STA 380 Homework 1

Summer Wang, Yawen Ye, Wenduo Wang

August 1, 2016

Probability practice

Part A.

Here's a question a friend of mine was asked when he interviewed at Google.

Visitors to your website are asked to answer a single survey question before they get access to the content on the page. Among all of the users, there are two categories: Random Clicker (RC), and Truthful Clicker (TC). There are two possible answers to the survey: yes and no. Random clickers would click either one with equal probability. You are also giving the information that the expected fraction of random clickers is 0.3.

After a trial period, you get the following survey results: 65% said Yes and 35% said No.

What fraction of people who are truthful clickers answered yes?

Given the information, we can make a set of simultaneous equations to solve the problem, which in our case is to get $P(Yes|TC)$ (actually one of them is sufficient, but let's solve both and see if the results are the same)

$$\begin{cases} P(TC) * P(Yes|TC) + P(RC) * P(Yes|RC) = 65\% \\ P(TC) * P(No|TC) + P(RC) * P(No|RC) = 35\% \end{cases}$$

Which is equivalent to:

$$\begin{cases} 0.7 * P(Yes|TC) + 0.3 * 0.5 = 65\% \\ 0.7 * (1 - P(Yes|TC)) + 0.3 * 0.5 = 35\% \end{cases}$$

Therefore $P(Yes|TC) = \frac{5}{7}$

Part B.

Imagine a medical test for a disease with the following two attributes: - The *sensitivity* is about 0.993. That is, if someone has the disease, there is a probability of 0.993 that they will test positive.
- The *specificity* is about 0.9999. This means that if someone doesn't have the disease, there is probability of 0.9999 that they will test negative.

In the general population, incidence of the disease is reasonably rare: about 0.0025% of all people have it (or 0.000025 as a decimal probability).

Suppose someone tests positive. What is the probability that they have the disease? In light of this calculation, do you envision any problems in implementing a universal testing policy for the disease?

The problem here is to solve the probability $P(have\ disease|positive\ test)$, which can be done using Bayes' theorem $P(A|B) = \frac{P(B|A)*P(A)}{\sum_i P(B|A_i)*P(A_i)}$.

$$P(have\, disease|positive\, test) = \frac{P(positive\ test|have\ disease)*P(have\ disease)}{P(positive\ test|have\ disease)*P(have\ disease)+P(positive\ test|no\ disease)*P(no\ disease)}$$

Which given the above information can be written as

$P(have\ disease|positive\ test) = \frac{0.993*0.000025}{0.993*0.000025+0.0001*0.999975} = 19.88824\%$

Exploratory analysis: green buildings

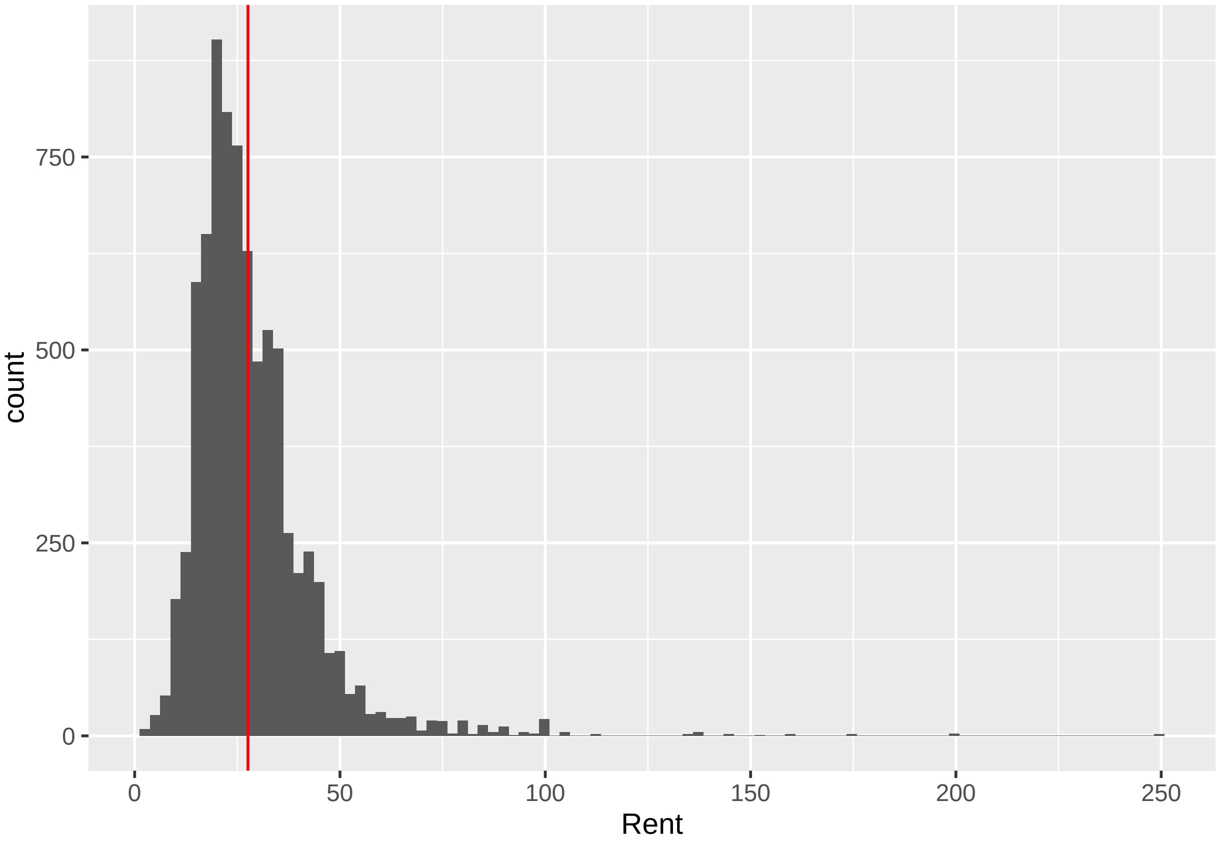
```
library(dplyr)
library(ggplot2)
library(gridExtra)
# library()
```

Before making an assertion on the investment return on getting a green certification, it is necessary to understand the provided data and see how it can be used for our purpose.

