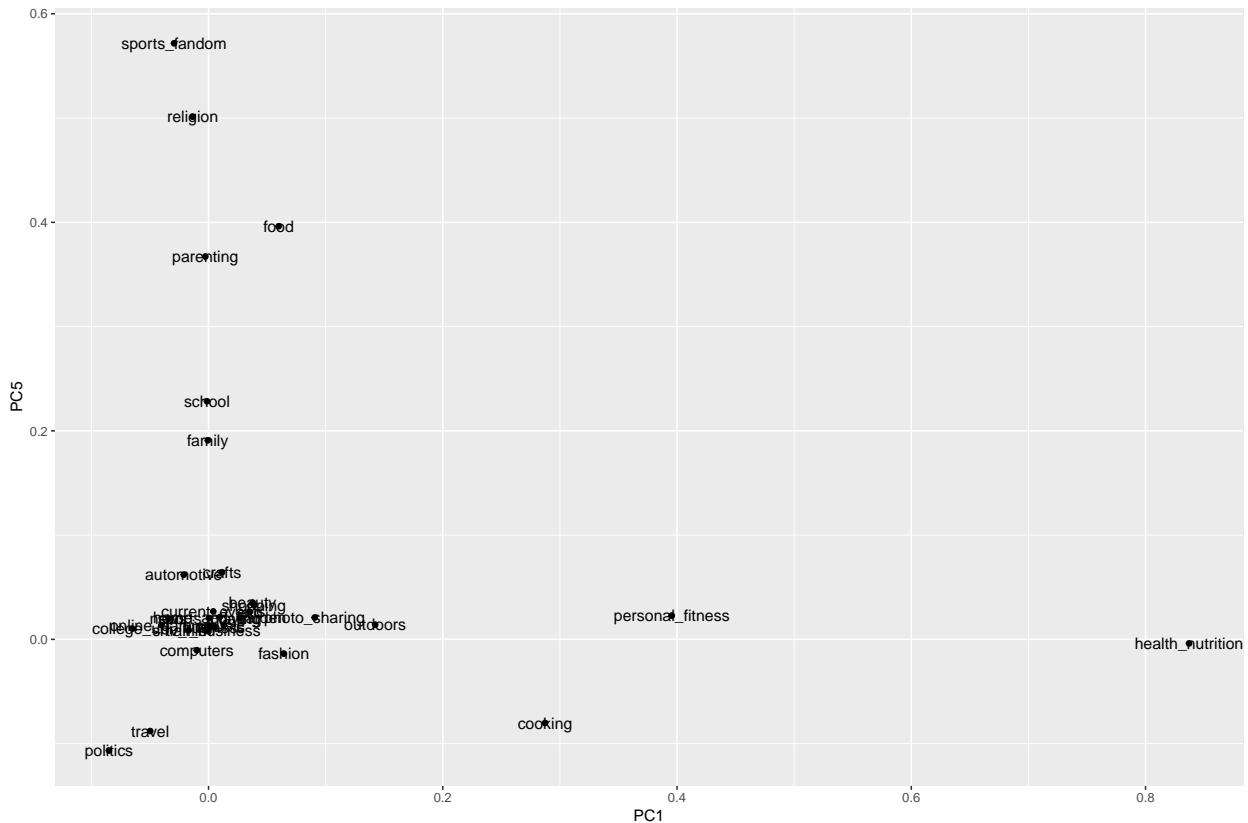
```

loadings_df <- as.data.frame(loadings)
ggplot(data = loadings_df , aes(x = PC1, y = PC4))+
  geom_point()+
  geom_text(label = rownames(loadings_df))

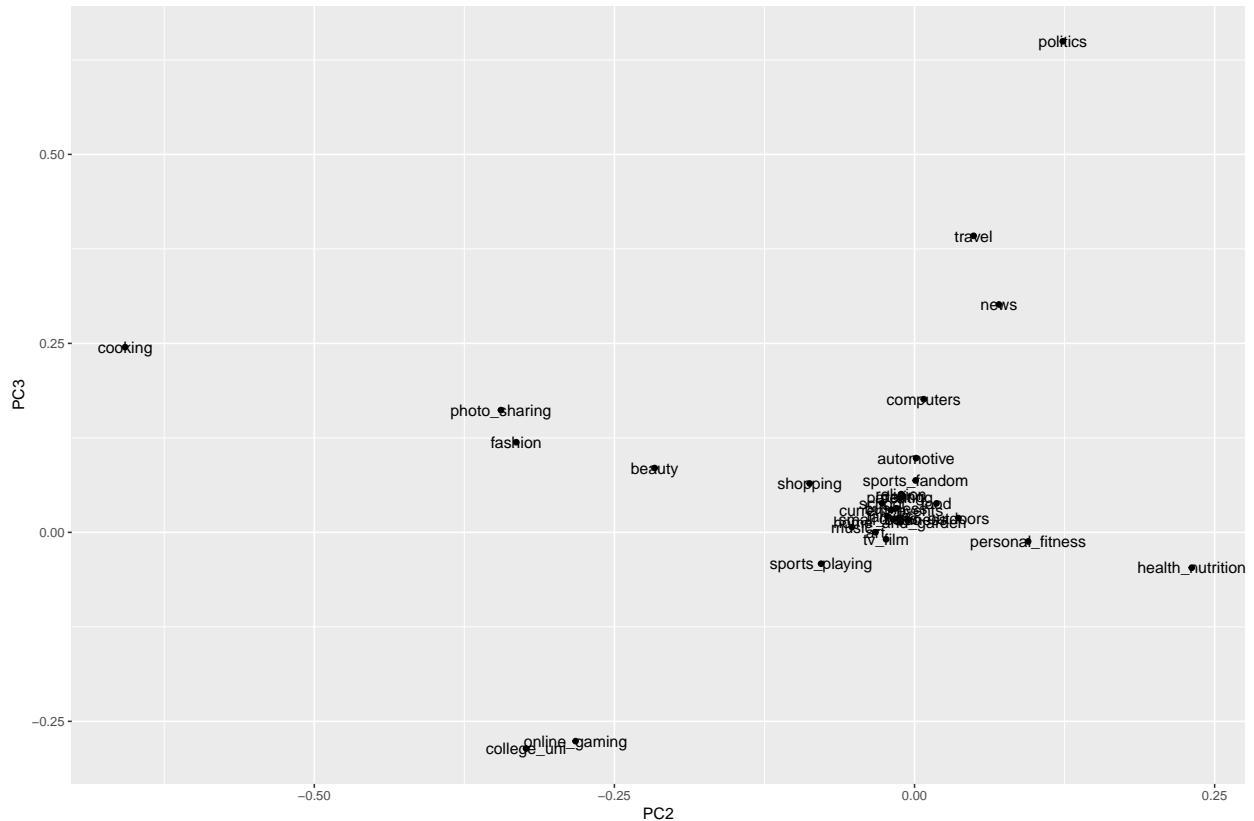
```



