Title: STA 380, Part 2: Exercises
Author: Ankit Muthiyan, Disha Gandhi, Kshitij Mahajan
Date: 08/12/2022
Github link:https://github.com/KshitijMahajan1998/STA-380-Part-2-Assignment-Disha-Ankit-Kshitij

Q1 ~ Probability practice

Question 1(A)

P(Rc) = 0.3 => P(Tc) = 0.7

Rc and Tc are mutually exclusive and exhaustive

P('yes'| Rc) = P('no'| Rc) = 0.5

p('yes') = 0.65

Let P('yes'|Tc) = x

Using the rule of total probability,

P('yes') = P('yes',Rc) + P ('yes',Tc) 0.65 = (o.3 * o.5) + (0.7 *x) #Using Baye's Rule 0.65 - 0.15 = 0.7x x= 5/7 = 71.4 %

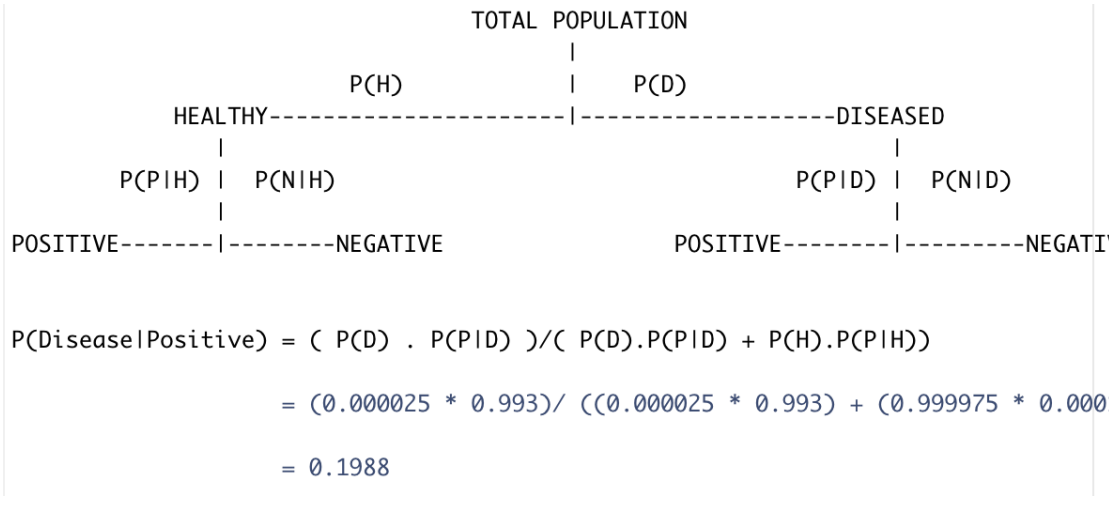
Question 1(B)

P(Positive|Disease) = 0.993 => P(Negative|Disease) = 0.007

P(Positive|Healthy) = 0.0001 ; P(Negative| Healthy) = 0.9999

P(Disease)=0.000025 ; P(Healthy)= 0.999975

Drawing a tree chart



$$P(Disease|Positive) = (P(D) . P(P|D))/(P(D).P(P|D) + P(H).P(P|H))$$

$$= (0.000025 * 0.993)/ ((0.000025 * 0.993) + (0.999975 * 0.0001))$$

$$= 0.1988$$

However, here we are assuming that the entire population underwent the testing- which is impossible. Hence, we are overestimating the false positives in the denominator.

Hence, the actual P(Disease|Positive) will likely be greater than 0.1988

Q2 ~ Wrangling the Billboard Top 100

PART A As per the ask of the question we are supposed to generate a table displaying the top 10 songs as measured by the number of weeks it was present on billboard. The first column presents **song name**, second column displays **performer name**, and third column displays the **number of weeks that song was present on the billboard**. We have taken into consideration that the combination of **song and performer should be unique** and the **count is displayed in descending order**.

```
##
## 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

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

## # A tibble: 29,356 x 3
## # Groups:   song [24,360]
##    song                                 performer                          count
##    <chr>                                <chr>                              <int>
##  1 Radioactive                          Imagine Dragons                       87
##  2 Sail                                 AWOLNATION                            79
##  3 Blinding Lights                      The Weeknd                            76
##  4 I'm Yours                            Jason Mraz                            76
##  5 How Do I Live                        LeAnn Rimes                           69
##  6 Counting Stars                       OneRepublic                           68
##  7 Party Rock Anthem                    LMFAO Featuring Lauren Bennett & G~    68
##  8 Foolish Games/You Were Meant For Me  Jewel                                 65
##  9 Rolling In The Deep                  Adele                                 65
## 10 Before He Cheats                     Carrie Underwood                      64
## # ... with 29,346 more rows
## # i Use `print(n = ...)` to see more rows
```

STEPS FOLLOWED IN THE ORDER:

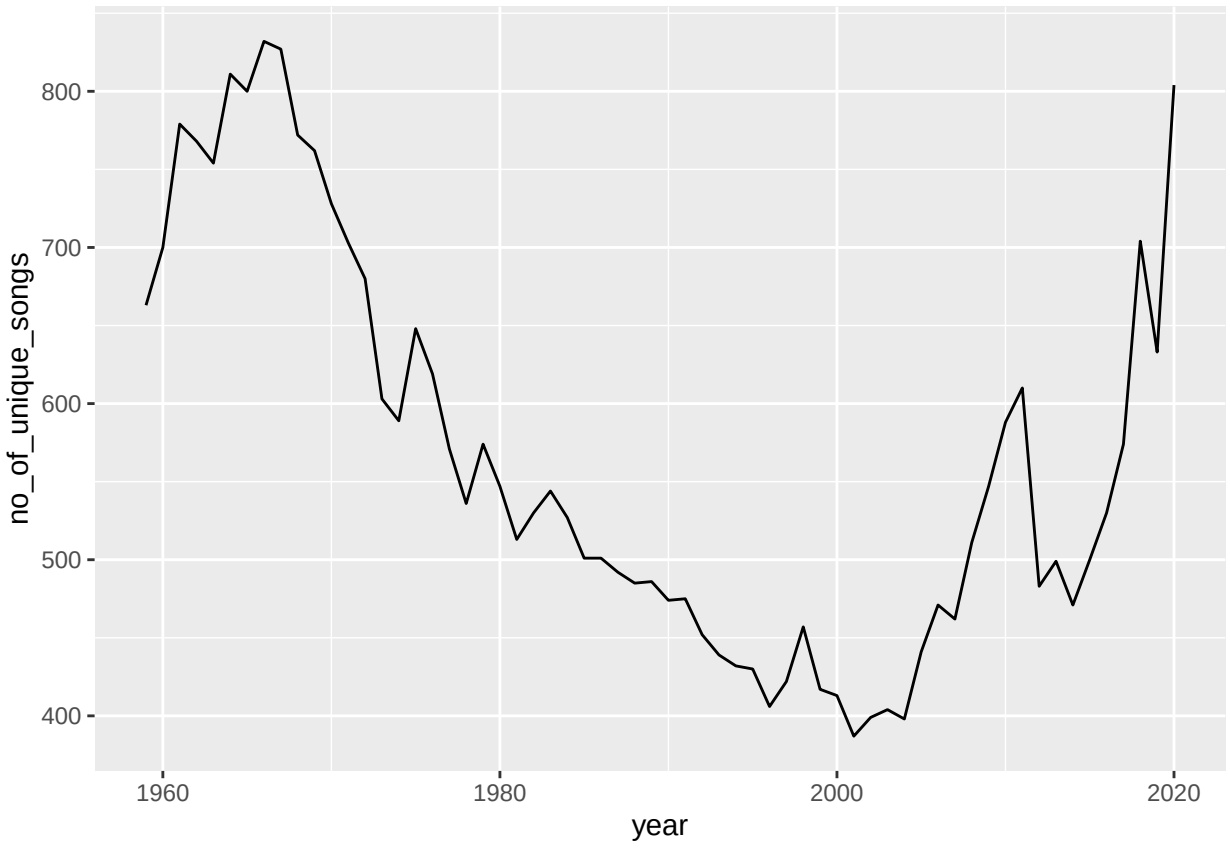
1) Loading Dplyr Library
2) Reading the csv and converting to df
3) Performing GroupBy on song and performer and then taking a count to find total presence on billboard for each song
4) Arranging the count in descending order
5) Displaying the top 10 rows from above table

PART B As per the ask of the question, we are supposed to report the musical diversity of the billboard for each year. That implies reporting the **unique songs that appeared on the billboard top 100 for each year**. We can see that second column denotes exactly that for each year.

```
## # A tibble: 62 x 2
##     year no_of_unique_songs
##    <int>              <int>
##  1  1959                663
##  2  1960                700
##  3  1961                779
##  4  1962                768
##  5  1963                754
##  6  1964                811
##  7  1965                800
##  8  1966                832
##  9  1967                827
## 10  1968                772
## # ... with 52 more rows
## # i Use 'print(n = ...)' to see more rows
```

STEPS FOLLOWED IN THE ORDER:

1) Remove songs from 1958 and 2021
2) Loading Dplyr Library
3) GroupBy on year and take a unique count of the songs
4) Plot the unique song count over the years on x axis



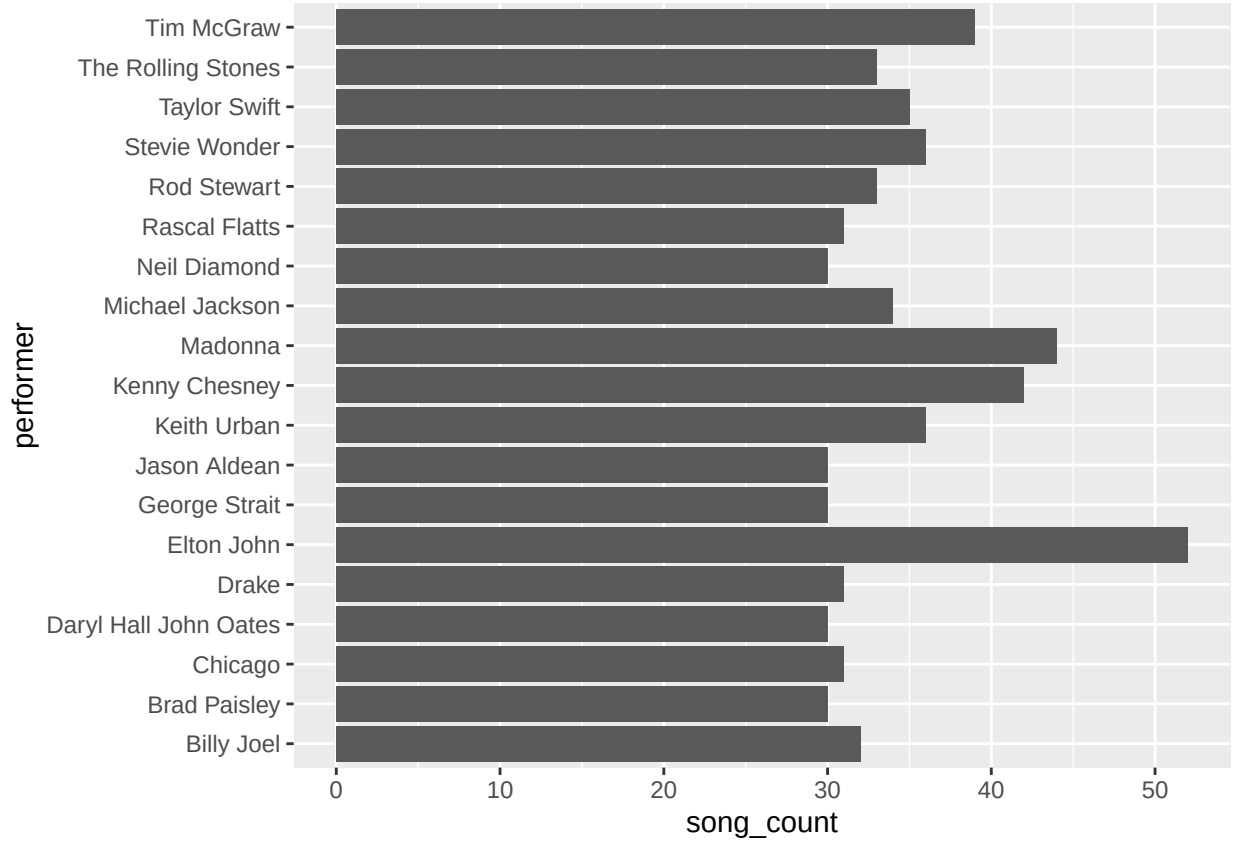
- We can see from the graph above that there is a **dip in the number of fresh songs in a year starting from the 80's till the previous decade from today**. That's strange because 80's gave us some pretty good songs to dance :) None the less as the data shows, **the dip is very gradual and goes to its lowest in 2001-2002** and takes a **sharp rise later to just increase but is again followed by a sharp decline in 2012-2013**.
- That can be possible because **billboard top 100 was launched in 50's** and during that time there were very **less artists dominating the inudustry and more of struggling artists**. Eventually with 2000's many well known and **talented artists starting coming up because of being found and backed up by top producers** and music industry was shaping up and people were idolizing artists. This might have led to the **increase in diversity**.

PART C As asked in the question, this plot is to tell us which artists have a 10 week hit implying they had their song on the billboard for atleast 10 weeks. Out of these artists there are 19 artists that have atleast 30 songs in the 10 week hit category. So what we see in the table and in the graph are those 19 artists along with **the number of songs for each that satisfy the two conditions of first being a ten week hit and second the artist having atleast 30 such songs**.

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

```
## # A tibble: 19 x 2
##    performer             song_count
##    <chr>                      <int>
##  1 Billy Joel                    32
```

```
##  2 Brad Paisley                30
##  3 Chicago                     31
##  4 Daryl Hall John Oates       30
##  5 Drake                       31
##  6 Elton John                  52
##  7 George Strait               30
##  8 Jason Aldean                30
##  9 Keith Urban                 36
## 10 Kenny Chesney               42
## 11 Madonna                     44
## 12 Michael Jackson             34
## 13 Neil Diamond                30
## 14 Rascal Flatts               31
## 15 Rod Stewart                 33
## 16 Stevie Wonder               36
## 17 Taylor Swift                35
## 18 The Rolling Stones          33
## 19 Tim McGraw                  39
```



STEPS FOLLOWED IN THE ORDER:

1) Grouping by the df with song and performer, applying a summarise on the unique count and filtering the unique combination with count atleast 10.
2) Grouping by the df obtained above by artist and then summarising further the count obtained in above step. Post which we apply a filter that the new summarise count by artist should be atleast 30. 3)The

19 artists obtained after this filtering are plotted on a bar graph and inverted with the axis for clarity. The song count is visible on the x axis

Q3 ~ Visual story telling part 1: green buildings

We do not agree with the findings of on-staff stats guru.

Though guru's inferences sound rational, he just took an average of the rent of non-green buildings and green buildings to then forecast premium rent rates of green buildings. He completely disregarded the 21 other variables that could have been used.

Simple Linear Regression

```
##                Estimate Std. Error  t value     Pr(>|t|)
## (Intercept)   0.7119434  0.1156967 6.153532 7.947381e-10
## green_rating1 2.4124581  0.3927573 6.142363 8.524150e-10
```

Coefficient of Green_Rating: 2.4125; P-Value: 8e-10

This model has an extremely high bias and has a very low R^2. This indicates that this model is fairly inadequate.

Now we will check whether any cofounding variables are present or not.

```
## 1 ) Multiple Linear Regression, green-building rating with CS_PropertyID : Non Confounding
## 2 ) Multiple Linear Regression, green-building rating with cluster : Non Confounding
## 3 ) Multiple Linear Regression, green-building rating with size : Non Confounding
## 4 ) Multiple Linear Regression, green-building rating with empl_gr : Non Confounding
## 5 ) Multiple Linear Regression, green-building rating with leasing_rate : Non Confounding
## 6 ) Multiple Linear Regression, green-building rating with stories : Non Confounding
## 7 ) Multiple Linear Regression, green-building rating with age : Non Confounding
## 8 ) Multiple Linear Regression, green-building rating with renovated : Non Confounding
## 9 ) Multiple Linear Regression, green-building rating with class_a : Confounding
## 10 ) Multiple Linear Regression, green-building rating with class_b : Non Confounding
## 11 ) Multiple Linear Regression, green-building rating with net : Non Confounding
## 12 ) Multiple Linear Regression, green-building rating with amenities : Non Confounding
## 13 ) Multiple Linear Regression, green-building rating with cd_total_07 : Non Confounding
## 14 ) Multiple Linear Regression, green-building rating with hd_total07 : Non Confounding
## 15 ) Multiple Linear Regression, green-building rating with total_dd_07 : Non Confounding
## 16 ) Multiple Linear Regression, green-building rating with Precipitation : Non Confounding
## 17 ) Multiple Linear Regression, green-building rating with Gas_Costs : Non Confounding
## 18 ) Multiple Linear Regression, green-building rating with Electricity_Costs : Non Confounding
```

From these models, we can conclude that whether a building is in **Class_A** category is an underlying confounding variable with the *Rent* and *Green Ratings*

This is rational.Class_A property is the most desirable property.It had an upper hand when compared with another properties in the cluster in terms of amenities, services, and technological aspects. Hence, Class_A listings will obviously cost more.All these factors combined will result in the increase in price and also enhances the chances of property qualifying as a green building.

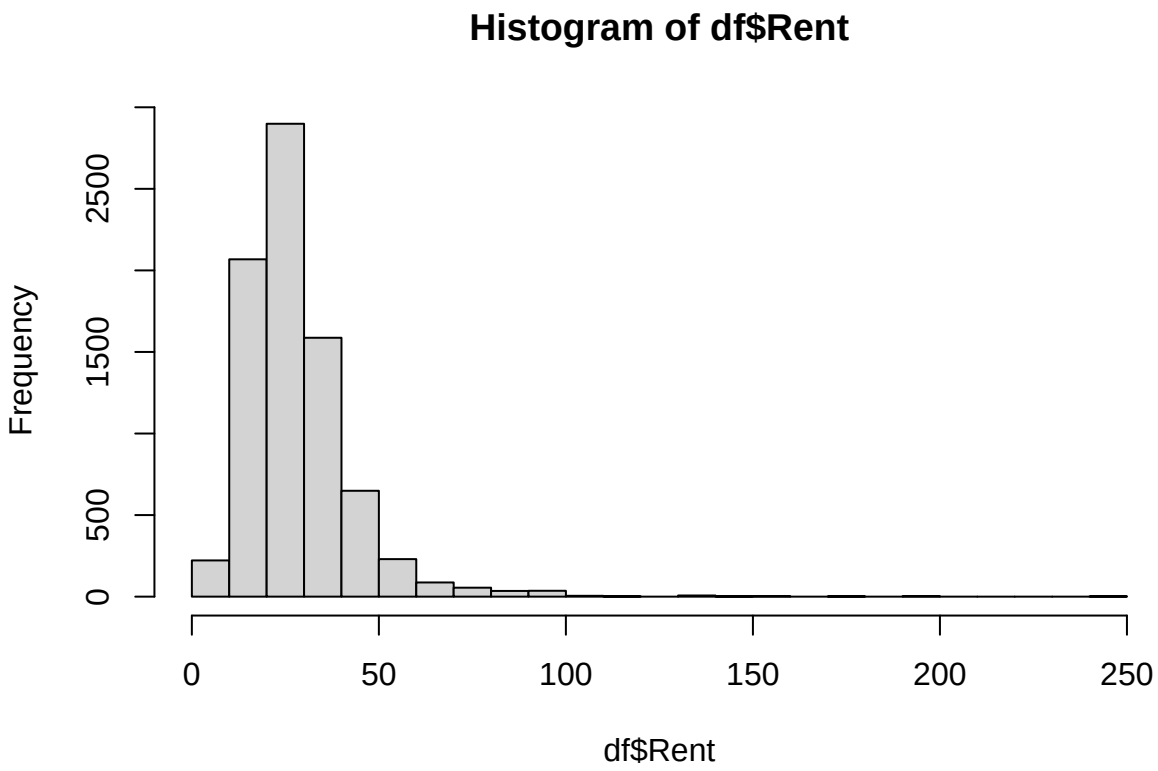
Data Pre-processing and Visualization of Data Distribution

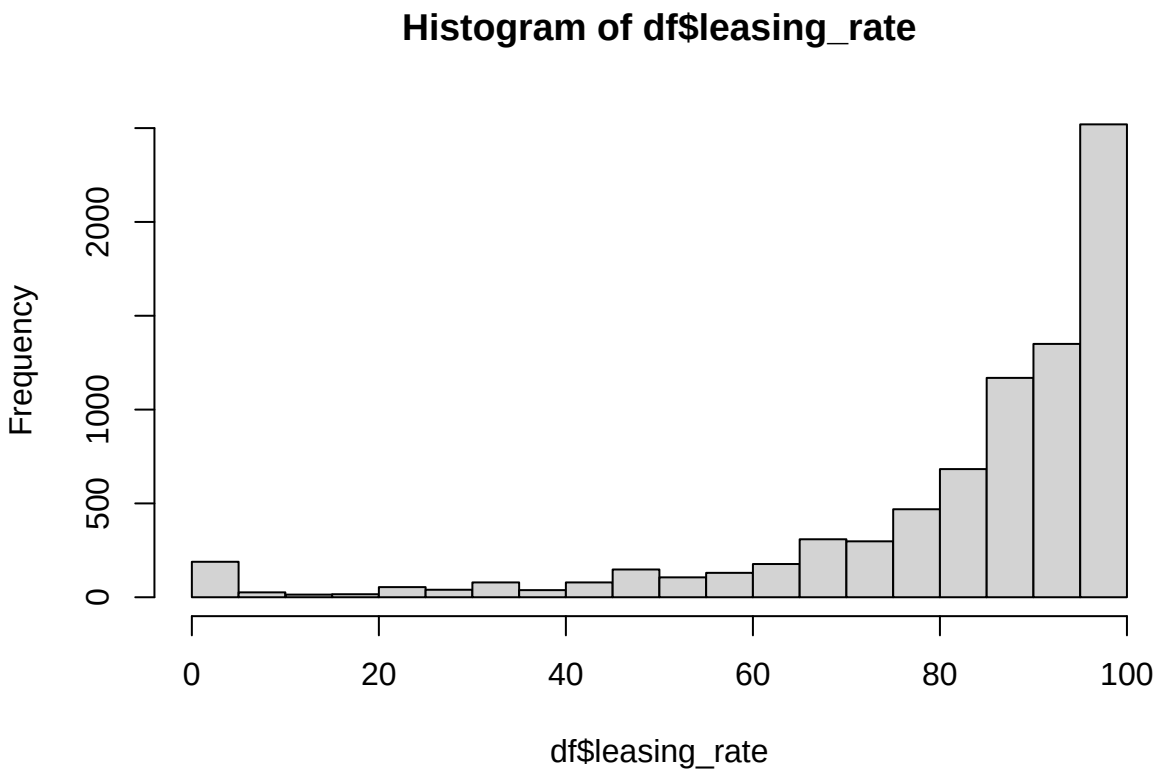
A few columns were added to the dataframe to combine multiple redundant columns.Binary continous values were converted to categorical values.