The `greenbuildings` dataset includes the records of rental buildings across US, including physical and environmental attributes, and rent rates. It provides a benchmark to forecast the potential benefit of obtaining a green certification in our question.

Let's explore the dataset a little bit and see where the analyst's forecast of 27.6 USD/sqft per year sits in the rent spectrum.

```
green <- read.csv("data/greenbuildings.csv", header=TRUE)
ggplot(data=green, aes(x=Rent)) + geom_histogram(bins=100) + geom_vline(xintercept=27.6, color="red")
```



```
logistic_fit <- glm(Rent~., data=green, family="gaussian")
confint(logistic_fit)
```

```
>>> Waiting for profiling to be done...
>>>                               2.5 %         97.5 %
>>> (Intercept)       -1.031011e+01 -6.319176e+00
>>> CS_PropertyID     -1.270033e-08  6.044450e-07
>>> cluster            1.966512e-04  1.309730e-03
>>> size               5.455665e-06  8.027414e-06
>>> empl_gr            3.118021e-02  9.781244e-02
>>> leasing_rate      -9.961513e-04  1.990324e-02
>>> stories           -6.640729e-02 -3.024430e-03
>>> age               -2.173899e-02 -3.248869e-03
>>> renovated         -6.492764e-01  3.643434e-01
>>> class_a            2.014465e+00  3.730122e+00
>>> class_b            5.147248e-01  1.858202e+00
>>> LEED              -5.143667e+00  8.897289e+00
>>> Energystar        -7.695159e+00  7.269774e+00
>>> green_rating      -6.826477e+00  8.220367e+00
>>> net               -3.721208e+00 -1.396973e+00
>>> amenities          1.766230e-01  1.163947e+00
>>> cd_total_07       -4.118391e-04  1.621855e-04
>>> hd_total07         3.595592e-04  7.112730e-04
>>> total_dd_07                   NA            NA
>>> Precipitation      1.671143e-02  7.988095e-02
>>> Gas_Costs         -5.096047e+02 -2.021921e+02
>>> Electricity_Costs  1.396967e+02  2.374278e+02
>>> cluster_rent       9.805372e-01  1.036251e+00
```

The histogram shows a reasonable positioning of the rent forecast by the analyst, because 27.6 seems like a "default" value even without any further information. But that is utilizing the information in our case. So let's examine how to make our forecast more accurate, and evaluate the analyst's report.

From the coefficient intervals of the logistic regression model, it is seen that 10 out of the 23 attributes are statistically insignificant. Among the other 13 variables, only a part are relevant in this case which are known for the new project for comparison/prediction purpose.

```
+ size
+ empl_gr(to be inferred)
+ age
+ LEED
+ amenities
+ hd_total07(to be inferred)
+ Precipitation
+ Gas_Costs
+ Electricity_Costs
+ cluster_rent
```

Therefore we only select these variables and again fit a logistic regression model.

```
green_2 <- select(green, Rent, size, empl_gr, age, LEED, amenities, hd_total07, Precipitation, Gas_Costs
logistic_fit_2 <- glm(Rent~., data=green_2, family="gaussian")
cat("Coefficient of LEED =", logistic_fit_2$coefficients["LEED"])
print("")
confint(logistic_fit_2)
```

```
>>> Waiting for profiling to be done...

>>> Coefficient of LEED = 2.863394[1] ""
>>>                         2.5 %       97.5 %
>>> (Intercept)       -7.396322e+00 -3.935383e+00
>>> size               5.579480e-06  7.174320e-06
>>> empl_gr            2.320881e-02  8.377537e-02
>>> age               -3.525165e-02 -2.078762e-02
>>> LEED               3.217144e-01  5.405074e+00
>>> amenities          5.606118e-01  1.500518e+00
>>> hd_total07         3.963291e-04  7.131531e-04
>>> Precipitation      3.129444e-03  6.525951e-02
>>> Gas_Costs         -4.924881e+02 -2.171006e+02
>>> Electricity_Costs  1.433174e+02  2.385684e+02
>>> cluster_rent       1.004355e+00  1.057112e+00
```

Now it is clear all the variables in the new model are statistically significantly correlated with `Rent`, and the coefficient confidence interval of each variable is printed out. In our case, we are predicting how much extra profit can be made by obtaining the green certification, that is to say *LEED = 1*. Looking at the coefficient of `LEED` and its confidence interval, it is safe to assert that having the green certification statistically increases the unit rent by $2.86, which turns out higher than the analyst's reasoning, therefore her conclusion is further supported. But let's go further and include some specific data to predict what the rent should be in our case.

Below are some data available for the new building, some are stated in the text, while some are inferred using Austin local statistics.

```
+ size = 250000
+ empl_gr = 4% # Bureau of Labor Statistics
+ age = 0 # growing year by year.
+ LEED = 1
+ amenities = 1
+ hd_total07 = 365*0.75 =  # guess
+ Precipitation = 34.25 # http://www.usclimatedata.com/
+ Gas_Costs = 0.03 # take the max to assess influence
+ Electricity_Costs = 0.06 # take the max to assess influence
+ cluster_rent = 47 or 36 or 27 # http://www.austintenantadvisors.com/austin-office-lease-rates/
```

Pass the data into a new dataframe and predict `Rent` on it.

```r
austin_green <- data.frame(size=rep(250000, 3), empl_gr=rep(0.04, 3), age=rep(0, 3), LEED=rep(1 ,3), ame

austin_ngreen <- austin_green
austin_ngreen$LEED <- 0

rent_green <- predict(logistic_fit_2, newdata=austin_green)
rent_ngreen <- predict(logistic_fit_2, newdata=austin_ngreen)
rent_prediction <- data.frame(Green=rent_green, Plain=rent_ngreen)
row.names(rent_prediction) <- c("Class A", "Class B", "Class C")
print(rent_prediction)
```

