STA 380 Part 2 Exercises Juwon Lee, Aakash Talathi, Milan Patel, Teja Sirigina

https://github.com/juwon0502/STA-380-pt-2-Exercises (https://github.com/juwon0502/STA-380-pt-2-Exercises)

Probability Practice

Part A.

Given information:

- p(random) = 0.3
- p(truthful) = 0.7
- p(yes) = 0.65
- p(no) = 0.35
- p(yes|random) = 0.3 * 0.5 = 0.15
- p(no|random) = 0.3 * 0.5 = 0.15

We want to figure out p(yes|truthful)

- p(yes|truthful) = p(yes and truthful)/p(truthful)
- p(yes) - p(yes and random) = p(yes and truthful)
- 0.65 - 0.15 = 0.5

Plug in:

- p(yes|truthful) = 0.5/0.7

```
0.5/0.7
```

```
## [1] 0.7142857
```

Part B.

Given information:

- p(test positive|has disease) = 0.993
- p(test negative | doesn’t have disease) = 0.9999
- p(has disease) = 0.000025

Therefore:

- p(has disease and tests positive) = 0.000025 * 0.993
- p(has disease and tests negative) = 0.000025 * 0.007
- p(does not have disease and tests positive) = 0.999975 * 0.0001

- p(does not have disease and tests negative) = 0.999975 * 0.9999

```
0.000025 * 0.993
```

```
## [1] 2.4825e-05
```

```
0.000025 * 0.007
```

```
## [1] 1.75e-07
```

```
0.999975 * 0.0001
```

```
## [1] 9.99975e-05
```

```
0.999975 * 0.9999
```

```
## [1] 0.999875
```

We want to figure out p(has disease | tests positive)

- p(has disease | tests positive) = p(has disease and tests positive)/p(tests positive)
- p(tests positive) = p(does not have disease and tests positive) + p(has disease and tests positive)

  = 0.000024825 + 0.000099975 = 0.0001248

Therefore:

- p(has disease | tests positive) = p(has disease and tests positive)/p(tests positive) = 0.000024825 / 0.0001248 = 0.1989

Wrangling the Billboard Top 100

```
library(readr)
```

```
## Warning: package 'readr' was built under R version 4.0.5
```

```
billboard <- read_csv("billboard.csv")
```

```
## New names:
## * `` -> ...1
```

```
## Rows: 327895 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): url, week_id, song, performer, song_id
## dbl (8): ...1, week_position, instance, previous_week_position, peak_positio...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(billboard,10)
```

...1 <dbl>	url <chr>	week_id <chr>	week_position <dbl>
1	http://www.billboard.com/charts/hot-100/1965-07-17	7/17/1965	34
2	http://www.billboard.com/charts/hot-100/1965-07-24	7/24/1965	22
3	http://www.billboard.com/charts/hot-100/1965-07-31	7/31/1965	14
4	http://www.billboard.com/charts/hot-100/1965-08-07	8/7/1965	10
5	http://www.billboard.com/charts/hot-100/1965-08-14	8/14/1965	8
6	http://www.billboard.com/charts/hot-100/1965-08-21	8/21/1965	8
7	http://www.billboard.com/charts/hot-100/1965-08-28	8/28/1965	14
8	http://www.billboard.com/charts/hot-100/1965-09-04	9/4/1965	36
9	http://www.billboard.com/charts/hot-100/1997-04-19	4/19/1997	97
10	http://www.billboard.com/charts/hot-100/1997-04-26	4/26/1997	90

1-10 of 10 rows | 1-4 of 13 columns

Part A.

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.5
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v dplyr   1.0.7
## v tibble  3.1.6     v stringr 1.4.0
## v tidyr   1.2.0     v forcats 0.5.1
## v purrr   0.3.4
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
## Warning: package 'tibble' was built under R version 4.0.5
```

```
## Warning: package 'tidyr' was built under R version 4.0.5
```

```
## Warning: package 'purrr' was built under R version 4.0.5
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
```

```
## Warning: package 'stringr' was built under R version 4.0.5
```

```
## Warning: package 'forcats' was built under R version 4.0.5
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
billboard %>% group_by(performer, song) %>% summarize(count = n()) %>%
  arrange(-count) %>% head(10)
```

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

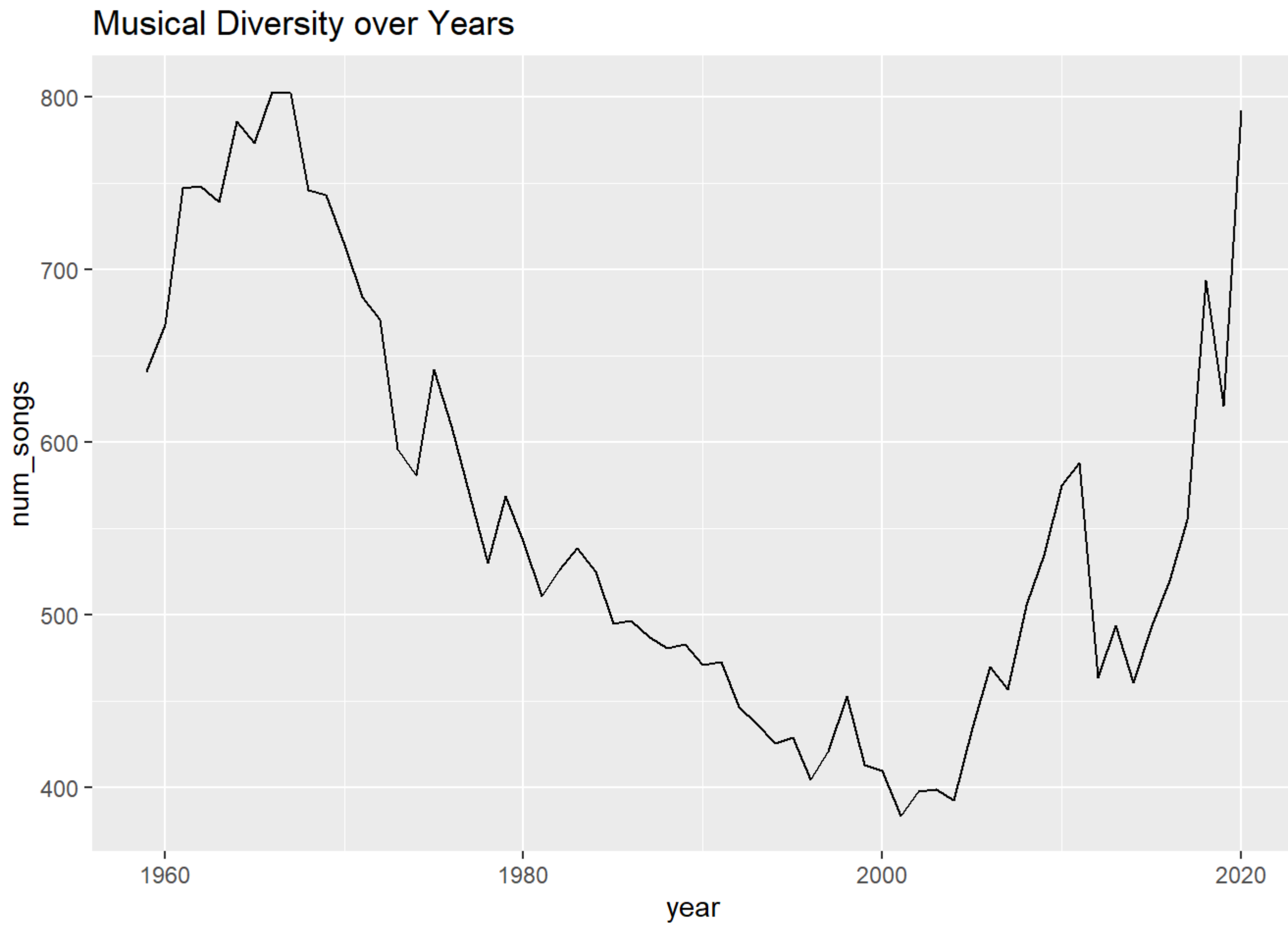
performer <chr>	song <chr>	c <
Imagine Dragons	Radioactive	
AWOLNATION	Sail	
Jason Mraz	I'm Yours	
The Weeknd	Blinding Lights	
LeAnn Rimes	How Do I Live	
LMFAO Featuring Lauren Bennett & GoonRock	Party Rock Anthem	
OneRepublic	Counting Stars	
Adele	Rolling In The Deep	
Jewel	Foolish Games/You Were Meant For Me	
Carrie Underwood	Before He Cheats	

1-10 of 10 rows

This table represents the top 10 most popular songs since 1958. The count is the number of weeks the song appeared on the Billboard top 100 list.

Part B

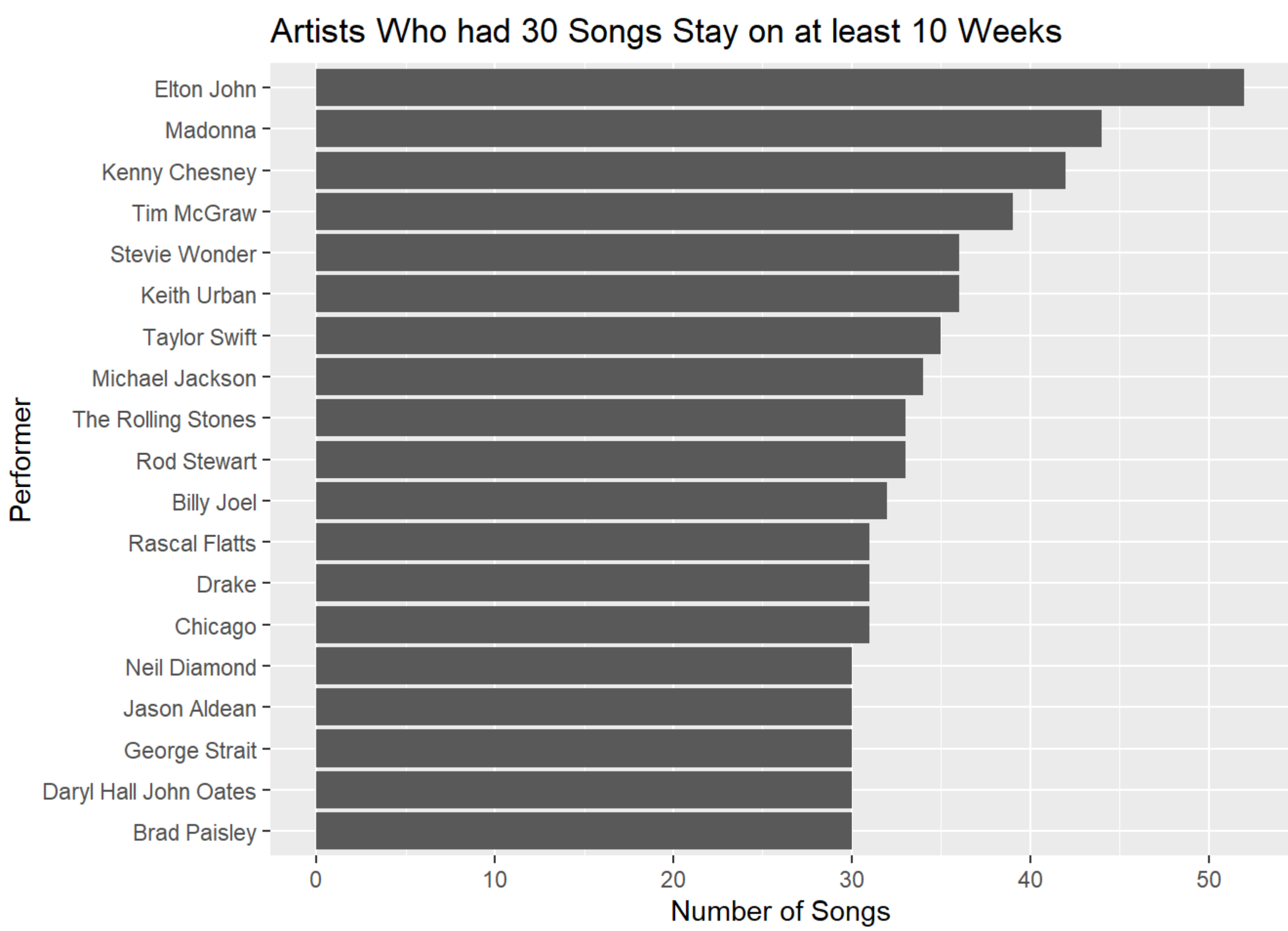
```
billboard %>% filter(year > 1958 & year < 2021) %>%
  group_by(year) %>%
  summarize(num_songs = n_distinct(song)) %>%
  ggplot() + geom_line(aes(year, num_songs)) +
  ggtitle("Musical Diversity over Years")
```



Part C

```
billboard %>% group_by(song, performer) %>%
  summarize(max_weeks = max(weeks_on_chart)) %>%
  filter(max_weeks >= 10) %>%
  group_by(performer) %>% summarize(number_songs_ten_weeks = n()) %>%
  filter(number_songs_ten_weeks >= 30) %>% arrange(-number_songs_ten_weeks) %>%
  ggplot(aes(y = reorder(performer, number_songs_ten_weeks),
             x = number_songs_ten_weeks)) +
  geom_bar(stat = "identity") +
  ylab("Performer") +
  xlab("Number of Songs") +
  ggtitle("Artists Who had 30 Songs Stay on at least 10 Weeks")
```

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



Visual story telling part 1: green buildings

```
library(readr)
greenbuildings <- read_csv("greenbuildings.csv")
```

```
## Rows: 7894 Columns: 23
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (23): CS_PropertyID, cluster, size, empl_gr, Rent, leasing_rate, stories...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
greenbuildings %>% head(10)
```

CS_PropertyID <dbl>	cluster <dbl>	size <dbl>	empl... <dbl>	Rent <dbl>	leasing_rate <dbl>	stories <dbl>	... <dbl>	renovated <dbl>	class_a <dbl>
379105	1	260300	2.22	38.56	91.39	14	16	0	1
122151	1	67861	2.22	28.57	87.14	5	27	0	0
379839	1	164848	2.22	33.31	88.94	13	36	1	0
94614	1	93372	2.22	35.00	97.04	13	46	1	0
379285	1	174307	2.22	40.69	96.58	16	5	0	1
94765	1	231633	2.22	43.16	92.74	14	20	0	1
236739	6	210038	4.01	12.50	94.33	11	38	0	0
234578	6	225895	4.01	14.77	91.02	15	24	0	1
42087	6	912011	4.01	17.00	99.32	31	34	0	1
233989	6	518578	4.01	17.00	93.54	21	36	1	1

1-10 of 10 rows | 1-10 of 23 columns

```
### see how leasing rate correlates with the rest of the variables

lesstenpct=greenbuildings %>% filter(leasing_rate<=10)

cor(lesstenpct[sapply(lesstenpct, is.numeric)],use="complete.obs")[,6]
```

```
## Warning in cor(lesstenpct[sapply(lesstenpct, is.numeric)], use =
## "complete.obs"): the standard deviation is zero
```

```
##     CS_PropertyID           cluster              size           empl_gr 
##       -0.17990645       -0.05930760        0.22640870       -0.02408887 
##              Rent      leasing_rate           stories               age 
##        0.01306946        1.00000000        0.27228111       -0.07193576 
##         renovated           class_a           class_b              LEED 
##        0.21374311        0.18560543        0.07772456                NA 
##        Energystar      green_rating               net         amenities 
##       -0.03289791       -0.03289791       -0.03289791        0.32987392 
##       cd_total_07        hd_total07       total_dd_07     Precipitation 
##        0.02942398       -0.02906541       -0.01147026       -0.13743538 
##         Gas_Costs Electricity_Costs      cluster_rent 
##       -0.14631376       -0.06237857       -0.13768930
```

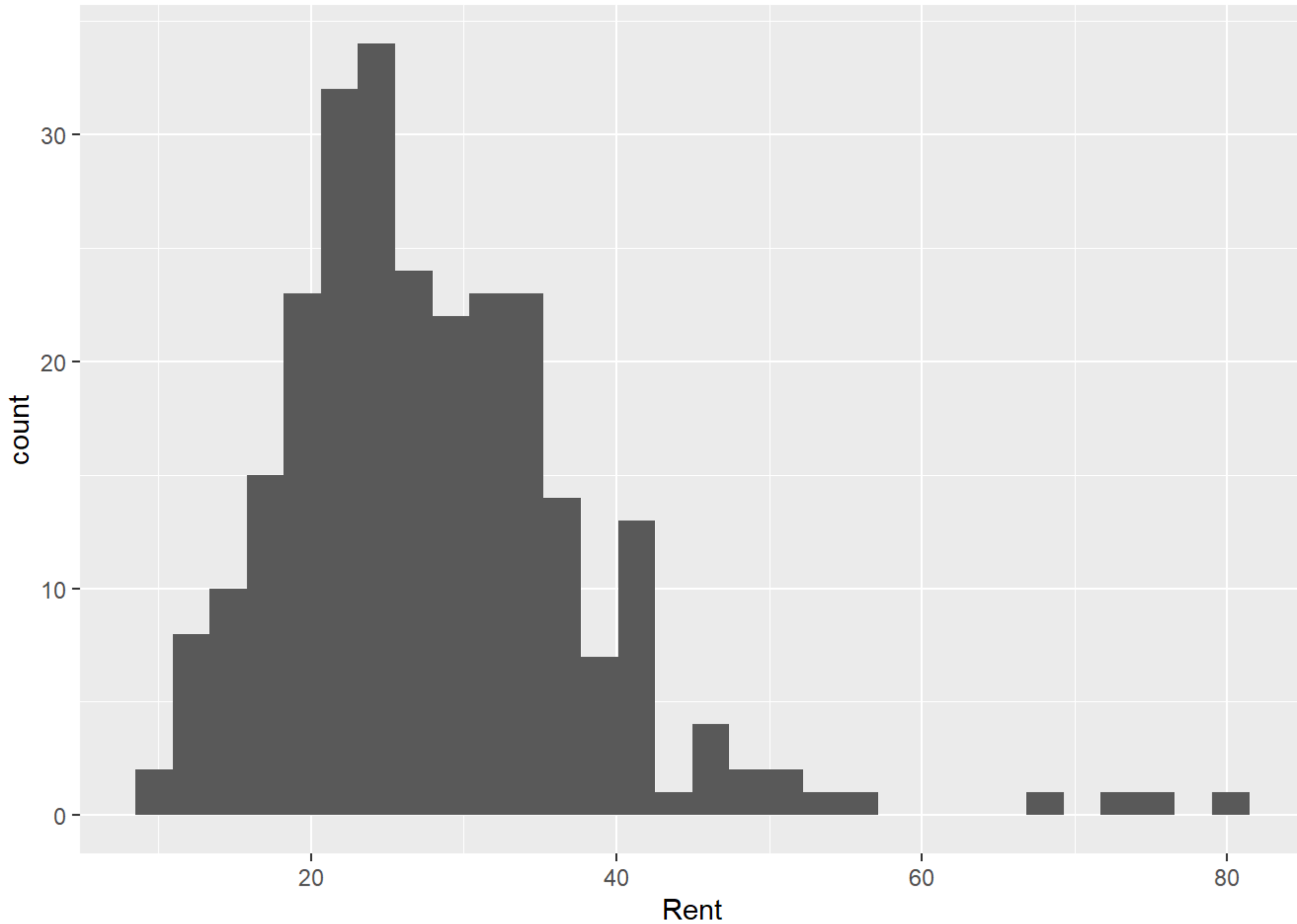
```
moretenpct=greenbuildings %>% filter(leasing_rate>10)

cor(moretenpct[sapply(moretenpct, is.numeric)],use="complete.obs")[,6]
```

```
##     CS_PropertyID           cluster              size           empl_gr 
##      -0.052218347       0.005563464       0.171208141      -0.040143237 
##              Rent      leasing_rate           stories               age 
##       0.178815521       1.000000000       0.171219443      -0.131089232 
##         renovated           class_a           class_b              LEED 
##      -0.026941187       0.187001244      -0.063678538       0.015352215 
##        Energystar      green_rating               net         amenities 
##       0.083125581       0.083723198       0.015524343       0.142021607 
##       cd_total_07        hd_total07       total_dd_07     Precipitation 
##      -0.025529485       0.007151803      -0.007061384       0.029015296 
##         Gas_Costs Electricity_Costs      cluster_rent 
##       0.047223786       0.067979687       0.170212818
```

Since the stats guru scrubbed the data clean of buildings with less than 10%, we wanted to see if these data entries were correlated with any of the other variables, and how the correlation differs between the less than 10% full buildings and the more than 10% full buildings. We noticed that in the less than 10% full correlation data, the cluster rent was negatively correlated at -0.137, whereas in the more than 10% correlation data it is positively correlated at 0.17. Other variables of note include amenities and renovated, both of which differ by around 20%. This suggests there may be a structural difference between the two groups and as such we cannot drop the data in which less then 10 percent of the building is being leased.

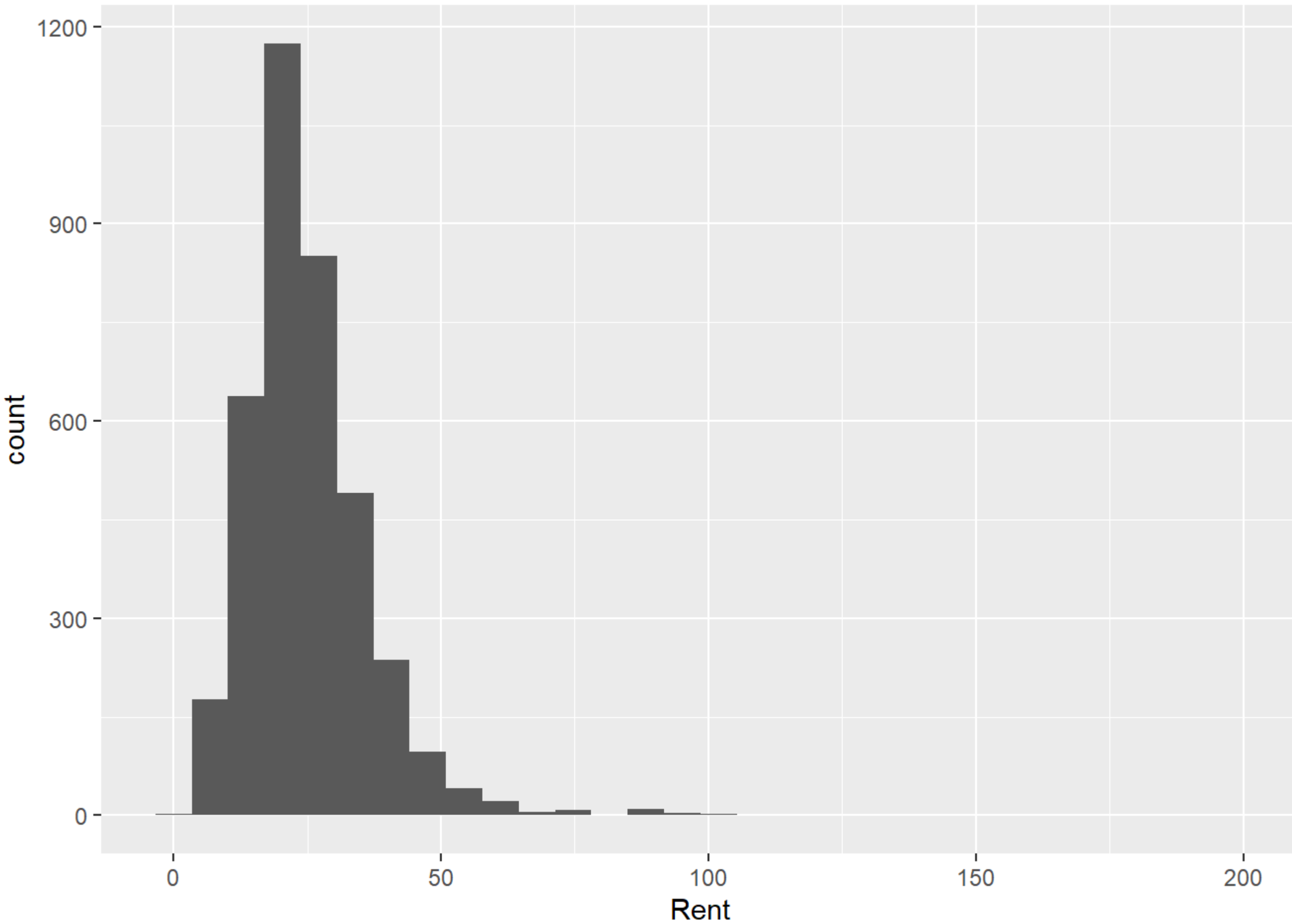
```
greenbuildings %>% filter(leasing_rate <= 90) %>%
  filter(green_rating == 1) %>%
  ggplot(aes (x = Rent)) + geom_histogram()
```

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




