Visual story telling part 1: green buildings

An Austin real-estate developer is interested in the possible economic impact of "going green" in her latest project: a new 15-story mixed-use building on East Cesar Chavez, just across I-35 from downtown. Will investing in a green building be worth it, from an economic perspective? The baseline construction costs are $100 million, with a 5% expected premium for green certification.

Extract the buildings with green ratings

Let's also look at the distribution of rents for green buildings: not normally distributed. The developer's staff is correct on the median rent price for green buildings.

```
print('Mean rent per sq ft for Green Buildings is')
```

```
## [1] "Mean rent per sq ft for Green Buildings is"
```

```
mean(green_only$Rent)
```

```
## [1] 30.01603
```

```
print('Median rent per sq ft for Green Buildings is')
```

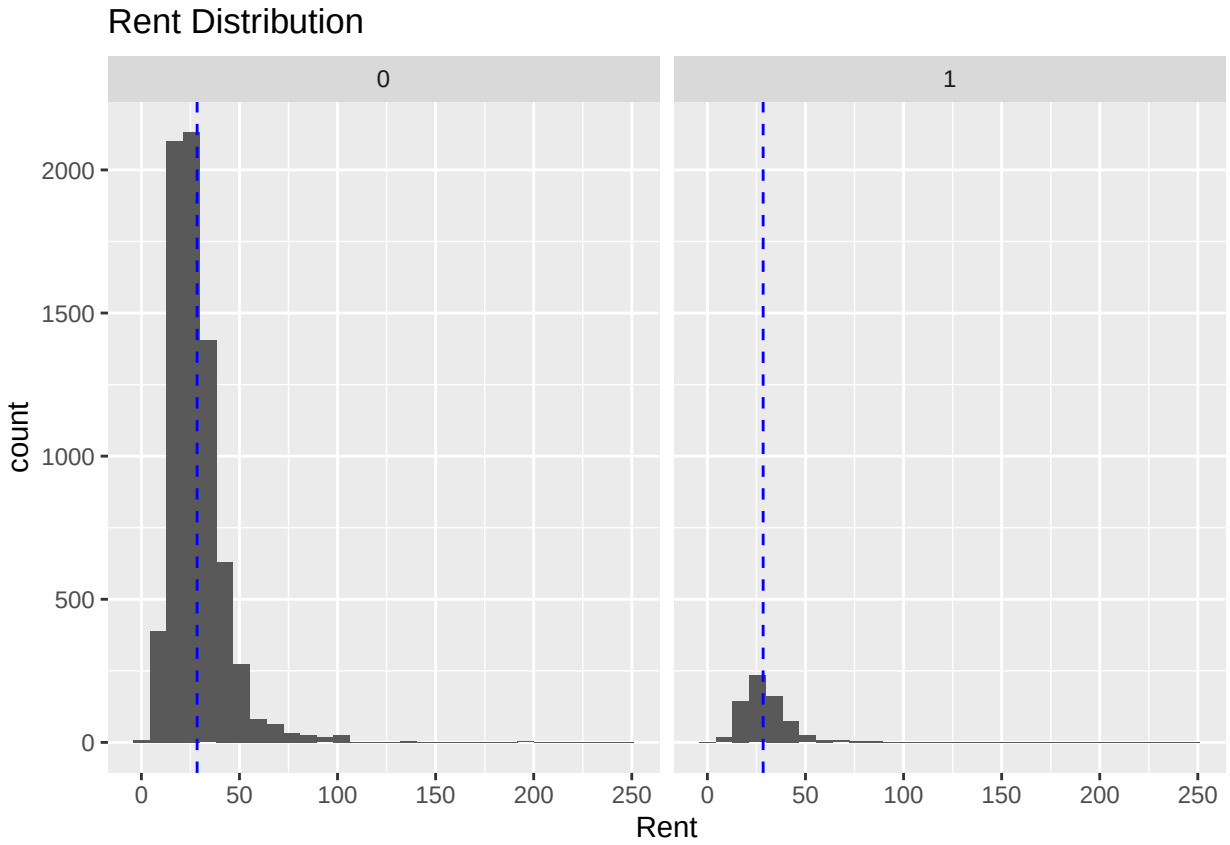
```
## [1] "Median rent per sq ft for Green Buildings is"
```

```
median(green_only$Rent)
```

```
## [1] 27.6
```

```
ggplot(green, aes(x=Rent)) +
  geom_histogram(position="dodge") +
  facet_grid(~green_rating) +
  geom_vline(aes(xintercept=mean(Rent)), color="blue", linetype="dashed") +
  labs(
    title = "Rent Distribution"
    )
```

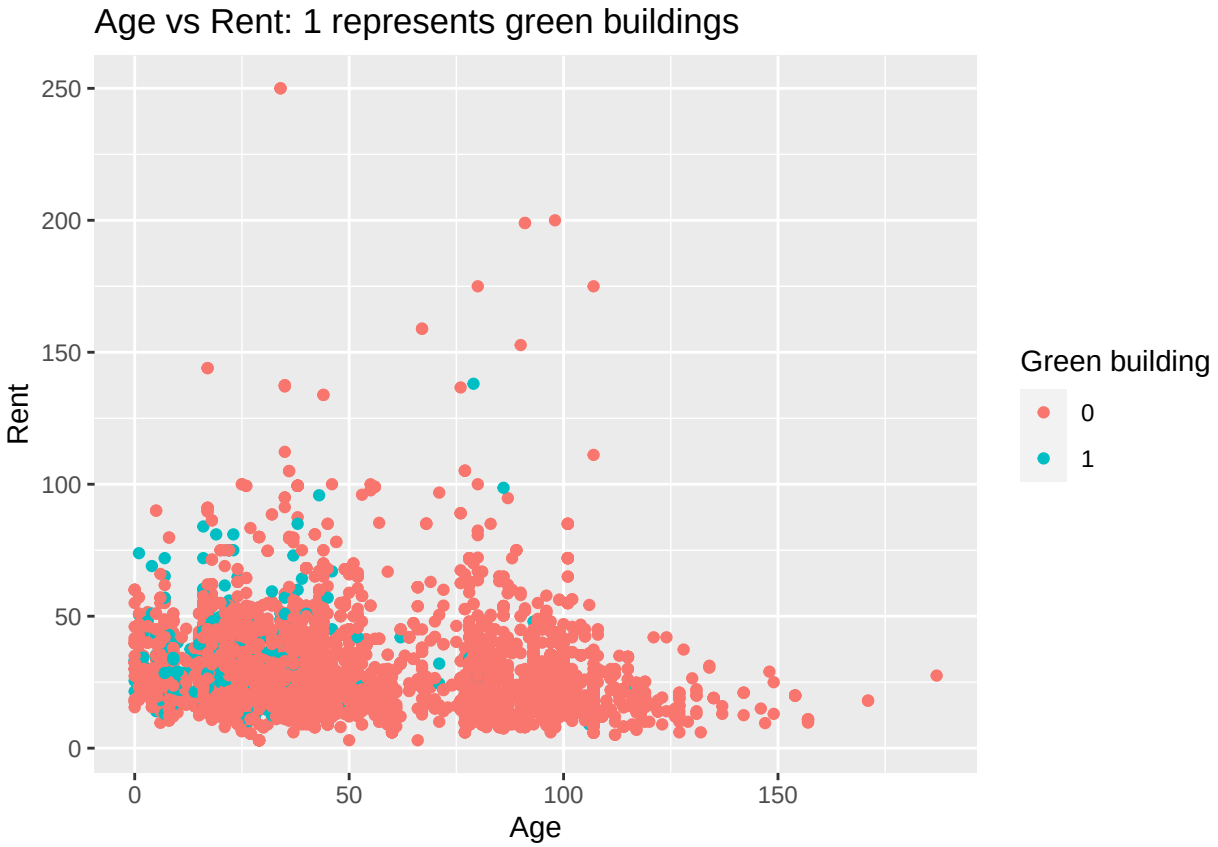
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Let's look at relationship of age versus rent!

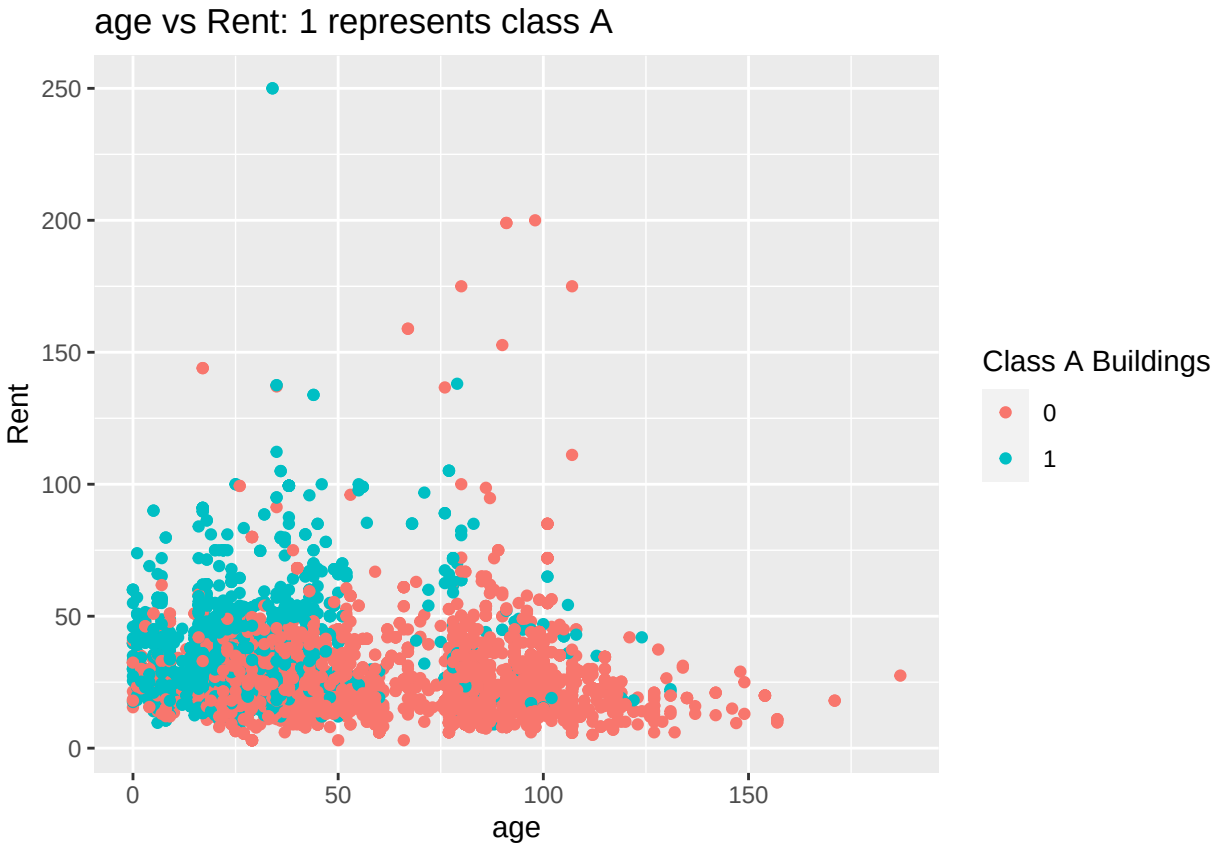
Green buildings tend to be newer. The higher rent price for green buildings could potentially be caused by the age of the buildings.

```
green$green_rating = as.factor(green$green_rating)
ggplot(data = green) +
  geom_point(mapping=aes(x = age, y = Rent, colour = green_rating))+
  labs(
    x = "Age",
    y = 'Rent',
    title = 'Age vs Rent: 1 represents green buildings',
    color = 'Green building')
```



Also, we noticed that Class A buildings tend to be newer and have higher rent, which makes sense because landlords tend to charge higher rent for higher quality properties.

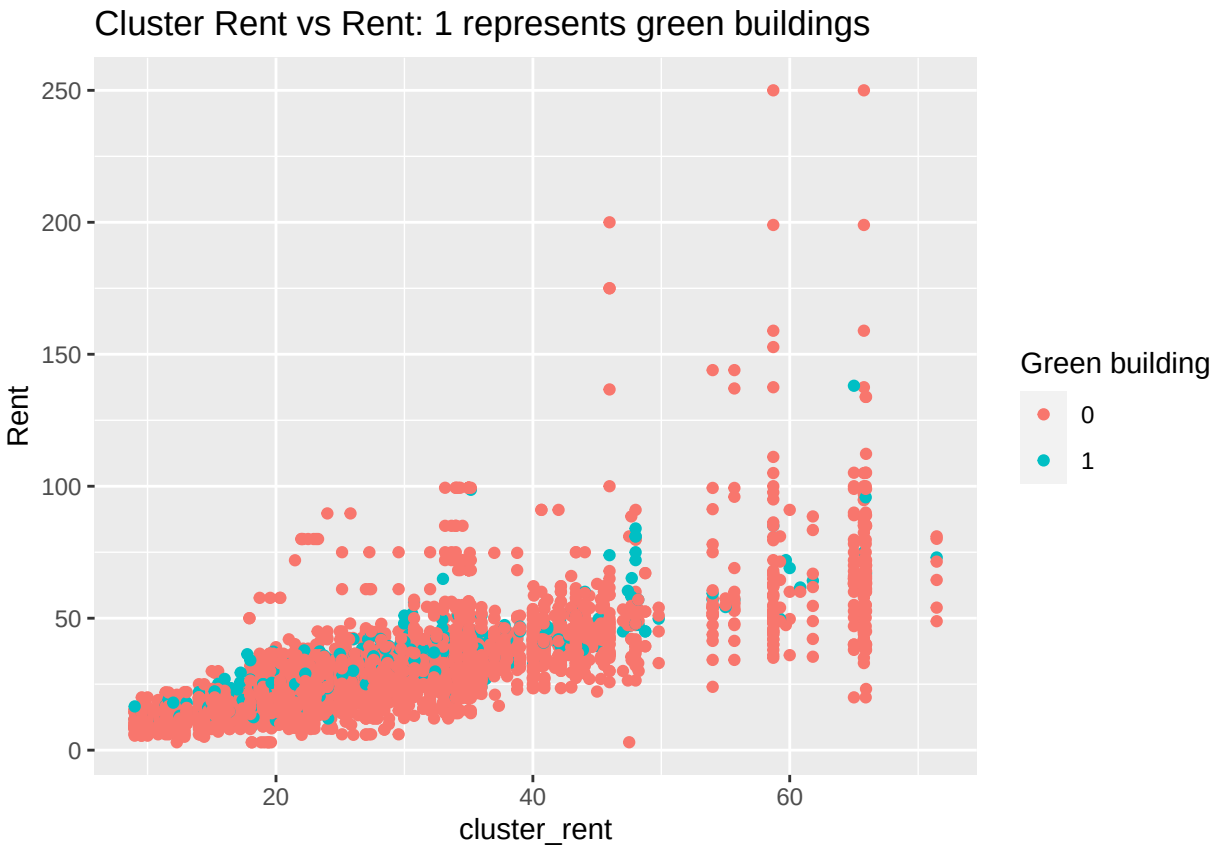
```
green$class_a = as.factor(green$class_a)
ggplot(data = green) +
  geom_point(mapping=aes(x = age, y = Rent, colour = class_a))+
  labs(
    x = "age",
    y = 'Rent',
    title = 'age vs Rent: 1 represents class A',
    color = 'Class A Buildings')
```



Let's examine the relationship cluster rent versus rent

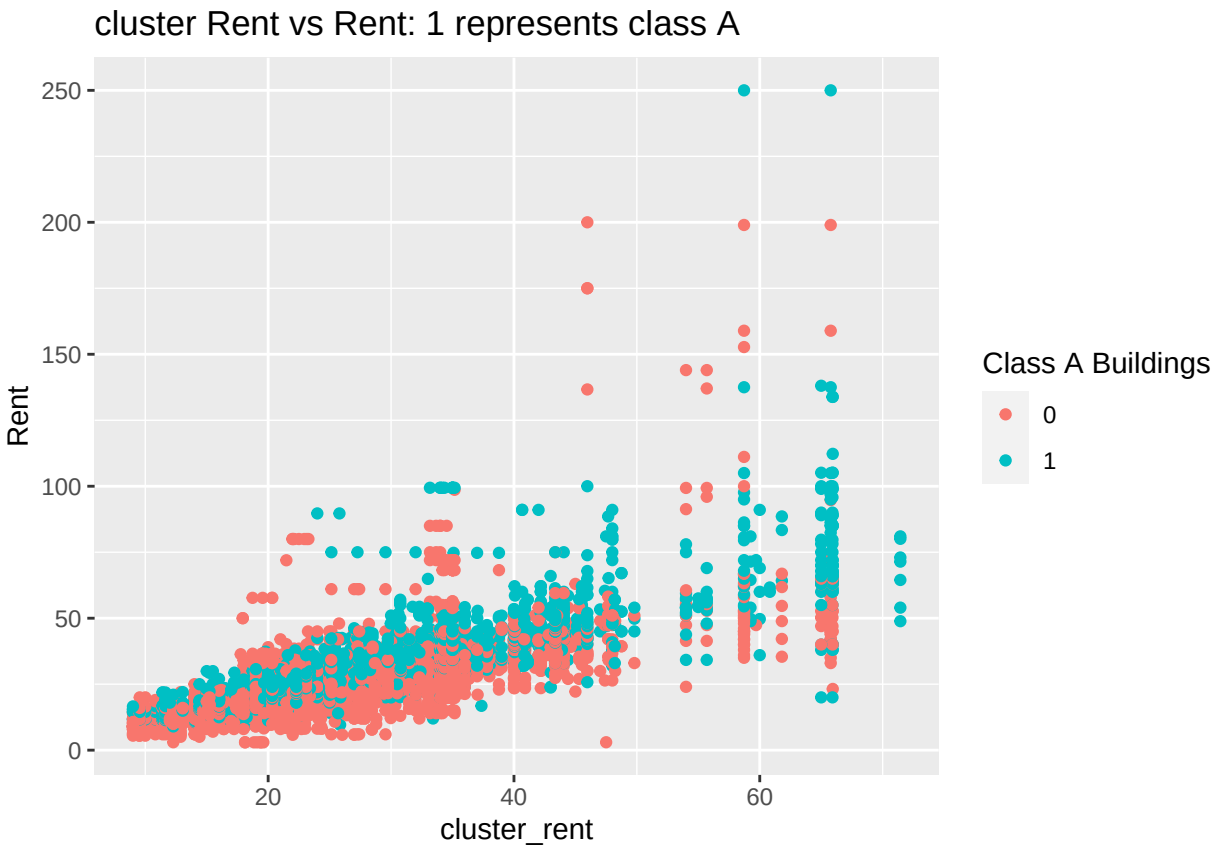
There are a lot less green buildings but the rent seems to be in the similar range compared to non-green buildings as well.

```
green$green_rating = as.factor(green$green_rating)
ggplot(data = green) +
  geom_point(mapping=aes(x = cluster_rent, y = Rent, colour = green_rating))+
  labs(
    x = "cluster_rent",
    y = 'Rent',
    title = 'Cluster Rent vs Rent: 1 represents green buildings',
    color = 'Green building')
```



Unlike the distribution of green buildings vs non-green buildings, the distribution of Class A buildings vs non-Class A buildings is more evenly spread out; cluster_rent and rent are highly correlated.

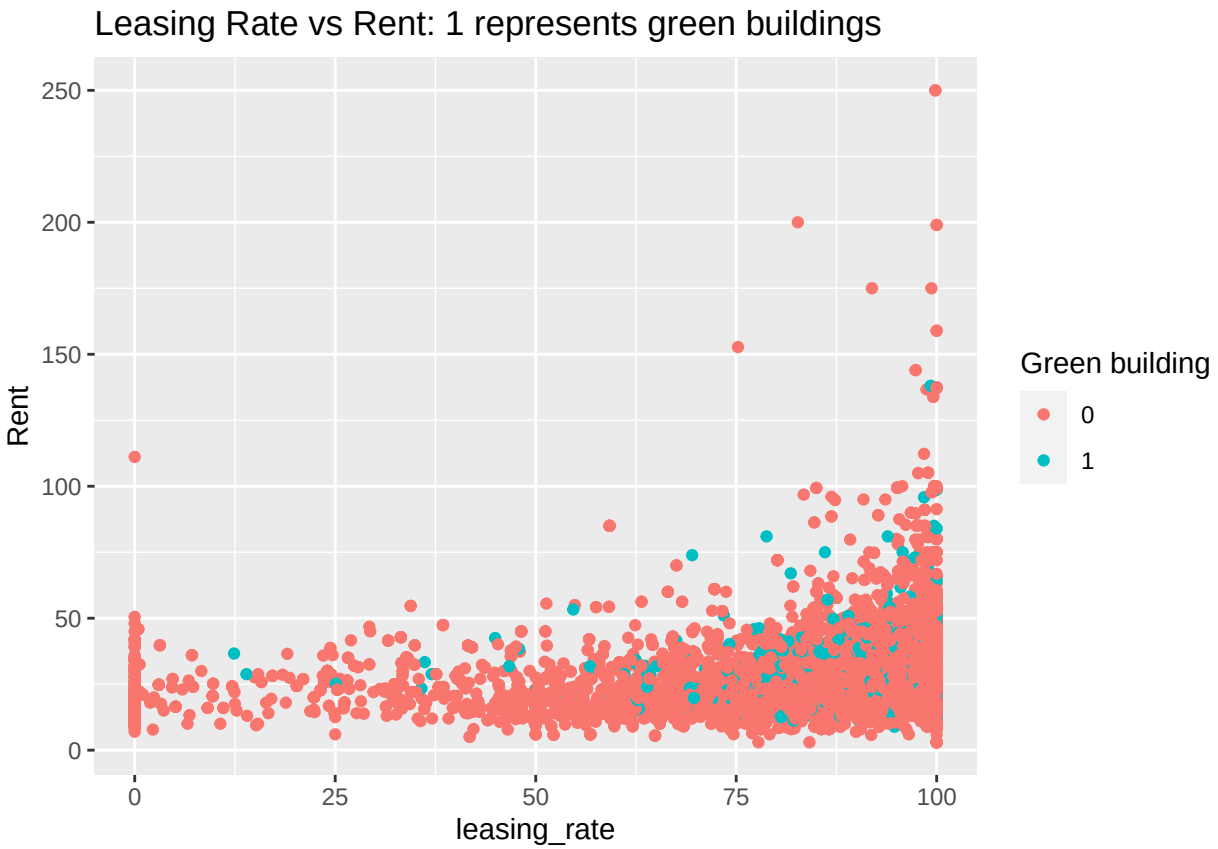
```
green$class_a = as.factor(green$class_a)
ggplot(data = green) +
  geom_point(mapping=aes(x = cluster_rent, y = Rent, colour = class_a))+
  labs(
    x = "cluster_rent",
    y = 'Rent',
    title = 'cluster Rent vs Rent: 1 represents class A',
    color = 'Class A Buildings')
```



Let's examine the relationship leasing rate versus rent

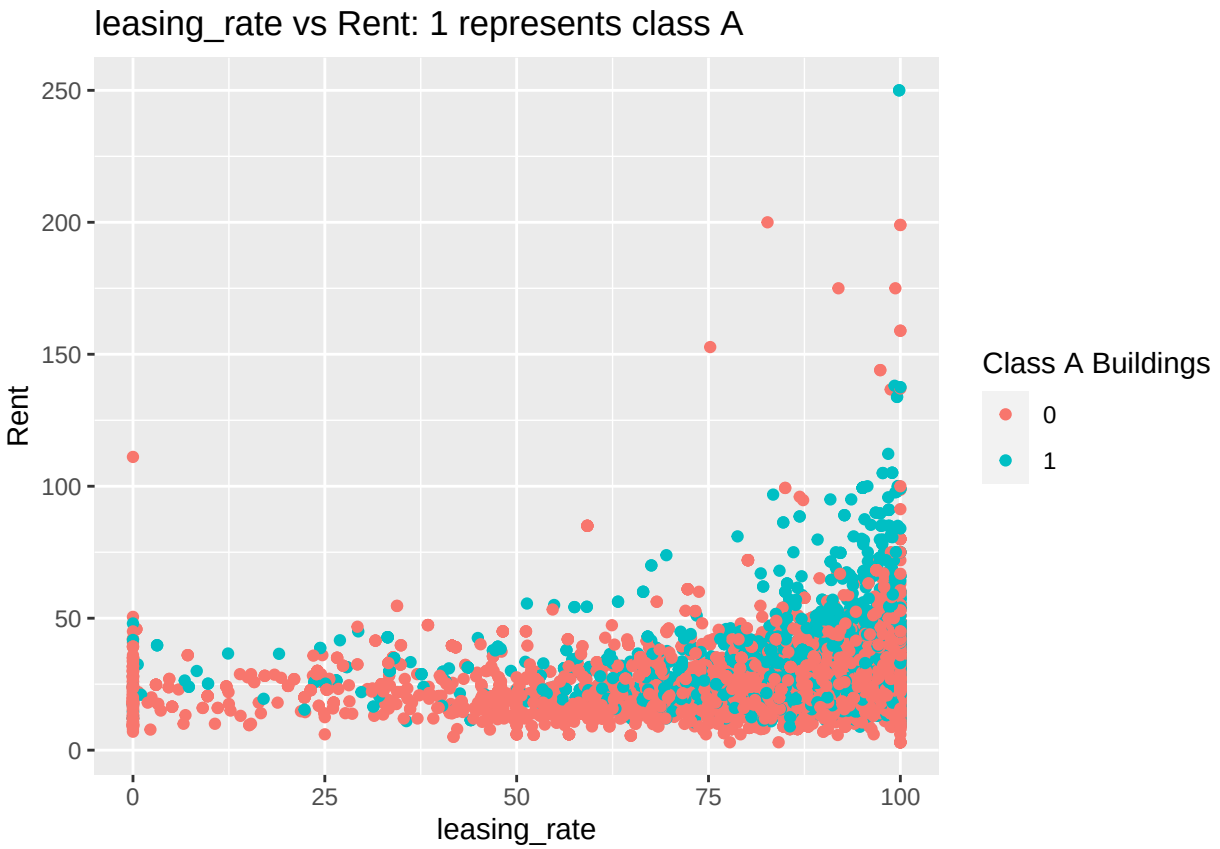
Even though we see significantly less green buildings data points here, the green buildings data points are scattered around the higher leasing rate range. In other words, green buildings have higher leasing rates.

```
green$green_rating = as.factor(green$green_rating)
ggplot(data = green) +
  geom_point(mapping=aes(x = leasing_rate, y = Rent, colour = green_rating))+
  labs(
    x = "leasing_rate",
    y = 'Rent',
    title = 'Leasing Rate vs Rent: 1 represents green buildings',
    color = 'Green building')
```



Similarly, Class A buildings data points are more concentrated in the higher leasing rate range. In other words, Class A buildings (higher quality buildings) have higher leasing rates.

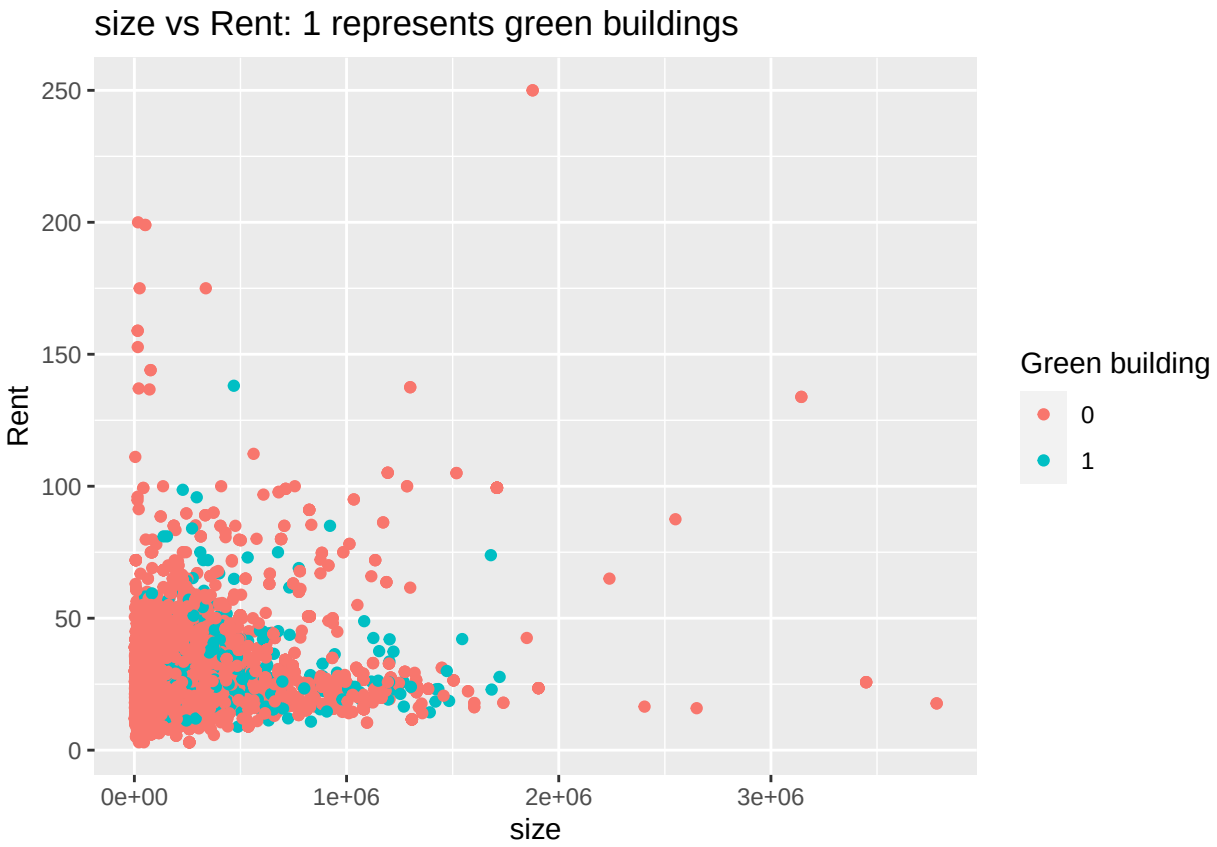
```
green$class_a = as.factor(green$class_a)
ggplot(data = green) +
  geom_point(mapping=aes(x = leasing_rate, y = Rent, colour = class_a))+
  labs(
    x = "leasing_rate",
    y = 'Rent',
    title = 'leasing_rate vs Rent: 1 represents class A',
    color = 'Class A Buildings')
```



Let's examine the relationship size versus rent

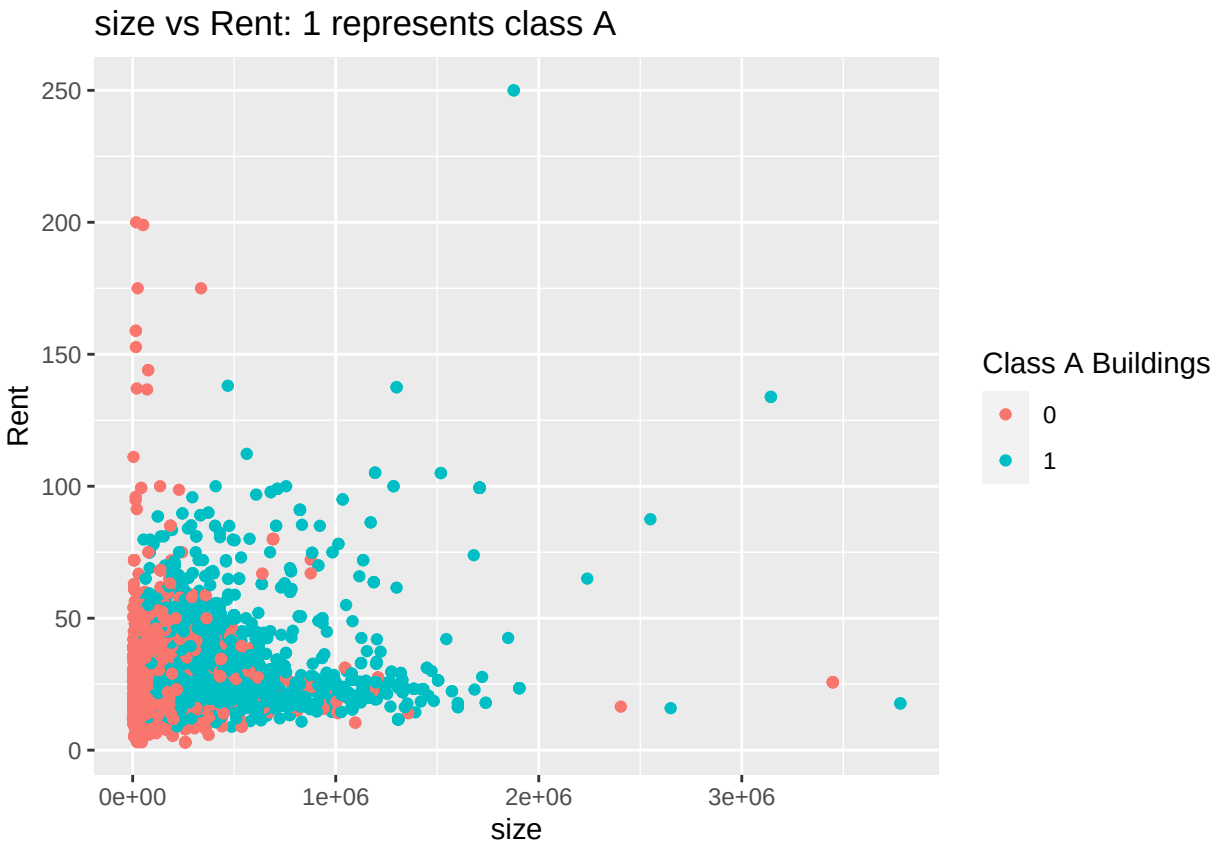
There are a handful of non-green buildings that seem to be outliers. Their rents are quite low considering its size relatively. No distinct difference when we compare green vs non-green here.

```
green$green_rating = as.factor(green$green_rating)
ggplot(data = green) +
  geom_point(mapping=aes(x = size, y = Rent, colour = green_rating))+
  labs(
    x = "size",
    y = 'Rent',
    title = 'size vs Rent: 1 represents green buildings',
    color = 'Green building')
```



Class A buildings tend to have larger size and charge higher rent.

```
green$class_a = as.factor(green$class_a)
ggplot(data = green) +
  geom_point(mapping=aes(x = size, y = Rent, colour = class_a))+
  labs(
    x = "size",
    y = 'Rent',
    title = 'size vs Rent: 1 represents class A',
    color = 'Class A Buildings')
```



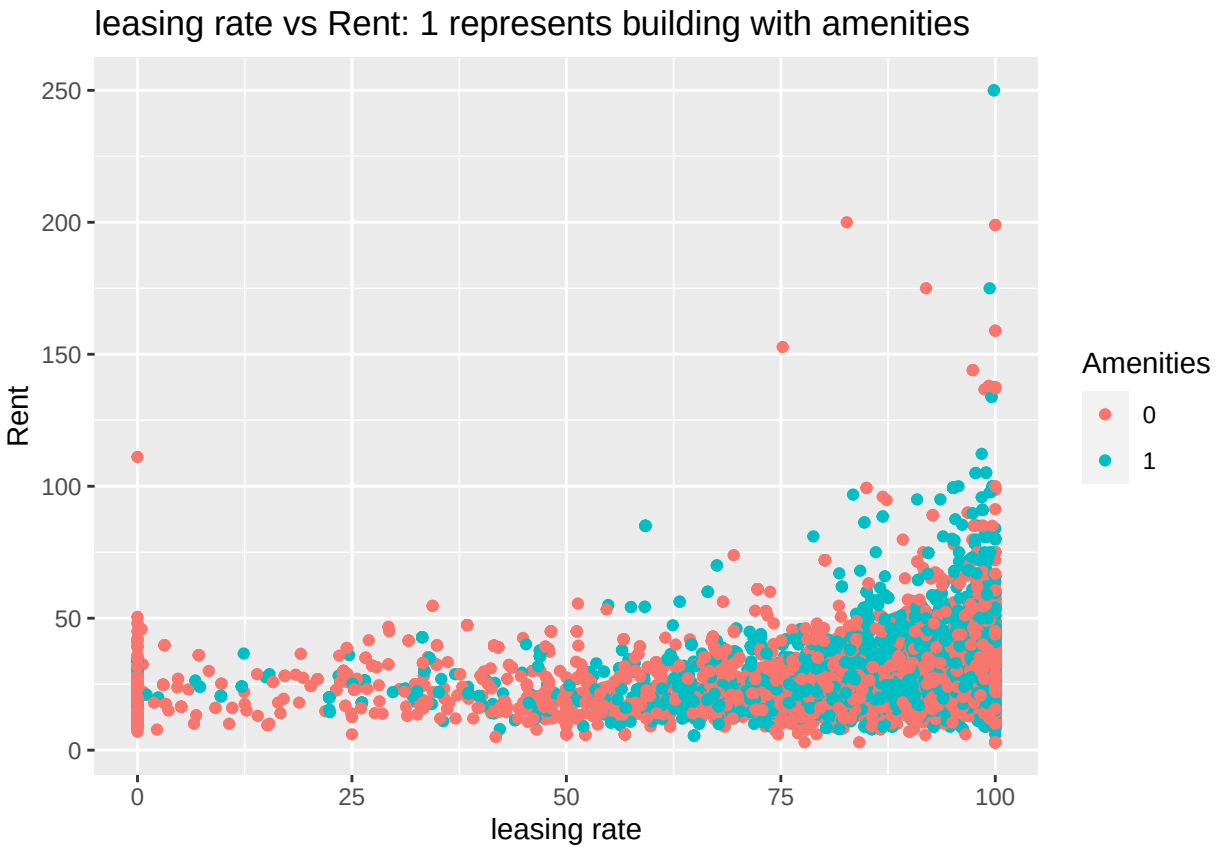
Let's look at Class A buildings and Green buildings