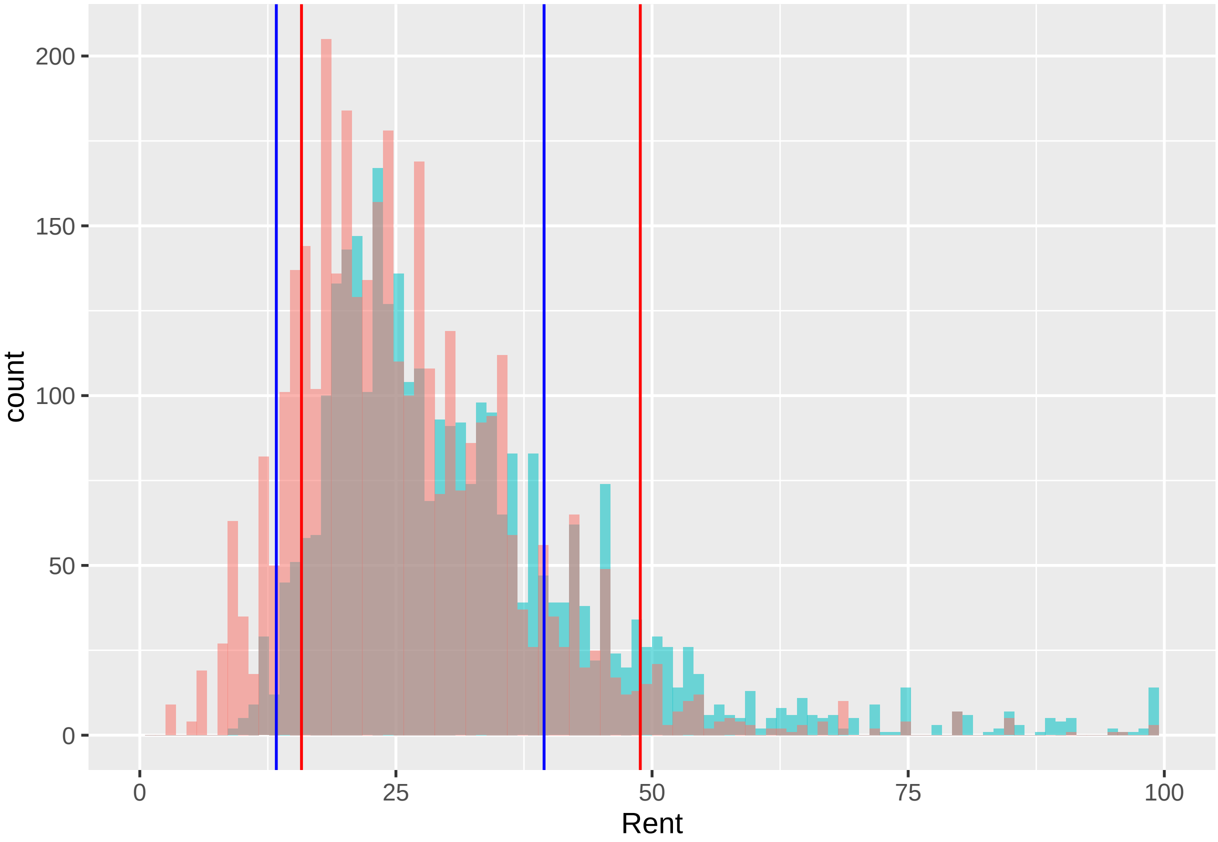
```
>>>            Green    Plain
>>> Class A 50.40484 47.54145
>>> Class B 39.06678 36.20338
>>> Class C 29.79018 26.92678
```

Here we can see `Rent` depends on the class of the building, and the variation is considerable, yet whichever class it is in our case, the predicted `Rent` should be higher than the analyst's assertion, and thus the investment is even more profitable given the data.

However, suppose the above local data is not available, the analyst's conclusion could have been invalidated. For example, since we know the actual rent highly depends on the class of the building, let's assess this effect on `Rent`.

```r
mean_a <- mean(green[green$class_a==1,]$Rent)
mean_b <- mean(green[green$class_b==1,]$Rent)
```

```
sd_a <- sd(green[green$class_a==1,]$Rent)
sd_b <- sd(green[green$class_b==1,]$Rent)

ggplot(data=green, aes(x=Rent)) +
    geom_histogram(data=green[green$class_a==1,], bins=100, aes(fill="red", alpha=0.4)) +
    geom_histogram(data=green[green$class_b==1,], bins=100, aes(fill="blue", alpha=0.4)) +
    geom_vline(xintercept=c(mean_a-sd_a, mean_a+sd_a), color="red") +
    geom_vline(xintercept=c(mean_b-sd_b, mean_b+sd_b), color="blue") +
    scale_x_continuous(limits=c(0, 100)) +
    theme(legend.position="none")
```



In the above graph, Class A buildings are rendered in red and Class B in blue, and for both classes, the mean rent and standard deviation is calculated and marked on the graph. The red vertical lines are $mean(Rent) \pm sd(Rent)$ for Class A, and the blue lines are that for Class B. Without the knowledge of local real estate market, the analyst could have been too optimistic about the project profitability, if the building were to be offered at $20/sqft a year.

Bootstrapping

Load necessary libraries.