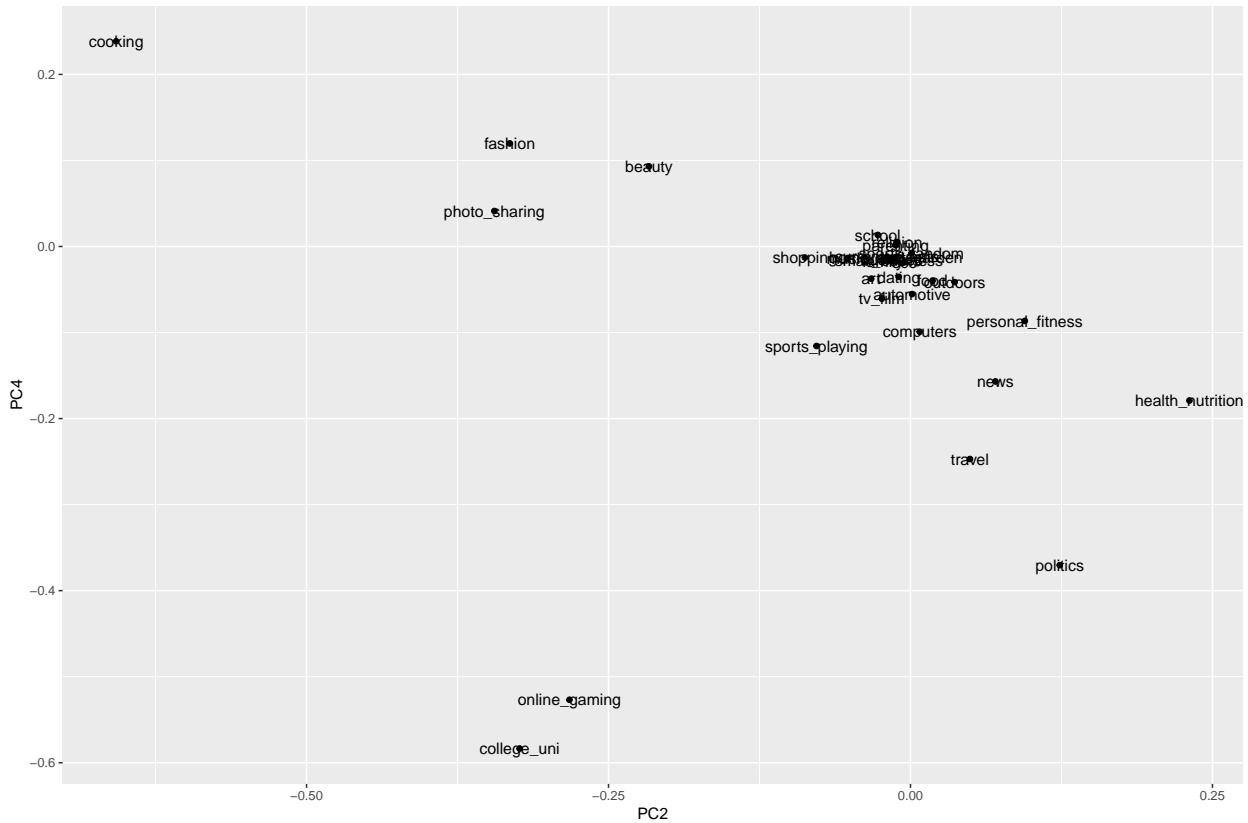
```
loadings_df <- as.data.frame(loadings)
ggplot(data = loadings_df , aes(x = PC1, y = PC5))+  
  geom_point() +  
  geom_text(label = rownames(loadings_df))
```



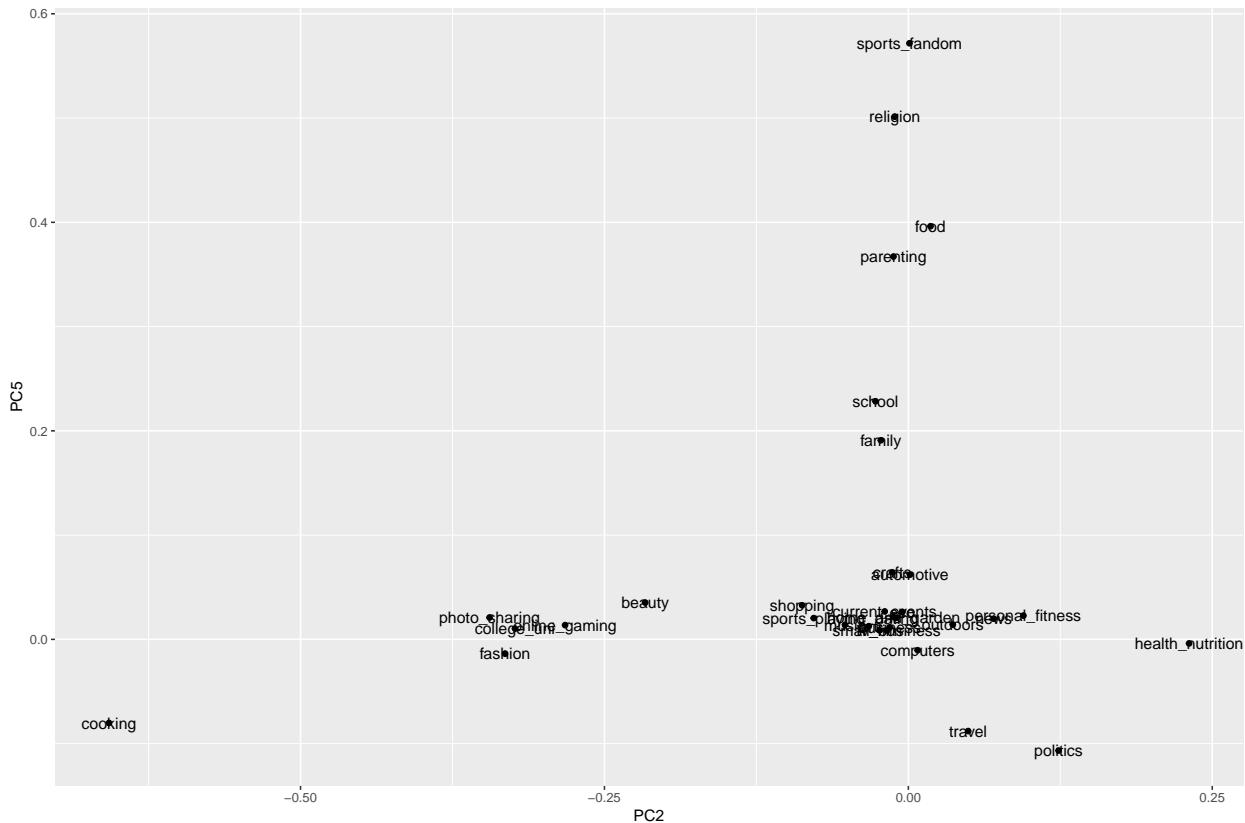
```
loadings_df <- as.data.frame(loadings)
ggplot(data = loadings_df , aes(x = PC2, y = PC3))+
  geom_point()+
  geom_text(label = rownames(loadings_df))
```