The proportion of Class A buildings is higher in green buildings. Based on what we observed above, Class A buildings tend to have higher rents, the higher median rent price in green buildings could potentially due to the higher proportion of Class A properties in green buildings. Class A is a potential confounding variable.

```
green$class_a = as.factor(green$class_a)
ggplot(green, aes(class_a, ..count..)) +
  geom_bar(aes(fill = green_rating), position = "dodge")+
  labs(
    x="0: Non-Class A Building, 1: Class A Buildings",
    y='Number of buildings',
    title = 'Class A vs Green Buildings',
    fill='Green building')
```

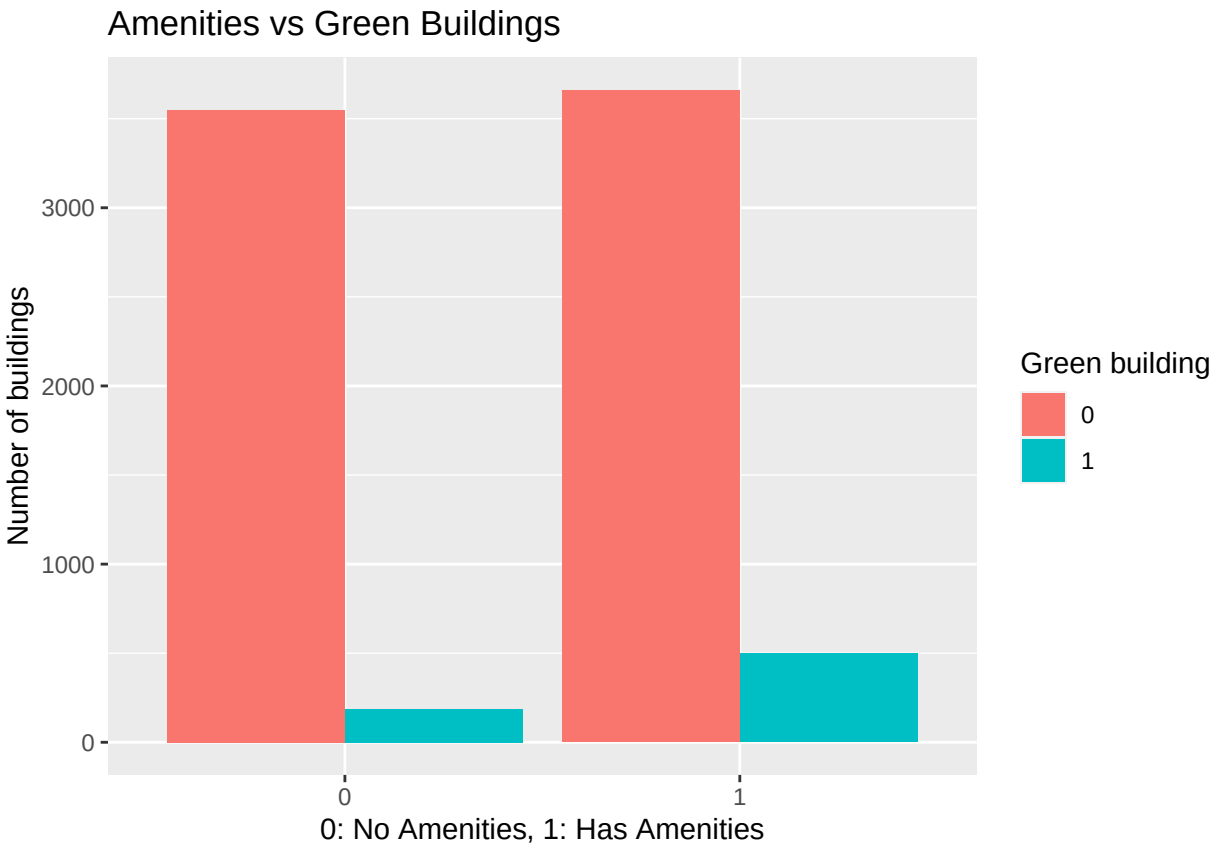
Let's look at buildings with amenities and Green buildings
Unsurprisingly, buildings with amenities have higher leasing rates.

```
green$green_rating = as.factor(green$green_rating)
green$amenities = as.factor(green$amenities)
ggplot(data = green) +
  geom_point(mapping=aes(x = leasing_rate, y = Rent, colour = amenities))+
  labs(
    x = "leasing rate",
    y = 'Rent',
    title = 'leasing rate vs Rent: 1 represents building with amenities',
    color = 'Amenities')
```



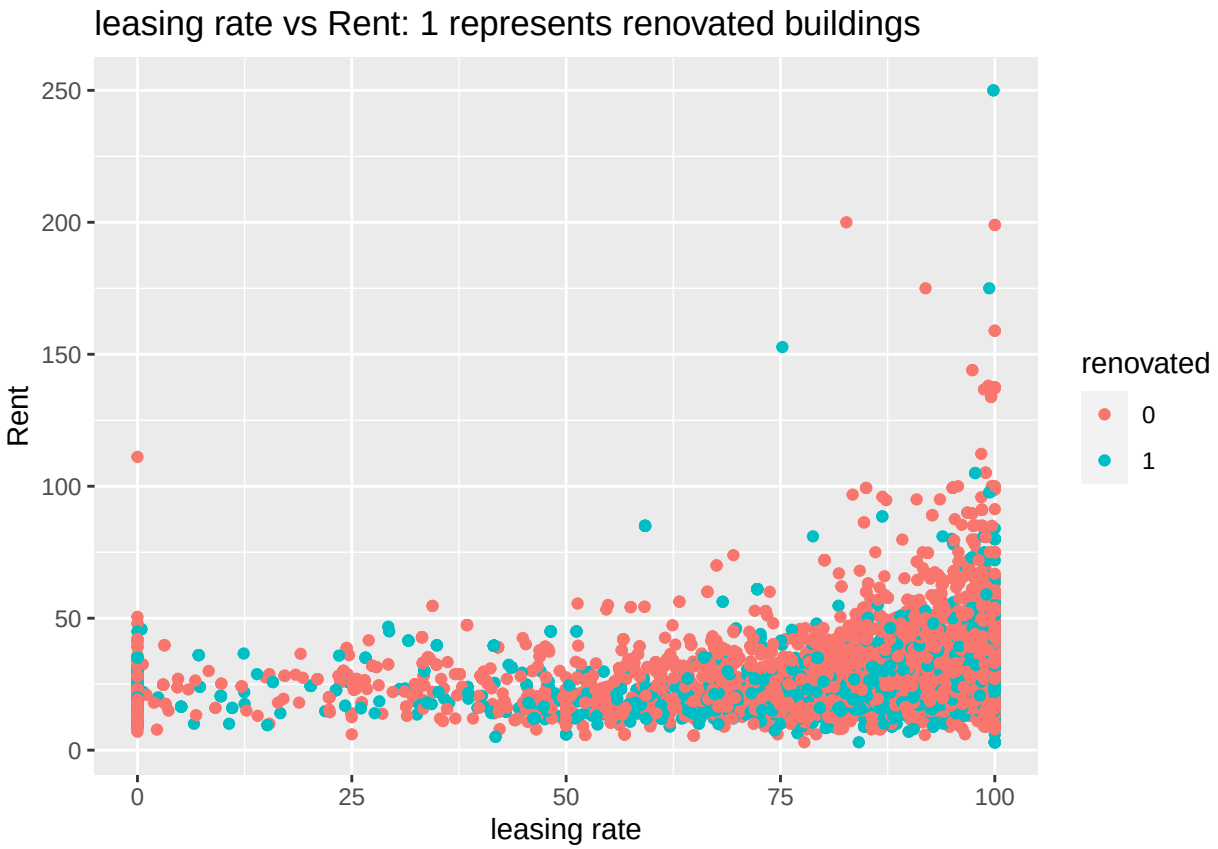
The proportion of buildings with amenities is higher in green buildings. Based on what we observed above, buildings with amenities tend to have higher rents, the higher median rent price in green buildings could potentially due to the higher proportion of buildings with amenities in green buildings. Amenities is another potential confounding variable.

```
green$class_a = as.factor(green$class_a)
green$amenities = as.factor(green$amenities)
ggplot(green, aes(amenities, ..count..)) +
  geom_bar(aes(fill = green_rating), position = "dodge")+
  labs(
    x="0: No Amenities, 1: Has Amenities",
    y='Number of buildings',
    title = 'Amenities vs Green Buildings',
    fill='Green building')
```



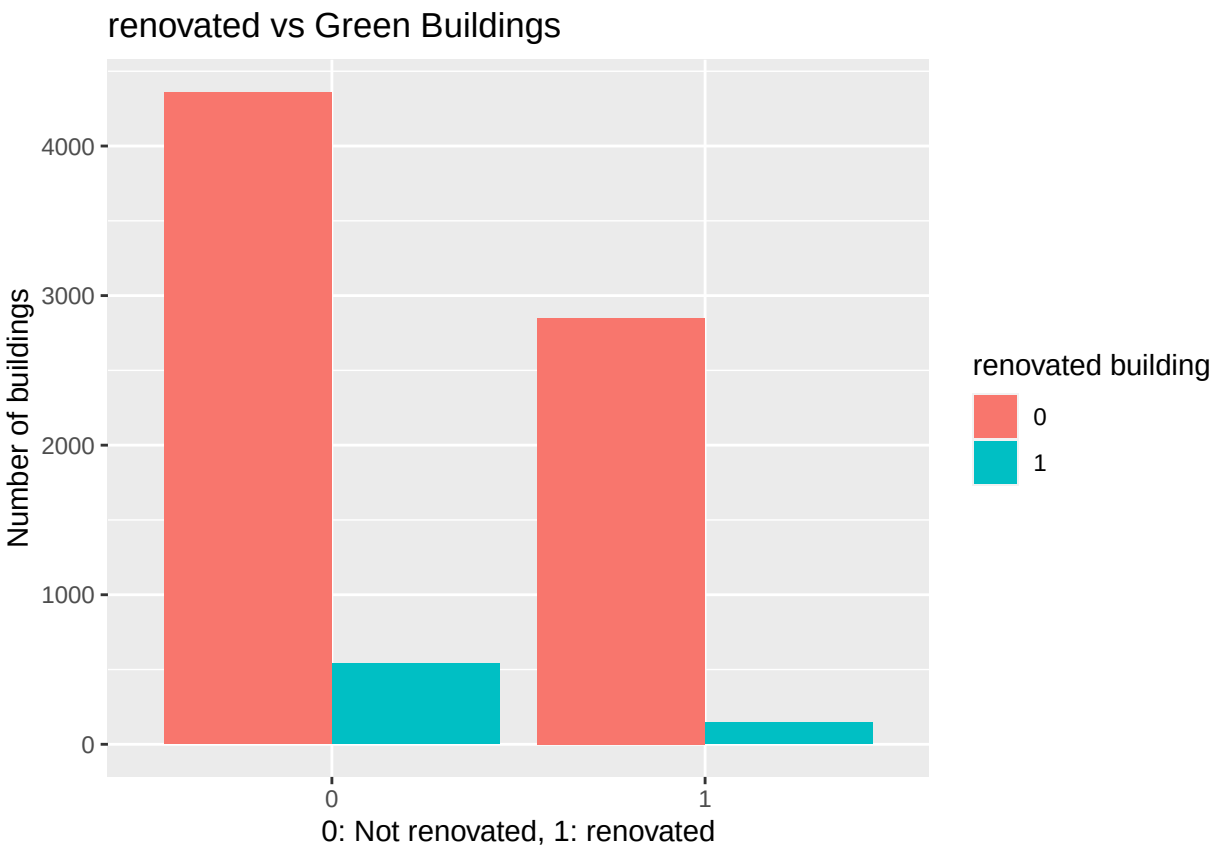
Let's look at the renovated buildings and Green buildings

```
green$green_rating = as.factor(green$green_rating)
green$renovated = as.factor(green$renovated)
ggplot(data = green) +
  geom_point(mapping=aes(x = leasing_rate, y = Rent, colour = renovated))+
  labs(
    x = "leasing rate",
    y = 'Rent',
    title = 'leasing rate vs Rent: 1 represents renovated buildings',
    color = 'renovated')
```



The proportion of renovated buildings is lower in green buildings. This might be because green buildings are newer which don't need a renovation yet.

```
green$class_a = as.factor(green$class_a)
green$renovated = as.factor(green$renovated)
ggplot(green, aes(renovated, ..count..)) +
  geom_bar(aes(fill = green_rating), position = "dodge")+
  labs(
    x="0: Not renovated, 1: renovated",
    y='Number of buildings',
    title = 'renovated vs Green Buildings',
    fill='renovated building')
```



Isolate Variables

Here we are only comparing the median rent prices for Green and Non-Green Buildings, and the difference is $2.60, as stated by the stat guru.

```
green_only = subset(green, green_rating==1)
nongreen_only = subset(green, green_rating==0)
print('Median rent difference in Green vs Non-Green')
```

```
## [1] "Median rent difference in Green vs Non-Green"
```

```
median(green_only$Rent) - median(nongreen_only$Rent)
```

```
## [1] 2.6
```

Let's isolate the Class variables: Class A, Class B
If we compare rent prices of green and non-green properties in Class A buildings only, the difference is only $0.24. Initially, the difference in rent prices when we compare the entire dataset the rent difference was $2.60. Here when we isolate the Class A variable, we see a minor difference.

```
green$class_a = as.factor(green$class_a)
A_only = subset(green, class_a==1)
nonA_only = subset(green, class_a==0)
classA = aggregate(Rent~ green_rating, A_only, median)
classA
```

```
##   green_rating  Rent
## 1            0 28.20
## 2            1 28.44
```

If we compare rent prices of green and non-green properties in Class B buildings only, the difference is $1.10. Initially, the difference in rent prices when we compare the entire dataset the rent difference was $2.60. Here when we isolate the Class B variable, we see a smaller difference.

```
green$class_b = as.factor(green$class_b)
B_only = subset(green, class_b==1)
nonB_only = subset(green, class_b==0)
classB = aggregate(Rent~ green_rating, B_only, median)
classB
```

```
##   green_rating Rent
## 1            0 24.0
## 2            1 25.1
```

Let's isolate the 'renovated' variable Renovated is potentially another confounding variable. The rent difference here is $3.545, which is slightly higher than using the entire dataset. The higher rent price for green buildings could be due to its renovated status.

```
renovated_only = subset(green, renovated==1)
nonrenovated_only = subset(green, renovated==0)
Renovated = aggregate(Rent~ green_rating, renovated_only, median)
Renovated
```

```
##   green_rating   Rent
## 1            0 23.500
## 2            1 27.045
```

Conclusion

There are flaws in the analysis by the stat guru because he failed to take in to consideration of all the other variables, just to name a few confounding variables: Class, renovated, Age, Size. The higher rent for green buildings is not solely due to its green status, it is because most green buildings are newer, have amenities, higher class.

Visual story telling part 2: flights at ABIA

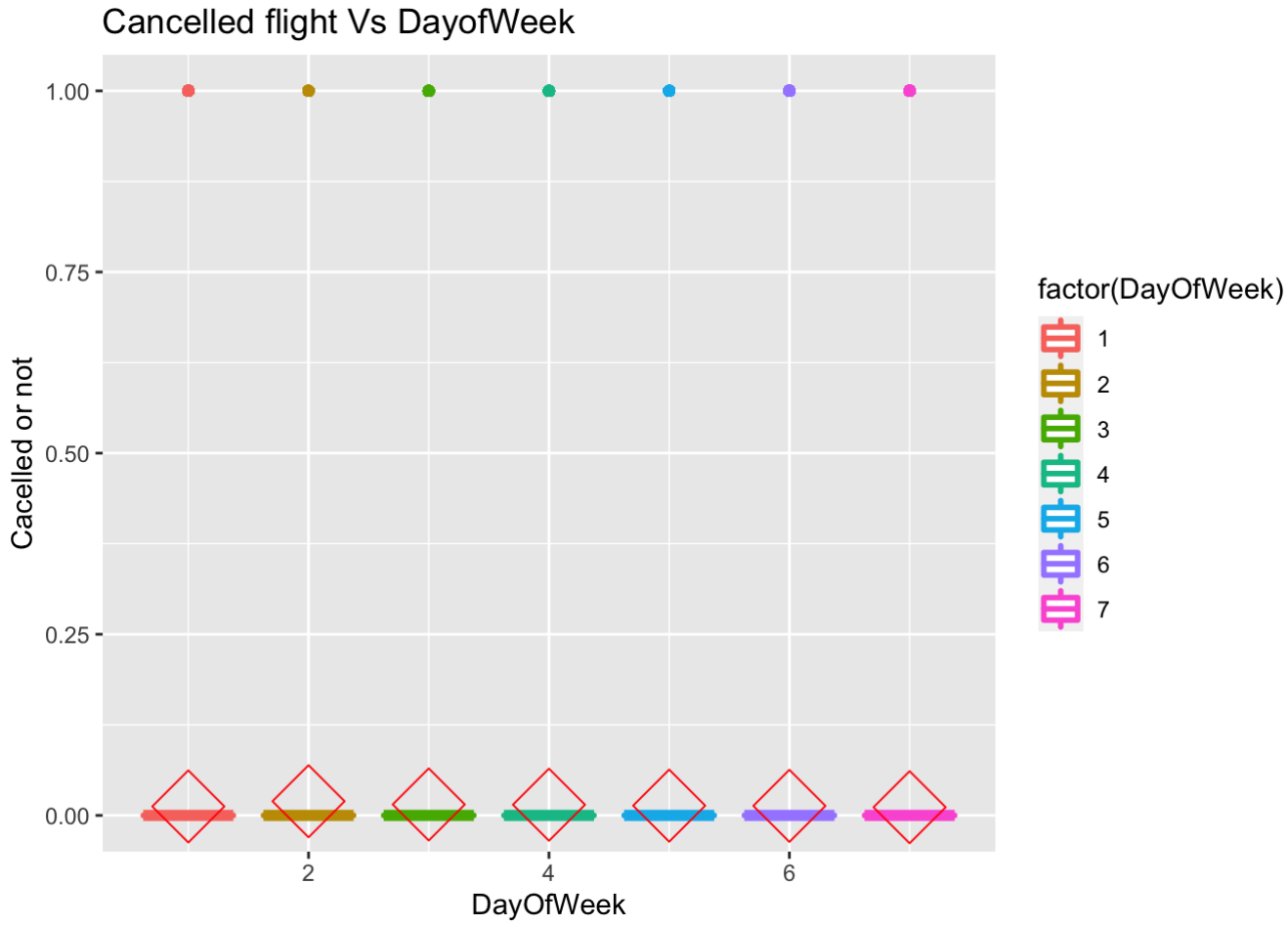
Story About Delay

```
library(dplyr)
library(ggplot2)
library("ggpubr")
library(ggmap)
```

```
Graph1 = ggplot(ABIA, aes(y=Cancelled,x=DayOfWeek))+
   geom_boxplot(aes(colour = factor(DayOfWeek)), size = 1)+
          stat_summary(fun.y=mean, geom="point", shape=23, size=10, color = "red")+
          ggtitle("Cancelled flight Vs DayofWeek") +
          ylab("Cacelled or not")
```

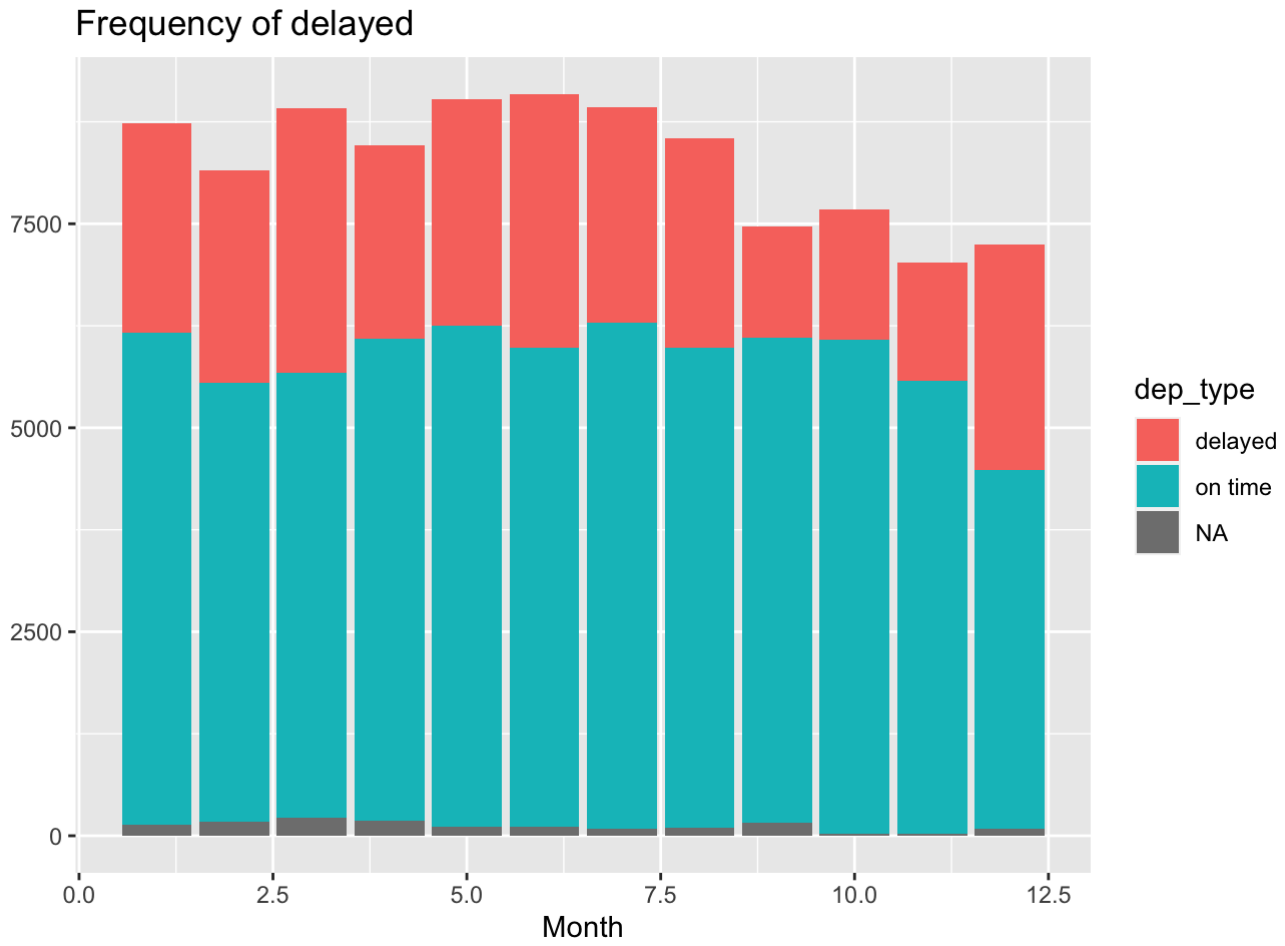
```
## Warning: `fun.y` is deprecated. Use `fun` instead.
```

```
Graph1
```



From this graph, we can tell that the delay of flight has no significant relationship with day of the week. We will further explor the dataset by showing graphs of other variabels.

```
ABIA_delay= ABIA%>%
mutate(dep_type=ifelse(DepDelay<5, "on time", "delayed"))
qplot(x=Month, fill=dep_type, data=ABIA_delay, geom="bar", main="Frequency of delayed")
```

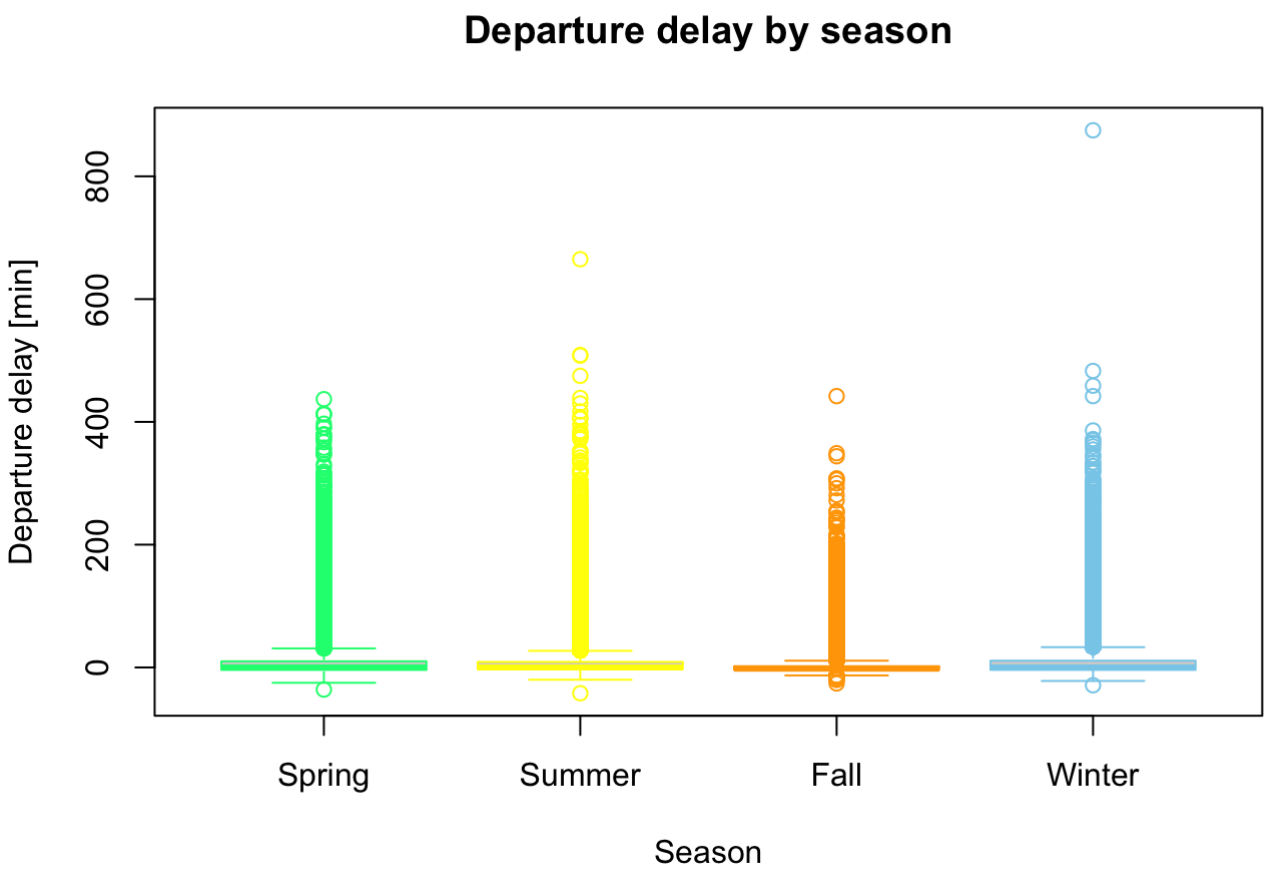


From this graph, we are able to see that passenger who travel in summer are more likely to encounter delay of departure time. I have revised the graph to show on time flights in blue and delayed flights in red. There is a significant decrease of number of delays after July.

```
ABIA$Season <- ABIA$Month
#recoding the spring months
ABIA$Season [ABIA$Season == 3] <- 100
ABIA$Season [ABIA$Season == 4] <- 100
ABIA$Season [ABIA$Season == 5] <- 100

#recoding the summer months
ABIA$Season [ABIA$Season == 6] <- 200
ABIA$Season [ABIA$Season == 7] <- 200
ABIA$Season [ABIA$Season == 8] <- 200

#recoding the fall months
ABIA$Season [ABIA$Season == 9] <- 300
ABIA$Season [ABIA$Season == 10] <- 300
ABIA$Season [ABIA$Season == 11] <- 300

#recoding the winter months
ABIA$Season [ABIA$Season == 12] <- 400
ABIA$Season [ABIA$Season == 1] <- 400
ABIA$Season [ABIA$Season == 2] <- 400

#checking whether everything went fine
table(ABIA$Season)
```

```
##
##   100   200   300   400
## 26400 26574 22156 24130
```

```
boxplot(formula = DepDelay ~ Season,
           data = ABIA,
        main = 'Departure delay by season',
        xlab = 'Season',
        ylab = 'Departure delay [min]',
        border = c('springgreen', 'yellow', 'orange', 'skyblue'),
        names = c('Spring', 'Summer', 'Fall', 'Winter'))
```



This graph could be sued to create a visualization of how delay is related to different seasons. The mean of delay is all around zero for all seasons but there are some outliers that are extremely high for both summer and winter. I'll have to look at the summary of the data in order to make a conclusion.

```
aggregated.mean.sd.median <- cbind(
mean = aggregate(formula = DepDelay ~ Season,
           data = ABIA,
           FUN = mean,
            na.rm = T),
sd = aggregate(formula = DepDelay ~ Season,
           data = ABIA,
           FUN = sd,
            na.rm = T),
median= aggregate(formula = DepDelay ~ Season,
           data = ABIA,
           FUN = median,
            na.rm = T)
)

aggregated.mean.sd.median
```

```
##   mean.Season mean.DepDelay sd.Season sd.DepDelay median.Season median.DepDelay
## 1         100     10.115349       100    31.72364           100               0
## 2         200     10.923777       200    34.41609           200               0
## 3         300      3.847408       300    21.63533           300              -2
## 4         400     11.125332       400    33.54880           400               0
```

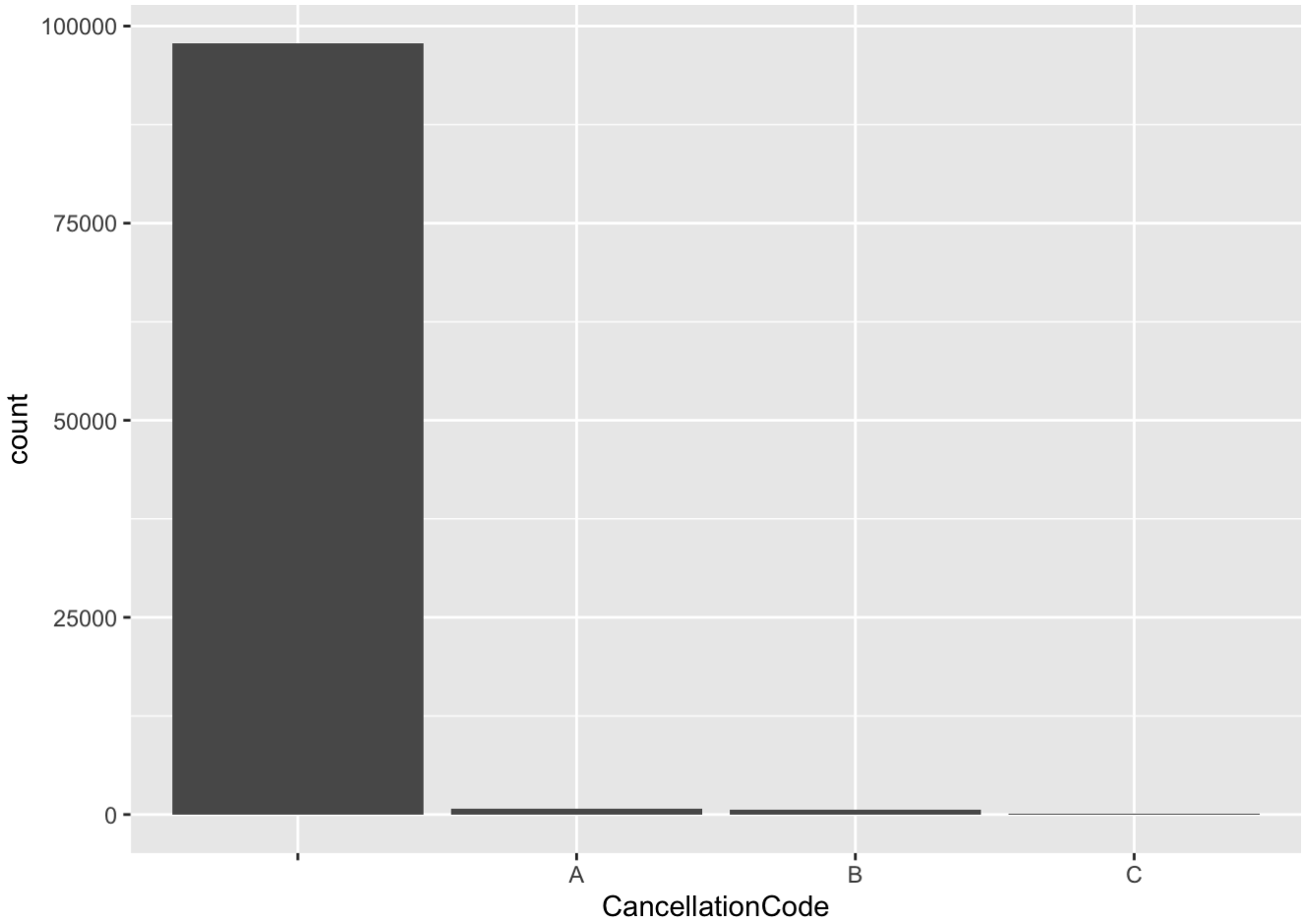
This chart tells us that Fall is the season that people are less likely to encounter delay of flights. Summer and Winter are the seasons that people will face some delay.