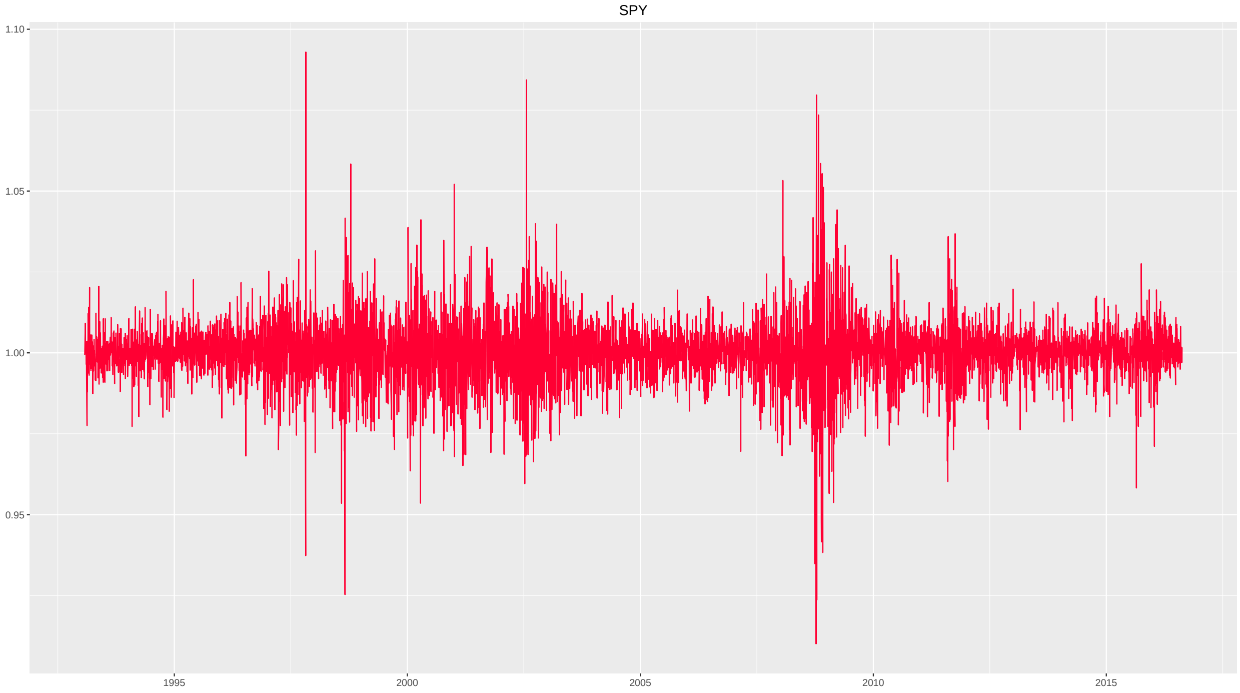
```
library(fImport)
library(ggplot2)
library(reshape2)
library(dplyr)
library(gridExtra)
```

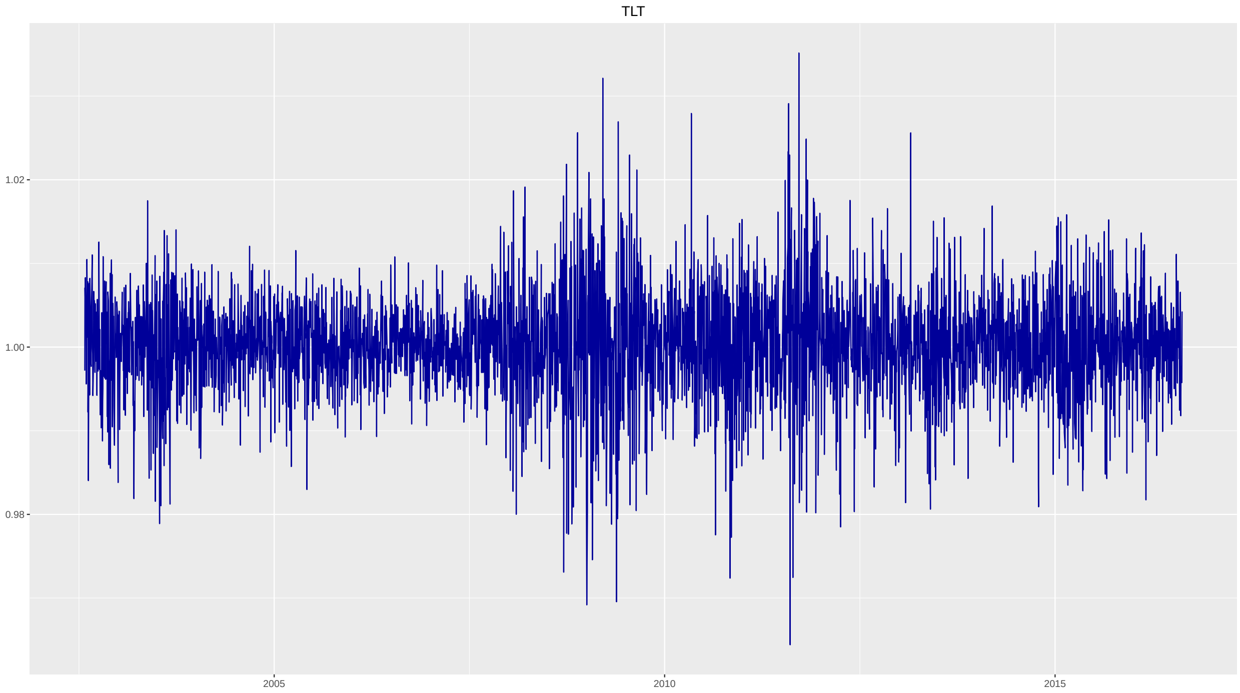
Define a helper function to download instrument indices, and transform the `timeSeries` object to dataframe.

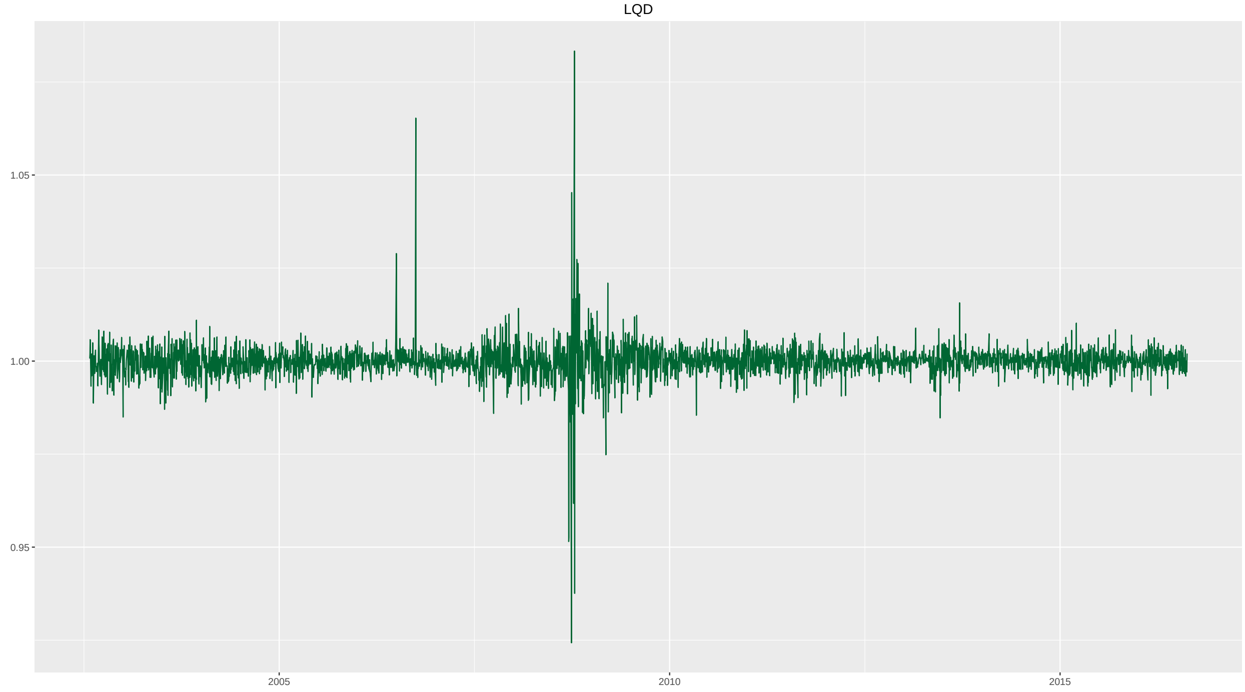
```
import_data <- function(ticker){
    df <- yahooImport(ticker, from=1990-01-01, to=Sys.timeDate())@data
    df$Date <- as.Date(row.names(df))
    df$Return <- df$Close/df$Open
    df <- as.data.frame(df)
    df <- select(df, Date, Return)
    return(df)
}

df_SPY <- import_data("SPY")
df_TLT <- import_data("TLT")
df_LQD <- import_data("LQD")
df_EEM <- import_data("EEM")
df_VNQ <- import_data("VNQ")
```