```
greenbuildings %>% filter(leasing_rate <= 90) %>%
  filter(green_rating == 0) %>%
  ggplot(aes (x = Rent)) + geom_histogram()
```

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

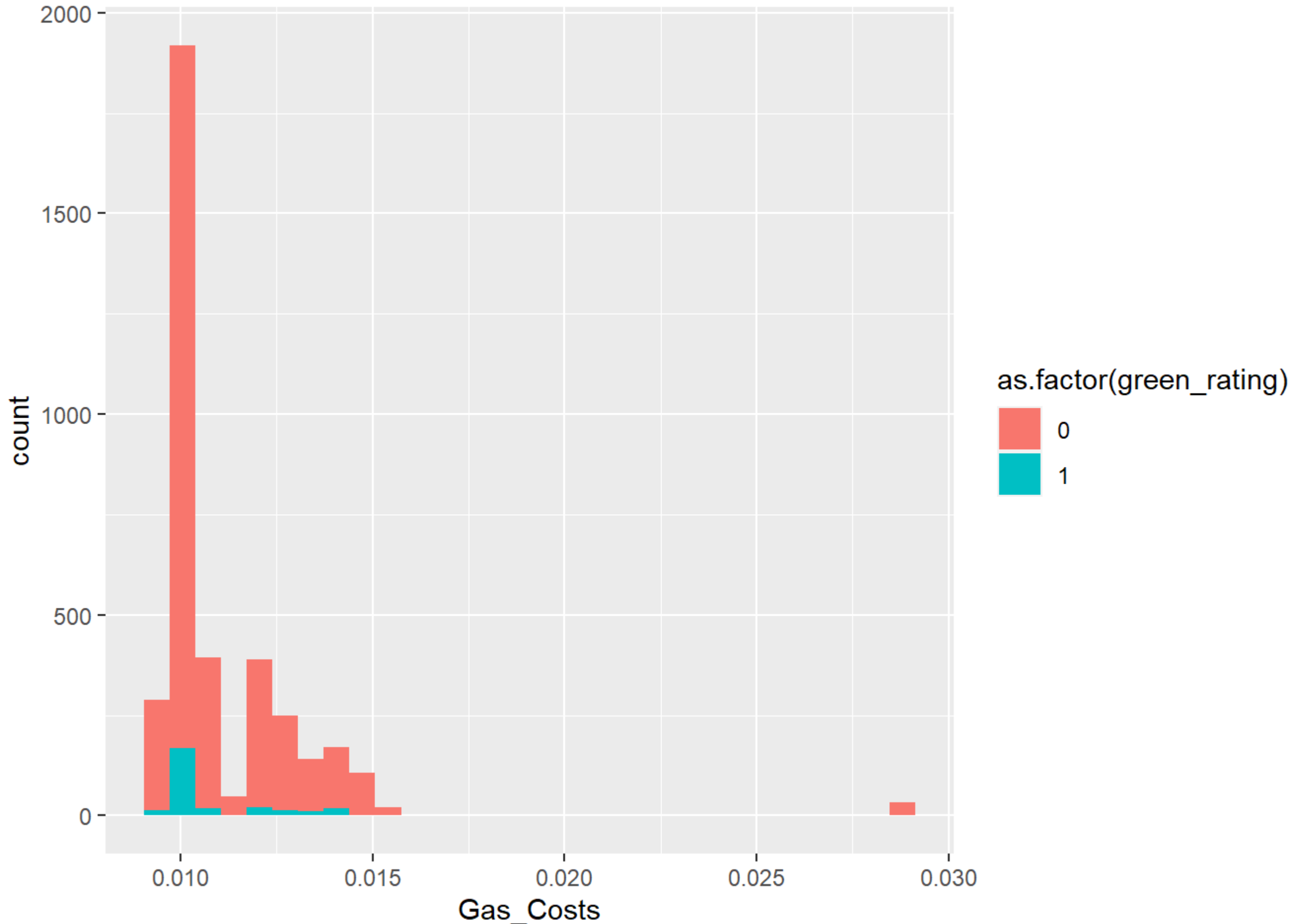


Just to check the Excel Guru's reasoning, the distributions were plotted.

We can see if utility costs will differ, which can affect the developer's decision

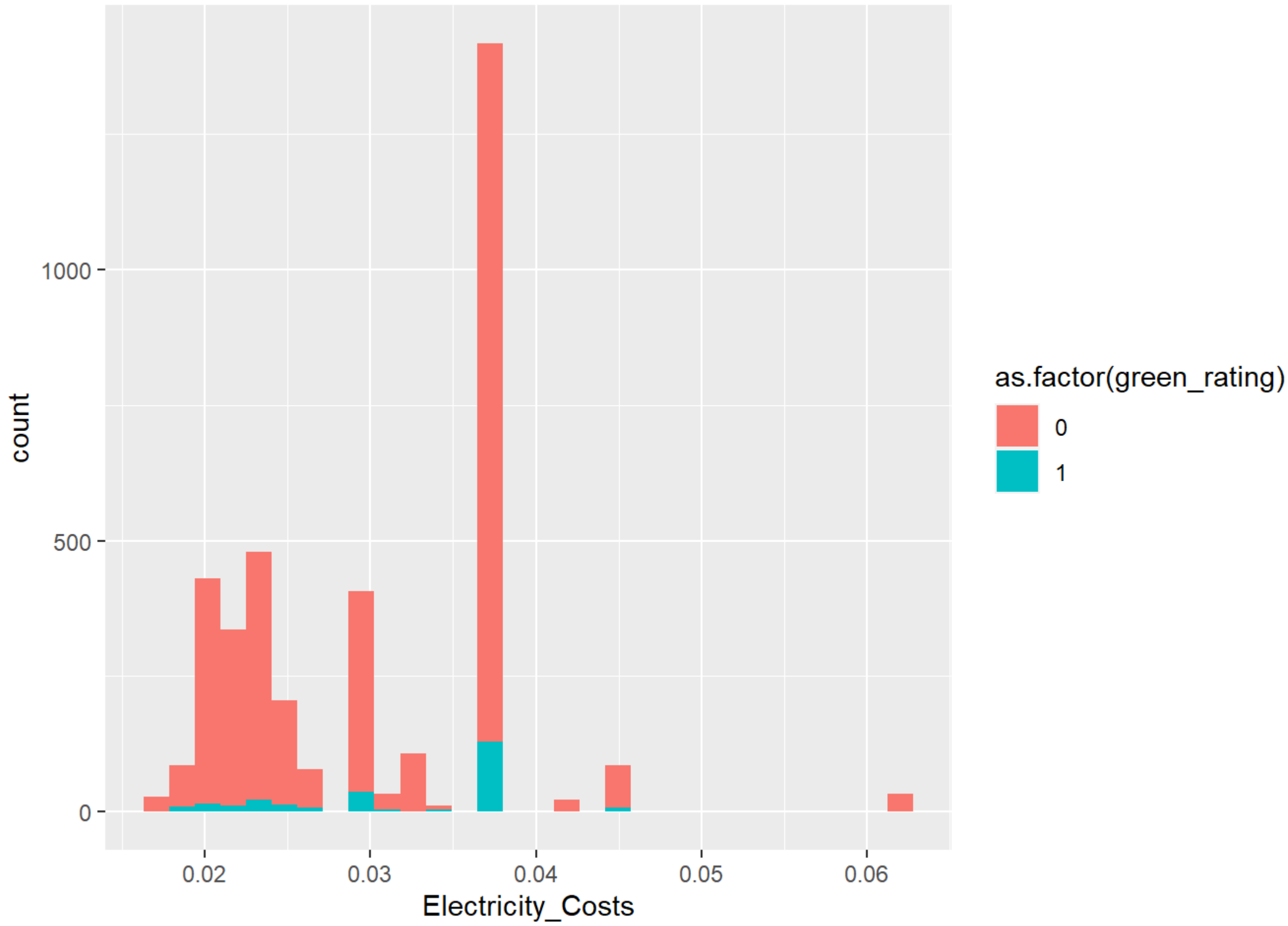
```
greenbuildings %>% filter(leasing_rate <= 90) %>%
  ggplot(aes (x = Gas_Costs, fill = as.factor(green_rating))) +
  geom_histogram(position = "identity")
```

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




```
greenbuildings %>% filter(leasing_rate <= 90) %>%
  ggplot(aes (x = Electricity_Costs, fill = as.factor(green_rating))) +
  geom_histogram(position = "identity")
```

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



These two histograms show us that overall, the cost of electricity and gas is similar for both green buildings and non-green buildings.

The information we can determine from our analysis supports the Excel Guru.

```
greenbuildings %>% select(Rent, green_rating) %>% group_by(green_rating) %>%
  summarize(x = median(Rent))
```

green_rating <dbl>	x <dbl>
0	25.0
1	27.6

2 rows

Visual story telling part 2: Capital Metro data

```
capmetro <- read_csv("capmetro_UT.csv")
```

```
## Rows: 5824 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): day_of_week, month, weekend
## dbl  (4): boarding, alighting, temperature, hour_of_day
## dttm (1): timestamp
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

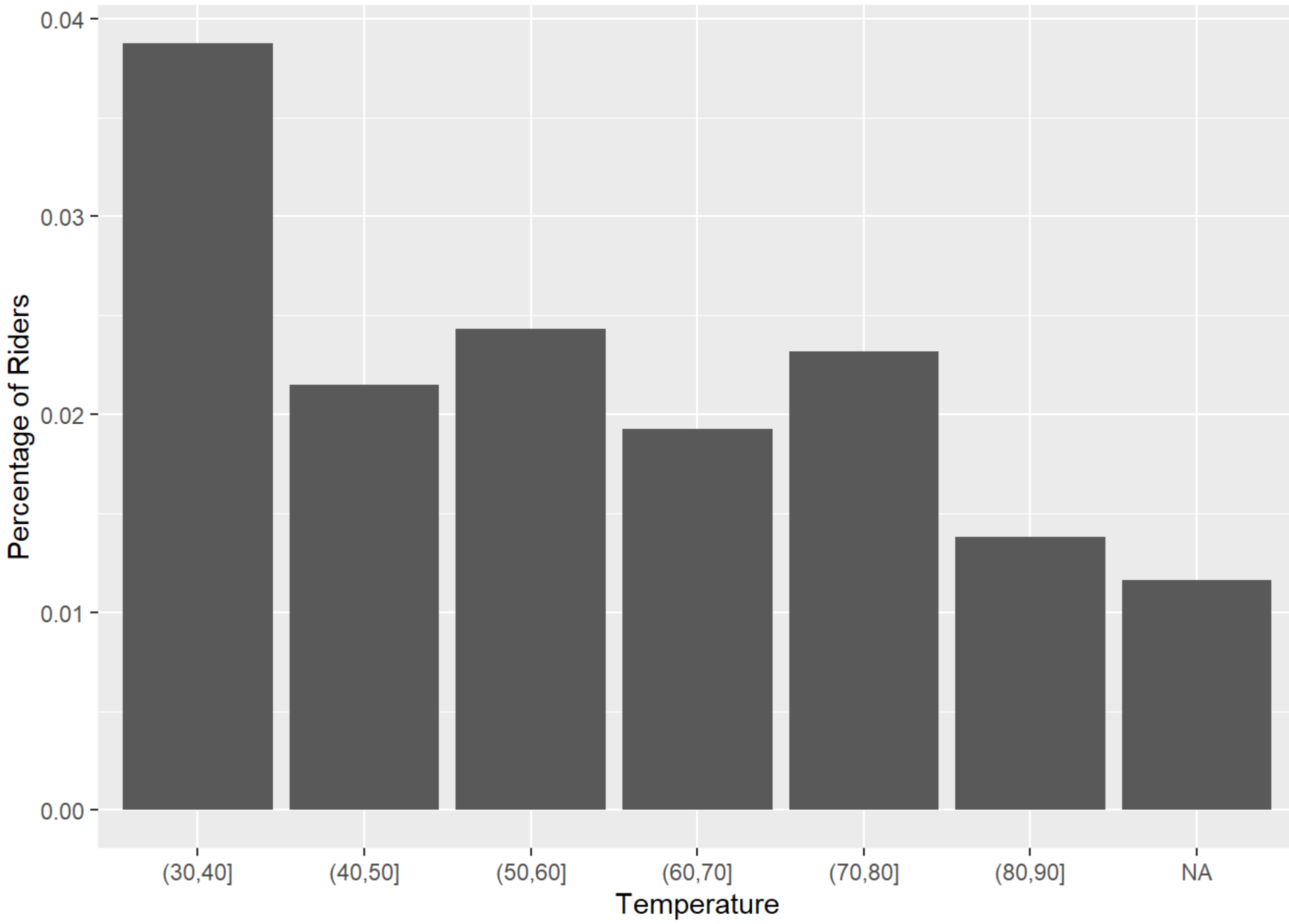
```
capmetro %>% head(10)
```

timestamp <dttm>	boarding <dbl>	alighting <dbl>	day_of_week <chr>	temperature <dbl>	hour_of_day <dbl>	mo... <chr>	weel <chr
2018-09-01 06:00:00	0	1	Sat	74.82	6	Sep	week
2018-09-01 06:15:00	2	1	Sat	74.82	6	Sep	week
2018-09-01 06:30:00	3	4	Sat	74.82	6	Sep	week
2018-09-01 06:45:00	3	4	Sat	74.82	6	Sep	week
2018-09-01 07:00:00	2	4	Sat	74.39	7	Sep	week
2018-09-01 07:15:00	4	4	Sat	74.39	7	Sep	week
2018-09-01 07:30:00	3	12	Sat	74.39	7	Sep	week
2018-09-01 07:45:00	8	4	Sat	74.39	7	Sep	week
2018-09-01 08:00:00	4	15	Sat	75.72	8	Sep	week
2018-09-01 08:15:00	7	10	Sat	75.72	8	Sep	week

1-10 of 10 rows

How does weather affect the number of students who take the bus?

```
capmetro %>% mutate(temps = cut(temperature, breaks = c(30, 40, 50, 60, 70, 80, 90))) %>% group_
by(temps) %>%
  summarize(num_occurance = n(), num_riders = sum(boarding)) %>%
  mutate(riders_per_temp = num_occurance/num_riders) %>%
  ggplot(aes(x = temps, y = riders_per_temp)) + geom_bar(stat = "identity") +
  xlab("Temperature") + ylab("Percentage of Riders")
```

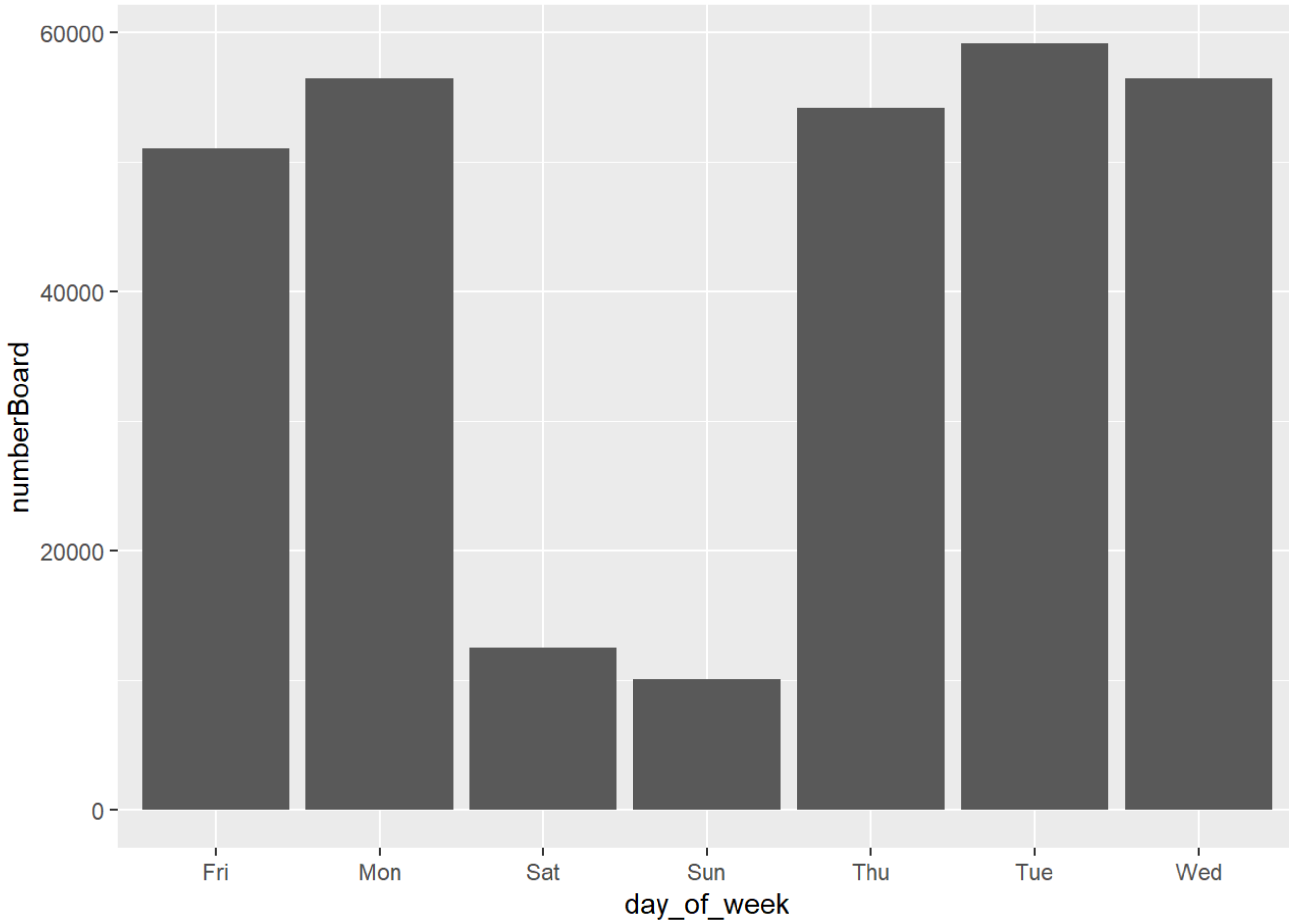


Overall, we can see that students will take the bus the most when it is very cold outside. We expected that very hot days would also have the most number of riders, however it was the lowest bucket. One possible reason for this is that the hottest time of the day is typically between 4-6 which is not a time most students travel meaning that overall, they wouldn’t take the bus.

The Y axis was also scaled since there were less occurrences of very hot and very cold days.

What about day of the week?

```
day = capmetro$day_of_week

capmetro %>% group_by(day_of_week) %>% summarize(numberBoard = sum(boarding)) %>% ggplot(aes(x =
day_of_week, y = numberBoard)) + geom_bar(stat = "identity")
```



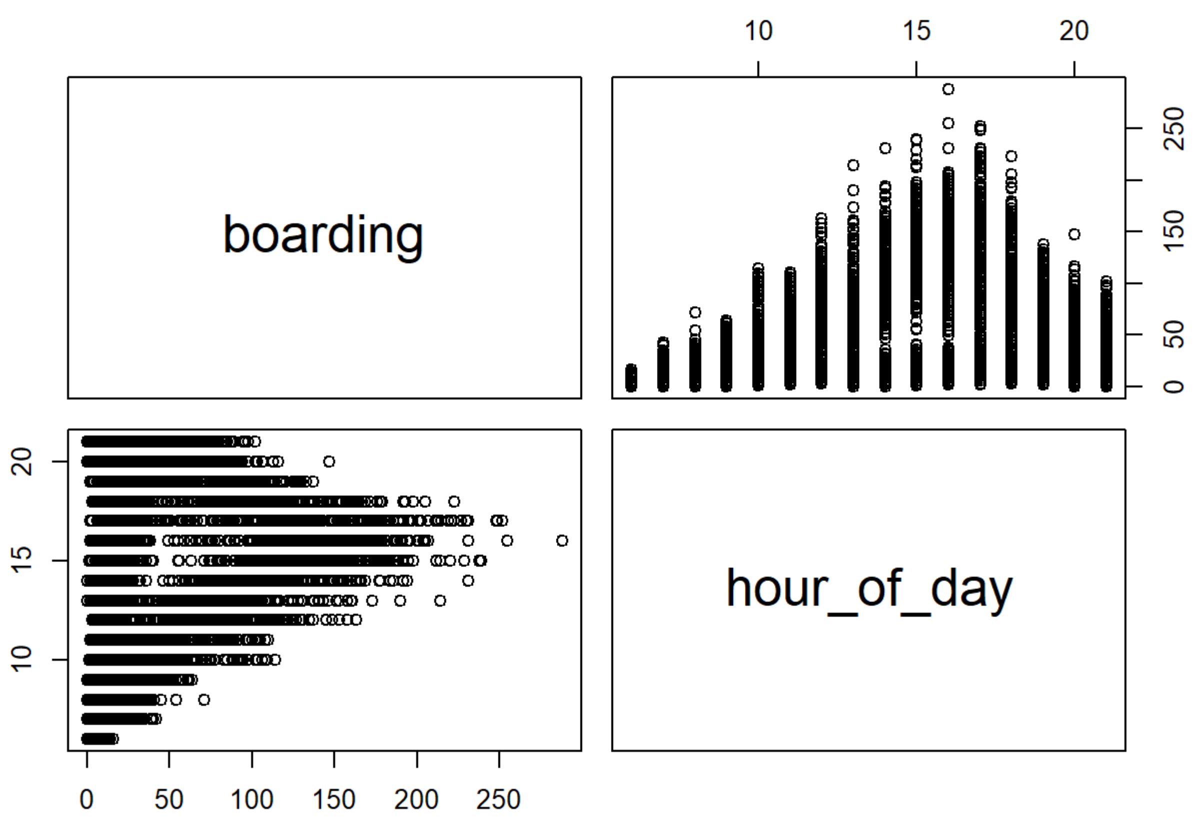
Based on this graph we can see bus ridership at UT is much higher during the weekdays, which is what we expected given that school is not in session during those days.

