Title: STA 380N Intro to ML Take Home Exam 2

Author: Samarth Mishra (sm79247), Muskan Agarwal (ma64547), Sreekar Lanka (sl54387), Rishabh Tiwari (rt27739)

Date : 2022-08-15

Output : pdf_document

GitHub Repository Link : https://github.com/smjohn98/STA380_Take-Home-Exam-2

Probability Practice

Part A - 71.4%

Part B - 19.9%

Wrangling the Billboard Top 100

Part A:

```
##
## 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

## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v stringr 1.4.0
## v tidyr   1.2.0     v forcats 0.5.1
## v readr   2.1.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
##
## Attaching package: 'kableExtra'
##
##
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

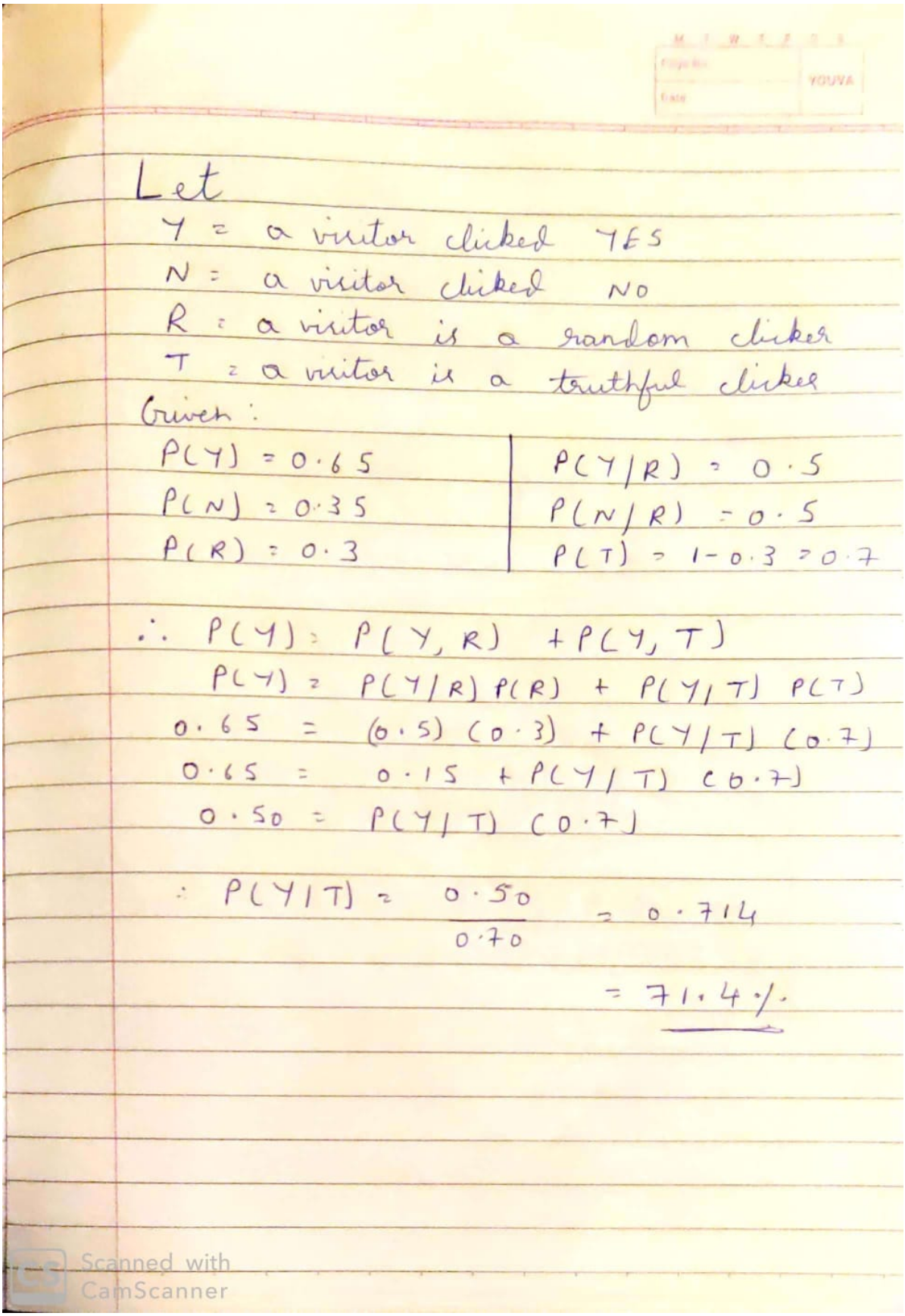


Figure 1: Caption for the picture.

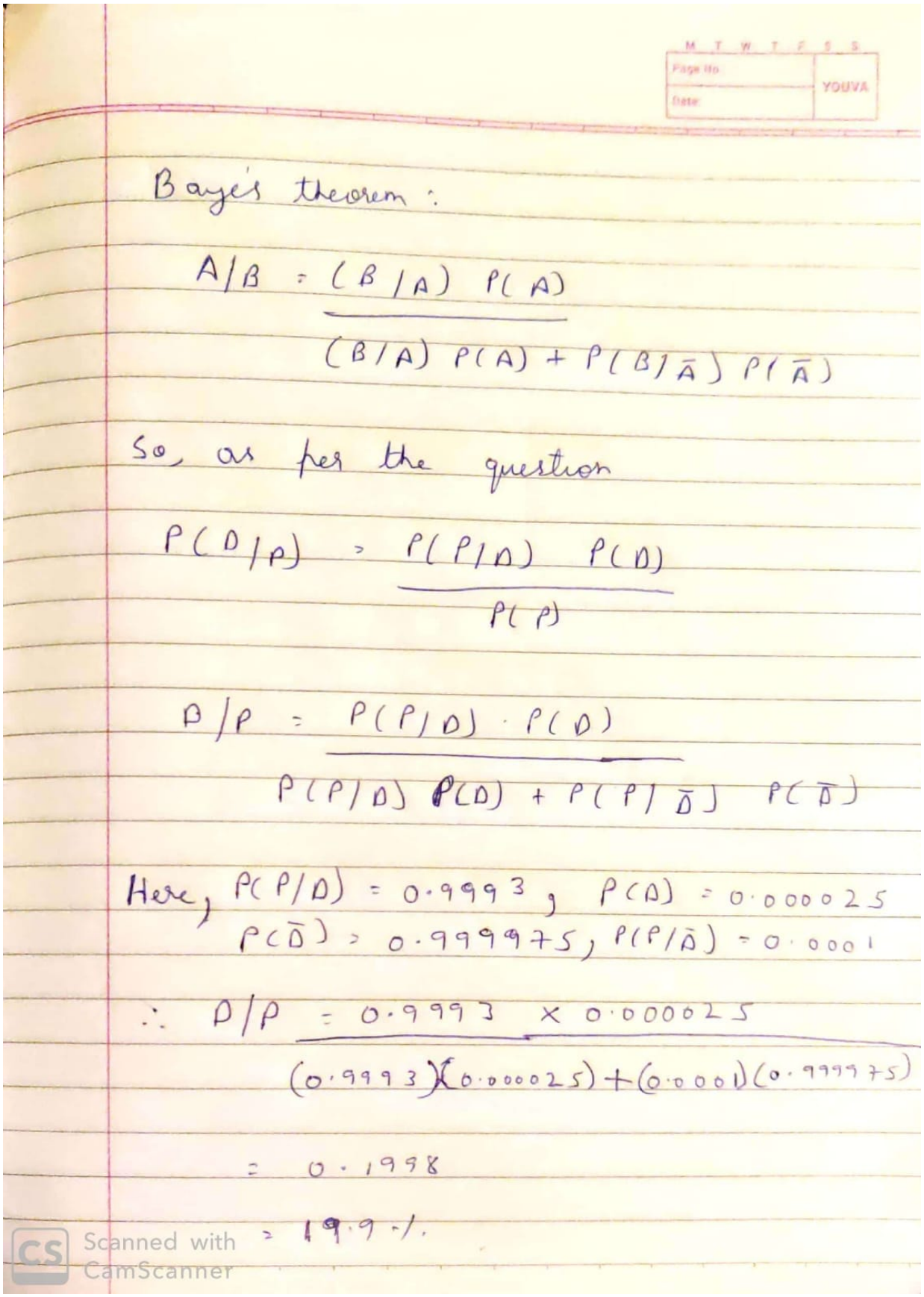
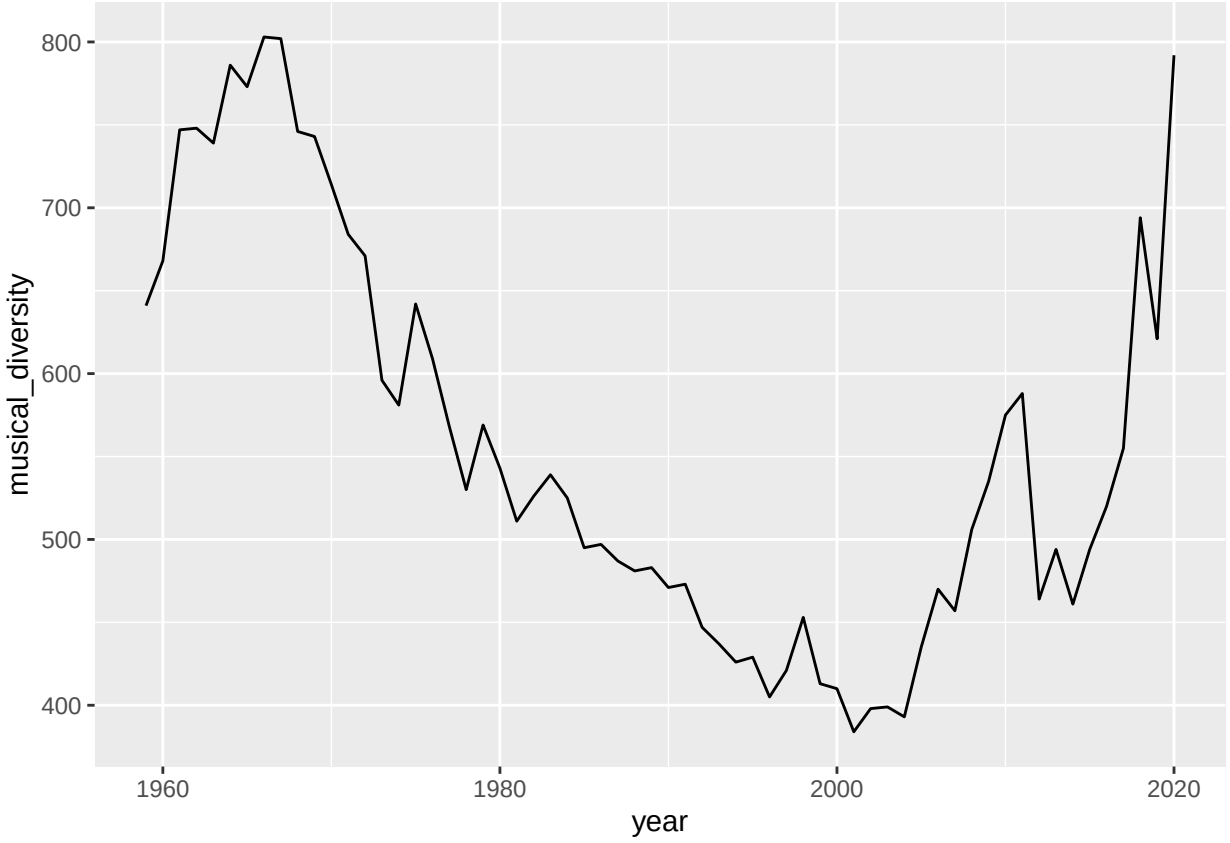


Figure 2: Caption for the picture.

Performer	Song	Count
Imagine Dragons	Radioactive	87
AWOLNATION	Sail	79
Jason Mraz	I'm Yours	76
The Weeknd	Blinding Lights	76
LeAnn Rimes	How Do I Live	69
LMFAO Featuring Lauren Bennett & GoonRock	Party Rock Anthem	68
OneRepublic	Counting Stars	68
Adele	Rolling In The Deep	65
Jewel	Foolish Games/You Were Meant For Me	65
Carrie Underwood	Before He Cheats	64

Imagine Dragons's Radioactive was the most popular song since 1958, appearing on the Billboard Top 100 for a total of 87 weeks.

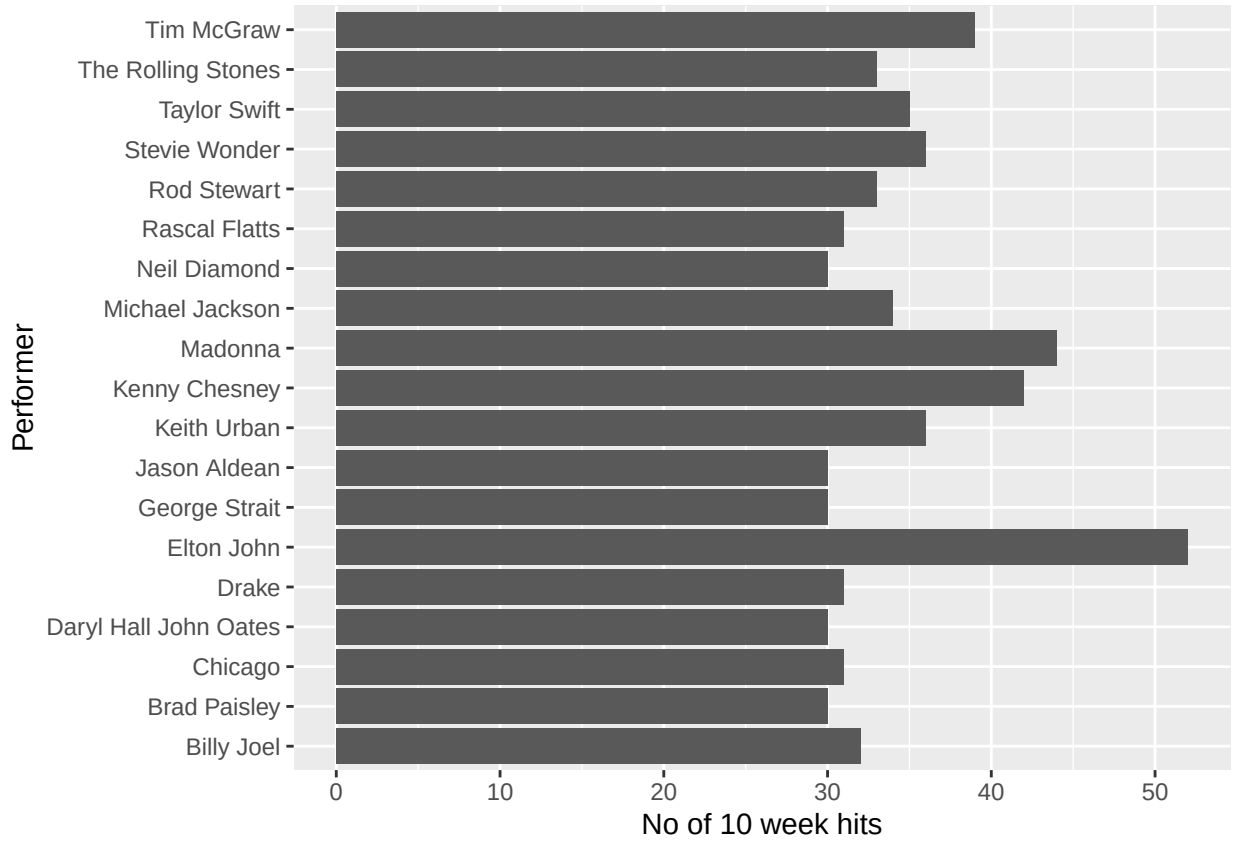
Part B:




```
## $x
## [1] "Year"
##
## $y
## [1] "Musical Diversity"
##
## $caption
## [1] "The musical diversity peaked in 1970 with about 800 unique songs that appeared in the Billboard
##
## attr(,"class")
```

```
## [1] "labels"
```

c) Part C



Visual story telling part 1: green buildings

Loading the libraries and reading the csv