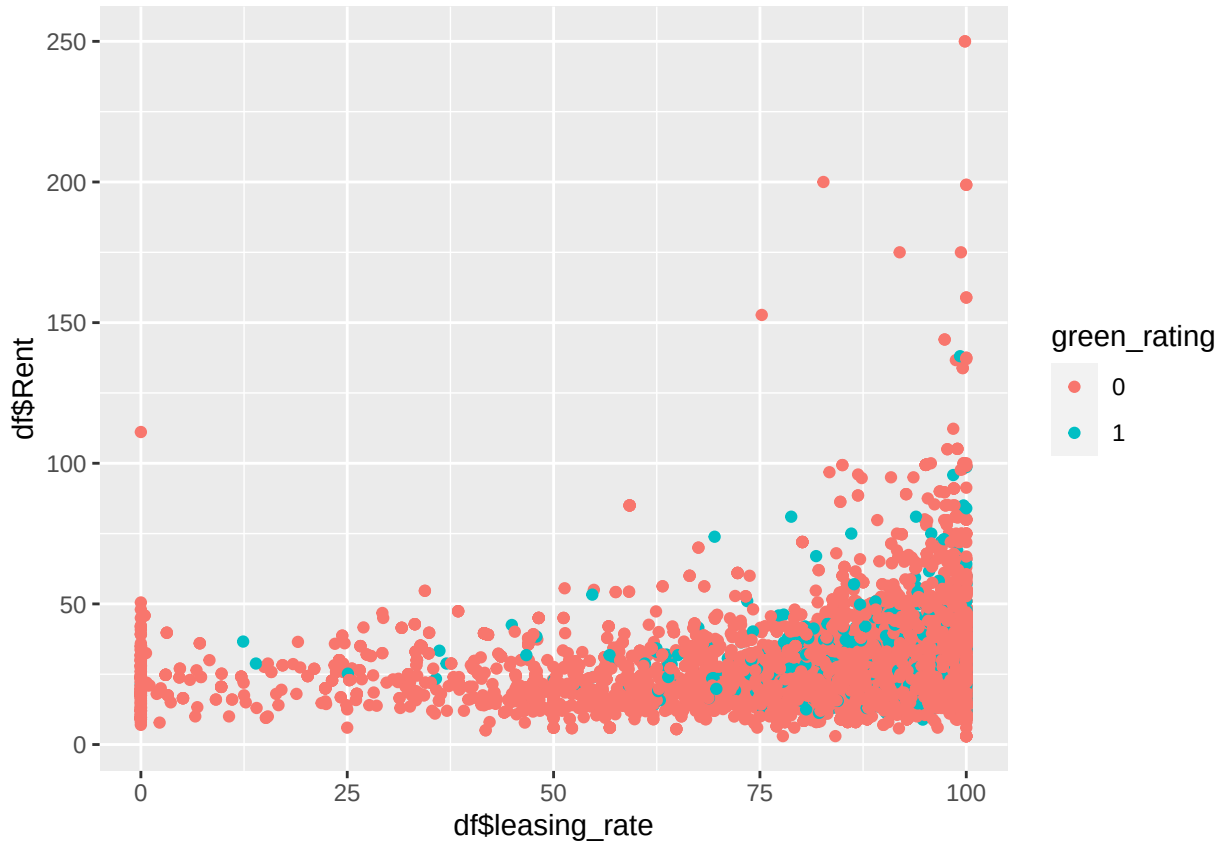
First, several sets of varibles were considered. Then their correlations with the rent and their effect on the rent was observed by plotting different graphs.Secondly, Few graphs were plotted to see the overall distribution of data.