Reaons for cancelling flights

```
flights.per.Cancelreason <- cbind (Frequency  = table(ABIA$CancellationCode), RelFreq =
prop.table (table(ABIA$CancellationCode)))

flights.per.Cancelreason
```

```
##   Frequency     RelFreq
##       97840 0.985694137
## A       719 0.007243603
## B       605 0.006095104
## C        96 0.000967157
```

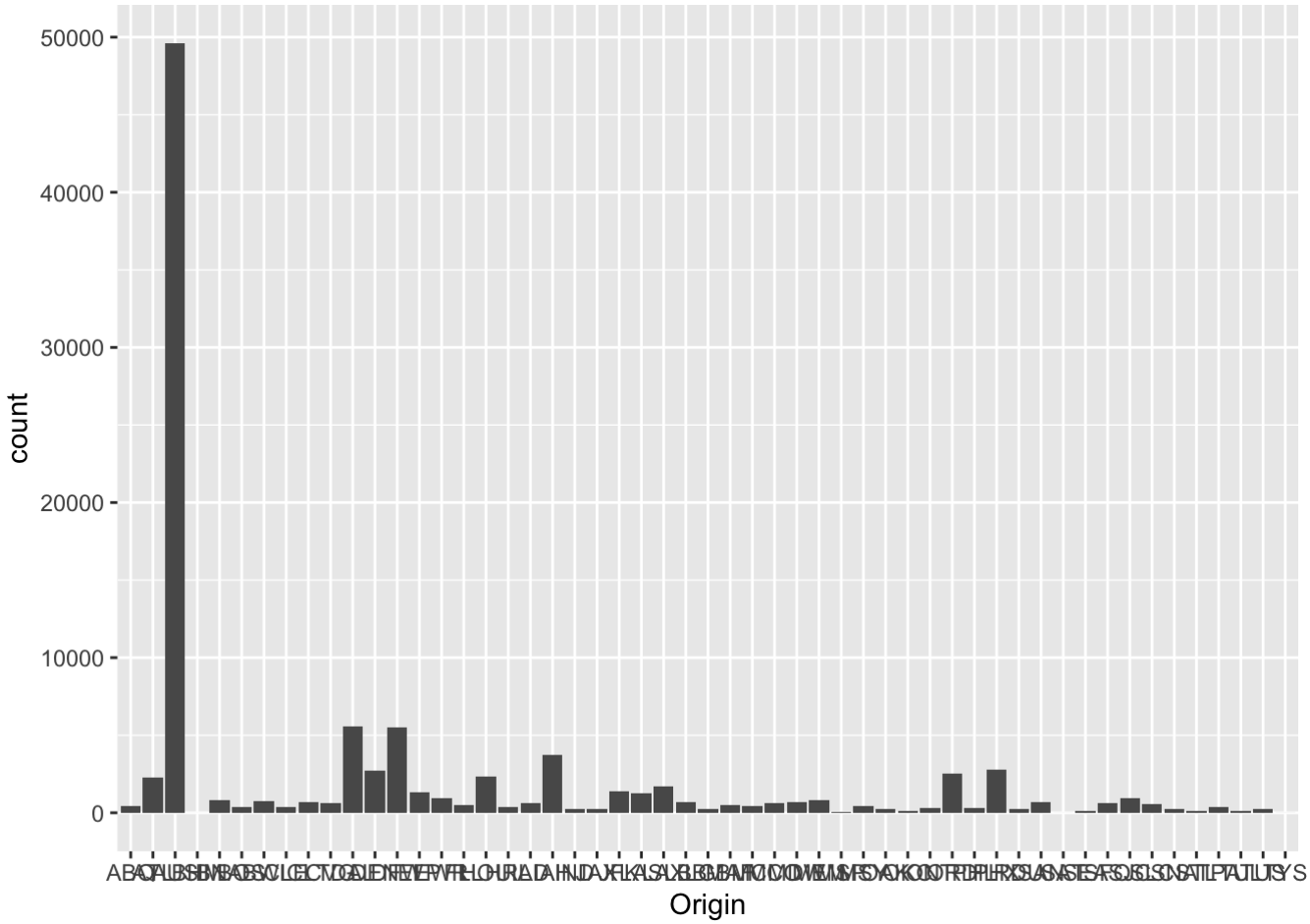
```
Graph6 = ggplot(ABIA, aes(CancellationCode))+
   geom_bar()

Graph6
```



There have been 719 cases because of Carrier, 605 because of Weather and 96 because of National Aviation System.

Destination and Origin of flights

```
Graph4 = ggplot(ABIA, aes(Origin))+
   geom_bar()

Graph4
```

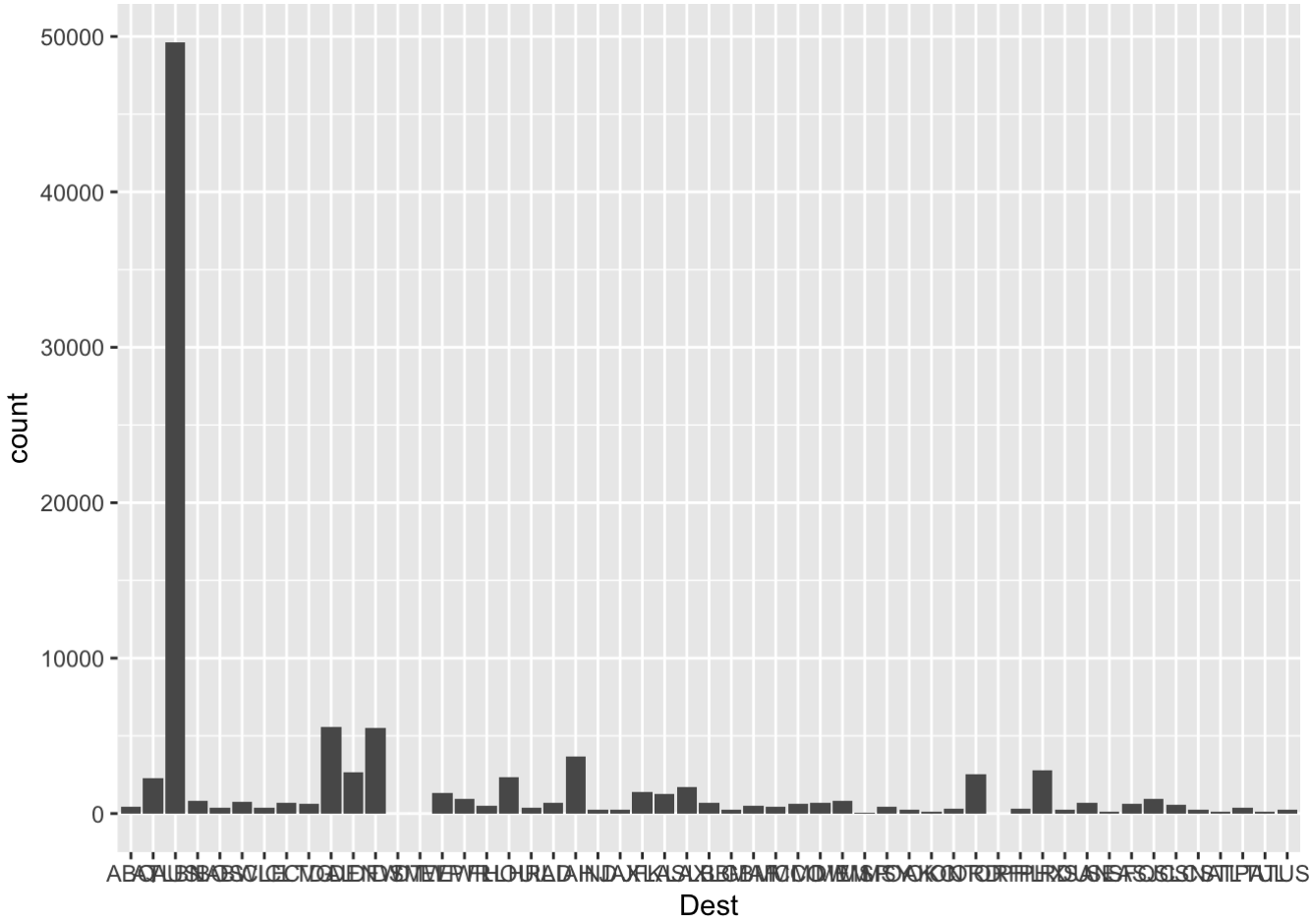



```
flights.per.Origin <- cbind (Frequency  = table(ABIA$Origin), RelFreq = prop.table (tabl
e(ABIA$Origin)))

flights.per.Origin
```

```
##     Frequency      RelFreq
## ABQ       433 4.362281e-03
## ATL      2255 2.271811e-02
## AUS     49623 4.999295e-01
## BHM         1 1.007455e-05
## BNA       795 8.009269e-03
## BOS       368 3.707435e-03
## BWI       728 7.334274e-03
## CLE       380 3.828330e-03
## CLT       660 6.649204e-03
## CVG       653 6.578682e-03
## DAL      5583 5.624622e-02
## DEN      2719 2.739271e-02
## DFW      5508 5.549063e-02
## ELP      1344 1.354020e-02
## EWR       939 9.460004e-03
## FLL       481 4.845859e-03
## HOU      2310 2.327221e-02
## HRL       366 3.687286e-03
## IAD       631 6.357042e-03
## IAH      3704 3.731614e-02
## IND       218 2.196252e-03
## JAX       229 2.307072e-03
## JFK      1356 1.366109e-02
## LAS      1232 1.241185e-02
## LAX      1732 1.744912e-02
## LBB       690 6.951441e-03
## LGB       245 2.468265e-03
## MAF       471 4.745114e-03
## MCI       459 4.624219e-03
## MCO       632 6.367117e-03
## MDW       713 7.183155e-03
## MEM       835 8.412251e-03
## MSP        54 5.440258e-04
## MSY       443 4.463026e-03
## OAK       236 2.377594e-03
## OKC        87 8.764860e-04
## ONT       304 3.062664e-03
## ORD      2515 2.533750e-02
## PHL       290 2.921620e-03
## PHX      2786 2.806770e-02
## RDU       231 2.327221e-03
## SAN       715 7.203304e-03
## SAT         2 2.014910e-05
## SEA       147 1.480959e-03
## SFO       609 6.135402e-03
## SJC       968 9.752166e-03
## SLC       550 5.541003e-03
## SNA       246 2.478340e-03
## STL        95 9.570824e-04
## TPA       367 3.697360e-03
## TUL        90 9.067097e-04
```

```
## TUS       229 2.307072e-03
## TYS         3 3.022366e-05
```

```
Graph5 = ggplot(ABIA, aes(Dest))+
   geom_bar()

Graph5
```




```
flights.per.Dest <- cbind (Frequency  = table(ABIA$Dest), RelFreq = prop.table (table(AB
IA$Dest)))

flights.per.Dest
```

```
##     Frequency      RelFreq
## ABQ       435 4.382430e-03
## ATL      2252 2.268789e-02
## AUS     49637 5.000705e-01
## BNA       792 7.979045e-03
## BOS       368 3.707435e-03
## BWI       730 7.354423e-03
## CLE       380 3.828330e-03
## CLT       659 6.639130e-03
## CVG       653 6.578682e-03
## DAL      5573 5.614548e-02
## DEN      2673 2.692928e-02
## DFW      5506 5.547048e-02
## DSM         1 1.007455e-05
## DTW         1 1.007455e-05
## ELP      1349 1.359057e-02
## EWR       949 9.560750e-03
## FLL       481 4.845859e-03
## HOU      2319 2.336289e-02
## HRL       367 3.697360e-03
## IAD       670 6.749950e-03
## IAH      3691 3.718517e-02
## IND       218 2.196252e-03
## JAX       226 2.276849e-03
## JFK      1358 1.368124e-02
## LAS      1231 1.240177e-02
## LAX      1733 1.745920e-02
## LBB       692 6.971590e-03
## LGB       245 2.468265e-03
## MAF       470 4.735039e-03
## MCI       459 4.624219e-03
## MCO       632 6.367117e-03
## MDW       712 7.173081e-03
## MEM       834 8.402176e-03
## MSP        55 5.541003e-04
## MSY       444 4.473101e-03
## OAK       236 2.377594e-03
## OKC        88 8.865605e-04
## ONT       305 3.072738e-03
## ORD      2514 2.532742e-02
## ORF         1 1.007455e-05
## PHL       290 2.921620e-03
## PHX      2783 2.803748e-02
## RDU       231 2.327221e-03
## SAN       719 7.243603e-03
## SEA       149 1.501108e-03
## SFO       610 6.145477e-03
## SJC       968 9.752166e-03
## SLC       548 5.520854e-03
## SNA       245 2.468265e-03
## STL        95 9.570824e-04
## TPA       367 3.697360e-03
```

```
## TUL        88 8.865605e-04
## TUS       228 2.296998e-03
```

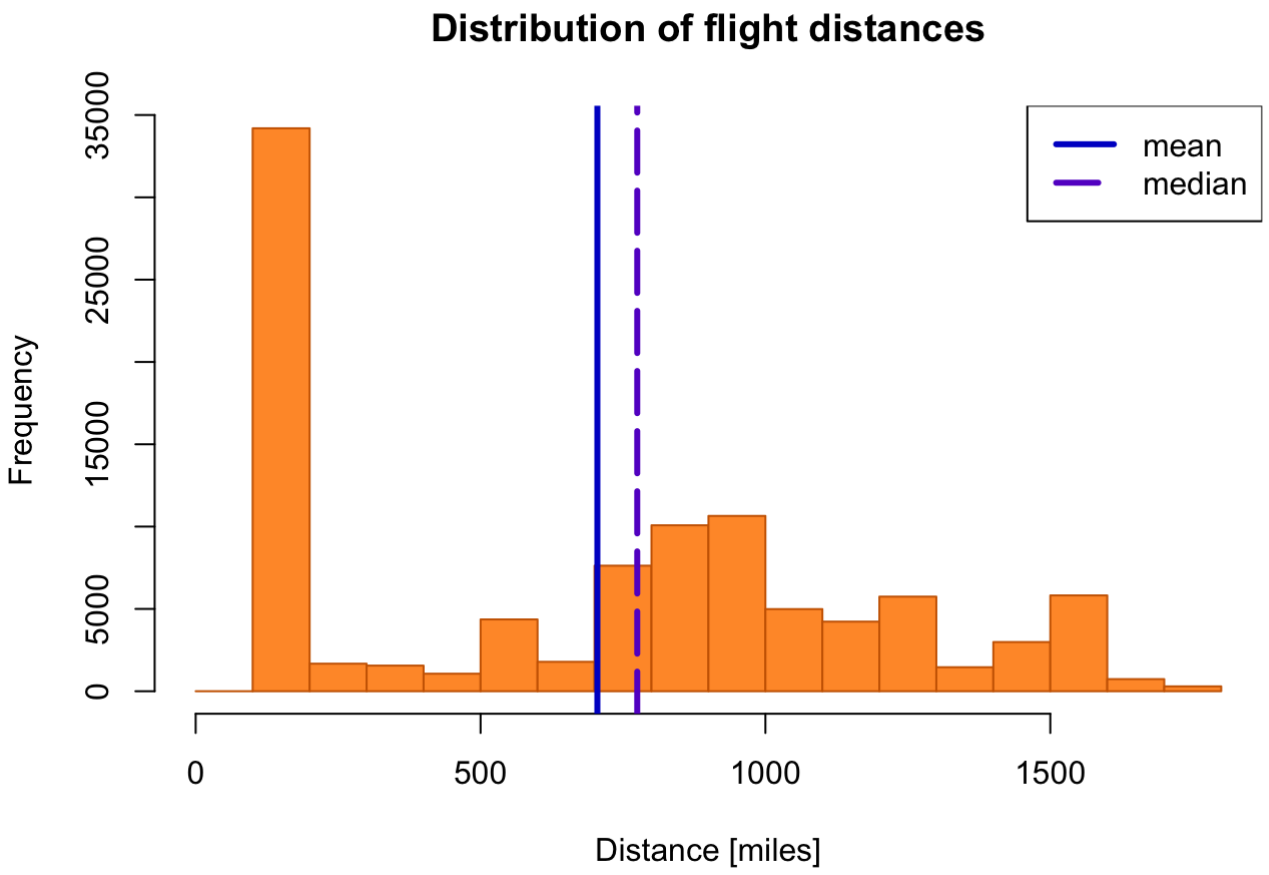
Most of flights are origin from AUS but their final destination is still AUS. That is caused by the operation of round trips offerd by Airlines.

Distribution of flight distance

```
library(Rgb)
```

```
## Warning: package 'Rgb' was built under R version 4.0.2
```

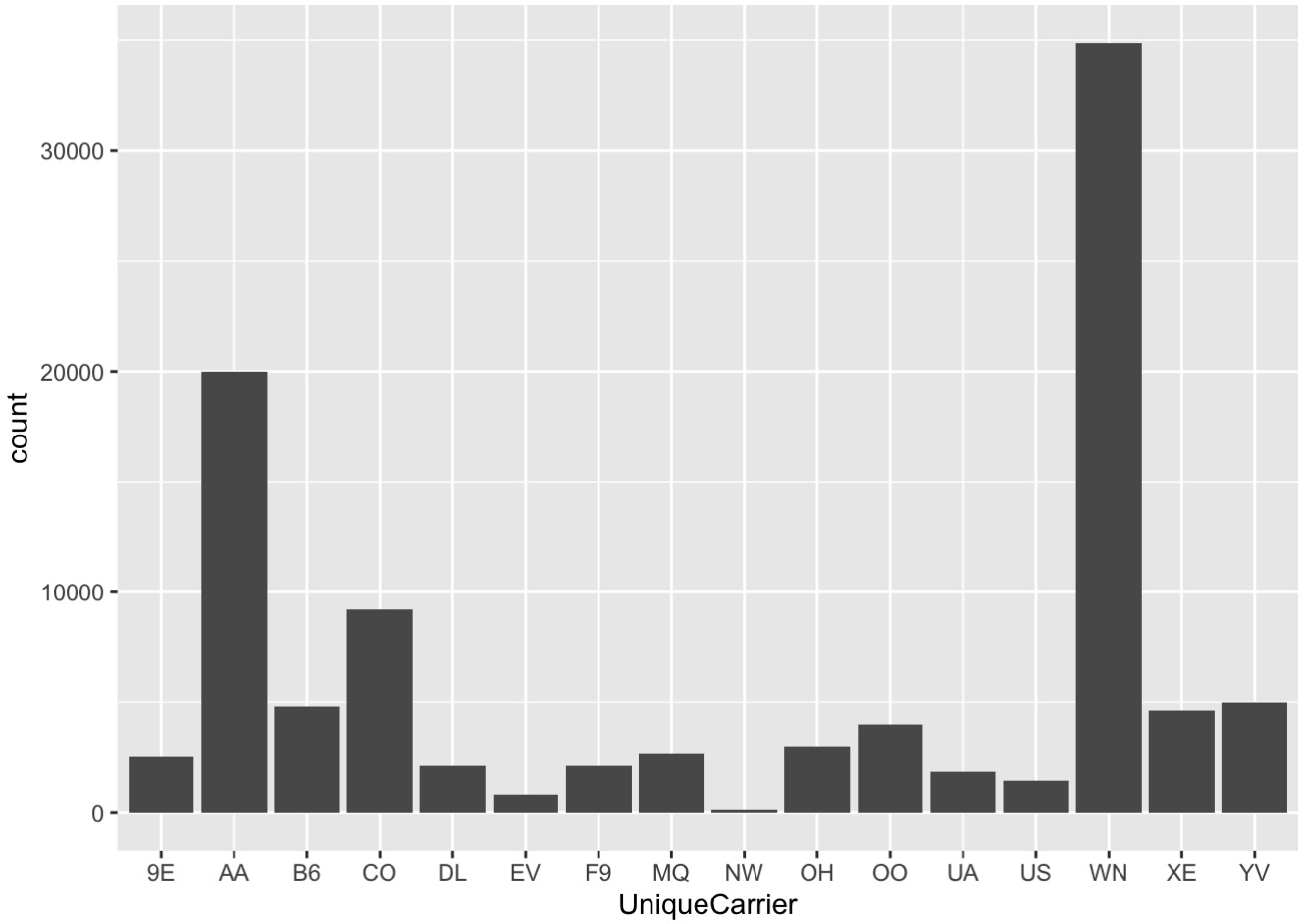
```
hist(ABIA$Distance,
     main = 'Distribution of flight distances',
     xlab = 'Distance [miles]',
    col = '#FF9933',
     border = '#CC6600')

abline(v=mean(ABIA$Distance),
       col = '#0000CC',
       lwd = 3)

abline(v= median(ABIA$Distance),
       col = '#6600CC',
       lty = 5,
       lwd = 3)
legend('topright',
       legend = c('mean', 'median'),
       lty = c(1,5),
       lwd = c(3,3),
       col = c('#0000CC', '#6600CC'))
```



The frequency of flight have mean and median around 750 miles but most frequent flights are below 250 miles.

Two Most Popular Airlines

```
Graph3 = ggplot(ABIA, aes(UniqueCarrier))+
   geom_bar()

Graph3
```




```
flights.per.carrier <- cbind (Frequency  = table(ABIA$UniqueCarrier), RelFreq = prop.tab
le (table(ABIA$UniqueCarrier)))

flights.per.carrier
```

```
##    Frequency     RelFreq
## 9E      2549 0.025680032
## AA     19995 0.201440661
## B6      4798 0.048337699
## CO      9230 0.092988112
## DL      2134 0.021499093
## EV       825 0.008311505
## F9      2132 0.021478944
## MQ      2663 0.026828531
## NW       121 0.001219021
## OH      2986 0.030082611
## OO      4015 0.040449325
## UA      1866 0.018799113
## US      1458 0.014688696
## WN     34876 0.351360064
## XE      4618 0.046524280
## YV      4994 0.050312311
```

From this graph and chart we can see that American Airlines and Southwest Airlines are the two most busy carrier at Austin Bergstrom Interational Airport in 2008. American Airlines has abourt 20% and Southwest Airlines has about 35%.

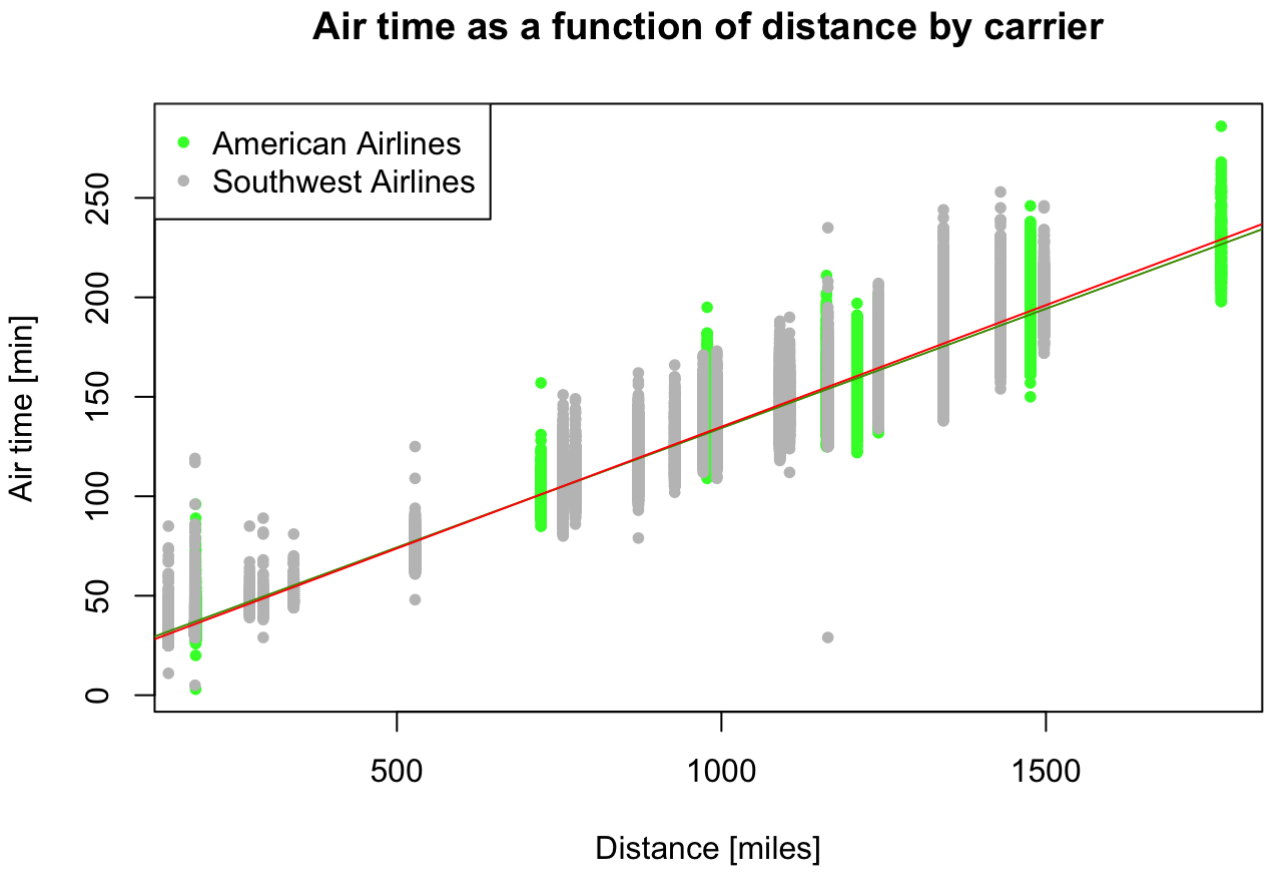
```
airtime.lm.AA <- lm(formula = AirTime ~ Distance, data = ABIA, subset = UniqueCarrier ==
'AA')
summary (airtime.lm.AA)
```

```
##
## Call:
## lm(formula = AirTime ~ Distance, data = ABIA, subset = UniqueCarrier ==
##     "AA")
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -41.469  -5.102  -0.102   3.898  63.309
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.430e+01  1.090e-01   131.1   <2e-16 ***
## Distance    1.200e-01  1.321e-04   909.0   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.587 on 19399 degrees of freedom
##   (594 observations deleted due to missingness)
## Multiple R-squared:  0.9771, Adjusted R-squared:  0.9771
## F-statistic: 8.263e+05 on 1 and 19399 DF,  p-value: < 2.2e-16
```

```
airtime.lm.WN <- lm(formula = AirTime ~ Distance, data = ABIA, subset = UniqueCarrier ==
'WN')
summary (airtime.lm.WN)
```

```
##
## Call:
## lm(formula = AirTime ~ Distance, data = ABIA, subset = UniqueCarrier ==
##     "WN")
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -126.029   -4.616   -0.759    4.227   83.227
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.266e+01  9.125e-02   138.7   <2e-16 ***
## Distance    1.223e-01  1.225e-04   998.2   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.741 on 34631 degrees of freedom
##   (243 observations deleted due to missingness)
## Multiple R-squared:  0.9664, Adjusted R-squared:  0.9664
## F-statistic: 9.964e+05 on 1 and 34631 DF,  p-value: < 2.2e-16
```

```
library(Rgb)
AA <- subset(ABIA, UniqueCarrier == 'AA')
WN <- subset(ABIA, UniqueCarrier == 'WN')

plot (x = AA$Distance,
      y = AA$AirTime,
      xlab = 'Distance [miles]',
      ylab = 'Air time [min]',
      main = 'Air time as a function of distance by carrier',
      pch=20,
      col='#33FF33'
      )
points (x = WN$Distance,
      y = WN$AirTime,
     pch=20,
      col='#C0C0C0'
      )

abline (airtime.lm.AA , col = '#4C9900')
abline (airtime.lm.WN, col = '#FF0000')

legend ('topleft',
        legend = c('American Airlines', 'Southwest Airlines'),
        col = c('#33FF33', '#C0C0C0'),
        pch = 20)
```



Conclusion

We can observe that people tend to choose Southwest Airlines for short distance travel which are less than 500 miles. For those flights that has travel distance more than 500 miles, there will be some flights of American Airlines but American Airlines have more air time than Southwest Airlines when traveling for similar distance. I believe that is why most people still chose Southwest Airlines to travel for longer distance.

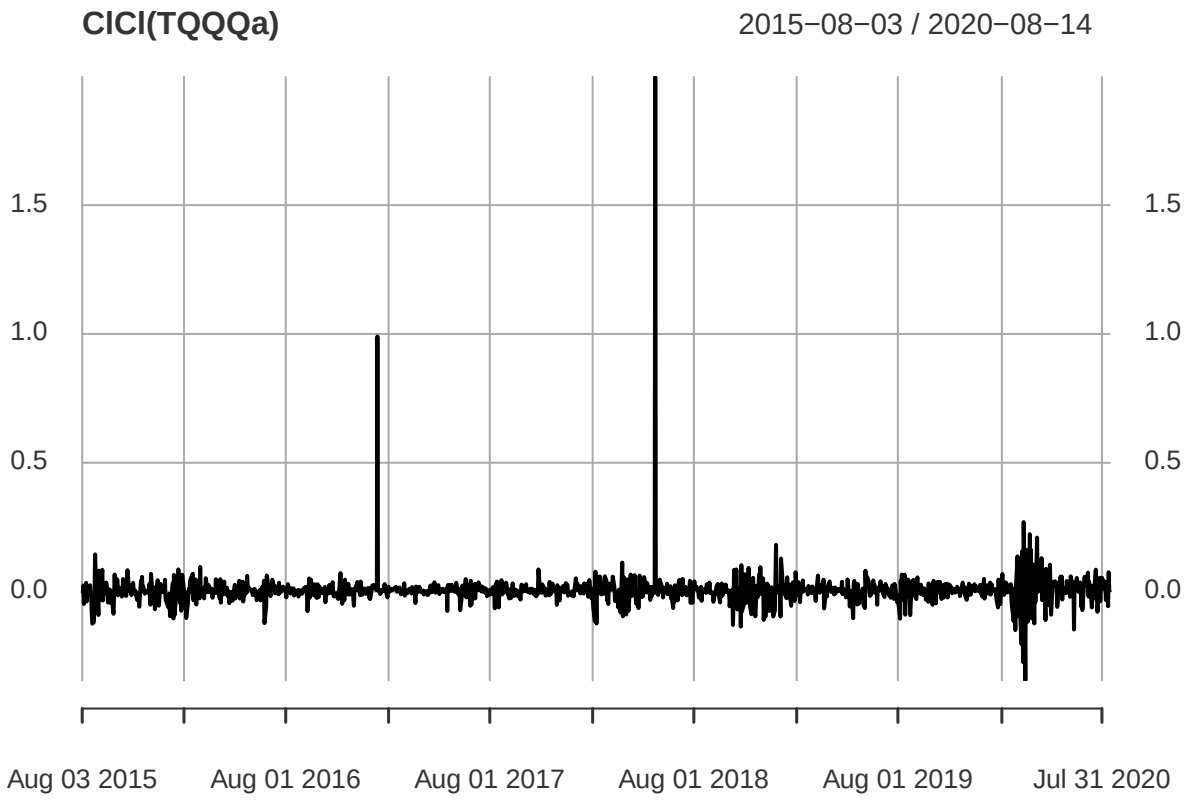
Portfolio modeling

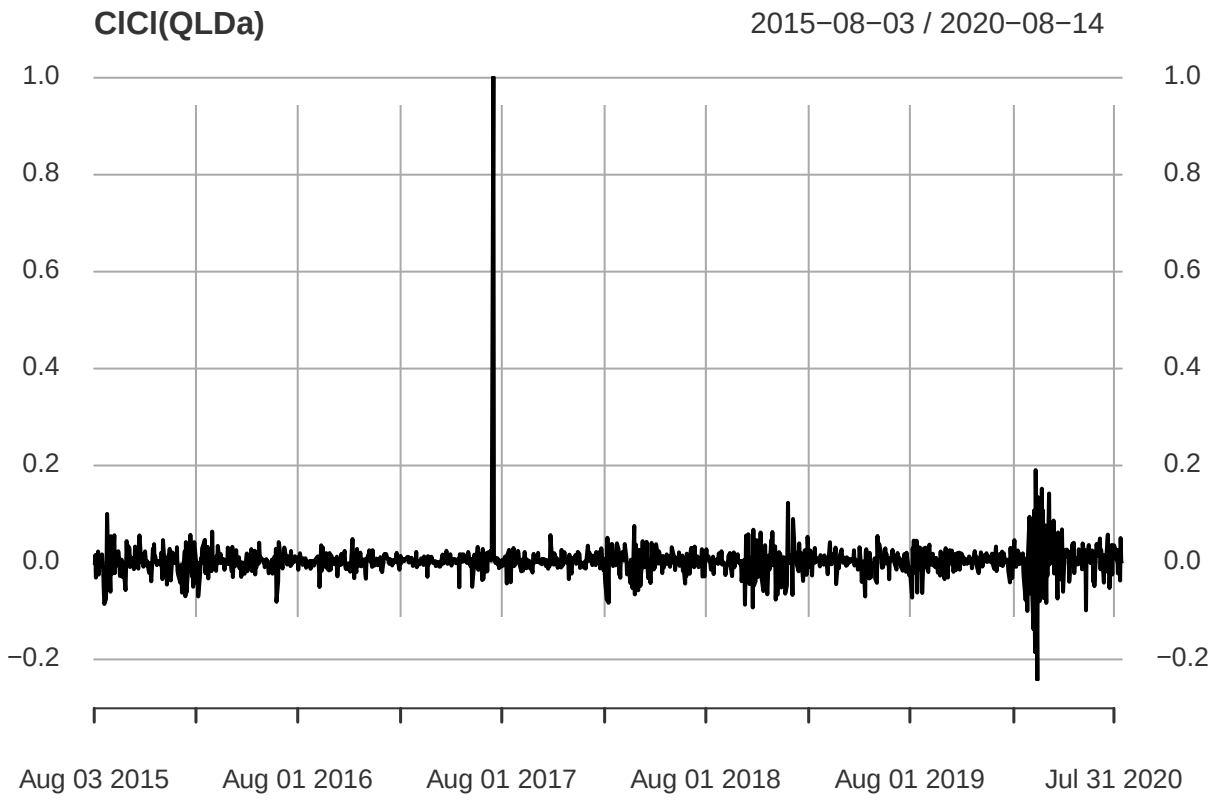
Aggresive Portfolio

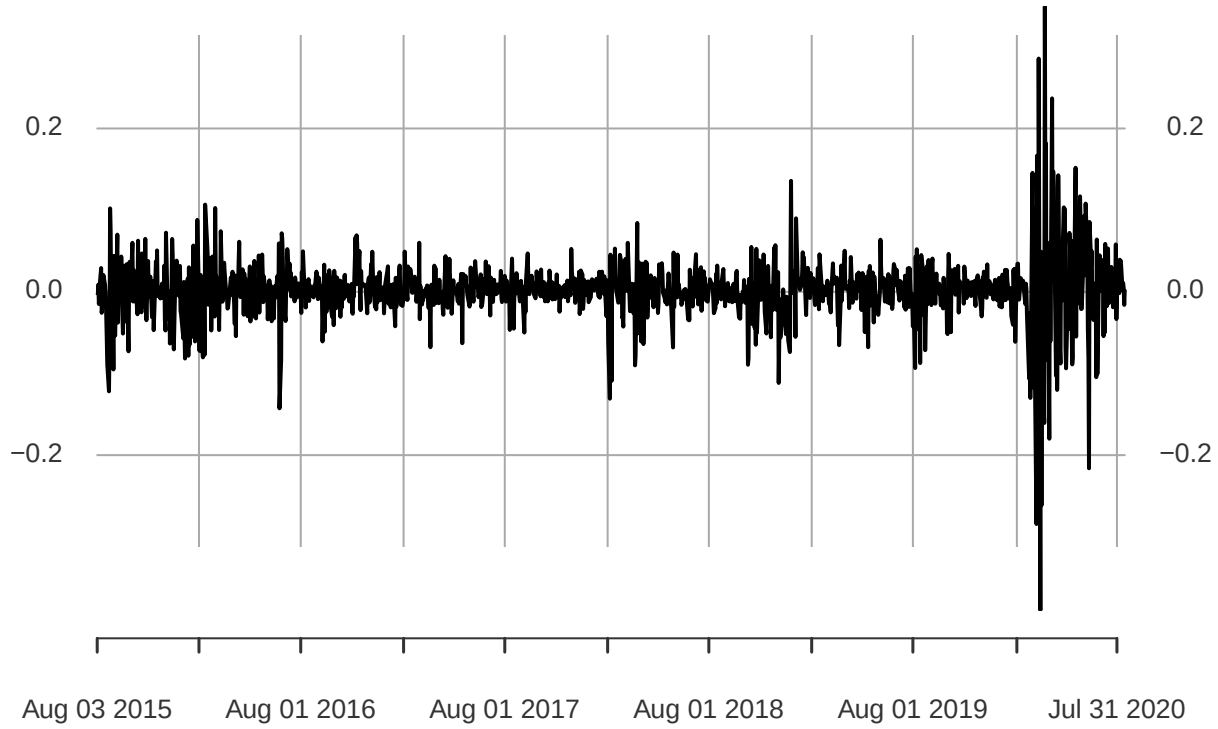
```
## Loading required package: dplyr

##
## 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

## Loading required package: lattice

## Loading required package: ggformula

## Loading required package: ggplot2

## Loading required package: ggstance

##
## Attaching package: 'ggstance'

## The following objects are masked from 'package:ggplot2':
##
##     geom_errorbarh, GeomErrorbarh

##
## New to ggformula?  Try the tutorials:
##  learnr::run_tutorial("introduction", package = "ggformula")
##  learnr::run_tutorial("refining", package = "ggformula")

## Loading required package: mosaicData

## Loading required package: Matrix

## Registered S3 method overwritten by 'mosaic':
##   method                           from
##   fortify.SpatialPolygonsDataFrame ggplot2

##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affected by this.
##
## Note: If you use the Matrix package, be sure to load it BEFORE loading mosaic.
##
## Have you tried the ggformula package for your plots?
```

```
##
## Attaching package: 'mosaic'

## The following object is masked from 'package:Matrix':
##
##     mean

## The following object is masked from 'package:ggplot2':
##
##     stat

## The following objects are masked from 'package:dplyr':
##
##     count, do, tally

## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median, prop.test,
##     quantile, sd, t.test, var

## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum

## Loading required package: xts

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

##
## Attaching package: 'xts'

## The following objects are masked from 'package:dplyr':
##
##     first, last

## Loading required package: TTR

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

## Version 0.4-0 included new data defaults. See ?getSymbols.
```