And what about time of day?

```
model1=lm(data=capmetro,boarding~hour_of_day)
summary(model1)
```

```
##
## Call:
## lm(formula = boarding ~ hour_of_day, data = capmetro)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -79.46 -30.20 -11.56  23.08 227.17
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.2027     1.8532   0.649    0.516
## hour_of_day   3.7266     0.1299  28.686   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 45.7 on 5822 degrees of freedom
## Multiple R-squared:  0.1238, Adjusted R-squared:  0.1237
## F-statistic: 822.9 on 1 and 5822 DF,  p-value: < 2.2e-16
```

```
pairs(boarding~hour_of_day, data=capmetro)
```



As we can see from this graph, boarding increases steadily over the course of the day (starting at 6:00 AM), and at around the 16th hour of the day, or 4:00 PM, the boarding peaks and begins to decrease fairly sharply until the end of the day (9:00 PM).

Portfolio modeling

We decided to use the ETFs SPY, VOO, QQQ, ARKK, and VNQ as they are all different types of ETFs which has SPY, a very safe ETF, and ARKK, a much more risky ETF.

```
library(mosaic)
```

```
## Warning: package 'mosaic' was built under R version 4.0.5
```

```
## 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 thi
s.
```

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

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

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

```
## 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: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
```

```r
library(quantmod)
```

```
## Loading required package: xts
```

```
## Warning: package 'xts' was built under R version 4.0.5
```

```
## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 4.0.5
```

```
##
## 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
```

```
## Warning: package 'TTR' was built under R version 4.0.5
```

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

```r
library(foreach)
```

```
## Warning: package 'foreach' was built under R version 4.0.5
```

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

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

```
myETFs = c("SPY", "VOO", "QQQ", "ARKK", "VNQ")
getSymbols(myETFs, from = "2017-07-31", to = "2022-07-31")
```

```
## [1] "SPY"  "VOO"  "QQQ"  "ARKK" "VNQ"
```

```
SPYa = adjustOHLC(SPY)
```

```
## Warning in read.table(file = file, header = header, sep = sep,
## quote = quote, : incomplete final line found by readTableHeader
## on 'https://query2.finance.yahoo.com/v7/finance/download/SPY?
## period1=-2208988800&period2=1660521600&interval=1d&events=split'
```

```
## Warning in read.table(file = file, header = header, sep = sep,
## quote = quote, : incomplete final line found by readTableHeader
## on 'https://query1.finance.yahoo.com/v7/finance/download/SPY?
## period1=-2208988800&period2=1660521600&interval=1d&events=split'
```

```
QQQa = adjustOHLC(QQQ)
```

```
## Warning in read.table(file = file, header = header, sep = sep,
## quote = quote, : incomplete final line found by readTableHeader
## on 'https://query2.finance.yahoo.com/v7/finance/download/QQQ?
## period1=-2208988800&period2=1660521600&interval=1d&events=split'
```

```
## Warning in read.table(file = file, header = header, sep = sep,
## quote = quote, : incomplete final line found by readTableHeader
## on 'https://query2.finance.yahoo.com/v7/finance/download/QQQ?
## period1=-2208988800&period2=1660521600&interval=1d&events=split'
```

```
VOOa = adjustOHLC(VOO)
```

```
## Warning in read.table(file = file, header = header, sep = sep,
## quote = quote, : incomplete final line found by readTableHeader
## on 'https://query2.finance.yahoo.com/v7/finance/download/VOO?
## period1=-2208988800&period2=1660521600&interval=1d&events=split'
```

```
## Warning in read.table(file = file, header = header, sep = sep,
## quote = quote, : incomplete final line found by readTableHeader
## on 'https://query2.finance.yahoo.com/v7/finance/download/VOO?
## period1=-2208988800&period2=1660521600&interval=1d&events=split'
```

```
ARKKa = adjustOHLC(ARKK)
```

```
## Warning in read.table(file = file, header = header, sep = sep,
## quote = quote, : incomplete final line found by readTableHeader
## on 'https://query1.finance.yahoo.com/v7/finance/download/ARKK?
## period1=-2208988800&period2=1660521600&interval=1d&events=split'
```

```
## Warning in read.table(file = file, header = header, sep = sep,
## quote = quote, : incomplete final line found by readTableHeader
## on 'https://query1.finance.yahoo.com/v7/finance/download/ARKK?
## period1=-2208988800&period2=1660521600&interval=1d&events=split'
```

```
VNQa = adjustOHLC(VNQ)
```

```
## Warning in read.table(file = file, header = header, sep = sep,
## quote = quote, : incomplete final line found by readTableHeader
## on 'https://query1.finance.yahoo.com/v7/finance/download/VNQ?
## period1=-2208988800&period2=1660521600&interval=1d&events=split'
```

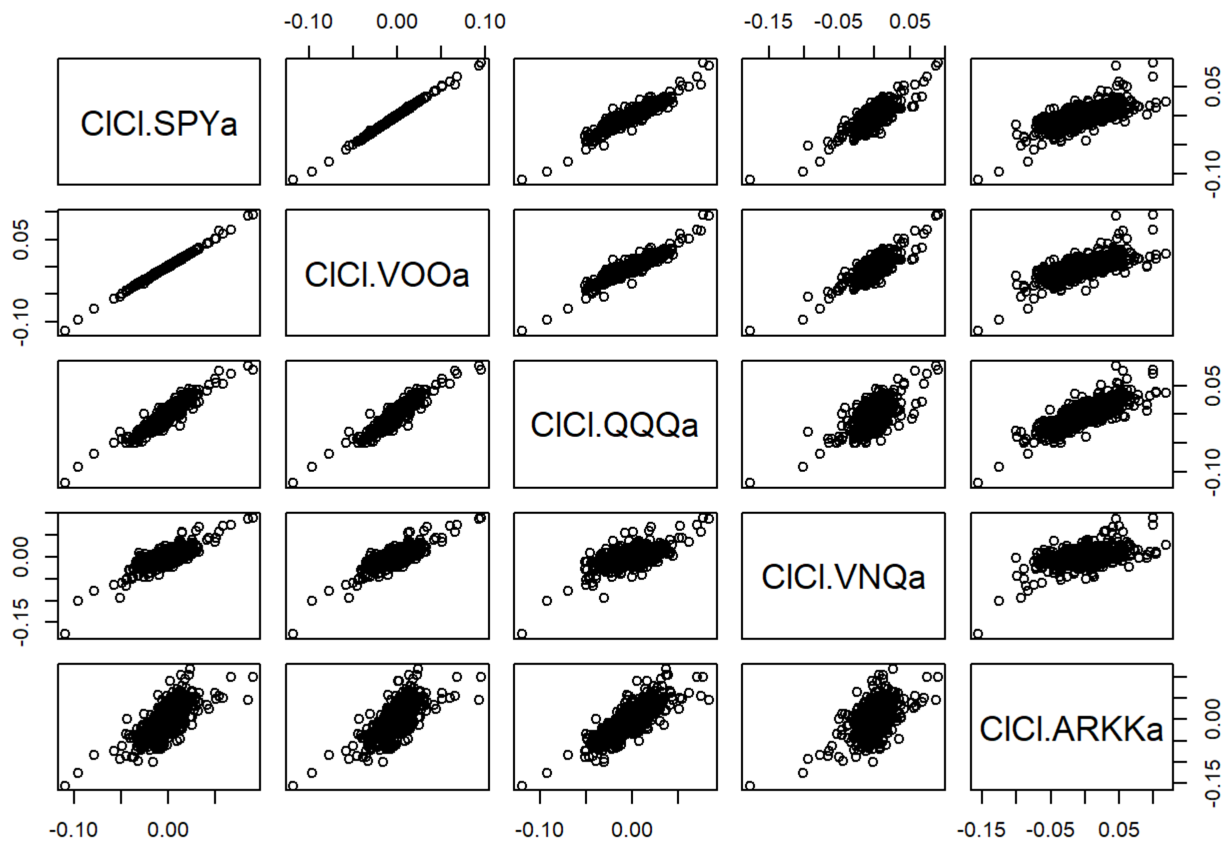
```
## Warning in read.table(file = file, header = header, sep = sep,
## quote = quote, : incomplete final line found by readTableHeader
## on 'https://query1.finance.yahoo.com/v7/finance/download/VNQ?
## period1=-2208988800&period2=1660521600&interval=1d&events=split'
```

```
all_returns = cbind(ClCl(SPYa),
                                    ClCl(VOOa),
                                    ClCl(QQQa),
                                    ClCl(VNQa),
                                    ClCl(ARKKa))

head(all_returns)
```

```
##                ClCl.SPYa     ClCl.VOOa    ClCl.QQQa    ClCl.VNQa  ClCl.ARKKa
## 2017-07-31            NA            NA           NA           NA          NA
## 2017-08-01  0.0022288082  0.0022502427  0.002304001  0.004509815 0.002633212
## 2017-08-02  0.0004851811  0.0005723311  0.002716627 -0.009097306 0.003771010
## 2017-08-03 -0.0019398440 -0.0017599173 -0.003890226 -0.002027018 0.007480209
## 2017-08-04  0.0018221452  0.0016748501  0.001813202  0.003345353 0.012152489
## 2017-08-07  0.0018592255  0.0016281352  0.006334870 -0.001071731 0.018749967
```

```
all_returns = as.matrix(na.omit(all_returns))
```

```
pairs(all_returns)
```



Low-Risk Model

```
return.today = resample(all_returns, 1, orig.ids=FALSE)

# Update the value of your holdings
# Assumes an equal allocation to each asset
total_wealth = 100000
my_weights = c(0.8,0.05,0.05, 0.05, 0.05)
holdings = total_wealth*my_weights
holdings = holdings*(1 + return.today)

# Compute your new total wealth
holdings
```

```
##            ClCl.SPYa ClCl.VOOa ClCl.QQQa ClCl.VNQa ClCl.ARKKa
## 2020-02-10  80597.23  5035.863  5060.428   5055.98   5094.019
```

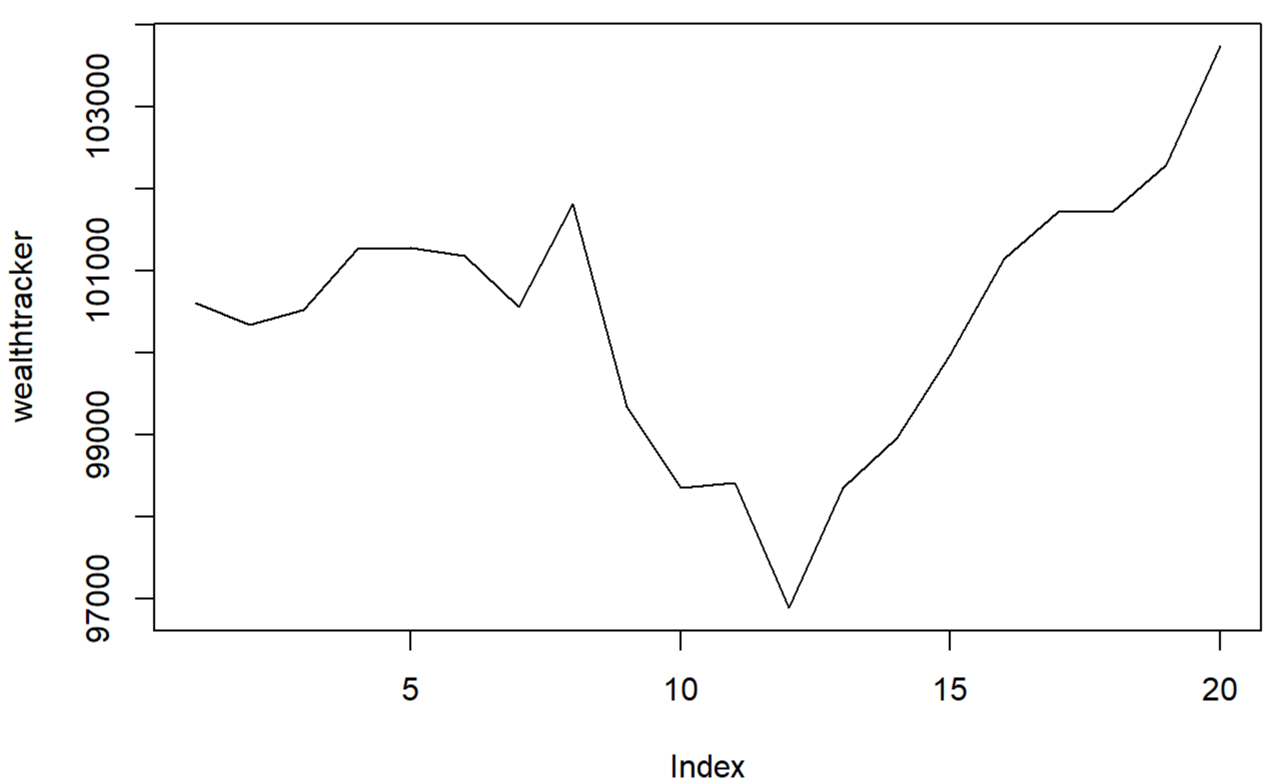
```
total_wealth = sum(holdings)
total_wealth
```

```
## [1] 100843.5
```

```
## begin block
total_wealth = 100000
weights = c(0.8, 0.05, 0.05, 0.05, 0.05)
holdings = weights * total_wealth
n_days = 20  # capital T in the notes
wealthtracker = rep(0, n_days) # Set up a placeholder to track total wealth
for(today in 1:n_days) {
    return.today = resample(all_returns, 1, orig.ids=FALSE)  # sampling from R matrix in notes
    holdings = holdings + holdings*return.today
    total_wealth = sum(holdings)
    wealthtracker[today] = total_wealth
}
total_wealth
```

```
## [1] 103738
```

```
plot(wealthtracker, type='l')
```

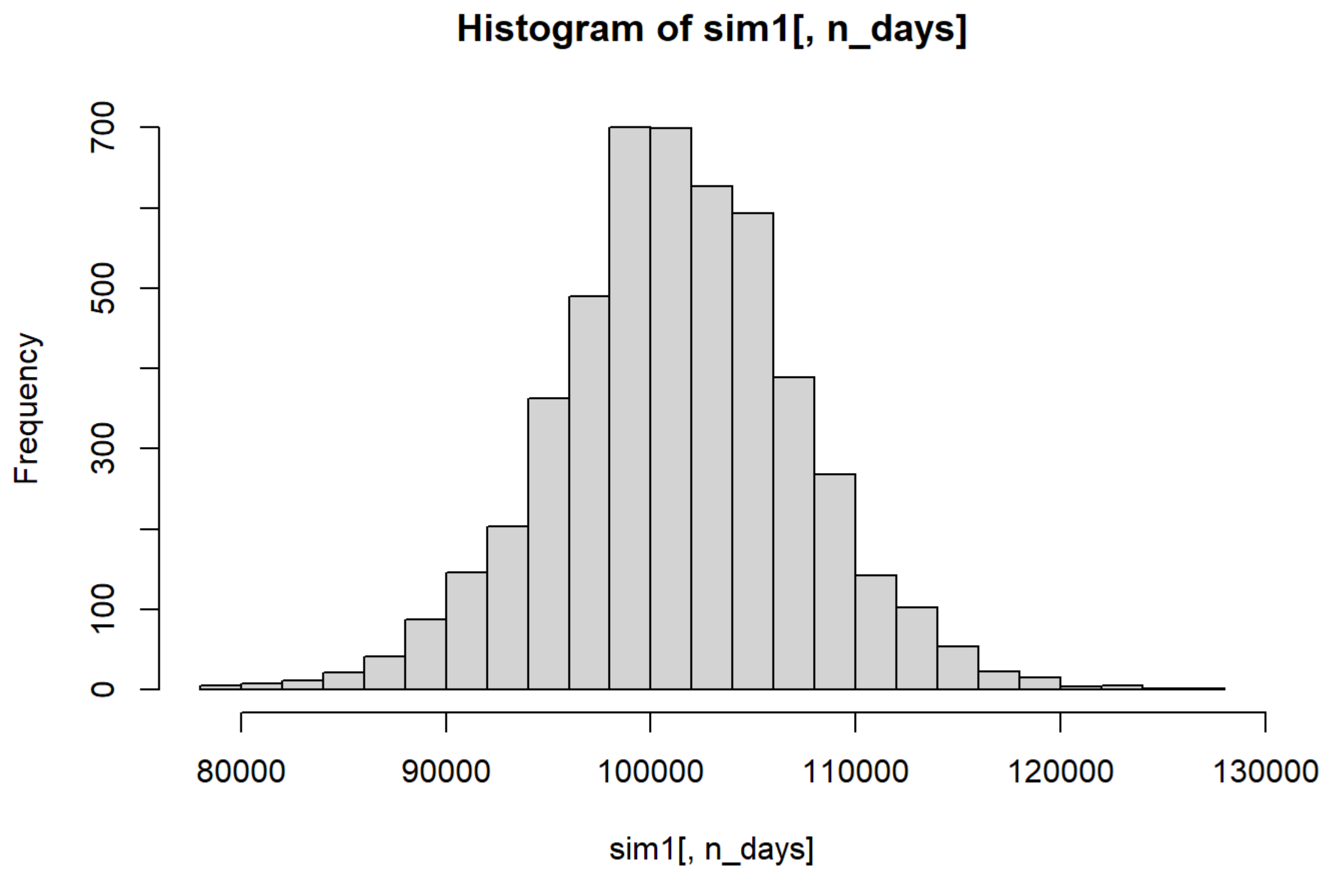


```
## end block
```

```
initial_wealth = 100000
sim1 = foreach(i=1:5000, .combine='rbind') %do% {
    total_wealth = initial_wealth
    weights = c(0.8, 0.05, 0.05, 0.05, 0.05)
    holdings = weights * total_wealth
    n_days = 20
    wealthtracker = rep(0, n_days)
    for(today in 1:n_days) {
        return.today = resample(all_returns, 1, orig.ids=FALSE)
        holdings = holdings + holdings*return.today
        total_wealth = sum(holdings)
        wealthtracker[today] = total_wealth
    }
    wealthtracker
}

# each row is a simulated trajectory
# each column is a data
head(sim1)
```

```
##               [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
## result.1  99540.04  99706.53 100332.24  99003.73  97440.48  97738.65  99214.36
## result.2 100757.99 100739.21 100858.19 101135.88 101693.26 100798.13 101292.92
## result.3  99832.61  99198.36  98973.96  99071.06  99305.60  99545.03 100256.64
## result.4 100637.02  98713.70  98753.43 101786.72 101954.20 102162.65 101622.72
## result.5 101720.18 102010.35 102026.43 102528.84 102511.86 102184.30 102605.43
## result.6 100804.18 100983.75  97912.39  98965.96  98226.38  99837.29 100439.54
##               [,8]      [,9]    [,10]    [,11]    [,12]    [,13]     [,14]
## result.1  99473.63  99677.89 100422.2 106283.7 106506.9 108253.5 108345.48
## result.2 100676.73 100732.69 100551.9 100453.6 100264.6  99303.8  99059.34
## result.3 101047.03 102434.51 103500.9 105240.9 104290.7 105306.3 105407.96
## result.4 101925.66 101898.63 101414.5 101964.4 104989.6 104035.6 103649.62
## result.5 101234.87 101943.62 102384.2 105596.5 104085.7 103447.7 102388.77
## result.6 100259.44 100718.21 100769.1 100843.2 100576.8 100729.7 100517.10
##              [,15]     [,16]     [,17]     [,18]     [,19]     [,20]
## result.1 110013.01 112009.15 112242.21 111970.14 111132.54 113262.77
## result.2  96464.73  96968.34  96979.34  96666.70  95860.48  95648.25
## result.3 105556.57 106079.35 105703.19 105885.69 105191.05  97003.82
## result.4 104452.00 104723.63 101734.80 102804.14  99739.67 100207.57
## result.5  98435.26  99947.34  99260.51  97726.02  98628.93  98199.14
## result.6 101210.85 101564.33 100556.95 100565.20 100760.22 100299.08
```

```
hist(sim1[,n_days], 25)
```



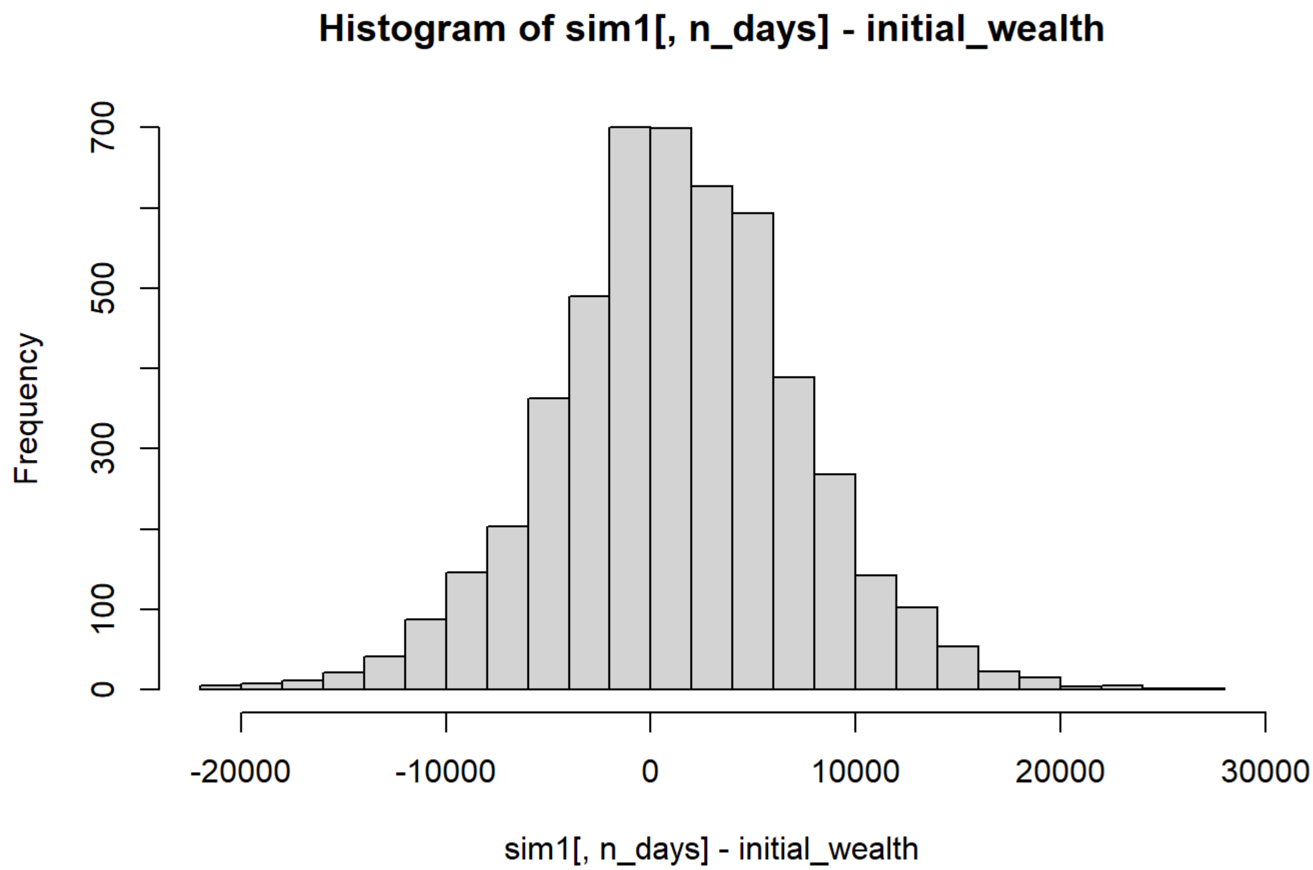

```
# Profit/loss
mean(sim1[,n_days])
```

```
## [1] 101264.7
```

```
mean(sim1[,n_days] - initial_wealth)
```

```
## [1] 1264.709
```

```
hist(sim1[,n_days]- initial_wealth, breaks=30)
```




```
# 5% value at risk:
quantile(sim1[,n_days]- initial_wealth, prob=0.05)
```

```
##        5%
## -8836.039
```

We expect lose only around $8000 in the worst case scenario.