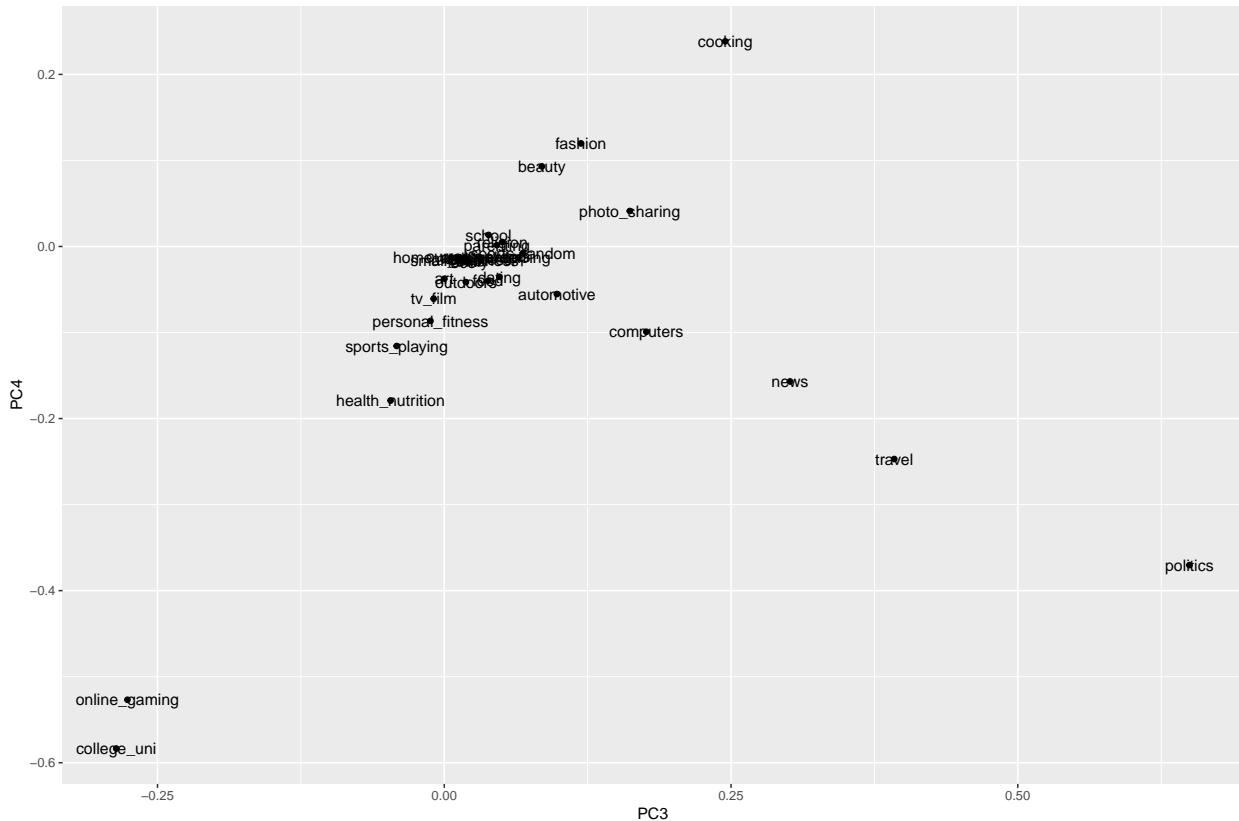
```
loadings_df <- as.data.frame(loadings)
ggplot(data = loadings_df , aes(x = PC2, y = PC4))+
  geom_point()+
  geom_text(label = rownames(loadings_df))
```



```
loadings_df <- as.data.frame(loadings)
ggplot(data = loadings_df , aes(x = PC2, y = PC5))+
  geom_point()+
  geom_text(label = rownames(loadings_df))
```



```
loadings_df <- as.data.frame(loadings)
ggplot(data = loadings_df , aes(x = PC3, y = PC4))+
  geom_point()+
  geom_text(label = rownames(loadings_df))
```

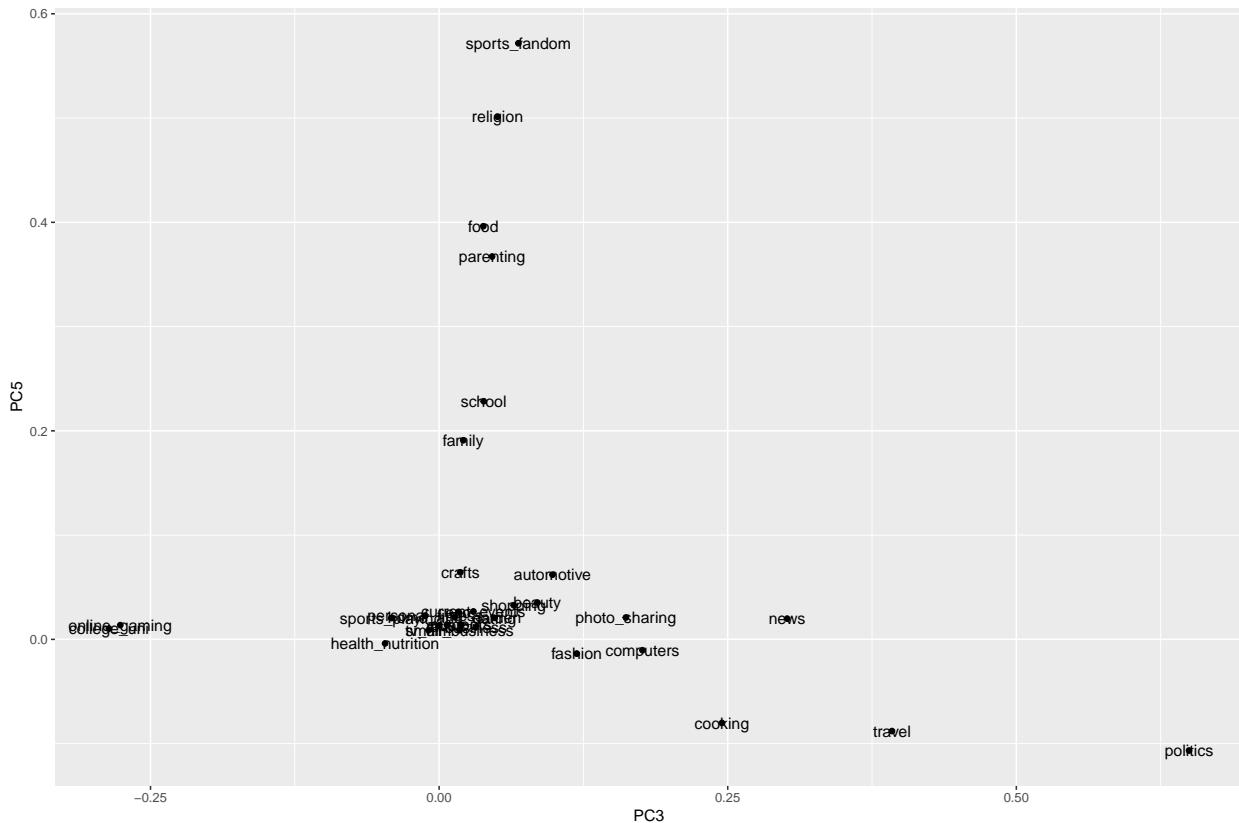


```

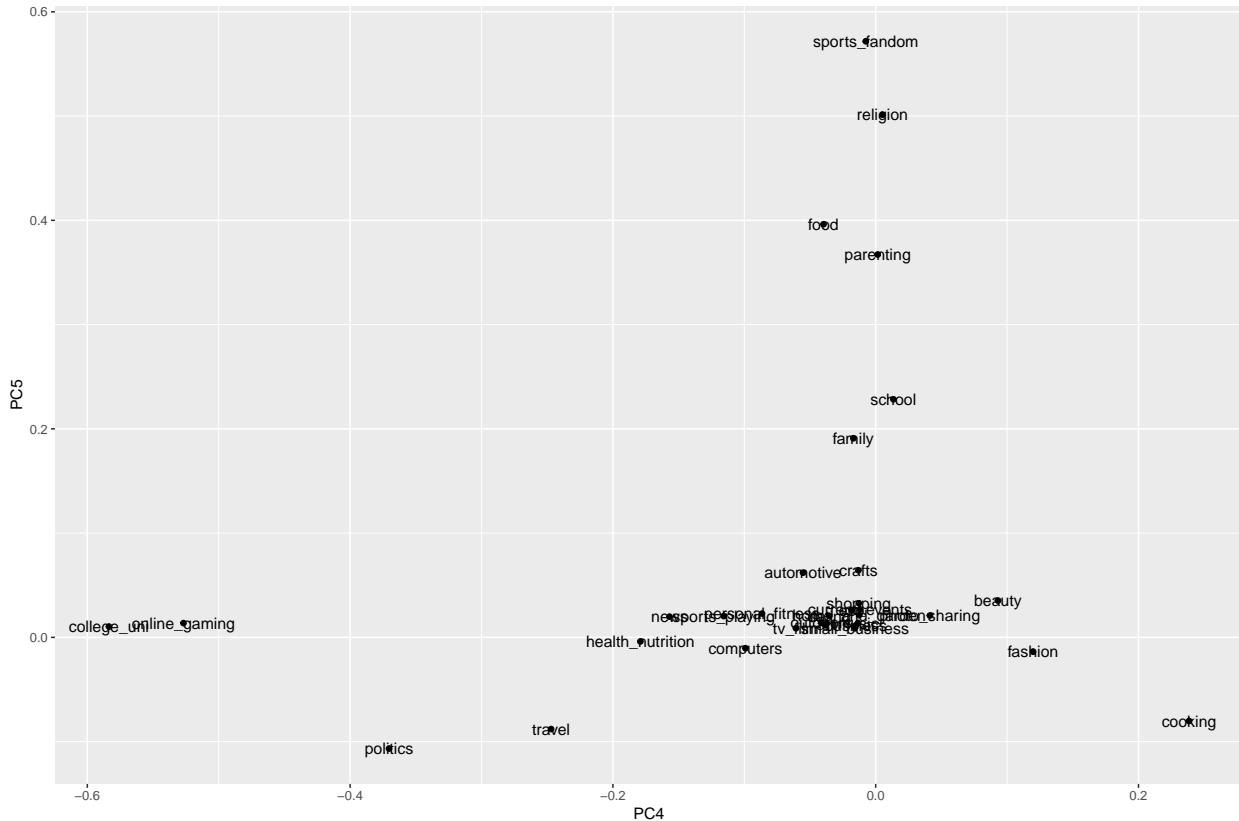
loadings_df <- as.data.frame(loadings)
ggplot(data = loadings_df , aes(x = PC3, y = PC5))+  

  geom_point() +  

  geom_text(label = rownames(loadings_df))
  