```
## 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
```

Converting categorical columns to numerical

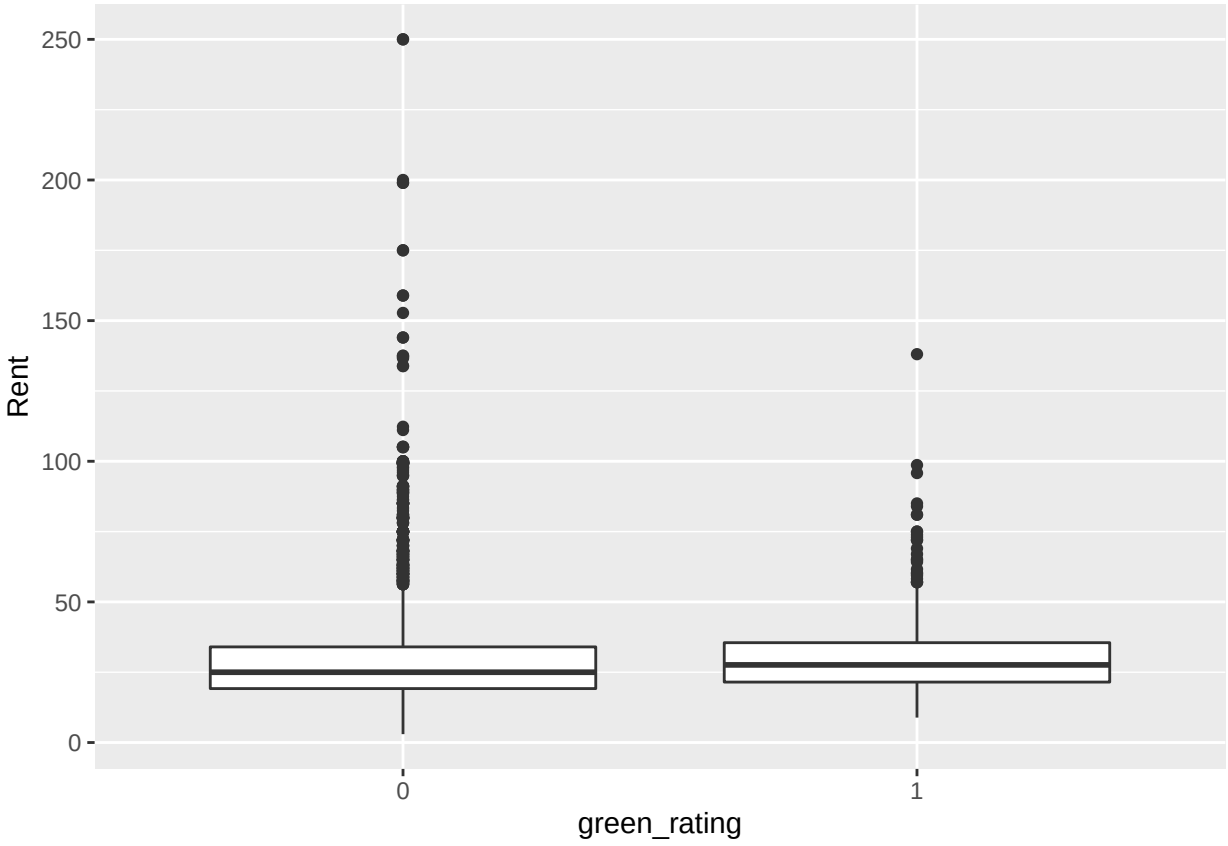
Creating a new column for class

Verifying first condition told by the guru

```
## # A tibble: 2 x 4
##   green_rating med_rent mean_rent count
##   <fct>           <dbl>     <dbl> <int>
## 1 0                25.0      28.4  6995
## 2 1                27.6      30.0   684
```

From above we can notice that the rent per square foot is more for buildings which have green rating

Plotting the data on a box plot to check for outliers:



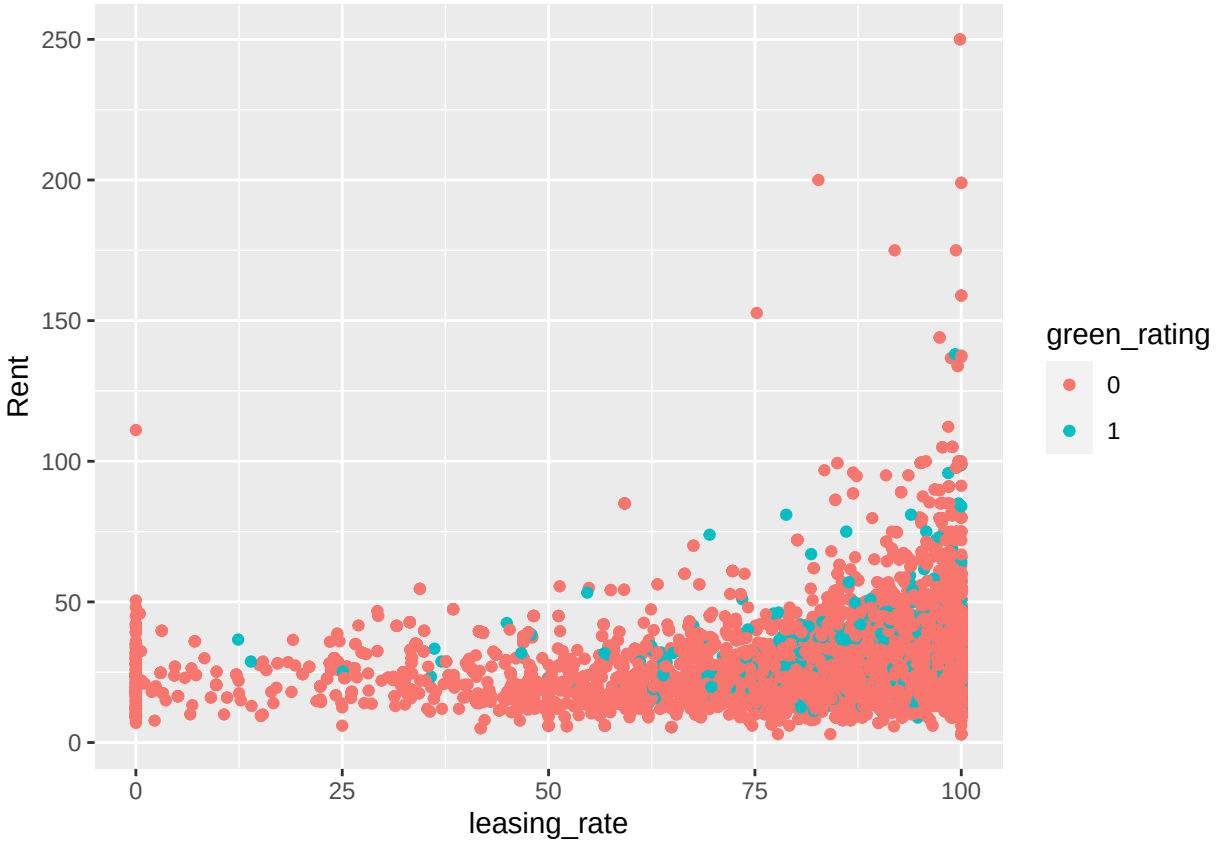
From the above box plot we notice that non - green buildings have more outliers hence it was a good decision by guru to take the median of the rent rather than mean.

Let us compare few variables below to cross check with the findings of guru

Age vs leasing rates

There does not seem to be any relationship between these two variables

green_building vs leasing rate



Here we can notice that until 70% the leasing rate does not vary too much but once it crosses 70% it starts increasing.

Let us check if their is any connection between the leasing rate of different types of buildings

```
## # A tibble: 2 x 3
##   green_rating median_rate count
##   <fct>              <dbl> <int>
## 1 0                   89.2  7209
## 2 1                   92.9   685
```

Here from the table we can notice that the leasing rate is higher for green buildings but I think it will be hard to verify this because we have small amount of data. Therefore guru's concept of 90% occupancy seems to be aligned with the data.

Search for confounding factors:

1) Size

```
## # A tibble: 2 x 3
##   green_rating median_size count
##   <fct>              <int> <int>
## 1 0                 118696  7209
## 2 1                 241150   685
```

From above table we can analyze that green buildings have larger area so it can be a good confounding factor.

Lets have a look at the plot:

From the plot we can see that as he size increases the rent also increases.

2) Amenities

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

```
## # A tibble: 4 x 4
## # Groups:   green_rating [2]
##   green_rating amenities `median(Rent)` count
##   <fct>        <fct>              <dbl> <int>
## 1 0            0                   25    3550
## 2 0            1                   25    3659
## 3 1            0                   27     187
## 4 1            1                   27.8   498
```

There is no deviation in the rent whether the amenities are included or not. So this cannot be a confounding variable.

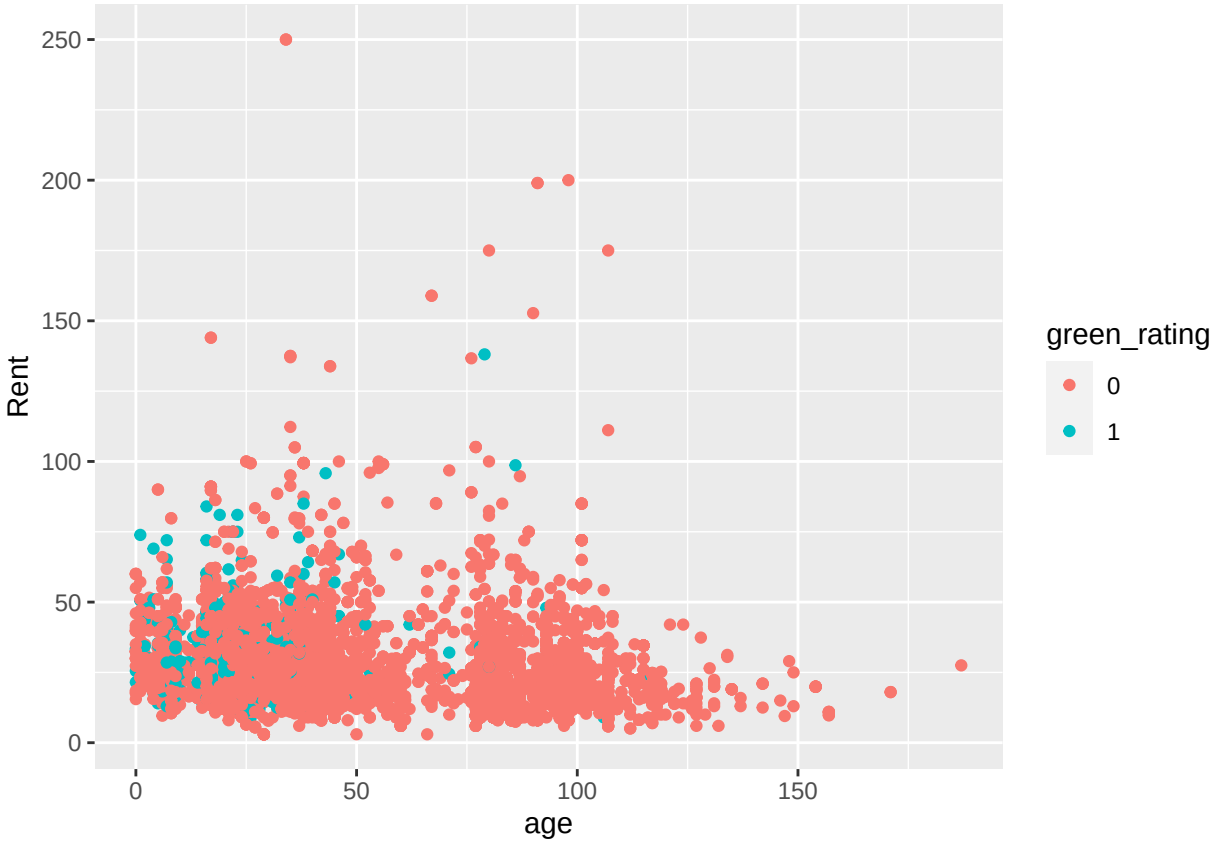
3) Age

```
## # A tibble: 2 x 4
##   green_rating `median(Rent)` `median(age)` count
```

```
##   <fct>                 <dbl>         <int> <int>
## 1 0                      25             37  7209
## 2 1                      27.6           22   685
```

There is no deviation in the rent with respect to the age of the building. Hence it cannot be a confounding variable

Let us plot it and observe for any trend



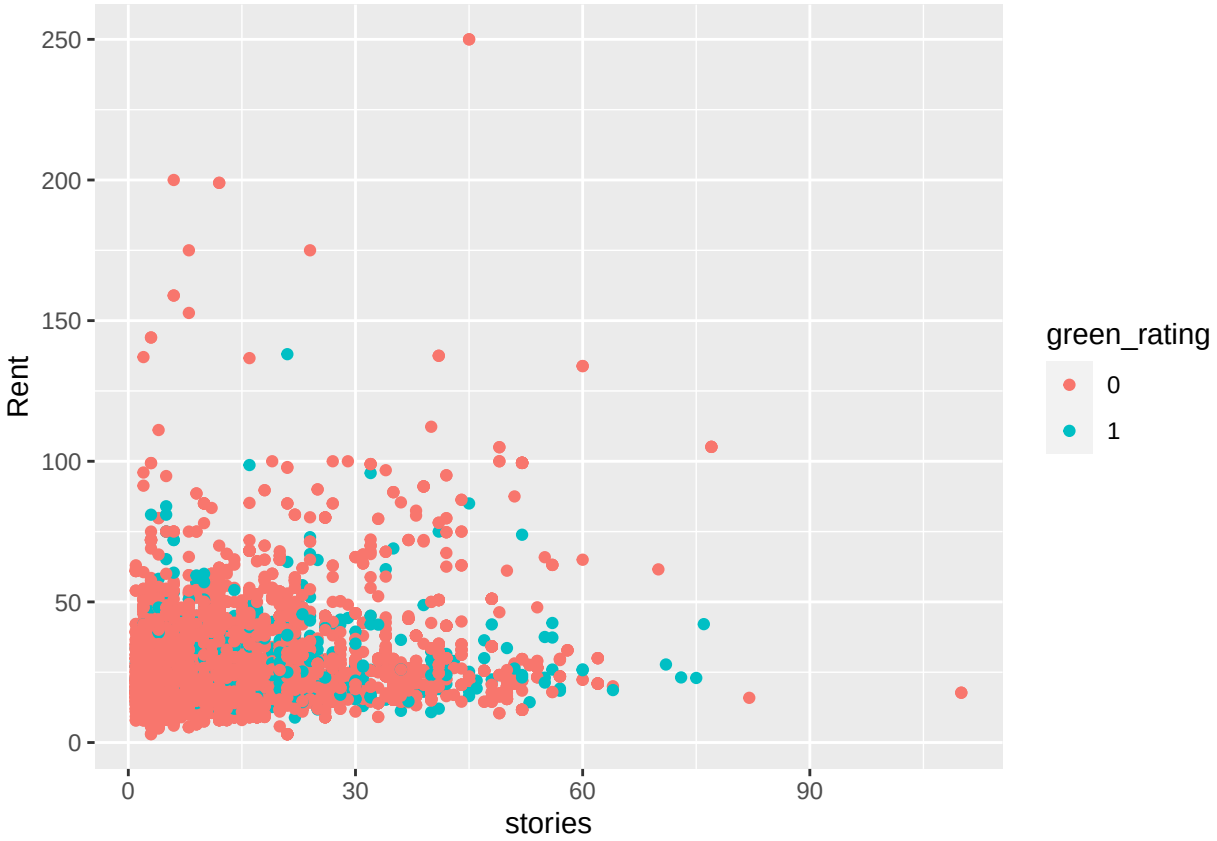
There is not possible trend that can be observed from the plot

4) Stories