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v tibble  3.1.8     v purrr   0.3.4
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## 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.
##
##
## Attaching package: 'mosaic'
##
##
## The following object is masked from 'package:Matrix':
##
##     mean
##
##
## The following object is masked from 'package:purrr':
##
##     cross
##
##
## 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
```




Histogram of df$leasing_rate
Frequency
0
500
1000
2000
0
20
40
60
80
100
df$leasing_rate

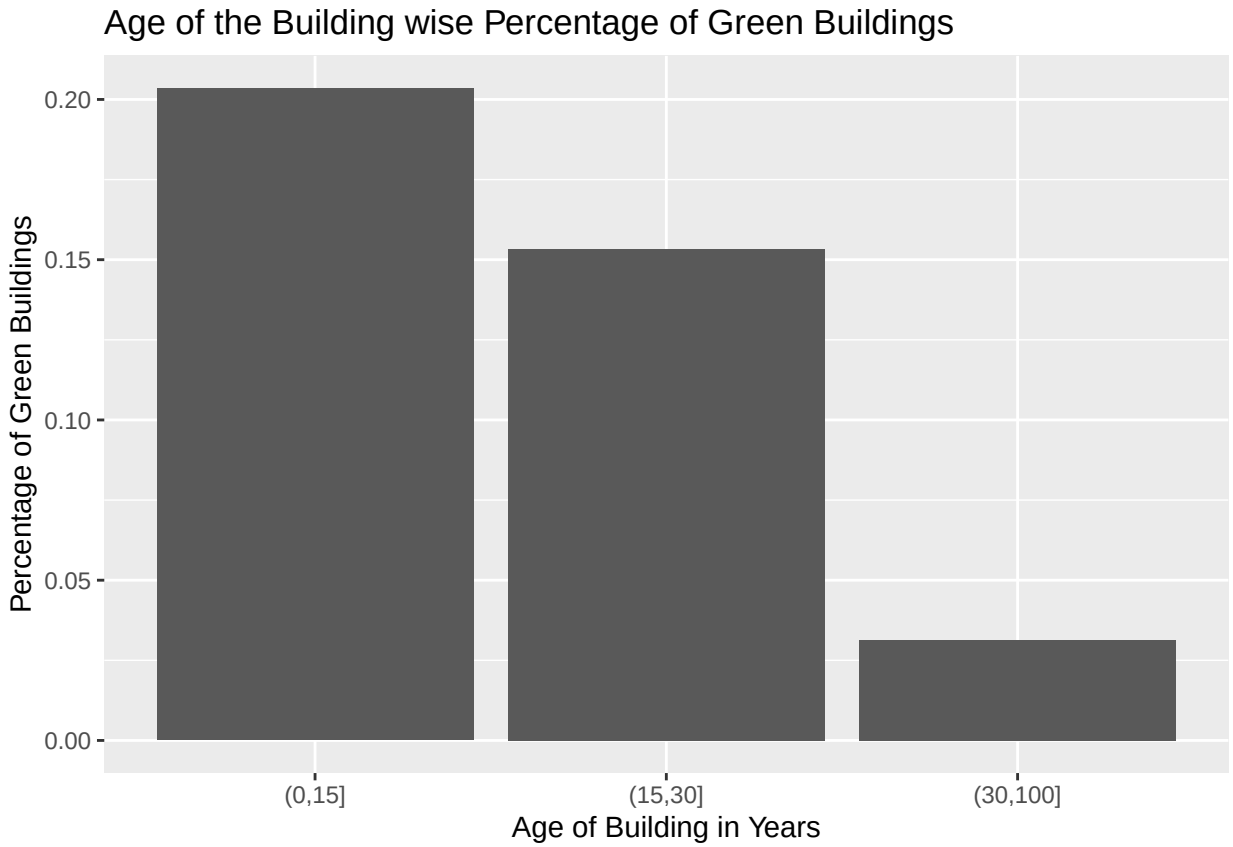


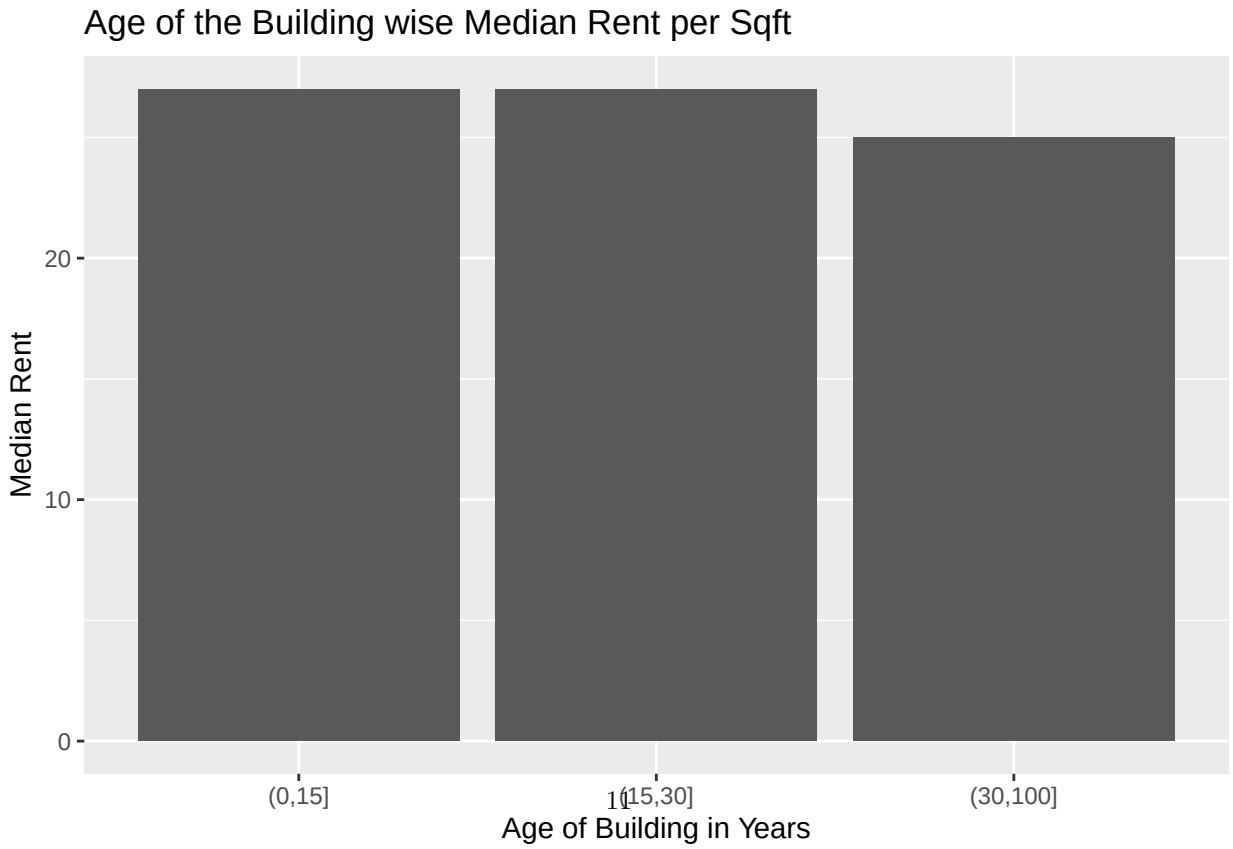
Convert continous variables into categorical variables.

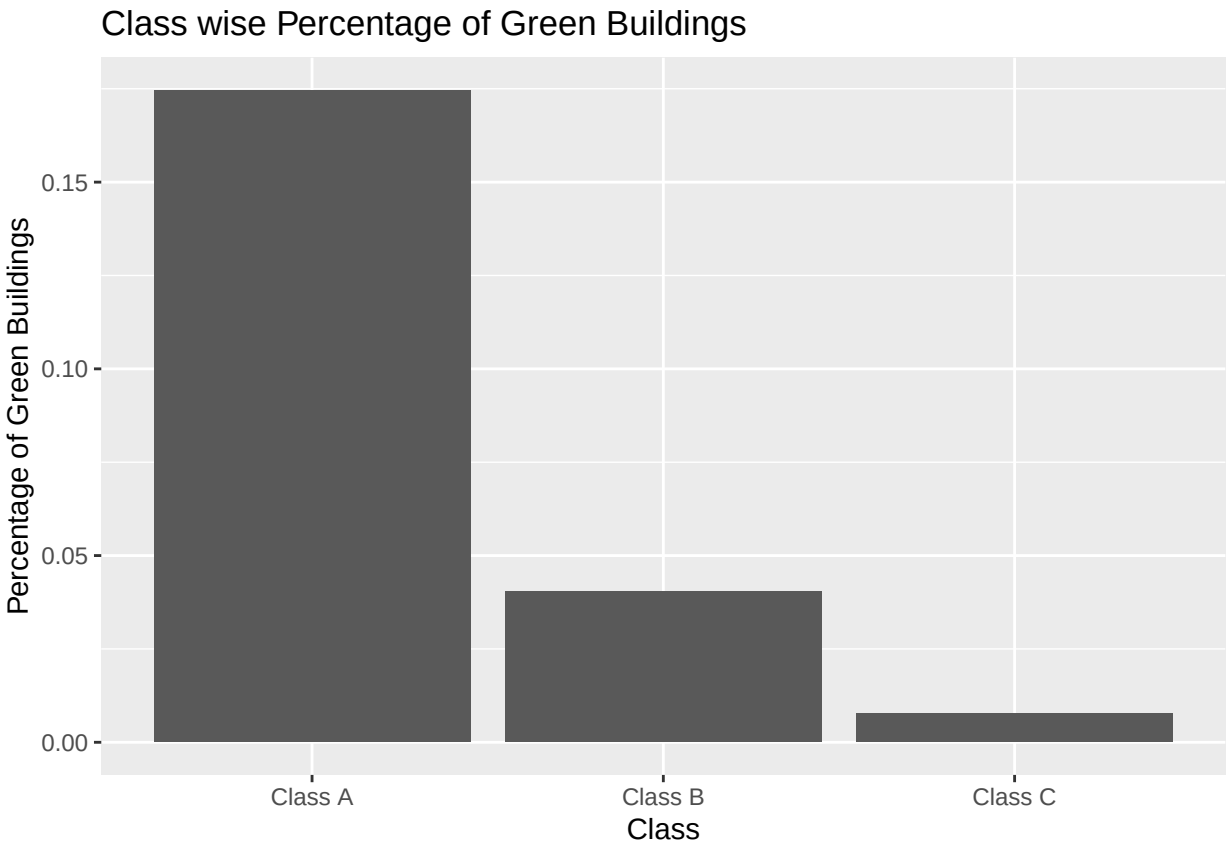
Convert continous variables such as Age, electricity price and the total degree days so that we can use them for visualization part.

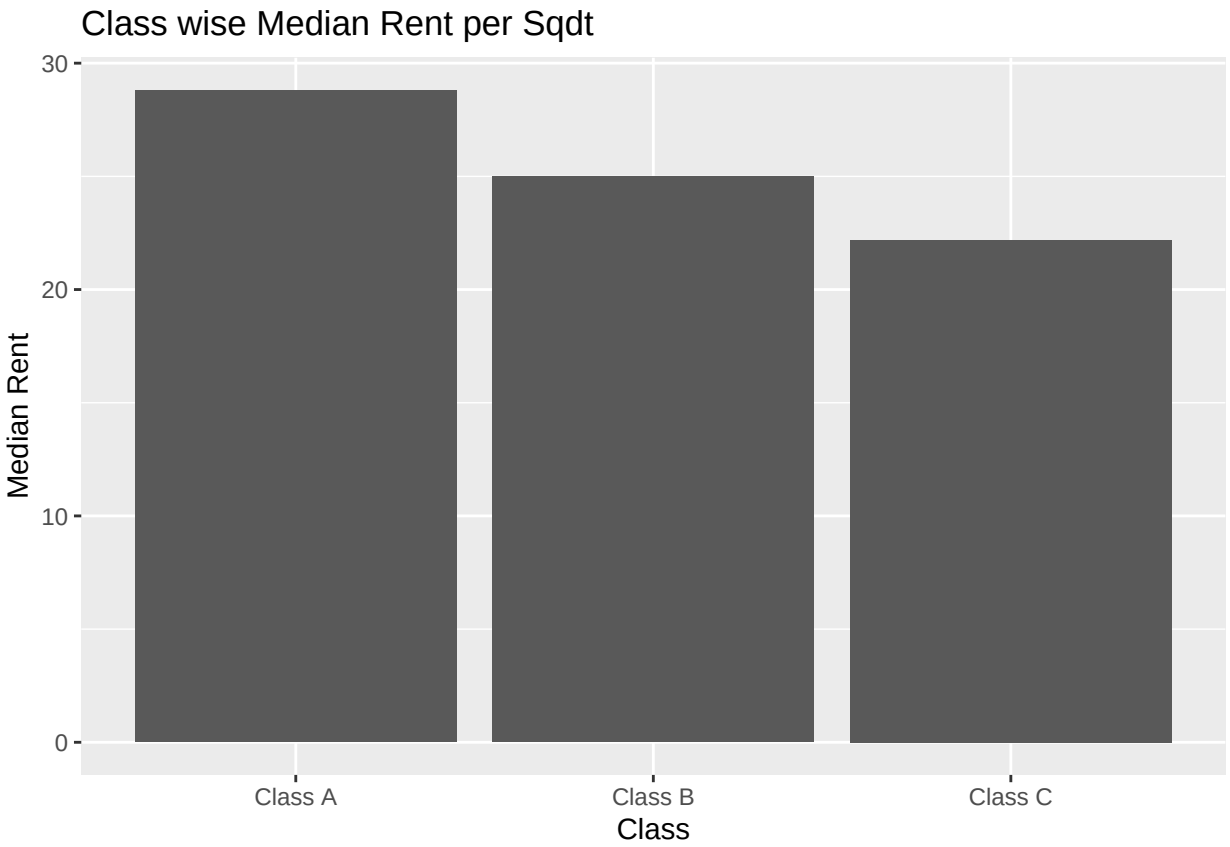
Data Analysis and Visualization

Distribution of Green Buidlings across different categories

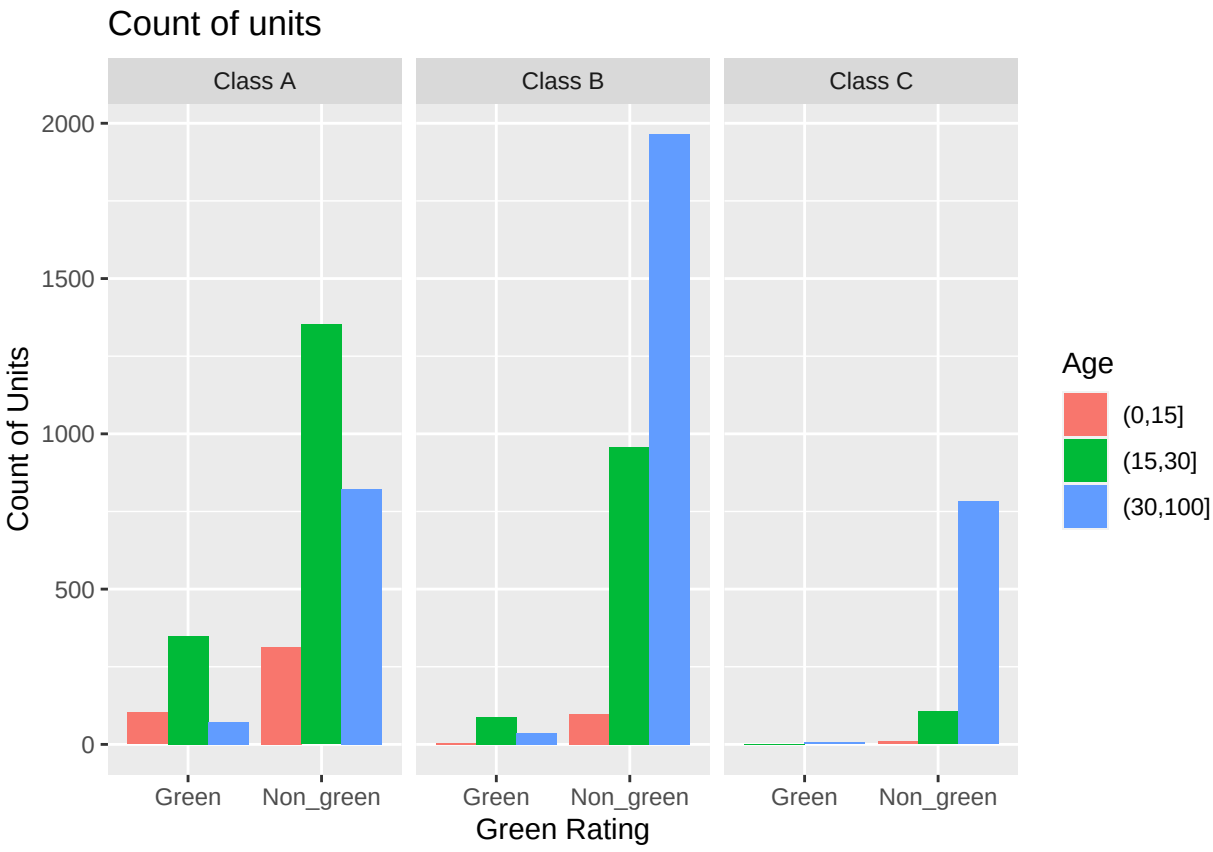


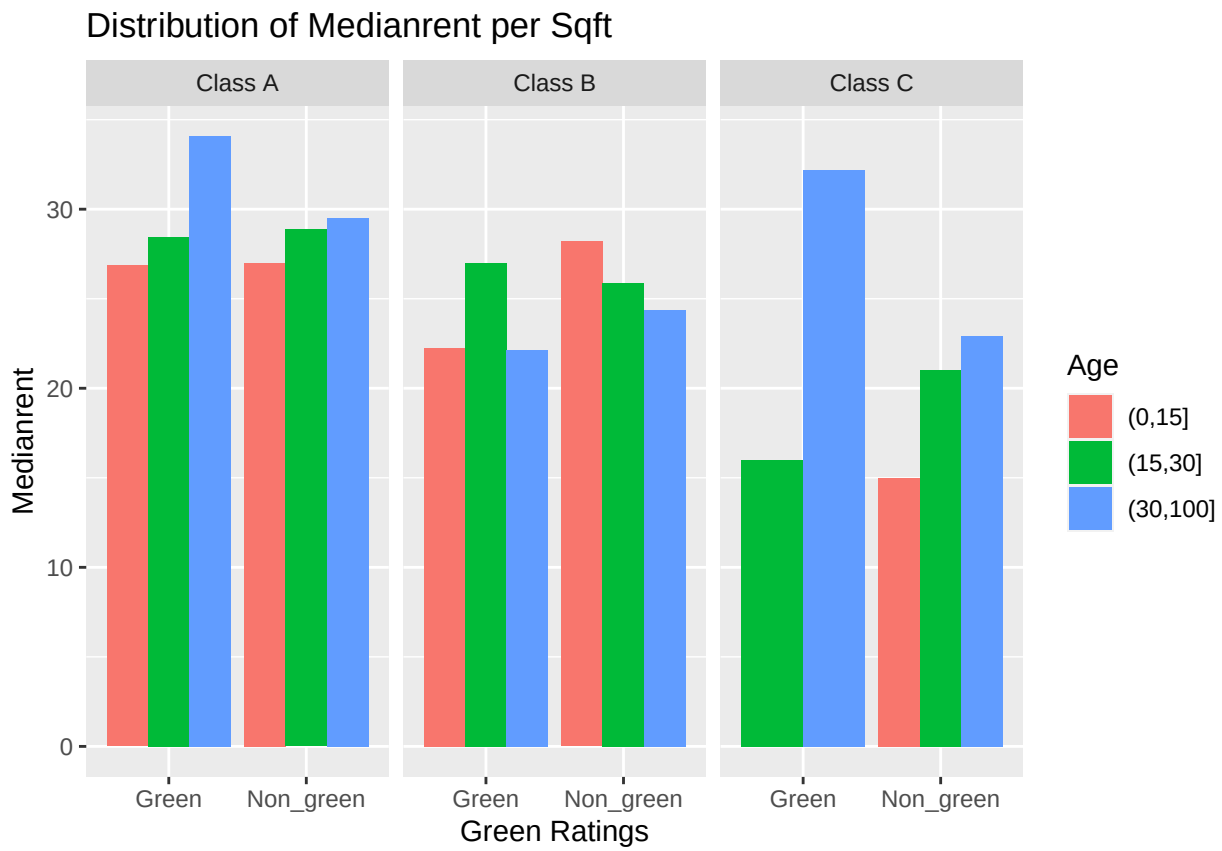



Class wise Percentage of Green Buildings
Percentage of Green Buildings
0.15
0.10
0.05
0.00
Class A
Class B
Class C
Class




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


Count of units
Class A
Class B
Class C
2000
1500
1000
500
0
Count of Units
Green
Non_green
Green Rating
Age
(0,15]
(15,30]
(30,100]

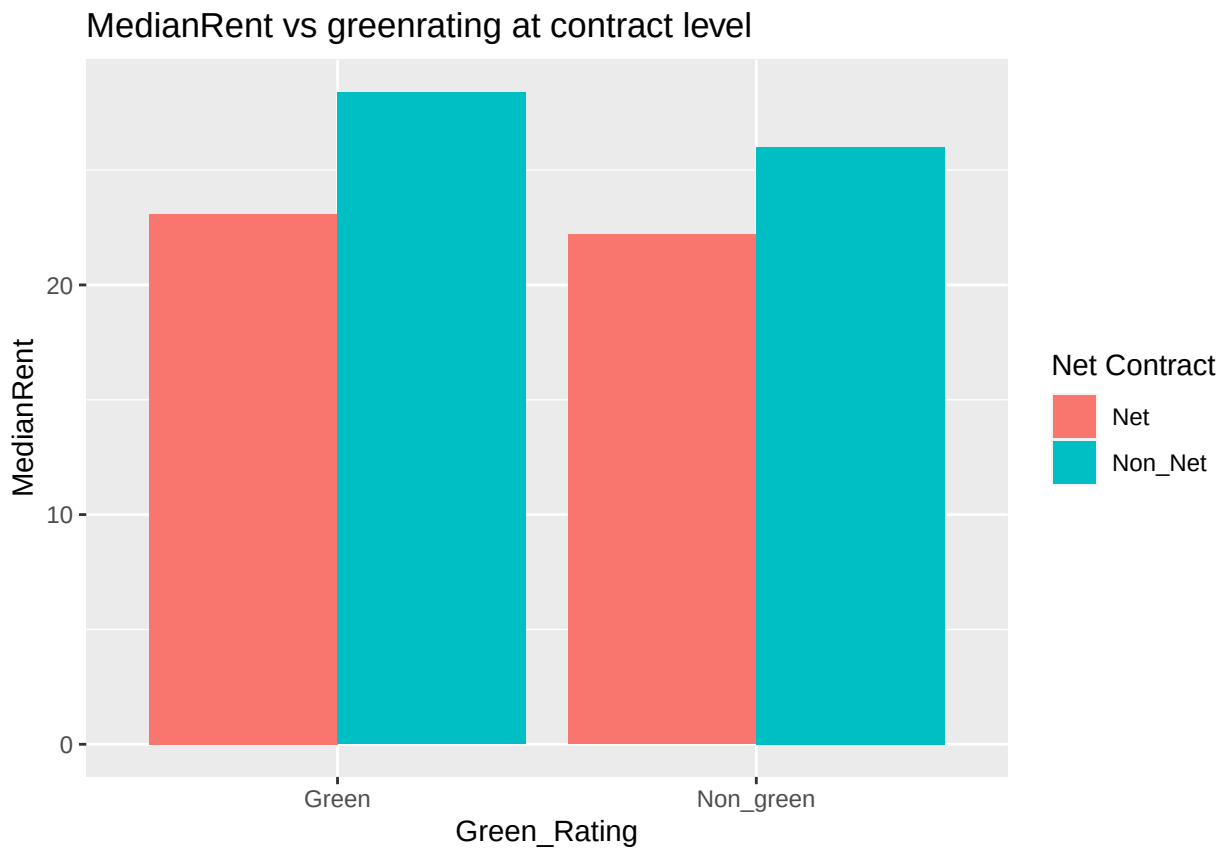


From the graphs it can be concluded that there is no considerable increase in Median rent for the non-green building to green building for Classes A and B across all age groups. This observation is sufficient to negate the conclusion of stats guru. Though his observation was right, it was due to higher percentage of Class A buildings and recent buildings in green rated buildings than solely because of it being a green building. It is recommended that stats guru should sub-divide the analysis across the class of building and age group so that he can improve his flawed suggestion.

Net Contract Level Analysis

We analysed further to see if there is any relation between the contract type charges and the green buildings.

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

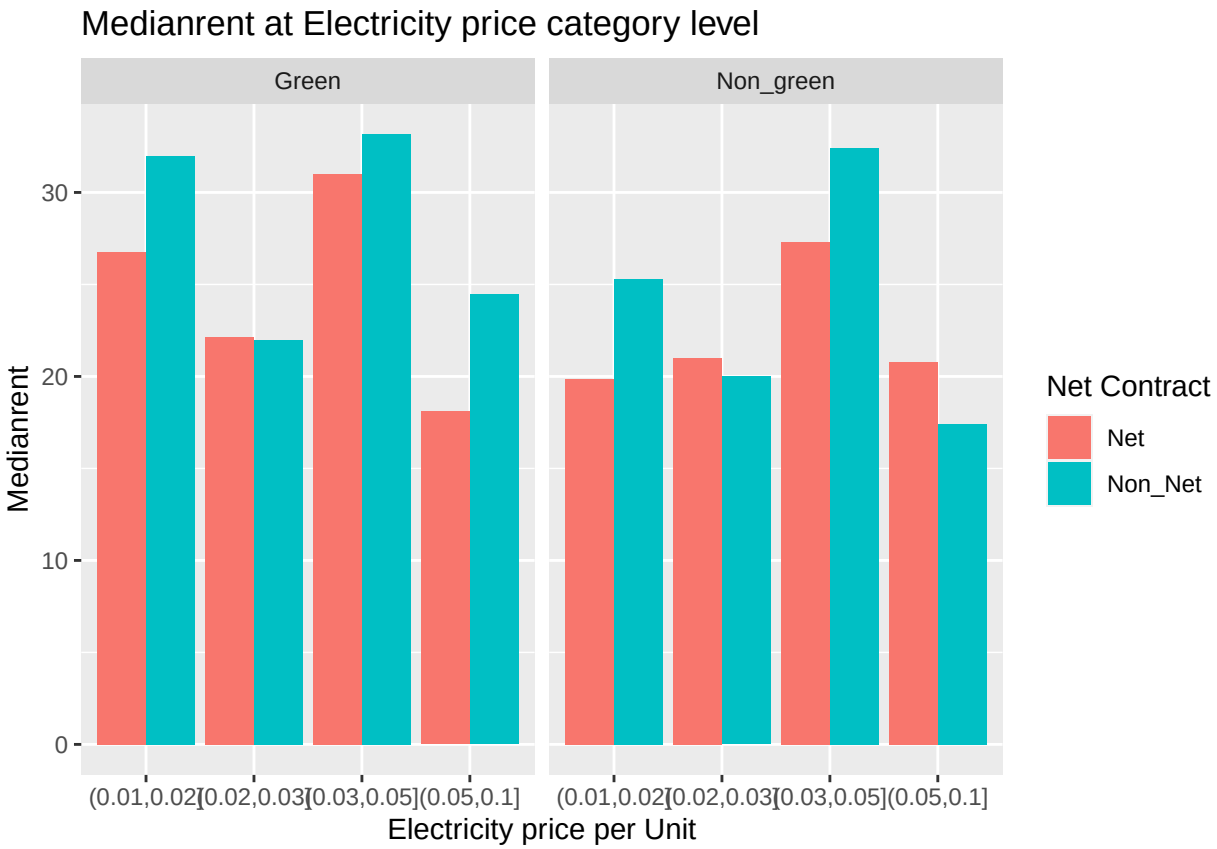


In the case of tenants availing net-rental contracts,they end up paying the same median rent for both green and non-green building. The underlying assumption that green buildings should have lesser difference between net rental contacts and non net rental contracts, being that green building reduce the consumption power accounting to thier ecofriendly design,was not observed. Rather this difference was more significant in the case of green buildings.

Net Contract Level Analysis with relation to Electricity Prices

We plotted below plots to explore an option of non net contract offering.This can help in the coming years in competitive pricing as the electricity and gas prices hike.To add to this, green building usually consumes lesser electricity and gas.

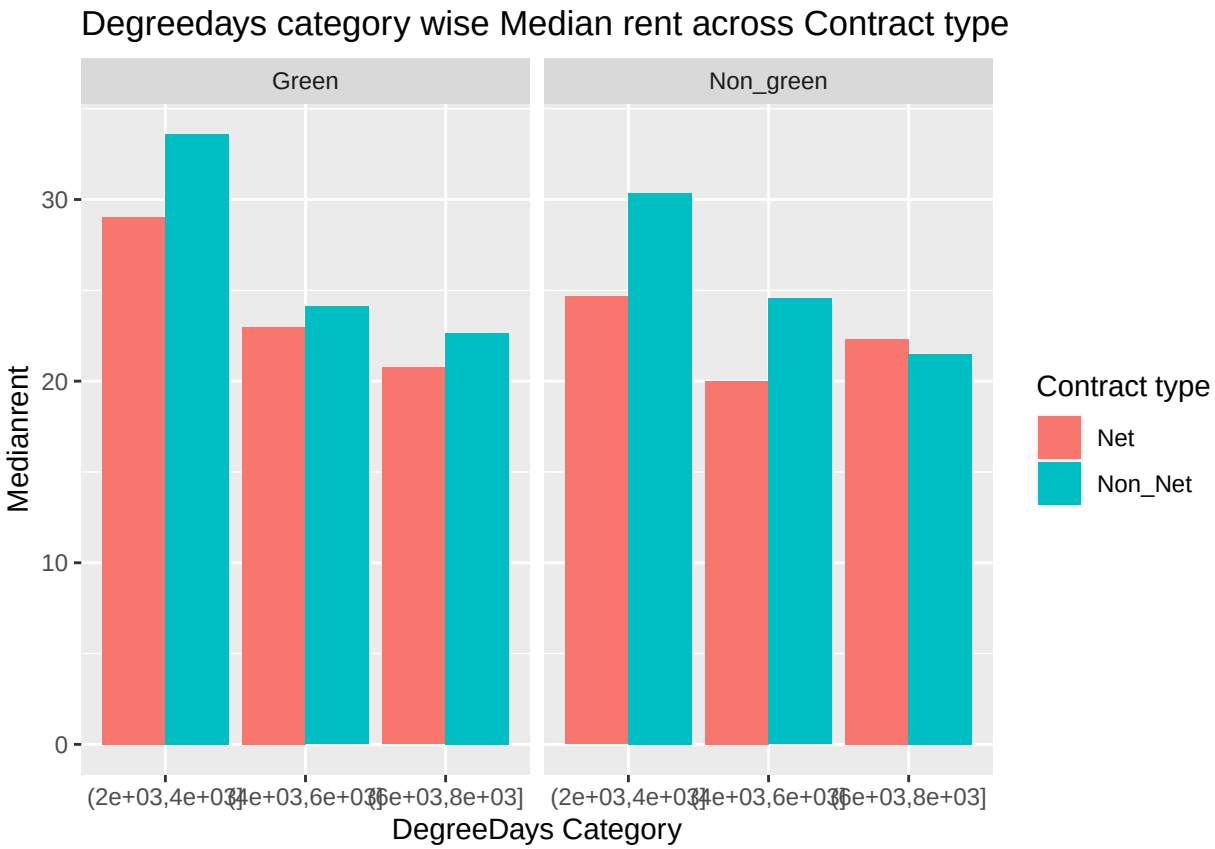
```
## `summarise()` has grouped output by 'green_rating', 'net'. You can override
## using the `.groups` argument.
```



The aim was to compare Net Contract to Non Net contract and see if there is any correlation between the electricity charges with the difference between the net contract and non net contract type. The difference of rent between the netcontract and non net contract type is higher for Non_green buildings [Electricty bracket of 0.03-0.05]

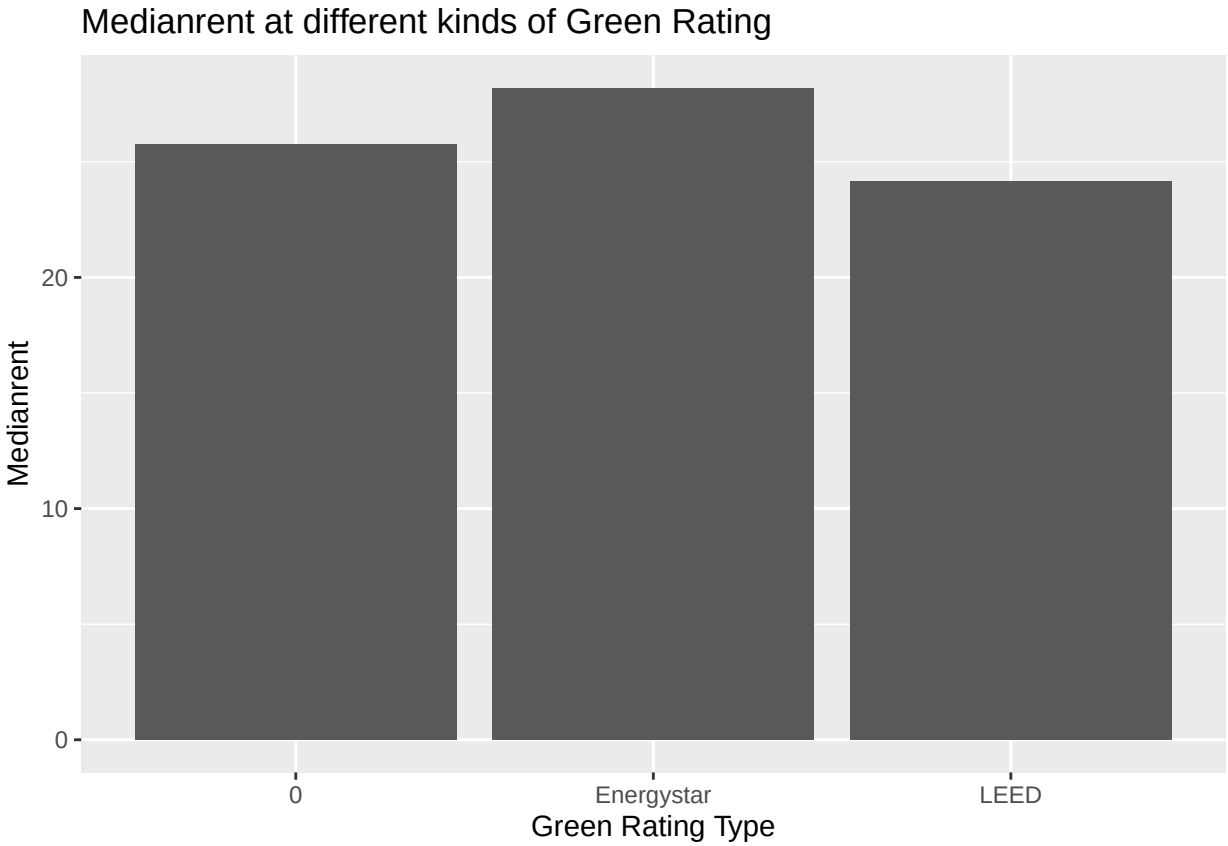
Degree Days with Contract rent with Medianrent Comparison

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



With the help of graphs above, we can conclude that the difference betweeen net and non net contracts is higher for Non green buildings.This further emphasizes that the green rated building need less maintenance and tenants end up spending minimal amount for the services.

GreenRating Vs Medianrent per Sqft



These graphs show the median rent of two different types of green buildings. It is evident that green building with Energystar certification end up getting more rent than leed certified green building. Hence, if decided to go ahead with green building, green building should be Energystar certified.

Conclusion

Just constructing a green building might not be sufficient for it to be profitable. We need to consider several other factors to make this plan profitable.Those points are as follow: 1. Building should be energystar certified. 2. It should be rented out as Non net contract. 3. Tenant should be charged for utilities at the market rates. This ensures that company could make money on the utilities charge and compensate for the initial investment in the coming years as the electricity and gas prices tend to hike.

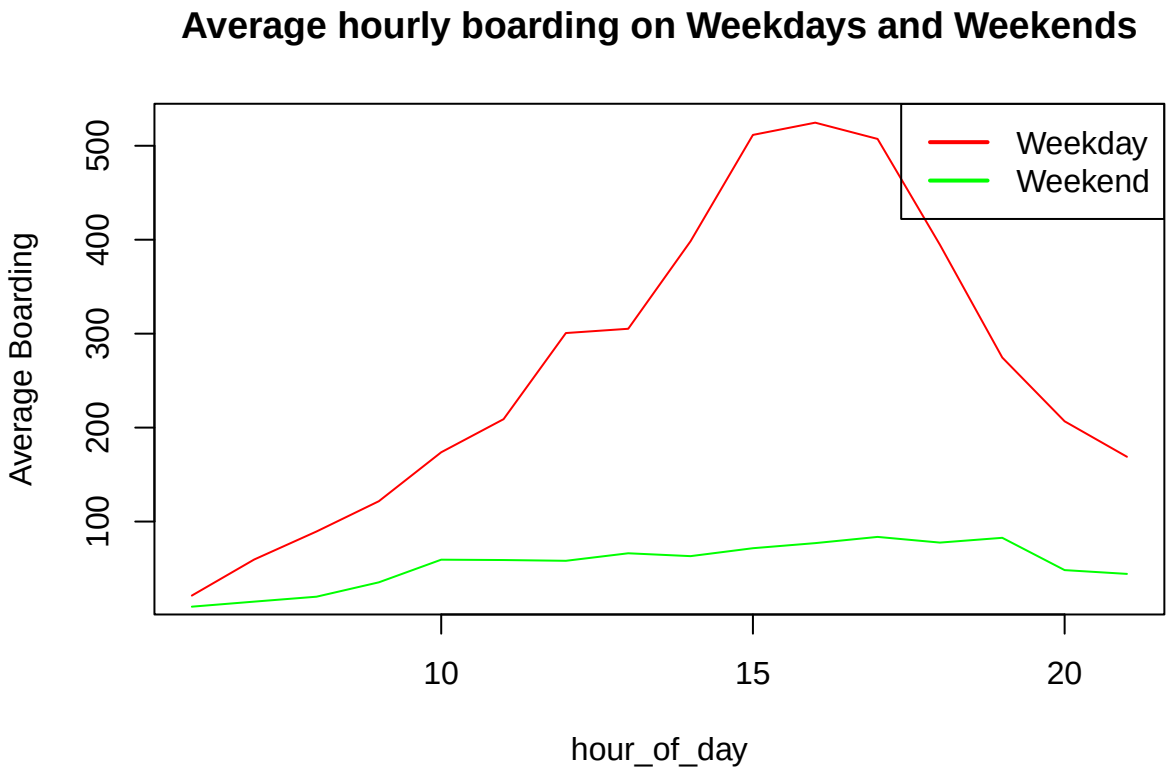
Q4 ~ Visual story telling part 2: Capital Metro data

- Total hourly boarding and alighting for weekdays and weekends:

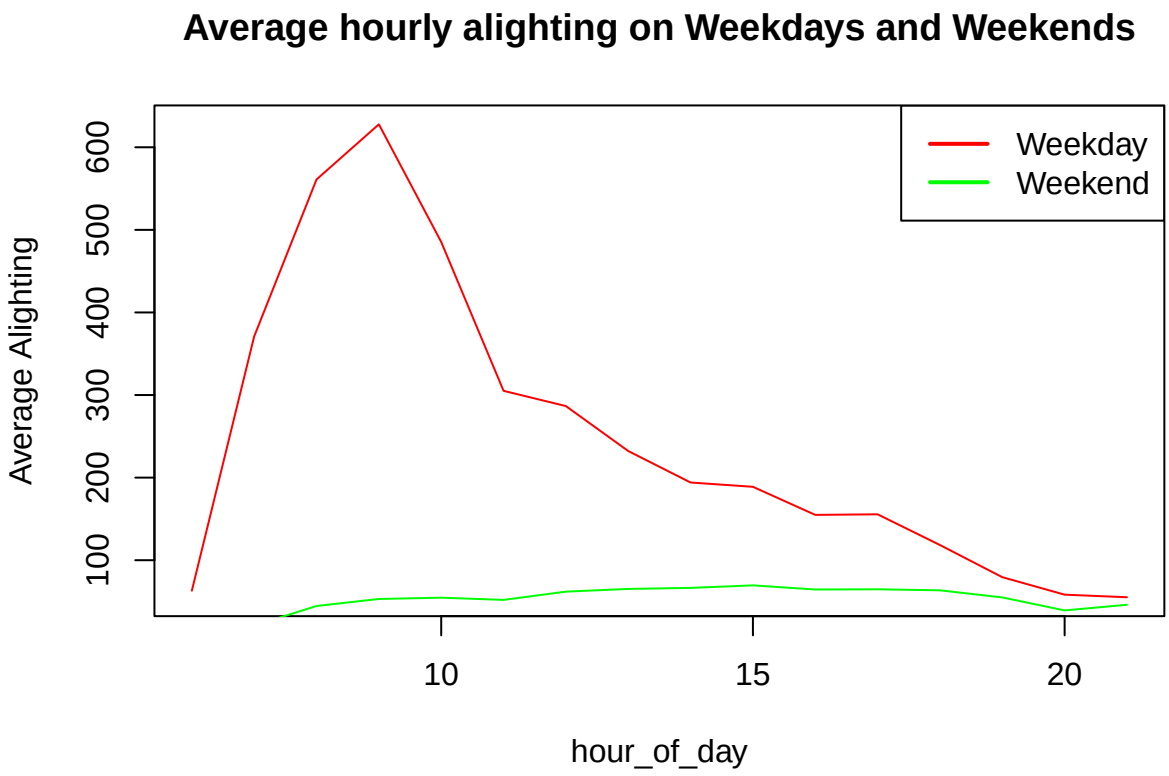
```
## # A tibble: 32 x 4
## # Groups:   hour_of_day [16]
##    hour_of_day weekend boarding alighting
##          <int> <chr>      <int>     <int>
##  1           6 weekday     1379      4102
##  2           6 weekend      244       241
##  3           7 weekday     3873     24114
```