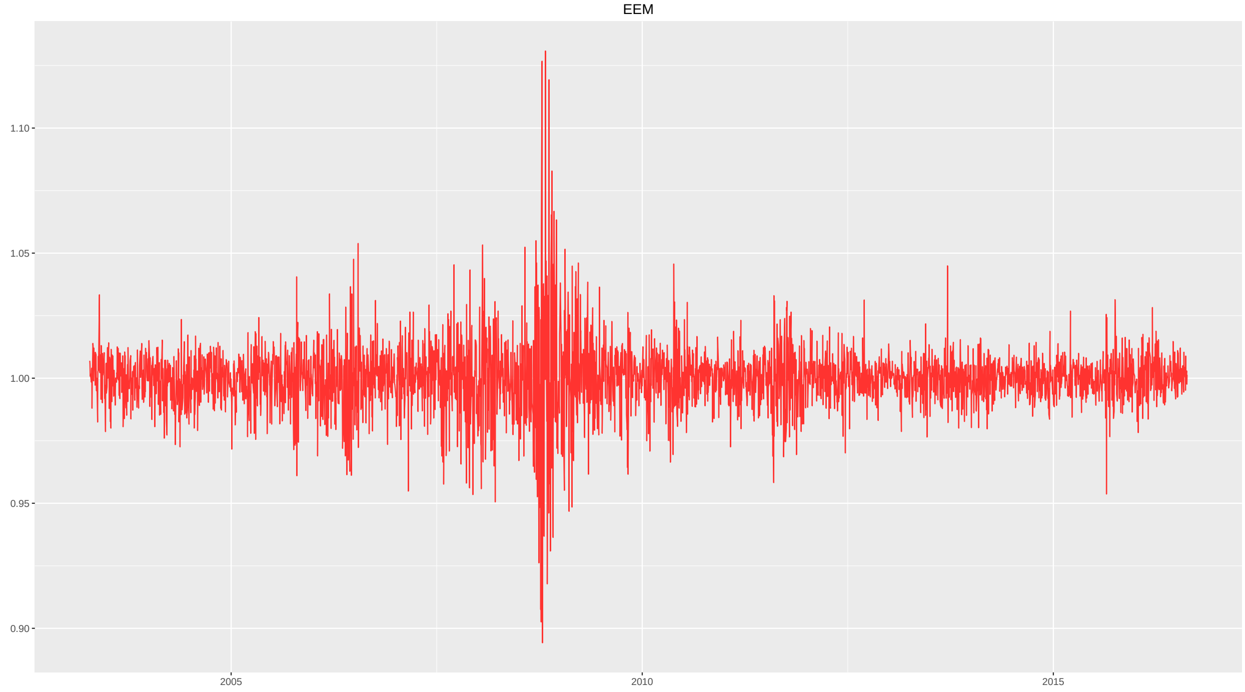
Let's plot the historical daily return curves of the 5 instruments.