```
## # A tibble: 2 x 4
##   green_rating `median(stories)` `median(Rent)` count
##   <fct>                    <int>          <dbl> <int>
## 1 0                           10           25    7209
## 2 1                           11           27.6   685
```

Green buildings tend to have one floor extra but there is not difference between in the median rent

Let us look at the plot

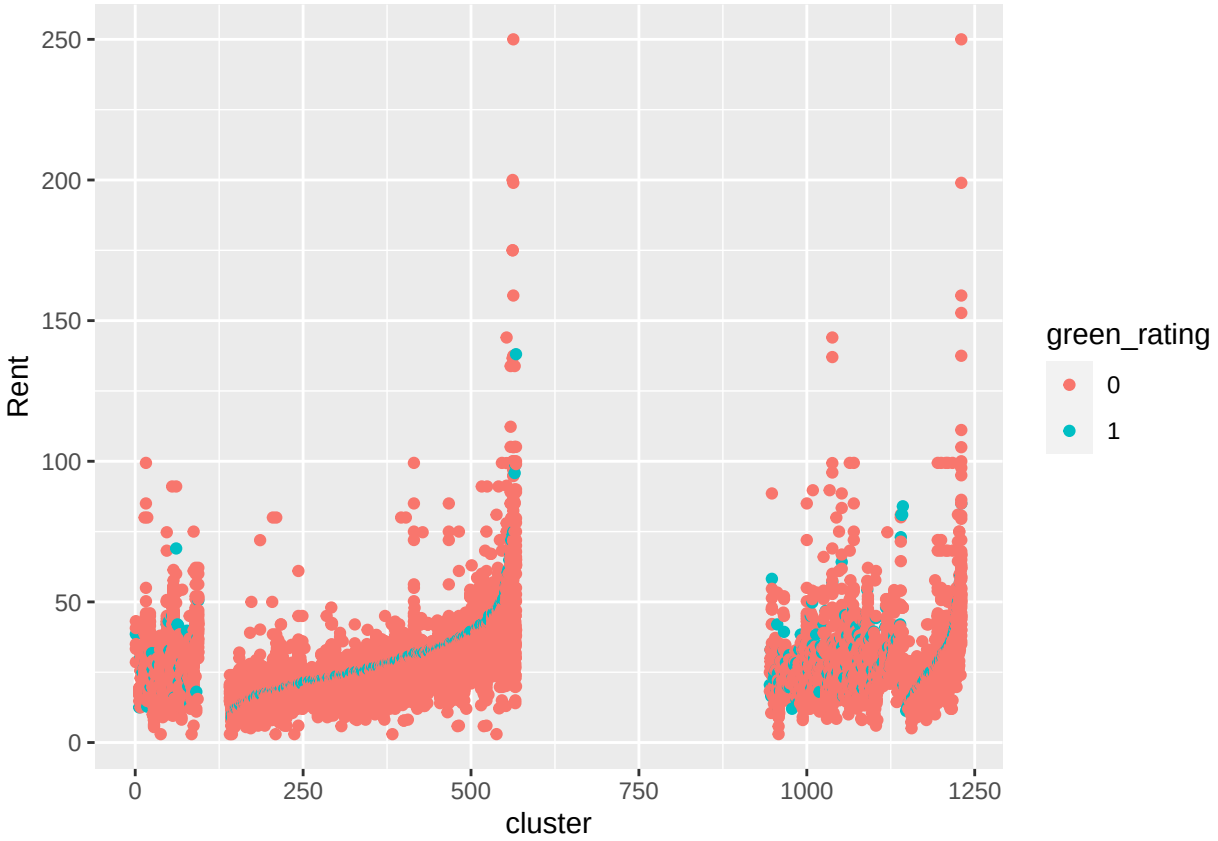


As per the plot we can notice that there is slight increase as you increase number of stories

Recommendations :

```
## # A tibble: 2 x 3
##   green_rating `median(Rent)` count
##   <fct>                 <dbl> <int>
## 1 0                      34.2  1492
## 2 1                      39.4   131
```

Lets look at the plot



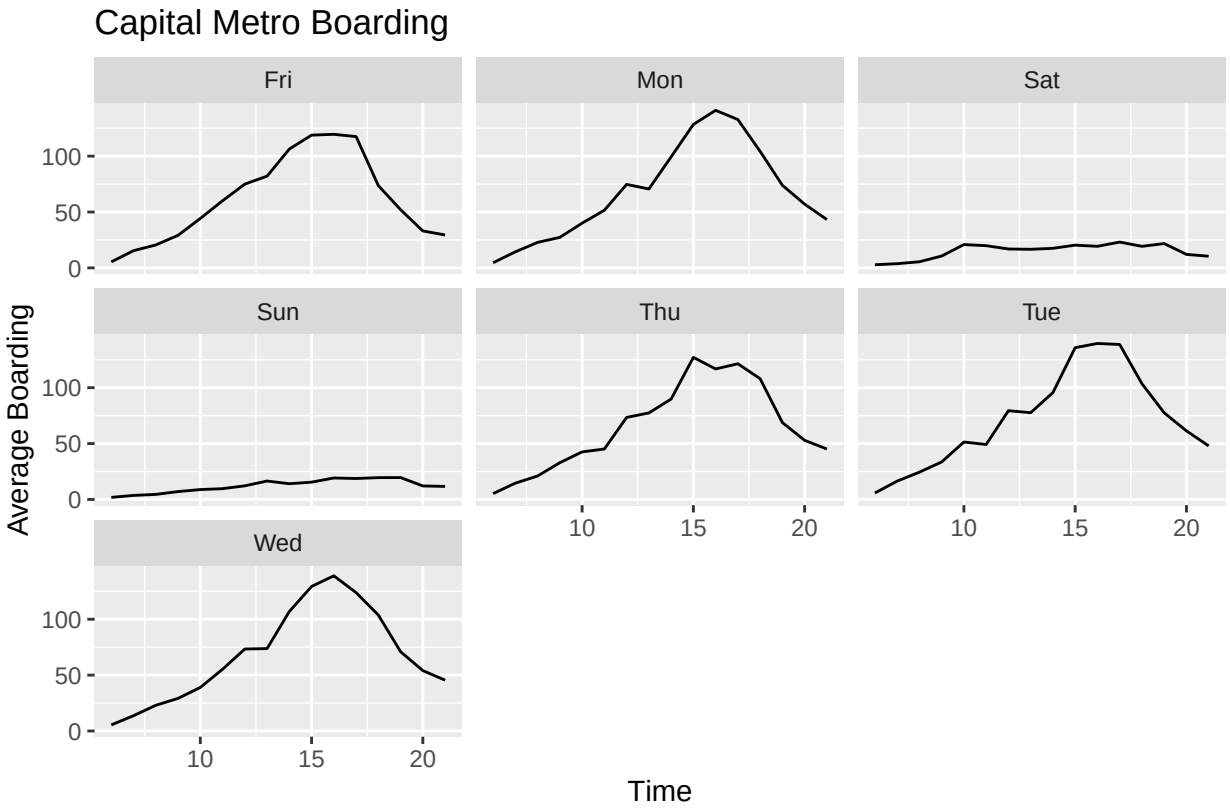
There is a difference of $5 between green and non-green buildings for clusters 430 - 600, possibly due to increased environmental awareness. We could thus generate an additional revenue of $5 * 250,000 = $1.25 million per year

Key Takeaways :

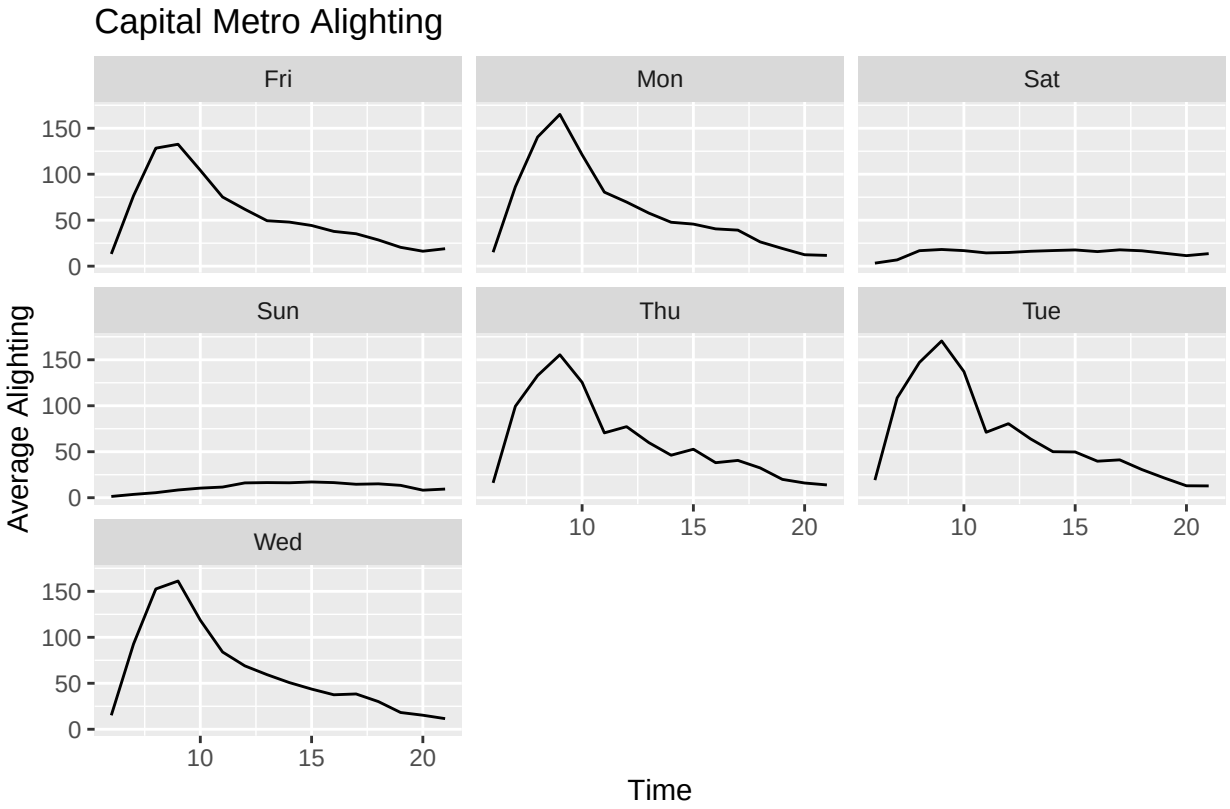
- There is an additional $2.6 in revenue for green buildings and it goes upto$5 for clusters 430 - 600.
- There is a positive relatio bwtween rent and occupancy rates. Green buildings have a better rate of occupancy
- There is no relation between age and occupancy rates.
- Also we found that rent and size have a subtle positive relationship. Green buildings 100,000 sq. ft. larger than non-green buildings, so this can be a confounding factor.

Guru's assumptions of being able to recuperate costs in around 8-9 years seems to be true. The average age for green buildings is currently 22 years, as the popularity of green buildings increase, we expect them to age and thus, the guru's assumption that the building would generate revenue for 30 years holds true.

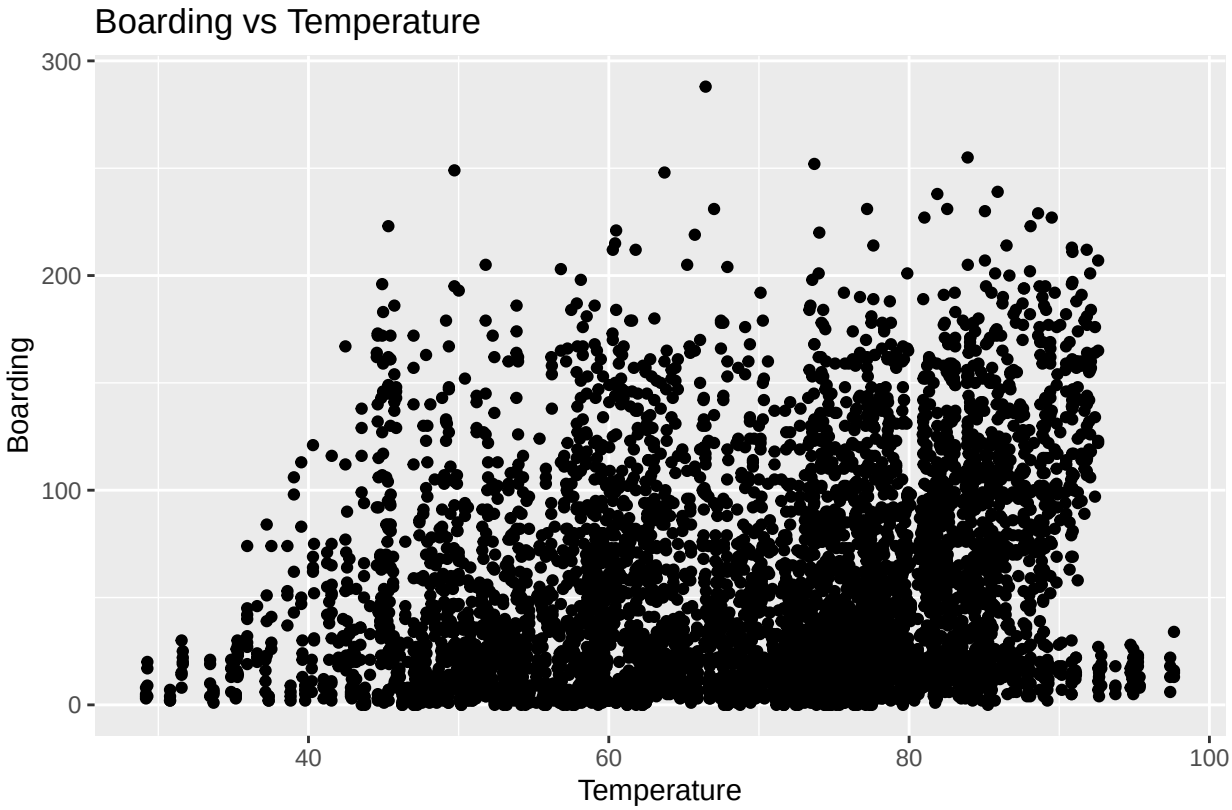
Visual story telling part 2: Capital Metro data



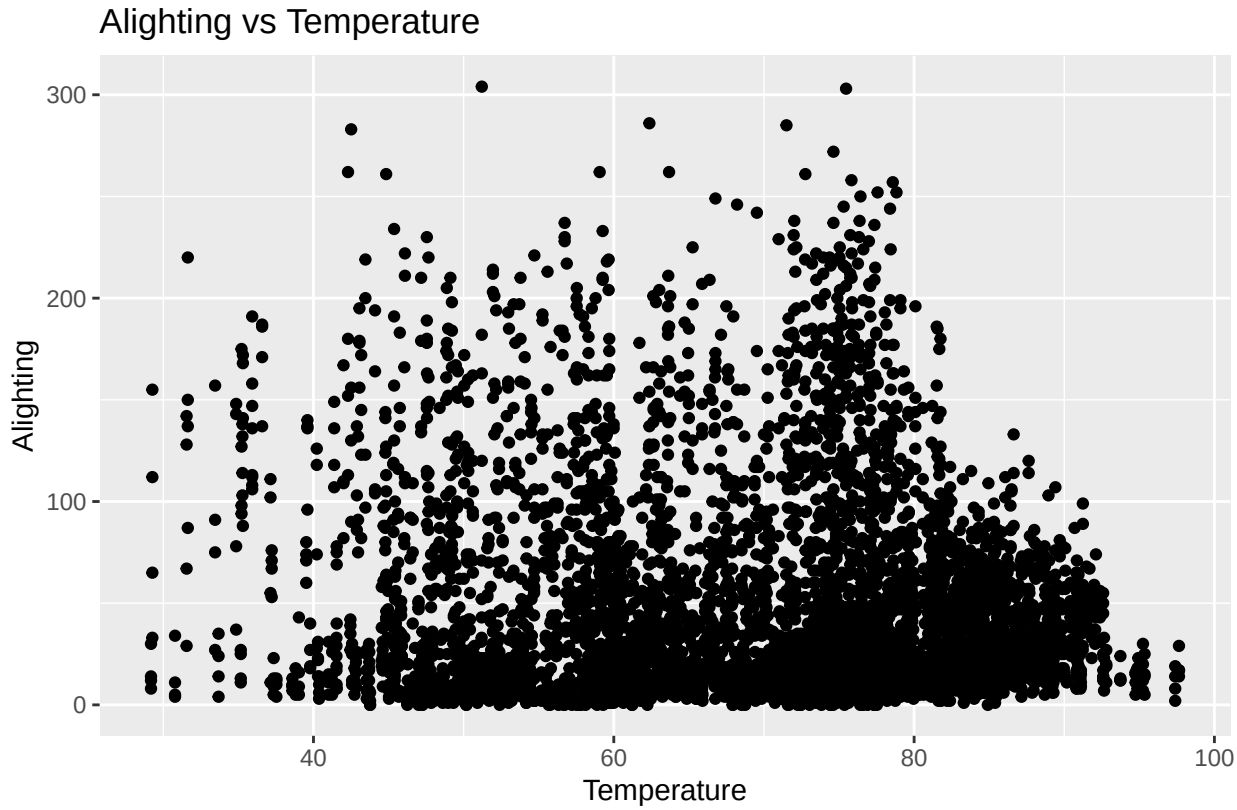
g steadily increases after 6:00 on all working days peaking at around 15:00−16:00 while being very low on weekends



ighting steadily increases after 6:00 peaking at around 8:00−9:00 on all weekdays while being very low on weekends



From the scatter plot above, the boarding looks independent of the temperature



From the scatter plot above, the alighting also looks independent of the temperature

Portfolio Modeling

Safe Portfolio (Low Risk ETF's)

VOO - Vanguard 500 Index Fund ETF - Invests in stocks in the S&P 500 Index, representing 500 of the largest U.S. companies

SCHA - Schwab US Small-Cap ETF - The fund's goal is to track as closely as possible, before fees and expenses, the total return of the Dow Jones U.S. Small-Cap Total Stock Market Index.

VXUS - Vanguard Total International Stock Index Fund ETF - Seeks to track the performance of the FTSE Global All Cap ex US Index, which measures the investment return of stocks issued by companies located outside the United States

VWO - Vanguard Emerging Markets Stock Index Fund ETF - Invests in stocks of companies located in emerging markets around the world, such as China, Brazil, Taiwan, and South Africa.

AOR - iShares Core Growth Allocation ETF - The investment seeks to track the investment results of the S&P Target Risk Growth Index composed of a portfolio of underlying equity and fixed income funds intended to represent a growth allocation target risk strategy.

High Risk ETF's

TQQQ- ProShares UltraPro QQQ - The index includes 100 of the largest domestic and international non-financial companies listed on The Nasdaq Stock Market based on market capitalization.

QLD - ProShares Ultra QQQ - ProShares Ultra QQQ seeks daily investment results, before fees and expenses, that correspond to two times (2x) the daily performance of the Nasdaq-100 Index

SPXL - Direxion Daily S&P 500 Bull 3X Shares - The Direxion Daily S&P 500® Bull (SPXL) and Bear (SPXS) 3X Shares seeks daily investment results, before fees and expenses, of 300%, or 300% of the inverse (or opposite), of the performance of the S&P 500® Index.

SPUU - Direxion Daily S&P 500 Bull 2X Shares ETF - The Direxion Daily S&P 500® Bull 2X Shares seeks daily investment results, before fees and expenses, of 200% of the performance of the S&P 500® Index.

SWAN - BlackSwan ETF - The BlackSwan ETF seeks investment results that correspond to the S-Network BlackSwan Core Index (the Index)

NTSX - WisdomTree U.S. Efficient Core Fund - The WisdomTree U.S. Efficient Core Fund* seeks total return by investing in large-capitalization U.S. equity securities and U.S. Treasury futures contracts.

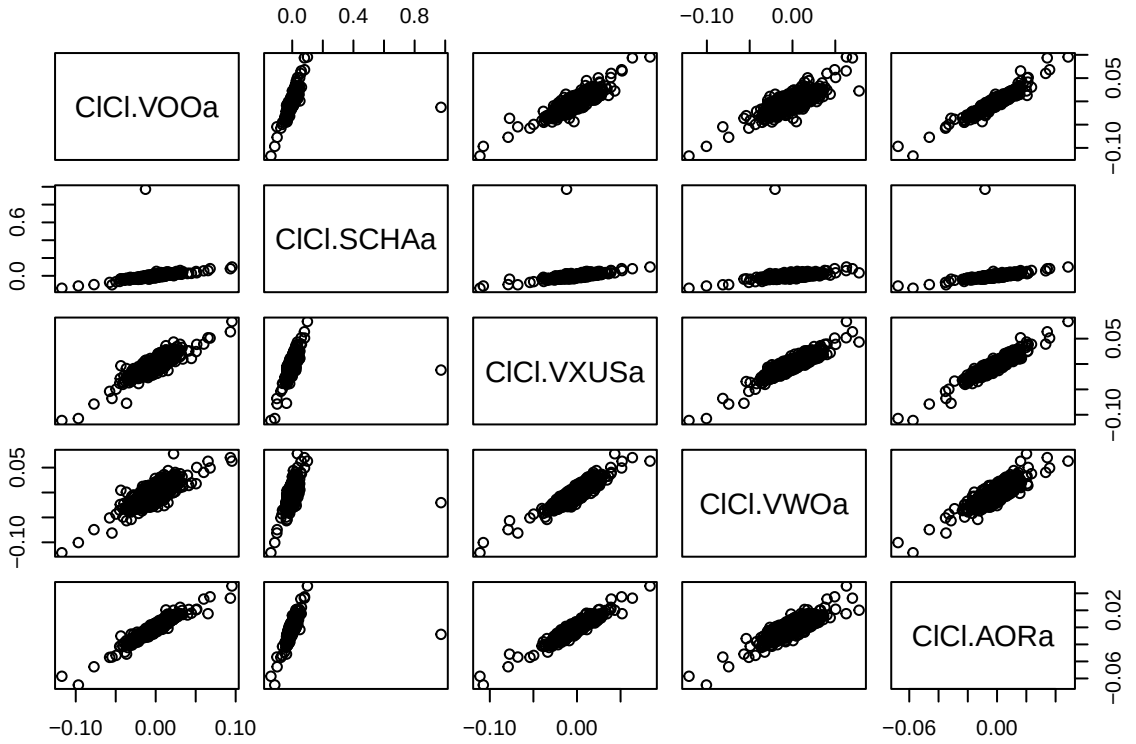
Medium Risk ETF's (Combination of High Risk and Safe ETF's)

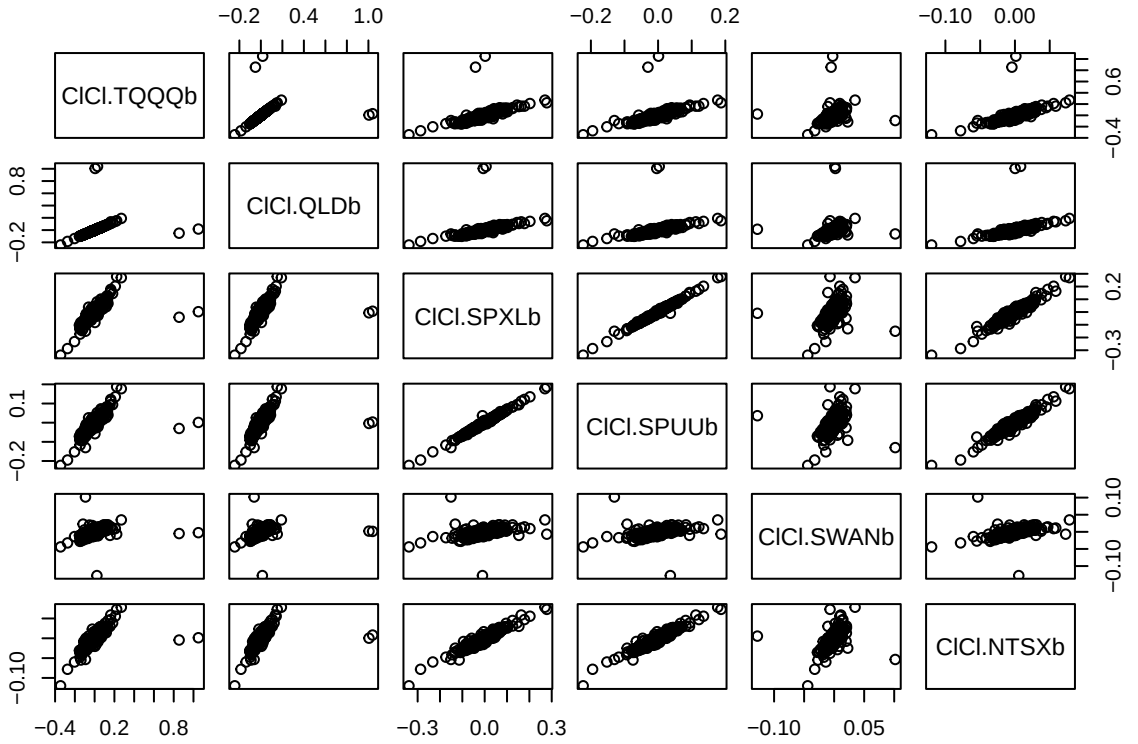
```
## Loading required package: xts