```
##  4           7 weekend      384       542
##  5           8 weekday     5812     36466
##  6           8 weekend      520      1158
##  7           9 weekday     7908     40805
##  8           9 weekend      918      1379
##  9          10 weekday    11294     31549
## 10          10 weekend     1545      1420
## # ... with 22 more rows
## # i Use 'print(n = ...)' to see more rows
```

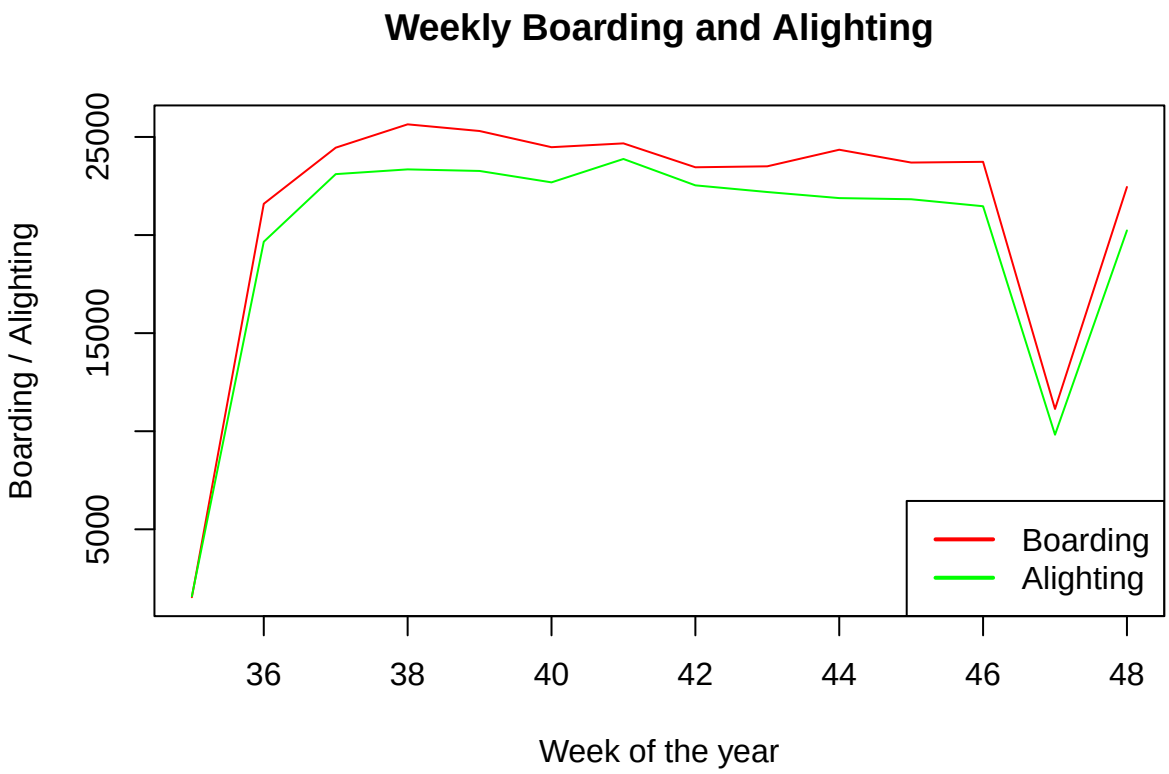
- Average hourly boarding and alighting:



- On weekdays, the average boarding peaks at hour 4-5 PM, which is 525
- This makes sense as students will be heading back after classes
- The boarding on weekends is considerably lower, the maximum is between 5-7 PM



- On weekdays, the average alighting peaks at 9-10 AM, which is 628
- This can be explained as students will be coming to classes from their homes
- The alighting on weekends is again considerably lower, the maximum is at 3-4 PM
- Checking boarding and alighting trends across the weeks:



- There is a sharp drop in ridership during week 47 (Nov 18-24, 2018). This corresponds to the Thanksgiving Week!
- Thanksgiving was on Thursday, November 22, 2018
- For the first week (Week 35) the data is only captured for 1 day, hence the low ridership

Q5 ~ Portfolio modeling

Following is the list of ETF's I considered for making three portfolio's for comparison and analyzing. Our idea on making portfolios was based on type of ETF's rather than sectors and so we have mixed sectors but a different combination of type of ETF's. This is because even if one sector gets affected due to any natural/climatic/environmental issue, other investments in the same portfolio will balance it out. And also we will come to know which type of ETF's benefit the most.

PORTFOLIO 1 The focus for this portfolio is on **Large Cap Growth Equities** and large investment with slow but high returns. This portfolio is preferable for people who are risk averse and chances of profit here are extremely high.

1) VOO: Vanguard S&P 500 This is based on S&P index and includes companies listed majorly in tech. It is diversified and has equities of 500 such high floating companies.

2) QQQ: Invesco Trust This is based on Nasdaq index and includes stocks from major tech companies listed on world's most popular index. It has a high volume of trades on a daily basis that makes it easy to rely on.

3) SPY: SPDR S&P 500 This is again based on S&P index and is one of the most largest, heavily traded and most liquid ETF. It is a heavy investment but it is very popular making it a safe bet.

4) CCOR: Core Alternative In this category of ETF's, this one is quite cheap and hence can help in balancing the investment. And also this ETF is very diversified across all sectors making it a safe blind bet.

5) IVV: iShares Core S&P 500 ETF This one is isued by BlackRock which has a very influential portfolio and is again based on S&P index. It has a huge volume too but the investment is quite high. The avg return is relatively high compared to other ETF's in the same category.

```
##
## 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

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

##            QQQ.Open QQQ.High  QQQ.Low QQQ.Close QQQ.Volume QQQ.Adjusted
## 2017-08-08 139.3822 140.5318 139.0151  139.4112   30810400     139.4112
## 2017-08-09 138.2616 139.2953 137.9428  139.2277   36169500     139.2277
## 2017-08-10 138.4355 138.5321 136.1073  136.2426   63331700     136.2426
## 2017-08-11 136.4261 137.5757 136.1170  137.2762   45862500     137.2762
## 2017-08-14 138.3679 139.2566 138.3292  139.0441   34161000     139.0442
## 2017-08-15 139.3919 139.4016 138.7640  139.1407   25509200     139.1407
```

- To start building our portfolio in R, we used getSymbols method to load the tickers with respective returns and generate a dataframe for each ETF.
- We then operated on the same dataframe to adjust for OHLC (Open,High,Low,Close) returns with any split and dividend returns for that day using adjustOHLC function.
- Above is QQQ ETF with it's first five rows showing how the returns look like for each day, with each row representing each day.
- By default the data source is yahoo.

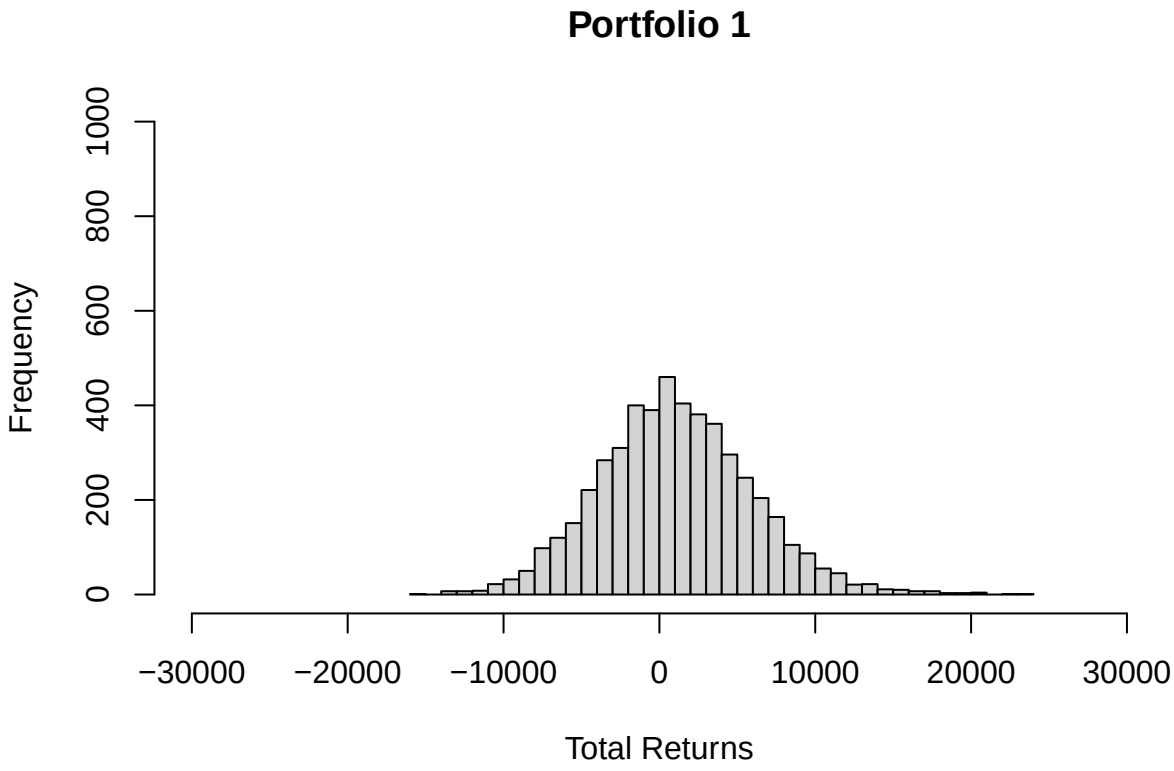
```
##                ClCl.VOOa     ClCl.QQQa     ClCl.SPYa   ClCl.CCORa     ClCl.IVVa
## 2017-08-09 -5.721895e-04 -0.0013166309 -4.042304e-05 -0.003956953 -2.008395e-04
## 2017-08-10 -1.374151e-02 -0.0214404393 -1.411529e-02  0.002006380 -1.385986e-02
## 2017-08-11  1.339718e-03  0.0075870879  1.476863e-03 -0.002402844  1.018454e-03
## 2017-08-14  9.811341e-03  0.0128781627  9.913149e-03  0.002408631  9.970686e-03
## 2017-08-15  8.834519e-05  0.0006948239 -1.216760e-04  0.002803444 -8.060604e-05
## 2017-08-16  1.589759e-03  0.0017357495  1.744380e-03 -0.001198123  1.813468e-03
```

- This is how the returns look like when computed via a percentage change.

- We will now sample any day out of the available data for a 20 day trading period and simulate this exercise for about 5000 times which is nothing but bootstap resampling to track how our wealth changes and what returns and var we get.
- We have taken initial wealth to be 100,000 and a 20 day trading period as mentioned in the question. And the weightage allocated between these 5 ETF's is 12.5%, 25%, 25%, 25%, 12.5% respectively in the order above.

```
##
## Attaching package: 'foreach'

## The following objects are masked from 'package:purrr':
##
##     accumulate, when
```




```
## [1] "The VAR observed in portfolio 1 is  -6687.35230025611"

## [1] "The average estimated return observed in portfolio 1 is  1069.14914946383"

## [1] "The upper bound to return observed in portfolio 1 is  9281.76999134452"
```

PORTFOLIO 2 The focus for this portfolio is on **Corporate and Treasury Bonds** and steady returns on a long term basis. They might not be quick like short cap market ETF's but have a guaranteed return irrespective of the movement of stock market.

1) VCSH: Vanguard Short Term Corporate Bond This ETF is based on corporate bonds and is known for it's large volume and short term investment. It is tracking investments in Bloomberg index.

2) IBDN: iShares iBonds Dec 2022 Corporate Bond This ETF also shares the same trait as VCSH but is very cheaper compared to VCSH. It has really good returns and YTD.

3) SPSB: SPDR Portfolio Short Term Corporate Bond This ETF is based on corporate bonds and is known for it's large volume and high liquidity. It is tracking investments in Bloomberg index.

4) HYG: iShares iBoxx $ High Yield Corporate Bond The benefits from this ETF depends on the knowledge of the investor and their willingness to take risk. The profit or loss will depend on the market and volume.

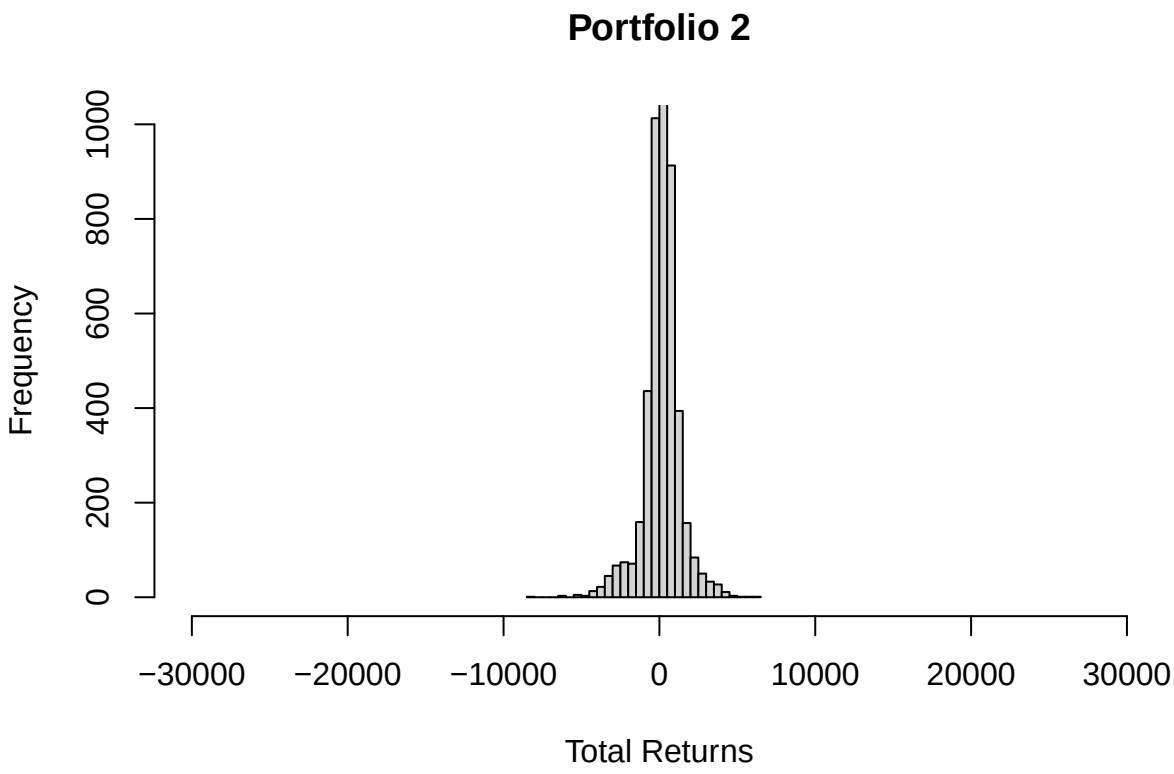
5) VCIT: Vanguard Intermediate-Term Corporate Bond This has a medium length term for investment and is very cost efficient to other ETF's in the same category. It is again based on Bloomberg index like it's peers.

```
##            VCSH.Open VCSH.High VCSH.Low VCSH.Close VCSH.Volume VCSH.Adjusted
## 2017-08-08  71.38990  71.38990 71.31866   71.31866     1075600      71.31866
## 2017-08-09  71.33648  71.37210 71.31866   71.33648     1645200      71.33649
## 2017-08-10  71.33648  71.34538 71.29195   71.30976     1037100      71.30978
## 2017-08-11  71.38990  71.39881 71.32757   71.38100     1166700      71.38100
## 2017-08-14  71.37210  71.38990 71.33648   71.36319     1517800      71.36322
## 2017-08-15  71.31866  71.32757 71.28305   71.29195     1055400      71.29195
```

- Above is VCSH ETF with it's first five rows showing how the returns look like for each day, with each row representing each day.

```
##               ClCl.VCSHa    ClCl.IBDNa    ClCl.SPSBa     ClCl.HYGa
## 2017-08-09  0.0002497815  0.0011913820  0.0013059419 -0.0038601725
## 2017-08-10 -0.0003744726  0.0011900437 -0.0003260515 -0.0063824711
## 2017-08-11  0.0009990259  0.0007923930  0.0000000000  0.0017206011
## 2017-08-14 -0.0002495634 -0.0003958828  0.0003261579  0.0052673650
## 2017-08-15 -0.0009982655 -0.0019801584 -0.0003260515  0.0003417132
## 2017-08-16  0.0011242069  0.0019840872  0.0003261579  0.0001138921
##               ClCl.VCITa
## 2017-08-09  0.0000000000
## 2017-08-10 -0.0001139180
## 2017-08-11  0.0013669324
## 2017-08-14 -0.0005687976
## 2017-08-15 -0.0018211245
## 2017-08-16  0.0035348348
```

- This is how the returns look like when computed via a percentage change.
- We will now sample any day out of the available data for a 20 day trading period and simulate this exercise for about 5000 times which is nothing but bootstap resampling to track how our wealth changes and what returns and var we get.
- We have taken initial wealth to be 100,000 and a 20 day trading period as mentioned in the question. And the weightage allocated between these 5 ETF's is 25%, 25%, 25%, 12.5%, 12.5% respectively in the order above.




```
## [1] "The VAR observed in portfolio 2 is  -1900.09671232285"
```

```
## [1] "The average estimated return observed in portfolio 2 is  163.5243071812"
```

```
## [1] "The upper bound to return observed in portfolio 2 is  1846.93736011974"
```

PORTFOLIO 3 The focus for this portfolio is on **Large Cap Blend Equities** awhich are index focussed and only deal with large cap equities. This type is assigned to portfolios where neither growth nor value characteristics dominate and it represents the overall US market in terms of proportion make us inclined towards investing in this and studying it.

1) SCHD: Schwab US Dividend Equity This is based on the very famous Dow Jones index and is well known for the stable returns.

2) SPLG: SPDR Portfolio S&P 500 It is based on S&P 500 index and is world wide known.

3) VTV: Vanguard Value The companies included in this ETF are very stable and safe. On top of it, this is very diverse at some of the lowest prices with good returns.

4) IWD: iShares Russell 1000 Value This ETF is tilted towards financial companies and has a large cap of companies even greater than 650. They tend to be heavy on securities and occupy a large portion in portfolios.

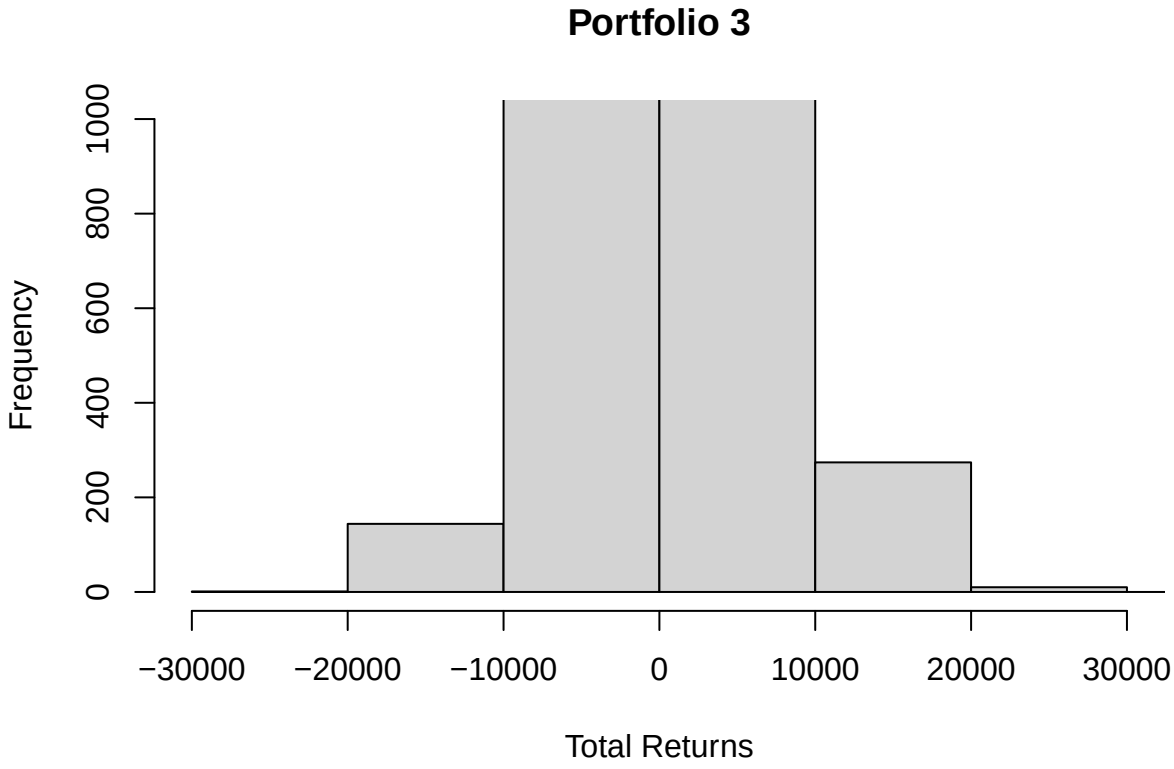
```
##            SCHD.Open SCHD.High SCHD.Low SCHD.Close SCHD.Volume SCHD.Adjusted
## 2017-08-08  39.26088  39.36325 39.13291   39.20116      379700      39.20116
## 2017-08-09  39.15850  39.22675 39.09025   39.21822      372500      39.21820
```

```
## 2017-08-10  39.09025  39.14144 38.93669   38.95375      427100      38.95374
## 2017-08-11  38.95375  39.07319 38.92816   38.93669      370500      38.93669
## 2017-08-14  39.11584  39.27794 39.10731   39.22675      380600      39.22676
## 2017-08-15  39.29500  39.29500 39.14144   39.22675      291900      39.22676
```

- Above is SCHD ETF with it's first five rows showing how the returns look like for each day, with each row representing each day.