Equal Investing

```
return.today = resample(all_returns, 1, orig.ids=FALSE)

# Update the value of your holdings
# Assumes an equal allocation to each asset
total_wealth = 100000
my_weights = c(0.2,0.2,0.2, 0.2, 0.2)
holdings = total_wealth*my_weights
holdings = holdings*(1 + return.today)

# Compute your new total wealth
holdings
```

```
##            ClCl.SPYa ClCl.VOOa ClCl.QQQa ClCl.VNQa ClCl.ARKKa
## 2021-07-23  20205.76  20199.81  20233.55  20165.74   20118.31
```

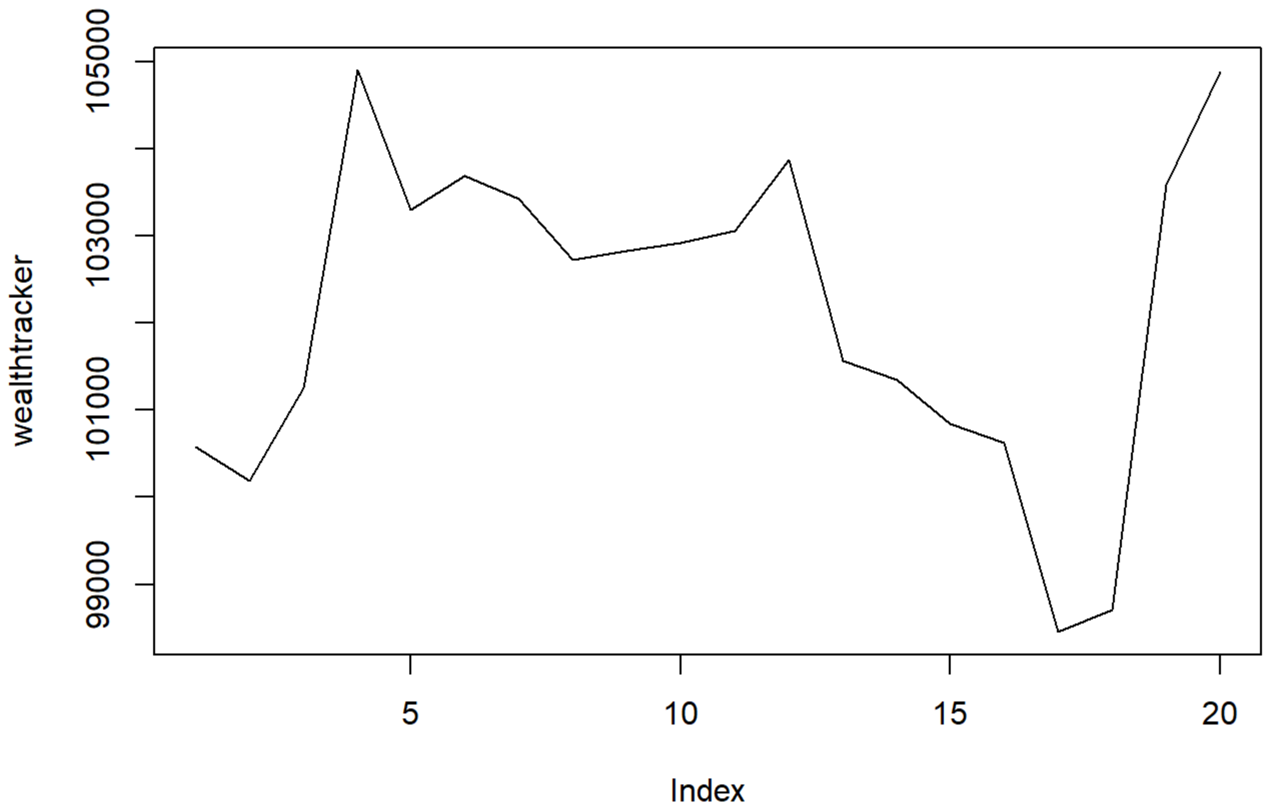
```
total_wealth = sum(holdings)
total_wealth
```

```
## [1] 100923.2
```

```
## begin block
total_wealth = 100000
weights = c(0.2,0.2,0.2, 0.2, 0.2)
holdings = weights * total_wealth
n_days = 20  # capital T in the notes
wealthtracker = rep(0, n_days) # Set up a placeholder to track total wealth
for(today in 1:n_days) {
    return.today = resample(all_returns, 1, orig.ids=FALSE)  # sampling from R matrix in notes
    holdings = holdings + holdings*return.today
    total_wealth = sum(holdings)
    wealthtracker[today] = total_wealth
}
total_wealth
```

```
## [1] 104886.1
```

```
plot(wealthtracker, type='l')
```

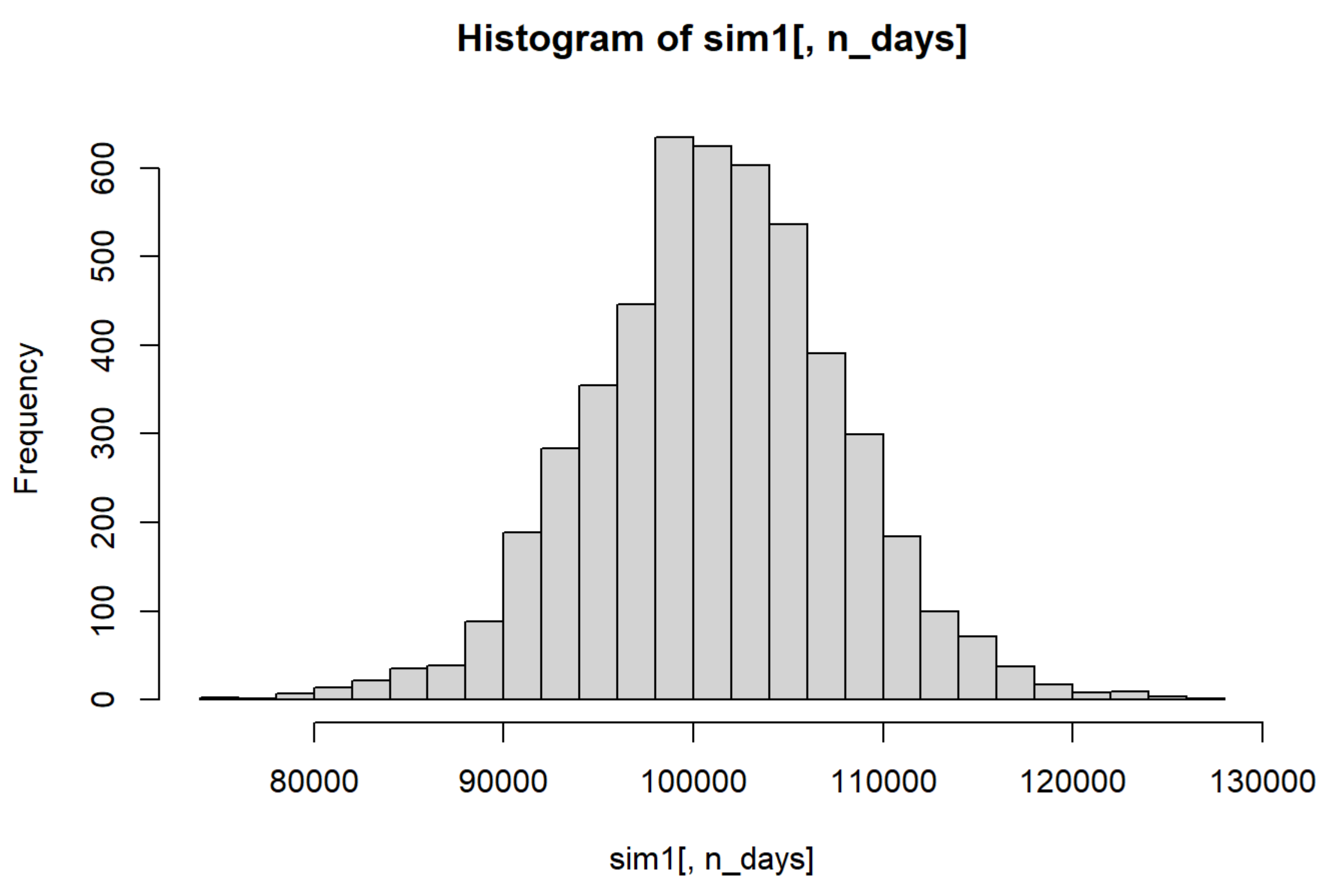



```
## end block
```

```
initial_wealth = 100000
sim1 = foreach(i=1:5000, .combine='rbind') %do% {
    total_wealth = initial_wealth
    weights = c(0.2,0.2,0.2,0.2,0.2)
    holdings = weights * total_wealth
    n_days = 20
    wealthtracker = rep(0, n_days)
    for(today in 1:n_days) {
        return.today = resample(all_returns, 1, orig.ids=FALSE)
        holdings = holdings + holdings*return.today
        total_wealth = sum(holdings)
        wealthtracker[today] = total_wealth
    }
    wealthtracker
}

# each row is a simulated trajectory
# each column is a data
head(sim1)
```

```
##               [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
## result.1  98707.56  98373.91  98030.85  97912.81  99720.99 100082.19  99844.42
## result.2 100381.43 101597.40 101619.97 101014.10 100992.77 102725.72 103518.48
## result.3 100758.23 102211.48 101976.25 101368.52  99672.00  99590.53  91927.71
## result.4  94325.67  98045.80 101494.24 100793.57 101110.83 101957.11 101233.04
## result.5 100352.27  99957.13 100753.34  99449.27  98989.45  99742.39 100561.89
## result.6 102812.18 103148.48 102541.44 104586.82 105506.51 105704.96 103655.15
##               [,8]      [,9]     [,10]     [,11]     [,12]     [,13]     [,14]
## result.1  99969.80 100356.89 101532.79 101980.74 101691.89 103890.52 102207.32
## result.2 105246.87 105909.73 106016.82 106176.97 106276.00 106068.57 108606.08
## result.3  92787.90  92619.56  92565.42  92740.63  90963.12  92300.29  93099.70
## result.4  97401.08  97274.04  97816.34  98342.21  98716.91  98037.94  96769.51
## result.5 101088.06 102559.62 103948.03 104285.45 105894.25 110072.59 109377.12
## result.6 103541.91 100984.28  98656.21  99834.24 102055.04 100897.75  98774.82
##              [,15]     [,16]     [,17]     [,18]     [,19]     [,20]
## result.1 103269.44 102979.79 102853.16 102729.58 104094.96 103278.61
## result.2 109060.74 108993.12 108620.31 109367.35 109102.76 110688.70
## result.3  94282.54  94714.24  95158.96  95266.66  95068.99  93969.55
## result.4  98296.77  95746.21  97561.56  98301.22  98104.43  98166.70
## result.5 109069.60 111118.71 110607.18 110726.71 110486.37 108037.64
## result.6  99170.68  99506.24  97796.64  97482.13  97565.49  98185.16
```

```
hist(sim1[,n_days], 25)
```



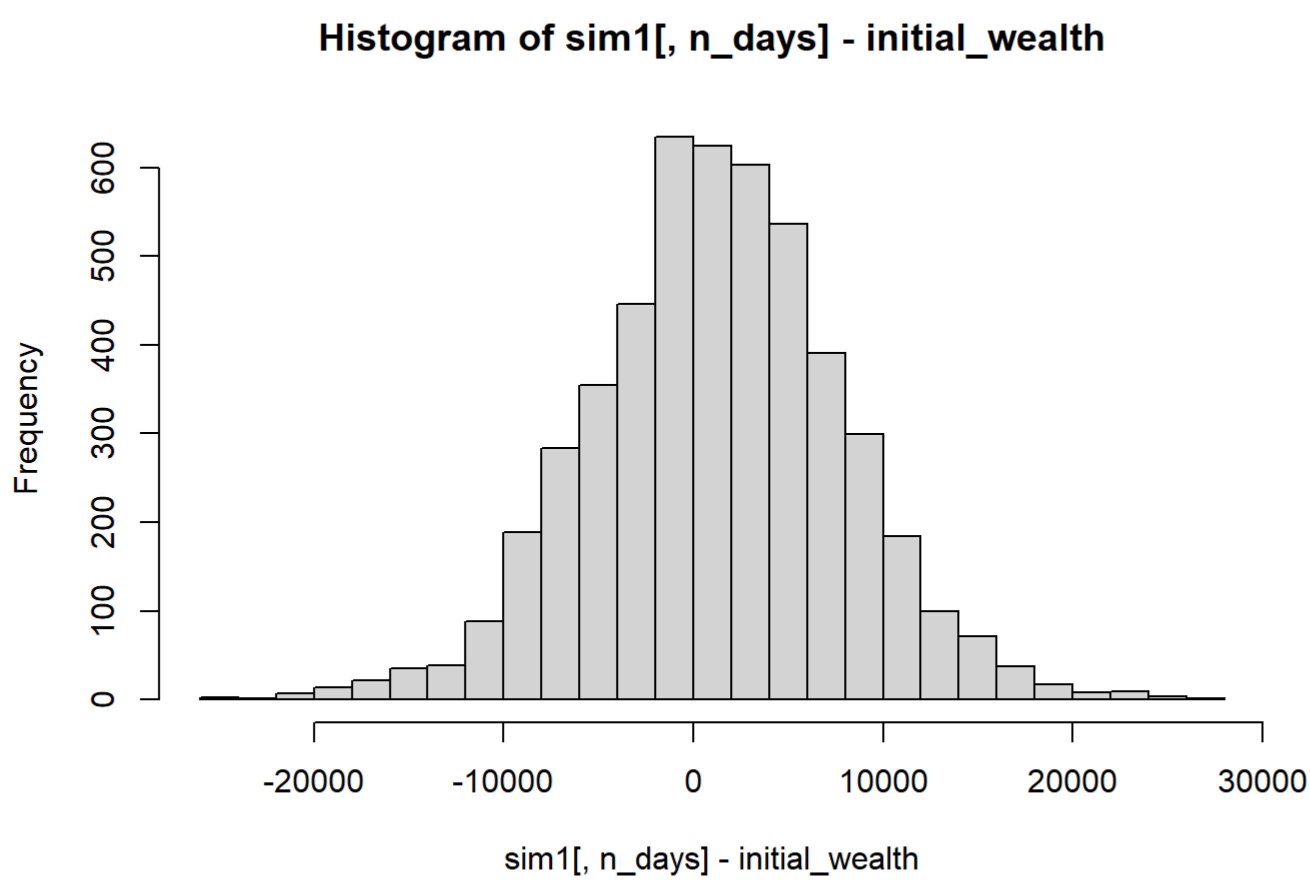
```
# Profit/loss
mean(sim1[,n_days])
```

```
## [1] 101210.5
```

```
mean(sim1[,n_days] - initial_wealth)
```

```
## [1] 1210.46
```

```
hist(sim1[,n_days]- initial_wealth, breaks=30)
```




```
# 5% value at risk:
quantile(sim1[,n_days]- initial_wealth, prob=0.05)
```

```
##        5%
## -9443.639
```

We expect to lose around $10,000 at the worst case scenario

High-Risk Model

```
return.today = resample(all_returns, 1, orig.ids=FALSE)

# Update the value of your holdings
# Assumes an equal allocation to each asset
total_wealth = 100000
my_weights = c(0.05,0.05,0.05, 0.8, 0.05)
holdings = total_wealth*my_weights
holdings = holdings*(1 + return.today)

# Compute your new total wealth
holdings
```

```
##            ClCl.SPYa ClCl.VOOa ClCl.QQQa ClCl.VNQa ClCl.ARKKa
## 2019-10-15  5049.501  5048.367  5063.583  80103.15   5105.301
```

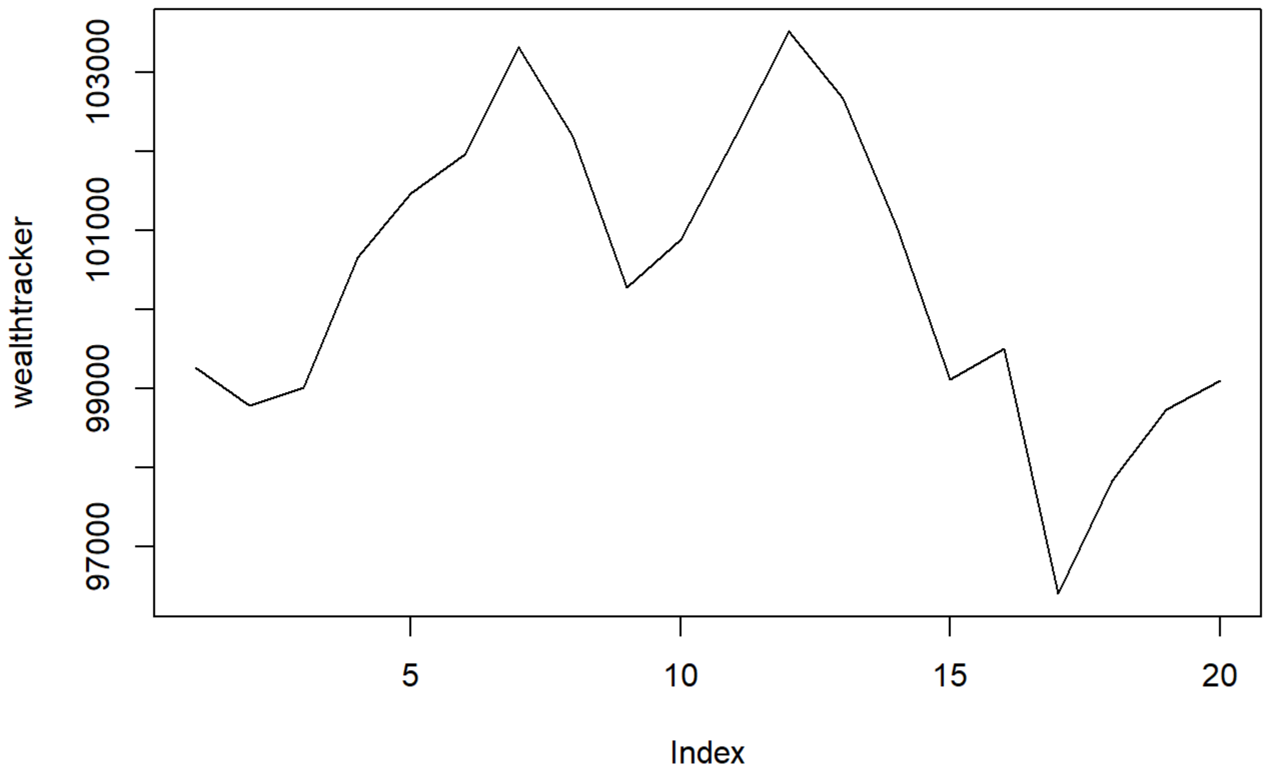
```
total_wealth = sum(holdings)
total_wealth
```

```
## [1] 100369.9
```

```
## begin block
total_wealth = 100000
weights = c(0.05,0.05,0.05, 0.8, 0.05)
holdings = weights * total_wealth
n_days = 20  # capital T in the notes
wealthtracker = rep(0, n_days) # Set up a placeholder to track total wealth
for(today in 1:n_days) {
    return.today = resample(all_returns, 1, orig.ids=FALSE)  # sampling from R matrix in notes
    holdings = holdings + holdings*return.today
    total_wealth = sum(holdings)
    wealthtracker[today] = total_wealth
}
total_wealth
```

```
## [1] 99105.53
```

```
plot(wealthtracker, type='l')
```