## 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

##
## Attaching package: 'foreach'

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

## pausing 1 second between requests for more than 5 symbols
## pausing 1 second between requests for more than 5 symbols
## pausing 1 second between requests for more than 5 symbols
## pausing 1 second between requests for more than 5 symbols
```

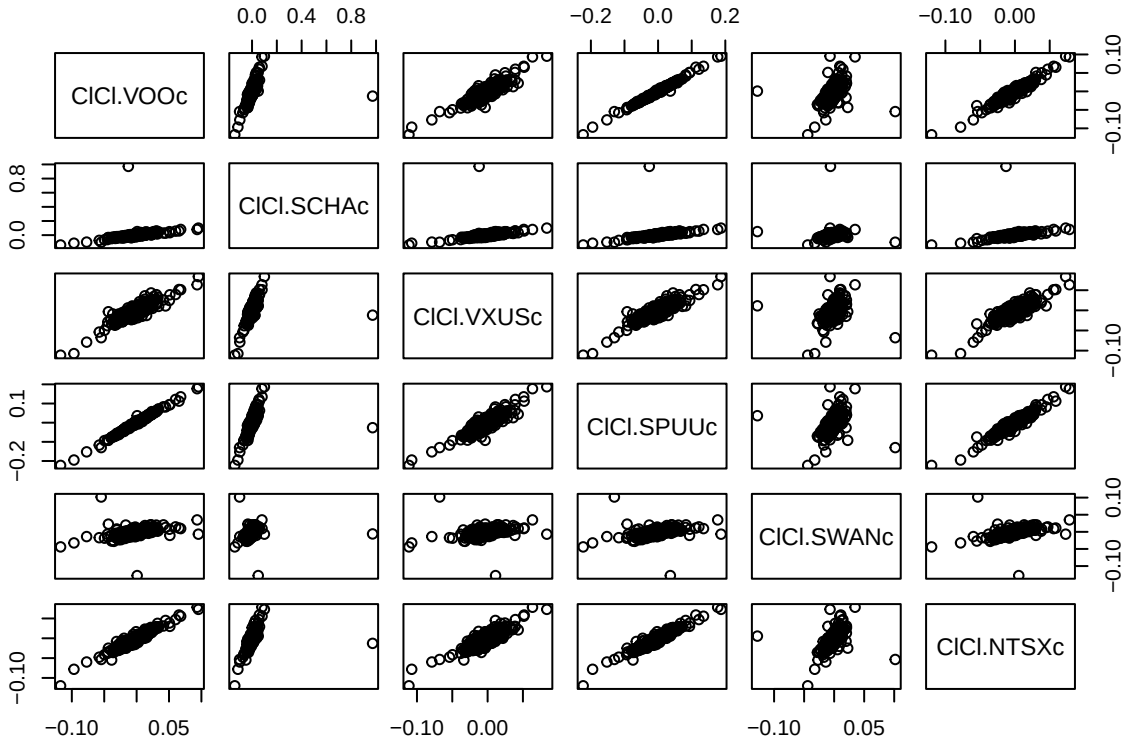
Combining close to close changes in a single matrix

```
##                ClCl.VOOa   ClCl.SCHAa    ClCl.VXUSa    ClCl.VWOa     ClCl.AORa
## 2014-01-03 -0.0008948816  0.003651038 -0.0003886514 -0.001259421  0.0010468726
## 2014-01-06 -0.0025077502 -0.006892629 -0.0029159798 -0.009583909 -0.0020915556
## 2014-01-07  0.0062253562  0.008097205  0.0048742444  0.004074357  0.0060257797
## 2014-01-08  0.0004163712  0.002103691 -0.0005821304 -0.001775273 -0.0026042447
## 2014-01-09  0.0005946721  0.001335878 -0.0029119978 -0.006351626 -0.0007832637
## 2014-01-10  0.0023177036  0.007051610  0.0103192755  0.019432319  0.0065325320

##               ClCl.TQQQb     ClCl.QLDb   ClCl.SPXLb   ClCl.SPUUb   ClCl.SWANb
## 2018-11-07  0.0919627589  0.0618173598  0.062333560  0.040594040  0.020351198
## 2018-11-08 -0.0170523752 -0.0122089532 -0.004802652 -0.007231228 -0.005475166
## 2018-11-09 -0.0508054523 -0.0339893813 -0.027905959 -0.008817309 -0.001966221
## 2018-11-12 -0.0878403581 -0.0591162630 -0.058061793 -0.045832509 -0.009850276
## 2018-11-13 -0.0004089143  0.0003776882 -0.005499473 -0.005472233 -0.005173060
## 2018-11-14 -0.0253630599 -0.0167399883 -0.020276543 -0.013653923  0.000000000
##              ClCl.NTSXb
## 2018-11-07  0.013185002
## 2018-11-08  0.005693371
## 2018-11-09 -0.006874282
## 2018-11-12  0.000000000
## 2018-11-13 -0.012214943
## 2018-11-14 -0.004122012

##               ClCl.VOOc   ClCl.SCHAc   ClCl.VXUSc   ClCl.SPUUc   ClCl.SWANc
## 2018-11-07  0.021121734  0.017453505  0.013417541  0.040594040  0.020351198
## 2018-11-08 -0.001704381 -0.003374634 -0.011292874 -0.007231228 -0.005475166
## 2018-11-09 -0.009234848 -0.015801298 -0.010437161 -0.008817309 -0.001966221
## 2018-11-12 -0.019307559 -0.019782167 -0.015920378 -0.045832509 -0.009850276
## 2018-11-13 -0.001677245 -0.002193536  0.005257796 -0.005472233 -0.005173060
## 2018-11-14 -0.006840297 -0.006888494  0.003017542 -0.013653923  0.000000000
##              ClCl.NTSXc
## 2018-11-07  0.013185002
## 2018-11-08  0.005693371
## 2018-11-09 -0.006874282
## 2018-11-12  0.000000000
## 2018-11-13 -0.012214943
## 2018-11-14 -0.004122012
```

Computing the returns from the closing prices


ClCl.VOOa
ClCl.SCHAa
ClCl.VXUSa
ClCl.VWOa
ClCl.AORa
0.0 0.4 0.8
−0.10 0.00
−0.10 0.00 0.10
−0.10 0.00
−0.06 0.00
0.0 0.6
−0.10 0.05
−0.10 0.05
−0.10 0.05
−0.06 0.02


ClCl.TQQQb
ClCl.QLDb
ClCl.SPXLb
ClCl.SPUUb
ClCl.SWANb
ClCl.NTSXb
−0.2 0.4 1.0
−0.2 0.0 0.2
−0.10 0.00
−0.4 0.6
−0.2 0.8
−0.3 0.2
−0.2 0.1
−0.10 0.10
−0.10
−0.4 0.2 0.8
−0.3 0.0 0.3
−0.10 0.05



Simulate a random day

Update the value of my holdings

```
## [1] 100188.2
```

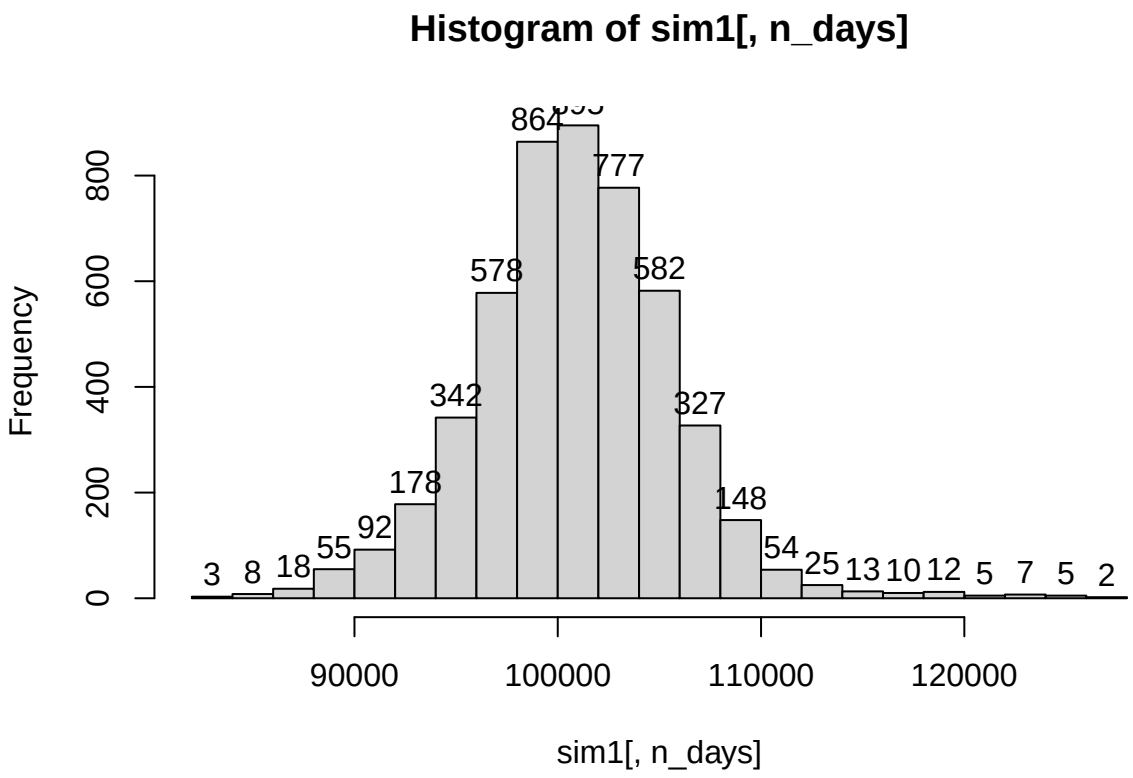
```
## [1] 89938.29
```