```
##               ClCl.SCHDa    ClCl.SPLGa     ClCl.VTVa     ClCl.IWDa
## 2017-08-09  0.0004352557  0.0010378276 -0.0004068450 -0.0013675556
## 2017-08-10 -0.0067435500 -0.0114038877 -0.0114977311 -0.0116397899
## 2017-08-11 -0.0004380420  0.0002621690 -0.0011322800 -0.0019051004
## 2017-08-14  0.0074496277  0.0129303163  0.0087592538  0.0082422088
## 2017-08-15  0.0000000000 -0.0034500948 -0.0001021759 -0.0006884261
## 2017-08-16  0.0023923445  0.0002596503  0.0003065693  0.0007750280
```

- This is how the returns look like when computed via a percentage change.
- We will now sample any day out of the available data for a 20 day trading period and simulate this exercise for about 5000 times which is nothing but bootstap resampling to track how our wealth changes and what returns and var we get.




```
## [1] "The VAR observed in portfolio 3 is  -8421.03437418188"
```

```
## [1] "The average estimated return observed in portfolio 3 is  1811.42619999786"
```

```
## [1] "The upper bound to return observed in portfolio 3 is  11245.951298032"
```

- What we observe in the above results as obtained from all three portfolios is that the **Value at Risk is lowest for Portfolio 2** and **highest for Portfolio 3**. So a **risk averse person would choose Portfolio 2** and a **risk taking person would be inclined towards Portfolio 3.**
- **The average estimated returns from Portfolio 3 is the highest as compared to Portfolio 1 and 2** but when comparing it with VaR we can see that even though Portfolio 3 has great returns the risk associated with it is high and so **a risk averse person would probably be happy with less returns and less risk.** But a risk taking person would be encouraged by the high returns and be motivated to take such a high amount of risk because as you can see the difference in VaR is also very high in between them.
- The upper bound return is very high for Portfolio 1 and 3 which goes in proportion to other parameters and says a lot about how **Portfolio 3 is beneficial in best returns but just that it is very risky.**
- Portfolio 1 has a very normal distribution curve for returns compared to the skewed graphs for Portfolio 2 and 3 which makes us favor Portfolio 3 a little less. And since **Portfolio 2 has a left skewed graph, the lower VaR becomes obvious.**
- We chose Large Cap Blend Equities in Portfolio 3 which cover the whole stock market with proportion to size and growth and hence this ETF is more diversified implying large risk which is also proven by the results we observed above. And Portfolio 2 is about corporate bonds which are short term less risky investments. Since they also have treasury bonds in it, the low risk is automatically implied.

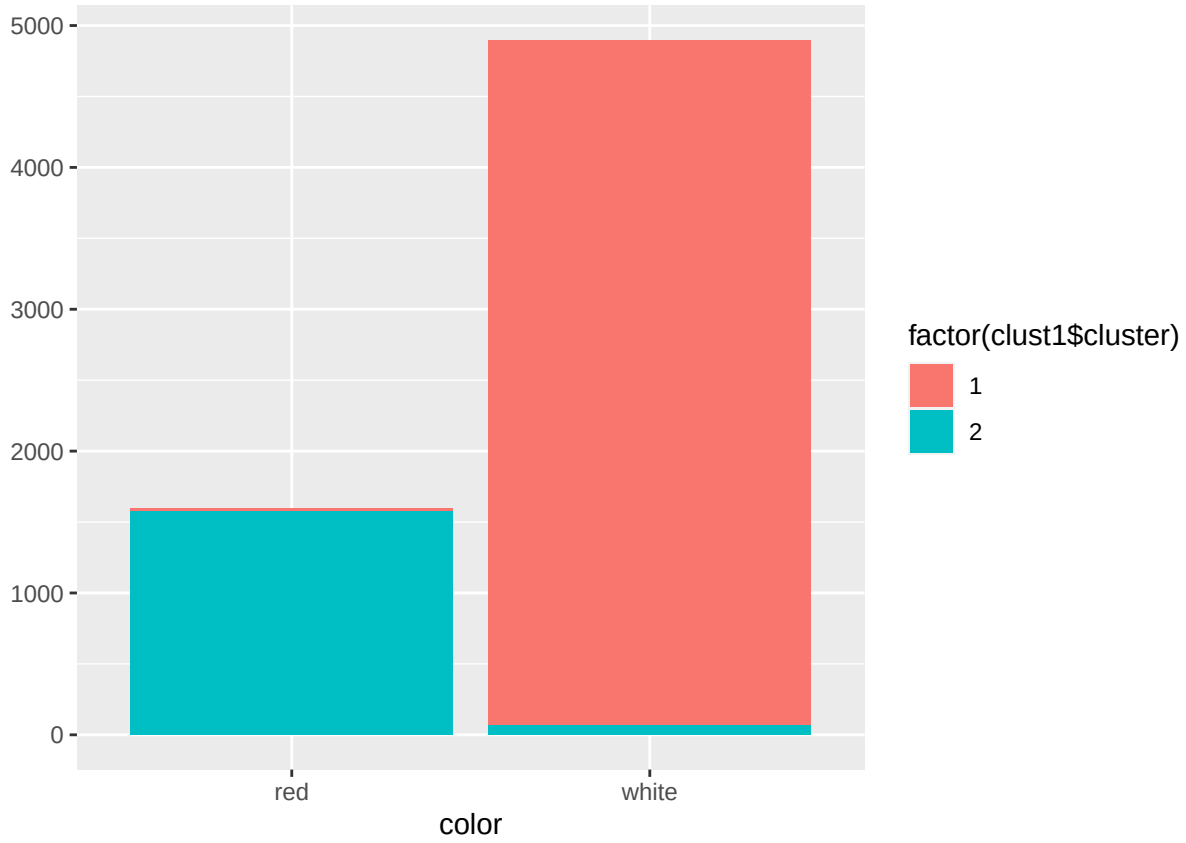
Q6 ~ Clustering and PCA

To start with the problem, we performed following steps.

1.Importred the necessary libraries. 2.Loaded the dataset in a dataframe called “winedf”. 3. As we are going to perform the unsupervised learning algorithm, dataset should be unlabeled. Hence, we removed the last two columns from the dataset, which were dependant varibales. 4. We performed factoring relavant factor variable such as quality. 5.Scaled the numeric data to mean = 0 and s.d = 1

We selected K-Means algorithm as a clustering algorithm. The K-Means algorithm has 2 centers and 50 random initial centroids.

This is the plot of result of the K-Means clustering algorithm on two clusters.



From the graph,we can observe what the k-means clustering algorithm have clustred in cluster 1 and 2.We can also see the wine type, whether it's red or white. We can conclude that, attributes in cluster 1 are most likely corresponding to red wine properties, and attributes of cluster 2 are most likely corresponding to the white wine properties.

We have represented the above plot in the tabular form.

```
##
##            1    2
##   red     24 1575
##   white 4830   68
```

We can treat this table as a confusion matrix to calculate the accuracy of the K-Means algorithm. Cluster 1 clusterd 4854 wines and Cluster 2 clustered 1643 wines in total. But, cluster 1 clustered 24 wines inaccurately. On the other hand, cluster 2 clustered 68 wines inaccurately.

We further implemented the probablity table of the above result.

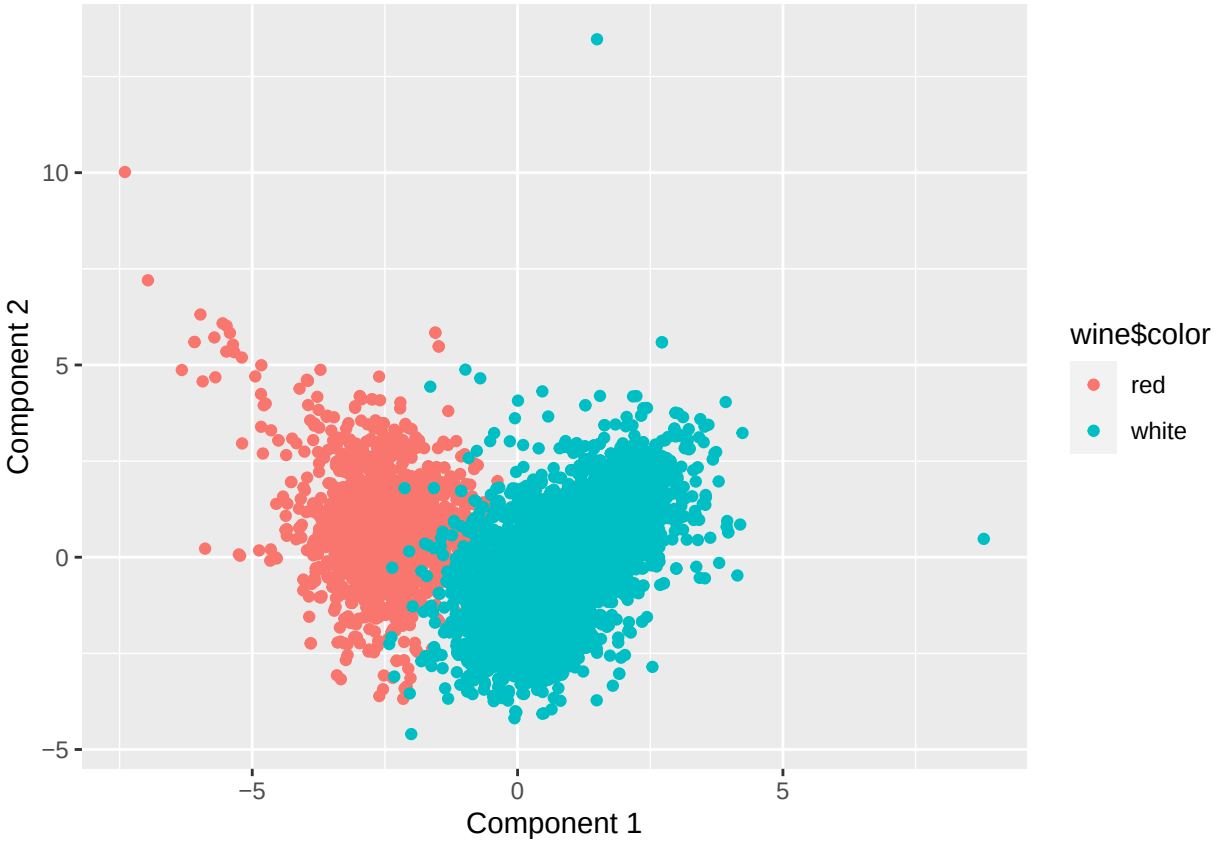
```
##
```

```
##                  1          2
##   red   0.01500938 0.98499062
##   white 0.98611678 0.01388322
```

Accuracy of cluster 1 is 98.61 percent and accuracy of cluster 2 is 98.49 percent.

Now, we will apply PCA on the given wine dataset

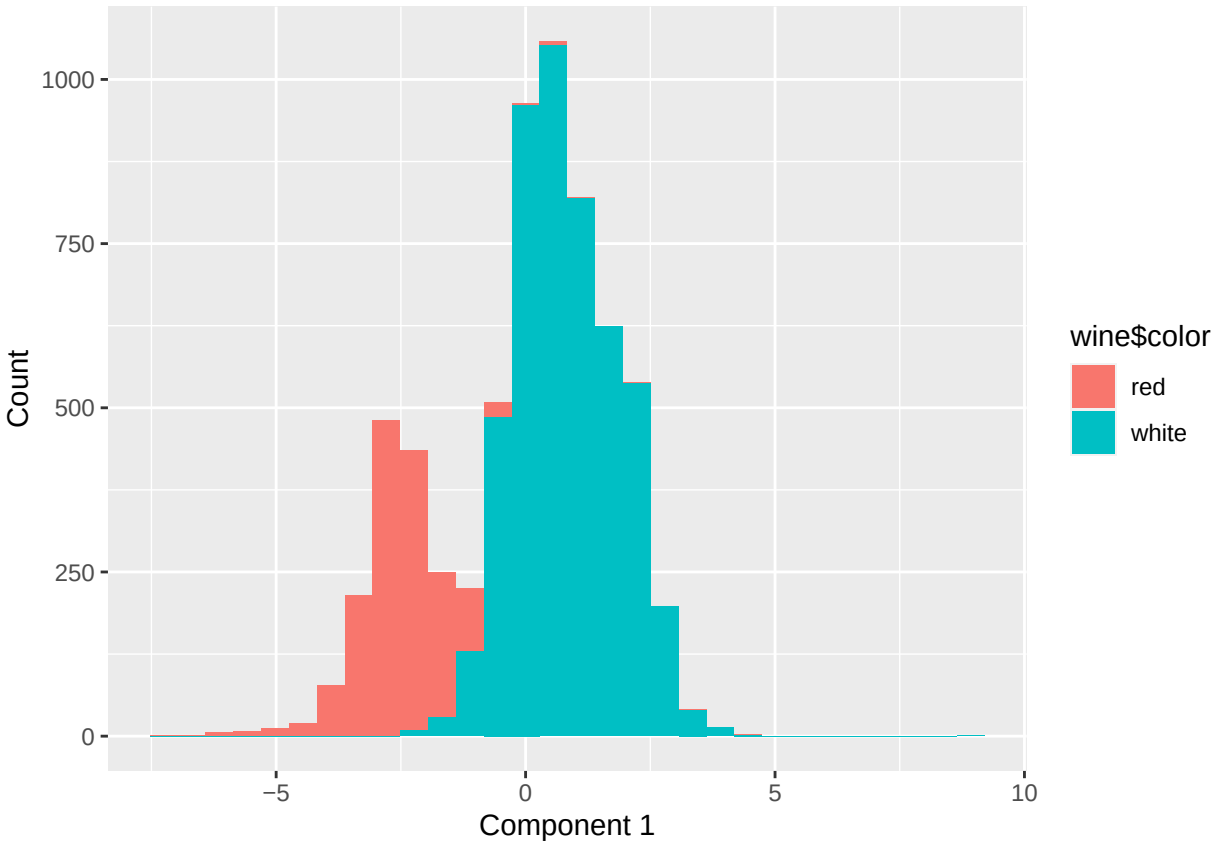
This plot shows the first two Principal Components along with the datapoints.



From the above graph, we can see that there are two groupings with some overlap, between the two principal components. We can conclude that, first two principal components satisfactorily identify the wine type at the expense of some minimal missclassification.

Let's plot first component against count.

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



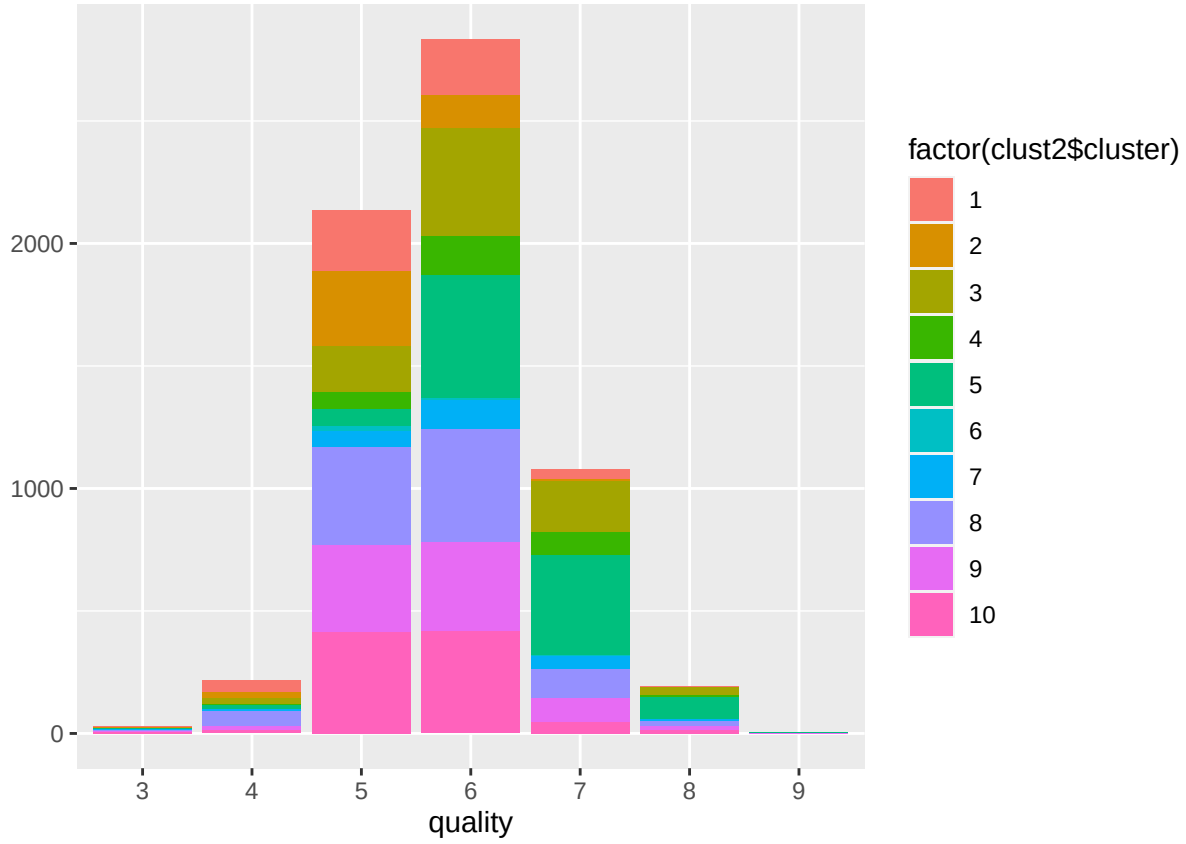
From the above graph, we can see that just the first principle component does a decent job at identifying two groups and predicting the wine type. However, two principal components capture the groups well, as seen above.

Both K-Means clustering algorithm and PCA can be implemented to classify wine into two types- red or white.

Now,let's check if K-Means clustering algorithm and PCA are capable of distinguishing the higher quality wines from the lower quality wines.

First, we will implement K-Means clustering algorithm.The K-Means algorithm has 2 centers and 50 random initial centroids.

This is the plot of result of the K-Means clustering algorithm on 10 clusters.



For distingusihing higher quality wines from lowe quality wines, the K-means algorithm does not seem capable of classifying the wines accurately. It clusters each of the distinct quality wines into different clusters, which are not representative of the clusters.

We have represented the above plot in the tabular form.

```
##
##       1   2   3   4   5   6   7   8   9  10
##   3   7   1   2   0   2   1   4   4   3   6
##   4  50  22  23   7  14   2   6  62  18  12
##   5 252 305 187  72  68  19  68 398 355 414
##   6 232 134 440 161 500   7 121 461 364 416
##   7  42   8 210  90 411   1  56 115 101  45
##   8   4   0  32   8  93   0   6  20  17  13
```

```
##   9   0   0   0   0   4   0   0   0   1   0
```

From this table, we can conclude that K-Mean clustering algorithm is pretty inaccurate in classifying the wine on basis of quality. We can see a very distributed grouping of wine quality amongst the indentified 10 clusters.

We further implemented the probablity table of the above result.