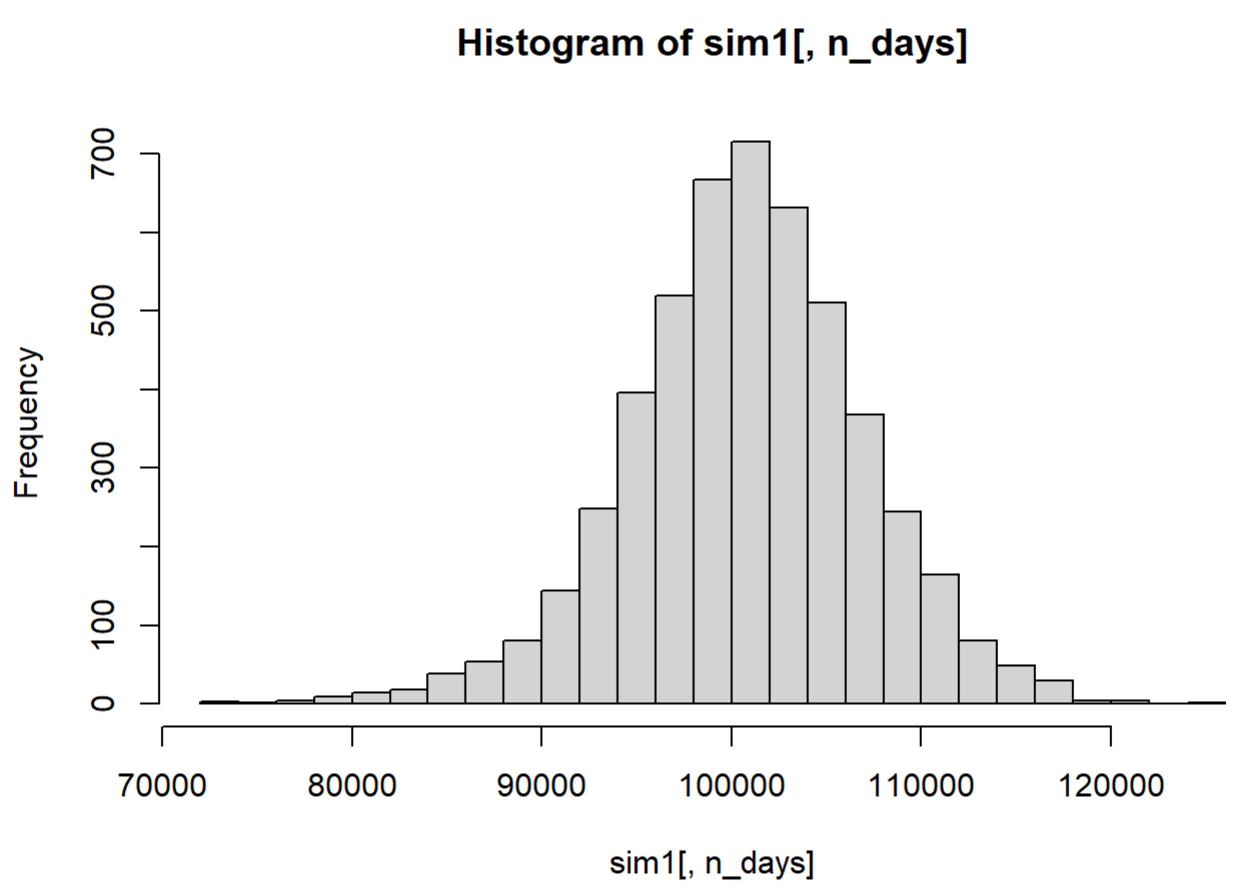



```
## end block
```

```
initial_wealth = 100000
sim1 = foreach(i=1:5000, .combine='rbind') %do% {
    total_wealth = initial_wealth
    weights = c(0.05,0.05,0.05, 0.8, 0.05)
    holdings = weights * total_wealth
    n_days = 20
    wealthtracker = rep(0, n_days)
    for(today in 1:n_days) {
        return.today = resample(all_returns, 1, orig.ids=FALSE)
        holdings = holdings + holdings*return.today
        total_wealth = sum(holdings)
        wealthtracker[today] = total_wealth
    }
    wealthtracker
}

# each row is a simulated trajectory
# each column is a data
head(sim1)
```

```
##               [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
## result.1  99621.83  98472.27  96957.75  96357.10  96027.70  97883.69 100322.21
## result.2 100827.70  97638.74  97839.37  98611.60  98249.83  98939.12  97814.34
## result.3  99970.03 100352.50  99640.34  97809.64  98183.58  98410.63  98150.14
## result.4 100280.72 100511.22  99733.74 100726.12 101680.47 101764.05 100846.80
## result.5 100219.71 100717.78 100971.38 100851.83 101702.97 100229.23  99615.87
## result.6 100284.90 102382.53 102006.60 102892.83 102680.07 102120.84 102103.43
##               [,8]      [,9]     [,10]     [,11]     [,12]     [,13]     [,14]
## result.1  99902.00  99934.94 100452.61 109502.05 110029.37 109545.70 110097.67
## result.2  98366.57  99059.96 100055.16 100115.67  97555.37  98890.09  98504.20
## result.3 105465.26 106494.04 107591.16 109088.39 108716.31 108138.32 115447.66
## result.4 100982.66 100946.81 102357.79 104219.85 102978.73 103651.85 101920.26
## result.5 100680.12  99969.30  99625.33  99862.88  98568.34  98762.28  98841.46
## result.6 102072.09 103935.59 104161.45 104399.39 104016.31 103978.24 104101.19
##              [,15]     [,16]     [,17]     [,18]     [,19]     [,20]
## result.1 110076.55 108307.68 108833.25 109683.93 109794.73 107433.59
## result.2  97079.93  97039.08  97186.18  98806.27  99458.98  99514.49
## result.3 116201.07 115275.88 113746.01 114554.01 111120.17 111315.75
## result.4 101869.21 101063.75 100010.91 100857.13 101578.71 100780.57
## result.5  97987.82  98896.64  99263.47  99667.44  98156.19  98951.53
## result.6 101884.54 102244.99 102268.96 101636.20 103698.22 103193.96
```

```
hist(sim1[,n_days], 25)
```



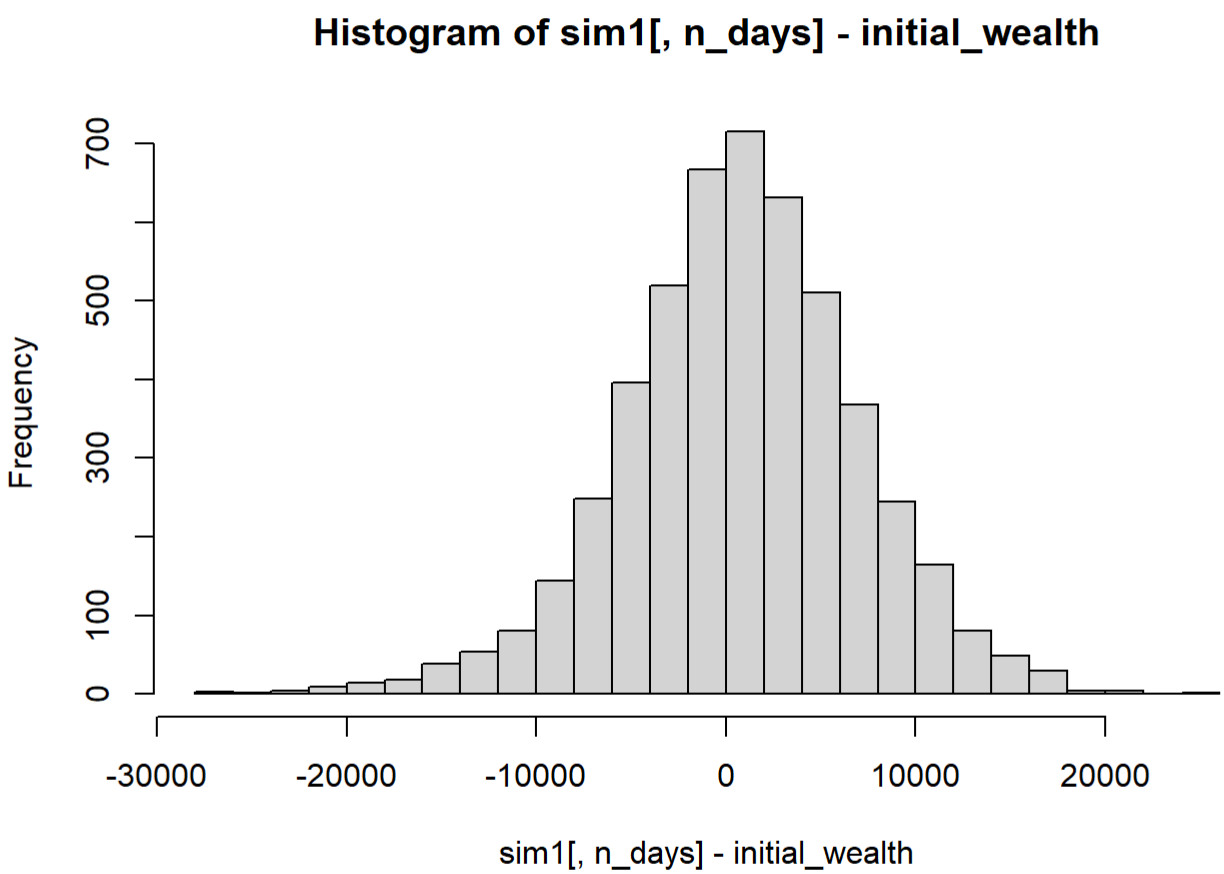
```
# Profit/loss
mean(sim1[,n_days])
```

```
## [1] 100763.6
```

```
mean(sim1[,n_days] - initial_wealth)
```

```
## [1] 763.5912
```

```
hist(sim1[,n_days]- initial_wealth, breaks=30)
```




```
# 5% value at risk:
quantile(sim1[,n_days]- initial_wealth, prob=0.05)
```

```
##        5%
## -9451.151
```

We expect to lose around $10,000 at the worst case scenario.

Clustering and PCA

```
# PCA
wine <- read_csv("wine.csv")
```

```
## Rows: 6497 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): color
## dbl (12): fixed.acidity, volatile.acidity, citric.acid, residual.sugar, chlo...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
library(ggplot2)
library(tidyverse)
head(wine)
```

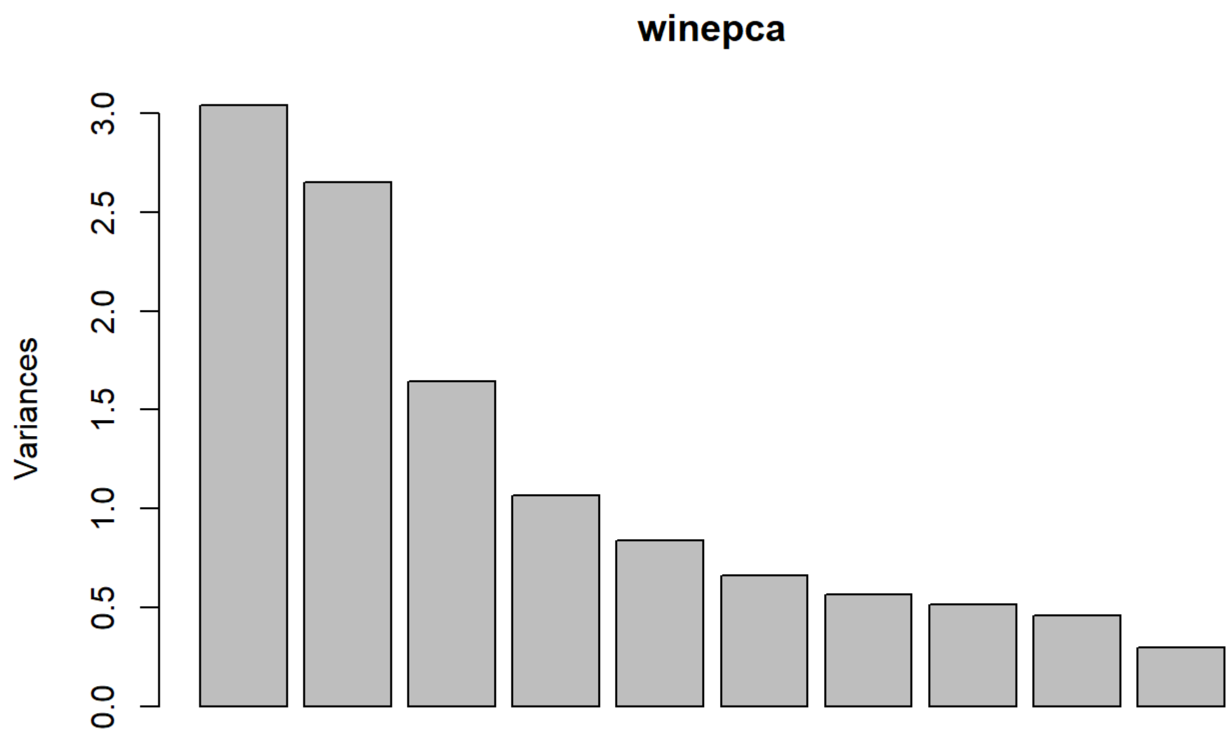
fixed.acidity <dbl>	volatile.acidity <dbl>	citric.acid <dbl>	residual.sugar <dbl>	chlorides <dbl>	free.sulfur.dic
7.4	0.70	0.00	1.9	0.076	
7.8	0.88	0.00	2.6	0.098	
7.8	0.76	0.04	2.3	0.092	
11.2	0.28	0.56	1.9	0.075	
7.4	0.70	0.00	1.9	0.076	
7.4	0.66	0.00	1.8	0.075	

6 rows | 1-6 of 13 columns

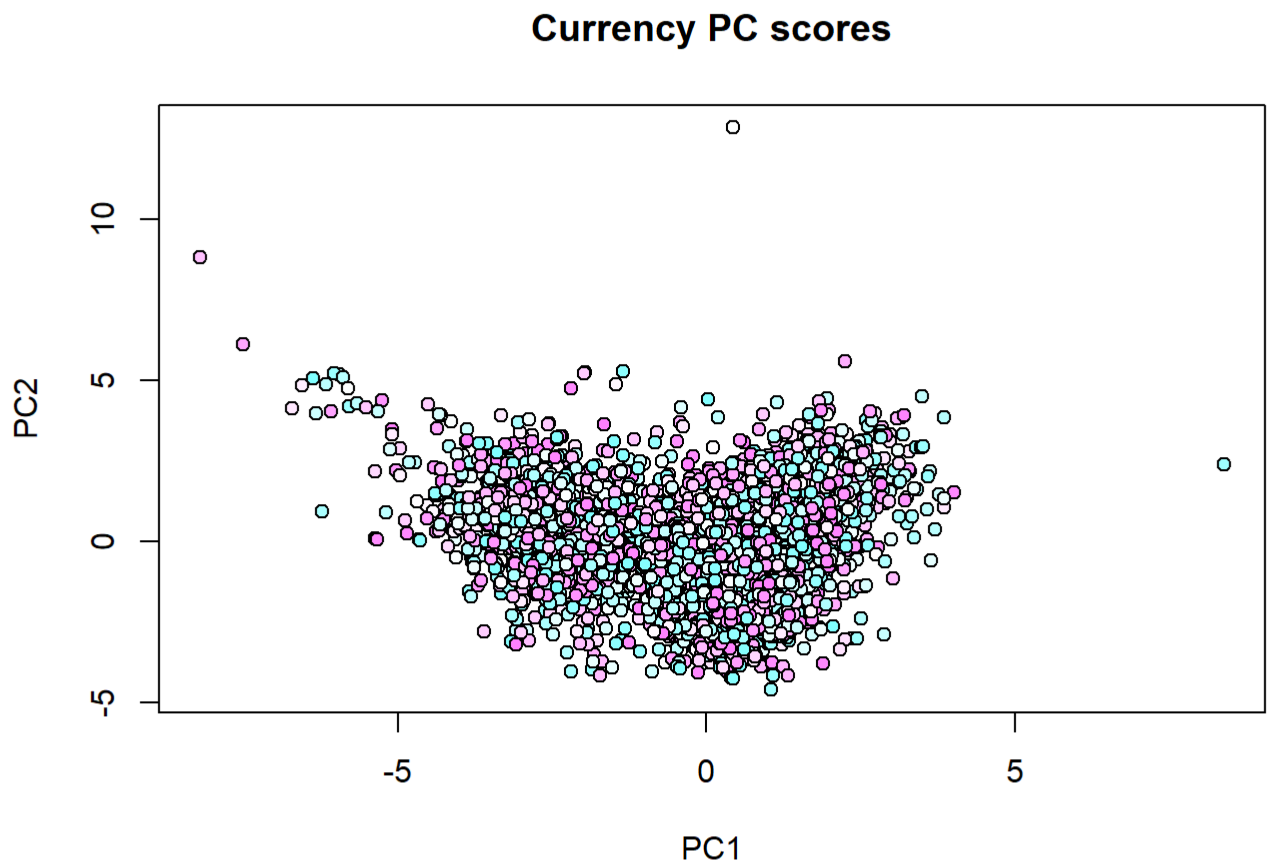
```
Z = scale(wine[,1:12], center=TRUE, scale=FALSE)
winepca = prcomp(Z, rank=6, scale=TRUE)
summary(winepca)
```

```
## Importance of first k=6 (out of 12) components:
##                           PC1    PC2    PC3     PC4     PC5     PC6
## Standard deviation     1.7440 1.6278 1.2812 1.03374 0.91679 0.81265
## Proportion of Variance 0.2535 0.2208 0.1368 0.08905 0.07004 0.05503
## Cumulative Proportion  0.2535 0.4743 0.6111 0.70013 0.77017 0.82520
```

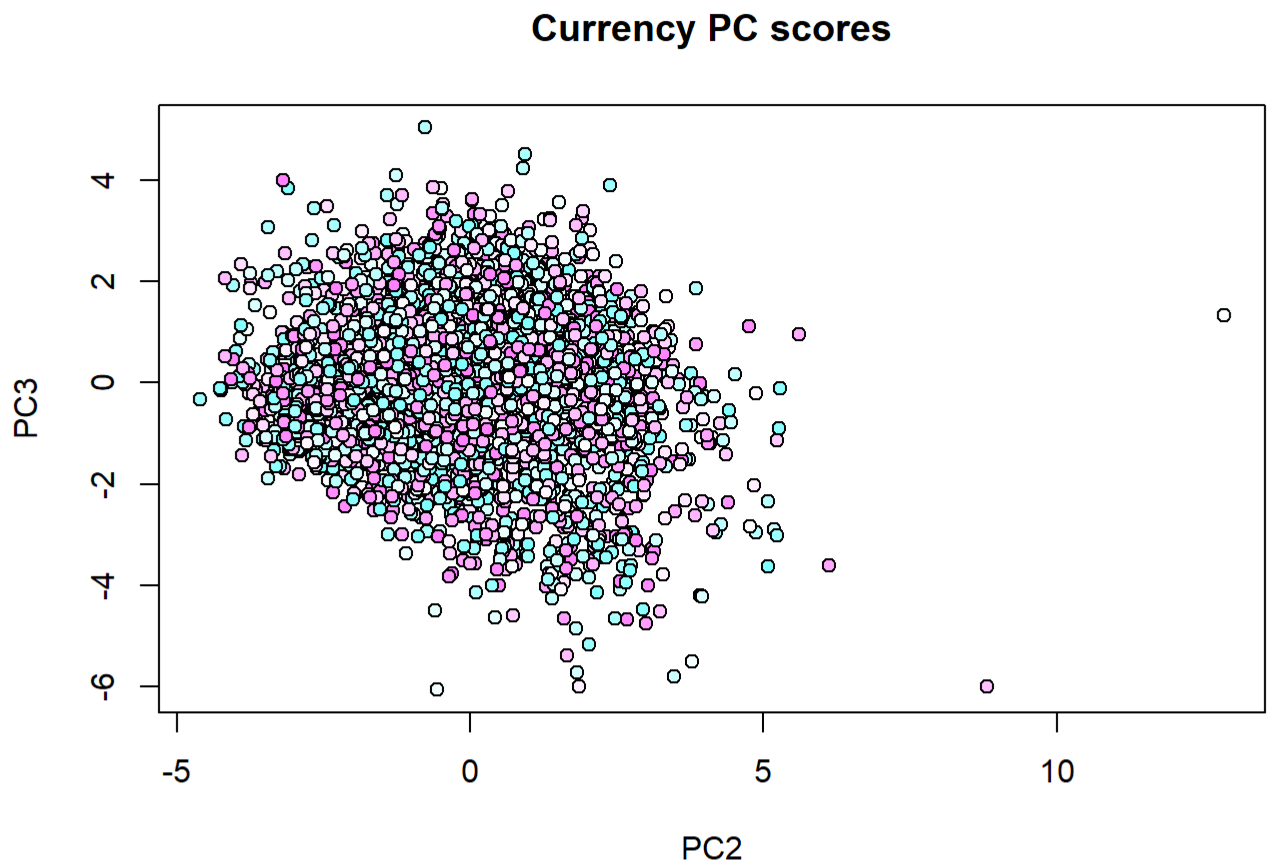
```
plot(winepca)
```



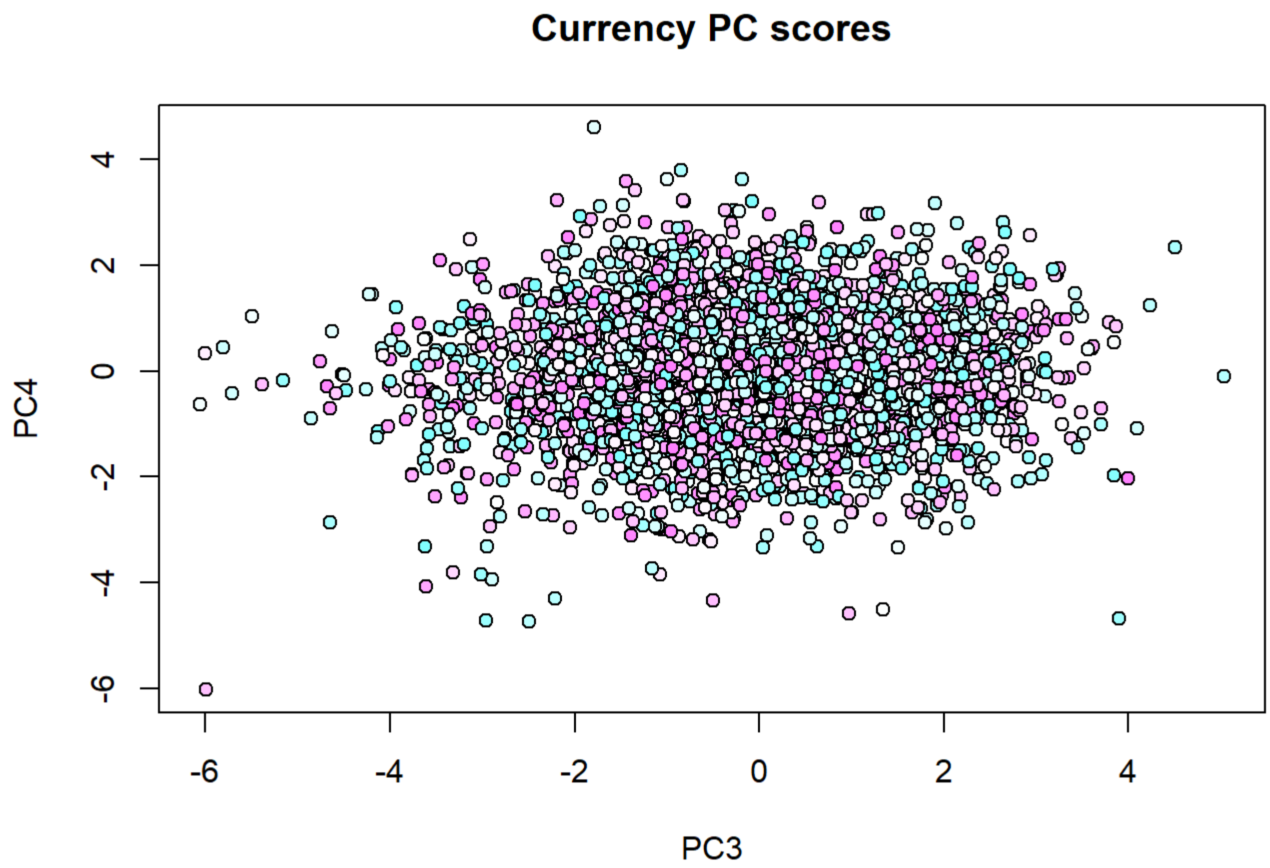

```
scores=predict(winepca)
plot(scores[,1:2], pch=21, bg=cm.colors(120)[120:1], main="Currency PC scores")
```