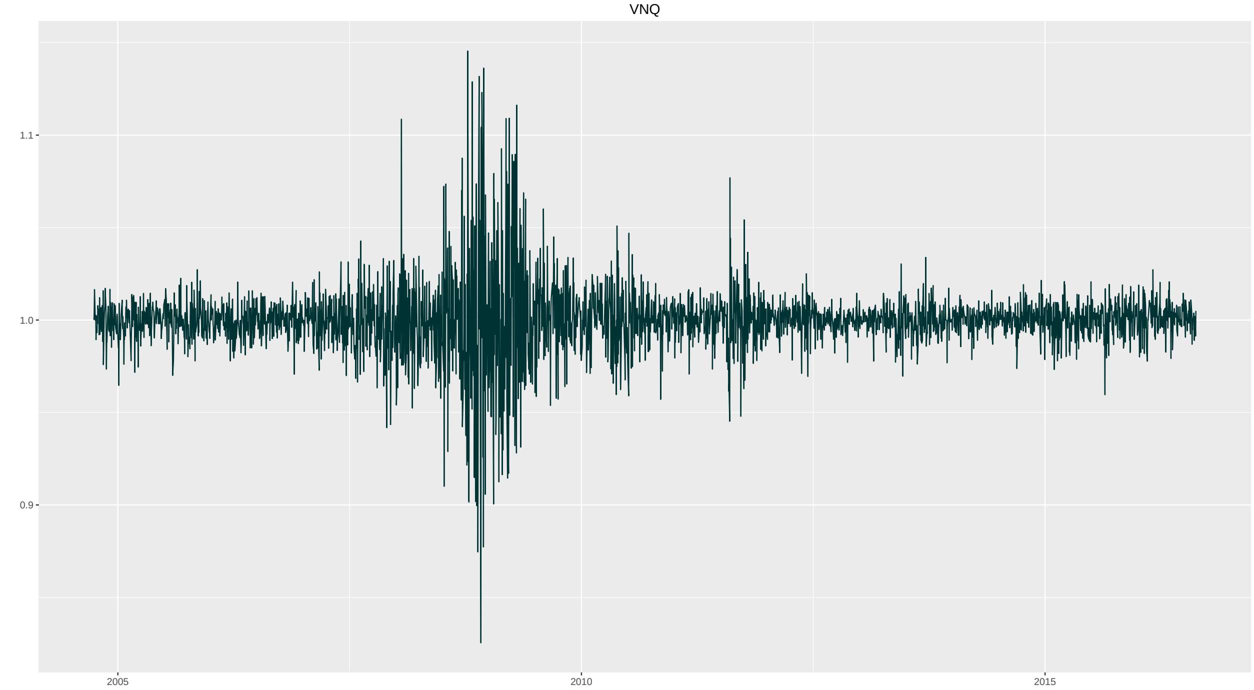
```
p_spy <- ggplot(data=df_SPY, aes(x=Date, y=Return))
p_spy <- p_spy + geom_line(colour="#FF0033") + guides(colour=FALSE) +
    labs(title="SPY", x=" ", y=" ")

p_tlt <- ggplot(data=df_TLT, aes(x=Date, y=Return))
p_tlt <- p_tlt + geom_line(colour="#000099") + guides(colour=FALSE) +
    labs(title="TLT", x=" ", y=" ")

p_lqd <- ggplot(data=df_LQD, aes(x=Date, y=Return))
p_lqd <- p_lqd + geom_line(colour="#006633") + guides(colour=FALSE) +
    labs(title="LQD", x=" ", y=" ")

p_eem <- ggplot(data=df_EEM, aes(x=Date, y=Return))
p_eem <- p_eem + geom_line(colour="#FF3330") + guides(colour=FALSE) +
    labs(title="EEM", x=" ", y=" ")

p_vnq <- ggplot(data=df_VNQ, aes(x=Date, y=Return))
p_vnq <- p_vnq + geom_line(colour="#003333") + guides(colour=FALSE) +
    labs(title="VNQ", x=" ", y=" ")
```


SPY
1.10
1.05
1.00
0.95
1995
2000
2005
2010
2015


TLT
1.02
1.00
0.98
2005
2010
2015

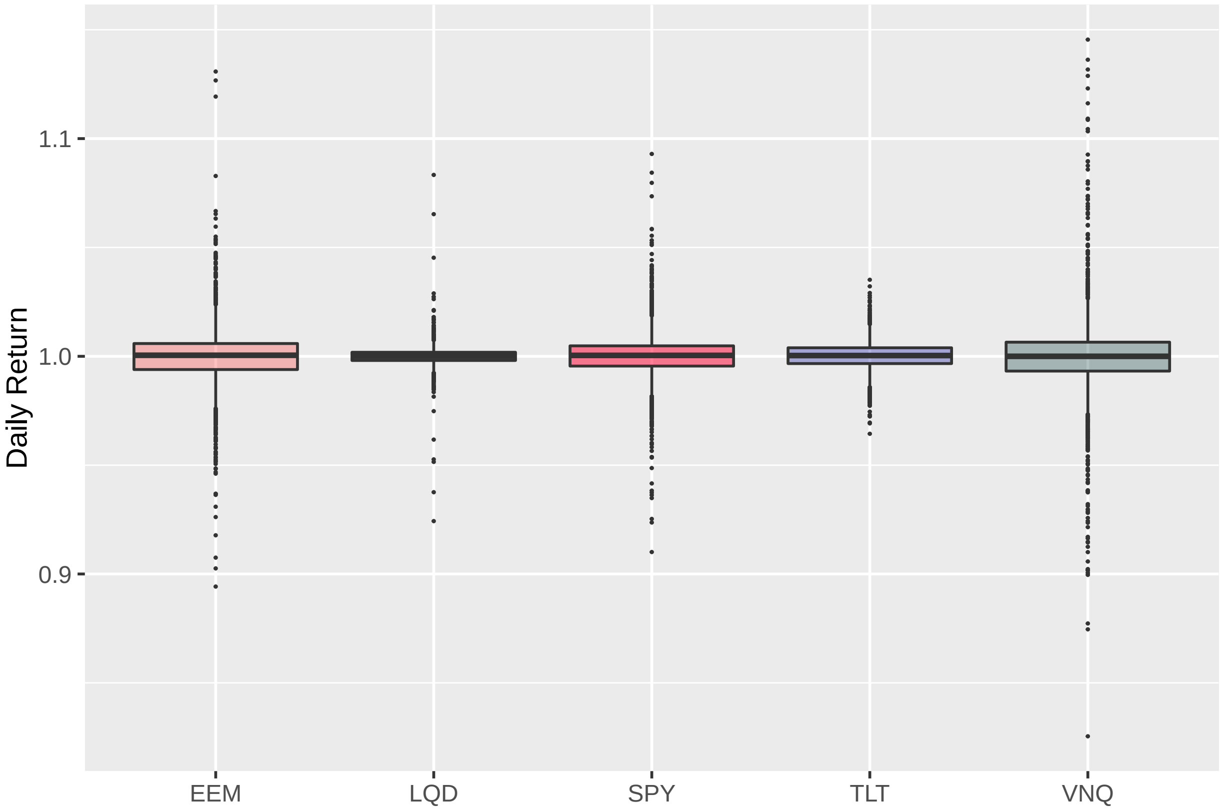

LQD


EEM



To make horizontal comparison between the 5 instruments, let's create a boxplot.

```r
ggplot(data=df_SPY, aes(y=Return)) +
    geom_boxplot(aes(x="SPY"), fill="#FF0033", outlier.size=.1, alpha=0.5) +
    geom_boxplot(data=df_TLT, aes(x="TLT"), fill="#000099", outlier.size=.1, alpha=0.3) +
    geom_boxplot(data=df_LQD, aes(x="LQD"), fill="#006633", outlier.size=.1, alpha=0.3) +
    geom_boxplot(data=df_EEM, aes(x="EEM"), fill="#FF3330", outlier.size=.1, alpha=0.3) +
    geom_boxplot(data=df_VNQ, aes(x="VNQ"), fill="#003333", outlier.size=.1, alpha=0.3) +
    labs(x=" ", y="Daily Return")
```



```r
return_sd <- sapply(list(df_SPY$Return, df_TLT$Return, df_LQD$Return, df_EEM$Return, df_VNQ$Return), sd
names(return_sd) <- c("SPY", "TLT", "LQD", "EEM", "VNQ")
return_sd
```

```
>>>         SPY         TLT         LQD         EEM         VNQ
>>> 0.009963649 0.006361433 0.004565711 0.013196954 0.018698470
```

From the boxplot, it is observed TLT offers the least variation and VNQ has the largest fluctuation.