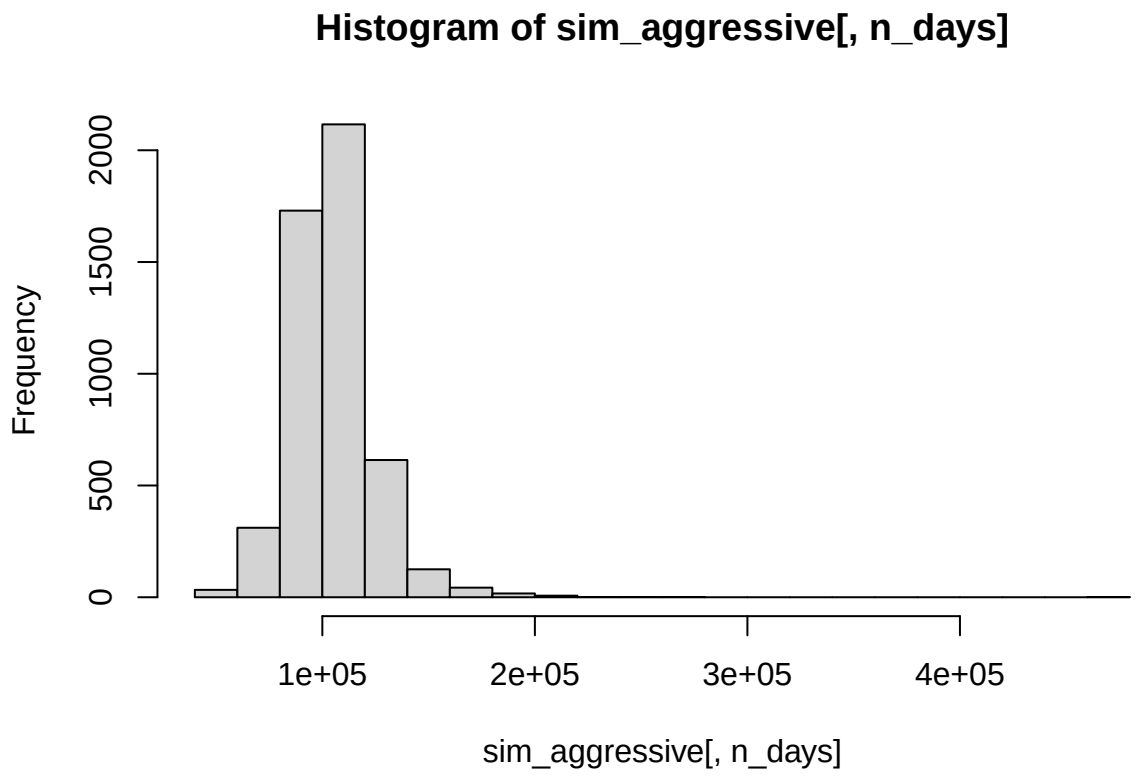
```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.

## Warning in read.table(file = file, header = header, sep = sep,
## quote = quote, : incomplete final line found by readTableHeader
## on 'https://query2.finance.yahoo.com/v7/finance/download/QLD?
## period1=-2208988800&period2=1597536000&interval=1d&events=split&crumb=nOzoZfrfynX'

## Warning in read.table(file = file, header = header, sep = sep,
## quote = quote, : incomplete final line found by readTableHeader
## on 'https://query2.finance.yahoo.com/v7/finance/download/QLD?
## period1=-2208988800&period2=1597536000&interval=1d&events=split&crumb=nOzoZfrfynX'
```




ClCl(QLDa)
2015−08−03 / 2020−08−14
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
Aug 03 2015
Aug 01 2016
Aug 01 2017
Aug 01 2018
Aug 01 2019
Jul 31 2020

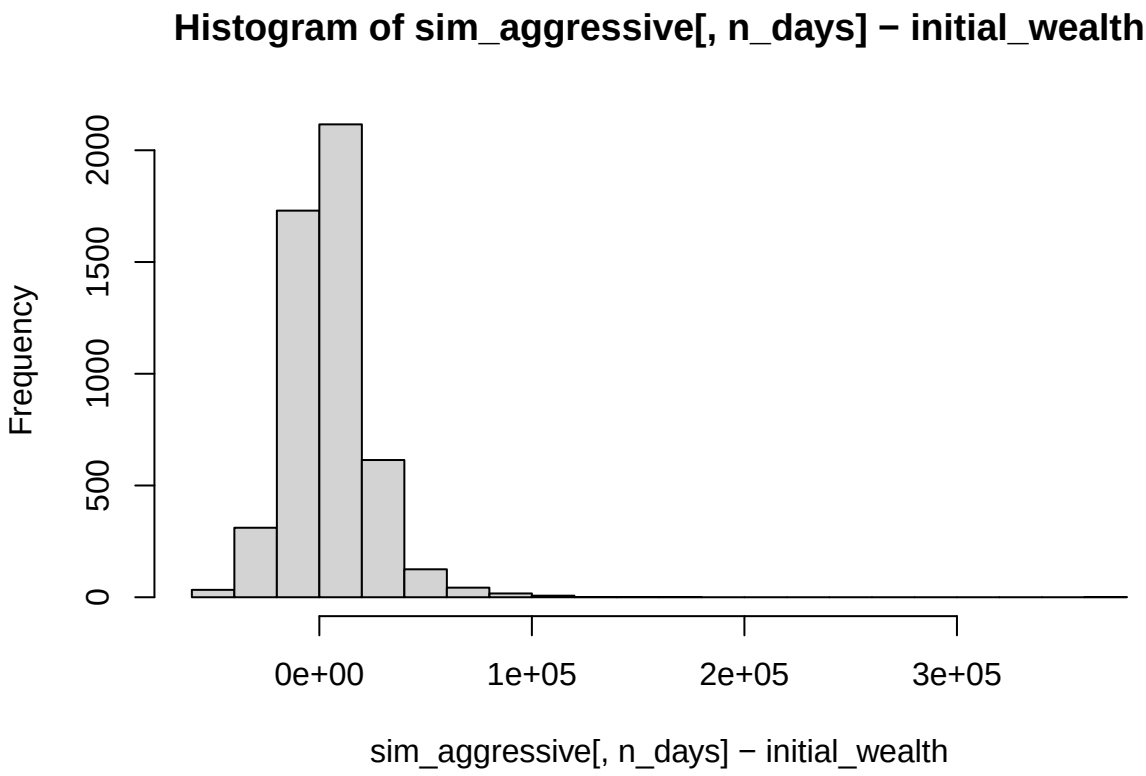



```
##               [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
## result.1 101307.84 104055.39 104059.24 102538.13 102702.18 103082.21 101120.90
## result.2 103400.08 106600.61 106296.04 111616.09 109454.85 110702.67 111054.84
## result.3 100930.64 101470.74 102578.91 104723.89 104849.64 106830.82 108834.21
## result.4  98704.62  91965.27  95177.72  94803.65  95291.18  94098.99  94860.11
## result.5  98721.10 101931.98  98732.55 100311.52  99144.13 108054.39 107885.28
## result.6  99295.83 101356.01 101655.90 101654.01 100473.73 159876.80 161163.96
##               [,8]      [,9]     [,10]     [,11]     [,12]     [,13]     [,14]
## result.1 100019.55 100562.77 101470.60 100384.09 101148.26 100487.28 103994.62
## result.2 110955.14 106891.93 103336.29 104798.75 105066.87 106373.20 108251.07
## result.3 109175.91 109172.54 115174.39 112378.85 109911.07 114069.67 117615.58
## result.4  96821.39  95584.21  95828.78  88899.43  90185.21  89477.73  92781.81
## result.5 106557.57 101242.06 100616.80 102030.80 100272.59 102961.63 104958.25
## result.6 165698.39 166513.32 169046.88 167960.72 169203.01 172348.94 172220.31
##              [,15]     [,16]     [,17]     [,18]     [,19]     [,20]
## result.1 111692.33 110948.2 109448.95 109803.42 111633.97 112158.66
## result.2 106608.83 106300.7 110590.74 111479.58 106854.23 113551.20
## result.3 122416.03 123718.6 129912.90 131952.20 135658.05 134774.15
## result.4  93799.72  93888.2  92659.94  92329.44  91445.58  95650.53
## result.5 106366.02 107522.7 110477.88 108087.13 105734.08 109528.77
## result.6 172772.14 169680.3 175246.09 176084.45 180086.47 183088.14
```




```
## [1] 104709.1
```

```
## [1] 4709.133
```



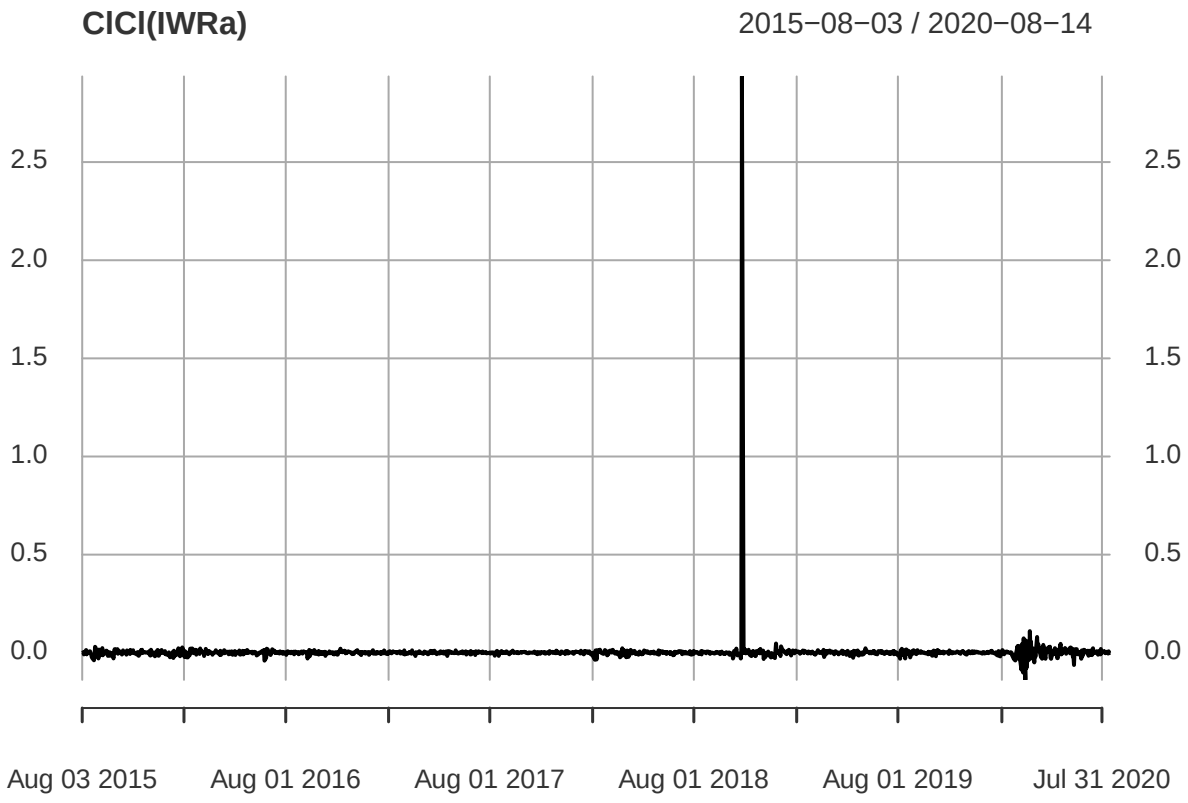

```
##        5%
## -22646.87
```

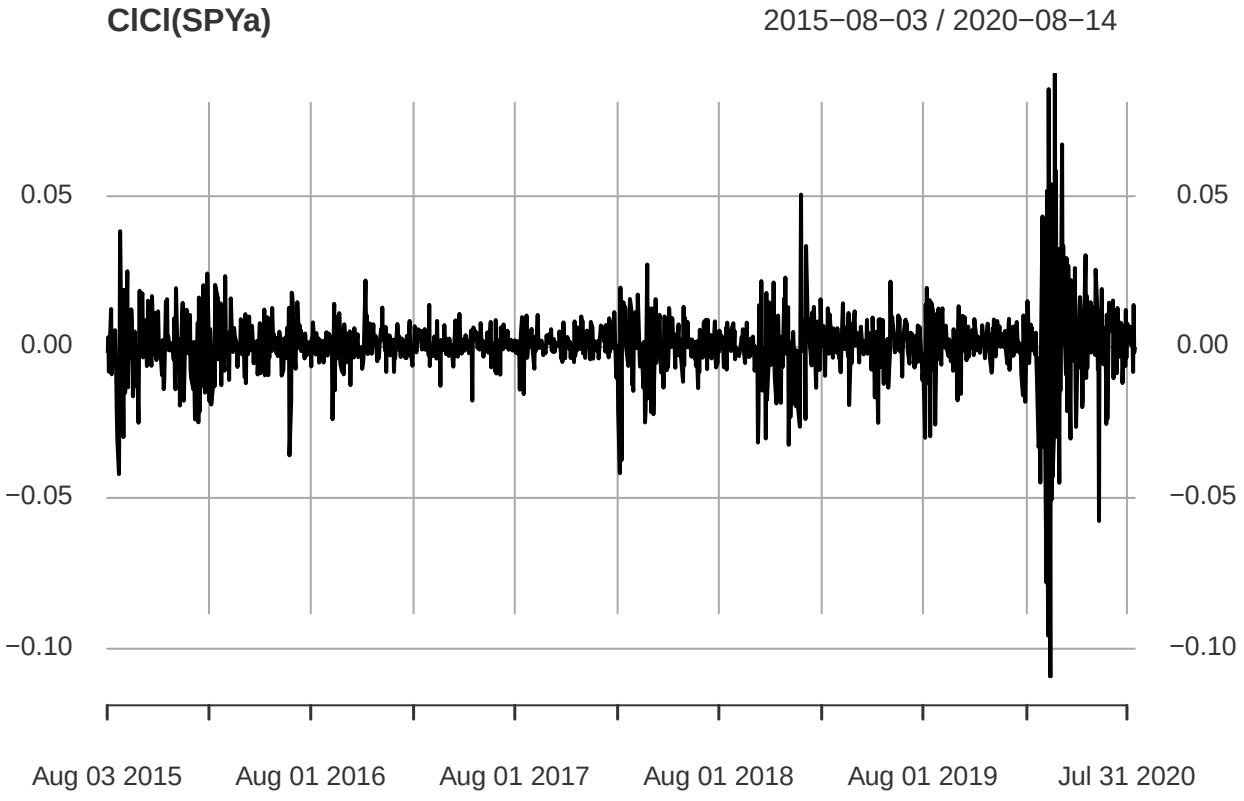
First, constructed our portfolio using three leveraged funds, we can see from the CLCL plot that TQQQ,QLD,FAS are all volatile. As a result, using bootstrap resampling to estimate the 4-week VAR, we got a VAR of $21087 at 5% level of the aggressive portfolio.

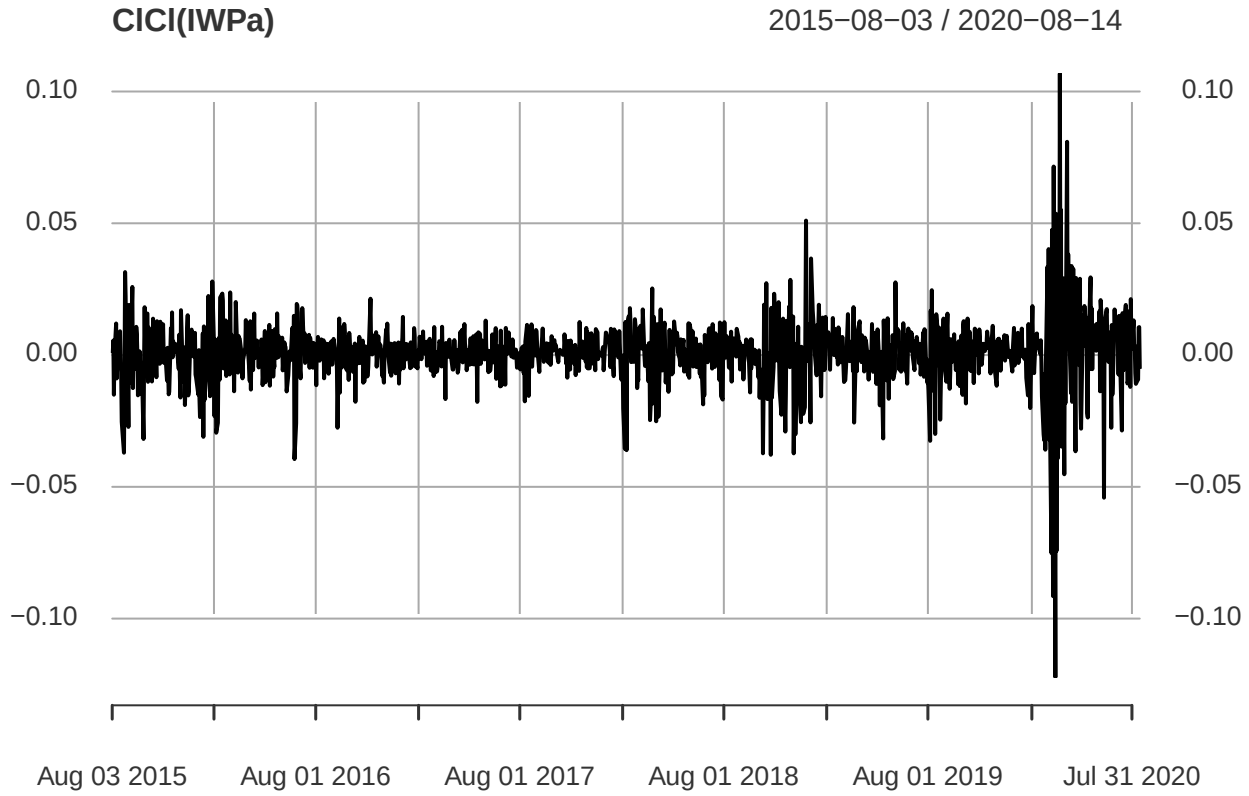
Diverse Portfolio

```
## Warning in read.table(file = file, header = header, sep = sep,
## quote = quote, : incomplete final line found by readTableHeader
## on 'https://query1.finance.yahoo.com/v7/finance/download/IWR?
## period1=-2208988800&period2=1597536000&interval=1d&events=split&crumb=nOzoZfrfynX'

## Warning in read.table(file = file, header = header, sep = sep,
## quote = quote, : incomplete final line found by readTableHeader
## on 'https://query1.finance.yahoo.com/v7/finance/download/IWR?
## period1=-2208988800&period2=1597536000&interval=1d&events=split&crumb=nOzoZfrfynX'

## Warning in read.table(file = file, header = header, sep = sep,
## quote = quote, : incomplete final line found by readTableHeader
## on 'https://query1.finance.yahoo.com/v7/finance/download/SPY?
## period1=-2208988800&period2=1597536000&interval=1d&events=split&crumb=nOzoZfrfynX'
```

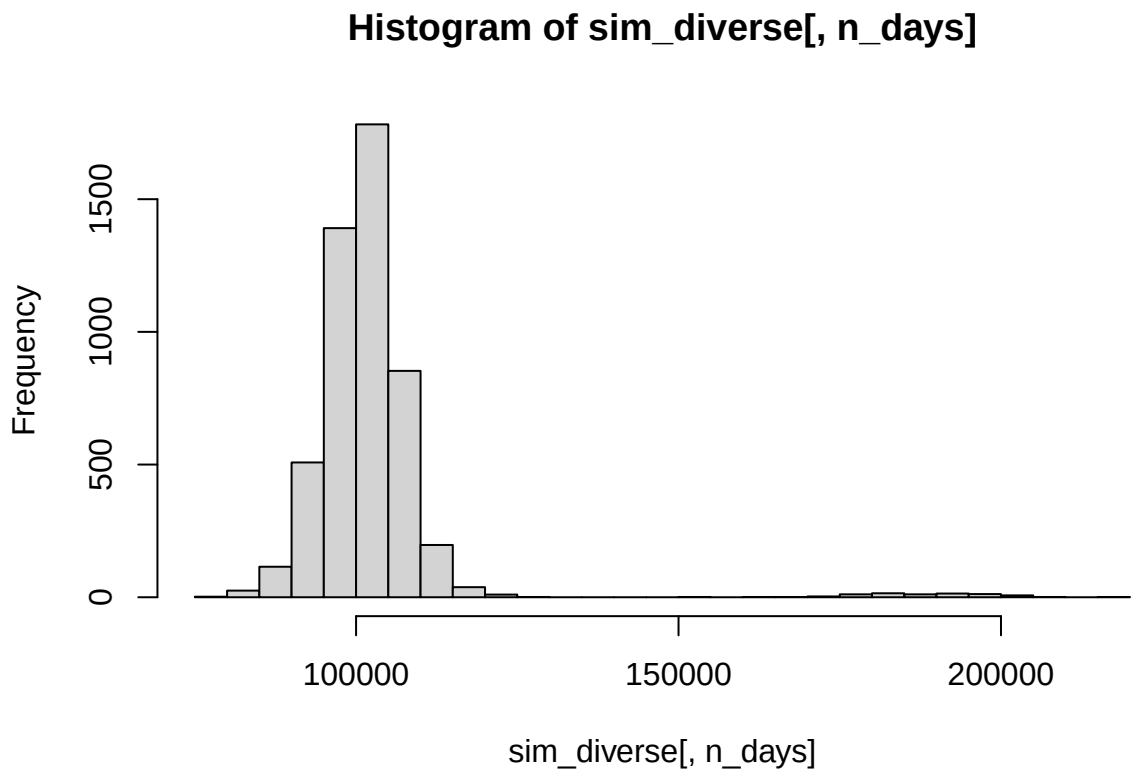
```
## Warning in read.table(file = file, header = header, sep = sep,
## quote = quote, : incomplete final line found by readTableHeader
## on 'https://query2.finance.yahoo.com/v7/finance/download/SPY?
## period1=-2208988800&period2=1597536000&interval=1d&events=split&crumb=nOzoZfrfynX'

## Warning in read.table(file = file, header = header, sep = sep,
## quote = quote, : incomplete final line found by readTableHeader
## on 'https://query2.finance.yahoo.com/v7/finance/download/IWP?
## period1=-2208988800&period2=1597536000&interval=1d&events=split&crumb=nOzoZfrfynX'

## Warning in read.table(file = file, header = header, sep = sep,
## quote = quote, : incomplete final line found by readTableHeader
## on 'https://query2.finance.yahoo.com/v7/finance/download/IWP?
## period1=-2208988800&period2=1597536000&interval=1d&events=split&crumb=nOzoZfrfynX'
```




ClCl(SPYa)
2015−08−03 / 2020−08−14
0.05
0.00
−0.05
−0.10
Aug 03 2015
Aug 01 2016
Aug 01 2017
Aug 01 2018
Aug 01 2019
Jul 31 2020

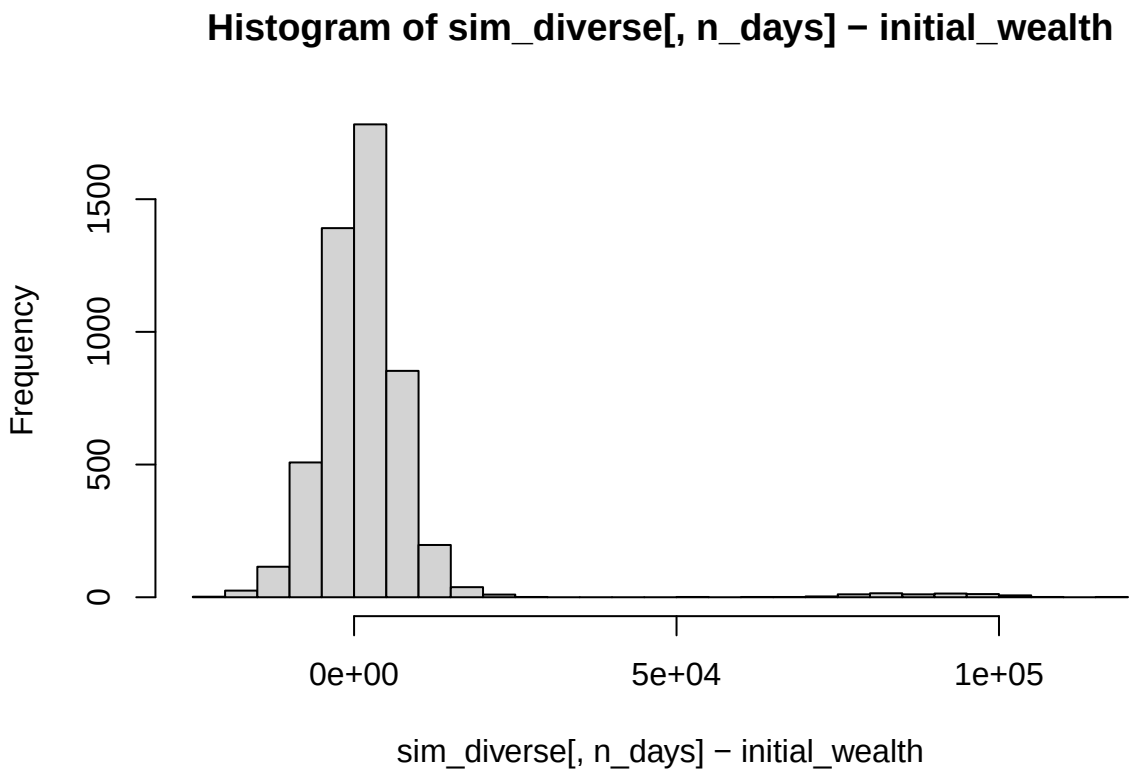



```
##               [,1]      [,2]     [,3]     [,4]     [,5]     [,6]      [,7]
## result.1 100004.84 101049.78 100637.9 100570.0 101341.4 100432.0  97565.52
## result.2 101562.19 102311.22 103275.7 100851.5 100626.8 102081.4 100272.51
## result.3  99609.23 101081.94 101274.8 103230.5 103679.1 103018.0 101913.37
## result.4 100681.49 108366.70 109507.5 109503.6 107604.9 107527.6 108582.77
## result.5 102479.53 103419.55 102799.8 102719.1 102284.4 101550.1 101080.28
## result.6  99527.58  99643.99 100412.3 101186.8 101719.6 103132.5 103664.25
##               [,8]      [,9]     [,10]     [,11]     [,12]     [,13]     [,14]
## result.1  97393.68  97202.48  99525.60  98359.04  99762.51  99779.74 100197.66
## result.2  87924.53  87795.74  87799.74  86923.50  86537.26  86309.42  86120.12
## result.3 101694.08 102151.53  98176.81  98108.82  98605.31  98533.29  98311.92
## result.4 108377.34 108794.57 109630.88 108350.00 108614.27 109208.95 108697.68
## result.5 101550.20 101386.73 100669.53 100167.18 100906.42  97993.00  98682.18
## result.6 104008.66 102465.60 102076.24 100526.24  99993.30 100049.05 100916.26
##              [,15]     [,16]     [,17]     [,18]     [,19]     [,20]
## result.1 100919.48 101431.66 101621.12 101769.95 101792.71 100622.00
## result.2  85500.42  85577.56  84797.96  88065.43  89693.44  89606.23
## result.3  97279.98  98296.50  98166.02  98119.74  97439.52  97558.73
## result.4 108111.49 108060.53 106437.21 108374.60 107962.63 107723.71
## result.5  96365.39  96391.67  96379.15  97305.22  97233.38  98983.44
## result.6 100798.03 101030.39 100960.19 100548.02 100275.64 100780.04
```




```
## [1] 102372.8
```

```
## [1] 2372.772
```



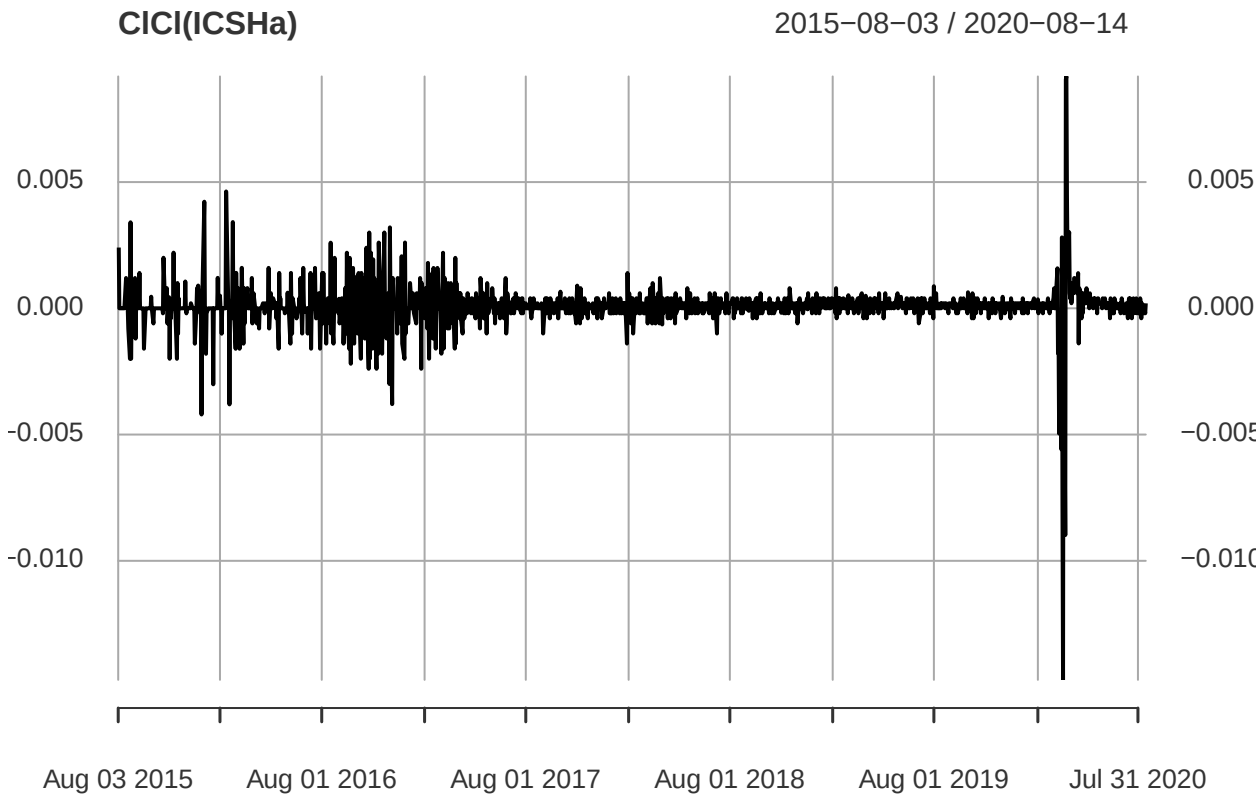

```
##        5%
## -8154.182
```

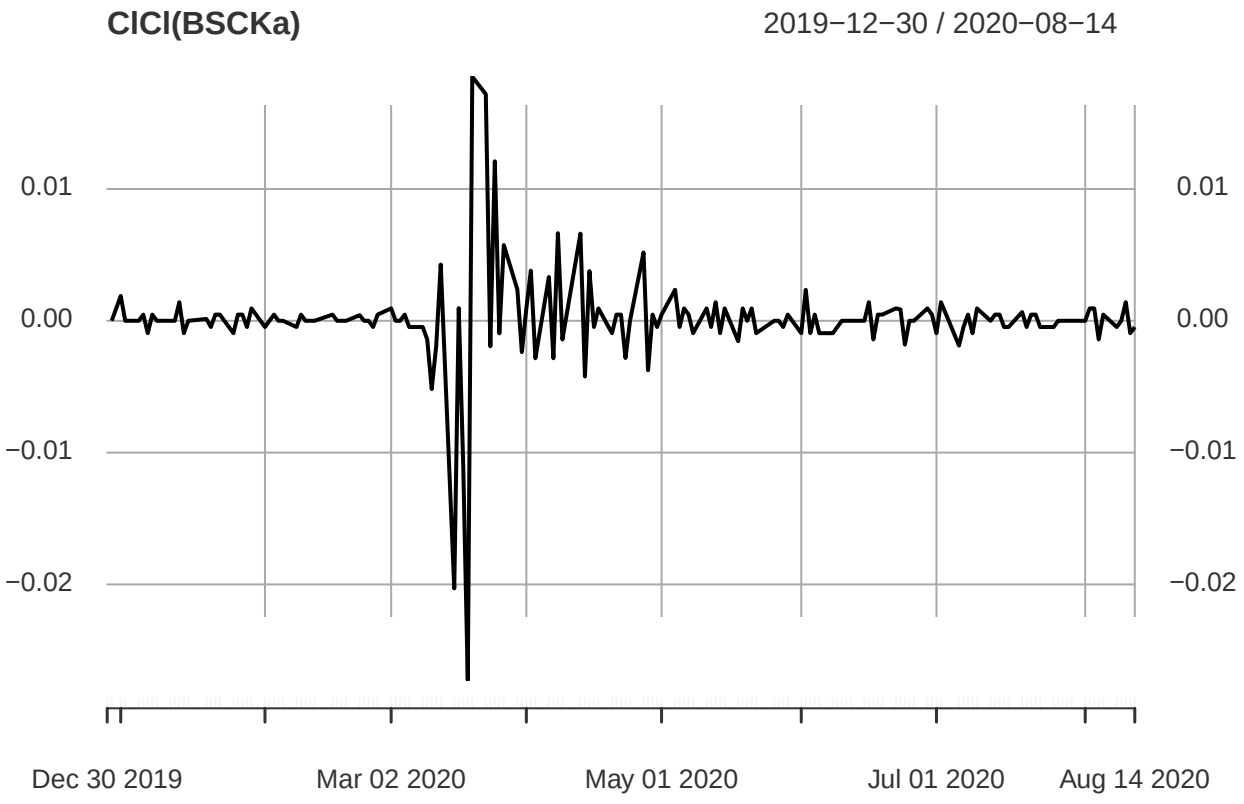
Secondly, we tried to construct our portfolio using all index funds, we can see from the CLCL plot that the three index funds did a good job diversifying systematic risk comparing to our aggressive portfolio. As a result, using bootstrap resampling to estimate the 4-week VAR, we got a VAR of $8129 at 5% level of the diverse portfolio.