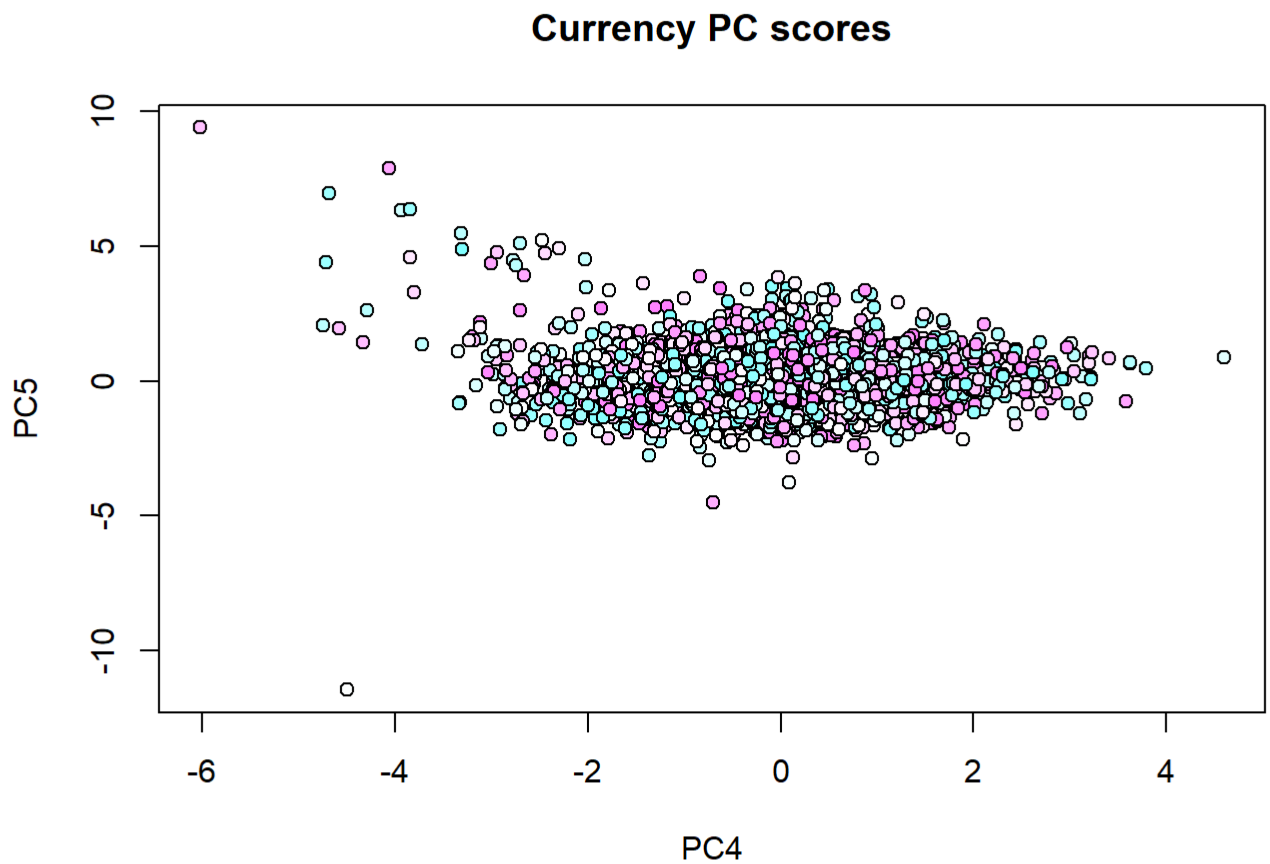

```
plot(scores[,2:3], pch=21, bg=cm.colors (120)[120:1], main="Currency PC scores")
```



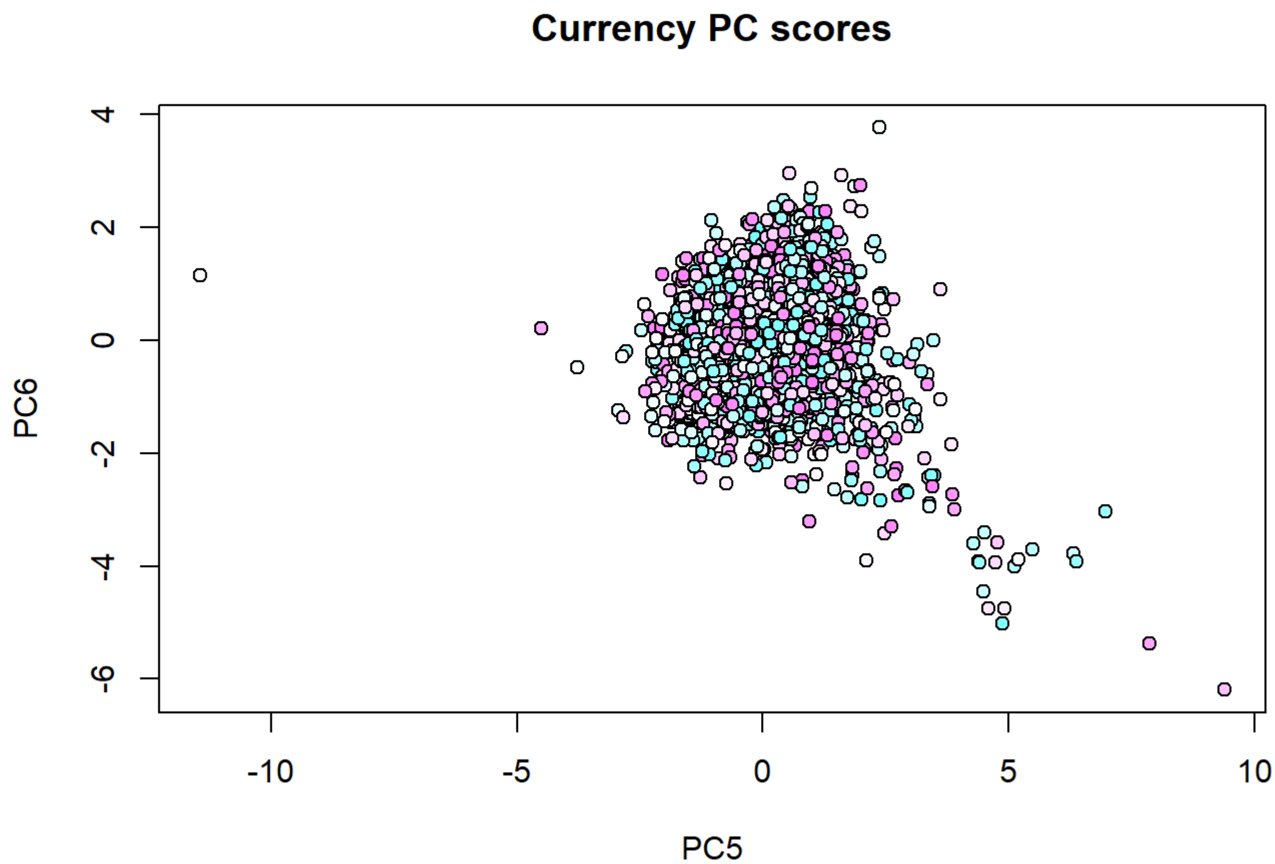

```
plot(scores[,3:4], pch=21, bg=cm.colors(120)[120:1], main="Currency PC scores")
```




```
plot(scores[,4:5], pch=21, bg=cm.colors(120)[120:1], main="Currency PC scores")
```




```
plot(scores[,5:6], pch=21, bg=cm.colors(120)[120:1], main="Currency PC scores")
```




```
# K Means Clustering:

# Loading package
library(ClusterR)
```

```
## Warning: package 'ClusterR' was built under R version 4.0.5
```

```
## Loading required package: gtools
```

```
## Warning: package 'gtools' was built under R version 4.0.5
```

```
##
## Attaching package: 'gtools'
```

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

```
library(cluster)
```

```
# Removing initial label of
# Species from original dataset
wine <- read_csv("wine.csv")
```

```
## Rows: 6497 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): color
## dbl (12): fixed.acidity, volatile.acidity, citric.acid, residual.sugar, chlo...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
wine_1 <- wine[,1:12]
```

```
# Fitting K-Means clustering Model
# to training dataset
set.seed(240) # Setting seed
kmeans.re <- kmeans(wine_1, centers = 2, nstart = 25)
kmeans.re
```

```
## K-means clustering with 2 clusters of sizes 3689, 2808
##
## Cluster means:
##   fixed.acidity volatile.acidity citric.acid residual.sugar  chlorides
## 1      6.904812        0.2871659   0.3397642       7.244809 0.04859257
## 2      7.623219        0.4086378   0.2908725       3.076425 0.06580983
##   free.sulfur.dioxide total.sulfur.dioxide   density       pH sulphates
## 1            39.75590            155.69246 0.9947903 3.190808 0.4999485
## 2            18.39868             63.26318 0.9945736 3.254882 0.5724145
##    alcohol  quality
## 1 10.25932 5.824343
## 2 10.79722 5.810541
##
## Clustering vector:
##    [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2
##   [38] 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2
##   [75] 2 2 2 2 2 1 2 2 2 2 2 2 1 2 1 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2
##  [112] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2
##  [149] 2 2 2 2 2 2 1 1 1 1 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [186] 2 2 2 1 1 1 2 1 2 2 1 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 2 2 1 2 2
##  [223] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2
##  [260] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [297] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
##  [334] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [371] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2 2 2 2
##  [408] 2 2 2 2 2 2 2 1 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [445] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [482] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2
##  [519] 2 2 2 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [556] 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
##  [593] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [630] 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [667] 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2
##  [704] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [741] 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2
##  [778] 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [815] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [852] 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [889] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [926] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [963] 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1000] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1037] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1074] 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1111] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1148] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1185] 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1222] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1259] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1296] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1333] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1370] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2
```

```
## [1407] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1444] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1481] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1518] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1555] 2 2 2 2 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1592] 2 2 2 2 2 2 2 2 1 1 2 1 1 2 1 1 1 1 2 2 2 1 1 1 2 2 1 1 2 2 1 1 1 1 1 1 1
## [1629] 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 2 1 1
## [1666] 1 1 2 2 1 1 1 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1
## [1703] 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 2
## [1740] 2 1 2 2 2 1 1 2 2 1 1 1 1 2 2 1 1 1 1 1 1 1 2 1 1 1 1 2 1 2 1 2 2 1 1 1 2
## [1777] 1 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 1
## [1814] 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 2 2 1 1
## [1851] 1 1 2 1 1 2 2 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 1 1 1 1 1 1 1 1
## [1888] 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 2 1 1 1 1
## [1925] 1 1 1 1 2 2 1 2 1 2 2 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2
## [1962] 1 1 2 1 1 1 1 2 1 1 1 1 2 2 1 2 1 1 2 1 1 1 1 2 1 1 1 1 1 2 1 2 1 1 2 2 1
## [1999] 2 2 1 2 1 1 1 2 1 1 2 1 1 2 2 1 1 2 1 2 1 1 1 1 1 1 1 1 1 2 1 1 2 1 1 2 2
## [2036] 1 1 1 2 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1 2 1 1 1 1 2
## [2073] 1 2 1 2 1 1 1 1 2 1 1 1 1 1 2 1 1 2 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [2110] 1 1 1 2 1 1 1 1 2 2 1 1 2 2 2 1 2 1 2 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1
## [2147] 2 1 1 1 2 1 1 1 2 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1 1 2
## [2184] 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 2 1 2 1 1 1 1 2 1 1 1 1
## [2221] 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1
## [2258] 1 1 2 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [2295] 1 2 1 1 1 1 1 2 1 1 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 1
## [2332] 2 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 1
## [2369] 1 1 1 1 1 1 2 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 1 1 1 2 1 1 1 1 1 1
## [2406] 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 2 1 2 1 1 1 2 2 1 1 2 2 1 2 1 1 1 1 1 1 1
## [2443] 2 1 1 1 2 1 2 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 2 2 1 1 2 1 2
## [2480] 1 1 1 1 1 2 1 2 1 2 1 1 1 1 1 2 1 2 2 1 1 1 1 1 1 2 2 1 1 1 1 1 1 2 2 2 1
## [2517] 1 2 2 1 1 1 1 1 2 2 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 2 1 1 2 1 1 2 2 1 1 2 1
## [2554] 1 1 1 2 2 1 1 2 1 1 1 2 1 1 2 2 2 2 1 2 1 2 1 1 1 2 2 1 2 2 1 1 1 1 1 1 1
## [2591] 2 1 2 1 1 2 1 1 1 2 1 1 1 1 2 1 2 1 1 2 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2
## [2628] 1 1 1 1 1 1 1 1 2 2 2 2 1 2 2 1 2 2 2 2 2 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 1
## [2665] 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1
## [2702] 1 1 1 1 1 2 2 1 2 2 1 2 2 2 2 1 2 1 2 1 2 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1
## [2739] 2 2 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 2 1 2 1 2 1 1 1 1 2 1 1
## [2776] 1 1 1 2 1 1 2 1 1 1 1 2 1 2 1 1 2 1 1 1 1 1 2 2 2 2 1 2 2 1 1 1 1 1 2 1 1
## [2813] 2 2 1 2 1 1 2 1 1 1 1 2 2 2 2 2 2 1 1 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 2 1
## [2850] 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [2887] 1 2 2 2 1 2 2 2 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 2 1 2 1 1 1
## [2924] 1 1 1 1 2 2 2 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1
## [2961] 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 1 1 1 2 1 1 1 2 2 2 1 2 1 1
## [2998] 1 2 1 1 1 1 2 1 2 2 1 1 2 2 1 1 1 2 1 1 1 2 1 2 2 1 1 1 1 1 2 2 1 2 2 2 1
## [3035] 2 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1
## [3072] 2 1 2 1 1 2 1 1 1 1 1 2 1 1 2 1 1 1 1 2 1 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 1
## [3109] 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 1 2 2 1 2 2 2 2 1
## [3146] 1 1 2 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 2 1 1 2 1 1 1 1 1 1 2 1 2 1 1 1 1 2 1
## [3183] 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 2 1 1 2 2 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1
## [3220] 2 1 1 1 1 1 1 1 2 2 1 2 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 2
## [3257] 1 1 1 1 1 1 1 1 1 2 1 2 1 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [3294] 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 2 1 1 1 2
```

```
## [3331] 1 1 1 1 1 2 1 2 2 1 1 1 1 1 1 2 1 2 1 1 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 1 1
## [3368] 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1
## [3405] 1 1 1 1 1 1 1 2 2 2 1 1 1 2 1 1 2 2 1 1 2 1 1 1 1 1 1 1 1 1 1 2 2 1 2 1 2
## [3442] 1 1 2 1 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1
## [3479] 2 1 2 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 2 1 1 1 1 2 1 1 2 1 1 1 2
## [3516] 1 2 1 1 1 1 1 2 2 2 2 1 1 1 1 1 2 1 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [3553] 1 1 1 2 2 1 2 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2
## [3590] 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 2 1 1 1 2 2 1 2 1 1 1 1 1 1 1 1 2
## [3627] 1 1 1 1 1 2 1 1 2 1 1 2 2 1 2 2 2 1 2 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 2 1 1
## [3664] 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1
## [3701] 2 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [3738] 2 2 1 1 1 2 1 1 2 1 2 2 2 1 2 2 1 1 2 2 2 2 1 2 1 2 1 1 1 1 1 1 1 1 1 1 1
## [3775] 1 1 1 1 2 1 1 1 1 2 2 2 1 2 1 1 1 1 1 2 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [3812] 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2
## [3849] 1 1 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 2 1 1 1 2 1 1 2 2 1 1 1 1
## [3886] 1 1 1 2 2 1 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1
## [3923] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 1 2 1 1 2 1 1 1 1 1 2 2 1 1 2 1 1
## [3960] 1 2 1 1 1 1 1 1 1 1 2 1 2 2 1 1 1 2 1 1 1 1 2 2 2 1 1 1 2 2 1 1 1 1 1 1 1
## [3997] 2 2 2 2 2 1 1 1 1 2 2 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [4034] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 2 1 1
## [4071] 1 2 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1
## [4108] 1 1 1 1 2 1 1 2 1 1 2 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 2 1 2 1 1 2 1 1 1 2 1
## [4145] 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 2 1 1 1 2 1 1 1 1 2 2 1 1 1 1 1 2 1 1 1
## [4182] 1 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 2 2 1 1 1 2 1 1 2 1 2 1 1 1 1
## [4219] 1 1 1 1 1 2 1 1 1 1 1 2 2 1 1 1 2 1 1 2 2 2 1 2 1 1 1 2 1 1 1 1 1 2 1 1 1
## [4256] 1 1 1 2 1 1 1 2 2 1 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1
## [4293] 1 2 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 2 1 1 1 2 1 1 2
## [4330] 1 1 1 2 1 1 1 2 1 2 1 1 1 2 2 2 1 1 1 1 1 1 1 2 2 1 1 2 2 1 1 1 1 1 1 1 2
## [4367] 1 1 1 1 1 1 2 1 1 1 1 2 2 1 1 1 1 1 1 2 1 1 1 1 1 1 2 1 2 2 1 1 1 1 1 2 2
## [4404] 2 1 1 1 1 1 2 1 2 1 2 2 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2 2 2
## [4441] 2 2 2 2 2 2 2 1 1 1 2 1 2 2 1 1 2 1 1 1 2 2 2 2 1 1 1 1 2 1 2 1 2 1 2 1 1
## [4478] 1 2 2 2 1 2 1 2 2 2 2 1 1 1 1 1 2 1 1 1 2 1 2 2 1 2 1 1 1 2 2 2 1 1 2 1 2
## [4515] 2 1 1 2 1 2 1 1 1 1 1 2 1 1 1 1 2 1 1 2 2 2 1 2 2 1 2 1 2 1 1 2 2 1 1 2 2
## [4552] 2 2 2 1 2 2 2 2 1 1 2 1 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 2 1 1 2 1 1 1 2
## [4589] 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1 1 1 2 2 2 2 2 2 1 1 2 2 2 1 2 2 1
## [4626] 1 1 1 1 1 1 2 1 1 2 1 1 2 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 2 1 2 1 1 2 1 1 1
## [4663] 1 1 1 1 1 1 2 1 2 1 1 1 2 1 1 2 1 2 1 2 2 2 2 2 1 2 2 2 1 1 1 2 2 2 1 1 1
## [4700] 1 2 1 1 1 1 1 1 1 1 2 2 1 1 1 1 2 2 1 2 1 1 2 2 1 1 1 2 2 1 1 1 1 1 1 1 2
## [4737] 1 1 1 1 2 1 2 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 2 1 1
## [4774] 1 1 2 1 2 2 2 2 1 2 2 1 2 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 2
## [4811] 1 1 1 2 2 1 2 2 2 2 2 1 1 1 2 1 1 1 1 1 2 2 2 1 1 1 1 1 2 1 1 1 2 2 1 1 1
## [4848] 1 1 1 1 2 2 1 1 1 1 1 1 1 2 1 2 1 1 2 1 1 1 1 2 2 1 1 2 1 1 1 1 1 1 1 1 2
## [4885] 1 1 1 1 1 2 2 1 2 1 1 1 1 1 2 2 2 2 2 1 2 1 1 1 2 1 1 2 2 1 2 2 2 2 1 1 2
## [4922] 2 2 1 1 1 2 1 1 1 1 1 1 2 1 1 1 2 2 1 2 2 1 1 1 1 1 2 2 1 2 2 2 1 1 1 2 2
## [4959] 2 2 2 1 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 2 1 1 1
## [4996] 1 1 1 2 1 1 1 2 2 1 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1
## [5033] 2 2 2 2 1 1 1 2 1 2 2 1 1 1 2 1 1 2 2 1 2 2 2 1 1 1 1 2 1 1 2 1 1 1 1 2 1
## [5070] 1 2 1 2 1 1 2 1 1 2 2 1 2 2 2 1 2 1 2 2 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1
## [5107] 1 1 1 1 2 1 2 2 2 1 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 2 2 1 2 2 1 1 1
## [5144] 1 1 1 1 1 1 1 1 1 1 2 1 2 2 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 2 1
## [5181] 1 2 2 2 2 1 1 1 1 2 1 1 1 2 1 1 1 1 1 2 1 1 1 2 1 1 2 1 1 1 2 1 1 1 2 1 1
## [5218] 1 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 1 2 2 1 2 2 1 1 1 1 1 1 2 2 1 1 1 1 2 1 2
```

```
## [5255] 1 1 1 1 1 1 1 2 1 1 1 1 1 2 2 1 2 2 2 1 1 2 1 2 2 1 1 1 1 1 1 1 2 1 1 1 1
## [5292] 2 1 2 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 1 2
## [5329] 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 2 2 2 2 1 1
## [5366] 1 1 1 1 1 1 1 2 1 2 1 2 2 2 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 2 2 1 1 2 2 1
## [5403] 2 2 2 2 2 2 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 2 1 2 1 2 1 1 1 1 1 1 2 2 1 1
## [5440] 2 1 1 2 2 1 2 1 2 2 2 1 2 1 1 1 2 2 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 2 1
## [5477] 2 1 2 1 1 1 1 1 2 1 2 1 1 2 1 1 2 1 2 1 1 1 1 2 2 2 2 2 2 2 2 1 1 1 1 2 2
## [5514] 1 2 1 1 1 2 1 1 2 2 2 2 2 2 1 1 2 2 1 2 2 1 2 1 2 1 1 1 2 1 1 1 1 1 2 1 1
## [5551] 1 2 1 1 2 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 1 2 2 1
## [5588] 1 2 1 2 2 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 2 1 1 1 1 1 1 2 1 1 1 2 2 2 1 1 1
## [5625] 2 1 1 2 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 1 2 1 1 1 2 1 1 1 2 2 2
## [5662] 1 1 2 1 1 1 1 1 1 2 1 2 2 1 1 2 2 2 1 1 1 1 2 2 2 2 2 1 1 2 1 2 2 2 1 2 1
## [5699] 1 2 1 1 2 2 1 1 1 2 1 1 2 2 2 2 2 1 1 1 1 1 1 1 2 2 1 1 1 1 2 1 1 1 1 2 1
## [5736] 1 1 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 1 1 2 1
## [5773] 2 1 1 1 1 1 1 1 2 1 2 2 1 2 1 1 2 2 1 2 2 2 2 2 1 1 2 2 2 1 1 1 2 1 1 2 2
## [5810] 1 1 1 2 1 1 1 1 1 1 2 1 1 2 2 1 1 2 1 1 1 2 2 1 1 1 1 2 1 2 1 1 1 2 1 1 2
## [5847] 1 1 1 2 2 2 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2
## [5884] 2 2 2 1 2 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 2 1 1 2 1 2 1 1 2 1 1 1 1
## [5921] 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 2 2 2 1 1 1 2 1 1 1 1 1 1 1
## [5958] 1 1 1 1 1 1 1 1 1 1 2 1 1 2 1 2 1 1 1 2 1 1 1 1 1 2 1 1 1 2 2 1 1 1 1 1 1
## [5995] 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2
## [6032] 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2
## [6069] 2 2 2 2 1 1 1 2 1 1 1 1 1 1 2 1 2 2 2 1 1 1 2 2 1 2 1 2 1 2 2 1 1 1 2 1 1
## [6106] 2 2 1 2 1 2 2 2 1 1 2 2 2 1 1 1 2 1 1 1 1 2 2 1 1 1 1 1 2 1 1 2 1 2 2 2 2
## [6143] 2 2 2 2 2 1 2 2 1 2 2 2 2 1 1 1 2 2 2 1 1 1 1 1 1 1 1 2 2 2 2 2 1 2 2 2 1
## [6180] 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 2 1 2 1 1 1 1 1 2 2 1 2 2 2 2 1 1 1 1 2
## [6217] 1 1 1 1 2 1 1 2 1 1 2 2 2 1 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 2 1 2 2 1 1 2
## [6254] 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 2 2 1 1 1 1 1 1 1 1
## [6291] 1 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1 1 2 1 2 1 1 2 1 2 2 1 1 2 2 2 1 1 2 1 2 1
## [6328] 1 2 2 1 1 1 2 2 2 1 1 2 1 1 1 1 2 1 1 2 1 1 1 1 2 2 2 2 1 1 1 1 1 1 2 1 1
## [6365] 2 1 1 1 1 1 1 1 2 1 1 1 2 2 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1
## [6402] 1 1 2 2 1 2 2 1 1 2 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 2 1 1 1 2 2 2 2 1 1
## [6439] 2 1 1 2 1 1 2 1 2 1 1 1 1 2 2 2 1 1 1 1 1 1 2 2 2 1 2 1 2 1 2 1 2 1 1 1 2
## [6476] 1 2 2 1 1 1 1 1 1 1 1 2 1 1 2 1 1 2 1 1 2 2
## 
## Within cluster sum of squares by cluster:
## [1] 5337874 3256954
##  (between_SS / total_SS =  62.6 %)
## 
## Available components:
## 
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
# Cluster identification for
# each observation
kmeans.re$cluster
```

```
##    [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2
##   [38] 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2
##   [75] 2 2 2 2 2 1 2 2 2 2 2 2 1 2 1 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2
##  [112] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2
##  [149] 2 2 2 2 2 2 1 1 1 1 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [186] 2 2 2 1 1 1 2 1 2 2 1 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 2 2 1 2 2
##  [223] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2
##  [260] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [297] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
##  [334] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [371] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2 2 2 2
##  [408] 2 2 2 2 2 2 2 1 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [445] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [482] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2
##  [519] 2 2 2 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [556] 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
##  [593] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [630] 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [667] 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2
##  [704] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [741] 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2
##  [778] 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [815] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [852] 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [889] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [926] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [963] 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1000] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1037] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1074] 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1111] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1148] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1185] 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1222] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1259] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1296] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1333] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1370] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2
## [1407] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1444] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1481] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1518] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1555] 2 2 2 2 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1592] 2 2 2 2 2 2 2 2 1 1 2 1 1 2 1 1 1 1 2 2 2 1 1 1 2 2 1 1 2 2 1 1 1 1 1 1 1
## [1629] 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 2 1 1
## [1666] 1 1 2 2 1 1 1 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1
## [1703] 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 2
## [1740] 2 1 2 2 2 1 1 2 2 1 1 1 1 2 2 1 1 1 1 1 1 1 2 1 1 1 1 2 1 2 1 2 2 1 1 1 2
## [1777] 1 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 1
## [1814] 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 2 2 1 1
## [1851] 1 1 2 1 1 2 2 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 1 1 1 1 1 1 1 1
## [1888] 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 2 1 1 1 1
```

```
## [1925] 1 1 1 1 2 2 1 2 1 2 2 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2
## [1962] 1 1 2 1 1 1 1 2 1 1 1 1 2 2 1 2 1 1 2 1 1 1 1 2 1 1 1 1 1 2 1 2 1 1 2 2 1
## [1999] 2 2 1 2 1 1 1 2 1 1 2 1 1 2 2 1 1 2 1 2 1 1 1 1 1 1 1 1 1 2 1 1 2 1 1 2 2
## [2036] 1 1 1 2 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1 2 1 1 1 1 2
## [2073] 1 2 1 2 1 1 1 1 2 1 1 1 1 1 2 1 1 2 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [2110] 1 1 1 2 1 1 1 1 2 2 1 1 2 2 2 1 2 1 2 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1
## [2147] 2 1 1 1 2 1 1 1 2 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1 1 2
## [2184] 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 2 1 2 1 1 1 1 2 1 1 1 1
## [2221] 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1
## [2258] 1 1 2 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [2295] 1 2 1 1 1 1 1 2 1 1 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 1
## [2332] 2 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 1
## [2369] 1 1 1 1 1 1 2 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 1 1 1 2 1 1 1 1 1 1
## [2406] 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 2 1 2 1 1 1 2 2 1 1 2 2 1 2 1 1 1 1 1 1 1
## [2443] 2 1 1 1 2 1 2 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 2 2 1 1 2 1 2
## [2480] 1 1 1 1 1 2 1 2 1 2 1 1 1 1 1 2 1 2 2 1 1 1 1 1 1 2 2 1 1 1 1 1 1 2 2 2 1
## [2517] 1 2 2 1 1 1 1 1 2 2 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 2 1 1 2 1 1 2 2 1 1 2 1
## [2554] 1 1 1 2 2 1 1 2 1 1 1 2 1 1 2 2 2 2 1 2 1 2 1 1 1 2 2 1 2 2 1 1 1 1 1 1 1
## [2591] 2 1 2 1 1 2 1 1 1 2 1 1 1 1 2 1 2 1 1 2 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2
## [2628] 1 1 1 1 1 1 1 1 2 2 2 2 1 2 2 1 2 2 2 2 2 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 1
## [2665] 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1
## [2702] 1 1 1 1 1 2 2 1 2 2 1 2 2 2 2 1 2 1 2 1 2 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1
## [2739] 2 2 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 2 1 2 1 2 1 1 1 1 2 1 1
## [2776] 1 1 1 2 1 1 2 1 1 1 1 2 1 2 1 1 2 1 1 1 1 1 2 2 2 2 1 2 2 1 1 1 1 1 2 1 1
## [2813] 2 2 1 2 1 1 2 1 1 1 1 2 2 2 2 2 2 1 1 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 2 1
## [2850] 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [2887] 1 2 2 2 1 2 2 2 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 2 1 2 1 1 1
## [2924] 1 1 1 1 2 2 2 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1
## [2961] 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 1 1 1 2 1 1 1 2 2 2 1 2 1 1
## [2998] 1 2 1 1 1 1 2 1 2 2 1 1 2 2 1 1 1 2 1 1 1 2 1 2 2 1 1 1 1 1 2 2 1 2 2 2 1
## [3035] 2 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1
## [3072] 2 1 2 1 1 2 1 1 1 1 1 2 1 1 2 1 1 1 1 2 1 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 1
## [3109] 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 1 2 2 1 2 2 2 2 1
## [3146] 1 1 2 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 2 1 1 2 1 1 1 1 1 1 2 1 2 1 1 1 1 2 1
## [3183] 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 2 1 1 2 2 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1
## [3220] 2 1 1 1 1 1 1 1 2 2 1 2 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 2
## [3257] 1 1 1 1 1 1 1 1 1 2 1 2 1 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [3294] 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 2 1 1 1 2
## [3331] 1 1 1 1 1 2 1 2 2 1 1 1 1 1 1 2 1 2 1 1 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 1 1
## [3368] 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1
## [3405] 1 1 1 1 1 1 1 2 2 2 1 1 1 2 1 1 2 2 1 1 2 1 1 1 1 1 1 1 1 1 1 2 2 1 2 1 2
## [3442] 1 1 2 1 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1
## [3479] 2 1 2 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 2 1 1 1 1 2 1 1 2 1 1 1 2
## [3516] 1 2 1 1 1 1 1 2 2 2 2 1 1 1 1 1 2 1 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [3553] 1 1 1 2 2 1 2 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2
## [3590] 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 2 1 1 1 2 2 1 2 1 1 1 1 1 1 1 1 2
## [3627] 1 1 1 1 1 2 1 1 2 1 1 2 2 1 2 2 2 1 2 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 2 1 1
## [3664] 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1
## [3701] 2 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [3738] 2 2 1 1 1 2 1 1 2 1 2 2 2 1 2 2 1 1 2 2 2 2 1 2 1 2 1 1 1 1 1 1 1 1 1 1 1
## [3775] 1 1 1 1 2 1 1 1 1 2 2 2 1 2 1 1 1 1 1 2 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [3812] 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2
```