Let's also calculate the mean daily return for each instrument.

```r
return_mean <- sapply(list(df_SPY$Return, df_TLT$Return, df_LQD$Return, df_EEM$Return, df_VNQ$Return), m
names(return_mean) <- c("SPY", "TLT", "LQD", "EEM", "VNQ")
return_mean
```

```
>>>       SPY       TLT       LQD       EEM       VNQ
>>> 1.0000261 1.0001763 0.9999080 0.9999243 0.9997875
```

To construct the "safe" portfolio, we here choose the 3 instruments with lowest standard deviation, i.e. SPY, TLT and LQD, and assign higher weight to lower sd, so therefore SPY: 10%, TLT: 40% and LQD: 50%.

For the "aggressive" portfolio, we use the other 2 instruments, and assign higher weight to higher mean return (lower mean loss). So let's say EEM: 60% and VNQ: 40%.

To simulate the performance of the 3 portfolios requires bootstrapping, let's import the necessary libraries first.

```r
library(foreach)
library(mosaic)
```

Then define a helper function to make bootstrap sampling on each instrument. Take 1000 samples of 20-day day-by-day returns.

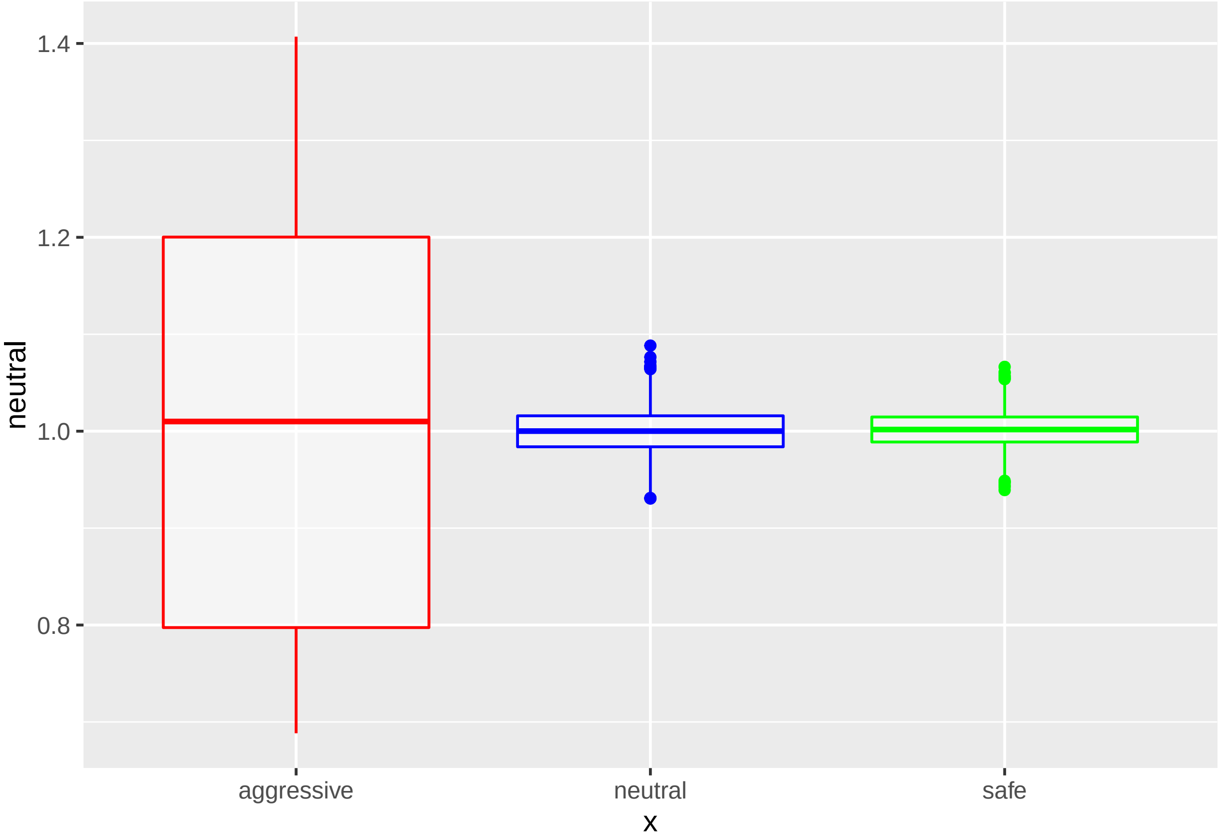
```
sample_return <- function(v){
    boot <- foreach(i=1:1000, .combine="c") %do% {
        daybyday_bootstrap <- resample(v$Return, 20, replace=TRUE)
        prod(daybyday_bootstrap)
    }
}

df_return <- sapply(list(df_SPY, df_TLT, df_LQD, df_EEM, df_VNQ), sample_return)
colnames(df_return) <- c("SPY", "TLT", "LQD", "EEM", "VNQ")
df_return <- as.data.frame(df_return)
```

Then generate portfolio returns according to predefined weights.

```
return_neutral <- rowSums(rep(0.2, 5) * df_return)
return_safe <- rowSums(c(0.1, 0.4, 0.5) * select(df_return, SPY, TLT, LQD))
return_aggressive <- rowSums(c(0.6, 0.4) * select(df_return, EEM, VNQ))
portfolio_return <- as.data.frame(cbind(return_neutral, return_safe, return_aggressive))
colnames(portfolio_return) <- c("neutral", "safe", "aggressive")
```