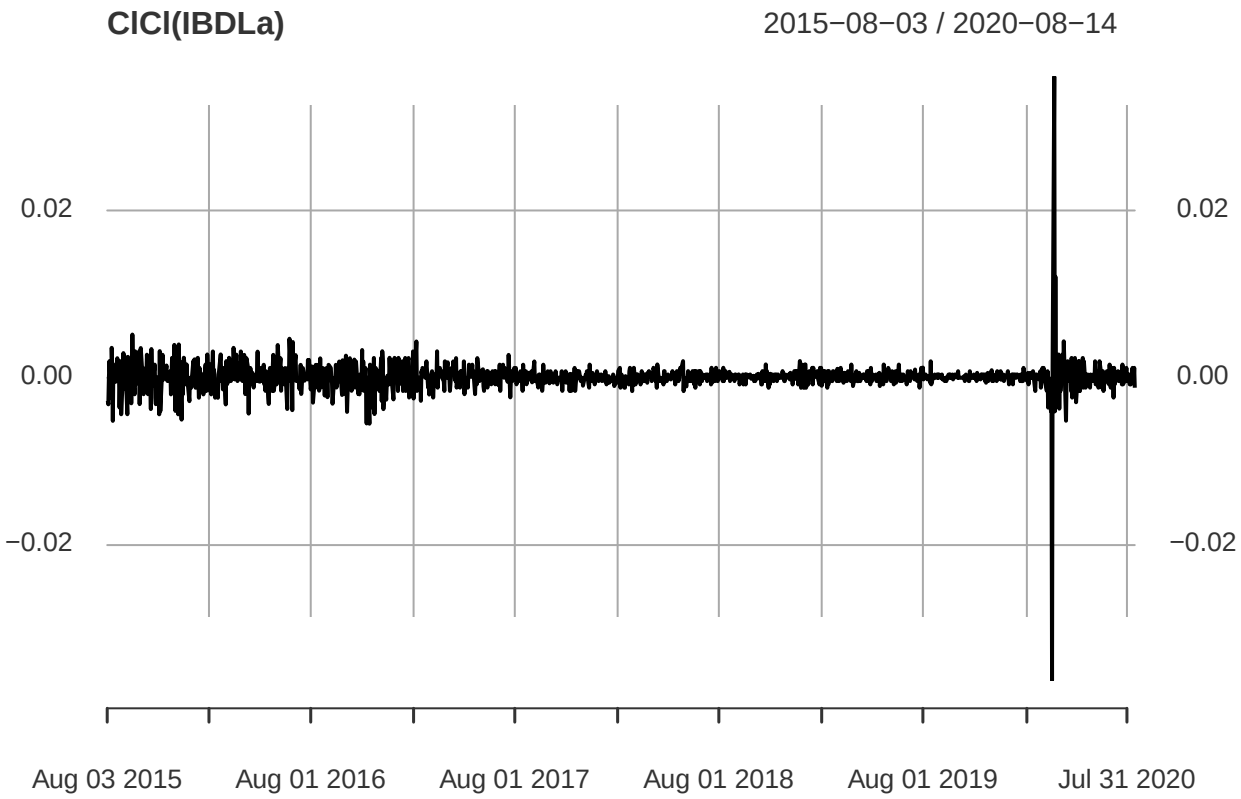
Safe Portfolio

```
## Warning in read.table(file = file, header = header, sep = sep,
## quote = quote, : incomplete final line found by readTableHeader
## on 'https://query2.finance.yahoo.com/v7/finance/download/ICSH?
## period1=-2208988800&period2=1597536000&interval=1d&events=split&crumb=nOzoZfrfynX'

## Warning in read.table(file = file, header = header, sep = sep,
## quote = quote, : incomplete final line found by readTableHeader
## on 'https://query2.finance.yahoo.com/v7/finance/download/ICSH?
## period1=-2208988800&period2=1597536000&interval=1d&events=split&crumb=nOzoZfrfynX'

## Warning in read.table(file = file, header = header, sep = sep,
## quote = quote, : incomplete final line found by readTableHeader
## on 'https://query2.finance.yahoo.com/v7/finance/download/BSCK?
## period1=-2208988800&period2=1597536000&interval=1d&events=split&crumb=nOzoZfrfynX'
```

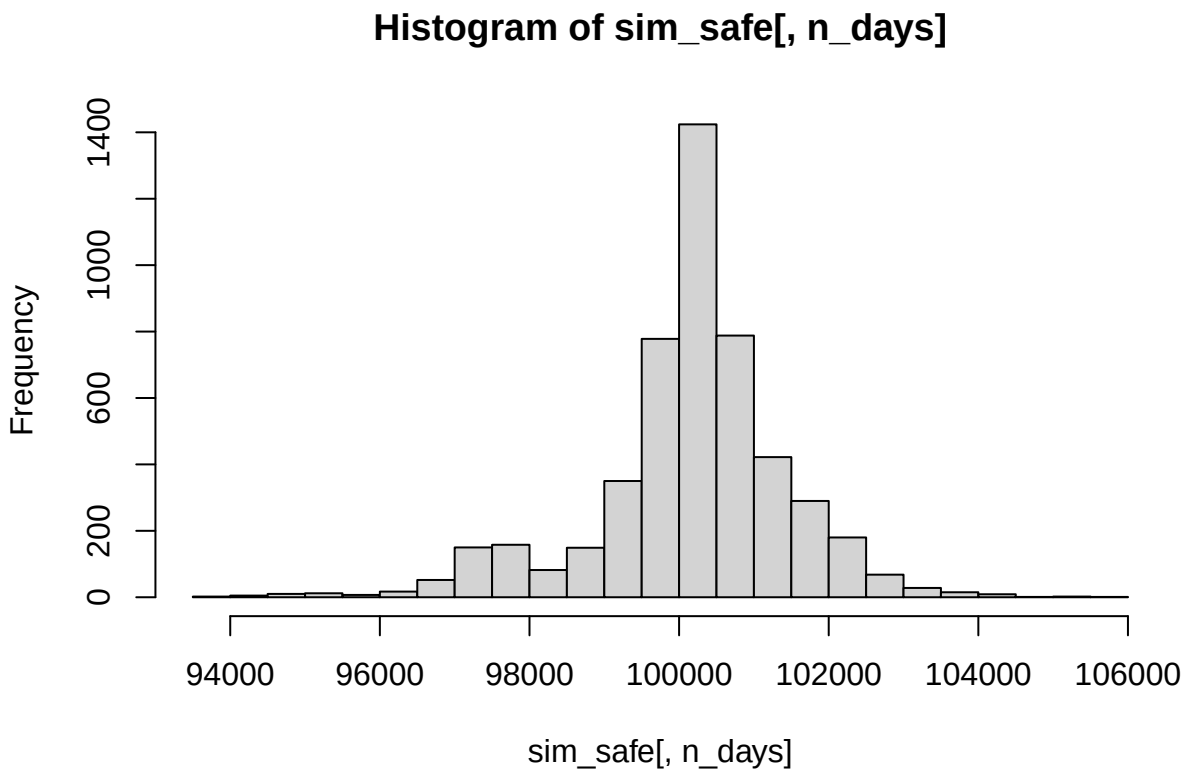
```
## Warning in read.table(file = file, header = header, sep = sep,
## quote = quote, : incomplete final line found by readTableHeader
## on 'https://query1.finance.yahoo.com/v7/finance/download/BSCK?
## period1=-2208988800&period2=1597536000&interval=1d&events=split&crumb=nOzoZfrfynX'

## Warning in read.table(file = file, header = header, sep = sep,
## quote = quote, : incomplete final line found by readTableHeader
## on 'https://query1.finance.yahoo.com/v7/finance/download/IBDL?
## period1=-2208988800&period2=1597536000&interval=1d&events=split&crumb=nOzoZfrfynX'

## Warning in read.table(file = file, header = header, sep = sep,
## quote = quote, : incomplete final line found by readTableHeader
## on 'https://query1.finance.yahoo.com/v7/finance/download/IBDL?
## period1=-2208988800&period2=1597536000&interval=1d&events=split&crumb=nOzoZfrfynX'
```




ClCl(BSCKa)
2019−12−30 / 2020−08−14
0.01
0.00
−0.01
−0.02
Dec 30 2019
Mar 02 2020
May 01 2020
Jul 01 2020
Aug 14 2020

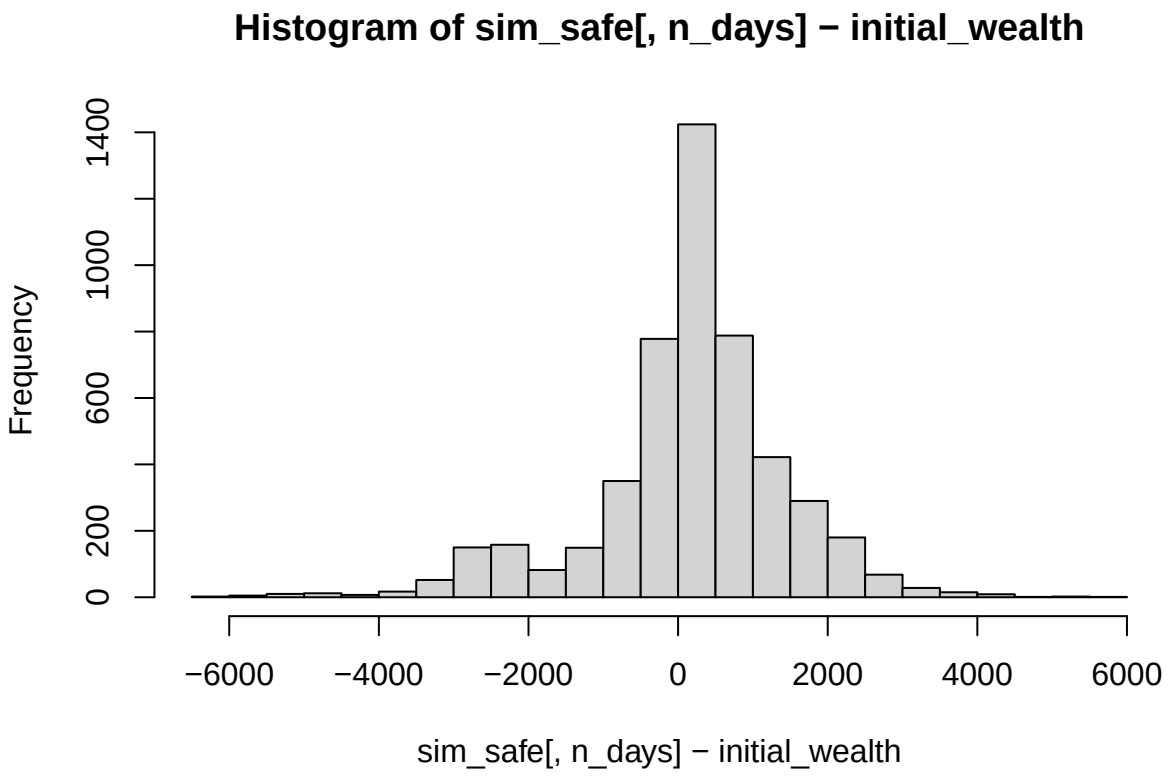



```
##               [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
## result.1 100031.64 100005.50 100154.58 100252.24 100206.40 100894.75 100900.72
## result.2  99930.48  99919.69  99963.69 100187.53 100225.16 100186.73 100180.31
## result.3  99998.29 100011.66  99985.96  99909.04  99911.67  99978.37 100009.90
## result.4 100033.44 100044.08 100134.83 100140.79 100146.75 100149.49 100193.71
## result.5 100225.55 100169.79 100231.39 100229.15 100282.98 100192.49 100212.40
## result.6  99982.39  99823.34  99787.54  99741.43  99785.07  99825.19  99814.33
##               [,8]      [,9]     [,10]     [,11]     [,12]     [,13]     [,14]
## result.1 100903.70 101054.09 101048.13 101018.85 100975.40 100883.98 100975.42
## result.2 100220.21 100176.80 100143.45 100118.37 100090.83 100076.77 100142.10
## result.3 100033.27 100027.35  99154.36  99191.78  99115.10  99173.96  99183.92
## result.4 100237.97 100183.99 100150.97 100007.55  99962.49  99970.06  99991.66
## result.5 100234.14 100249.89 100225.73 100219.68 100208.33 100272.96 100294.75
## result.6  99796.97  99863.66  99911.33  99933.03  99955.96  99958.72  99957.13
##              [,15]     [,16]     [,17]     [,18]     [,19]     [,20]
## result.1 100978.36 100922.19 100878.47 100905.77 100999.49 101041.76
## result.2  99431.65  99497.19  99544.89  99643.93  99691.57  99689.42
## result.3  99221.40  99144.65  99438.74  99446.63  99391.31  99385.35
## result.4 100039.45  99985.75  99842.83  99794.58  99769.38  99779.18
## result.5 100310.34 100256.76 100250.40 100224.94 100251.45 100344.22
## result.6 100004.36  99988.62  99992.34 100009.77 100049.44  99338.39
```




```
## [1] 100162.2
```

```
## [1] 162.2485
```




```
##        5%
## -2505.089
```

Secondly, we tried to construct our portfolio using all moneymarket/treasury funds, we can see from the CLCL plot that the three bond funds do not fluctuate a lot on a day to day basis comparing to the other two portfolios. As a result, using bootstrap resampling to estimate the 4-week VAR, we got a VAR of $2601 at 5% level of the safe portfolio.

Market Segementation

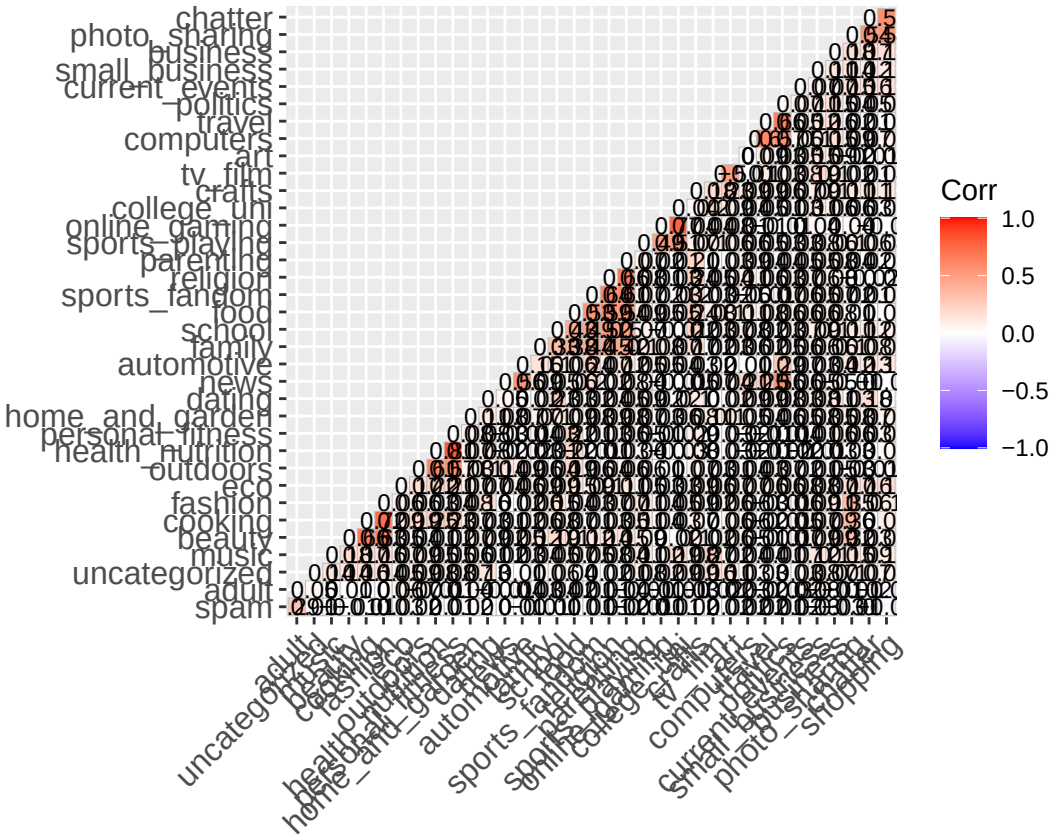
```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
## -- Attaching packages ---------------------------------------------------------------------- tidyverse
```

```
## v tibble  3.0.1     v purrr   0.3.4
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

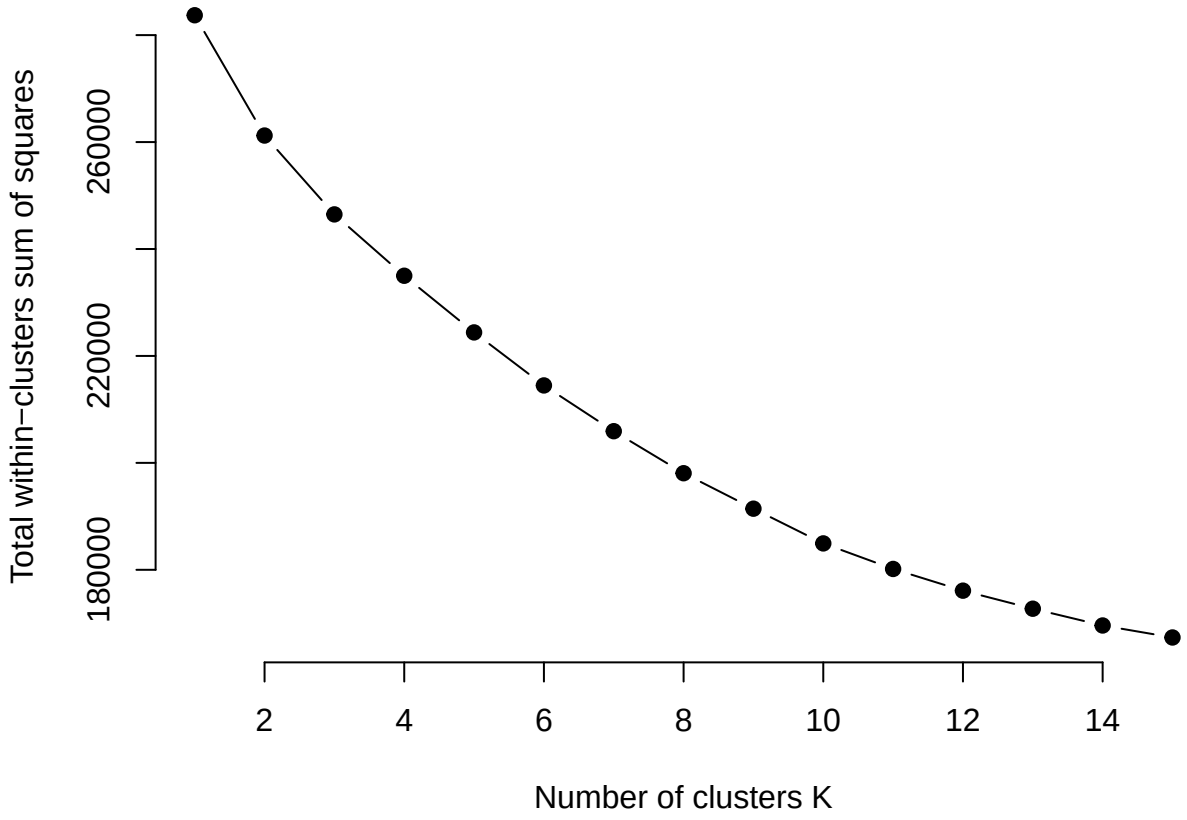
```
## -- Conflicts ------------------------------------------------------------------------ tidyverse_confli
## x purrr::accumulate()        masks foreach::accumulate()
## x mosaic::count()            masks dplyr::count()
```

```
## x purrr::cross()             masks mosaic::cross()
## x mosaic::do()               masks dplyr::do()
## x tidyr::expand()            masks Matrix::expand()
## x dplyr::filter()            masks stats::filter()
## x xts::first()               masks dplyr::first()
## x ggstance::geom_errorbarh() masks ggplot2::geom_errorbarh()
## x dplyr::lag()               masks stats::lag()
## x xts::last()                masks dplyr::last()
## x tidyr::pack()              masks Matrix::pack()
## x mosaic::stat()             masks ggplot2::stat()
## x mosaic::tally()            masks dplyr::tally()
## x tidyr::unpack()            masks Matrix::unpack()
## x purrr::when()              masks foreach::when()

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

##
## Attaching package: 'reshape2'

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



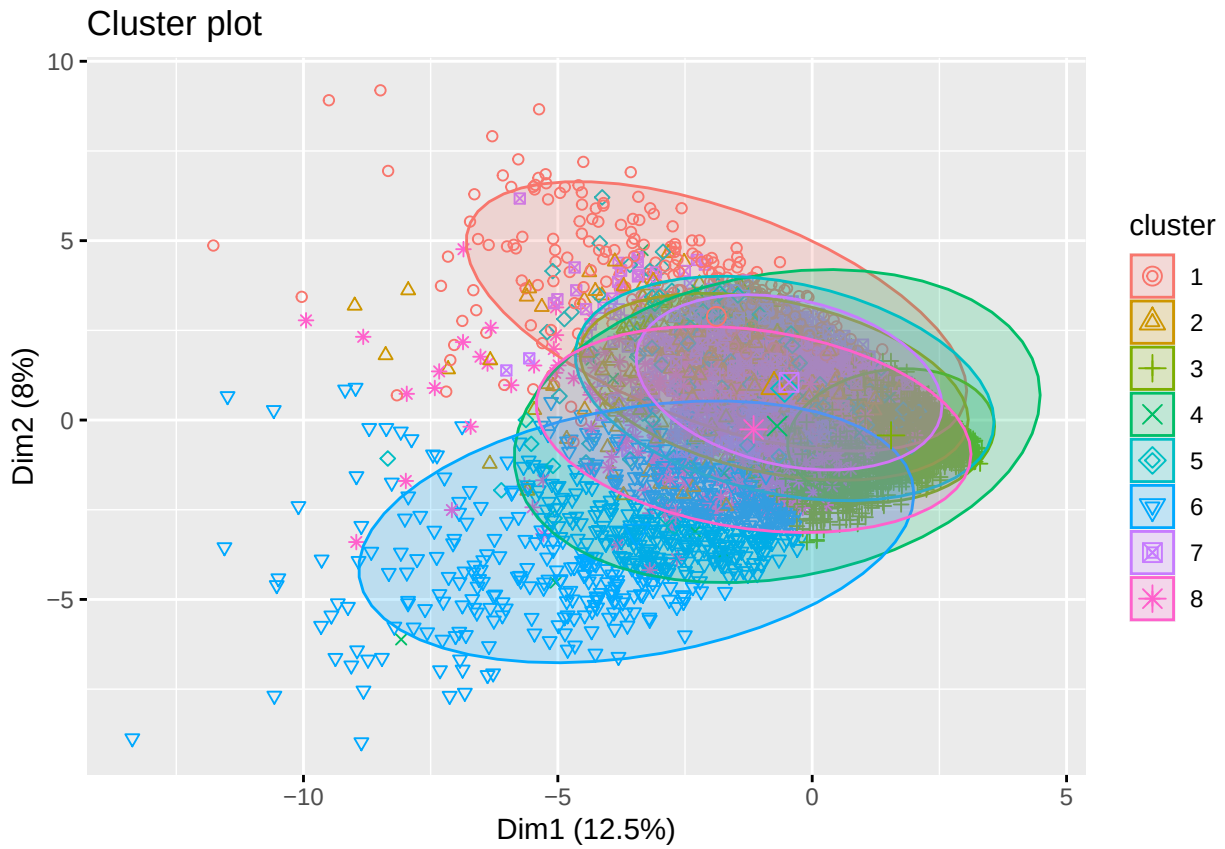
We first ran a correlation plot, from which we can see explore the correlations between the features. Our plot indicates that pairs in red such as Chatter/Shopping, photo_sharing/chatter, photo_sharing/shopping,

travel/politics, computers/politics,computers/travel, new/politics, online_gaming/college_uni,sport_playing/college_uni all have a somewhat strong positive correlation with each other. Furthermore we can a large proportion of the pairs are mingled together, thus this mingled situation leads us to consider that clustering could be a good direction to explore further.

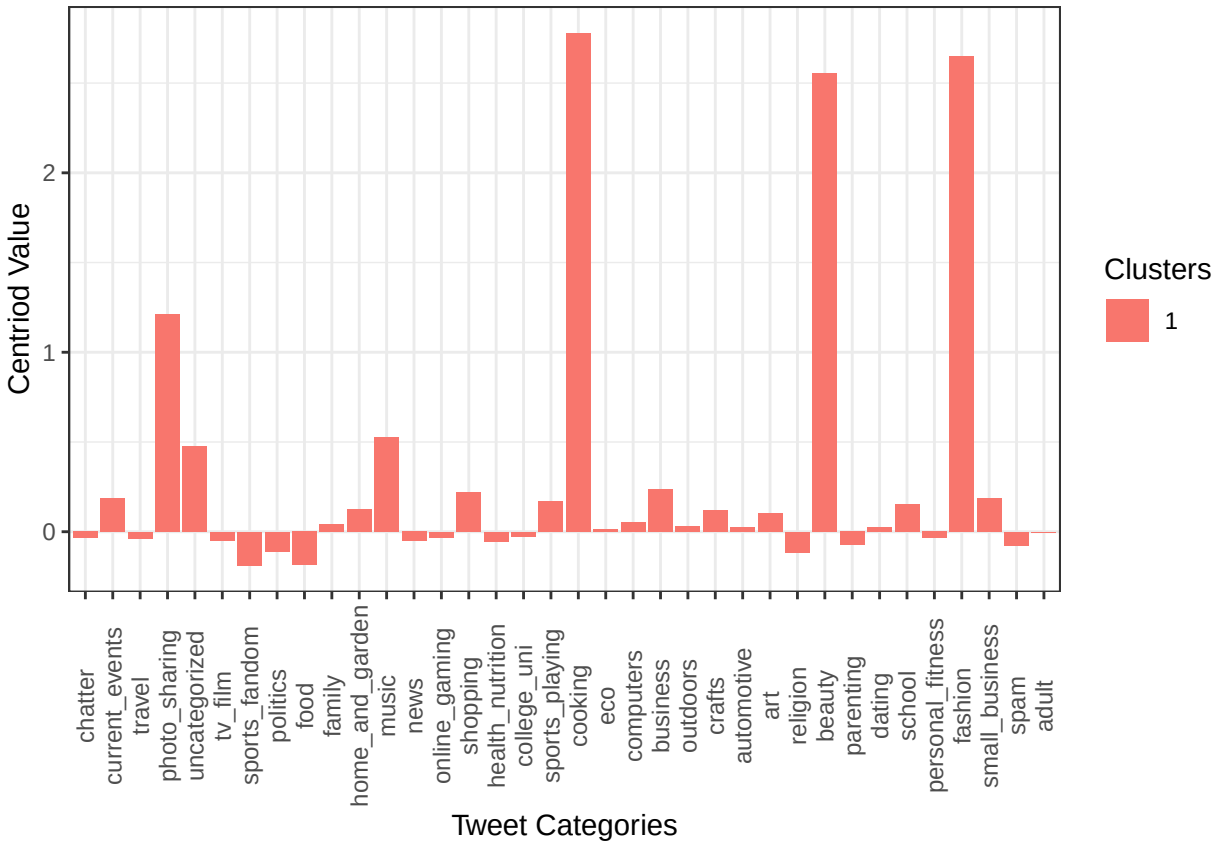


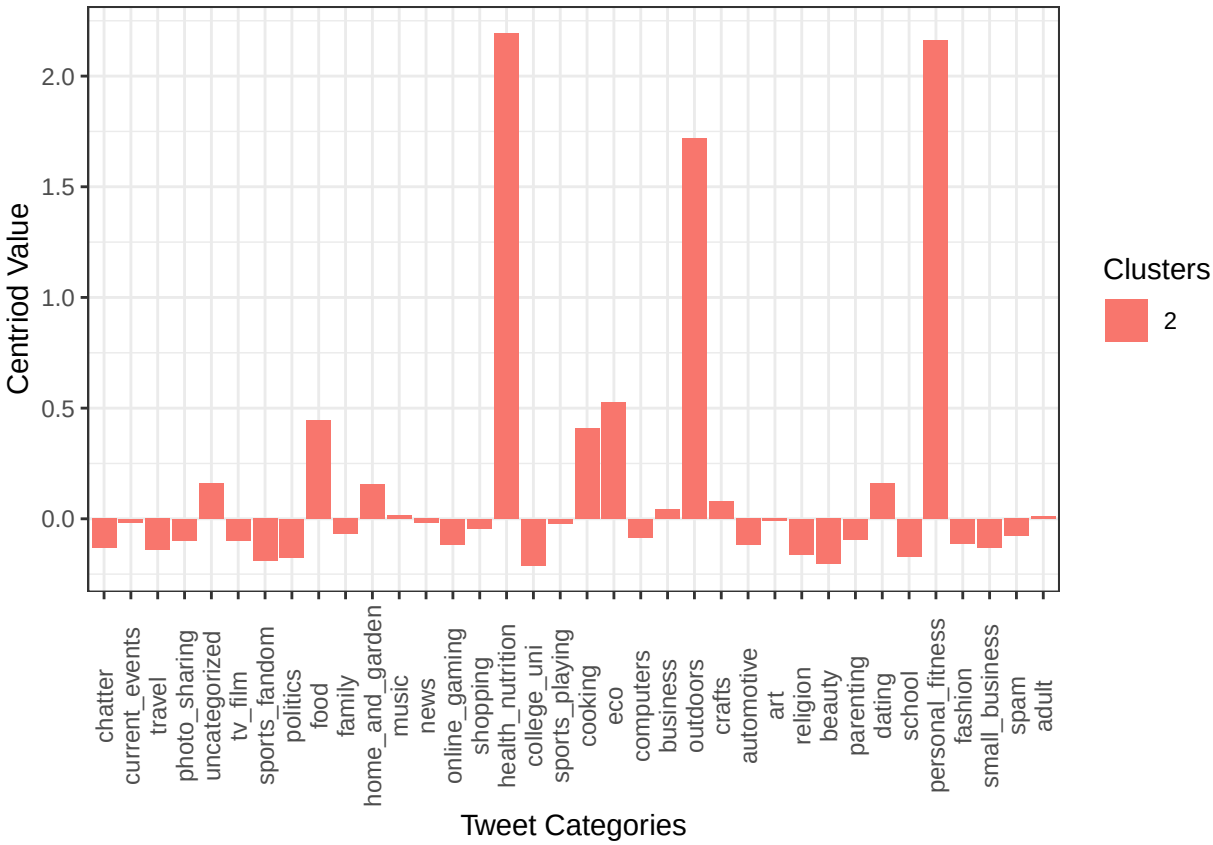

```
## Warning: argument frame is deprecated; please use ellipse instead.
```

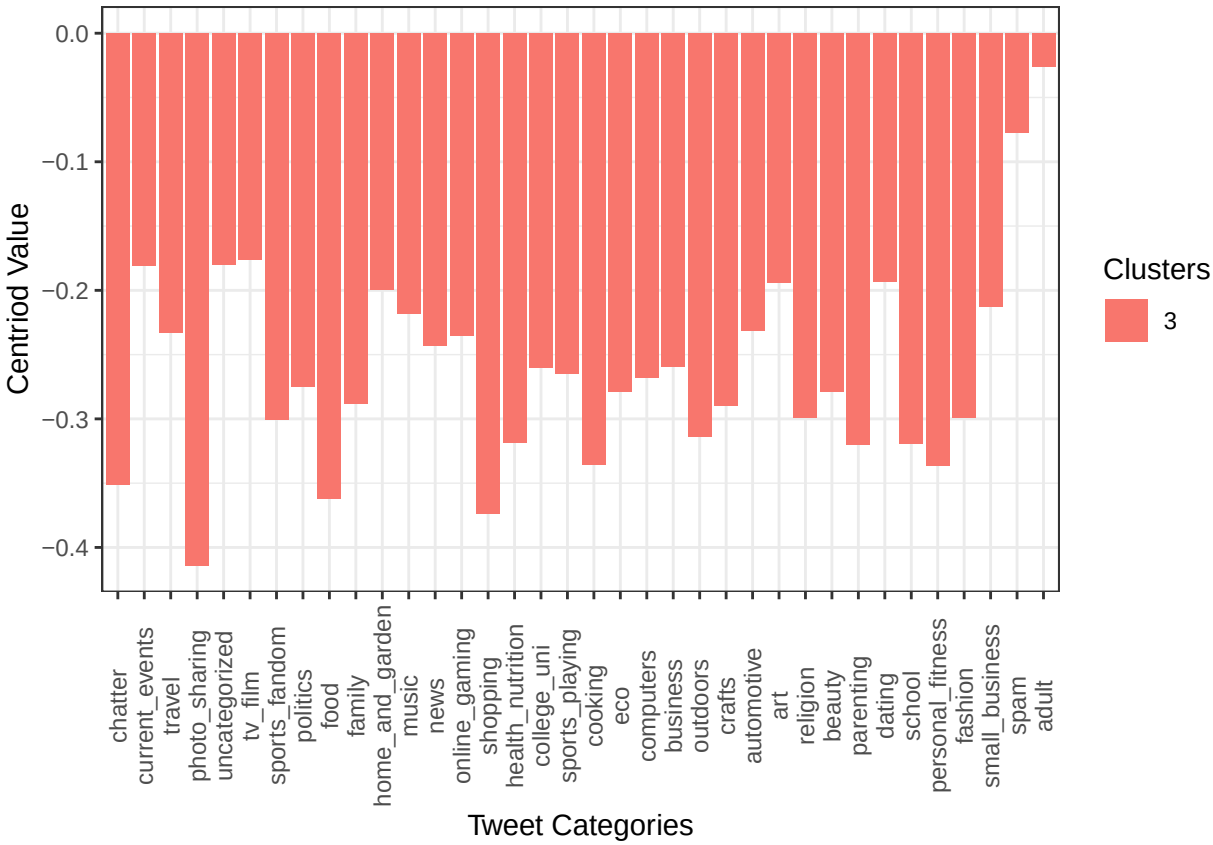
```
## Warning: argument frame.type is deprecated; please use ellipse.type instead.
```

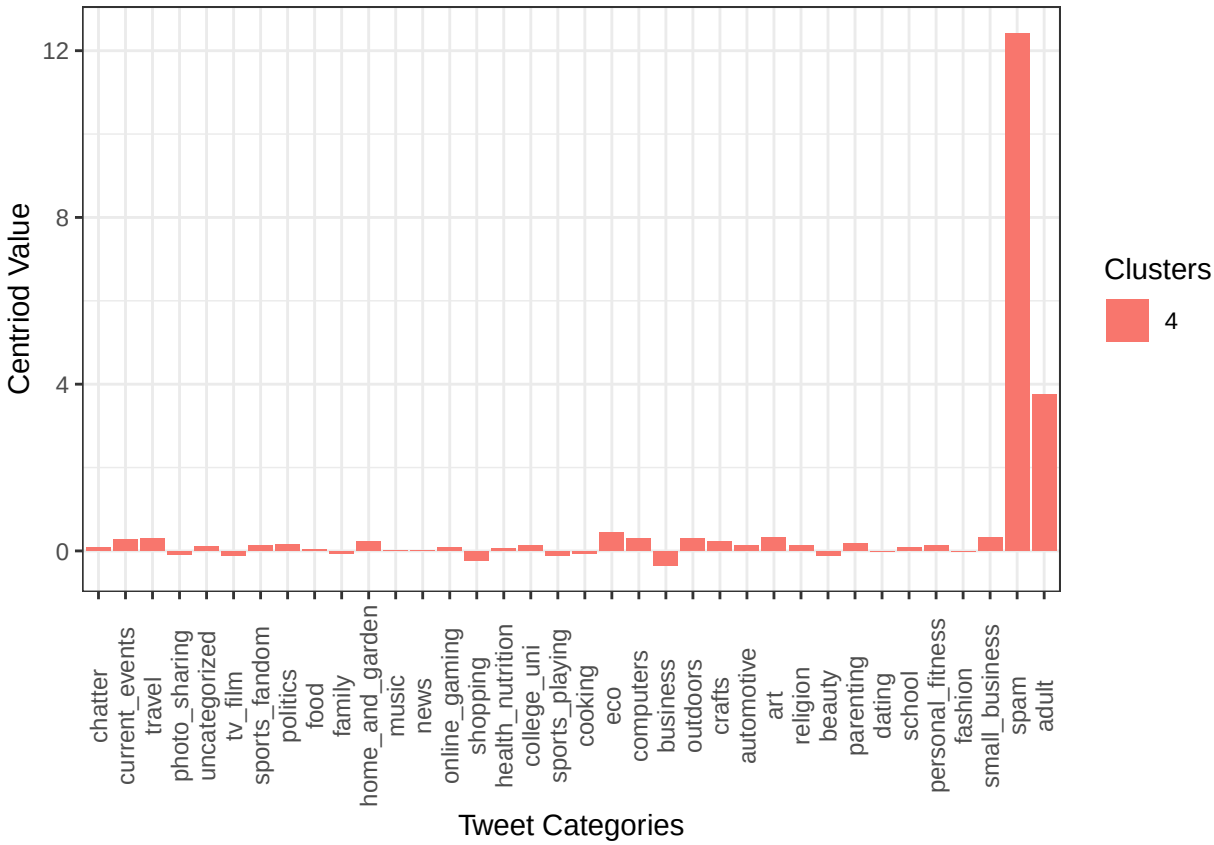


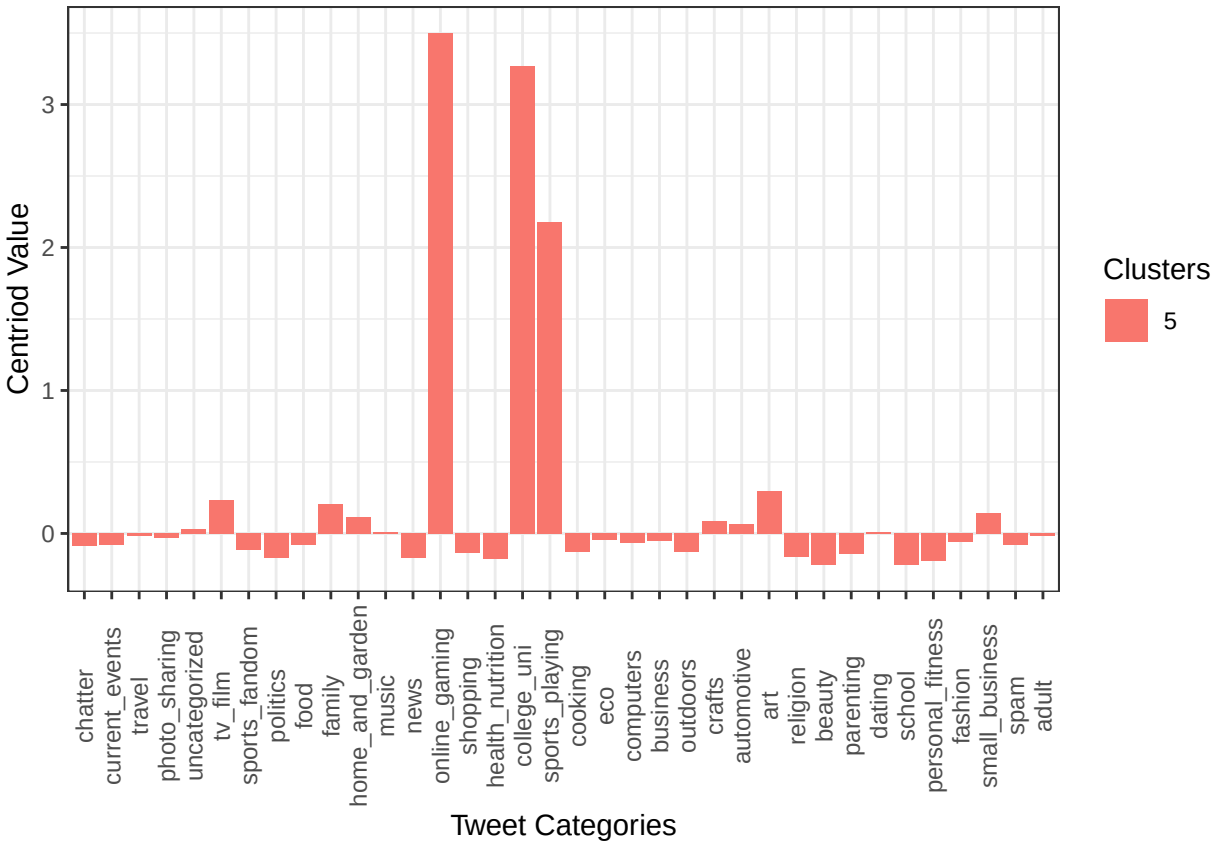
We choose 8 cluster for our model because we think it provides the best balance among the different number of clusters.