```



```
loadings_df <- as.data.frame(loadings)
ggplot(data = loadings_df , aes(x = PC4, y = PC5))+  
  geom_point() +  
  geom_text(label = rownames(loadings_df))
```



Observations: - PC4 vs PC5 : Separates the family cluster from other clusters; also, to the left is the student cluster

PC4 vs PC2 : Separates the social media influences cluster

A more efficient approach maybe is to cluster these PCs and see which of them are similar

```
library(LICORS)
clust = kmeanspp(loadings_df, k=6) # Add min thresholds
clust$size

## [1] 18 3 4 3 2 2

z = order(clust$cluster)
clust$cluster[z]

##   current_events      tv_film       family home_and_garden
##                 1             1             1                  1
##       music      shopping sports_playing           eco
##                 1             1             1                  1
##       computers    business     outdoors      crafts
##                 1             1             1                  1
##   automotive          art        beauty      dating
##                 1             1             1                  1
##       school small_business photo_sharing      cooking
```

```

##          1          1          2          2
## fashion sports_fandom food religion
##          2          3          3          3
## parenting travel politics news
##          3          4          4          4
## health_nutrition personal_fitness online_gaming college_uni
##          5          5          6          6

## No matter how many clusters we try to create, the clusters are very unbalanced indicating that PC are

# The clusters with less size are clearly distinguishable

```

We try to look at all of these PCs to get a concrete sense of what ‘trait’ do they distinguish but are unable to decipher clearly.

Since this is a marketing segmentation problem, we will require very good understanding on what the individual PCs are trying to distinguish.

As this clarity is not present, we abandon the idea of PCs completely and try Kmeans++ on the entire data.

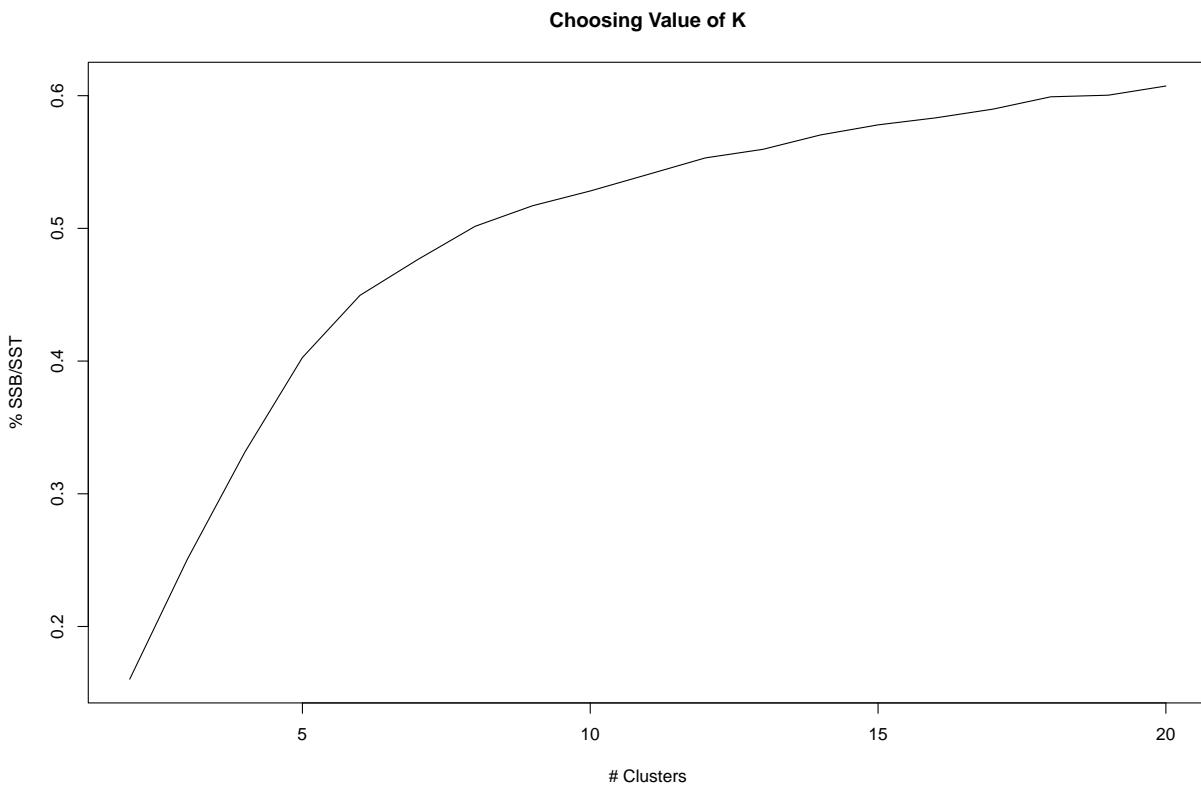
K Means for Entire data

```

ratio = c()
clust_counter = c(2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20)
for ( clust_count in clust_counter)
{
  clust_total = kmeanspp(Z, k=clust_count)
  #clust_df = cbind(Z,clust_total$cluster)
  a = clust_total$betweenss/ clust_total$totss
  ratio = append(ratio,a)
}

plot(x = clust_counter,
      y = ratio,
      type = 'l',
      xlab = '# Clusters',
      ylab = '% SSB/SST',
      main = 'Choosing Value of K'
)

```



Along expected lines, optimal number of clusters = 6

Let us fit the final K means model to cluster it

```
clust = kmeanspp(Z, k=6)
### Top category
a = colnames(analysis_df)[max.col(analysis_df, ties.method = "first")]
clust$size

## [1] 494 618 2675 545 881 379

### final df post clustering
clustering_df = cbind(analysis_df, clust$cluster, a)

df_1 = clustering_df %>% group_by(`clust$cluster`, a) %>% dplyr::summarise(rcds = n())

## `summarise()` has grouped output by 'clust$cluster'. You can override using the
## '.groups' argument.
```

Now let us try to look at the Top 5 most common Tags in Each Cluster

```
top_cat <- df_1 %>%
  dplyr::group_by(`clust$cluster`) %>%
  dplyr::mutate(my_ranks = order(order(rcds, `clust$cluster`))) %>%
  dplyr::filter(my_ranks<=5)
```