```
##
##               1           2           3           4           5
##   3 0.2333333333 0.0333333333 0.0666666667 0.0000000000 0.0666666667
##   4 0.2314814815 0.1018518519 0.1064814815 0.0324074074 0.0648148148
##   5 0.1178671656 0.1426566885 0.0874649205 0.0336763330 0.0318054256
##   6 0.0818053597 0.0472496474 0.1551480959 0.0567700987 0.1763046544
##   7 0.0389249305 0.0074142725 0.1946246525 0.0834105653 0.3809082484
##   8 0.0207253886 0.0000000000 0.1658031088 0.0414507772 0.4818652850
##   9 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.8000000000
##
##               6           7           8           9           10
##   3 0.0333333333 0.1333333333 0.1333333333 0.1000000000 0.2000000000
##   4 0.0092592593 0.0277777778 0.2870370370 0.0833333333 0.0555555556
##   5 0.0088868101 0.0318054256 0.1861552853 0.1660430309 0.1936389149
##   6 0.0024682652 0.0426657264 0.1625528914 0.1283497884 0.1466854725
##   7 0.0009267841 0.0518999073 0.1065801668 0.0936051900 0.0417052827
##   8 0.0000000000 0.0310880829 0.1036269430 0.0880829016 0.0673575130
##   9 0.0000000000 0.0000000000 0.0000000000 0.2000000000 0.0000000000
```

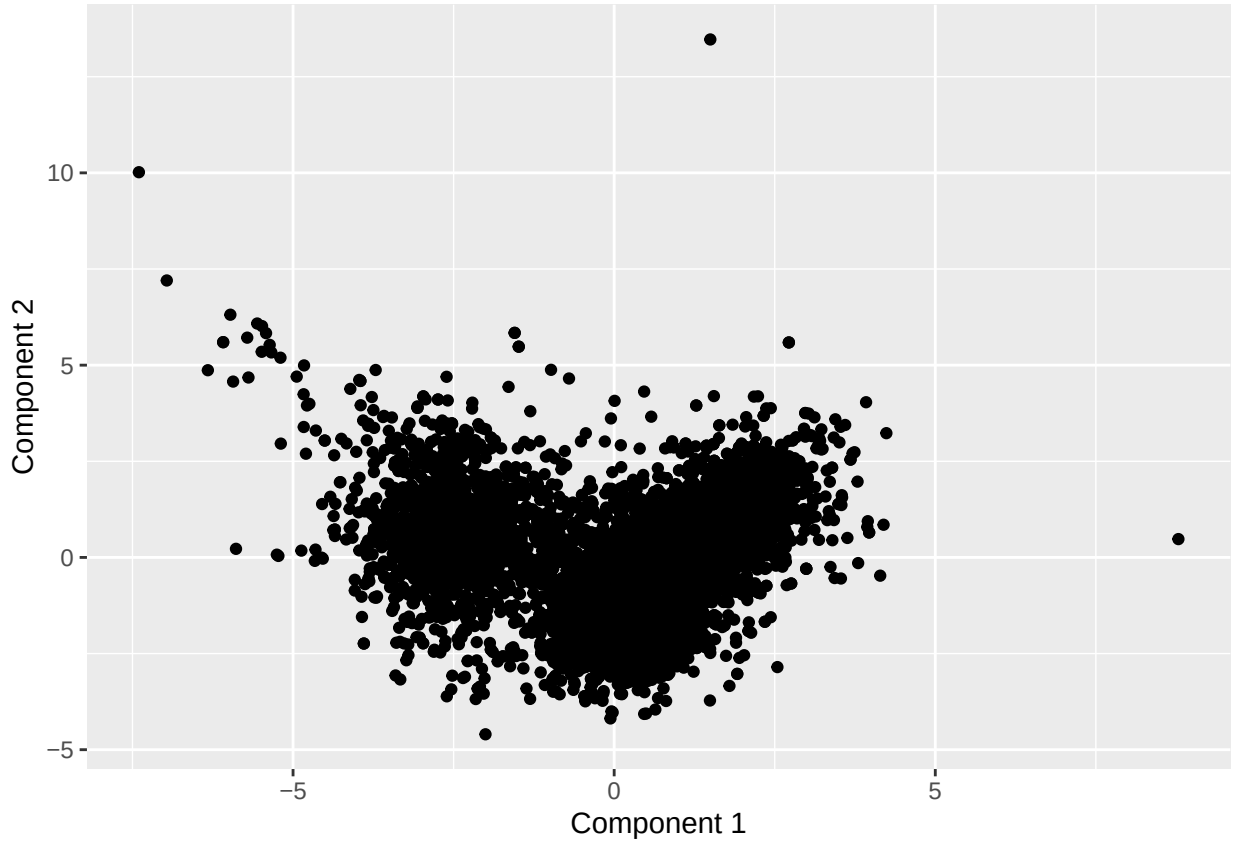
Above table represents the probabalistic accuracy of the clusters. We are not acheiving good accuracy scores for any quality within any cluster.

K means clustering is not capable of distinguishing the higher quality wines from the lower quality wines.

Now,let's check if PCA is capable of distinguishing the higher quality wines from the lower quality wines.

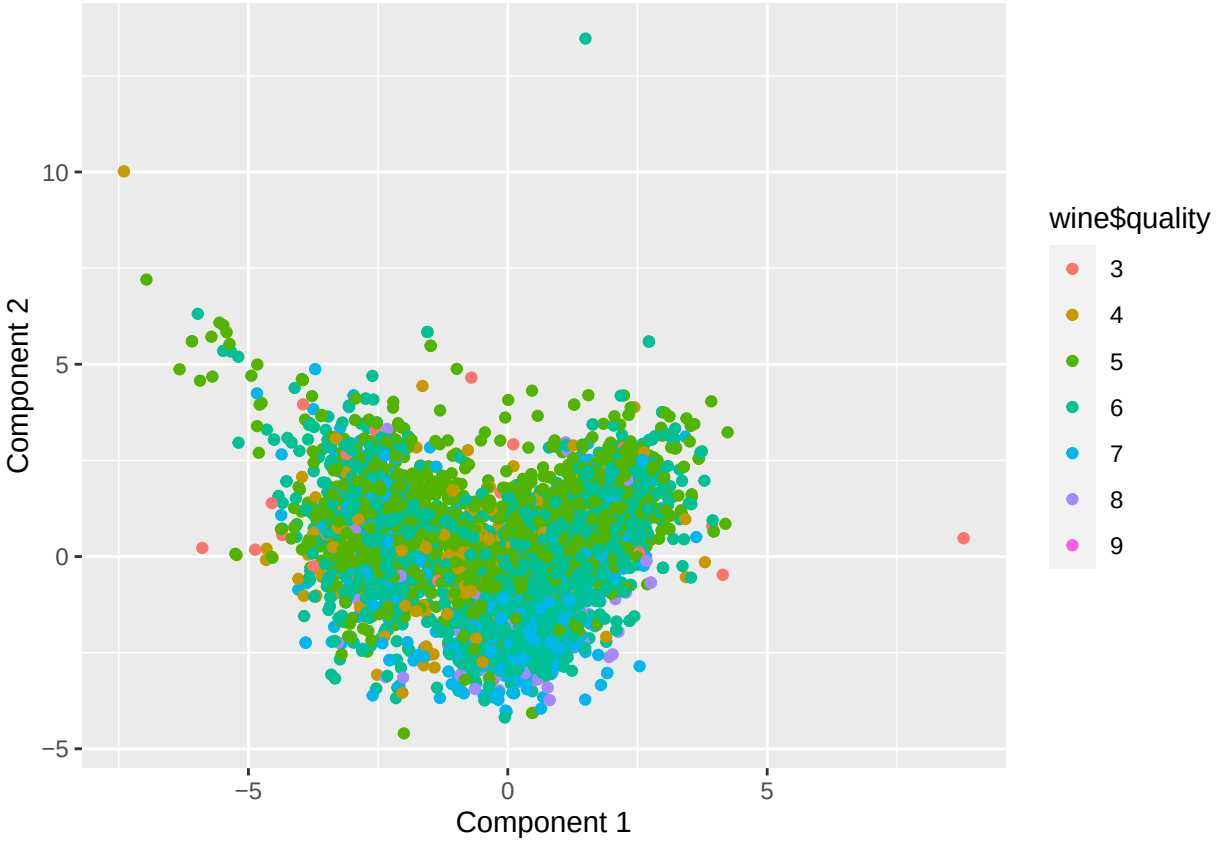
Now, we will apply PCA on the given wine dataset

This plot shows the first two Principal Components along with the datapoints.



From the above graph, we can see that there are two groupings with some overlap, between the two principal components.

We can superimpose the original qualities of the wine onto the above two Principal Components to see if the groupings can be used to identify quality of wines.



From the above graph, we can see that two principal components are not capable of distinguishing the higher quality wines from the lower quality wines.

Finally we can draw following conclusions:

1.Both K-Means clustering and PCA methods are capable of distinguishing the red wines from the white wines. 2.Both K-Means clustering and PCA methods are not capable of distinguishing the higher quality wines from the lower quality wines. 3.For this dataset, PCA makes more sense to us for two reasons: 1.PCA is able to characterize the attributes which give each type of wine its properties. 2.K-Means clustering algorithm would not give desirable results with the variation of K value.

Q7 ~ Market segmentation

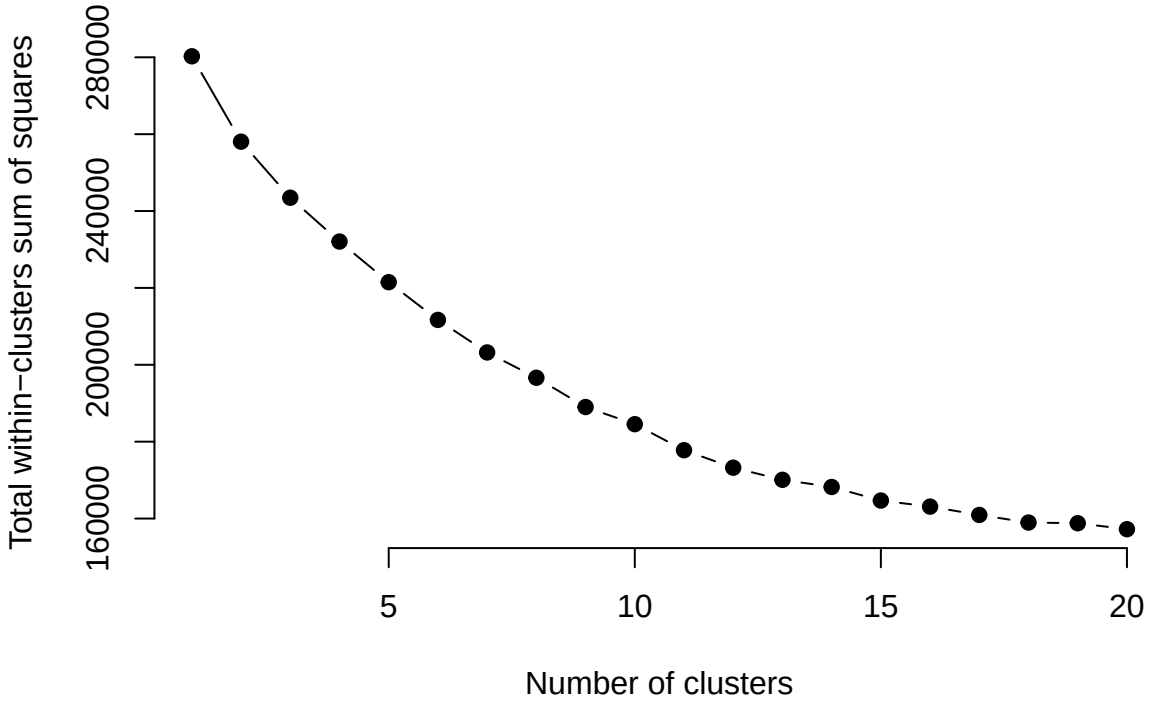
```
## [1] 7882   36
```

- Data has 7882 rows with 37 columns
- There are not many spam tweets

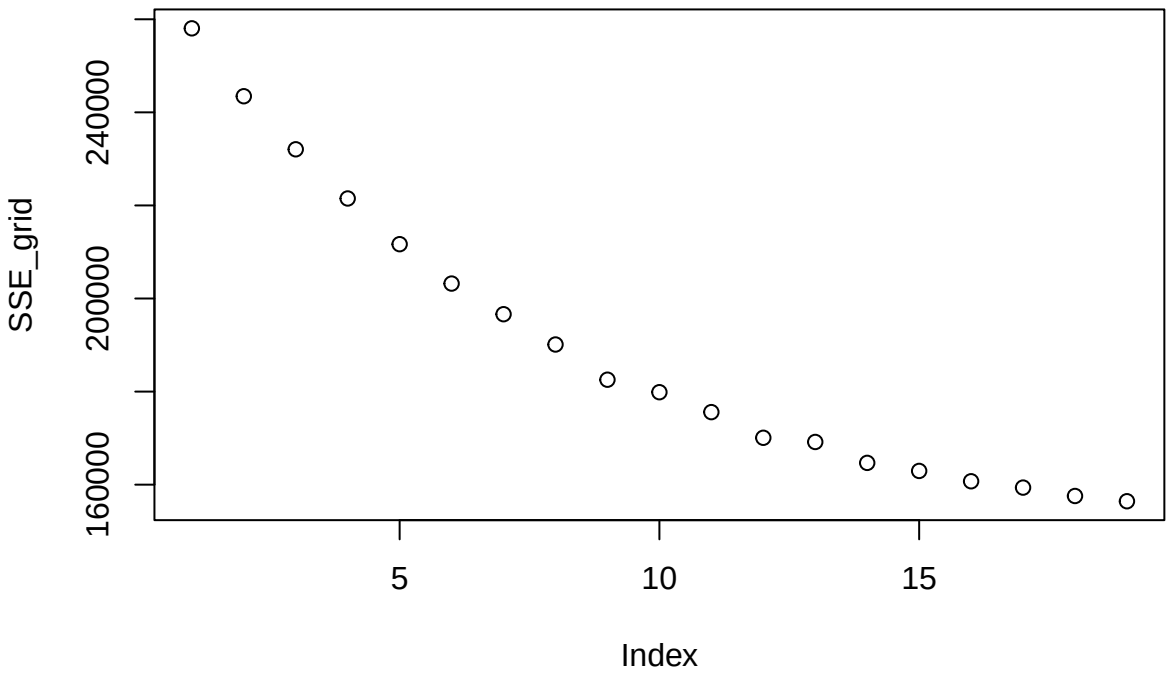
- Removing probable bots based on adult tweets

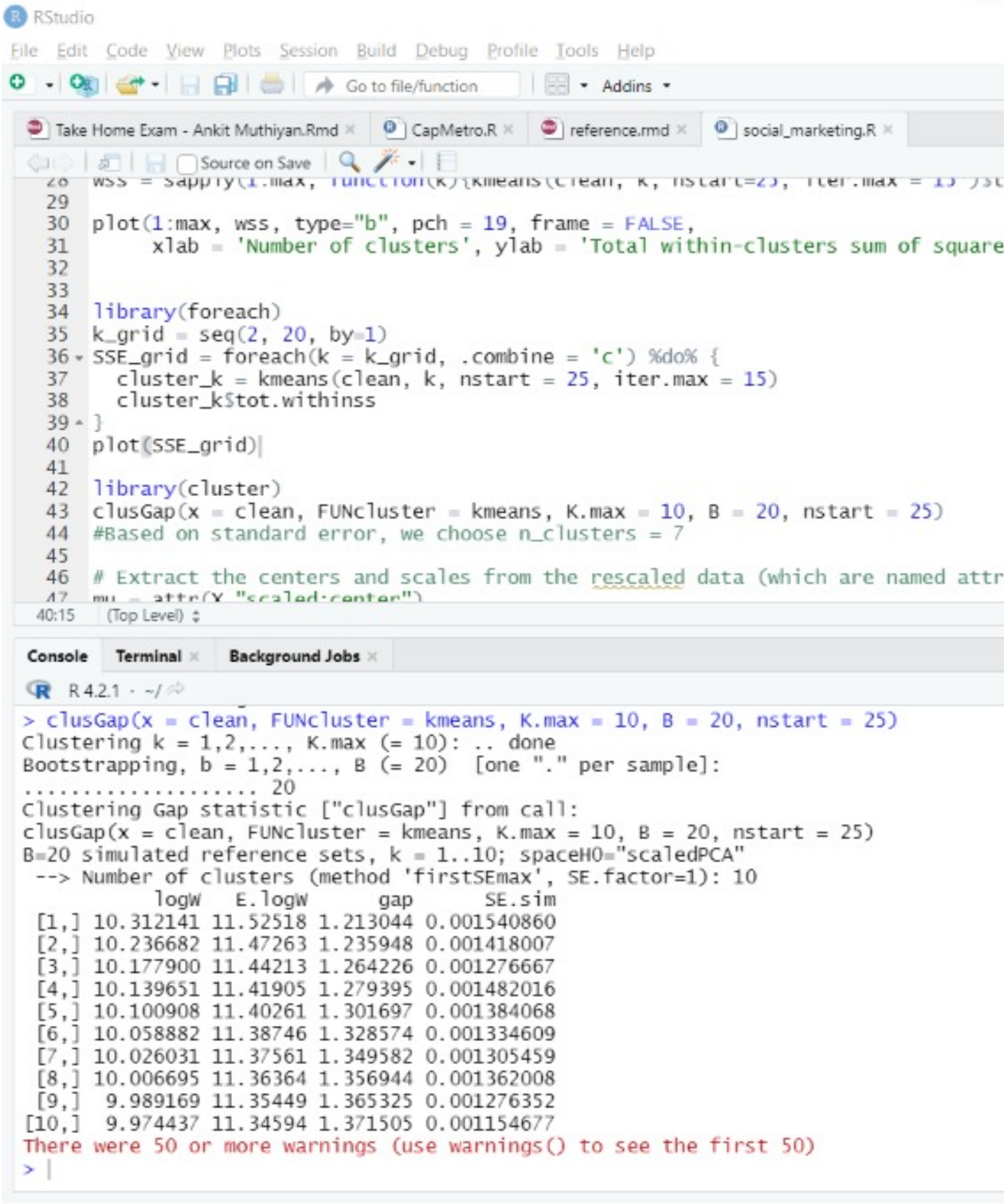
```
## [1] 7786   36
```

- Cleaned data contains 7786 rows. 96 probable bots removed
- Now scaling the data and running K-means clustering to get market segments
- Computing and plotting wss for K = 2 to 20



- Plotting SSE grid:


SSE_grid
160000
200000
240000
5
10
15
Index


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Take Home Exam - Ankit Muthiyan.Rmd
CapMetro.R
reference.rmd
social_marketing.R
Source on Save
wss = sapply(1:max, function(k){kmeans(clean, k, nstart=25, iter.max = 15)$t
plot(1:max, wss, type="b", pch = 19, frame = FALSE,
xlab = 'Number of clusters', ylab = 'Total within-clusters sum of square
library(foreach)
k_grid = seq(2, 20, by=1)
SSE_grid = foreach(k = k_grid, .combine = 'c') %do% {
cluster_k = kmeans(clean, k, nstart = 25, iter.max = 15)
cluster_k$tot.withinss
}
plot(SSE_grid)
library(cluster)
clusGap(x = clean, FUNcluster = kmeans, K.max = 10, B = 20, nstart = 25)
#Based on standard error, we choose n_clusters = 7
# Extract the centers and scales from the rescaled data (which are named attr
mu = attr(X,"scaled:center")
40:15
(Top Level)
Console
Terminal
Background Jobs
R 4.2.1 · ~/
> clusGap(x = clean, FUNcluster = kmeans, K.max = 10, B = 20, nstart = 25)
Clustering k = 1,2,..., K.max (= 10): .. done
Bootstrapping, b = 1,2,..., B (= 20) [one "." per sample]:
.... 20
Clustering Gap statistic ["clusGap"] from call:
clusGap(x = clean, FUNcluster = kmeans, K.max = 10, B = 20, nstart = 25)
B=20 simulated reference sets, k = 1..10; spaceH0="scaledPCA"
--> Number of clusters (method 'firstSEmax', SE.factor=1): 10
logW E.logW gap SE.sim
[1,] 10.312141 11.52518 1.213044 0.001540860
[2,] 10.236682 11.47263 1.235948 0.001418007
[3,] 10.177900 11.44213 1.264226 0.001276667
[4,] 10.139651 11.41905 1.279395 0.001482016
[5,] 10.100908 11.40261 1.301697 0.001384068
[6,] 10.058882 11.38746 1.328574 0.001334609
[7,] 10.026031 11.37561 1.349582 0.001305459
[8,] 10.006695 11.36364 1.356944 0.001362008
[9,] 9.989169 11.35449 1.365325 0.001276352
[10,] 9.974437 11.34594 1.371505 0.001154677
There were 50 or more warnings (use warnings() to see the first 50)
>

- Based on standard error and previous plots, we choose n_clusters = 7
- Running k-means++ with 7 clusters:
- Analyzing the clusters:

	x
chatter	3.2
health_nutrition	1.8
photo_sharing	1.6
current_events	1.3
college_uni	1.1
travel	1.0

** Cluster1: chatter, politics, news, sports_fandom, photo_sharing, health_nutrition ** Count = 555 ** Politically active and opinionated

	x
tv_film	4.9
college_uni	4.8
art	4.3
chatter	4.0
online_gaming	3.1
photo_sharing	2.4

** Cluster2: chatter, photo_sharing, health_nutrition, cooking, sports_fandom, politics ** Count = 3683; this is the largest cluster ** Photo sharing, Health, nutrition and cooking enthusiasts

	x
sports_fandom	6.1
religion	5.5
food	4.7
parenting	4.2
chatter	3.9
school	2.7

** Cluster3: health_nutrition, chatter, personal_fitness, cooking, photo_sharing, politics ** Count = 426 ** Similar to Cluster2, more into nutrition and fitness

	x
chatter	8.6
health_nutrition	5.4
photo_sharing	5.2
shopping	3.4
personal_fitness	3.0
cooking	2.0

** Cluster4: health_nutrition, chatter, personal_fitness, cooking, photo_sharing, politics ** Count = 926 ** Same top features, can be clubbed with cluster3

	x
cooking	11.4
photo_sharing	5.9
fashion	5.8
chatter	4.2
beauty	4.1
health_nutrition	2.6

** Cluster5: politics, travel, chatter, health_nutrition, photo_sharing, news ** Count = 353 ** Travelers, and politically inclined

	x
adult	5.9
chatter	4.3
health_nutrition	2.2
photo_sharing	2.2
travel	1.6
current_events	1.5

** Cluster6: chatter, photo_sharing, cooking, health_nutrition, shopping, politics ** Count = 1431 ** Photo, cooking and nutrition

	x
politics	9.2
travel	5.7
news	5.4
chatter	4.2
computers	2.5
automotive	2.4

** Cluster7: college_uni, online_gaming, chatter, photo_sharing, health_nutrition, cooking ** Count = 412 ** Students and gamers

- NutrientH20 needs to target the the above mentioned clusters.
- Important Recurring features: photo_sharing, health_nutrition, cooking, politics

Q8 ~ Author Attribution

The steps we will follow in this question are as follows:

1) We will first load the train and test directories into a corpus of documents for each author. There are 2500 documents in both train and test corpus and author names are being recorded for each of those 2500 documents. We have used readPlain function and basic lists for this task.

2) The next task was to clean and preprocess the text data in both test and train datasets in order to be able to perform any visualization or perform Machine Learning tasks. For this we have used the below functions. – tolower: Make everything lowercase – removeNumbers: Removing all numerical digits – removePunctuation: Removing all punctuation marks – stripWhitespace: Removing all extra whitespaces – removeWords: Removing all stopwords – ConvertStrings: Removing all extra characters because of conversion of corpus to lists.

3) The next task we did was generating document term matrix for both train and test datasets. Below are the details of the corpus matrix for train and test respectively.

```
## <<DocumentTermMatrix (documents: 2500, terms: 31920)>>
## Non-/sparse entries: 498537/79301463
## Sparsity           : 99%
## Maximal term length: 36
## Weighting          : term frequency (tf)
## Sample             :
##       Terms
## Docs   billion company market million new one percent said will year
##   111        0       0      3       2   3   7       0    6    4    0
##   130        0       1      1       5   0   3       2    1    1    3
##   142        3       1      3       1   2   0       1    9    6    7
##   2127       4       1      2       0   7   3       7    5   11    6
##   2133       2       0      1       0   0   2       5    7    3    3
##   2139       0       1      7       0   1   1      14   23    9    1
##   2140       4       0      0       0   2   5       3   12   15    2
##   599        1       5     11       0   0   3       0   16    1    1
##   763        0       0      4       0   3   0       0   20    3    9
##   766        0       0      7       0   4   1       0   20    4    9
```

```
## <<DocumentTermMatrix (documents: 2500, terms: 32706)>>
## Non-/sparse entries: 505668/81259332
## Sparsity           : 99%
## Maximal term length: 45
## Weighting          : term frequency (tf)
## Sample             :
##       Terms
## Docs   billion company market million new one percent said will year
##   1089       0       1      1       3   1   2       3   10    1    0
##   1430       2       6      3       1   4   4       8   13    3    2
##   1446       0       4      1       0   4   6       1    8    0    0
##   1861       0       0      1       1   1   2       0    7    8    0
##   1936       6       6      1       2   3   1       5   10   10    2
##   2104       0       0     11       0   4   2      12   25    8    1
##   3          0       1      0       1   4   4       2    9    2    0
##   771        0       0      8       0   6   6       7   28    2    9
##   798        0       0     15       0  14   5       0   24    8   21
##   874        0       0      0       0   1   2       0    6    6    0
```

4) We then removed sparse terms out of the matrix. This basically removes words which are very sparsely present in the dataset. We have set the threshold as 0.94.

5) We generated a frequency of word count matrix and converted it into a dataframe in order to process this frequency for further processes.

6) Below displayed are the top 500 frequent words in the train dataset.