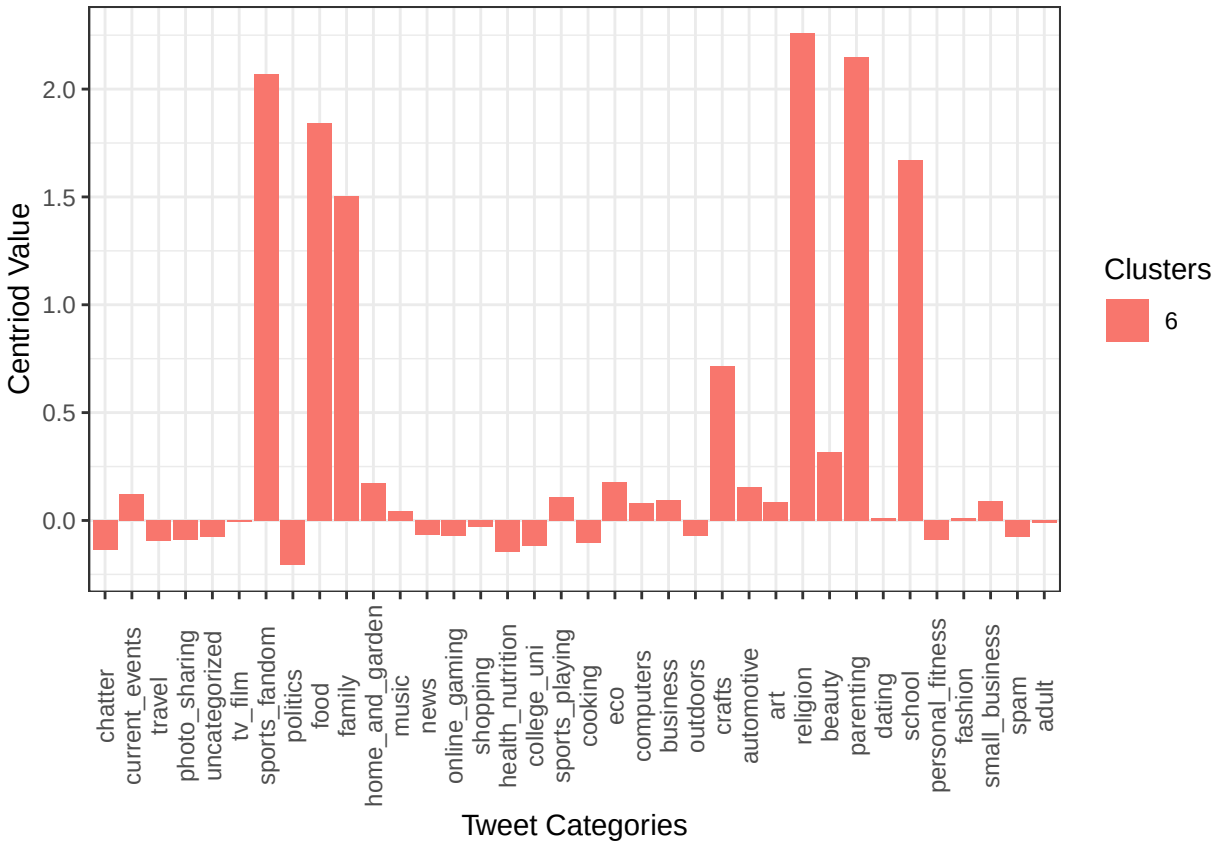

Centriod Value
2
1
0
chatter
current_events
travel
photo_sharing
uncategorized
tv_film
sports_fandom
politics
food
family
home_and_garden
music
news
online_gaming
shopping
health_nutrition
college_uni
sports_playing
cooking
eco
computers
business
outdoors
crafts
automotive
art
religion
beauty
parenting
dating
school
personal_fitness
fashion
small_business
spam
adult
Tweet Categories
Clusters
1

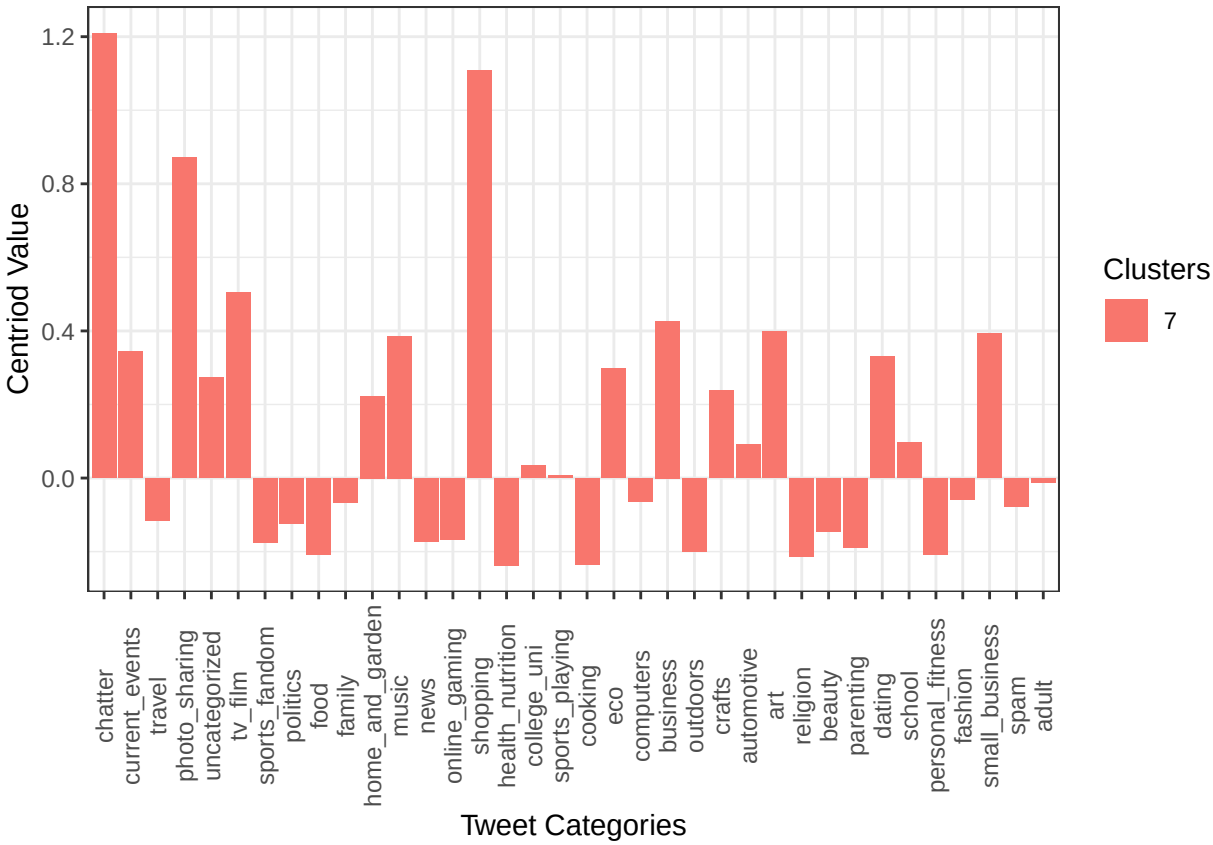

Centriod Value
2.0
1.5
1.0
0.5
0.0
chatter
current_events
travel
photo_sharing
uncategorized
tv_film
sports_fandom
politics
food
family
home_and_garden
music
news
online_gaming
shopping
health_nutrition
college_uni
sports_playing
cooking
eco
computers
business
outdoors
crafts
automotive
art
religion
beauty
parenting
dating
school
personal_fitness
fashion
small_business
spam
adult
Tweet Categories
Clusters
2

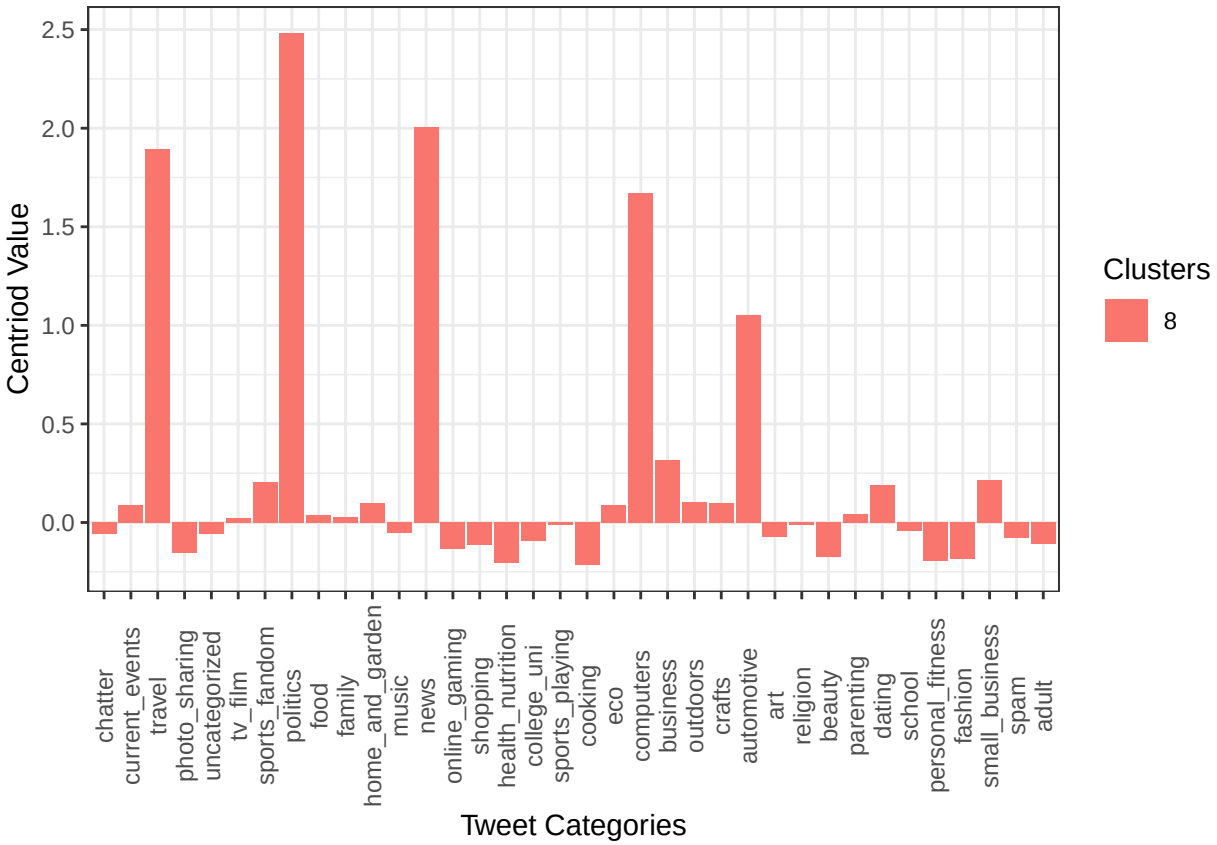

0.0
−0.1
−0.2
−0.3
−0.4
Centriod Value
chatter
current_events
travel
photo_sharing
uncategorized
tv_film
sports_fandom
politics
food
family
home_and_garden
music
news
online_gaming
shopping
health_nutrition
college_uni
sports_playing
cooking
eco
computers
business
outdoors
crafts
automotive
art
religion
beauty
parenting
dating
school
personal_fitness
fashion
small_business
spam
adult
Tweet Categories
Clusters
3


Centriod Value
12
8
4
0
chatter
current_events
travel
photo_sharing
uncategorized
tv_film
sports_fandom
politics
food
family
home_and_garden
music
news
online_gaming
shopping
health_nutrition
college_uni
sports_playing
cooking
eco
computers
business
outdoors
crafts
automotive
art
religion
beauty
parenting
dating
school
personal_fitness
fashion
small_business
spam
adult
Tweet Categories
Clusters
4


Centriod Value
3
2
1
0
chatter
current_events
travel
photo_sharing
uncategorized
tv_film
sports_fandom
politics
food
family
home_and_garden
music
news
online_gaming
shopping
health_nutrition
college_uni
sports_playing
cooking
eco
computers
business
outdoors
crafts
automotive
art
religion
beauty
parenting
dating
school
personal_fitness
fashion
small_business
spam
adult
Tweet Categories
Clusters
5


Centriod Value
2.0
1.5
1.0
0.5
0.0
chatter
current_events
travel
photo_sharing
uncategorized
tv_film
sports_fandom
politics
food
family
home_and_garden
music
news
online_gaming
shopping
health_nutrition
college_uni
sports_playing
cooking
eco
computers
business
outdoors
crafts
automotive
art
religion
beauty
parenting
dating
school
personal_fitness
fashion
small_business
spam
adult
Tweet Categories
Clusters
6


Centriod Value
1.2
0.8
0.4
0.0
chatter
current_events
travel
photo_sharing
uncategorized
tv_film
sports_fandom
politics
food
family
home_and_garden
music
news
online_gaming
shopping
health_nutrition
college_uni
sports_playing
cooking
eco
computers
business
outdoors
crafts
automotive
art
religion
beauty
parenting
dating
school
personal_fitness
fashion
small_business
spam
adult
Tweet Categories
Clusters
7



Finally we can see that our model filters out Spam/adult as a single cluster and shows that 1.photo_sharing/cooking/beauty/fashion, 2.health_nutrition/outdoors/personal_fitness/food, 3.online_gaming/college_uni/sports_playing, 4.sports_fandom/food/family/crafts/religion/parenting/school, 5.photo_sharing/tv_film/shopping, 6.travel/politics/news/computers/automotive as clusters. Since chatter and uncategorized does not have informative meaning for us, we did not consider those 2 features into account when writing this report. NutrientH20 could advertise to the user within the same cluster according to its specific product.

Data cleaning and framework

```
#Read all folders in C50 train
train=Sys.glob('D:/Austin/Summer Semester/Predictive Analytics/data/ReutersC50/C50train/*')
```

```
# Read the test
test=Sys.glob('D:/Austin/Summer Semester/Predictive Analytics/data/ReutersC50/C50test/*')
```

```
#Create training dataset
lab=NULL
comb_artist=NULL

# Read in the name in the train folder using for loop
for (name in train)
{
  author=substring(name,first=50)#first= ; ensure less than string length
  article=Sys.glob(paste0(name,'/*.txt'))
  comb_artist=append(comb_artist,article)
  lab=append(lab,rep(author,length(article)))
}
```

```
#Cleaning the file names
readerPlain <- function(fname)
  {
                readPlain(elem=list(content=readLines(fname)),
                            id=fname, language='en')
  }
comb = lapply(comb_artist, readerPlain)
names(comb) = comb_artist
names(comb) = sub('.txt', '', names(comb))
```

```
#Create a text mining corpus
corp_train=Corpus(VectorSource(comb))
```

```
# sparse item
dtm_tr=removeSparseTerms(dtm_train,.99)
tf_idf_mat = weightTfIdf(dtm_tr)
dtm_trr<-as.matrix(tf_idf_mat) #Matrix
tf_idf_mat #3394 words, 2500 documents
```

```
## <<DocumentTermMatrix (documents: 2500, terms: 3395)>>
## Non-/sparse entries: 382971/8104529
## Sparsity           : 95%
## Maximal term length: 53
## Weighting          : term frequency - inverse document frequency (normalized) (tf-idf)
```

```
comb_artist1=NULL
lab1=NULL
for (n in test)
{
```

```
  author1=substring(n,first=50)#first= ; ensure less than string length
  article1=Sys.glob(paste0(n,'/*.txt'))
  comb_artist1=append(comb_artist1,article1)
  lab1=append(lab1,rep(author1,length(article1)))
}
```

```
#Make sure the name is cleaned
comb1 = lapply(comb_artist1, readerPlain)
names(comb1) = comb_artist1
names(comb1) = sub('.txt', '', names(comb1))
```

```
#Create a text mining corpus
corp_ts=Corpus(VectorSource(comb1))
```

Data tokenization and pre-processing

```
## <<DocumentTermMatrix (documents: 2500, terms: 3395)>>
## Non-/sparse entries: 379314/8108186
## Sparsity           : 96%
## Maximal term length: 53
## Weighting          : term frequency - inverse document frequency (normalized) (tf-idf)
```

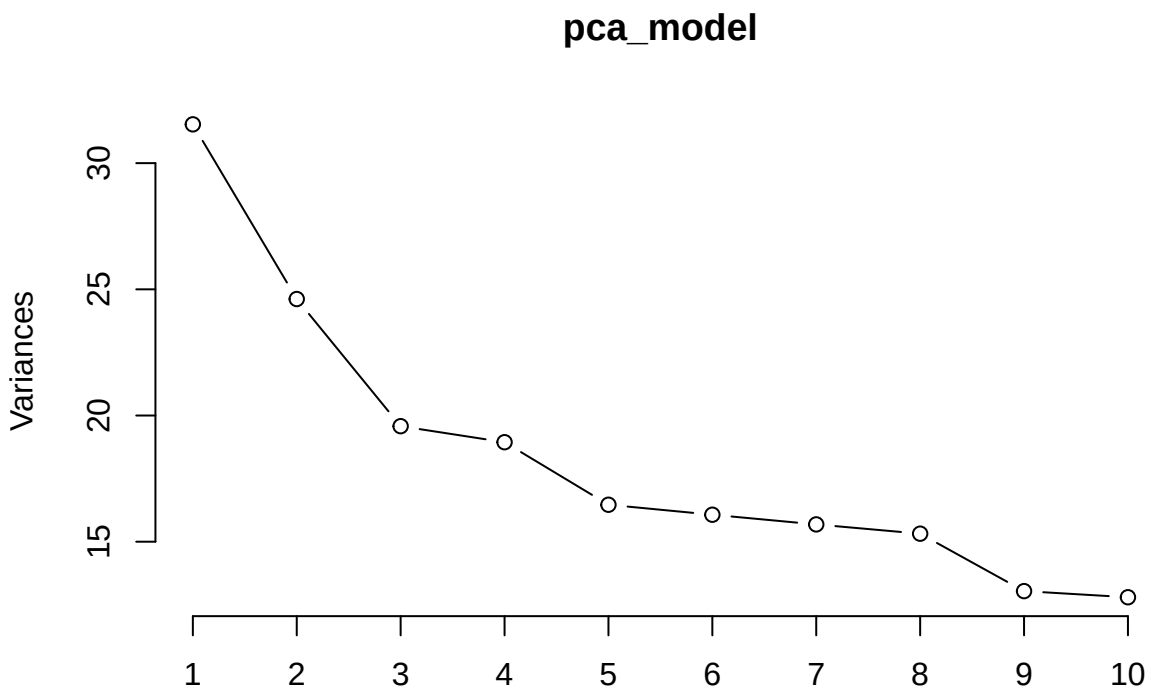
Dimension reduction

```
dtm_trr_one<-dtm_trr[,which(colSums(dtm_trr) != 0)]
dtm_tss_one<-dtm_tss[,which(colSums(dtm_tss) != 0)]
```

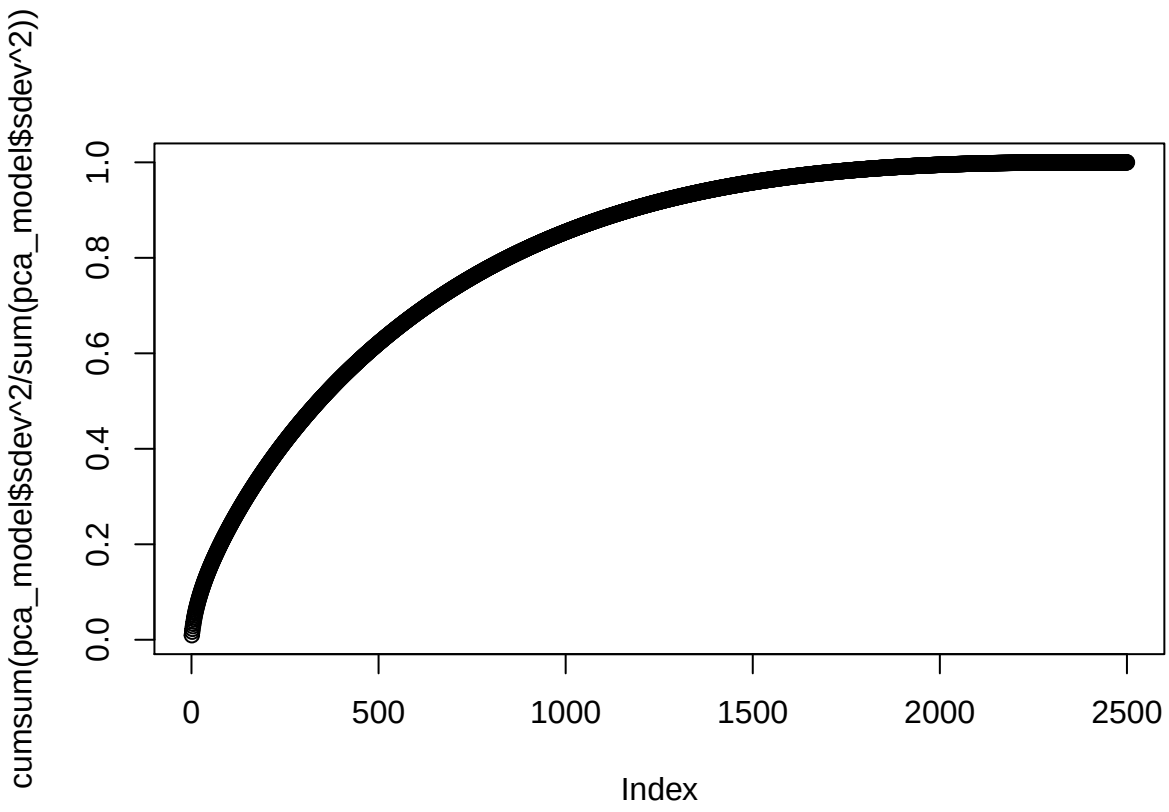
```
dtm_tss_one = dtm_tss_one[,intersect(colnames(dtm_tss_one),colnames(dtm_trr_one))]
dtm_trr_one = dtm_trr_one[,intersect(colnames(dtm_tss_one),colnames(dtm_trr_one))]
# total of 8317500 words
```

```
# use pca for dimension red
pca_model = prcomp(dtm_trr_one,scale=TRUE)
pca_predict=predict(pca_model,newdata = dtm_tss_one)
```

```
plot(pca_model,type='line')
```




```
var <- apply(pca_model$x, 2, var)
prop <- var / sum(var)
plot(cumsum(pca_model$sdev^2/sum(pca_model$sdev^2)))
```




```
#From the PCA result, the model has explained more than 75% of the variance at PC730 and more than 80%
```

```
# take only first 730 rows
class_train = data.frame(pca_model$x[,1:730])
class_train['author']=lab
tr_load = pca_model$rotation[,1:730]
ts_class_predict <- scale(dtm_tss_one) %*% tr_load
ts_class <- as.data.frame(ts_class_predict)
ts_class['author']=lab1
#ts_class['author']
```

#Use classification methods to find author of articles ##Naive Bayes

```
library('e1071')
library(caret)
naive_model=naiveBayes(as.factor(author)~.,data=class_train)
naive_prediction=predict(naive_model,ts_class)

# classify as factor and change into binary data
predicted_nb=naive_prediction
real_nb=as.factor(ts_class$author)
temp_nb<-as.data.frame(cbind(real_nb,predicted_nb))
temp_nb$dummy=ifelse(temp_nb$real_nb==temp_nb$predicted_nb,1,0)
sum(temp_nb$dummy)
```

```
## [1] 802
```

```
sum(temp_nb$dummy)*100/nrow(temp_nb)
```

```
## [1] 32.08
```

```
# The accuracy is 32.08%
```

##Random Forest

```
rc_predict<-predict(rc_model,data=ts_class)
rc_table<-as.data.frame(table(rc_predict,as.factor(ts_class$author)))
pred<-rc_predict
real<-as.factor(ts_class$author)
temp<-as.data.frame(cbind(real,pred))
temp$flag<-ifelse(temp$real==temp$pred,1,0)
sum(temp$flag)
```

```
## [1] 1877
```

```
sum(temp$flag)*100/nrow(temp)
```

```
## [1] 75.08
```

```
# Accuracy rate of Random Forest is 75% from 1875 correct predictions.
```

K-Nearest Neighbors

```
train_author=as.factor(class_train$author)
test_author=as.factor(ts_class$author)
train_knn = subset(class_train, select = -c(author))
test_knn = subset(ts_class,select=-c(author))
```

```
library(class)
set.seed(1)
knn_pred=knn(train_knn,test_knn,train_author,k=1)
```

```
temp_knn=as.data.frame(cbind(knn_pred,test_author))
temp_dummy<-ifelse(as.integer(knn_pred)==as.integer(test_author),1,0)
sum(temp_dummy)
```

```
## [1] 846
```

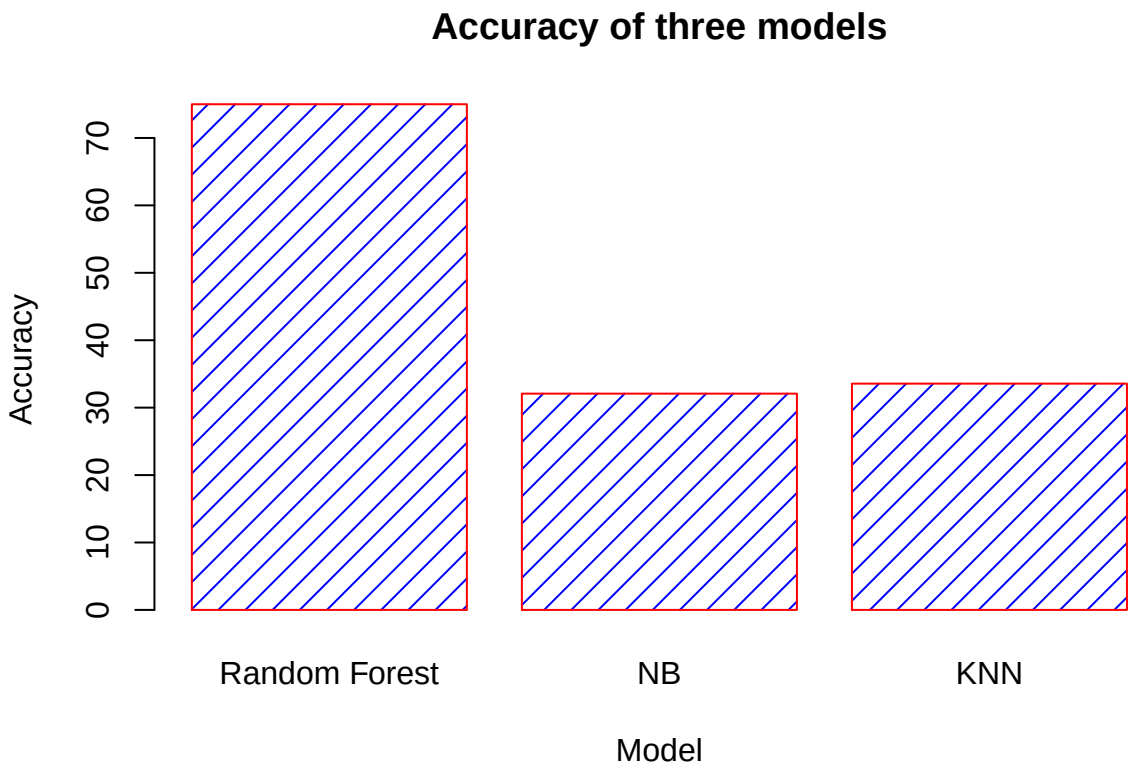
```
sum(temp_dummy)*100/nrow(temp_knn)
```

```
## [1] 33.84
```

```
#KNN presents an accuracy of 33.56% from 839 accuracy predictions.
```

Graphical summary

```
barplot(c(75,32.08,33.56),
main="Accuracy of three models",
xlab="Model",
ylab="Accuracy",
names.arg = c("Random Forest","NB","KNN"),
border="red",
col="blue",
density=10
)
```



Summary

In this test I used three supervised learning prediction methods (Random Forest, Naive Bayes and KNN) to attribute the author of the Reuters articles based on the contents. As shown in the graphical summary, Random Forest model has much higher predictability of 75% than the other two models.

Proj2_Question6

Question 6

load in packages

Presenting the structure of the raw dataset:

Each row will be a transaction class object and the apriori algorithm will be applied

```
##  chr [1:9835] "citrus fruit,semi-finished bread,margarine,ready soups" ...
```

```
##    Length     Class      Mode
##      9835 character character
```

From the dataset summary: 1. There are altogether 9835 transactions revealed. 2. More than 50% of baskets purchased no more than 3 items. 3. The most item in a basket of transaction is 32. 4. 2513 baskets have whole milk purchased and it is the most bought item. 5. Other items with top popularity are "other vegetables", "rolls/buns" and "soda". 6. The barplot shows the rest.