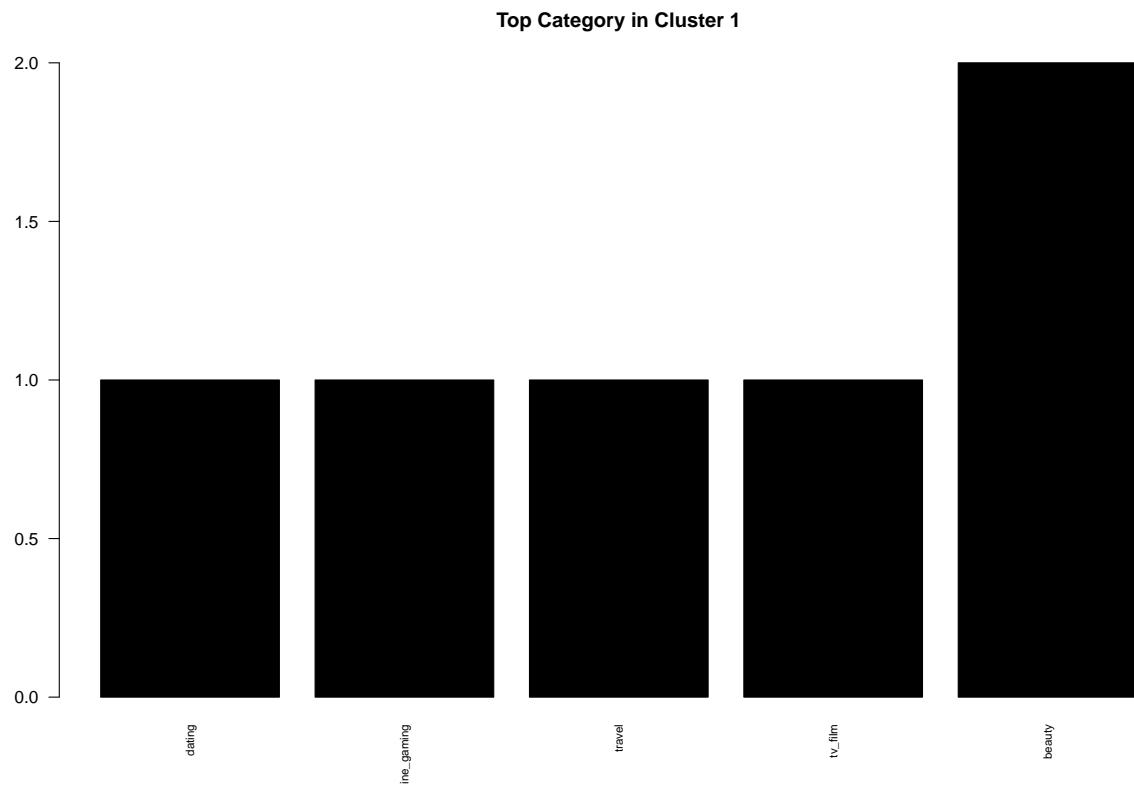
```

for (x in c(1,2,3,4,5,6))
{
  bar_df = top_cat %>% filter(`clust$cluster` == x) %>% dplyr::arrange(rcds)

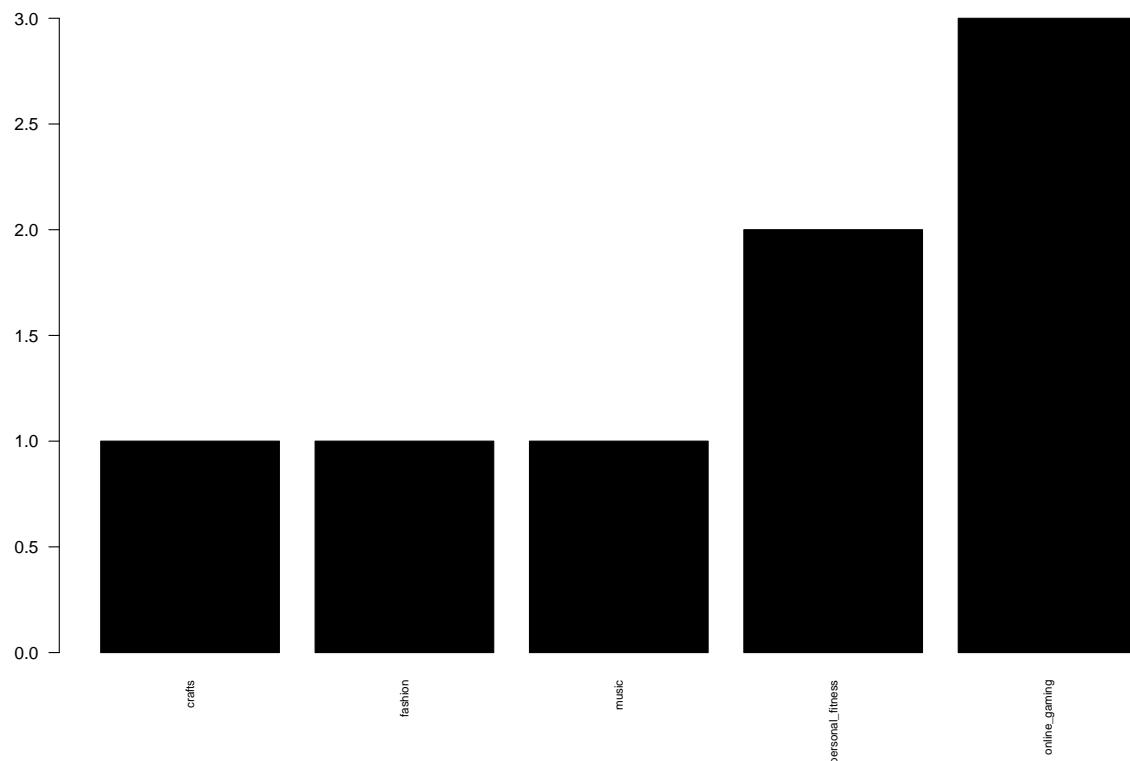
  bp = barplot( bar_df$rcds, names.arg = bar_df$a ,
    main = paste0("Top Category in Cluster ", x),
    col= 'Black', las=2,cex.names=.7)

}

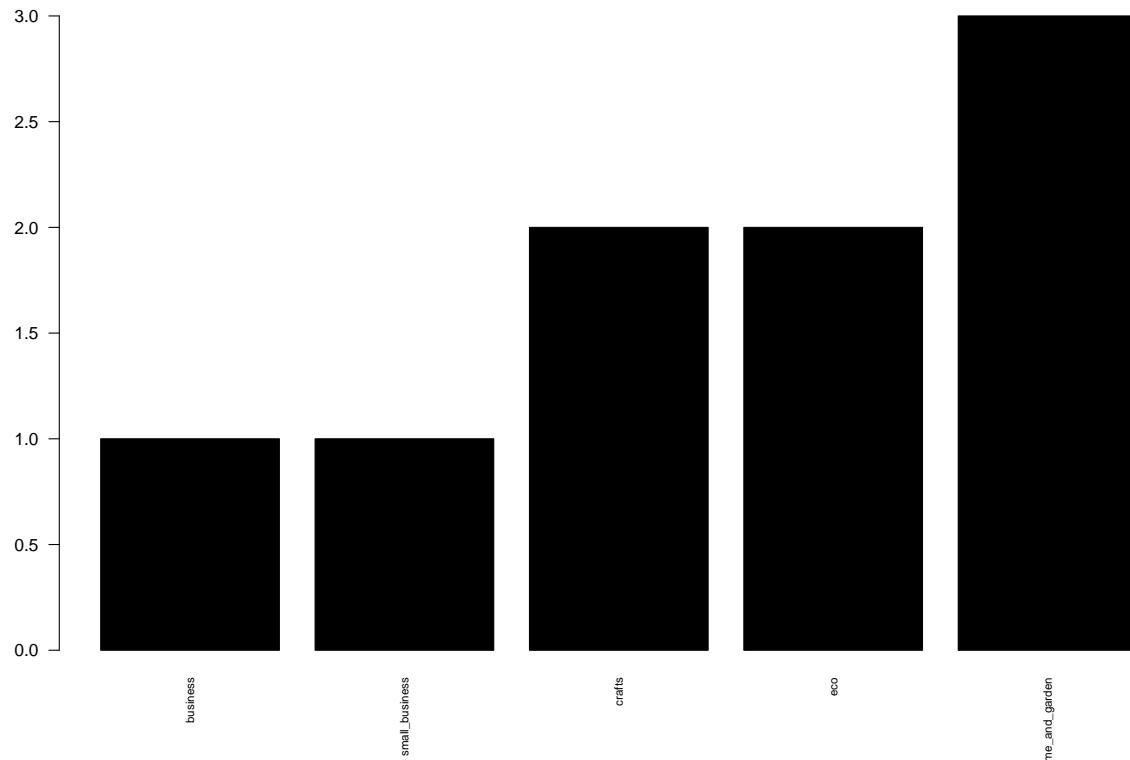
```