```
##   [1] "also"          "announced"     "business"      "character"
##   [5] "computer"      "director"      "early"         "fund"
##   [9] "get"           "group"         "internet"      "investors"
##  [13] "law"           "listauthor"    "local"         "lower"
##  [17] "major"         "may"           "meta"          "million"
```

```
##  [21] "money"          "month"          "national"       "net"
##  [25] "new"            "offer"          "one"            "said"
##  [29] "services"       "set"            "shares"         "state"
##  [33] "still"          "technology"     "third"          "trade"
##  [37] "tuesday"        "wednesday"      "world"          "can"
##  [41] "communications" "corp"           "earlier"        "even"
##  [45] "just"           "like"           "many"           "now"
##  [49] "people"         "plan"           "plans"          "president"
##  [53] "sector"         "service"        "software"       "system"
##  [57] "trading"        "use"            "will"           "executive"
##  [61] "companies"      "end"            "four"           "good"
##  [65] "government"     "including"      "international"  "market"
##  [69] "months"         "next"           "number"         "operating"
##  [73] "products"       "see"            "six"            "three"
##  [77] "two"            "week"           "work"           "year"
##  [81] "interest"       "statement"      "analyst"        "banks"
##  [85] "buy"            "company"        "exchange"       "financial"
##  [89] "however"        "last"           "much"           "officials"
##  [93] "sales"          "securities"     "states"         "take"
##  [97] "already"        "another"        "big"            "billion"
## [101] "court"          "expected"       "firms"          "foreign"
## [105] "former"         "future"         "general"        "investment"
## [109] "likely"         "markets"        "move"           "operations"
## [113] "part"           "say"            "told"           "united"
## [117] "added"          "chief"          "made"           "recent"
## [121] "since"          "stock"          "value"          "years"
## [125] "around"         "back"           "based"          "chairman"
## [129] "customers"      "friday"         "half"           "high"
## [133] "inc"            "results"        "time"           "first"
## [137] "growth"         "key"            "monday"         "news"
## [141] "saying"         "strong"         "well"           "bank"
## [145] "current"        "deal"           "economic"       "report"
## [149] "thursday"       "way"            "companys"       "domestic"
## [153] "going"          "industry"       "make"           "five"
## [157] "share"          "increase"       "reuters"        "long"
## [161] "meeting"        "official"       "think"          "spokesman"
## [165] "analysts"       "percent"        "amp"            "firm"
## [169] "largest"        "total"          "costs"          "due"
## [173] "pay"            "price"          "prices"         "ago"
## [177] "management"     "merger"         "cost"           "profit"
## [181] "second"         "agreement"      "reported"       "capital"
## [185] "czech"          "close"          "earnings"       "higher"
## [189] "rise"           "rose"           "british"        "bid"
## [193] "beijing"        "stake"          "profits"        "quarter"
## [197] "ltd"            "party"          "per"            "shareholders"
## [201] "pounds"         "cash"           "pence"          "talks"
## [205] "hong"           "kong"           "china"          "chinas"
## [209] "chinese"        "cents"          "tonnes"
```

7) Below displayed are the top 500 frequent words in the test dataset.

```
##   [1] "added"         "amp"           "banking"       "business"
##   [5] "character"     "chief"         "companies"     "dont"
##   [9] "earlier"       "earnings"      "even"          "exchange"
```

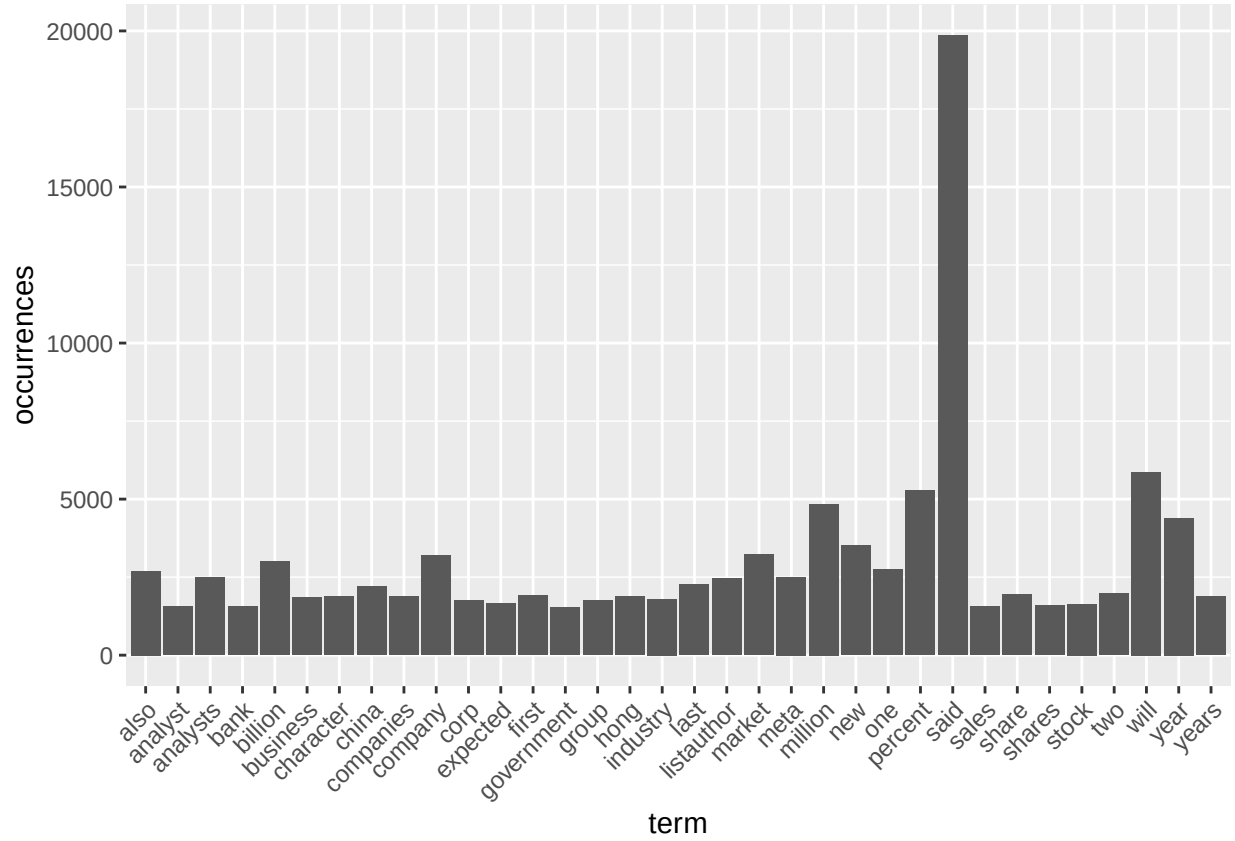
```
##  [13] "executive"     "financial"     "government"    "industry"
##  [17] "investors"     "just"          "last"          "likely"
##  [21] "listauthor"    "made"          "make"          "management"
##  [25] "many"          "may"           "meta"          "million"
##  [29] "month"         "months"        "new"           "north"
##  [33] "now"           "one"           "people"        "plan"
##  [37] "price"         "public"        "recent"        "said"
##  [41] "securities"    "since"         "still"         "stock"
##  [45] "think"         "time"          "tuesday"       "will"
##  [49] "also"          "bank"          "central"       "chairman"
##  [53] "current"       "declined"      "five"          "four"
##  [57] "members"       "much"          "number"        "report"
##  [61] "rights"        "spokesman"     "system"        "three"
##  [65] "thursday"      "top"           "two"           "years"
##  [69] "already"       "announced"     "another"       "around"
##  [73] "back"          "can"           "come"          "company"
##  [77] "computer"      "day"           "director"      "due"
##  [81] "end"           "firm"          "foreign"       "get"
##  [85] "high"          "inc"           "increase"      "international"
##  [89] "internet"      "late"          "like"          "long"
##  [93] "major"         "march"         "national"      "net"
##  [97] "network"       "offer"         "pay"           "per"
## [101] "percent"       "rate"          "second"        "seen"
## [105] "service"       "services"      "set"           "software"
## [109] "way"           "well"          "world"         "corp"
## [113] "going"         "including"     "key"           "president"
## [117] "states"        "trade"         "united"        "week"
## [121] "capital"       "first"         "group"         "groups"
## [125] "however"       "move"          "operations"    "part"
## [129] "party"         "products"      "sources"       "state"
## [133] "told"          "banks"         "firms"         "investment"
## [137] "largest"       "market"        "markets"       "six"
## [141] "total"         "wednesday"     "billion"       "debt"
## [145] "expected"      "growth"        "meeting"       "quarter"
## [149] "rates"         "reported"      "see"           "take"
## [153] "year"          "monday"        "officials"     "plans"
## [157] "say"           "statement"     "strong"        "analysts"
## [161] "costs"         "customers"     "interest"      "close"
## [165] "big"           "cents"         "future"        "money"
## [169] "news"          "prices"        "rise"          "south"
## [173] "analyst"       "cost"          "early"         "higher"
## [177] "economic"      "friday"        "general"       "good"
## [181] "cut"           "deal"          "merger"        "revenues"
## [185] "operating"     "political"     "reuters"       "stake"
## [189] "companys"      "official"      "next"          "local"
## [193] "sales"         "fell"          "profit"        "china"
## [197] "profits"       "share"         "rose"          "half"
## [201] "shareholders"  "shares"        "trading"       "results"
## [205] "saying"        "production"    "british"       "plc"
## [209] "pounds"        "sale"          "cash"          "bid"
## [213] "talks"         "ltd"           "hong"          "kong"
## [217] "beijing"       "chinese"       "chinas"        "kongs"
## [221] "tonnes"
```

8) Below displayed are the common words present in both the top 750 most frequent words list in train and test dataset.

```
##   [1] "also"          "business"      "character"     "group"
##   [5] "listauthor"    "major"         "may"           "meta"
##   [9] "million"       "new"           "one"           "said"
##  [13] "services"      "shares"        "state"         "still"
##  [17] "trade"         "tuesday"       "wednesday"     "can"
##  [21] "corp"          "just"          "many"          "now"
##  [25] "people"        "plans"         "president"     "trading"
##  [29] "will"          "executive"     "companies"     "end"
##  [33] "government"    "international" "market"        "months"
##  [37] "next"          "three"         "two"           "week"
##  [41] "year"          "analyst"       "banks"         "company"
##  [45] "financial"     "last"          "much"          "officials"
##  [49] "sales"         "states"        "take"          "another"
##  [53] "big"           "billion"       "expected"      "foreign"
##  [57] "investment"    "markets"       "say"           "told"
##  [61] "united"        "added"         "chief"         "made"
##  [65] "since"         "stock"         "years"         "around"
##  [69] "chairman"      "friday"        "inc"           "time"
##  [73] "first"         "growth"        "monday"        "news"
##  [77] "well"          "bank"          "deal"          "thursday"
##  [81] "industry"      "make"          "share"         "reuters"
##  [85] "think"         "analysts"      "percent"       "price"
##  [89] "prices"        "profit"        "earnings"      "british"
##  [93] "beijing"       "profits"       "quarter"       "pounds"
##  [97] "hong"          "kong"          "china"         "chinas"
## [101] "chinese"
```

9) Below displayed is the occurence of words having frequency above 1500 in train data and the top 5 words in both train and test dataset.

```
##    said    will percent million    year     new
##   19851    5873    5270    4848    4399    3513
```




```
##    said    will percent million    year     new
##   20122    5831    5597    4986    4247    3331
```

10) Below displayed is the wordcloud with words having minimum frequency of 1200 in train data and ignoring words which are less sparse and very high sparse in between 1 to 80% of documents.

```
## Loading required package: RColorBrewer

##
## Attaching package: 'wordcloud'

## The following object is masked from 'package:PerformanceAnalytics':
##
##     textplot
```

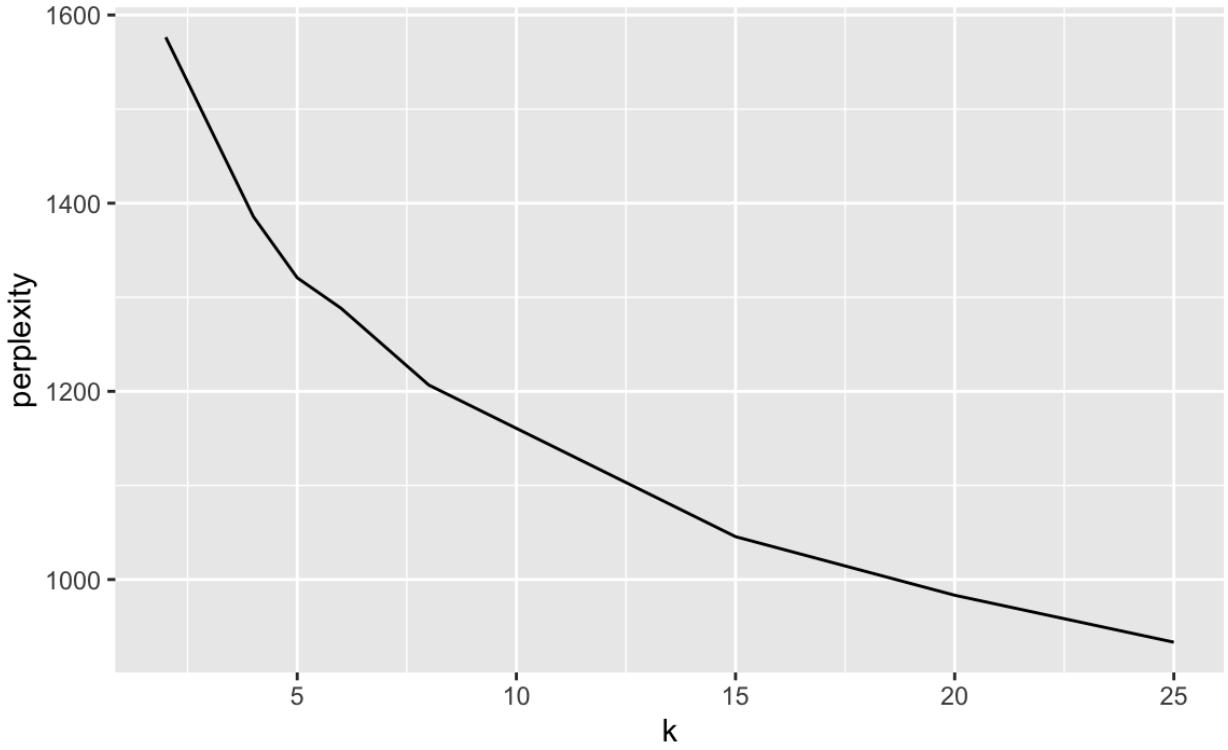
USE CASE 1 - UNSUPERVISED LEARNING

11) We chose to perform topic modelling which is a concept of Unsupervised learning using LDA algorithm. We saw the scores to be mad when tfidf score was used in place of frequency count hence we are only showing results with frequency count matrix.

12) We also had to do some additional stopwords removal before we could fit the data in our model.

13) In order to find the ideal number of clusters we ran the model fitting and topic prediction code for different number of clusters from 2 to 25 and observed where do we see the lowest perplexity score. Perplexity score helps us in finding the efficiency of LDA model and we usually plot it against the number of clusters and find where do we see the bends in the graph. That is because the perplexity score decreases with clusters because the data to be interpreted is increasing.

14) It takes a lot of time to run the model in a loop for different values of k, so we have included the code in include=FALSE and displayed the png we observed after successfully running the code.



15) As we can see from the above graph that with the rise in number of clusters the score is decreasing but it has small bends at 5, 8 and 10. We can try with further larger values of k to find when does the score start increasing again and even change the sample in train and test to find the ideal cluster number.

16) Below are the topics with k=5

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
[1,]	"tonnes"	"percent"	"billion"	"corp"	"china"
[2,]	"oil"	"million"	"group"	"company"	"hong"
[3,]	"told"	"year"	"company"	"inc"	"kong"
[4,]	"air"	"analysts"	"bank"	"companies"	"chinese"
[5,]	"workers"	"sales"	"banks"	"industry"	"government"
[6,]	"industry"	"share"	"financial"	"market"	"beijing"
[7,]	"three"	"quarter"	"british"	"internet"	"foreign"
[8,]	"agreement"	"market"	"pounds"	"services"	"chinas"
[9,]	"world"	"expected"	"market"	"business"	"people"
[10,]	"reuters"	"analyst"	"shares"	"computer"	"trade"

17) Below are the topics with k=8

```
      Topic 1      Topic 2    Topic 3     Topic 4      Topic 5
 [1,] "tonnes"     "million"  "stock"     "bank"       "government"
 [2,] "oil"        "percent"  "market"    "banks"      "court"
 [3,] "air"        "year"     "analysts"  "billion"    "czech"
 [4,] "workers"    "sales"    "shares"    "financial"  "former"
 [5,] "three"      "quarter"  "trading"   "investment" "wang"
 [6,] "plant"      "profits"  "price"     "foreign"    "years"
 [7,] "agreement"  "growth"   "analyst"   "companies"  "law"
 [8,] "production" "profit"   "prices"    "firms"      "minister"
 [9,] "told"       "share"    "investors" "fund"       "case"
[10,] "world"      "analysts" "gold"      "banking"    "state"
      Topic 6     Topic 7   Topic 8
 [1,] "corp"      "company" "china"
 [2,] "industry"  "group"   "hong"
 [3,] "internet"  "billion" "kong"
 [4,] "company"   "british" "chinese"
 [5,] "computer"  "pounds"  "beijing"
 [6,] "service"   "deal"    "chinas"
 [7,] "companies" "million" "official"
 [8,] "inc"       "percent" "people"
 [9,] "services"  "bid"     "officials"
[10,] "software"  "pence"   "trade"
```

18) Below are the topics with k=15

```
      Topic 1    Topic 2     Topic 3      Topic 4   Topic 5      Topic 6
 [1,] "sales"    "company"   "internet"   "million" "bank"       "workers"
 [2,] "analysts" "business"  "computer"   "percent" "banks"      "plant"
 [3,] "quarter"  "executive" "software"   "year"    "financial"  "agreement"
 [4,] "earnings" "chief"     "technology" "profits" "billion"    "union"
 [5,] "share"    "chairman"  "industry"   "profit"  "investment" "ford"
 [6,] "analyst"  "reuters"   "corp"       "half"    "fund"       "plants"
 [7,] "percent"  "told"      "inc"        "pounds"  "banking"    "work"
 [8,] "year"     "interview" "system"     "billion" "firms"      "strike"
 [9,] "expected" "group"     "microsoft"  "net"     "foreign"    "three"
[10,] "cents"    "years"     "customers"  "months"  "funds"      "car"
      Topic 7     Topic 8        Topic 9     Topic 10     Topic 11
 [1,] "law"       "company"      "stock"     "prices"     "czech"
 [2,] "court"     "shares"       "market"    "tonnes"     "government"
 [3,] "state"     "deal"         "trading"   "oil"        "percent"
 [4,] "companies" "bid"          "shares"    "year"       "minister"
 [5,] "rules"     "share"        "investors" "export"     "house"
 [6,] "decision"  "shareholders" "gold"      "industry"   "told"
 [7,] "case"      "offer"        "exchange"  "percent"    "trade"
 [8,] "whether"   "billion"      "markets"   "price"      "party"
 [9,] "time"      "group"        "percent"   "domestic"   "week"
[10,] "drug"      "pence"        "analysts"  "production" "social"
      Topic 12   Topic 13         Topic 14  Topic 15
 [1,] "air"      "market"         "china"   "years"
 [2,] "french"   "news"           "hong"    "wang"
 [3,] "billion"  "corp"           "kong"    "government"
 [4,] "france"   "services"       "chinese" "beijing"
 [5,] "airbus"   "mci"            "beijing" "rights"
 [6,] "francs"   "communications" "chinas"  "officials"
 [7,] "boeing"   "local"          "states"  "people"
 [8,] "european" "television"     "trade"   "former"
 [9,] "airlines" "atampt"         "united"  "communist"
[10,] "cargo"    "billion"        "kongs"   "party"
```

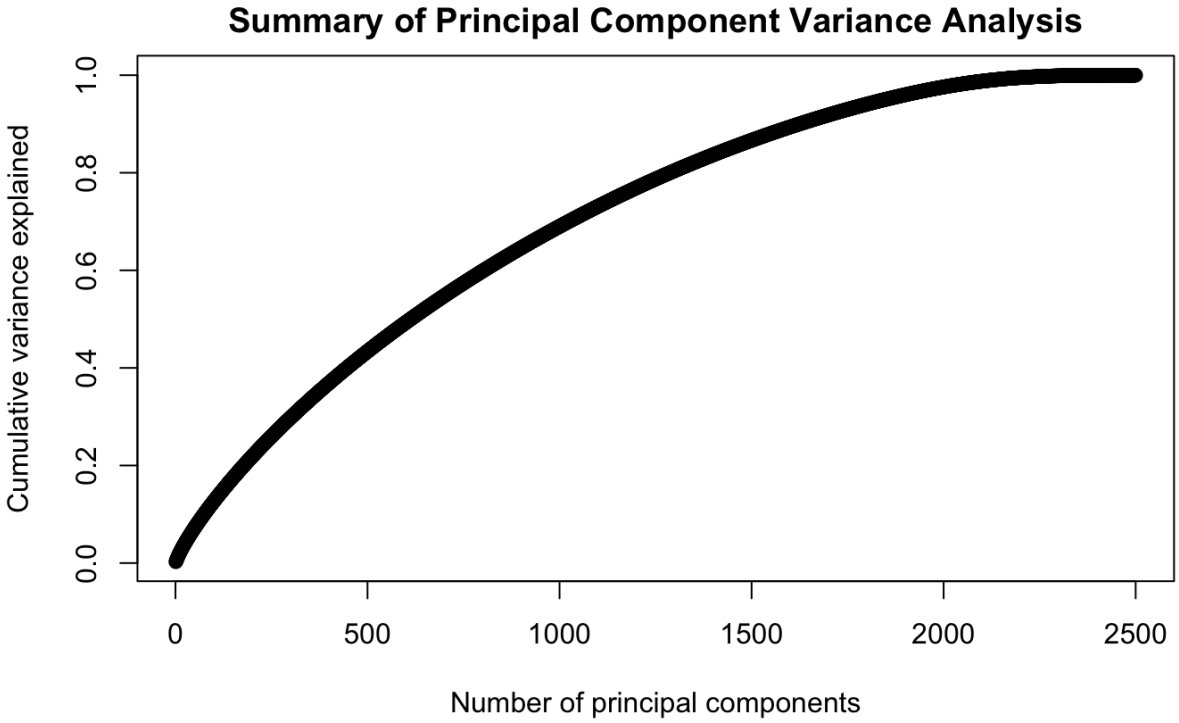
19) But after reading the top 10 words in the cases of topics 5, 8, and 15 we can see that with 8 clusters the topics are very accurate and the words make a lot of sense giving us the jist of what are the broad issues being spoken about in the articles. We can no doubt enhance it further by getting better computational resources to run the model with k as 50,60 or 100 or maybe even more. As we can see that even with higher number of clusters it is not such an ideal situation because as we see the topics list with k=15, topics 8 & 9 and topics 14 & 15 sre almost similar topics and don't have a very good distinguishing words. Hence we would definitely conclude with k=8 as a good number of clusters.

USE CASE 2 - SUPERVISED LEARNING

20) We now implemented the Supervised Machine Learning Use Case but in order to do that we have to first calculate TF-IDF scores for all features.

21) We also have to make sure that test and train data have the same number of features. Hence we remove the extra 55 words that are not present in training data.

22) We then implemented PCA to choose 1000 principal components to reduce the number of predictors. The graph below shows the variance on increasing the components and we can observe that 1000 is a huge number and we should rather keep it to 400-500 in future.



23) What then remained was to implement our model and we chose Random Forest Classifier and selected the number of tries as 6. The accuracy we achieved was 76.96% which is good as this is just the baseline model. Since the model takes a lot of time to run, we have included the screenshots of the output.