Try out different support and confidence levels with min length=2.0 and see the association rule with the apriori algorithm.

support=0.05, confidence=0.1

```
asso_rules1 = apriori(trans_grocery,
                      parameter=list(support=0.05, confidence=0.1, minlen=2.0))
```

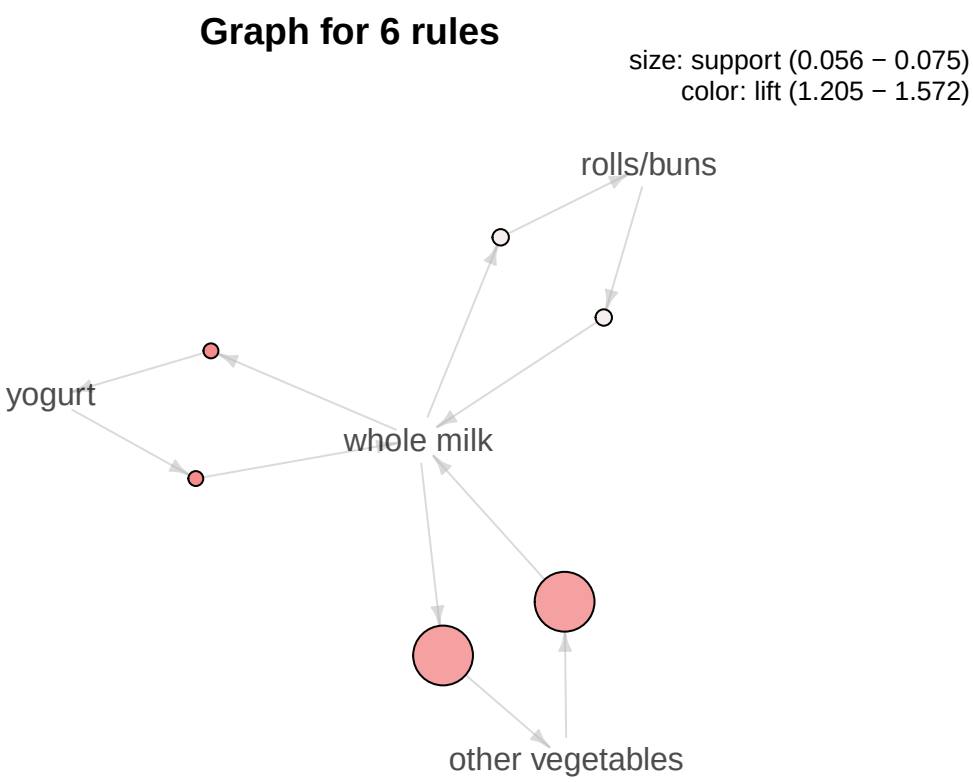
```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.1    0.1    1 none FALSE            TRUE       5    0.05      2
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
```

```
##
## Absolute minimum support count: 491
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [28 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 done [0.00s].
## writing ... [6 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
arules::inspect(asso_rules1)
```

```
##     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
```

```
plot(asso_rules1, method='graph')
```



Setting support=0.05 and confidence=0.1, we see only six rules. We also see the relationship between whole milk, other vegetables, yogurt and rolls/buns. This corresponds to the barplot we saw before. But we would like to dig deeper by changing support and lift.

support=0.03, confidence=0.08

```
asso_rules2 = apriori(trans_grocery,
                      parameter=list(support=0.03, confidence=0.08, minlen=1.8))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.08    0.1    1 none FALSE            TRUE       5    0.03      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 295
##
## set item appearances ...[0 item(s)] done [0.00s].
```

```
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [44 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [51 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

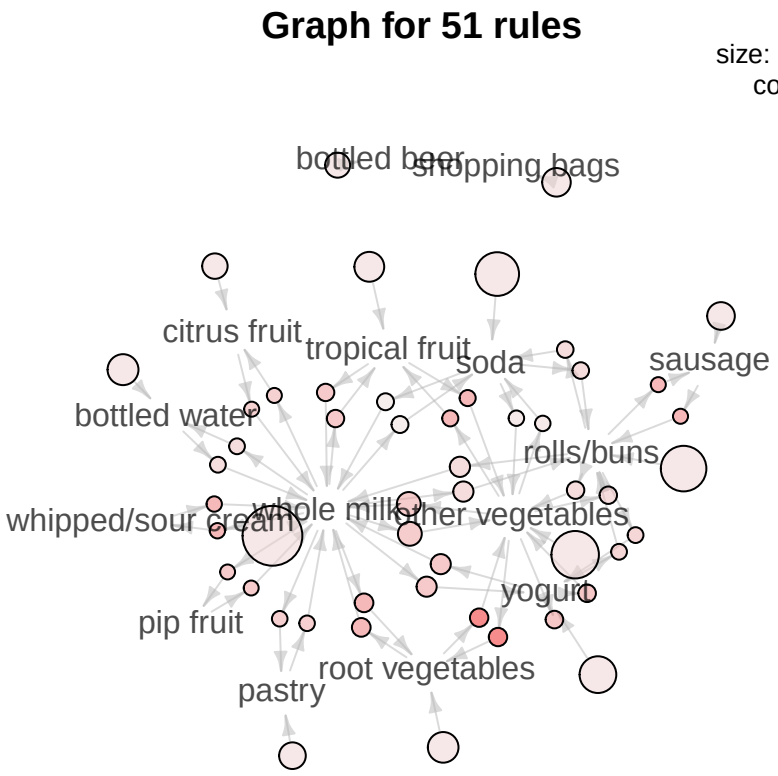
```
arules::inspect(asso_rules2)
```

```
##      lhs                     rhs                    support    confidence
## [1]  {}                   => {bottled beer}        0.08052872 0.08052872
## [2]  {}                   => {pastry}              0.08896797 0.08896797
## [3]  {}                   => {citrus fruit}        0.08276563 0.08276563
## [4]  {}                   => {shopping bags}       0.09852567 0.09852567
## [5]  {}                   => {sausage}             0.09395018 0.09395018
## [6]  {}                   => {bottled water}       0.11052364 0.11052364
## [7]  {}                   => {tropical fruit}      0.10493137 0.10493137
## [8]  {}                   => {root vegetables}     0.10899847 0.10899847
## [9]  {}                   => {soda}                0.17437722 0.17437722
## [10] {}                   => {yogurt}              0.13950178 0.13950178
## [11] {}                   => {rolls/buns}          0.18393493 0.18393493
## [12] {}                   => {other vegetables}    0.19349263 0.19349263
## [13] {}                   => {whole milk}          0.25551601 0.25551601
## [14] {whipped/sour cream} => {whole milk}          0.03223183 0.44964539
## [15] {whole milk}         => {whipped/sour cream}  0.03223183 0.12614405
## [16] {pip fruit}          => {whole milk}          0.03009659 0.39784946
## [17] {whole milk}         => {pip fruit}           0.03009659 0.11778750
## [18] {pastry}             => {whole milk}          0.03324860 0.37371429
## [19] {whole milk}         => {pastry}              0.03324860 0.13012336
## [20] {citrus fruit}       => {whole milk}          0.03050330 0.36855037
## [21] {whole milk}         => {citrus fruit}        0.03050330 0.11937923
## [22] {sausage}            => {rolls/buns}          0.03060498 0.32575758
## [23] {rolls/buns}         => {sausage}             0.03060498 0.16639027
## [24] {bottled water}      => {whole milk}          0.03436706 0.31094756
## [25] {whole milk}         => {bottled water}       0.03436706 0.13450060
## [26] {tropical fruit}     => {other vegetables}    0.03589222 0.34205426
## [27] {other vegetables}   => {tropical fruit}      0.03589222 0.18549658
## [28] {tropical fruit}     => {whole milk}          0.04229792 0.40310078
## [29] {whole milk}         => {tropical fruit}      0.04229792 0.16553920
## [30] {root vegetables}    => {other vegetables}    0.04738180 0.43470149
## [31] {other vegetables}   => {root vegetables}     0.04738180 0.24487651
## [32] {root vegetables}    => {whole milk}          0.04890696 0.44869403
## [33] {whole milk}         => {root vegetables}     0.04890696 0.19140470
## [34] {soda}               => {rolls/buns}          0.03833249 0.21982507
## [35] {rolls/buns}         => {soda}                0.03833249 0.20840243
## [36] {soda}               => {other vegetables}    0.03274021 0.18775510
## [37] {other vegetables}   => {soda}                0.03274021 0.16920652
## [38] {soda}               => {whole milk}          0.04006101 0.22973761
## [39] {whole milk}         => {soda}                0.04006101 0.15678472
## [40] {yogurt}             => {rolls/buns}          0.03436706 0.24635569
## [41] {rolls/buns}         => {yogurt}              0.03436706 0.18684356
## [42] {yogurt}             => {other vegetables}    0.04341637 0.31122449
## [43] {other vegetables}   => {yogurt}              0.04341637 0.22438255
## [44] {yogurt}             => {whole milk}          0.05602440 0.40160350
```

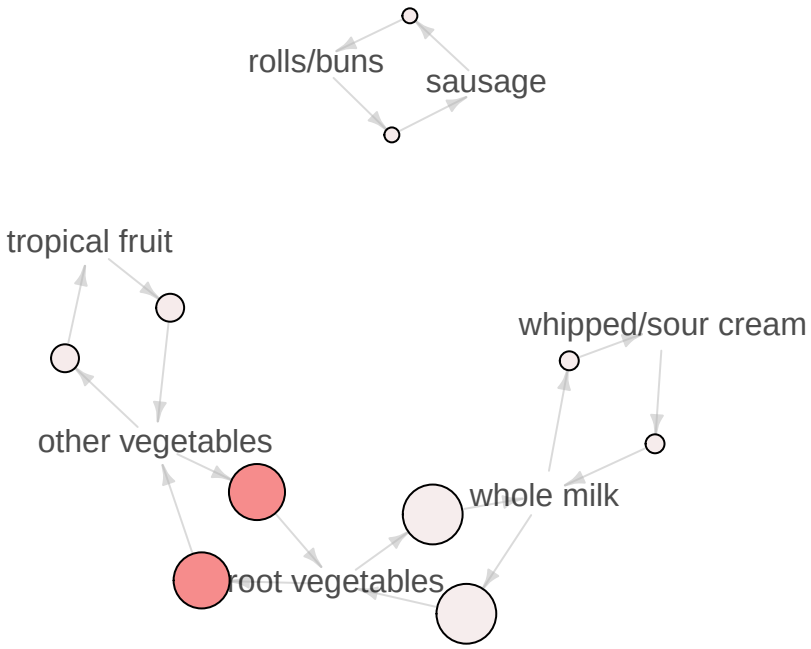
```
## [45] {whole milk}         => {yogurt}             0.05602440 0.21925985
## [46] {rolls/buns}         => {other vegetables}   0.04260295 0.23161968
## [47] {other vegetables}   => {rolls/buns}         0.04260295 0.22017867
## [48] {rolls/buns}         => {whole milk}         0.05663447 0.30790492
## [49] {whole milk}         => {rolls/buns}         0.05663447 0.22164743
## [50] {other vegetables}   => {whole milk}         0.07483477 0.38675775
## [51] {whole milk}         => {other vegetables}   0.07483477 0.29287704
##      coverage   lift      count
## [1]  1.00000000 1.0000000  792
## [2]  1.00000000 1.0000000  875
## [3]  1.00000000 1.0000000  814
## [4]  1.00000000 1.0000000  969
## [5]  1.00000000 1.0000000  924
## [6]  1.00000000 1.0000000 1087
## [7]  1.00000000 1.0000000 1032
## [8]  1.00000000 1.0000000 1072
## [9]  1.00000000 1.0000000 1715
## [10] 1.00000000 1.0000000 1372
## [11] 1.00000000 1.0000000 1809
## [12] 1.00000000 1.0000000 1903
## [13] 1.00000000 1.0000000 2513
## [14] 0.07168277 1.7597542  317
## [15] 0.25551601 1.7597542  317
## [16] 0.07564820 1.5570432  296
## [17] 0.25551601 1.5570432  296
## [18] 0.08896797 1.4625865  327
## [19] 0.25551601 1.4625865  327
## [20] 0.08276563 1.4423768  300
## [21] 0.25551601 1.4423768  300
## [22] 0.09395018 1.7710480  301
## [23] 0.18393493 1.7710480  301
## [24] 0.11052364 1.2169396  338
## [25] 0.25551601 1.2169396  338
## [26] 0.10493137 1.7677896  353
## [27] 0.19349263 1.7677896  353
## [28] 0.10493137 1.5775950  416
## [29] 0.25551601 1.5775950  416
## [30] 0.10899847 2.2466049  466
## [31] 0.19349263 2.2466049  466
## [32] 0.10899847 1.7560310  481
## [33] 0.25551601 1.7560310  481
## [34] 0.17437722 1.1951242  377
## [35] 0.18393493 1.1951242  377
## [36] 0.17437722 0.9703476  322
## [37] 0.19349263 0.9703476  322
## [38] 0.17437722 0.8991124  394
## [39] 0.25551601 0.8991124  394
## [40] 0.13950178 1.3393633  338
## [41] 0.18393493 1.3393633  338
## [42] 0.13950178 1.6084566  427
## [43] 0.19349263 1.6084566  427
## [44] 0.13950178 1.5717351  551
## [45] 0.25551601 1.5717351  551
## [46] 0.18393493 1.1970465  419
```

```
## [47] 0.19349263 1.1970465  419
## [48] 0.18393493 1.2050318  557
## [49] 0.25551601 1.2050318  557
## [50] 0.19349263 1.5136341  736
## [51] 0.25551601 1.5136341  736
```

```
plot(asso_rules2, method='graph')
```




```
plot(head(asso_rules2, 10, by='lift'), method='graph')
```



Reducing the total rule number to half by setting support = 0.03 and confidence = 0.08, we see a clearer graph. whole milk, other vegetables, rolls/buns are several of the obvious rules with the most connections. We would still like to explore with more rules.

support=0.01, confidence=0.05

```
asso_rules3 = apriori(trans_grocery,
                      parameter=list(support=0.01, confidence=0.05, minlen=1.5))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.05    0.1    1 none FALSE            TRUE       5    0.01      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 98
##
## set item appearances ...[0 item(s)] done [0.00s].
```

```
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.01s].
## sorting and recoding items ... [88 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [541 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
arules::inspect(asso_rules3)
```

```
##       lhs                          rhs                          support confidence   coverage
## [1]   {}                        => {canned beer}              0.07768175 0.07768175 1.00000000 1.000
## [2]   {}                        => {coffee}                   0.05805796 0.05805796 1.00000000 1.000
## [3]   {}                        => {beef}                     0.05246568 0.05246568 1.00000000 1.000
## [4]   {}                        => {curd}                     0.05327911 0.05327911 1.00000000 1.000
## [5]   {}                        => {napkins}                  0.05236401 0.05236401 1.00000000 1.000
## [6]   {}                        => {pork}                     0.05765125 0.05765125 1.00000000 1.000
## [7]   {}                        => {frankfurter}              0.05897306 0.05897306 1.00000000 1.000
## [8]   {}                        => {bottled beer}             0.08052872 0.08052872 1.00000000 1.000
## [9]   {}                        => {brown bread}              0.06487036 0.06487036 1.00000000 1.000
## [10]  {}                        => {margarine}                0.05856634 0.05856634 1.00000000 1.000
## [11]  {}                        => {butter}                   0.05541434 0.05541434 1.00000000 1.000
## [12]  {}                        => {newspapers}               0.07981698 0.07981698 1.00000000 1.000
## [13]  {}                        => {domestic eggs}            0.06344687 0.06344687 1.00000000 1.000
## [14]  {}                        => {fruit/vegetable juice}    0.07229283 0.07229283 1.00000000 1.000
## [15]  {}                        => {whipped/sour cream}       0.07168277 0.07168277 1.00000000 1.000
## [16]  {}                        => {pip fruit}                0.07564820 0.07564820 1.00000000 1.000
## [17]  {}                        => {pastry}                   0.08896797 0.08896797 1.00000000 1.000
## [18]  {}                        => {citrus fruit}             0.08276563 0.08276563 1.00000000 1.000
## [19]  {}                        => {shopping bags}            0.09852567 0.09852567 1.00000000 1.000
## [20]  {}                        => {sausage}                  0.09395018 0.09395018 1.00000000 1.000
## [21]  {}                        => {bottled water}            0.11052364 0.11052364 1.00000000 1.000
## [22]  {}                        => {tropical fruit}           0.10493137 0.10493137 1.00000000 1.000
## [23]  {}                        => {root vegetables}          0.10899847 0.10899847 1.00000000 1.000
## [24]  {}                        => {soda}                     0.17437722 0.17437722 1.00000000 1.000
## [25]  {}                        => {yogurt}                   0.13950178 0.13950178 1.00000000 1.000
## [26]  {}                        => {rolls/buns}               0.18393493 0.18393493 1.00000000 1.000
## [27]  {}                        => {other vegetables}         0.19349263 0.19349263 1.00000000 1.000
## [28]  {}                        => {whole milk}               0.25551601 0.25551601 1.00000000 1.000
## [29]  {hard cheese}             => {whole milk}               0.01006609 0.41078838 0.02450432 1.607
## [30]  {butter milk}             => {other vegetables}         0.01037112 0.37090909 0.02796136 1.916
## [31]  {other vegetables}        => {butter milk}              0.01037112 0.05359958 0.19349263 1.916
## [32]  {butter milk}             => {whole milk}               0.01159126 0.41454545 0.02796136 1.622
## [33]  {ham}                     => {whole milk}               0.01148958 0.44140625 0.02602949 1.727
## [34]  {sliced cheese}           => {whole milk}               0.01077783 0.43983402 0.02450432 1.721
## [35]  {oil}                     => {whole milk}               0.01128622 0.40217391 0.02806304 1.573
## [36]  {onions}                  => {other vegetables}         0.01423488 0.45901639 0.03101169 2.371
## [37]  {other vegetables}        => {onions}                   0.01423488 0.07356805 0.19349263 2.371
## [38]  {onions}                  => {whole milk}               0.01209964 0.39016393 0.03101169 1.526
## [39]  {berries}                 => {yogurt}                   0.01057448 0.31804281 0.03324860 2.279
## [40]  {yogurt}                  => {berries}                  0.01057448 0.07580175 0.13950178 2.279
## [41]  {berries}                 => {other vegetables}         0.01026945 0.30886850 0.03324860 1.596
## [42]  {other vegetables}        => {berries}                  0.01026945 0.05307409 0.19349263 1.596
## [43]  {berries}                 => {whole milk}               0.01179461 0.35474006 0.03324860 1.388
## [44]  {hamburger meat}          => {other vegetables}         0.01382816 0.41590214 0.03324860 2.149
```

```
## [45]  {other vegetables}         => {hamburger meat}           0.01382816 0.07146611 0.19349263 2.149
## [46]  {hamburger meat}           => {whole milk}               0.01474326 0.44342508 0.03324860 1.735
## [47]  {whole milk}               => {hamburger meat}           0.01474326 0.05769996 0.25551601 1.735
## [48]  {hygiene articles}         => {whole milk}               0.01281139 0.38888889 0.03294357 1.521
## [49]  {whole milk}               => {hygiene articles}         0.01281139 0.05013928 0.25551601 1.521
## [50]  {salty snack}              => {other vegetables}         0.01077783 0.28494624 0.03782410 1.471
## [51]  {other vegetables}         => {salty snack}              0.01077783 0.05570152 0.19349263 1.471
## [52]  {salty snack}              => {whole milk}               0.01118454 0.29569892 0.03782410 1.157
## [53]  {sugar}                    => {other vegetables}         0.01077783 0.31831832 0.03385867 1.645
## [54]  {other vegetables}         => {sugar}                    0.01077783 0.05570152 0.19349263 1.645
## [55]  {sugar}                    => {whole milk}               0.01504830 0.44444444 0.03385867 1.739
## [56]  {whole milk}               => {sugar}                    0.01504830 0.05889375 0.25551601 1.739
## [57]  {waffles}                  => {other vegetables}         0.01006609 0.26190476 0.03843416 1.353
## [58]  {other vegetables}         => {waffles}                  0.01006609 0.05202312 0.19349263 1.353
## [59]  {waffles}                  => {whole milk}               0.01270971 0.33068783 0.03843416 1.294
## [60]  {long life bakery product} => {other vegetables}         0.01067616 0.28532609 0.03741739 1.474
## [61]  {other vegetables}         => {long life bakery product} 0.01067616 0.05517604 0.19349263 1.474
## [62]  {long life bakery product} => {whole milk}               0.01352313 0.36141304 0.03741739 1.414
## [63]  {whole milk}               => {long life bakery product} 0.01352313 0.05292479 0.25551601 1.414
## [64]  {dessert}                  => {other vegetables}         0.01159126 0.31232877 0.03711235 1.614
## [65]  {other vegetables}         => {dessert}                  0.01159126 0.05990541 0.19349263 1.614
## [66]  {dessert}                  => {whole milk}               0.01372649 0.36986301 0.03711235 1.447
## [67]  {whole milk}               => {dessert}                  0.01372649 0.05372065 0.25551601 1.447
## [68]  {canned beer}              => {shopping bags}            0.01138790 0.14659686 0.07768175 1.487
## [69]  {shopping bags}            => {canned beer}              0.01138790 0.11558308 0.09852567 1.487
## [70]  {canned beer}              => {soda}                     0.01382816 0.17801047 0.07768175 1.020
## [71]  {soda}                     => {canned beer}              0.01382816 0.07930029 0.17437722 1.020
## [72]  {canned beer}              => {rolls/buns}               0.01128622 0.14528796 0.07768175 0.789
## [73]  {rolls/buns}               => {canned beer}              0.01128622 0.06135987 0.18393493 0.789
## [74]  {cream cheese }            => {yogurt}                   0.01240468 0.31282051 0.03965430 2.242
## [75]  {yogurt}                   => {cream cheese }            0.01240468 0.08892128 0.13950178 2.242
## [76]  {cream cheese }            => {other vegetables}         0.01372649 0.34615385 0.03965430 1.788
## [77]  {other vegetables}         => {cream cheese }            0.01372649 0.07094062 0.19349263 1.788
## [78]  {cream cheese }            => {whole milk}               0.01647178 0.41538462 0.03965430 1.625
## [79]  {whole milk}               => {cream cheese }            0.01647178 0.06446478 0.25551601 1.625
## [80]  {chicken}                  => {root vegetables}          0.01087951 0.25355450 0.04290798 2.326
## [81]  {root vegetables}          => {chicken}                  0.01087951 0.09981343 0.10899847 2.326
## [82]  {chicken}                  => {other vegetables}         0.01789527 0.41706161 0.04290798 2.155
## [83]  {other vegetables}         => {chicken}                  0.01789527 0.09248555 0.19349263 2.155
## [84]  {chicken}                  => {whole milk}               0.01759024 0.40995261 0.04290798 1.604
## [85]  {whole milk}               => {chicken}                  0.01759024 0.06884202 0.25551601 1.604
## [86]  {white bread}              => {soda}                     0.01026945 0.24396135 0.04209456 1.399
## [87]  {soda}                     => {white bread}              0.01026945 0.05889213 0.17437722 1.399
## [88]  {white bread}              => {other vegetables}         0.01372649 0.32608696 0.04209456 1.685
## [89]  {other vegetables}         => {white bread}              0.01372649 0.07094062 0.19349263 1.685
## [90]  {white bread}              => {whole milk}               0.01708185 0.40579710 0.04209456 1.588
## [91]  {whole milk}               => {white bread}              0.01708185 0.06685237 0.25551601 1.588
## [92]  {chocolate}                => {soda}                     0.01352313 0.27254098 0.04961871 1.562
## [93]  {soda}                     => {chocolate}                0.01352313 0.07755102 0.17437722 1.562
## [94]  {chocolate}                => {rolls/buns}               0.01179461 0.23770492 0.04961871 1.292
## [95]  {rolls/buns}               => {chocolate}                0.01179461 0.06412383 0.18393493 1.292
## [96]  {chocolate}                => {other vegetables}         0.01270971 0.25614754 0.04961871 1.323
## [97]  {other vegetables}         => {chocolate}                0.01270971 0.06568576 0.19349263 1.323
## [98]  {chocolate}                => {whole milk}               0.01667514 0.33606557 0.04961871 1.315
```

```
## [99]  {whole milk}              => {chocolate}              0.01667514 0.06526064 0.25551601 1.315
## [100] {coffee}                  => {rolls/buns}             0.01098119 0.18914186 0.05805796 1.028
## [101] {rolls/buns}              => {coffee}                 0.01098119 0.05970149 0.18393493 1.028
## [102] {coffee}                  => {other vegetables}       0.01342145 0.23117338 0.05805796 1.194
## [103] {other vegetables}        => {coffee}                 0.01342145 0.06936416 0.19349263 1.194
## [104] {coffee}                  => {whole milk}             0.01870869 0.32224168 0.05805796 1.261
## [105] {whole milk}              => {coffee}                 0.01870869 0.07321926 0.25551601 1.261
## [106] {frozen vegetables}       => {root vegetables}        0.01159126 0.24101480 0.04809354 2.211
## [107] {root vegetables}         => {frozen vegetables}      0.01159126 0.10634328 0.10899847 2.211
## [108] {frozen vegetables}       => {yogurt}                 0.01240468 0.25792812 0.04809354 1.848
## [109] {yogurt}                  => {frozen vegetables}      0.01240468 0.08892128 0.13950178 1.848
## [110] {frozen vegetables}       => {rolls/buns}             0.01016777 0.21141649 0.04809354 1.149
## [111] {rolls/buns}              => {frozen vegetables}      0.01016777 0.05527916 0.18393493 1.149
## [112] {frozen vegetables}       => {other vegetables}       0.01779359 0.36997886 0.04809354 1.911
## [113] {other vegetables}        => {frozen vegetables}      0.01779359 0.09196006 0.19349263 1.911
## [114] {frozen vegetables}       => {whole milk}             0.02043721 0.42494715 0.04809354 1.663
## [115] {whole milk}              => {frozen vegetables}      0.02043721 0.07998408 0.25551601 1.663
## [116] {beef}                    => {root vegetables}        0.01738688 0.33139535 0.05246568 3.040
## [117] {root vegetables}         => {beef}                   0.01738688 0.15951493 0.10899847 3.040
## [118] {beef}                    => {yogurt}                 0.01169293 0.22286822 0.05246568 1.597
## [119] {yogurt}                  => {beef}                   0.01169293 0.08381924 0.13950178 1.597
## [120] {beef}                    => {rolls/buns}             0.01362481 0.25968992 0.05246568 1.411
## [121] {rolls/buns}              => {beef}                   0.01362481 0.07407407 0.18393493 1.411
## [122] {beef}                    => {other vegetables}       0.01972547 0.37596899 0.05246568 1.943
## [123] {other vegetables}        => {beef}                   0.01972547 0.10194430 0.19349263 1.943
## [124] {beef}                    => {whole milk}             0.02125064 0.40503876 0.05246568 1.585
## [125] {whole milk}              => {beef}                   0.02125064 0.08316753 0.25551601 1.585
## [126] {curd}                    => {whipped/sour cream}     0.01047280 0.19656489 0.05327911 2.742
## [127] {whipped/sour cream}      => {curd}                   0.01047280 0.14609929 0.07168277 2.742
## [128] {curd}                    => {tropical fruit}         0.01026945 0.19274809 0.05327911 1.836
## [129] {tropical fruit}          => {curd}                   0.01026945 0.09786822 0.10493137 1.836
## [130] {curd}                    => {root vegetables}        0.01087951 0.20419847 0.05327911 1.873
## [131] {root vegetables}         => {curd}                   0.01087951 0.09981343 0.10899847 1.873
## [132] {curd}                    => {yogurt}                 0.01728521 0.32442748 0.05327911 2.325
## [133] {yogurt}                  => {curd}                   0.01728521 0.12390671 0.13950178 2.325
## [134] {curd}                    => {rolls/buns}             0.01006609 0.18893130 0.05327911 1.027
## [135] {rolls/buns}              => {curd}                   0.01006609 0.05472637 0.18393493 1.027
## [136] {curd}                    => {other vegetables}       0.01718353 0.32251908 0.05327911 1.666
## [137] {other vegetables}        => {curd}                   0.01718353 0.08880715 0.19349263 1.666
## [138] {curd}                    => {whole milk}             0.02613116 0.49045802 0.05327911 1.919
## [139] {whole milk}              => {curd}                   0.02613116 0.10226821 0.25551601 1.919
## [140] {napkins}                 => {tropical fruit}         0.01006609 0.19223301 0.05236401 1.831
## [141] {tropical fruit}          => {napkins}                0.01006609 0.09593023 0.10493137 1.831
## [142] {napkins}                 => {soda}                   0.01199797 0.22912621 0.05236401 1.313
## [143] {soda}                    => {napkins}                0.01199797 0.06880466 0.17437722 1.313
## [144] {napkins}                 => {yogurt}                 0.01230300 0.23495146 0.05236401 1.684
## [145] {yogurt}                  => {napkins}                0.01230300 0.08819242 0.13950178 1.684
## [146] {napkins}                 => {rolls/buns}             0.01169293 0.22330097 0.05236401 1.214
## [147] {rolls/buns}              => {napkins}                0.01169293 0.06357103 0.18393493 1.214
## [148] {napkins}                 => {other vegetables}       0.01443823 0.27572816 0.05236401 1.425
## [149] {other vegetables}        => {napkins}                0.01443823 0.07461902 0.19349263 1.425
## [150] {napkins}                 => {whole milk}             0.01972547 0.37669903 0.05236401 1.474
## [151] {whole milk}              => {napkins}                0.01972547 0.07719857 0.25551601 1.474
## [152] {pork}                    => {root vegetables}        0.01362481 0.23633157 0.05765125 2.168
```

```
## [153] {root vegetables}         => {pork}                0.01362481 0.12500000 0.10899847 2.168
## [154] {pork}                    => {soda}                0.01189629 0.20634921 0.05765125 1.183
## [155] {soda}                    => {pork}                0.01189629 0.06822157 0.17437722 1.183
## [156] {pork}                    => {rolls/buns}          0.01128622 0.19576720 0.05765125 1.064
## [157] {rolls/buns}              => {pork}                0.01128622 0.06135987 0.18393493 1.064
## [158] {pork}                    => {other vegetables}    0.02165735 0.37566138 0.05765125 1.941
## [159] {other vegetables}        => {pork}                0.02165735 0.11192853 0.19349263 1.941
## [160] {pork}                    => {whole milk}          0.02216573 0.38447972 0.05765125 1.504
## [161] {whole milk}              => {pork}                0.02216573 0.08674891 0.25551601 1.504
## [162] {frankfurter}             => {sausage}             0.01006609 0.17068966 0.05897306 1.816
## [163] {sausage}                 => {frankfurter}         0.01006609 0.10714286 0.09395018 1.816
## [164] {frankfurter}             => {root vegetables}     0.01016777 0.17241379 0.05897306 1.581
## [165] {root vegetables}         => {frankfurter}         0.01016777 0.09328358 0.10899847 1.581
## [166] {frankfurter}             => {soda}                0.01128622 0.19137931 0.05897306 1.097
## [167] {soda}                    => {frankfurter}         0.01128622 0.06472303 0.17437722 1.097
## [168] {frankfurter}             => {yogurt}              0.01118454 0.18965517 0.05897306 1.359
## [169] {yogurt}                  => {frankfurter}         0.01118454 0.08017493 0.13950178 1.359
## [170] {frankfurter}             => {rolls/buns}          0.01921708 0.32586207 0.05897306 1.771
## [171] {rolls/buns}              => {frankfurter}         0.01921708 0.10447761 0.18393493 1.771
## [172] {frankfurter}             => {other vegetables}    0.01647178 0.27931034 0.05897306 1.443
## [173] {other vegetables}        => {frankfurter}         0.01647178 0.08512874 0.19349263 1.443
## [174] {frankfurter}             => {whole milk}          0.02053889 0.34827586 0.05897306 1.363
## [175] {whole milk}              => {frankfurter}         0.02053889 0.08038201 0.25551601 1.363
## [176] {bottled beer}            => {bottled water}       0.01576004 0.19570707 0.08052872 1.770
## [177] {bottled water}           => {bottled beer}        0.01576004 0.14259430 0.11052364 1.770
## [178] {bottled beer}            => {soda}                0.01698017 0.21085859 0.08052872 1.209
## [179] {soda}                    => {bottled beer}        0.01698017 0.09737609 0.17437722 1.209
## [180] {bottled beer}            => {rolls/buns}          0.01362481 0.16919192 0.08052872 0.919
## [181] {rolls/buns}              => {bottled beer}        0.01362481 0.07407407 0.18393493 0.919
## [182] {bottled beer}            => {other vegetables}    0.01616675 0.20075758 0.08052872 1.037
## [183] {other vegetables}        => {bottled beer}        0.01616675 0.08355229 0.19349263 1.037
## [184] {bottled beer}            => {whole milk}          0.02043721 0.25378788 0.08052872 0.993
## [185] {whole milk}              => {bottled beer}        0.02043721 0.07998408 0.25551601 0.993
## [186] {brown bread}             => {sausage}             0.01067616 0.16457680 0.06487036 1.751
## [187] {sausage}                 => {brown bread}         0.01067616 0.11363636 0.09395018 1.751
## [188] {brown bread}             => {tropical fruit}      0.01067616 0.16457680 0.06487036 1.568
## [189] {tropical fruit}          => {brown bread}         0.01067616 0.10174419 0.10493137 1.568
## [190] {brown bread}             => {root vegetables}     0.01016777 0.15673981 0.06487036 1.438
## [191] {root vegetables}         => {brown bread}         0.01016777 0.09328358 0.10899847 1.438
## [192] {brown bread}             => {soda}                0.01260803 0.19435737 0.06487036 1.114
## [193] {soda}                    => {brown bread}         0.01260803 0.07230321 0.17437722 1.114
## [194] {brown bread}             => {yogurt}              0.01453991 0.22413793 0.06487036 1.606
## [195] {yogurt}                  => {brown bread}         0.01453991 0.10422741 0.13950178 1.606
## [196] {brown bread}             => {rolls/buns}          0.01260803 0.19435737 0.06487036 1.056
## [197] {rolls/buns}              => {brown bread}         0.01260803 0.06854616 0.18393493 1.056
## [198] {brown bread}             => {other vegetables}    0.01870869 0.28840125 0.06487036 1.490
## [199] {other vegetables}        => {brown bread}         0.01870869 0.09668944 0.19349263 1.490
## [200] {brown bread}             => {whole milk}          0.02521607 0.38871473 0.06487036 1.521
## [201] {whole milk}              => {brown bread}         0.02521607 0.09868683 0.25551601 1.521
## [202] {margarine}               => {bottled water}       0.01026945 0.17534722 0.05856634 1.586
## [203] {bottled water}           => {margarine}           0.01026945 0.09291628 0.11052364 1.586
## [204] {margarine}               => {root vegetables}     0.01108287 0.18923611 0.05856634 1.736
## [205] {root vegetables}         => {margarine}           0.01108287 0.10167910 0.10899847 1.736
## [206] {margarine}               => {soda}                0.01016777 0.17361111 0.05856634 0.995
```

```
## [207] {soda}                    => {margarine}          0.01016777 0.05830904 0.17437722 0.995
## [208] {margarine}               => {yogurt}             0.01423488 0.24305556 0.05856634 1.742
## [209] {yogurt}                  => {margarine}          0.01423488 0.10204082 0.13950178 1.742
## [210] {margarine}               => {rolls/buns}         0.01474326 0.25173611 0.05856634 1.368
## [211] {rolls/buns}              => {margarine}          0.01474326 0.08015478 0.18393493 1.368
## [212] {margarine}               => {other vegetables}   0.01972547 0.33680556 0.05856634 1.740
## [213] {other vegetables}        => {margarine}          0.01972547 0.10194430 0.19349263 1.740
## [214] {margarine}               => {whole milk}         0.02419929 0.41319444 0.05856634 1.617
## [215] {whole milk}              => {margarine}          0.02419929 0.09470752 0.25551601 1.617
## [216] {butter}                  => {whipped/sour cream} 0.01016777 0.18348624 0.05541434 2.559
## [217] {whipped/sour cream}      => {butter}             0.01016777 0.14184397 0.07168277 2.559
## [218] {butter}                  => {root vegetables}    0.01291307 0.23302752 0.05541434 2.137
## [219] {root vegetables}         => {butter}             0.01291307 0.11847015 0.10899847 2.137
## [220] {butter}                  => {yogurt}             0.01464159 0.26422018 0.05541434 1.894
## [221] {yogurt}                  => {butter}             0.01464159 0.10495627 0.13950178 1.894
## [222] {butter}                  => {rolls/buns}         0.01342145 0.24220183 0.05541434 1.316
## [223] {rolls/buns}              => {butter}             0.01342145 0.07296849 0.18393493 1.316
## [224] {butter}                  => {other vegetables}   0.02003050 0.36146789 0.05541434 1.868
## [225] {other vegetables}        => {butter}             0.02003050 0.10352076 0.19349263 1.868
## [226] {butter}                  => {whole milk}         0.02755465 0.49724771 0.05541434 1.946
## [227] {whole milk}              => {butter}             0.02755465 0.10783924 0.25551601 1.946
## [228] {newspapers}              => {bottled water}      0.01128622 0.14140127 0.07981698 1.279
## [229] {bottled water}           => {newspapers}         0.01128622 0.10211592 0.11052364 1.279
## [230] {newspapers}              => {tropical fruit}     0.01179461 0.14777070 0.07981698 1.408
## [231] {tropical fruit}          => {newspapers}         0.01179461 0.11240310 0.10493137 1.408
## [232] {newspapers}              => {root vegetables}    0.01148958 0.14394904 0.07981698 1.320
## [233] {root vegetables}         => {newspapers}         0.01148958 0.10541045 0.10899847 1.320
## [234] {newspapers}              => {soda}               0.01464159 0.18343949 0.07981698 1.051
## [235] {soda}                    => {newspapers}         0.01464159 0.08396501 0.17437722 1.051
## [236] {newspapers}              => {yogurt}             0.01535333 0.19235669 0.07981698 1.378
## [237] {yogurt}                  => {newspapers}         0.01535333 0.11005831 0.13950178 1.378
## [238] {newspapers}              => {rolls/buns}         0.01972547 0.24713376 0.07981698 1.343
## [239] {rolls/buns}              => {newspapers}         0.01972547 0.10724157 0.18393493 1.343
## [240] {newspapers}              => {other vegetables}   0.01931876 0.24203822 0.07981698 1.250
## [241] {other vegetables}        => {newspapers}         0.01931876 0.09984235 0.19349263 1.250
## [242] {newspapers}              => {whole milk}         0.02735130 0.34267516 0.07981698 1.341
## [243] {whole milk}              => {newspapers}         0.02735130 0.10704337 0.25551601 1.341
## [244] {domestic eggs}           => {citrus fruit}       0.01037112 0.16346154 0.06344687 1.974
## [245] {citrus fruit}            => {domestic eggs}      0.01037112 0.12530713 0.08276563 1.974
## [246] {domestic eggs}           => {tropical fruit}     0.01138790 0.17948718 0.06344687 1.710
## [247] {tropical fruit}          => {domestic eggs}      0.01138790 0.10852713 0.10493137 1.710
## [248] {domestic eggs}           => {root vegetables}    0.01433655 0.22596154 0.06344687 2.073
## [249] {root vegetables}         => {domestic eggs}      0.01433655 0.13152985 0.10899847 2.073
## [250] {domestic eggs}           => {soda}               0.01240468 0.19551282 0.06344687 1.121
## [251] {soda}                    => {domestic eggs}      0.01240468 0.07113703 0.17437722 1.121
## [252] {domestic eggs}           => {yogurt}             0.01433655 0.22596154 0.06344687 1.619
## [253] {yogurt}                  => {domestic eggs}      0.01433655 0.10276968 0.13950178 1.619
## [254] {domestic eggs}           => {rolls/buns}         0.01565836 0.24679487 0.06344687 1.341
## [255] {rolls/buns}              => {domestic eggs}      0.01565836 0.08512991 0.18393493 1.341
## [256] {domestic eggs}           => {other vegetables}   0.02226741 0.35096154 0.06344687 1.813
## [257] {other vegetables}        => {domestic eggs}      0.02226741 0.11508145 0.19349263 1.813
## [258] {domestic eggs}           => {whole milk}         0.02999492 0.47275641 0.06344687 1.850
## [259] {whole milk}              => {domestic eggs}      0.02999492 0.11738957 0.25551601 1.850
## [260] {fruit/vegetable juice}   => {citrus fruit}       0.01037112 0.14345992 0.07229283 1.733
```