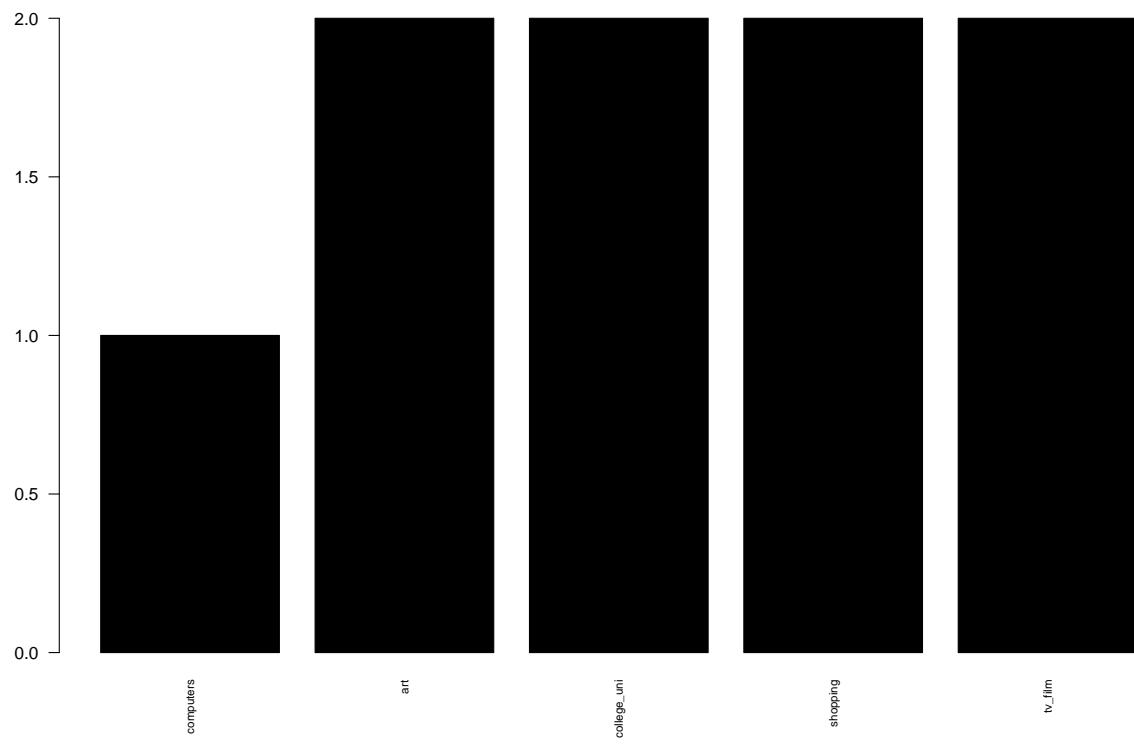
Top Category in Cluster 2



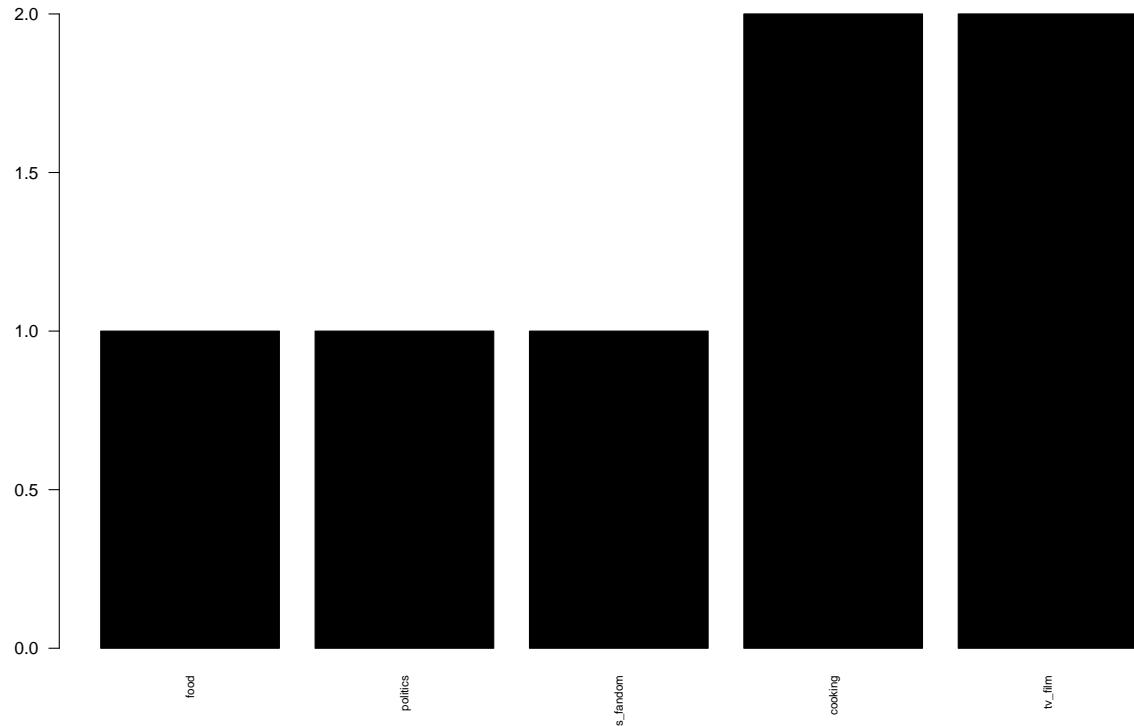
Top Category in Cluster 3

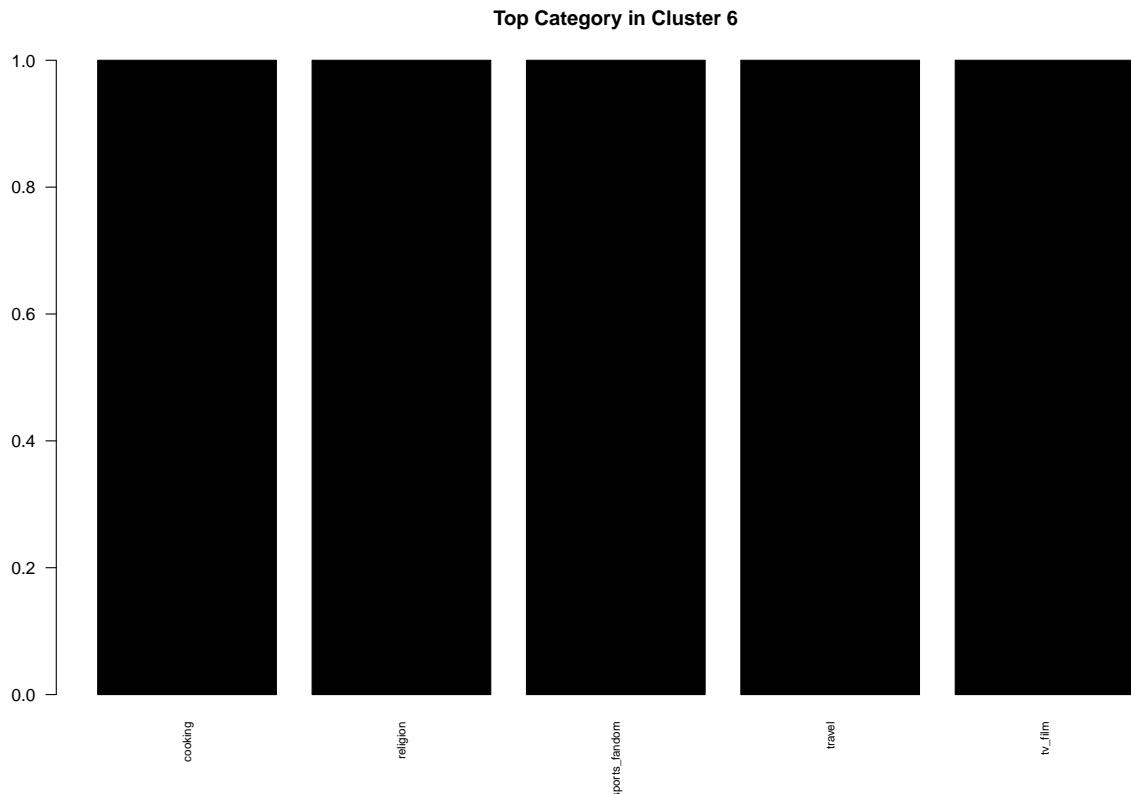


Top Category in Cluster 4



Top Category in Cluster 5





So, clearly these **clusters** are consistent with what we observed in the correlation plot during our **EDA**.

The Reuters Corpus

Step 1. Load files and perform pre-processing steps (for training data)

Step 2. Load files and perform pre-processing steps (for testing data)

Step 3. Peform PCA for Dimensionality reduction ** We will plot a variance plot::

We can select till PC800 as almost 80% of variance explained by them

Step 4. Build Random Forest models to identify authors The model built using Random forest predicted 1844 documents correctly with its respective authors and gave us accuracy of ~74%.