```
## [1] 114313.4
```

Portfolio -1 for Safe portfolio

```
##               [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
## result.1  96989.56  97124.44  96977.63  98159.95  97007.04  97699.66  97378.65
## result.2 100249.19 100143.85 100462.15 101908.46 101585.12 101054.26 100877.55
## result.3  99843.25  99956.04 100544.90 100428.07 100948.39  97168.72  95869.69
## result.4 100708.54 100084.93 100875.21 101032.49 100905.35 100540.44 100969.21
## result.5  99328.76 100425.11  99007.23  98721.18  98237.31  99207.03  99821.11
## result.6  99451.48  99492.14  97737.74  97923.86  98295.11  97973.80  97808.69
##               [,8]      [,9]     [,10]     [,11]     [,12]     [,13]     [,14]
## result.1  97083.72  97287.27  97264.65  96838.67  98007.35  97738.22  97584.62
## result.2 101885.94 102411.08 102050.00 102101.03 101194.86 101416.26 102665.27
## result.3  97492.31  97588.82  98460.69  98739.58  98606.79 100218.72  99960.60
## result.4 101497.10 101630.01 102324.68 101948.23  99539.85  99473.06 100265.76
## result.5 100758.69 100436.31 100862.48 100556.65 101167.75 100844.00 100934.05
## result.6  95612.89  96136.17  97136.23  98150.98  98314.69  98939.17  97659.89
```

```
##              [,15]     [,16]     [,17]     [,18]     [,19]     [,20]
## result.1  98479.61  98835.48  99173.55  98763.89  97902.83  98907.77
## result.2 102255.07 101487.62 101291.27 102260.49 101689.15 102331.68
## result.3 101263.79 100840.13 101013.82 102857.67 102741.16 102316.73
## result.4 100590.74  99833.01  99586.46  99331.03  99849.96  99801.90
## result.5 100885.42 100756.75 101070.20 100874.33 101176.63 101537.55
## result.6  98082.28  99305.93 100045.83  99624.66 100005.49 100152.12
```



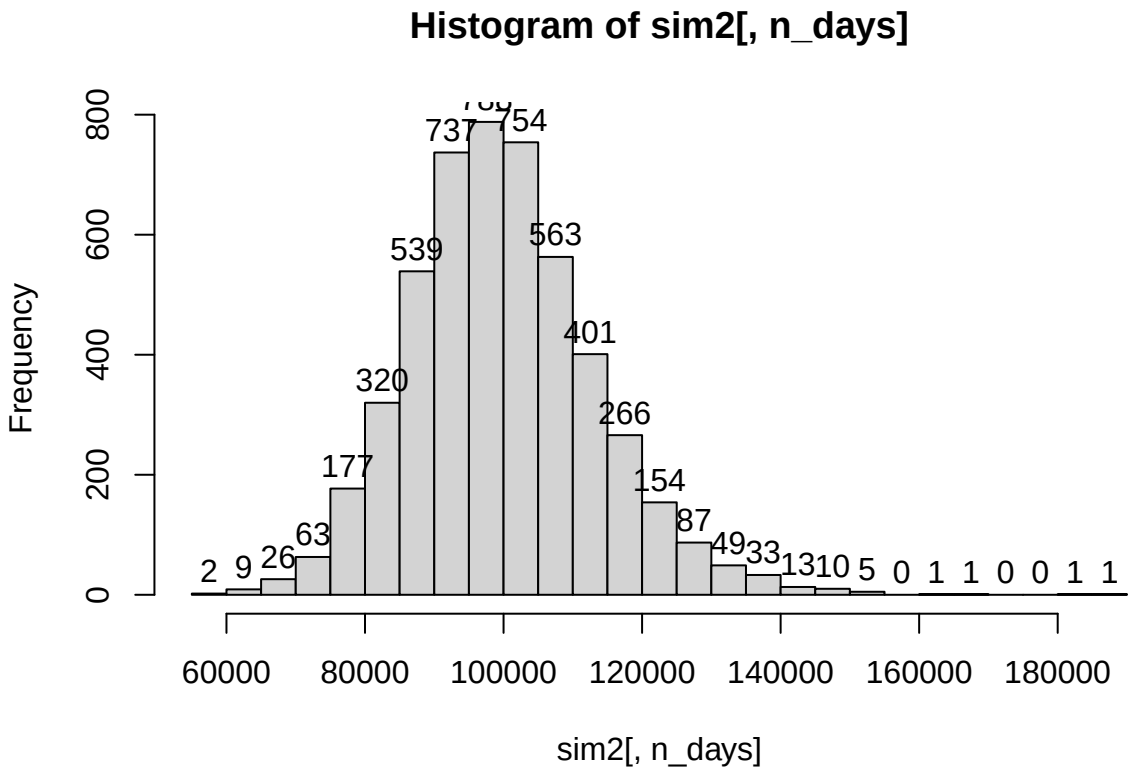

```
##        5%
## -6964.623
```

The VAR value of Safe Portfolio is :

Portfolio - 2 for High Risk ETF's

```
##              [,1]     [,2]     [,3]      [,4]      [,5]      [,6]      [,7]
## result.1 95902.93 96456.95 96928.37  98400.38 100193.86 100138.13  95901.68
## result.2 94280.54 92879.41 97641.69 102096.85 102805.52 101104.60 102418.63
## result.3 94867.48 96038.62 88476.79  86446.71  81199.69  82675.78  83281.85
## result.4 98883.04 97711.66 91282.11  92936.93  94170.53  93433.07  92594.97
## result.5 96529.11 96828.53 97117.63  98328.66  98344.05 101130.12 104603.64
## result.6 97802.05 98761.59 98313.18  98225.04  98550.60 101457.57 101403.73
##               [,8]      [,9]     [,10]     [,11]     [,12]     [,13]     [,14]
## result.1  92729.61  93244.74  93599.03  93956.11  94863.38  95615.69  95145.29
## result.2 101306.80 104183.26 101906.24 105399.12 102029.61 103041.99 103669.97
```

```
## result.3  84081.97  83343.35  81836.89  78506.64  79673.13  79949.55  80114.45
## result.4  92102.65  94113.51  93471.38  91700.42  90960.06  90894.74  92048.06
## result.5 106741.67 105258.67 104233.27 100782.43 103230.17 104373.22 105803.67
## result.6 101906.92 100129.07  98770.11 100476.61 104800.27 104511.17  95372.50
##              [,15]     [,16]     [,17]     [,18]     [,19]     [,20]
## result.1  95351.19  95271.42  98543.47 100248.82  99774.10  99854.95
## result.2 104274.99 103985.59 121959.49 123445.94 124470.60 128022.87
## result.3  82400.25  82087.25  82020.26  82777.00  82059.74  83484.72
## result.4  92559.58  92374.08  90007.00  91472.82  92426.35  93664.52
## result.5 106872.54 103416.18 107190.56 110287.38 106925.70 109474.47
## result.6  95441.57  96351.14  97293.03  97787.83  96637.11 100150.42
```



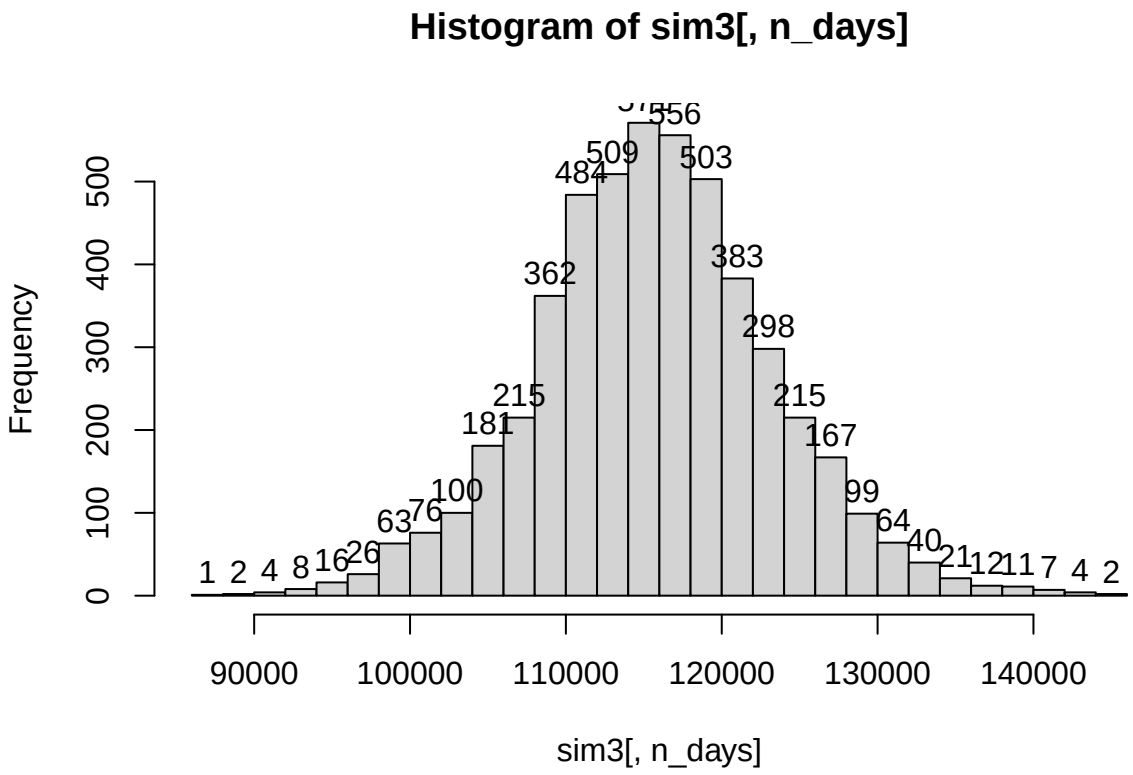

```
##       5%
## -20462.1
```

The VAR value of High Risk Portfolio is :

Portfolio - 3 for Medium Risk ETF's

```
##              [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]
## result.1 115069.4 112477.0 112738.8 112442.4 112737.5 111895.6 114110.7
## result.2 115026.6 114385.5 115469.3 116192.9 116124.0 116280.8 116614.5
## result.3 113591.3 112775.8 113617.2 112879.6 112102.3 112610.6 113017.9
## result.4 115444.2 114162.8 111628.6 112211.0 112352.6 111046.9 111106.9
## result.5 114857.1 115955.0 114521.6 114669.8 114385.2 113595.0 114080.6
```

```
## result.6 111987.4 111146.8 110575.3 110335.5 112206.6 111907.9 111871.1
##              [,8]     [,9]    [,10]    [,11]    [,12]     [,13]     [,14]
## result.1 114333.0 111903.8 113779.1 113558.6 112117.0 112284.91  98414.06
## result.2 118073.8 111127.0 109390.7 110794.8 110587.3 117622.67 117398.75
## result.3 113129.4 113359.9 111440.2 111621.3 100514.9  99988.22  99901.44
## result.4 110091.4 110692.8 110165.6 110166.5 109373.5 111217.55 112029.12
## result.5 114960.9 115741.0 116452.8 117213.0 117972.8 118228.74 118459.20
## result.6 112088.4 111282.8 112502.4 103882.9 104383.8 104541.92 108296.88
##              [,15]    [,16]    [,17]     [,18]     [,19]     [,20]
## result.1  99289.33 100209.4 100008.2  99835.33  99024.85  98965.33
## result.2 118416.86 118346.1 118491.8 119085.73 113689.22 115536.08
## result.3 102034.43 103262.0 102888.4 100847.80 100008.53  99610.07
## result.4 112245.63 112926.3 113548.0 113412.50 114368.93 114967.80
## result.5 122129.09 123689.8 124200.6 124107.46 125277.10 125968.46
## result.6 108270.38 107624.5 107395.8 107010.69 108118.53 108166.71
```




```
##       5%
## 3098.255
```

The VAR value of Medium Risk Portfolio is :

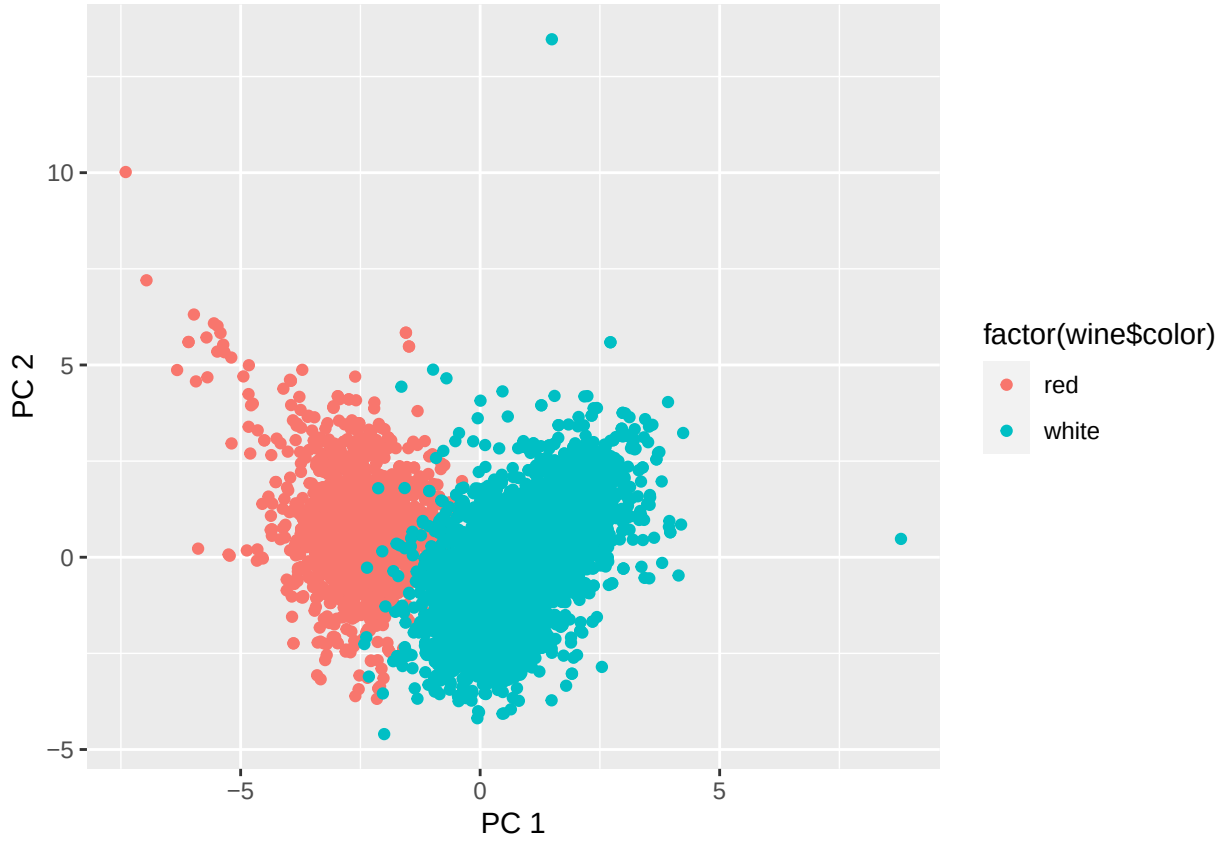
Clustering and PCA

PCA