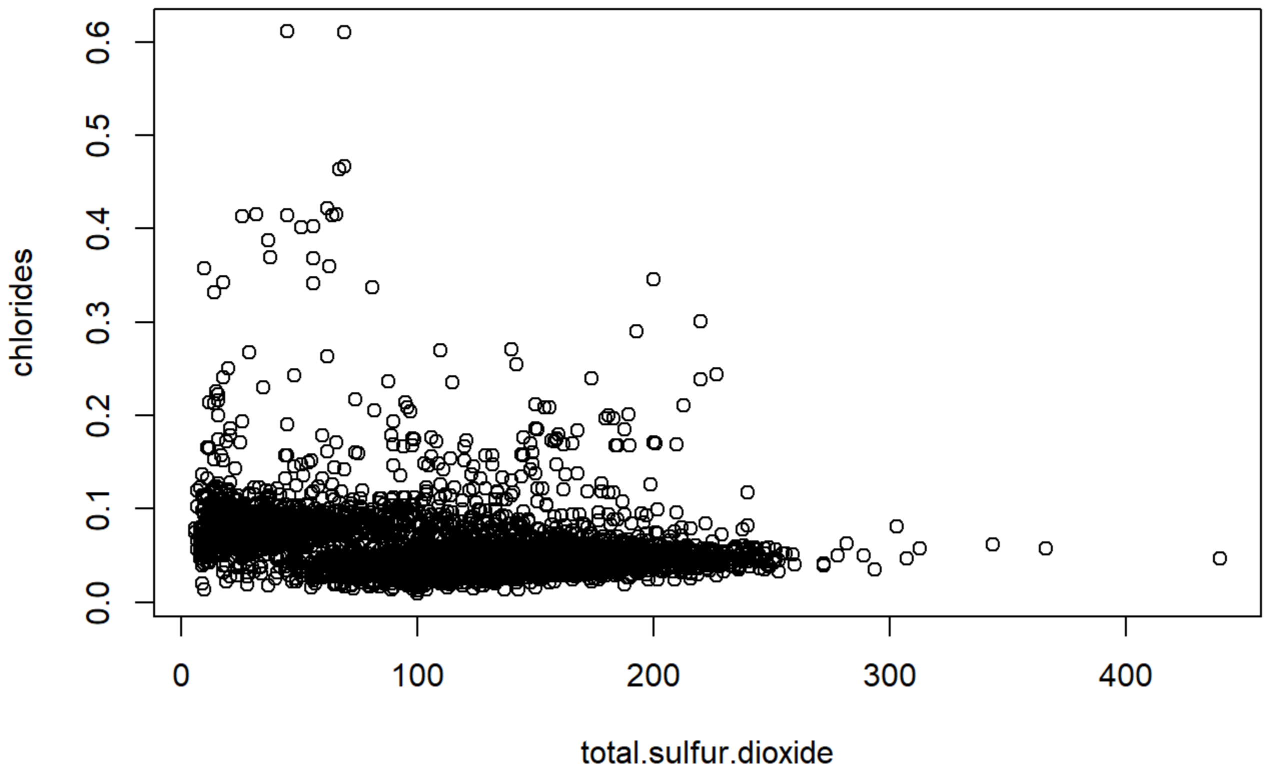
```
## [3849] 1 1 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 2 1 1 1 2 1 1 2 2 1 1 1 1
## [3886] 1 1 1 2 2 1 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1
## [3923] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 1 2 1 1 2 1 1 1 1 1 2 2 1 1 2 1 1
## [3960] 1 2 1 1 1 1 1 1 1 1 2 1 2 2 1 1 1 2 1 1 1 1 2 2 2 1 1 1 2 2 1 1 1 1 1 1 1
## [3997] 2 2 2 2 2 1 1 1 1 2 2 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [4034] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 2 1 1
## [4071] 1 2 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1
## [4108] 1 1 1 1 2 1 1 2 1 1 2 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 2 1 2 1 1 2 1 1 1 2 1
## [4145] 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 2 1 1 1 2 1 1 1 1 2 2 1 1 1 1 1 2 1 1 1
## [4182] 1 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 2 2 1 1 1 2 1 1 2 1 2 1 1 1 1 1
## [4219] 1 1 1 1 1 2 1 1 1 1 1 2 2 1 1 1 2 1 1 2 2 2 1 2 1 1 1 2 1 1 1 1 1 2 1 1 1
## [4256] 1 1 1 2 1 1 1 2 2 1 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1
## [4293] 1 2 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 2 1 1 1 2 1 1 2
## [4330] 1 1 1 2 1 1 1 2 1 2 1 1 1 2 2 2 1 1 1 1 1 1 1 2 2 1 1 2 2 1 1 1 1 1 1 1 2
## [4367] 1 1 1 1 1 1 2 1 1 1 1 2 2 1 1 1 1 1 1 2 1 1 1 1 1 1 2 1 2 2 1 1 1 1 1 2 2
## [4404] 2 1 1 1 1 1 2 1 2 1 2 2 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2 2 2
## [4441] 2 2 2 2 2 2 2 1 1 1 2 1 2 2 1 1 2 1 1 1 2 2 2 2 1 1 1 1 2 1 2 1 2 1 2 1 1
## [4478] 1 2 2 2 1 2 1 2 2 2 2 1 1 1 1 1 2 1 1 1 2 1 2 2 1 2 1 1 1 2 2 2 1 1 2 1 2
## [4515] 2 1 1 2 1 2 1 1 1 1 1 2 1 1 1 1 2 1 1 2 2 2 1 2 2 1 2 1 2 1 1 2 2 1 1 2 2
## [4552] 2 2 2 1 2 2 2 2 1 1 2 1 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 2 1 1 2 1 1 1 2
## [4589] 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1 1 1 2 2 2 2 2 2 1 1 2 2 2 1 2 2 1
## [4626] 1 1 1 1 1 1 2 1 1 2 1 1 2 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 2 1 2 1 1 2 1 1 1
## [4663] 1 1 1 1 1 1 2 1 2 1 1 1 2 1 1 2 1 2 1 2 2 2 2 2 1 2 2 2 1 1 1 2 2 2 1 1 1
## [4700] 1 2 1 1 1 1 1 1 1 1 2 2 1 1 1 1 2 2 1 2 1 1 2 2 1 1 1 2 2 1 1 1 1 1 1 1 2
## [4737] 1 1 1 1 2 1 2 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 2 1 1
## [4774] 1 1 2 1 2 2 2 2 1 2 2 1 2 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 2
## [4811] 1 1 1 2 2 1 2 2 2 2 2 1 1 1 2 1 1 1 1 1 2 2 2 1 1 1 1 1 2 1 1 1 2 2 1 1 1
## [4848] 1 1 1 1 2 2 1 1 1 1 1 1 1 2 1 2 1 1 2 1 1 1 1 2 2 1 1 2 1 1 1 1 1 1 1 1 2
## [4885] 1 1 1 1 1 2 2 1 2 1 1 1 1 1 2 2 2 2 2 1 2 1 1 1 2 1 1 2 2 1 2 2 2 2 1 1 2
## [4922] 2 2 1 1 1 2 1 1 1 1 1 1 2 1 1 1 2 2 1 2 2 1 1 1 1 1 2 2 1 2 2 2 1 1 1 2 2
## [4959] 2 2 2 1 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 2 1 1 1
## [4996] 1 1 1 2 1 1 1 2 2 1 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1
## [5033] 2 2 2 2 1 1 1 2 1 2 2 1 1 1 2 1 1 2 2 1 2 2 2 1 1 1 1 2 1 1 2 1 1 1 1 2 1
## [5070] 1 2 1 2 1 1 2 1 1 2 2 1 2 2 2 1 2 1 2 2 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1
## [5107] 1 1 1 1 2 1 2 2 2 1 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 2 2 1 2 2 1 1 1
## [5144] 1 1 1 1 1 1 1 1 1 1 2 1 2 2 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 2 1
## [5181] 1 2 2 2 2 1 1 1 1 2 1 1 1 2 1 1 1 1 1 2 1 1 1 2 1 1 2 1 1 1 2 1 1 1 2 1 1
## [5218] 1 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 1 2 2 1 2 2 1 1 1 1 1 1 2 2 1 1 1 1 2 1 2
## [5255] 1 1 1 1 1 1 1 2 1 1 1 1 1 2 2 1 2 2 2 1 1 2 1 2 2 1 1 1 1 1 1 1 2 1 1 1 1
## [5292] 2 1 2 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 1 2
## [5329] 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 2 2 2 2 1 1
## [5366] 1 1 1 1 1 1 1 2 1 2 1 2 2 2 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 2 2 1 1 2 2 1
## [5403] 2 2 2 2 2 2 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 2 1 2 1 2 1 1 1 1 1 1 2 2 1 1
## [5440] 2 1 1 2 2 1 2 1 2 2 2 1 2 1 1 1 2 2 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 2 1
## [5477] 2 1 2 1 1 1 1 1 2 1 2 1 1 2 1 1 2 1 2 1 1 1 1 2 2 2 2 2 2 2 2 1 1 1 1 2 2
## [5514] 1 2 1 1 1 2 1 1 2 2 2 2 2 2 1 1 2 2 1 2 2 1 2 1 2 1 1 1 2 1 1 1 1 1 2 1 1
## [5551] 1 2 1 1 2 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 1 2 2 1
## [5588] 1 2 1 2 2 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 2 1 1 1 1 1 1 2 1 1 1 2 2 2 1 1 1
## [5625] 2 1 1 2 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 1 2 1 1 1 2 1 1 1 2 2 2
## [5662] 1 1 2 1 1 1 1 1 1 2 1 2 2 1 1 2 2 2 1 1 1 1 2 2 2 2 2 1 1 2 1 2 2 2 1 2 1
## [5699] 1 2 1 1 2 2 1 1 1 2 1 1 2 2 2 2 2 1 1 1 1 1 1 1 2 2 1 1 1 1 2 1 1 1 1 2 1
## [5736] 1 1 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 1 1 2 1
```

```
## [5773] 2 1 1 1 1 1 1 1 2 1 2 2 1 2 1 1 2 2 1 2 2 2 2 2 1 1 2 2 2 1 1 1 2 1 1 2 2
## [5810] 1 1 1 2 1 1 1 1 1 1 2 1 1 2 2 1 1 2 1 1 1 2 2 1 1 1 1 2 1 2 1 1 1 2 1 1 2
## [5847] 1 1 1 2 2 2 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2
## [5884] 2 2 2 1 2 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 2 1 1 2 1 2 1 1 2 1 1 1 1
## [5921] 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 2 2 2 1 1 1 2 1 1 1 1 1 1 1
## [5958] 1 1 1 1 1 1 1 1 1 1 2 1 1 2 1 2 1 1 1 2 1 1 1 1 1 2 1 1 1 2 2 1 1 1 1 1 1
## [5995] 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2
## [6032] 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2
## [6069] 2 2 2 2 1 1 1 2 1 1 1 1 1 1 2 1 2 2 2 1 1 1 2 2 1 2 1 2 1 2 2 1 1 1 2 1 1
## [6106] 2 2 1 2 1 2 2 2 1 1 2 2 2 1 1 1 2 1 1 1 1 2 2 1 1 1 1 1 2 1 1 2 1 2 2 2 2
## [6143] 2 2 2 2 2 1 2 2 1 2 2 2 2 1 1 1 2 2 2 1 1 1 1 1 1 1 1 2 2 2 2 2 1 2 2 2 1
## [6180] 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 2 1 2 1 1 1 1 1 2 2 1 2 2 2 2 1 1 1 1 2
## [6217] 1 1 1 1 2 1 1 2 1 1 2 2 2 1 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 2 1 2 2 1 1 2
## [6254] 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 2 2 1 1 1 1 1 1 1 1
## [6291] 1 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1 1 2 1 2 1 1 2 1 2 2 1 1 2 2 2 1 1 2 1 2 1
## [6328] 1 2 2 1 1 1 2 2 2 1 1 2 1 1 1 1 2 1 1 2 1 1 1 1 2 2 2 2 1 1 1 1 1 1 2 1 1
## [6365] 2 1 1 1 1 1 1 1 2 1 1 1 2 2 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1
## [6402] 1 1 2 2 1 2 2 1 1 2 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 2 1 1 1 2 2 2 2 1 1
## [6439] 2 1 1 2 1 1 2 1 2 1 1 1 1 2 2 2 1 1 1 1 1 1 2 2 2 1 2 1 2 1 2 1 2 1 1 1 2
## [6476] 1 2 2 1 1 1 1 1 1 1 1 2 1 1 2 1 1 2 1 1 2 2
```

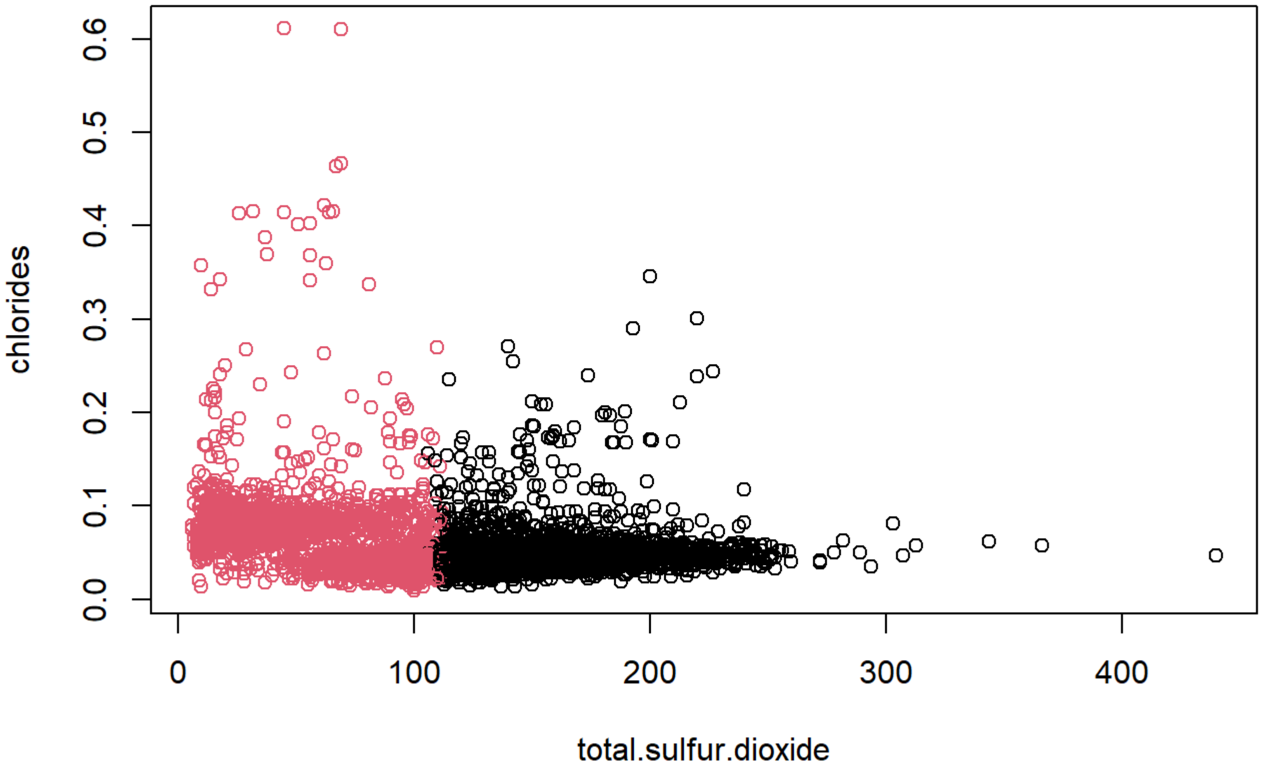
```r
# Confusion Matrix
cm <- table(wine$color, kmeans.re$cluster)
cm
```

```
## 
##           1    2
##   red     85 1514
##   white 3604 1294
```