```
## [261] {citrus fruit}              => {fruit/vegetable juice}    0.01037112 0.12530713 0.08276563 1.733
## [262] {fruit/vegetable juice}     => {shopping bags}            0.01067616 0.14767932 0.07229283 1.498
## [263] {shopping bags}             => {fruit/vegetable juice}    0.01067616 0.10835913 0.09852567 1.498
## [264] {fruit/vegetable juice}     => {sausage}                  0.01006609 0.13924051 0.07229283 1.481
## [265] {sausage}                   => {fruit/vegetable juice}    0.01006609 0.10714286 0.09395018 1.481
## [266] {fruit/vegetable juice}     => {bottled water}            0.01423488 0.19690577 0.07229283 1.781
## [267] {bottled water}             => {fruit/vegetable juice}    0.01423488 0.12879485 0.11052364 1.781
## [268] {fruit/vegetable juice}     => {tropical fruit}           0.01372649 0.18987342 0.07229283 1.809
## [269] {tropical fruit}            => {fruit/vegetable juice}    0.01372649 0.13081395 0.10493137 1.809
## [270] {fruit/vegetable juice}     => {root vegetables}          0.01199797 0.16596343 0.07229283 1.522
## [271] {root vegetables}           => {fruit/vegetable juice}    0.01199797 0.11007463 0.10899847 1.522
## [272] {fruit/vegetable juice}     => {soda}                     0.01840366 0.25457103 0.07229283 1.459
## [273] {soda}                      => {fruit/vegetable juice}    0.01840366 0.10553936 0.17437722 1.459
## [274] {fruit/vegetable juice}     => {yogurt}                   0.01870869 0.25879044 0.07229283 1.855
## [275] {yogurt}                    => {fruit/vegetable juice}    0.01870869 0.13411079 0.13950178 1.855
## [276] {fruit/vegetable juice}     => {rolls/buns}               0.01453991 0.20112518 0.07229283 1.093
## [277] {rolls/buns}                => {fruit/vegetable juice}    0.01453991 0.07904920 0.18393493 1.093
## [278] {fruit/vegetable juice}     => {other vegetables}         0.02104728 0.29113924 0.07229283 1.504
## [279] {other vegetables}          => {fruit/vegetable juice}    0.02104728 0.10877562 0.19349263 1.504
## [280] {fruit/vegetable juice}     => {whole milk}               0.02663955 0.36849508 0.07229283 1.442
## [281] {whole milk}                => {fruit/vegetable juice}    0.02663955 0.10425786 0.25551601 1.442
## [282] {whipped/sour cream}        => {citrus fruit}             0.01087951 0.15177305 0.07168277 1.833
## [283] {citrus fruit}              => {whipped/sour cream}       0.01087951 0.13144963 0.08276563 1.833
## [284] {whipped/sour cream}        => {tropical fruit}           0.01382816 0.19290780 0.07168277 1.838
## [285] {tropical fruit}            => {whipped/sour cream}       0.01382816 0.13178295 0.10493137 1.838
## [286] {whipped/sour cream}        => {root vegetables}          0.01708185 0.23829787 0.07168277 2.186
## [287] {root vegetables}           => {whipped/sour cream}       0.01708185 0.15671642 0.10899847 2.186
## [288] {whipped/sour cream}        => {soda}                     0.01159126 0.16170213 0.07168277 0.927
## [289] {soda}                      => {whipped/sour cream}       0.01159126 0.06647230 0.17437722 0.927
## [290] {whipped/sour cream}        => {yogurt}                   0.02074225 0.28936170 0.07168277 2.074
## [291] {yogurt}                    => {whipped/sour cream}       0.02074225 0.14868805 0.13950178 2.074
## [292] {whipped/sour cream}        => {rolls/buns}               0.01464159 0.20425532 0.07168277 1.110
## [293] {rolls/buns}                => {whipped/sour cream}       0.01464159 0.07960199 0.18393493 1.110
## [294] {whipped/sour cream}        => {other vegetables}         0.02887646 0.40283688 0.07168277 2.081
## [295] {other vegetables}          => {whipped/sour cream}       0.02887646 0.14923805 0.19349263 2.081
## [296] {whipped/sour cream}        => {whole milk}               0.03223183 0.44964539 0.07168277 1.759
## [297] {whole milk}                => {whipped/sour cream}       0.03223183 0.12614405 0.25551601 1.759
## [298] {pip fruit}                 => {pastry}                   0.01067616 0.14112903 0.07564820 1.586
## [299] {pastry}                    => {pip fruit}                0.01067616 0.12000000 0.08896797 1.586
## [300] {pip fruit}                 => {citrus fruit}             0.01382816 0.18279570 0.07564820 2.208
## [301] {citrus fruit}              => {pip fruit}                0.01382816 0.16707617 0.08276563 2.208
## [302] {pip fruit}                 => {sausage}                  0.01077783 0.14247312 0.07564820 1.516
## [303] {sausage}                   => {pip fruit}                0.01077783 0.11471861 0.09395018 1.516
## [304] {pip fruit}                 => {bottled water}            0.01057448 0.13978495 0.07564820 1.264
## [305] {bottled water}             => {pip fruit}                0.01057448 0.09567617 0.11052364 1.264
## [306] {pip fruit}                 => {tropical fruit}           0.02043721 0.27016129 0.07564820 2.574
## [307] {tropical fruit}            => {pip fruit}                0.02043721 0.19476744 0.10493137 2.574
## [308] {pip fruit}                 => {root vegetables}          0.01555669 0.20564516 0.07564820 1.886
## [309] {root vegetables}           => {pip fruit}                0.01555669 0.14272388 0.10899847 1.886
## [310] {pip fruit}                 => {soda}                     0.01331978 0.17607527 0.07564820 1.009
## [311] {soda}                      => {pip fruit}                0.01331978 0.07638484 0.17437722 1.009
## [312] {pip fruit}                 => {yogurt}                   0.01799695 0.23790323 0.07564820 1.705
## [313] {yogurt}                    => {pip fruit}                0.01799695 0.12900875 0.13950178 1.705
## [314] {pip fruit}                 => {rolls/buns}               0.01392984 0.18413978 0.07564820 1.001
```

```
## [315] {rolls/buns}          => {pip fruit}          0.01392984 0.07573245 0.18393493 1.00
## [316] {pip fruit}           => {other vegetables}   0.02613116 0.34543011 0.07564820 1.785
## [317] {other vegetables}    => {pip fruit}          0.02613116 0.13504992 0.19349263 1.785
## [318] {pip fruit}           => {whole milk}         0.03009659 0.39784946 0.07564820 1.557
## [319] {whole milk}          => {pip fruit}          0.03009659 0.11778750 0.25551601 1.557
## [320] {pastry}              => {shopping bags}      0.01189629 0.13371429 0.08896797 1.357
## [321] {shopping bags}       => {pastry}             0.01189629 0.12074303 0.09852567 1.357
## [322] {pastry}              => {sausage}            0.01250635 0.14057143 0.08896797 1.496
## [323] {sausage}             => {pastry}             0.01250635 0.13311688 0.09395018 1.496
## [324] {pastry}              => {tropical fruit}     0.01321810 0.14857143 0.08896797 1.415
## [325] {tropical fruit}      => {pastry}             0.01321810 0.12596899 0.10493137 1.415
## [326] {pastry}              => {root vegetables}    0.01098119 0.12342857 0.08896797 1.132
## [327] {root vegetables}     => {pastry}             0.01098119 0.10074627 0.10899847 1.132
## [328] {pastry}              => {soda}               0.02104728 0.23657143 0.08896797 1.356
## [329] {soda}                => {pastry}             0.02104728 0.12069971 0.17437722 1.356
## [330] {pastry}              => {yogurt}             0.01769192 0.19885714 0.08896797 1.425
## [331] {yogurt}              => {pastry}             0.01769192 0.12682216 0.13950178 1.425
## [332] {pastry}              => {rolls/buns}         0.02094560 0.23542857 0.08896797 1.279
## [333] {rolls/buns}          => {pastry}             0.02094560 0.11387507 0.18393493 1.279
## [334] {pastry}              => {other vegetables}   0.02257245 0.25371429 0.08896797 1.311
## [335] {other vegetables}    => {pastry}             0.02257245 0.11665791 0.19349263 1.311
## [336] {pastry}              => {whole milk}         0.03324860 0.37371429 0.08896797 1.462
## [337] {whole milk}          => {pastry}             0.03324860 0.13012336 0.25551601 1.462
## [338] {citrus fruit}        => {sausage}            0.01128622 0.13636364 0.08276563 1.451
## [339] {sausage}             => {citrus fruit}       0.01128622 0.12012987 0.09395018 1.451
## [340] {citrus fruit}        => {bottled water}      0.01352313 0.16339066 0.08276563 1.478
## [341] {bottled water}       => {citrus fruit}       0.01352313 0.12235511 0.11052364 1.478
## [342] {citrus fruit}        => {tropical fruit}     0.01992883 0.24078624 0.08276563 2.294
## [343] {tropical fruit}      => {citrus fruit}       0.01992883 0.18992248 0.10493137 2.294
## [344] {citrus fruit}        => {root vegetables}    0.01769192 0.21375921 0.08276563 1.961
## [345] {root vegetables}     => {citrus fruit}       0.01769192 0.16231343 0.10899847 1.961
## [346] {citrus fruit}        => {soda}               0.01281139 0.15479115 0.08276563 0.887
## [347] {soda}                => {citrus fruit}       0.01281139 0.07346939 0.17437722 0.887
## [348] {citrus fruit}        => {yogurt}             0.02165735 0.26167076 0.08276563 1.875
## [349] {yogurt}              => {citrus fruit}       0.02165735 0.15524781 0.13950178 1.875
## [350] {citrus fruit}        => {rolls/buns}         0.01677682 0.20270270 0.08276563 1.102
## [351] {rolls/buns}          => {citrus fruit}       0.01677682 0.09121061 0.18393493 1.102
## [352] {citrus fruit}        => {other vegetables}   0.02887646 0.34889435 0.08276563 1.803
## [353] {other vegetables}    => {citrus fruit}       0.02887646 0.14923805 0.19349263 1.803
## [354] {citrus fruit}        => {whole milk}         0.03050330 0.36855037 0.08276563 1.442
## [355] {whole milk}          => {citrus fruit}       0.03050330 0.11937923 0.25551601 1.442
## [356] {shopping bags}       => {sausage}            0.01565836 0.15892673 0.09852567 1.691
## [357] {sausage}             => {shopping bags}      0.01565836 0.16666667 0.09395018 1.691
## [358] {shopping bags}       => {bottled water}      0.01098119 0.11145511 0.09852567 1.008
## [359] {bottled water}       => {shopping bags}      0.01098119 0.09935603 0.11052364 1.008
## [360] {shopping bags}       => {tropical fruit}     0.01352313 0.13725490 0.09852567 1.308
## [361] {tropical fruit}      => {shopping bags}      0.01352313 0.12887597 0.10493137 1.308
## [362] {shopping bags}       => {root vegetables}    0.01281139 0.13003096 0.09852567 1.193
## [363] {root vegetables}     => {shopping bags}      0.01281139 0.11753731 0.10899847 1.193
## [364] {shopping bags}       => {soda}               0.02460600 0.24974200 0.09852567 1.433
## [365] {soda}                => {shopping bags}      0.02460600 0.14110787 0.17437722 1.433
## [366] {shopping bags}       => {yogurt}             0.01525165 0.15479876 0.09852567 1.109
## [367] {yogurt}              => {shopping bags}      0.01525165 0.10932945 0.13950178 1.109
## [368] {shopping bags}       => {rolls/buns}         0.01952211 0.19814241 0.09852567 1.077
```

```
## [369] {rolls/buns}              => {shopping bags}          0.01952211 0.10613599 0.18393493 1.07
## [370] {shopping bags}           => {other vegetables}       0.02318251 0.23529412 0.09852567 1.21
## [371] {other vegetables}        => {shopping bags}          0.02318251 0.11981083 0.19349263 1.21
## [372] {shopping bags}           => {whole milk}             0.02450432 0.24871001 0.09852567 0.97
## [373] {whole milk}              => {shopping bags}          0.02450432 0.09590131 0.25551601 0.97
## [374] {sausage}                 => {bottled water}          0.01199797 0.12770563 0.09395018 1.15
## [375] {bottled water}           => {sausage}                0.01199797 0.10855566 0.11052364 1.15
## [376] {sausage}                 => {tropical fruit}         0.01392984 0.14826840 0.09395018 1.41
## [377] {tropical fruit}          => {sausage}                0.01392984 0.13275194 0.10493137 1.41
## [378] {sausage}                 => {root vegetables}        0.01494662 0.15909091 0.09395018 1.45
## [379] {root vegetables}         => {sausage}                0.01494662 0.13712687 0.10899847 1.45
## [380] {sausage}                 => {soda}                   0.02430097 0.25865801 0.09395018 1.48
## [381] {soda}                    => {sausage}                0.02430097 0.13935860 0.17437722 1.48
## [382] {sausage}                 => {yogurt}                 0.01962379 0.20887446 0.09395018 1.49
## [383] {yogurt}                  => {sausage}                0.01962379 0.14067055 0.13950178 1.49
## [384] {sausage}                 => {rolls/buns}             0.03060498 0.32575758 0.09395018 1.77
## [385] {rolls/buns}              => {sausage}                0.03060498 0.16639027 0.18393493 1.77
## [386] {sausage}                 => {other vegetables}       0.02694459 0.28679654 0.09395018 1.48
## [387] {other vegetables}        => {sausage}                0.02694459 0.13925381 0.19349263 1.48
## [388] {sausage}                 => {whole milk}             0.02989324 0.31818182 0.09395018 1.24
## [389] {whole milk}              => {sausage}                0.02989324 0.11699164 0.25551601 1.24
## [390] {bottled water}           => {tropical fruit}         0.01850534 0.16743330 0.11052364 1.59
## [391] {tropical fruit}          => {bottled water}          0.01850534 0.17635659 0.10493137 1.59
## [392] {bottled water}           => {root vegetables}        0.01565836 0.14167433 0.11052364 1.29
## [393] {root vegetables}         => {bottled water}          0.01565836 0.14365672 0.10899847 1.29
## [394] {bottled water}           => {soda}                   0.02897814 0.26218951 0.11052364 1.50
## [395] {soda}                    => {bottled water}          0.02897814 0.16618076 0.17437722 1.50
## [396] {bottled water}           => {yogurt}                 0.02297916 0.20791168 0.11052364 1.49
## [397] {yogurt}                  => {bottled water}          0.02297916 0.16472303 0.13950178 1.49
## [398] {bottled water}           => {rolls/buns}             0.02419929 0.21895124 0.11052364 1.19
## [399] {rolls/buns}              => {bottled water}          0.02419929 0.13156440 0.18393493 1.19
## [400] {bottled water}           => {other vegetables}       0.02480935 0.22447102 0.11052364 1.16
## [401] {other vegetables}        => {bottled water}          0.02480935 0.12821860 0.19349263 1.16
## [402] {bottled water}           => {whole milk}             0.03436706 0.31094756 0.11052364 1.21
## [403] {whole milk}              => {bottled water}          0.03436706 0.13450060 0.25551601 1.21
## [404] {tropical fruit}          => {root vegetables}        0.02104728 0.20058140 0.10493137 1.84
## [405] {root vegetables}         => {tropical fruit}         0.02104728 0.19309701 0.10899847 1.84
## [406] {tropical fruit}          => {soda}                   0.02084392 0.19864341 0.10493137 1.13
## [407] {soda}                    => {tropical fruit}         0.02084392 0.11953353 0.17437722 1.13
## [408] {tropical fruit}          => {yogurt}                 0.02928317 0.27906977 0.10493137 2.00
## [409] {yogurt}                  => {tropical fruit}         0.02928317 0.20991254 0.13950178 2.00
## [410] {tropical fruit}          => {rolls/buns}             0.02460600 0.23449612 0.10493137 1.27
## [411] {rolls/buns}              => {tropical fruit}         0.02460600 0.13377557 0.18393493 1.27
## [412] {tropical fruit}          => {other vegetables}       0.03589222 0.34205426 0.10493137 1.76
## [413] {other vegetables}        => {tropical fruit}         0.03589222 0.18549658 0.19349263 1.76
## [414] {tropical fruit}          => {whole milk}             0.04229792 0.40310078 0.10493137 1.57
## [415] {whole milk}              => {tropical fruit}         0.04229792 0.16553920 0.25551601 1.57
## [416] {root vegetables}         => {soda}                   0.01860702 0.17070896 0.10899847 0.97
## [417] {soda}                    => {root vegetables}        0.01860702 0.10670554 0.17437722 0.97
## [418] {root vegetables}         => {yogurt}                 0.02582613 0.23694030 0.10899847 1.69
## [419] {yogurt}                  => {root vegetables}        0.02582613 0.18513120 0.13950178 1.69
## [420] {root vegetables}         => {rolls/buns}             0.02430097 0.22294776 0.10899847 1.21
## [421] {rolls/buns}              => {root vegetables}        0.02430097 0.13211719 0.18393493 1.21
## [422] {root vegetables}         => {other vegetables}       0.04738180 0.43470149 0.10899847 2.24
```

```
## [423] {other vegetables}         => {root vegetables}          0.04738180 0.24487651 0.19349263 2.24
## [424] {root vegetables}          => {whole milk}               0.04890696 0.44869403 0.10899847 1.75
## [425] {whole milk}               => {root vegetables}          0.04890696 0.19140470 0.25551601 1.75
## [426] {soda}                     => {yogurt}                   0.02735130 0.15685131 0.17437722 1.12
## [427] {yogurt}                   => {soda}                     0.02735130 0.19606414 0.13950178 1.12
## [428] {soda}                     => {rolls/buns}               0.03833249 0.21982507 0.17437722 1.19
## [429] {rolls/buns}               => {soda}                     0.03833249 0.20840243 0.18393493 1.19
## [430] {soda}                     => {other vegetables}         0.03274021 0.18775510 0.17437722 0.97
## [431] {other vegetables}         => {soda}                     0.03274021 0.16920652 0.19349263 0.97
## [432] {soda}                     => {whole milk}               0.04006101 0.22973761 0.17437722 0.89
## [433] {whole milk}               => {soda}                     0.04006101 0.15678472 0.25551601 0.89
## [434] {yogurt}                   => {rolls/buns}               0.03436706 0.24635569 0.13950178 1.33
## [435] {rolls/buns}               => {yogurt}                   0.03436706 0.18684356 0.18393493 1.33
## [436] {yogurt}                   => {other vegetables}         0.04341637 0.31122449 0.13950178 1.60
## [437] {other vegetables}         => {yogurt}                   0.04341637 0.22438255 0.19349263 1.60
## [438] {yogurt}                   => {whole milk}               0.05602440 0.40160350 0.13950178 1.57
## [439] {whole milk}               => {yogurt}                   0.05602440 0.21925985 0.25551601 1.57
## [440] {rolls/buns}               => {other vegetables}         0.04260295 0.23161968 0.18393493 1.19
## [441] {other vegetables}         => {rolls/buns}               0.04260295 0.22017867 0.19349263 1.19
## [442] {rolls/buns}               => {whole milk}               0.05663447 0.30790492 0.18393493 1.20
## [443] {whole milk}               => {rolls/buns}               0.05663447 0.22164743 0.25551601 1.20
## [444] {other vegetables}         => {whole milk}               0.07483477 0.38675775 0.19349263 1.51
## [445] {whole milk}               => {other vegetables}         0.07483477 0.29287704 0.25551601 1.51
## [446] {curd,
##        yogurt}                   => {whole milk}               0.01006609 0.58235294 0.01728521 2.27
## [447] {curd,
##        whole milk}               => {yogurt}                   0.01006609 0.38521401 0.02613116 2.76
## [448] {whole milk,
##        yogurt}                   => {curd}                     0.01006609 0.17967332 0.05602440 3.37
## [449] {other vegetables,
##        pork}                     => {whole milk}               0.01016777 0.46948357 0.02165735 1.83
## [450] {pork,
##        whole milk}               => {other vegetables}         0.01016777 0.45871560 0.02216573 2.37
## [451] {other vegetables,
##        whole milk}               => {pork}                     0.01016777 0.13586957 0.07483477 2.35
## [452] {butter,
##        other vegetables}         => {whole milk}               0.01148958 0.57360406 0.02003050 2.24
## [453] {butter,
##        whole milk}               => {other vegetables}         0.01148958 0.41697417 0.02755465 2.15
## [454] {other vegetables,
##        whole milk}               => {butter}                   0.01148958 0.15353261 0.07483477 2.77
## [455] {domestic eggs,
##        other vegetables}         => {whole milk}               0.01230300 0.55251142 0.02226741 2.16
## [456] {domestic eggs,
##        whole milk}               => {other vegetables}         0.01230300 0.41016949 0.02999492 2.11
## [457] {other vegetables,
##        whole milk}               => {domestic eggs}            0.01230300 0.16440217 0.07483477 2.59
## [458] {fruit/vegetable juice,
##        other vegetables}         => {whole milk}               0.01047280 0.49758454 0.02104728 1.94
## [459] {fruit/vegetable juice,
##        whole milk}               => {other vegetables}         0.01047280 0.39312977 0.02663955 2.03
## [460] {other vegetables,
##        whole milk}               => {fruit/vegetable juice}    0.01047280 0.13994565 0.07483477 1.93
## [461] {whipped/sour cream,
```

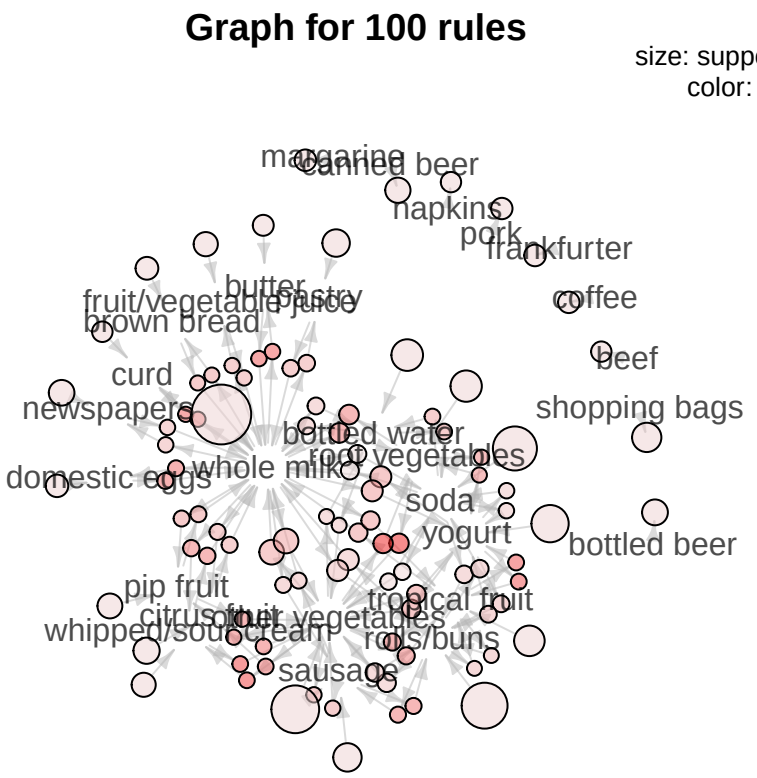
```
##        yogurt}                 => {other vegetables}        0.01016777 0.49019608 0.02074225 2.533
## [462] {other vegetables,
##        whipped/sour cream}     => {yogurt}                  0.01016777 0.35211268 0.02887646 2.524
## [463] {other vegetables,
##        yogurt}                 => {whipped/sour cream}      0.01016777 0.23419204 0.04341637 3.267
## [464] {whipped/sour cream,
##        yogurt}                 => {whole milk}              0.01087951 0.52450980 0.02074225 2.053
## [465] {whipped/sour cream,
##        whole milk}             => {yogurt}                  0.01087951 0.33753943 0.03223183 2.419
## [466] {whole milk,
##        yogurt}                 => {whipped/sour cream}      0.01087951 0.19419238 0.05602440 2.709
## [467] {other vegetables,
##        whipped/sour cream}     => {whole milk}              0.01464159 0.50704225 0.02887646 1.984
## [468] {whipped/sour cream,
##        whole milk}             => {other vegetables}        0.01464159 0.45425868 0.03223183 2.347
## [469] {other vegetables,
##        whole milk}             => {whipped/sour cream}      0.01464159 0.19565217 0.07483477 2.729
## [470] {other vegetables,
##        pip fruit}              => {whole milk}              0.01352313 0.51750973 0.02613116 2.025
## [471] {pip fruit,
##        whole milk}             => {other vegetables}        0.01352313 0.44932432 0.03009659 2.321
## [472] {other vegetables,
##        whole milk}             => {pip fruit}               0.01352313 0.18070652 0.07483477 2.388
## [473] {other vegetables,
##        pastry}                 => {whole milk}              0.01057448 0.46846847 0.02257245 1.833
## [474] {pastry,
##        whole milk}             => {other vegetables}        0.01057448 0.31804281 0.03324860 1.643
## [475] {other vegetables,
##        whole milk}             => {pastry}                  0.01057448 0.14130435 0.07483477 1.588
## [476] {citrus fruit,
##        root vegetables}        => {other vegetables}        0.01037112 0.58620690 0.01769192 3.029
## [477] {citrus fruit,
##        other vegetables}       => {root vegetables}         0.01037112 0.35915493 0.02887646 3.295
## [478] {other vegetables,
##        root vegetables}        => {citrus fruit}            0.01037112 0.21888412 0.04738180 2.644
## [479] {citrus fruit,
##        yogurt}                 => {whole milk}              0.01026945 0.47417840 0.02165735 1.855
## [480] {citrus fruit,
##        whole milk}             => {yogurt}                  0.01026945 0.33666667 0.03050330 2.413
## [481] {whole milk,
##        yogurt}                 => {citrus fruit}            0.01026945 0.18330309 0.05602440 2.214
## [482] {citrus fruit,
##        other vegetables}       => {whole milk}              0.01301474 0.45070423 0.02887646 1.763
## [483] {citrus fruit,
##        whole milk}             => {other vegetables}        0.01301474 0.42666667 0.03050330 2.205
## [484] {other vegetables,
##        whole milk}             => {citrus fruit}            0.01301474 0.17391304 0.07483477 2.101
## [485] {other vegetables,
##        sausage}                => {whole milk}              0.01016777 0.37735849 0.02694459 1.476
## [486] {sausage,
##        whole milk}             => {other vegetables}        0.01016777 0.34013605 0.02989324 1.757
## [487] {other vegetables,
##        whole milk}             => {sausage}                 0.01016777 0.13586957 0.07483477 1.446
## [488] {bottled water,
```

```
##        other vegetables}        => {whole milk}             0.01077783 0.43442623 0.02480935 1.700
## [489] {bottled water,
##        whole milk}              => {other vegetables}       0.01077783 0.31360947 0.03436706 1.620
## [490] {other vegetables,
##        whole milk}              => {bottled water}          0.01077783 0.14402174 0.07483477 1.303
## [491] {root vegetables,
##        tropical fruit}          => {other vegetables}       0.01230300 0.58454106 0.02104728 3.020
## [492] {other vegetables,
##        tropical fruit}          => {root vegetables}        0.01230300 0.34277620 0.03589222 3.144
## [493] {other vegetables,
##        root vegetables}         => {tropical fruit}         0.01230300 0.25965665 0.04738180 2.474
## [494] {root vegetables,
##        tropical fruit}          => {whole milk}             0.01199797 0.57004831 0.02104728 2.230
## [495] {tropical fruit,
##        whole milk}              => {root vegetables}        0.01199797 0.28365385 0.04229792 2.603
## [496] {root vegetables,
##        whole milk}              => {tropical fruit}         0.01199797 0.24532225 0.04890696 2.337
## [497] {tropical fruit,
##        yogurt}                  => {other vegetables}       0.01230300 0.42013889 0.02928317 2.171
## [498] {other vegetables,
##        tropical fruit}          => {yogurt}                 0.01230300 0.34277620 0.03589222 2.457
## [499] {other vegetables,
##        yogurt}                  => {tropical fruit}         0.01230300 0.28337237 0.04341637 2.700
## [500] {tropical fruit,
##        yogurt}                  => {whole milk}             0.01514997 0.51736111 0.02928317 2.024
## [501] {tropical fruit,
##        whole milk}              => {yogurt}                 0.01514997 0.35817308 0.04229792 2.567
## [502] {whole milk,
##        yogurt}                  => {tropical fruit}         0.01514997 0.27041742 0.05602440 2.577
## [503] {rolls/buns,
##        tropical fruit}          => {whole milk}             0.01098119 0.44628099 0.02460600 1.746
## [504] {tropical fruit,
##        whole milk}              => {rolls/buns}             0.01098119 0.25961538 0.04229792 1.411
## [505] {rolls/buns,
##        whole milk}              => {tropical fruit}         0.01098119 0.19389587 0.05663447 1.847
## [506] {other vegetables,
##        tropical fruit}          => {whole milk}             0.01708185 0.47592068 0.03589222 1.862
## [507] {tropical fruit,
##        whole milk}              => {other vegetables}       0.01708185 0.40384615 0.04229792 2.087
## [508] {other vegetables,
##        whole milk}              => {tropical fruit}         0.01708185 0.22826087 0.07483477 2.175
## [509] {root vegetables,
##        yogurt}                  => {other vegetables}       0.01291307 0.50000000 0.02582613 2.584
## [510] {other vegetables,
##        root vegetables}         => {yogurt}                 0.01291307 0.27253219 0.04738180 1.953
## [511] {other vegetables,
##        yogurt}                  => {root vegetables}        0.01291307 0.29742389 0.04341637 2.728
## [512] {root vegetables,
##        yogurt}                  => {whole milk}             0.01453991 0.56299213 0.02582613 2.203
## [513] {root vegetables,
##        whole milk}              => {yogurt}                 0.01453991 0.29729730 0.04890696 2.131
## [514] {whole milk,
##        yogurt}                  => {root vegetables}        0.01453991 0.25952813 0.05602440 2.381
## [515] {rolls/buns,
```

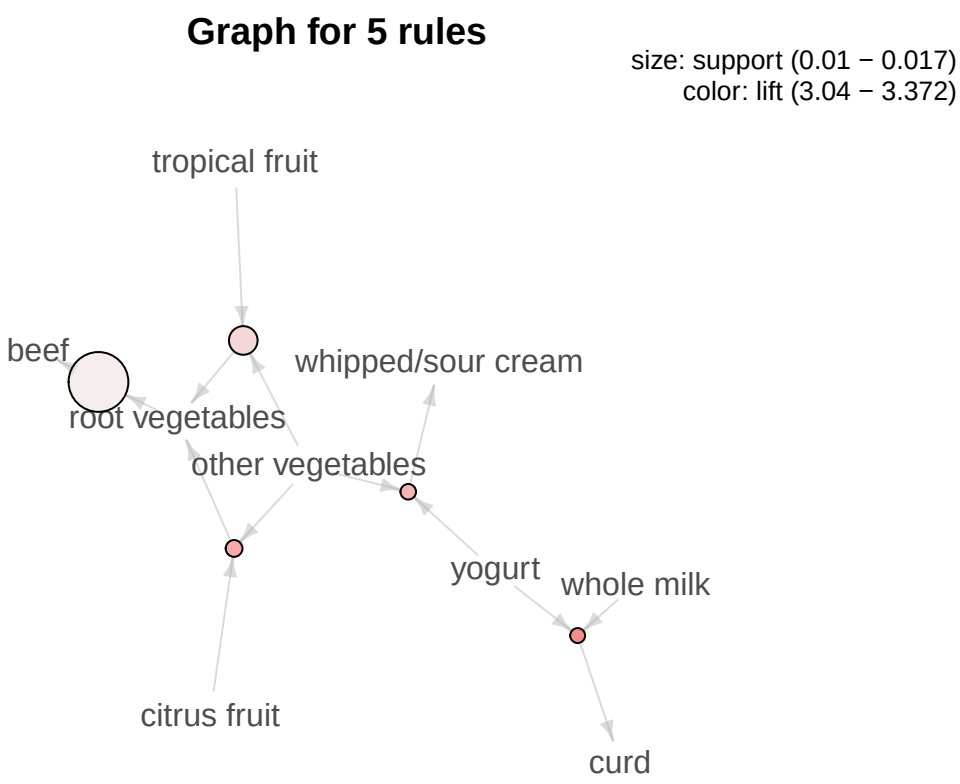
```
##        root vegetables}         => {other vegetables}         0.01220132 0.50209205 0.02430097 2.594
## [516] {other vegetables,
##        root vegetables}         => {rolls/buns}               0.01220132 0.25751073 0.04738180 1.400
## [517] {other vegetables,
##        rolls/buns}              => {root vegetables}          0.01220132 0.28639618 0.04260295 2.627
## [518] {rolls/buns,
##        root vegetables}         => {whole milk}               0.01270971 0.52301255 0.02430097 2.046
## [519] {root vegetables,
##        whole milk}              => {rolls/buns}               0.01270971 0.25987526 0.04890696 1.413
## [520] {rolls/buns,
##        whole milk}              => {root vegetables}          0.01270971 0.22441652 0.05663447 2.058
## [521] {other vegetables,
##        root vegetables}         => {whole milk}               0.02318251 0.48927039 0.04738180 1.914
## [522] {root vegetables,
##        whole milk}              => {other vegetables}         0.02318251 0.47401247 0.04890696 2.449
## [523] {other vegetables,
##        whole milk}              => {root vegetables}          0.02318251 0.30978261 0.07483477 2.842
## [524] {soda,
##        yogurt}                  => {whole milk}               0.01047280 0.38289963 0.02735130 1.498
## [525] {soda,
##        whole milk}              => {yogurt}                   0.01047280 0.26142132 0.04006101 1.873
## [526] {whole milk,
##        yogurt}                  => {soda}                     0.01047280 0.18693285 0.05602440 1.071
## [527] {other vegetables,
##        soda}                    => {whole milk}               0.01392984 0.42546584 0.03274021 1.665
## [528] {soda,
##        whole milk}              => {other vegetables}         0.01392984 0.34771574 0.04006101 1.797
## [529] {other vegetables,
##        whole milk}              => {soda}                     0.01392984 0.18614130 0.07483477 1.067
## [530] {rolls/buns,
##        yogurt}                  => {other vegetables}         0.01148958 0.33431953 0.03436706 1.727
## [531] {other vegetables,
##        yogurt}                  => {rolls/buns}               0.01148958 0.26463700 0.04341637 1.438
## [532] {other vegetables,
##        rolls/buns}              => {yogurt}                   0.01148958 0.26968974 0.04260295 1.933
## [533] {rolls/buns,
##        yogurt}                  => {whole milk}               0.01555669 0.45266272 0.03436706 1.771
## [534] {whole milk,
##        yogurt}                  => {rolls/buns}               0.01555669 0.27767695 0.05602440 1.509
## [535] {rolls/buns,
##        whole milk}              => {yogurt}                   0.01555669 0.27468582 0.05663447 1.969
## [536] {other vegetables,
##        yogurt}                  => {whole milk}               0.02226741 0.51288056 0.04341637 2.007
## [537] {whole milk,
##        yogurt}                  => {other vegetables}         0.02226741 0.39745917 0.05602440 2.054
## [538] {other vegetables,
##        whole milk}              => {yogurt}                   0.02226741 0.29755435 0.07483477 2.133
## [539] {other vegetables,
##        rolls/buns}              => {whole milk}               0.01789527 0.42004773 0.04260295 1.643
## [540] {rolls/buns,
##        whole milk}              => {other vegetables}         0.01789527 0.31597846 0.05663447 1.633
## [541] {other vegetables,
##        whole milk}              => {rolls/buns}               0.01789527 0.23913043 0.07483477 1.300
```

```
plot(asso_rules3, method='graph')
```

```
## Warning: plot: Too many rules supplied. Only plotting the best 100 rules using
## 'support' (change control parameter max if needed)
```




```
plot(head(asso_rules3, 5, by='lift'), method='graph')
```



Conclusion

From the association rule and the visualization graph, we get the following conclusions: 1. Bottled beer has high association with liquor and red/blush wine. People tend to buy bottled beer more if they buy the other two. 2. People are more likely to buy vegetables if they buy pip fruit, ham, fruit/vege juice and onions. 3. Whole milk has not much association with liquor or beer and is the most often bought item.