```
## Importance of components:
##                            PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.7407 1.5792 1.2475 0.98517 0.84845 0.77930 0.72330
## Proportion of Variance 0.2754 0.2267 0.1415 0.08823 0.06544 0.05521 0.04756
## Cumulative Proportion  0.2754 0.5021 0.6436 0.73187 0.79732 0.85253 0.90009
##                            PC8     PC9   PC10    PC11
## Standard deviation     0.70817 0.58054 0.4772 0.18119
## Proportion of Variance 0.04559 0.03064 0.0207 0.00298
## Cumulative Proportion  0.94568 0.97632 0.9970 1.00000
```

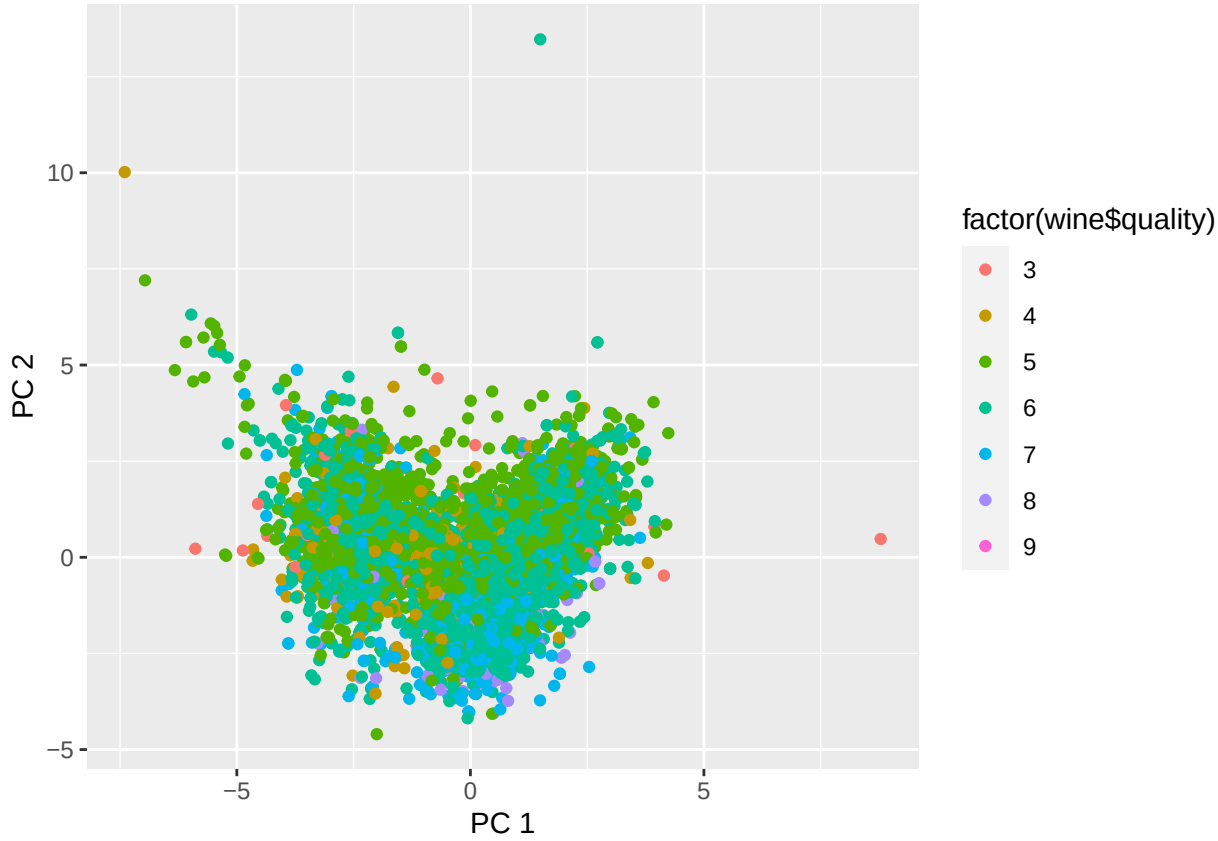
The first 2 principal components explain about half of the variance in the data. I will only take them into consideration when distinguishing between the wine color and quality

Distinguishing Red/ White Wine



We can see that the two principal components are superbly capable of distinguishing the red wine from the white wine.

Distinguishing Wine Quality



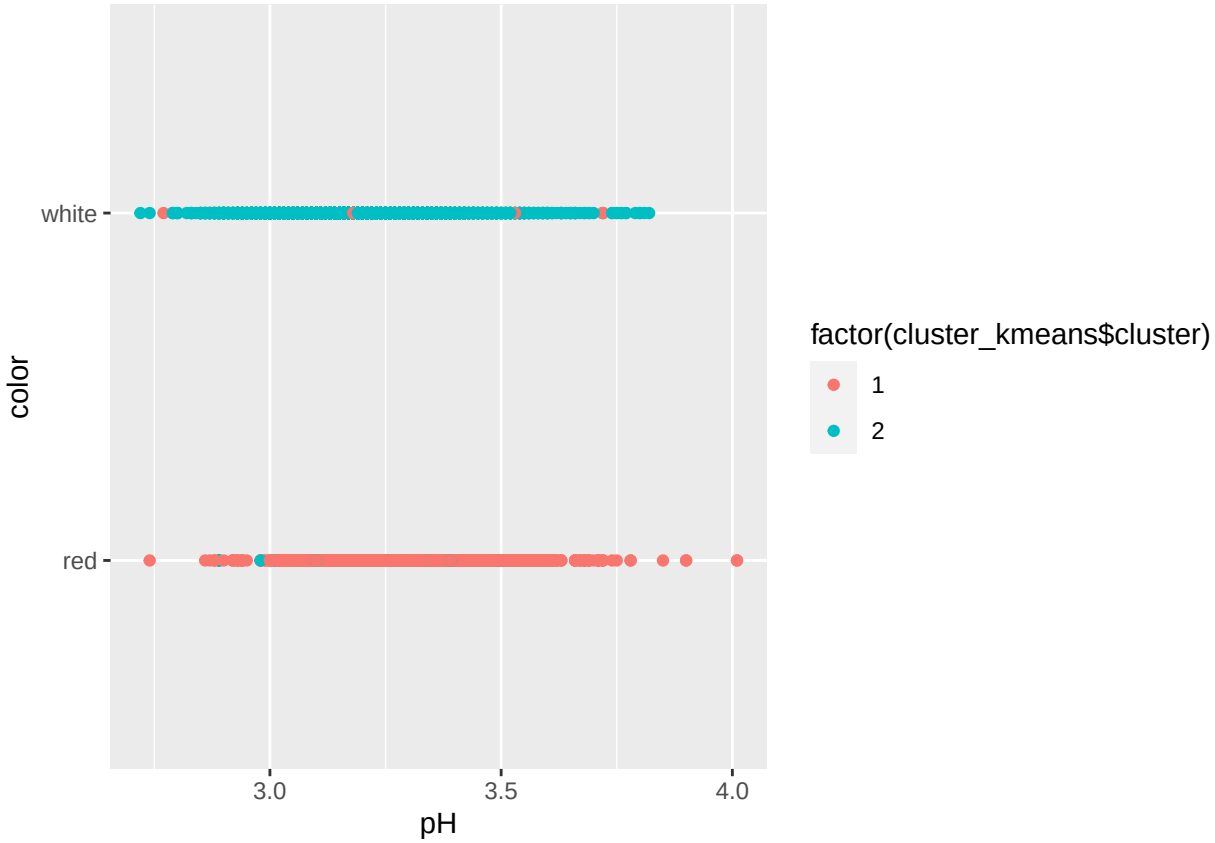
We see that it becomes hard to distinguish the wine quality using the two principal components

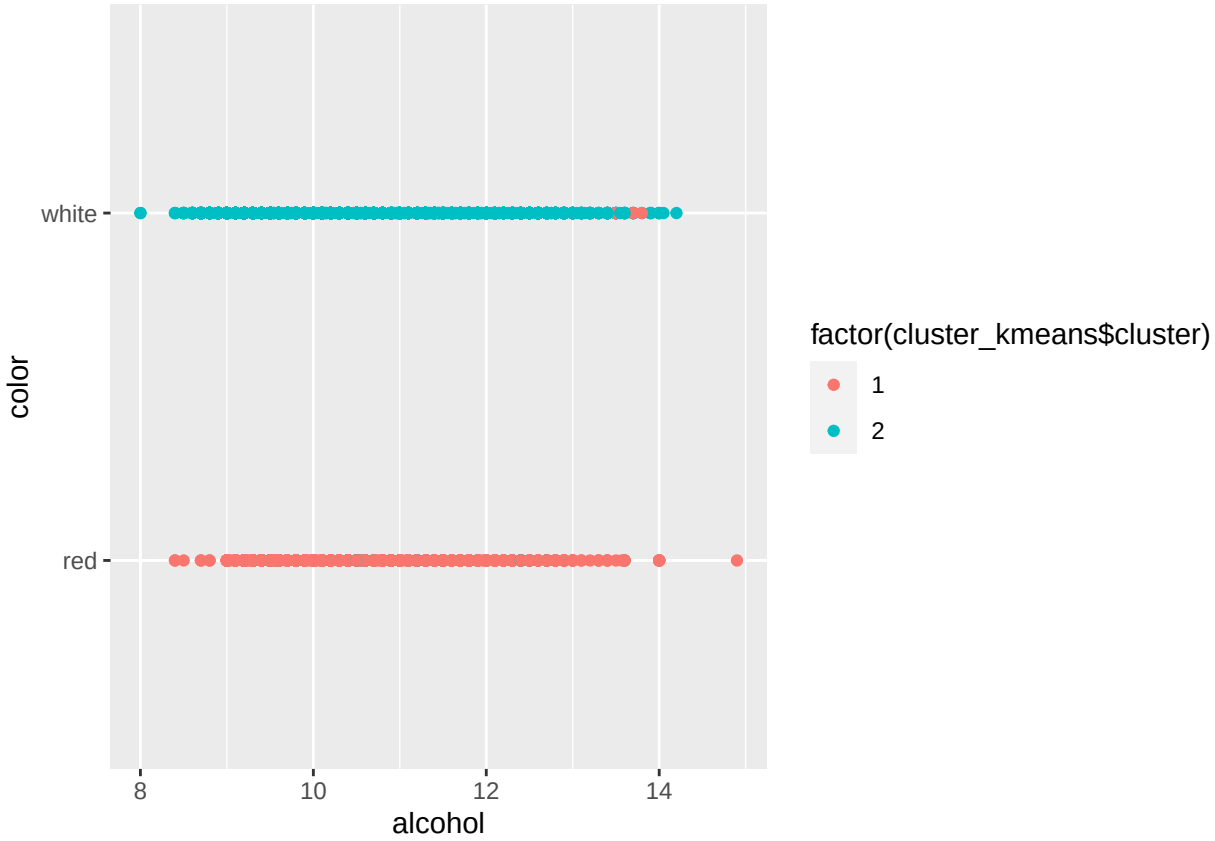
Clustering - K Means

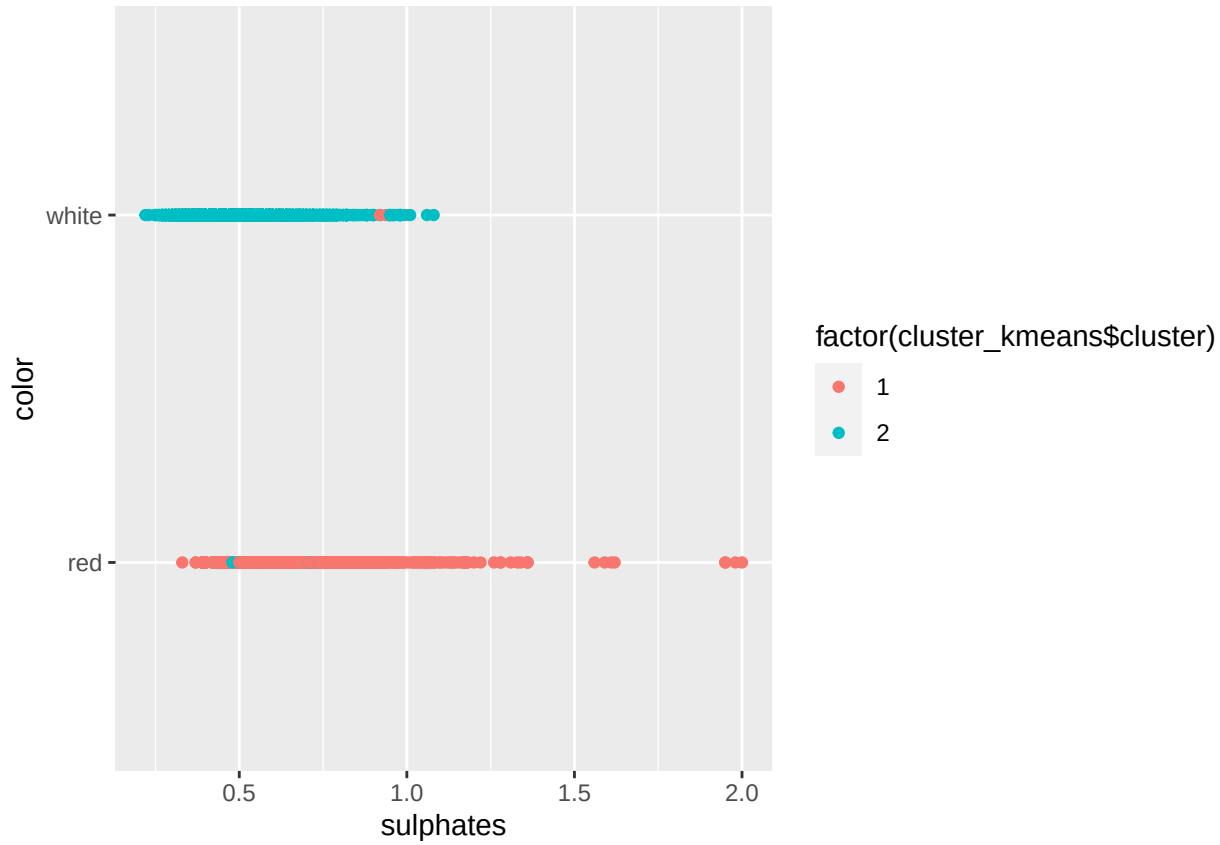
Distinguishing Red/ White Wine

We run a k-means model with k=2 (as we have two labels)

```
## Registered S3 methods overwritten by 'ggalt':
##   method                  from
##   grid.draw.absoluteGrob  ggplot2
##   grobHeight.absoluteGrob ggplot2
##   grobWidth.absoluteGrob  ggplot2
##   grobX.absoluteGrob      ggplot2
##   grobY.absoluteGrob      ggplot2