Association rule mining

Aim

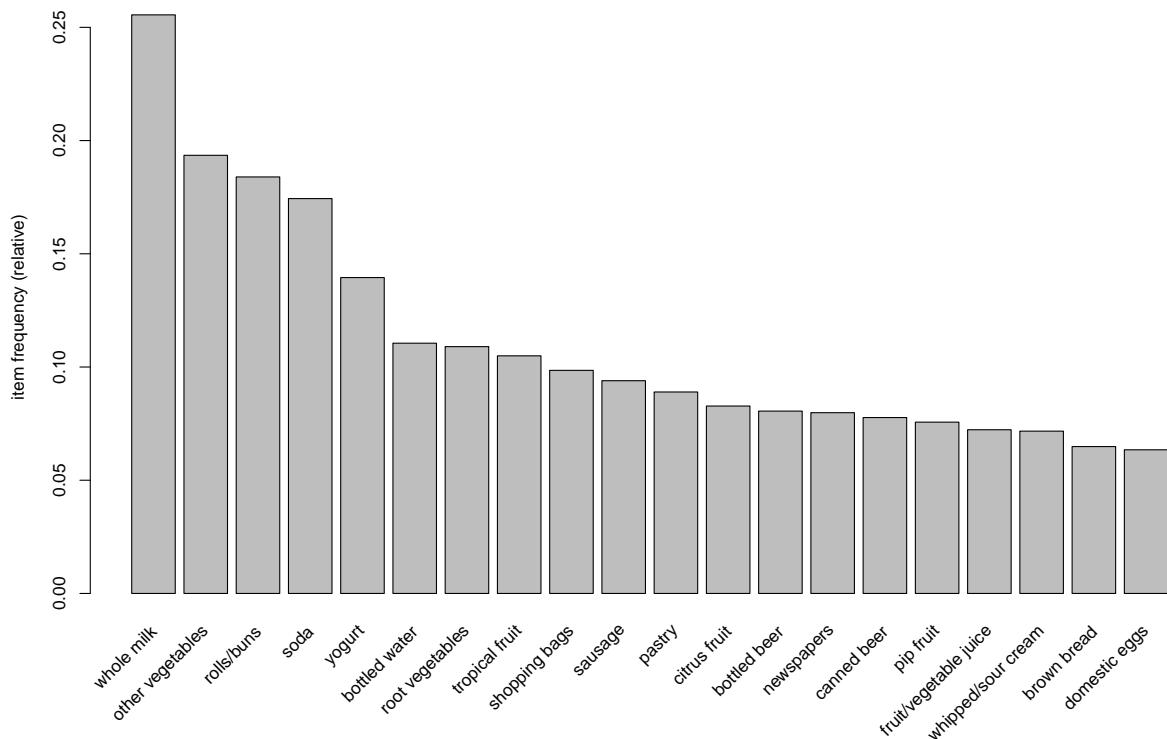
We have multiple shopping baskets of grocery purchases and we will find some interesting association rules between them using rule mining.

Let's see the raw dataset:

```
## [1] "citrus fruit,semi-finished bread,margarine,ready soups"
## [2] "tropical fruit,yogurt,coffee"
## [3] "whole milk"
## [4] "pip fruit,yogurt,cream cheese ,meat spreads"
## [5] "other vegetables,whole milk,condensed milk,long life bakery product"
## [6] "whole milk,butter,yogurt,rice,abrasive cleaner"
```

The apriori algorithm needs the data to be transformed into “transactions” class before we can use the raw data:

Let's plot the transformed dataset to look at the frequency



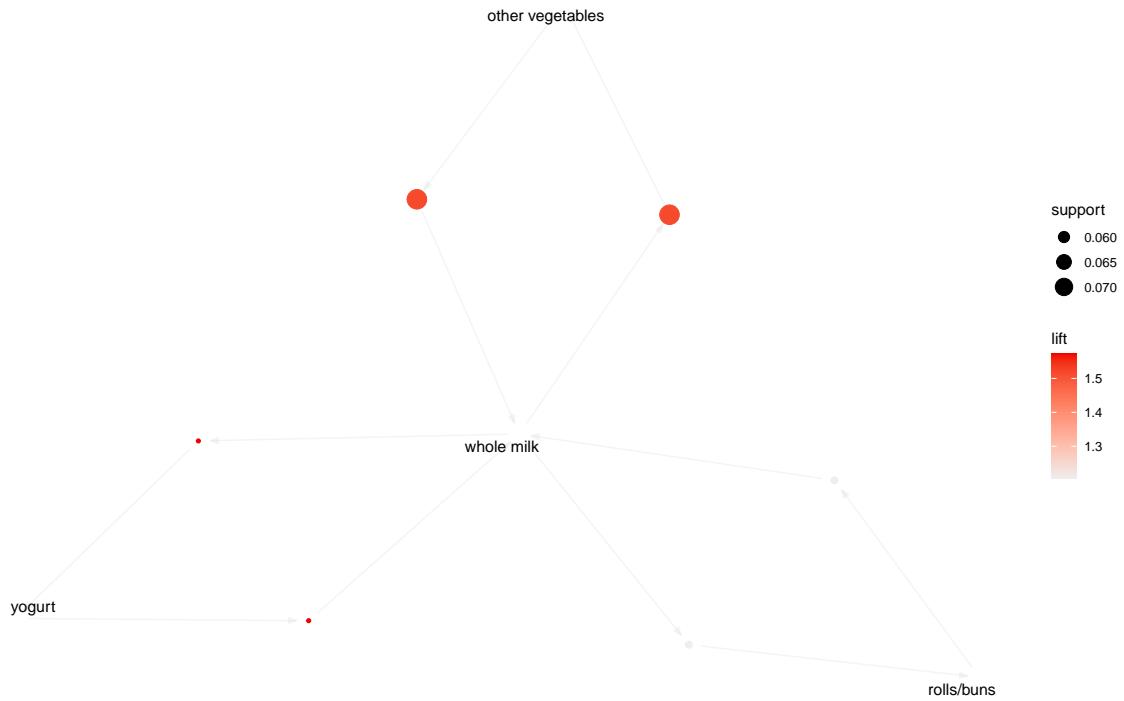
Here is a summary of the dataset:

1. Total of 9835 transactions in the dataset
2. Most of the transactions have 4 or lesser amount of items in the basket
3. The most frequently bought item is whole milk which was present in 2513 transactions

Let's try out the 'apriori' algo with support > 0.05, confidence > 0.1 and length <= 2

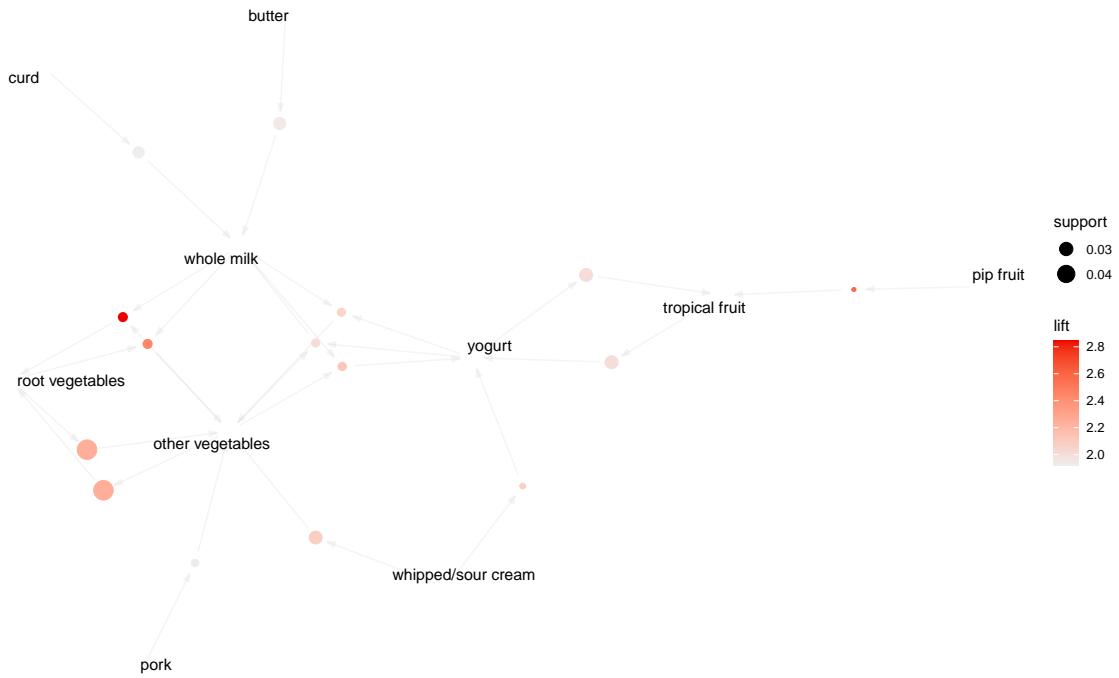
There are only 6 rules generated because of the high support and low confidence level. We also notice that most relationships in this item set include whole milk, yogurt and rolls/buns which is in accordance with the transaction frequency plot we saw earlier. These are some of the most frequently bought items.

```
##      lhs                  rhs          support    confidence coverage
## [1] {yogurt}            => {whole milk} 0.05602440 0.4016035 0.1395018
## [2] {whole milk}        => {yogurt}    0.05602440 0.2192598 0.2555160
## [3] {rolls/buns}        => {whole milk} 0.05663447 0.3079049 0.1839349
## [4] {whole milk}        => {rolls/buns} 0.05663447 0.2216474 0.2555160
## [5] {other vegetables} => {whole milk}  0.07483477 0.3867578 0.1934926
## [6] {whole milk}        => {other vegetables} 0.07483477 0.2928770 0.2555160
##      lift      count
## [1] 1.571735 551
## [2] 1.571735 551
## [3] 1.205032 557
## [4] 1.205032 557
## [5] 1.513634 736
## [6] 1.513634 736
```



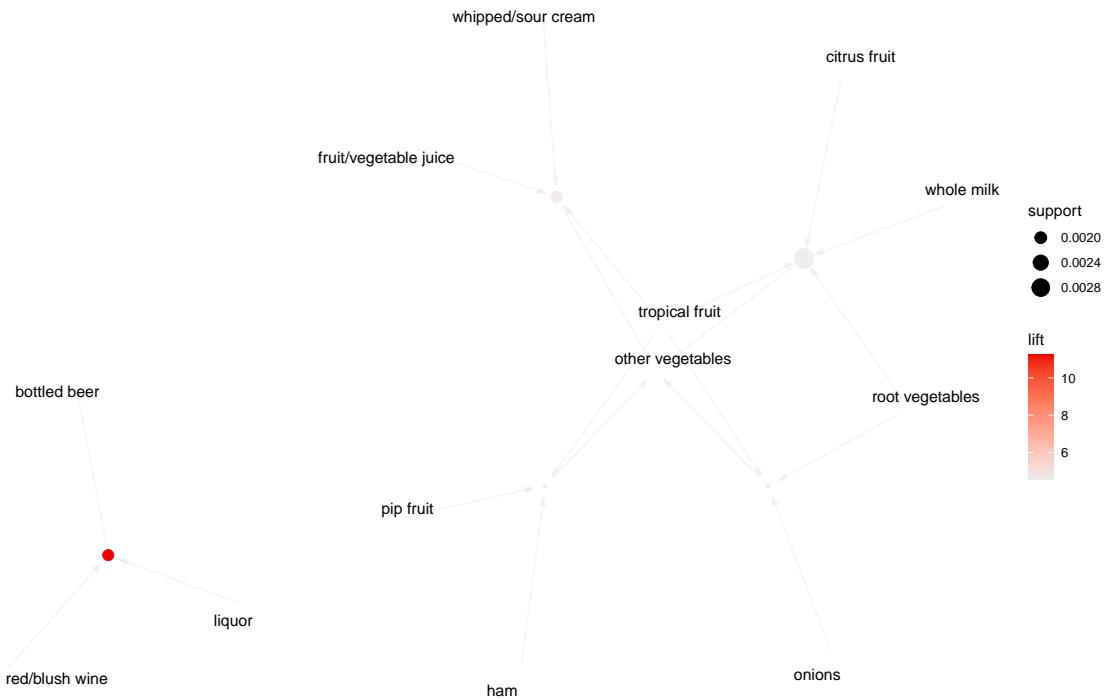
Let's try decreasing the support and increase confidence slightly with support > 0.02, confidence > 0.2 and length <= 2

Now we get 72 rules with a lot more items. Again we can notice that whole milk is a common occurrence.



** Let's try increasing the confidence level and decrease the support further. Let's explore rules with support > 0.0015, confidence > 0.8 and length <=2

This time we get 60 rules and again can notice that whole milk is the most frequent item as expected



Summary

From the association rules, some of the conclusions that can be drawn are:

1. People are more likely to buy bottled beer if they purchased red wine or liquor
2. People are more likely to buy vegetables when they buy vegetable/fruit juice
3. Whole milk is the most common item purchased by customers