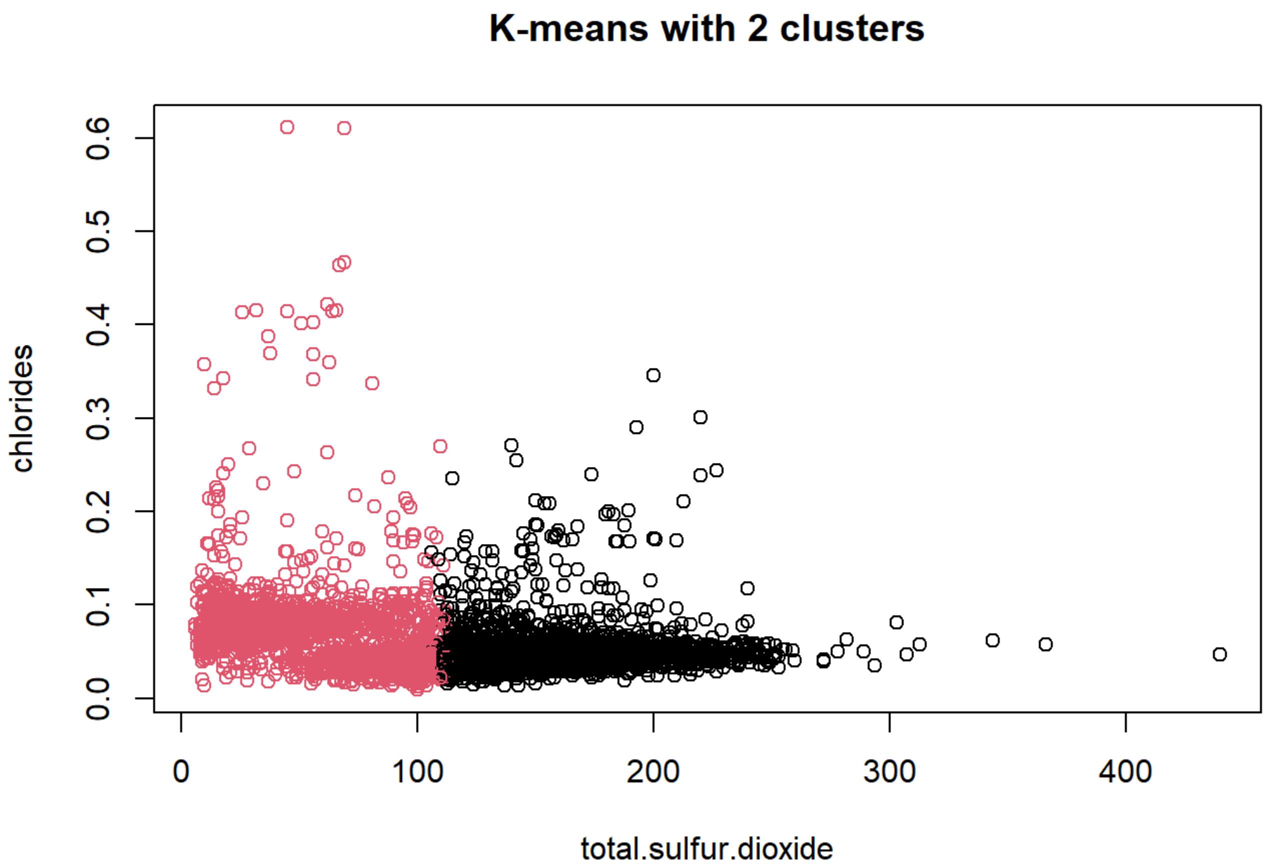
```r
# Model Evaluation and visualization
plot(wine_1[c("total.sulfur.dioxide", "chlorides")])
```




```
plot(wine_1[c("total.sulfur.dioxide", "chlorides")],
     col = kmeans.re$cluster)
```




```
plot(wine_1[c("total.sulfur.dioxide", "chlorides")],
     col = kmeans.re$cluster,
     main = "K-means with 2 clusters")
```



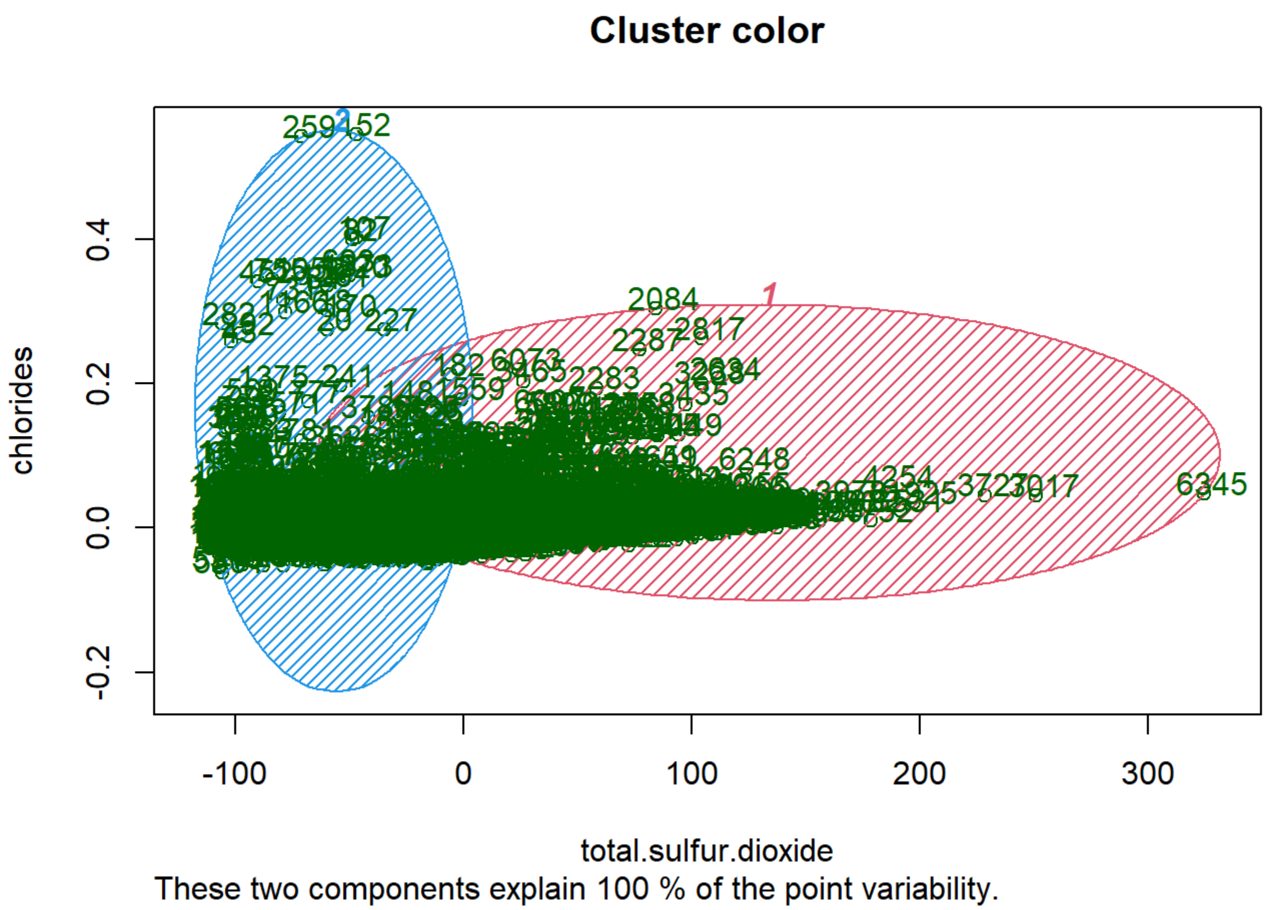

```
## Plotiing cluster centers
kmeans.re$centers
```

```
##   fixed.acidity volatile.acidity citric.acid residual.sugar  chlorides
## 1      6.904812        0.2871659   0.3397642       7.244809 0.04859257
## 2      7.623219        0.4086378   0.2908725       3.076425 0.06580983
##   free.sulfur.dioxide total.sulfur.dioxide   density       pH sulphates
## 1            39.75590            155.69246 0.9947903 3.190808 0.4999485
## 2            18.39868             63.26318 0.9945736 3.254882 0.5724145
##    alcohol  quality
## 1 10.25932 5.824343
## 2 10.79722 5.810541
```

```
kmeans.re$centers[, c("total.sulfur.dioxide", "chlorides")]
```

```
##   total.sulfur.dioxide  chlorides
## 1            155.69246 0.04859257
## 2             63.26318 0.06580983
```

```
## Visualizing clusters
y_kmeans <- kmeans.re$cluster
clusplot(wine_1[, c("total.sulfur.dioxide", "chlorides")],
         y_kmeans,
         lines = 0,
         shade = TRUE,
         color = TRUE,
         labels = 2,
         plotchar = FALSE,
         span = TRUE,
         main = paste("Cluster color"),
         xlab = 'total.sulfur.dioxide',
         ylab = 'chlorides')
```



The sum of squared on the Kmeans clustering is 62.6%.

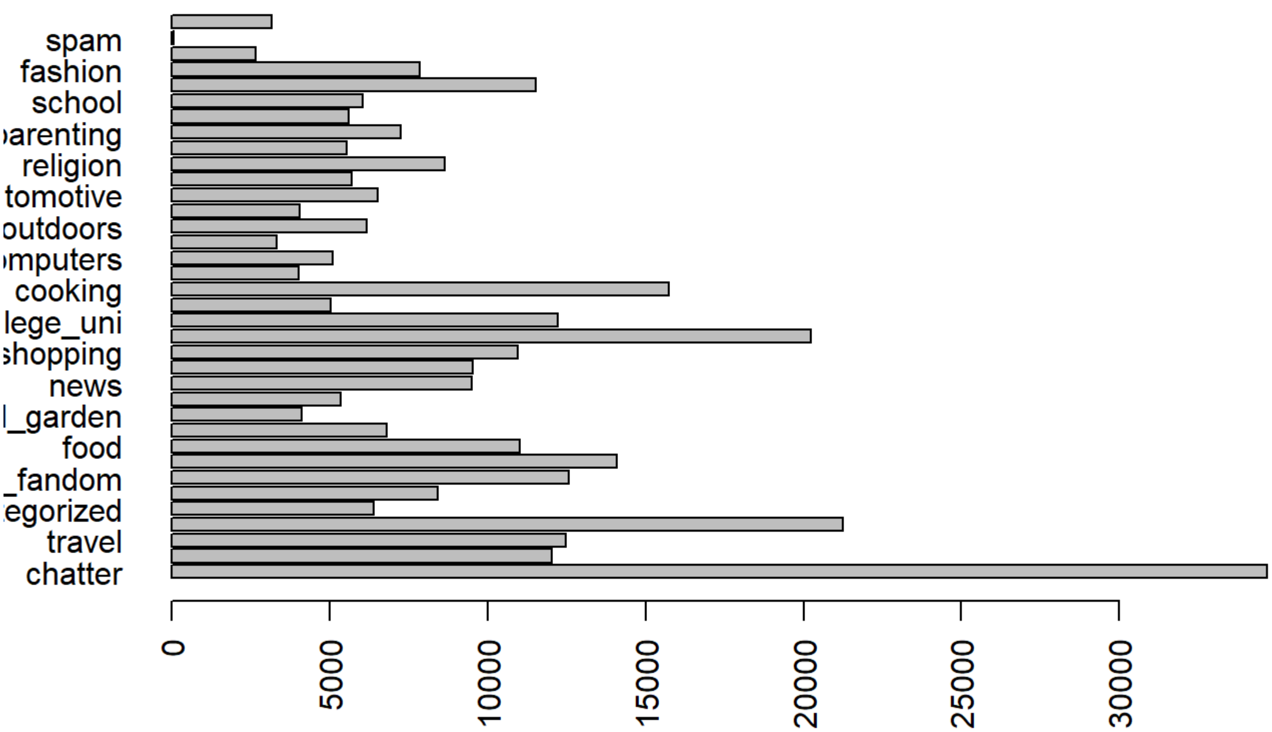
Market segmentation

```
library(readr)
library(ggplot2)
library(dplyr)
mkt <- read_csv("social_marketing.csv")
```

```
## New names:
## * `` -> ...1
```

```
## Rows: 7882 Columns: 37
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): ...1
## dbl (36): chatter, current_events, travel, photo_sharing, uncategorized, tv_...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# sumdata=data.frame(value=apply(mkt,2,sum))
# sumdata$key=rownames(sumdata)
# ggplot(data=sumdata, aes(x=key, y=value, fill=key)) +
# geom_bar(colour="black", stat="identity")


barplot(colSums(mkt[,2:37]),las=2, horiz = TRUE)
```




```
colSums(mkt[,2:37])
```

```
##          chatter   current_events           travel    photo_sharing
##            34671            12030            12493            21256
##    uncategorized          tv_film    sports_fandom         politics
##             6408             8436            12564            14098
##             food           family  home_and_garden            music
##            11015             6809             4104             5354
##             news    online_gaming         shopping health_nutrition
##             9502             9528            10951            20235
##      college_uni   sports_playing          cooking              eco
##            12213             5038            15750             4038
##        computers         business         outdoors           crafts
##             5116             3336             6169             4066
##       automotive              art         religion           beauty
##             6541             5713             8634             5558
##        parenting           dating           school personal_fitness
##             7262             5603             6051            11524
##          fashion   small_business             spam            adult
##             7855             2651               51             3179
```

As we can see from this bar chart which depicts the differing amounts of tweets per category, there are certain categorys with a larger amount of people interacting with that genre. Most notably, 'chatter' is the category with the largest number of tweets, but it is a very general category and therefore may have a disproportionate amount of tweets. Besides chatter, lots of people are tweeting about photo sharing, health and nutrition, politics, current events, and travel. The genres least discussed from this dataset are Eco, dating, adult, and crafts. What NutrientH2O can take away from this is how to adjust/structure their advertisements in a way which maximizes the engagement of users.

The Reuters corpus

```
# raw.files <- data_frame(filename = list.files('/Users/aakashtalathi/Desktop/ReutersC50/C50trai
n'))
#
#
# raw.file.paths <- raw.files  %>%
# mutate(filepath = paste0("/Users/aakashtalathi/Desktop/ReutersC50/C50train/", filename))
#
# for (x in raw.file.paths)
# {
#   q = x
#   raw.files.x <- data_frame(filenamex = list.files(x))
# }
# View(raw.files.x)
```

We had trouble loading in the raw files for this question so I wrote a brief analysis of the process we would have gone through had we successfully loaded them in. Rather then looking at individual texts of an author we believed it to be more benifical to group together each text as one large paragraph which is representative of the words the author would use. As we did not have a way to easily measure sentiment for this dataset we decided to calulate the IDF scores for each unique word used by the author, and this data would be what we would cluster on. This allows us to get a representation of how unique a authors vocabulary is as compared to their peers. Based on the results of clustering, we may be able to see groups of authors with similar vocabulary levels/usuage.

The question that will be answered in this analysis is what authors have similar unique vocabulary levels to each other? The approach that will be used is calculating the average of the TF-IDF of all words unique words an author uses. For example, if a an author uses the word “it” multiple times it will only be counted ones in terms of the TF-IDF. Essentially, we are creating a list of every unique word an author has ever used in the given texts and running the TF-IDF over every list of authors. We then calculate the average TF-IDF score per author. Using this technique we can hope to see a scatterplot of each author, grouped together with other authors with similar unique vocabulary words. If an author tends to use words that other authors do not, we will see them clustered together and vice versa. We believe that two or three clusters is the optimal level for this problem as there should not be micro clusters within the vocabulary levels of the authors.

In conclusion this analysis is important because it presents an mathematical grouping of authors who may be using more unique words, an indicator that they are writing about topics others aren’’t or using a wider range of vocabulary which may be indicative of certain writing features.

Association rule mining

```
# install.packages("arules")
# install.packages("arulesViz")

library(tidyverse)
library(igraph)
```

```
## Warning: package 'igraph' was built under R version 4.0.5
```

```
## 
## Attaching package: 'igraph'
```

```
## The following object is masked from 'package:gtools':
## 
##     permute
```

```
## The following object is masked from 'package:mosaic':
## 
##     compare
```

```
## The following objects are masked from 'package:dplyr':
## 
##     as_data_frame, groups, union
```

```
## The following objects are masked from 'package:purrr':
## 
##     compose, simplify
```

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

```
## The following object is masked from 'package:tibble':
##
##     as_data_frame
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
##
##     union
```

```
library(arules)  # has a big ecosystem of packages built around it
```

```
## Warning: package 'arules' was built under R version 4.0.5
```

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

```
## 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
```

```
library(arulesViz)
```

```
## Warning: package 'arulesViz' was built under R version 4.0.5
```

```
data("Groceries")

rules <- apriori(Groceries, parameter=list(support=.02, confidence=.15))
```

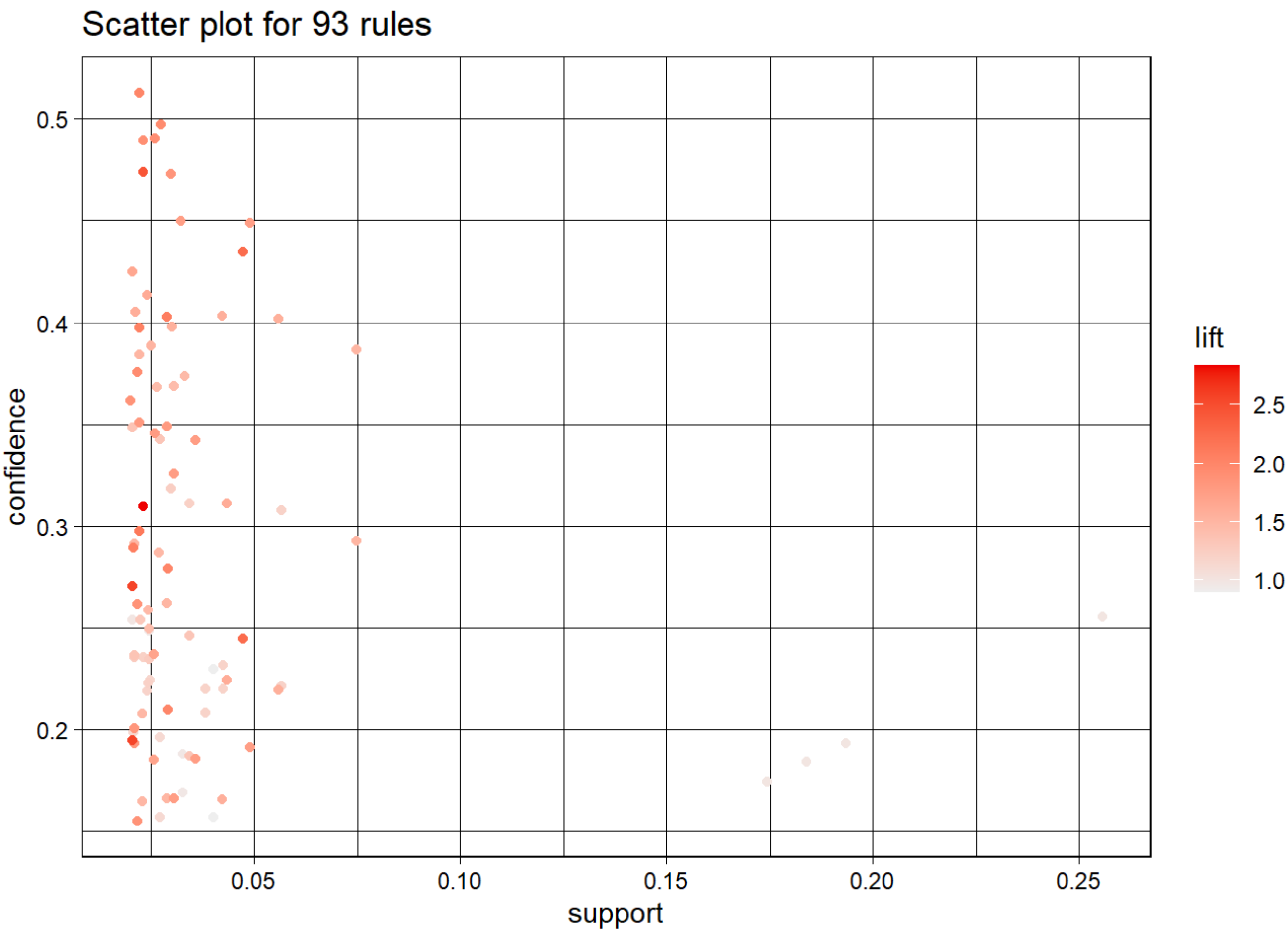
```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.15    0.1    1 none FALSE            TRUE       5    0.02      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 196
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.01s].
## sorting and recoding items ... [59 item(s)] done [0.00s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [93 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
arules::inspect(rules[1:20])
```

```
##      lhs                          rhs                support    confidence
## [1]  {}                        => {soda}             0.17437722 0.1743772
## [2]  {}                        => {rolls/buns}       0.18393493 0.1839349
## [3]  {}                        => {other vegetables} 0.19349263 0.1934926
## [4]  {}                        => {whole milk}       0.25551601 0.2555160
## [5]  {frozen vegetables}       => {whole milk}       0.02043721 0.4249471
## [6]  {beef}                    => {whole milk}       0.02125064 0.4050388
## [7]  {curd}                    => {whole milk}       0.02613116 0.4904580
## [8]  {pork}                    => {other vegetables} 0.02165735 0.3756614
## [9]  {pork}                    => {whole milk}       0.02216573 0.3844797
## [10] {frankfurter}             => {whole milk}       0.02053889 0.3482759
## [11] {bottled beer}            => {whole milk}       0.02043721 0.2537879
## [12] {brown bread}             => {whole milk}       0.02521607 0.3887147
## [13] {margarine}               => {whole milk}       0.02419929 0.4131944
## [14] {butter}                  => {other vegetables} 0.02003050 0.3614679
## [15] {butter}                  => {whole milk}       0.02755465 0.4972477
## [16] {newspapers}              => {whole milk}       0.02735130 0.3426752
## [17] {domestic eggs}           => {other vegetables} 0.02226741 0.3509615
## [18] {domestic eggs}           => {whole milk}       0.02999492 0.4727564
## [19] {fruit/vegetable juice} => {other vegetables} 0.02104728 0.2911392
## [20] {fruit/vegetable juice} => {whole milk}       0.02663955 0.3684951
##      coverage   lift      count
## [1]  1.00000000 1.0000000 1715
## [2]  1.00000000 1.0000000 1809
## [3]  1.00000000 1.0000000 1903
## [4]  1.00000000 1.0000000 2513
## [5]  0.04809354 1.6630940  201
## [6]  0.05246568 1.5851795  209
## [7]  0.05327911 1.9194805  257
## [8]  0.05765125 1.9414764  213
## [9]  0.05765125 1.5047187  218
## [10] 0.05897306 1.3630295  202
## [11] 0.08052872 0.9932367  201
## [12] 0.06487036 1.5212930  248
## [13] 0.05856634 1.6170980  238
## [14] 0.05541434 1.8681223  197
## [15] 0.05541434 1.9460530  271
## [16] 0.07981698 1.3411103  269
## [17] 0.06344687 1.8138238  219
## [18] 0.06344687 1.8502027  295
## [19] 0.07229283 1.5046529  207
## [20] 0.07229283 1.4421604  262
```

What our association rules tell us is that people buy soda, rolls/buns, other vegetables, and whole milk the most. This is why we can assume with nothing else in their basket, they would get these items. Whole milk seems to be something people get frequently regardless of other items as many association rules are recommending whole milk. Some interesting rules to be seen are pork and other vegetables as typically you would combine both in a dish. However, for other proteins such as beef, the first recommendation remains as whole milk. Butter can be an association to begetables as people tend to cook vegetables in butter.

```
plot(rules)
```



Overall, high lift gives us low support (although everything is pretty low on support)