##
## Attaching package: 'ggalt'

## The following objects are masked from 'package:ggformula':
##
##     stat_ash, StatAsh
```


white
red
color
3.0
3.5
4.0
pH
factor(cluster_kmeans$cluster)
1
2

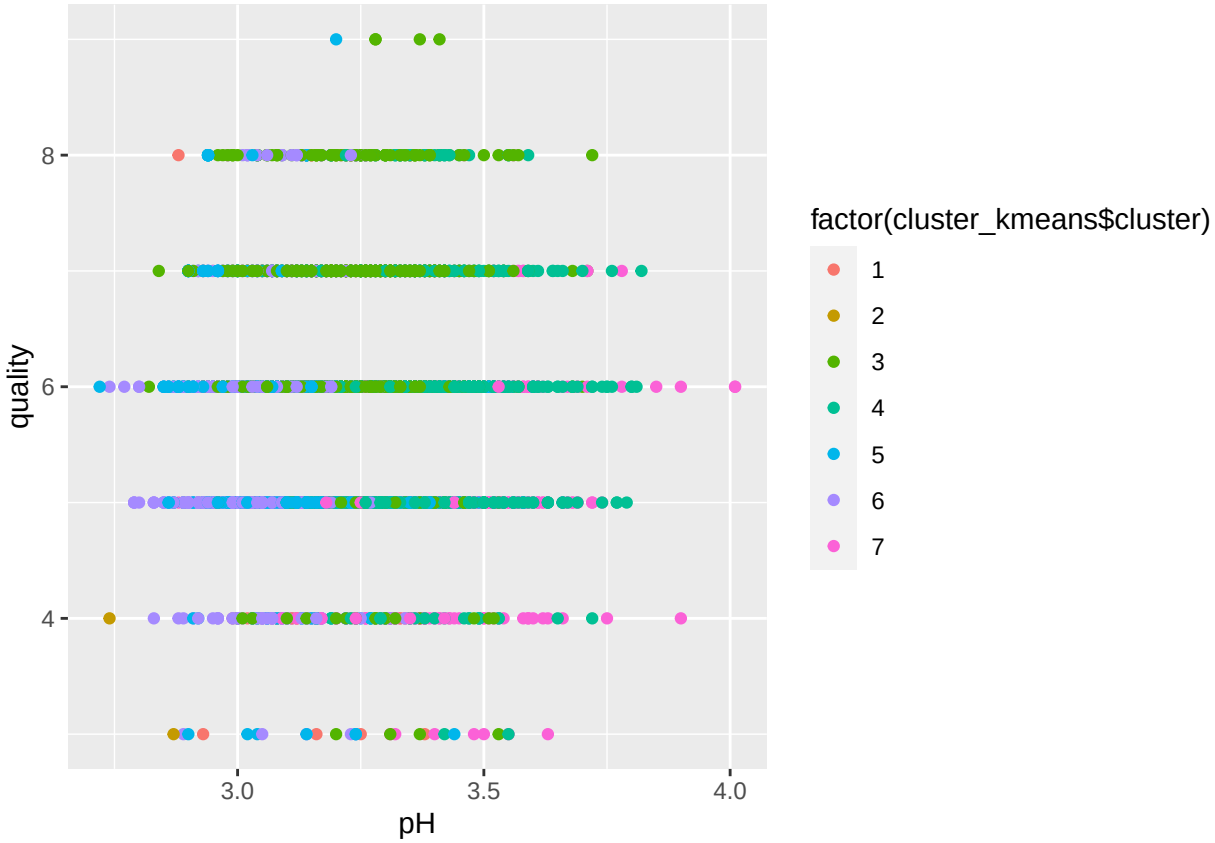

white
red
color
8
10
12
14
alcohol
factor(cluster_kmeans$cluster)
1
2

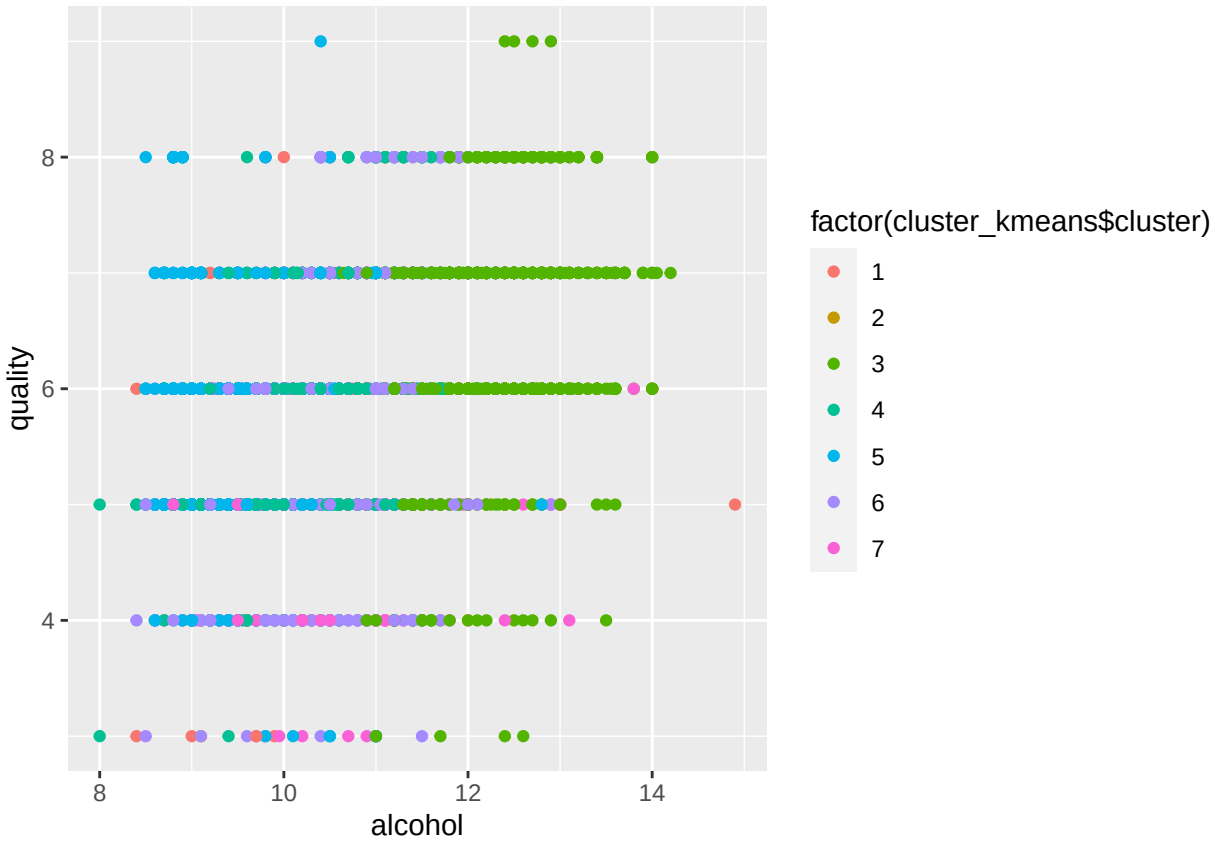


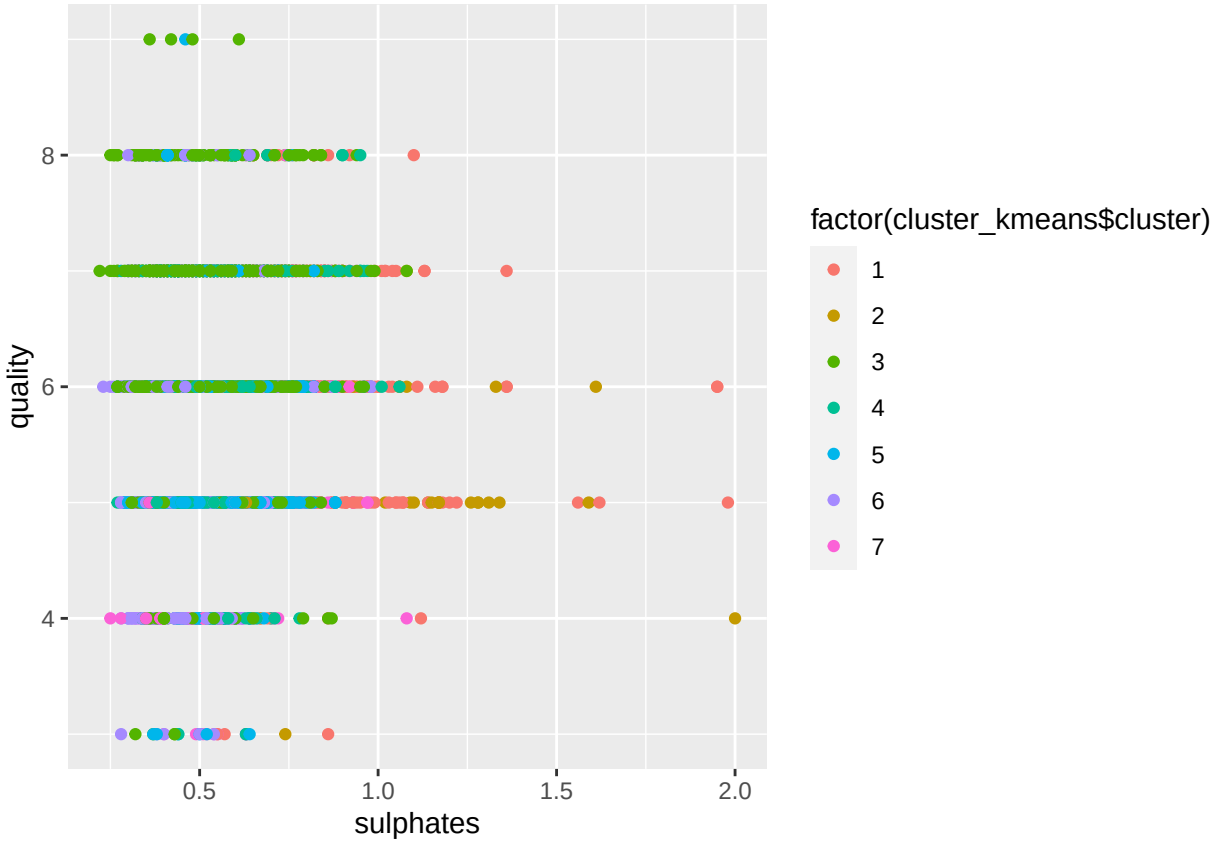
Since there are many variables, we only choose pH value, alcohol and sulphates as x-axis variables, and see whether the cluster model can distinguish their colors. We can see that k-mean clusters are able to distinguish the red and wine colors almost perfectly.

Distinguishing Wine Quality

We run a k-means clustering algorithm with no of clusters=7 (as we have 7 unique wine quality ratings)


quality
8
6
4
3.0
3.5
4.0
pH
factor(cluster_kmeans$cluster)
1
2
3
4
5
6
7


quality
8
6
4
8
10
12
14
alcohol
factor(cluster_kmeans$cluster)
1
2
3
4
5
6
7



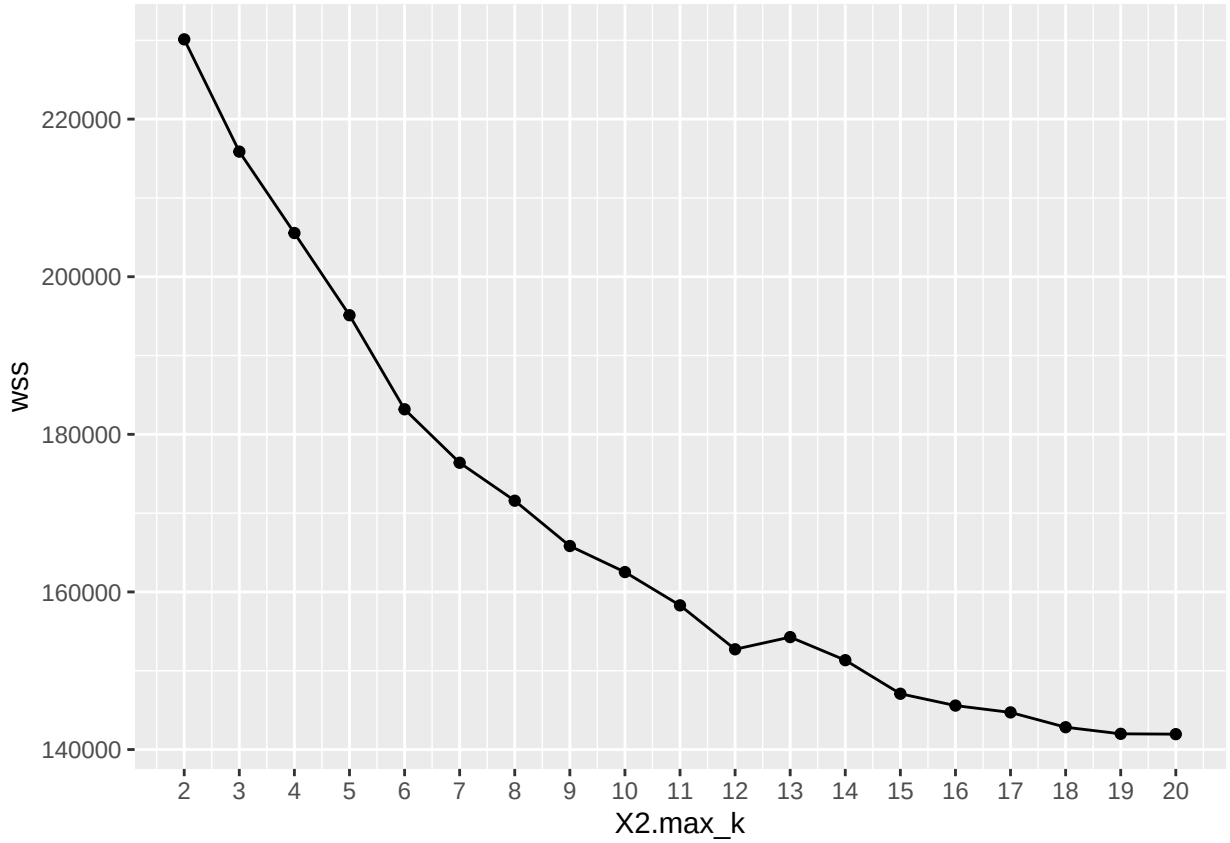
We see that the k-means is unable to distinguish between the different ratings of wine quality though clusters.

We observe both the PCA and K-Means techniques are able to distinguish between the red and white wine colors, however, both the models are unable to distinguish between wines of different quality ratings

Market Segmentation

Reading the Social Marketing csv file and cleaning the data

Running the K-Means Algorithm and finding the optimal k



Looking at the elbow plot, we can take k=4 and form 4 groups/ clusters

We planned to run a KNN model with k=4 and try to distinguish between these 4 groups

We have plotted the top 5 most important features that dominate each of the clusters to check for any interesting market segments. These features could give an idea about the type

```
## [1] "outdoors"         "fashion"          "personal_fitness" "health_nutrition"
## [5] "cooking"
```

Cluster 1 seems to include those segments of people who are fitness-conscious and concerned about their health

```
## [1] "current_events" "art"            "tv_film"        "college_uni"
## [5] "online_gaming"
```

Cluster 2 seems to include the younger population who are college-going and interested in online gaming, TV films and art

```
## [1] "school"        "food"          "sports_fandom" "parenting"
## [5] "religion"
```

Cluster 3 seems may include parents who have school-going children. The segment seems to cater towards the family folks

```
## [1] "automotive" "computers"  "travel"     "news"       "politics"
```

Cluster 4 may involve the techie population interested in computers and automotive. There also seems to be an inherent affinity towards news and politics for this cluster.

Reuters Corpus

Aim : Based on the term frequency of each word, try to predict which author does the a particular set of words belong to.

```
## Loading required package: NLP

##
## Attaching package: 'NLP'

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

##
## Attaching package: 'tm'

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

##
## Attaching package: 'proxy'

## The following object is masked from 'package:Matrix':
##
##     as.matrix

## The following objects are masked from 'package:stats':
##
##     as.dist, dist

## The following object is masked from 'package:base':
##
##     as.matrix

##   [1] "aronPressman"      "lanCrosby"         "lexanderSmith"
##   [4] "enjaminKangLim"    "ernardHickey"      "radDorfman"
##   [7] "arrenSchuettler"   "avidLawder"        "dnaFernandes"
##  [10] "ricAuchard"        "umikoFujisaki"     "rahamEarnshaw"
##  [13] "eatherScoffield"   "anLopatka"         "aneMacartney"
##  [16] "imGilchrist"       "oWinterbottom"     "oeOrtiz"
##  [19] "ohnMastrini"       "onathanBirt"       "arlPenhaul"
##  [22] "eithWeir"          "evinDrawbaugh"     "evinMorrison"
##  [25] "irstinRidley"      "ouroshKarimkhany"  "ydiaZajc"
##  [28] "ynneO'Donnell"     "ynnleyBrowning"    "arcelMichelson"
##  [31] "arkBendeich"       "artinWolk"         "atthewBunce"
##  [34] "ichaelConnor"      "ureDickie"         "ickLouth"
##  [37] "atriciaCommins"    "eterHumphrey"      "ierreTran"
##  [40] "obinSidel"         "ogerFillion"       "amuelPerry"
##  [43] "arahDavison"       "cottHillis"        "imonCowell"
```

```
##  [46] "anEeLyn"           "heresePoletti"     "imFarrand"
##  [49] "oddNissen"         "illiamKazer"       "AaronPressman"
##  [52] "AlanCrosby"        "AlexanderSmith"    "BenjaminKangLim"
##  [55] "BernardHickey"     "BradDorfman"       "DarrenSchuettler"
##  [58] "DavidLawder"       "EdnaFernandes"     "EricAuchard"
##  [61] "FumikoFujisaki"    "GrahamEarnshaw"    "HeatherScoffield"
##  [64] "JanLopatka"        "JaneMacartney"     "JimGilchrist"
##  [67] "JoWinterbottom"    "JoeOrtiz"          "JohnMastrini"
##  [70] "JonathanBirt"      "KarlPenhaul"       "KeithWeir"
##  [73] "KevinDrawbaugh"    "KevinMorrison"     "KirstinRidley"
##  [76] "KouroshKarimkhany" "LydiaZajc"         "LynneO'Donnell"
##  [79] "LynnleyBrowning"   "MarcelMichelson"   "MarkBendeich"
##  [82] "MartinWolk"        "MatthewBunce"      "MichaelConnor"
##  [85] "MureDickie"        "NickLouth"         "PatriciaCommins"
##  [88] "PeterHumphrey"     "PierreTran"        "RobinSidel"
##  [91] "RogerFillion"      "SamuelPerry"       "SarahDavison"
##  [94] "ScottHillis"       "SimonCowell"       "TanEeLyn"
##  [97] "TheresePoletti"    "TimFarrand"        "ToddNissen"
## [100] "WilliamKazer"