```
Warning: package 'randomForest' was built under R version 4.1.2randomForest 4.7-1.1
Type rfNews() to see new features/changes/bug fixes.

Attaching package: 'randomForest'

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

    margin

The following object is masked from 'package:dplyr':

    combine

[1] 0.7696
```

Q9 ~ Association rule mining

Include the required libraries and load the groceries.txt file.

```
##
## Attaching package: 'arules'

## The following object is masked from 'package:tm':
##
##     inspect

## The following objects are masked from 'package:mosaic':
##
##     inspect, lhs, rhs

## The following object is masked from 'package:dplyr':
##
##     recode

## The following objects are masked from 'package:base':
##
##     abbreviate, write
```

We have picked following thresholds

MaxLength = 3

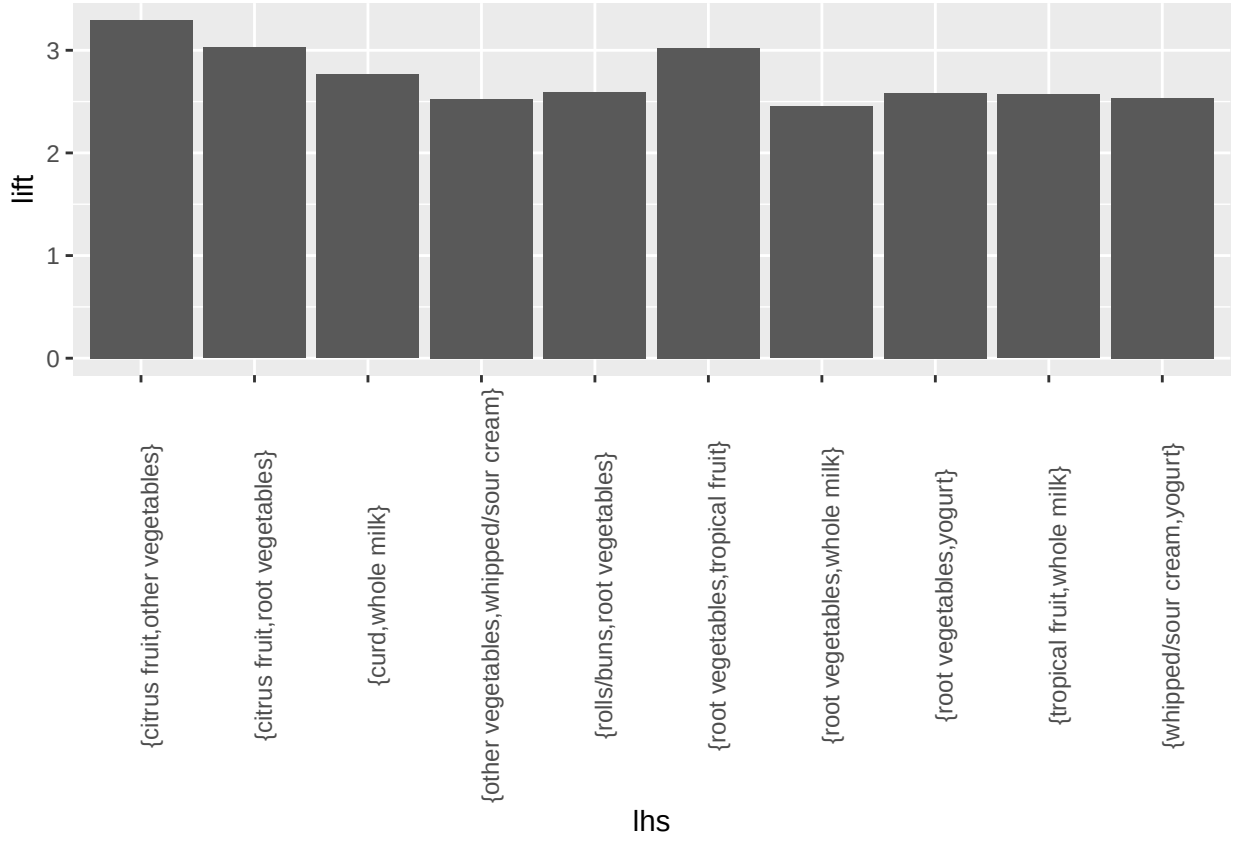
There can be max 3 items in the basket. Most of the dishes are made up of 3 core ingredients.Hence, it is rational to have some correlation in the baskets of these size. ### Support = 1 percent If support value is of 1 percent,we can capture as many baskets as possible with as many grocery items. ### Confidence = 0.35 We seletcted Confidence value of 0.35.Because, this value is neither too small to where it captured not truly correlated baskets nor too large of a confidence value to limit us to narrow of a scope.

Let's apply the apriori to find frequent item sets.

Get the top 10 baskets sorted by lift.

```
##                                       lhs                rhs    support
## 1        {citrus fruit,other vegetables}  {root vegetables} 0.01037112
## 2         {citrus fruit,root vegetables} {other vegetables} 0.01037112
## 3       {root vegetables,tropical fruit} {other vegetables} 0.01230300
## 4                       {curd,whole milk}           {yogurt} 0.01006609
## 5           {rolls/buns,root vegetables} {other vegetables} 0.01220132
## 6               {root vegetables,yogurt} {other vegetables} 0.01291307
## 7            {tropical fruit,whole milk}           {yogurt} 0.01514997
## 8            {whipped/sour cream,yogurt} {other vegetables} 0.01016777
## 9  {other vegetables,whipped/sour cream}           {yogurt} 0.01016777
## 10          {root vegetables,whole milk} {other vegetables} 0.02318251
##    confidence   coverage     lift count
## 1   0.3591549 0.02887646 3.295045   102
## 2   0.5862069 0.01769192 3.029608   102
```

```
## 3   0.5845411 0.02104728 3.020999   121
## 4   0.3852140 0.02613116 2.761356    99
## 5   0.5020921 0.02430097 2.594890   120
## 6   0.5000000 0.02582613 2.584078   127
## 7   0.3581731 0.04229792 2.567516   149
## 8   0.4901961 0.02074225 2.533410   100
## 9   0.3521127 0.02887646 2.524073   100
## 10  0.4740125 0.04890696 2.449770   228
```

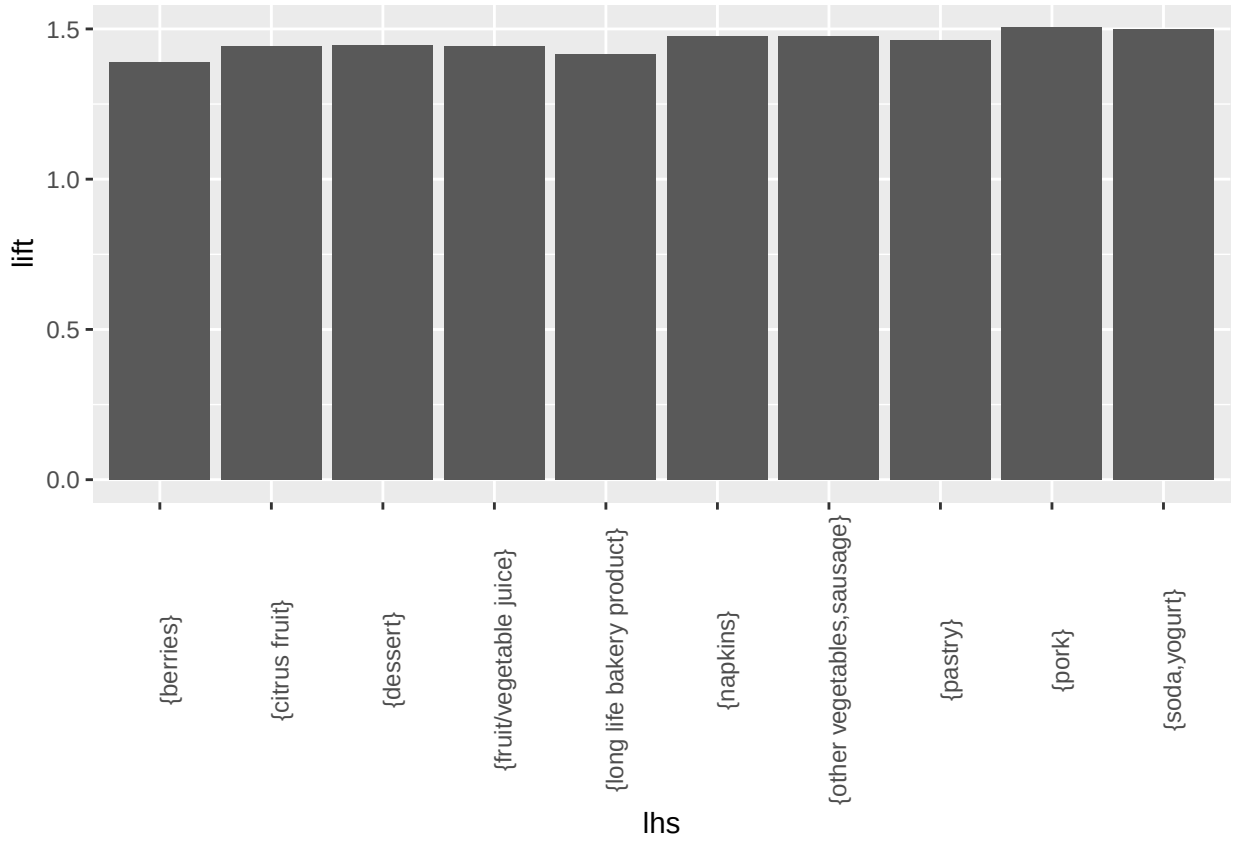


From the above table, we can see that the baskets showing the highest lift are easily correlated. For instance,Baskets with fruits and vegetables could have other types of vegetables.To add to this, baskets with curd and whole milk could have yogurt, given they are all dairy products. Seeing the confidence of these baskets, we can conclude that many of them have confidence of >0.5. This would indicate that shoppers are more likely than not to purchase the additional item.

Get the lowest 10 baskets sorted by lift.

```
##                           lhs          rhs    support confidence   coverage
## 1                   {berries} {whole milk} 0.01179461  0.3547401 0.03324860
## 2  {long life bakery product} {whole milk} 0.01352313  0.3614130 0.03741739
## 3     {fruit/vegetable juice} {whole milk} 0.02663955  0.3684951 0.07229283
## 4               {citrus fruit} {whole milk} 0.03050330  0.3685504 0.08276563
## 5                   {dessert} {whole milk} 0.01372649  0.3698630 0.03711235
## 6                    {pastry} {whole milk} 0.03324860  0.3737143 0.08896797
## 7                   {napkins} {whole milk} 0.01972547  0.3766990 0.05236401
## 8  {other vegetables,sausage} {whole milk} 0.01016777  0.3773585 0.02694459
```

```
## 9               {soda,yogurt} {whole milk} 0.01047280  0.3828996 0.02735130
## 10                     {pork} {whole milk} 0.02216573  0.3844797 0.05765125
##        lift count
## 1  1.388328   116
## 2  1.414444   133
## 3  1.442160   262
## 4  1.442377   300
## 5  1.447514   135
## 6  1.462587   327
## 7  1.474268   194
## 8  1.476849   100
## 9  1.498535   103
## 10 1.504719   218
```

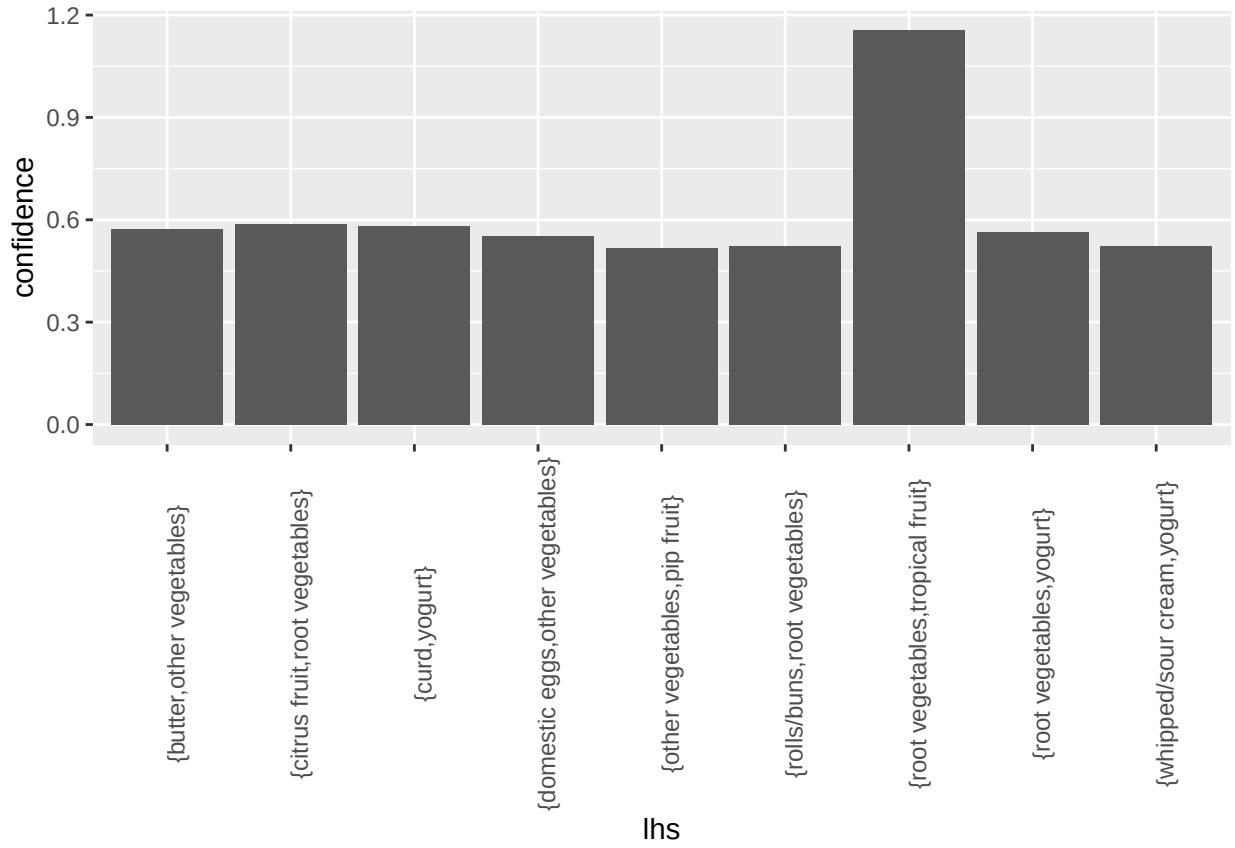


From the above table, we can see that these baskets still show correlation. For instance, observing the first row, we see that baskets of berries are slightly expected to also have whole milk - but it would certainly be possible to buy berries with milk.Each of the bottom 10 baskets sorted by lift have whole milk as the secondary item. This is rational since whole milk is a common thing to have in households and many people buy milk frequently.

It is logical for milk to have a low value of lift because of the high level of confidence that already exists for a customer to purchase milk before other items enhanvce it's likelihood. Lift cannot be high for products that have high purchase confidence.We also see that the confidence values of these baskets are slightly lower than that of the top 10 . Every value is less than 0.4. The confidence values are still high though. We can expect shoppers with the items in the "lhs" column in their baskets to have chance of 35 to 39 percent to buy whole milk.

Get the top 10 baskets sorted by confidence.

```
##                                lhs                rhs    support confidence
## 1    {citrus fruit,root vegetables} {other vegetables} 0.01037112  0.5862069
## 2  {root vegetables,tropical fruit} {other vegetables} 0.01230300  0.5845411
## 3                      {curd,yogurt}       {whole milk} 0.01006609  0.5823529
## 4         {butter,other vegetables}       {whole milk} 0.01148958  0.5736041
## 5  {root vegetables,tropical fruit}       {whole milk} 0.01199797  0.5700483
## 6          {root vegetables,yogurt}       {whole milk} 0.01453991  0.5629921
## 7  {domestic eggs,other vegetables}       {whole milk} 0.01230300  0.5525114
## 8       {whipped/sour cream,yogurt}       {whole milk} 0.01087951  0.5245098
## 9      {rolls/buns,root vegetables}       {whole milk} 0.01270971  0.5230126
## 10     {other vegetables,pip fruit}       {whole milk} 0.01352313  0.5175097
##      coverage     lift count
## 1  0.01769192 3.029608   102
## 2  0.02104728 3.020999   121
## 3  0.01728521 2.279125    99
## 4  0.02003050 2.244885   113
## 5  0.02104728 2.230969   118
## 6  0.02582613 2.203354   143
## 7  0.02226741 2.162336   121
## 8  0.02074225 2.052747   107
## 9  0.02430097 2.046888   125
## 10 0.02613116 2.025351   133
```

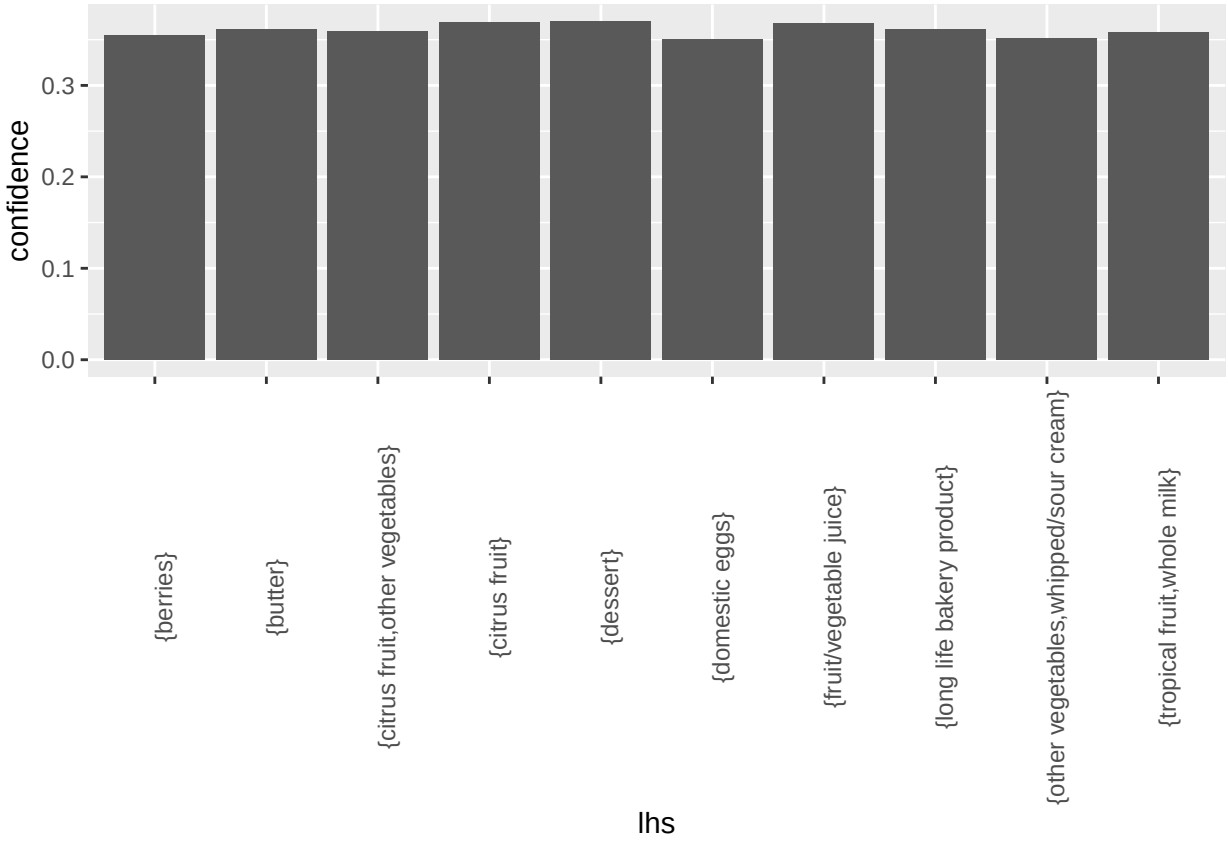


From the above table, we can see that the results are quite similar to what was observed previously. Top basket in this subset is the same as in the above. All confidence values are above 50 percent . This implies

that for the most of the time, shoppers with the items in the lhs column will end up purchasing the item in the rhs column. To add to this, as previously seen, whole milk showe up in a most of these top 10 baskets, which was expected.

Get the lowest 10 baskets sorted by confidence.

```
##                                       lhs                rhs    support
## 1                          {domestic eggs} {other vegetables} 0.02226741
## 2  {other vegetables,whipped/sour cream}           {yogurt} 0.01016777
## 3                                {berries}       {whole milk} 0.01179461
## 4            {tropical fruit,whole milk}           {yogurt} 0.01514997
## 5         {citrus fruit,other vegetables}  {root vegetables} 0.01037112
## 6              {long life bakery product}       {whole milk} 0.01352313
## 7                                 {butter} {other vegetables} 0.02003050
## 8                {fruit/vegetable juice}       {whole milk} 0.02663955
## 9                          {citrus fruit}       {whole milk} 0.03050330
## 10                              {dessert}       {whole milk} 0.01372649
##    confidence   coverage     lift count
## 1   0.3509615 0.06344687 1.813824   219
## 2   0.3521127 0.02887646 2.524073   100
## 3   0.3547401 0.03324860 1.388328   116
## 4   0.3581731 0.04229792 2.567516   149
## 5   0.3591549 0.02887646 3.295045   102
## 6   0.3614130 0.03741739 1.414444   133
## 7   0.3614679 0.05541434 1.868122   197
## 8   0.3684951 0.07229283 1.442160   262
## 9   0.3685504 0.08276563 1.442377   300
## 10  0.3698630 0.03711235 1.447514   135
```



From the above table, we can see the bottom 10 baskets based on confidence are quitesimilar to the bottom 10 baskets based on lift. Even though this table only shows the lowest confidence, all these are significant finds with at least 35 percent confidence that the rhs item is in the basket with the lhs items. Finally, the rhs contains whole milk, a product which many shoppers would already have in their baskets.