##
## Attaching package: 'modelr'

## The following objects are masked _by_ '.GlobalEnv':
##
##     sim1, sim2, sim3

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

## The following object is masked from 'package:ggformula':
##
##     na.warn

## <<SimpleCorpus>>
## Metadata:  corpus specific: 1, document level (indexed): 0
## Content:  documents: 5000

## <<DocumentTermMatrix (documents: 5000, terms: 45853)>>
## Non-/sparse entries: 1083147/228181853
## Sparsity           : 100%
## Maximal term length: 81
## Weighting          : term frequency (tf)
```

Removing words which are in the bottom 1%

```
## [1] "DocumentTermMatrix"    "simple_triplet_matrix"

## <<DocumentTermMatrix (documents: 5000, terms: 3378)>>
## Non-/sparse entries: 844216/16045784
## Sparsity           : 95%
## Maximal term length: 18
## Weighting          : term frequency (tf)
```

The number of terms reduced significantly

Predicting Author

We used Random Forest to predict the author based on the sets of words. We achieved an accuracy of:

```
## randomForest 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] 0.623
```

An accuracy of ~80% is achieved

Association Rule Mining

Step 1: Importing the important libraries and for association rule mining important libraries are arules and arulesViz

Step 2: Reading and cleaning of data - Reading the text file of grocery list. Converting the column values strings and splitting the list of items bought by ',' Creating a 'transactions' class for apriori algorithm.

Summary of the dataset is as below:

```
summary(grotrans)
```

```
## transactions as itemMatrix in sparse format with
##  9835 rows (elements/itemsets/transactions) and
##  169 columns (items) and a density of 0.02609146
##
## most frequent items:
##       whole milk other vegetables       rolls/buns             soda
##             2513             1903             1809             1715
##           yogurt          (Other)
##             1372            34055
##
## element (itemset/transaction) length distribution:
## sizes
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
## 2159 1643 1299 1005  855  645  545  438  350  246  182  117   78   77   55   46
##   17   18   19   20   21   22   23   24   26   27   28   29   32
```

```
##   29   14   14    9   11    4    6    1    1    1    1    3    1
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   3.000   4.409   6.000  32.000
##
## includes extended item information - examples:
##             labels
## 1 abrasive cleaner
## 2 artif. sweetener
## 3   baby cosmetics
```

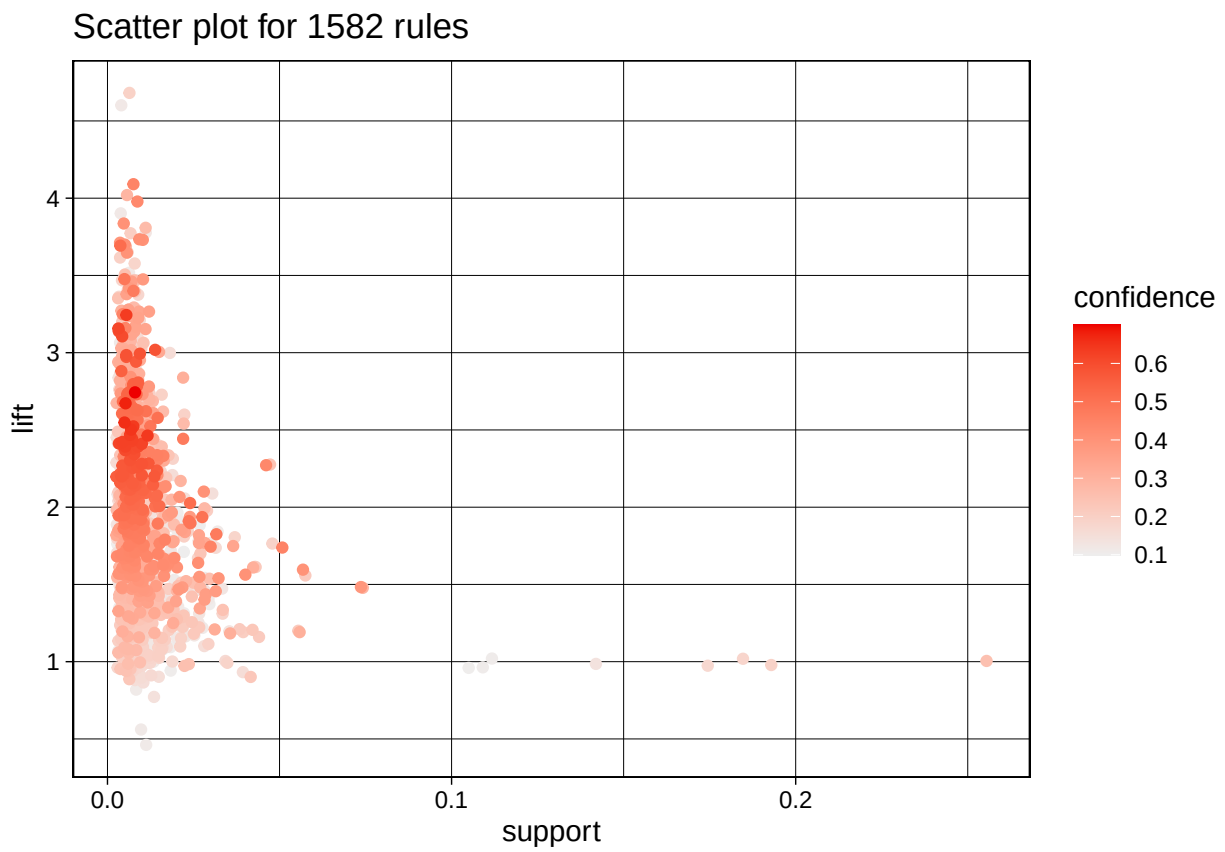
The summary of data tells us that most frequent items are Whole milk, Other vegetables followed by rolls/buns and yogurt. It also tels us that 2159 customers buy only 1 item.

Step 3: Run the apriori algorithm with support as 0.005, confidence as 0.1 and maxlen = 4

Step 4: The total association rules come out to be 1582, which is very high

Step 5: Plotting the association rules on support vs lift

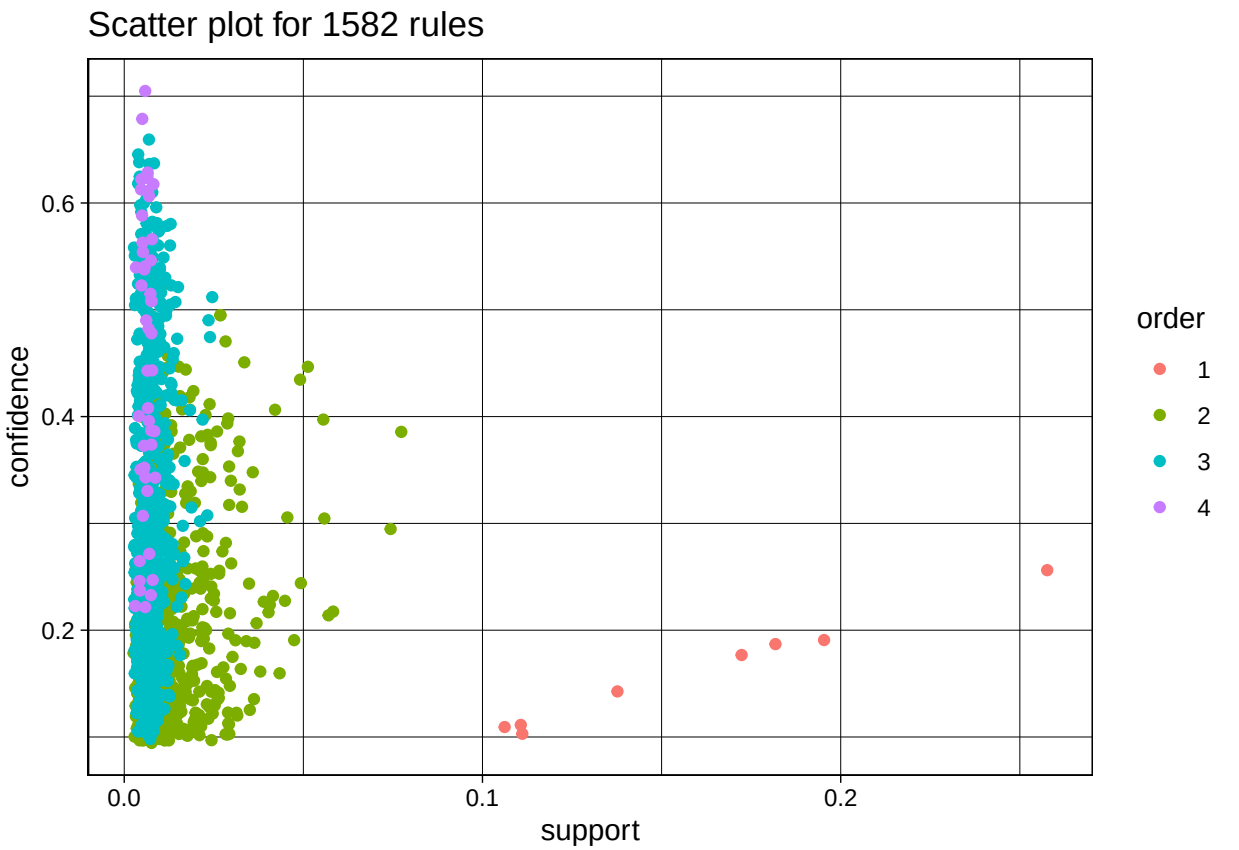
```
plot(grocrules, measure = c("support", "lift"), shading = "confidence")
```



In the above chart we can see that for lift > 2 the confidence is higher. So using the above chart, association rules with lift > 2 & confidence > 0.5 & support > 0.01 should give the important rules.

The two-key plot is below

```
# "two key" plot: coloring is by size (order) of item set
plot(grocrules, method='two-key plot')
```



We can say from the chart that order 4 and 3 has low support and high lift. And for order 2 and 1 the support is relatively high but confidence is lower.

```
# can now look at subsets driven by the plot
arules :: inspect(subset(grocrules, lift > 2 & confidence > 0.5 & support > 0.01))
```

```
##      lhs                                rhs                support
## [1]  {curd, yogurt}                  => {whole milk}       0.01006609
## [2]  {butter, other vegetables}      => {whole milk}       0.01148958
## [3]  {domestic eggs, other vegetables} => {whole milk}     0.01230300
## [4]  {whipped/sour cream, yogurt}    => {whole milk}       0.01087951
## [5]  {other vegetables, pip fruit}   => {whole milk}       0.01352313
## [6]  {citrus fruit, root vegetables} => {other vegetables} 0.01037112
## [7]  {root vegetables, tropical fruit} => {other vegetables} 0.01230300
## [8]  {root vegetables, tropical fruit} => {whole milk}     0.01199797
## [9]  {tropical fruit, yogurt}        => {whole milk}       0.01514997
## [10] {root vegetables, yogurt}       => {whole milk}       0.01453991
## [11] {rolls/buns, root vegetables}   => {other vegetables} 0.01220132
## [12] {rolls/buns, root vegetables}   => {whole milk}       0.01270971
## [13] {other vegetables, yogurt}      => {whole milk}       0.02226741
##      confidence coverage   lift     count
## [1]  0.5823529  0.01728521 2.279125  99
```

```
## [2]  0.5736041  0.02003050 2.244885 113
## [3]  0.5525114  0.02226741 2.162336 121
## [4]  0.5245098  0.02074225 2.052747 107
## [5]  0.5175097  0.02613116 2.025351 133
## [6]  0.5862069  0.01769192 3.029608 102
## [7]  0.5845411  0.02104728 3.020999 121
## [8]  0.5700483  0.02104728 2.230969 118
## [9]  0.5173611  0.02928317 2.024770 149
## [10] 0.5629921  0.02582613 2.203354 143
## [11] 0.5020921  0.02430097 2.594890 120
## [12] 0.5230126  0.02430097 2.046888 125
## [13] 0.5128806  0.04341637 2.007235 219
```

Using the thresholds for lift, support and confidence we get 13 association rules.

The one with max lift ~3 is {citrus fruit, root vegetables} => {other vegetables}. This shows if a customer comes to the store to buy citrus fruits and root vegetables, he/she would also buy other vegetables.

Another important rule is, {rolls/buns, root vegetables} => {other vegetables}. This seems like that customers who buy rolls/buns and root vegetables for a dish, also needs other veggies to complete the dish so he/she purchases that as well.

For rule, {curd, yogurt} => {whole milk} we can say that customers who buy curd and yogurt also buys whole milk. This might because they are interested in dairy products.

Below is the image from Gephi with lift>2 & support > 0.01 & confidence > 0.3.

The plot says that Other Vegetables and whole milk have a very thick line showing many edges between them so the betweenness is very high. The plot shows the frequent products of the grocery store.

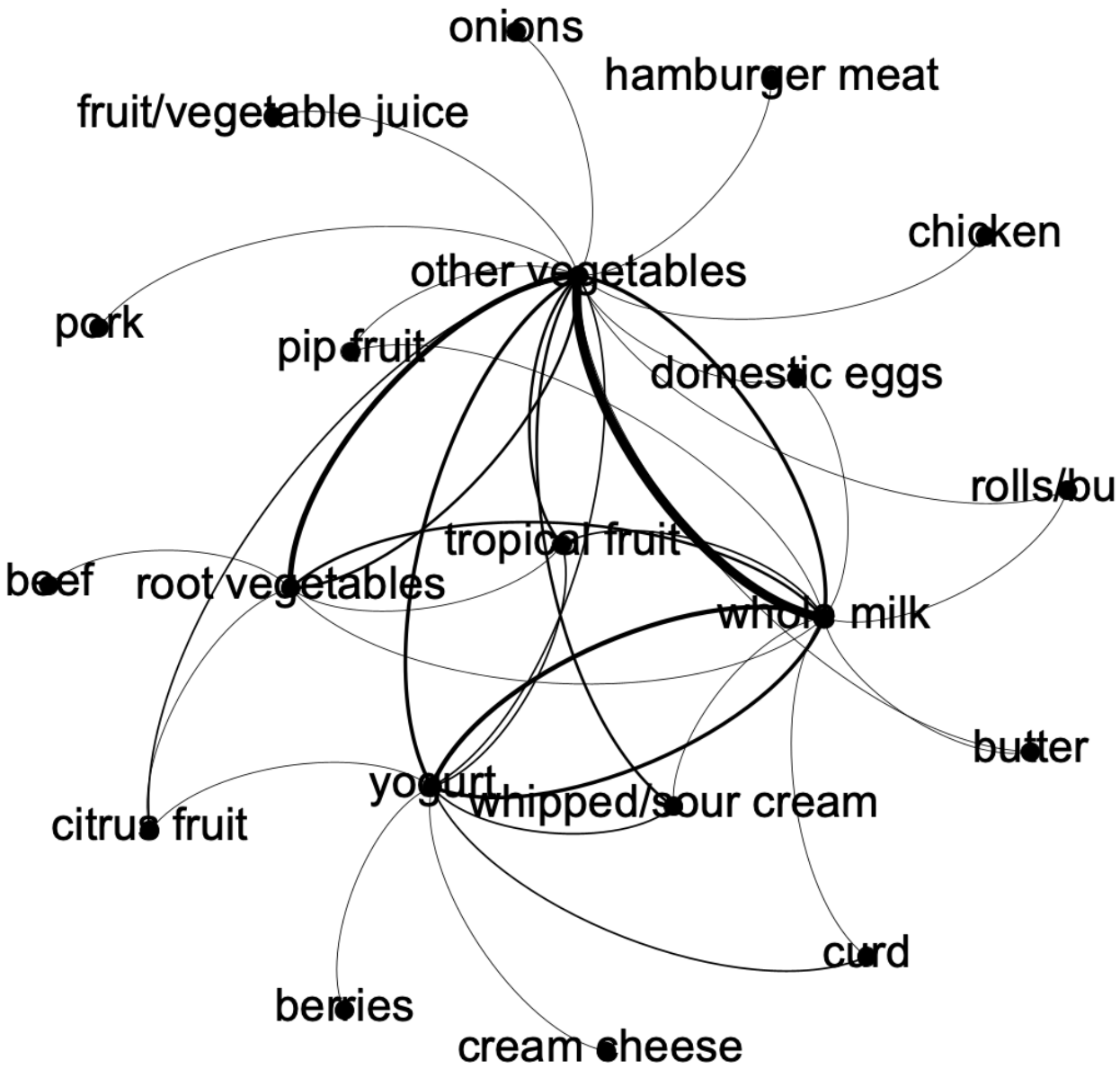


Figure 3: Caption for the picture.