Let's make a boxplot to compare the return characteristics of the 3 strategies.

```
ggplot(data=portfolio_return, aes(y=neutral)) +
    geom_boxplot(aes(x="neutral"), color="blue", alpha=0.5) +
    geom_boxplot(aes(x="safe", y=safe), color="green", alpha=0.5) +
    geom_boxplot(aes(x="aggressive", y=aggressive), color="red", alpha=0.5)
```



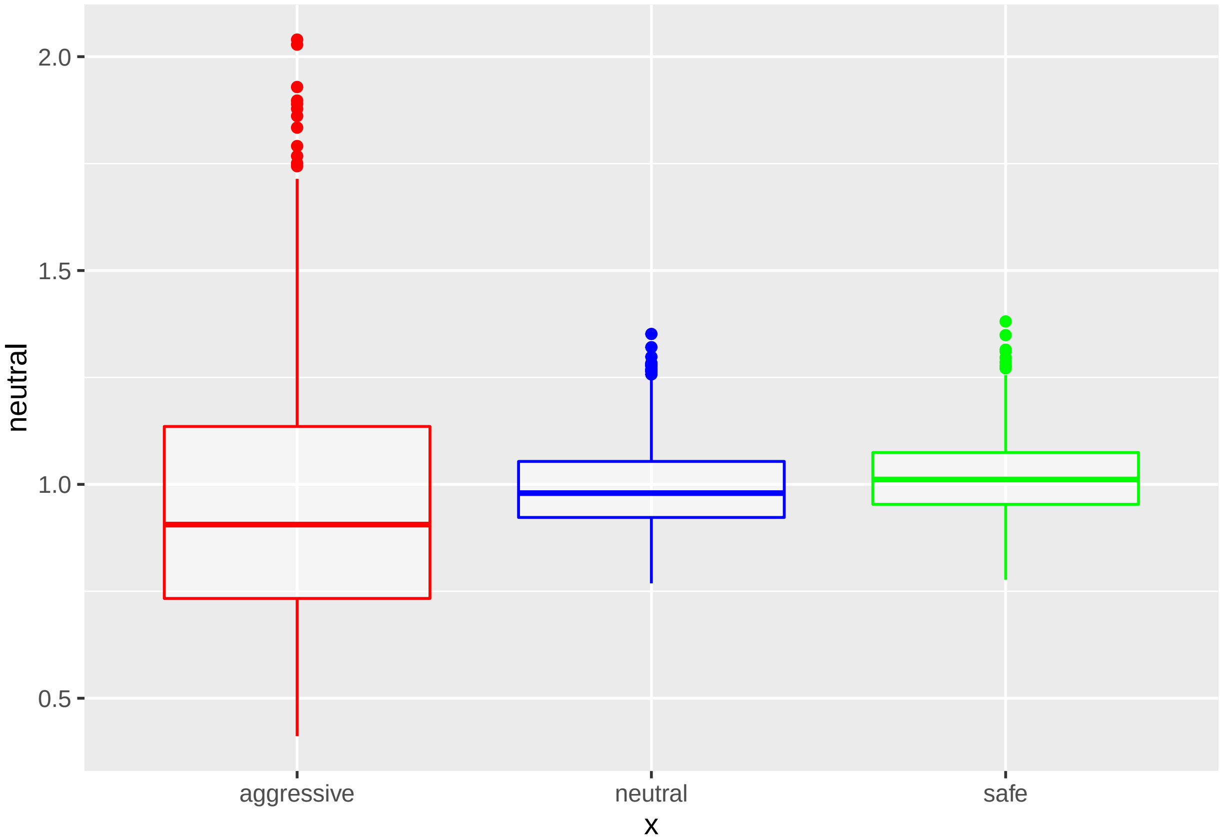
Value at risk at 5% level is the 5% quantile point of each portfolio, printed below.

```
portfolio_VaR <- 1e5 * (1- sapply(portfolio_return, function(v) quantile(x=v, probs=0.05)))
portfolio_VaR
```

```
>>>    neutral.5%       safe.5% aggressive.5%
>>>      3733.996      3373.969     25660.085
```

Investors are primarily concerned with annual returns of portfolios, including the standard deviation of each. Therefore we can repeat the above process but take 1000 bootstrap samples of 365-day day-by-day returns.

```
annual_return <- function(v){
    boot <- foreach(i=1:1000, .combine="c") %do% {
        daybyday_bootstrap <- resample(v$Return, 365, replace=TRUE)
        prod(daybyday_bootstrap)
    }
}

annual_return <- sapply(list(df_SPY, df_TLT, df_LQD, df_EEM, df_VNQ), annual_return)
colnames(annual_return) <- c("SPY", "TLT", "LQD", "EEM", "VNQ")
annual_return <- as.data.frame(annual_return)

annual_return_neutral <- rowSums(rep(0.2, 5) * annual_return)
annual_return_safe <- rowSums(c(0.1, 0.4, 0.5) * select(annual_return, SPY, TLT, LQD))
annual_return_aggressive <- rowSums(c(0.6, 0.4) * select(annual_return, EEM, VNQ))
portfolio_annual_return <- as.data.frame(cbind(annual_return_neutral, annual_return_safe, annual_return_
colnames(portfolio_annual_return) <- c("neutral", "safe", "aggressive")

ggplot(data=portfolio_annual_return, aes(y=neutral)) +
    geom_boxplot(aes(x="neutral"), color="blue", alpha=0.5) +
    geom_boxplot(aes(x="safe", y=safe), color="green", alpha=0.5) +
    geom_boxplot(aes(x="aggressive", y=aggressive), color="red", alpha=0.5)
```



```
portfolio_summary <- rbind(sapply(portfolio_annual_return, mean), sapply(portfolio_annual_return, sd))
row.names(portfolio_summary) <- c("Mean Return", "Std")
portfolio_summary
```

```
>>>                 neutral       safe aggressive
>>> Mean Return 0.99204163 1.01589779  0.9539374
>>> Std         0.09685765 0.09322608  0.2873082
```

Market segmentation

```
tweets <- read.csv("data/social_marketing.csv")
```

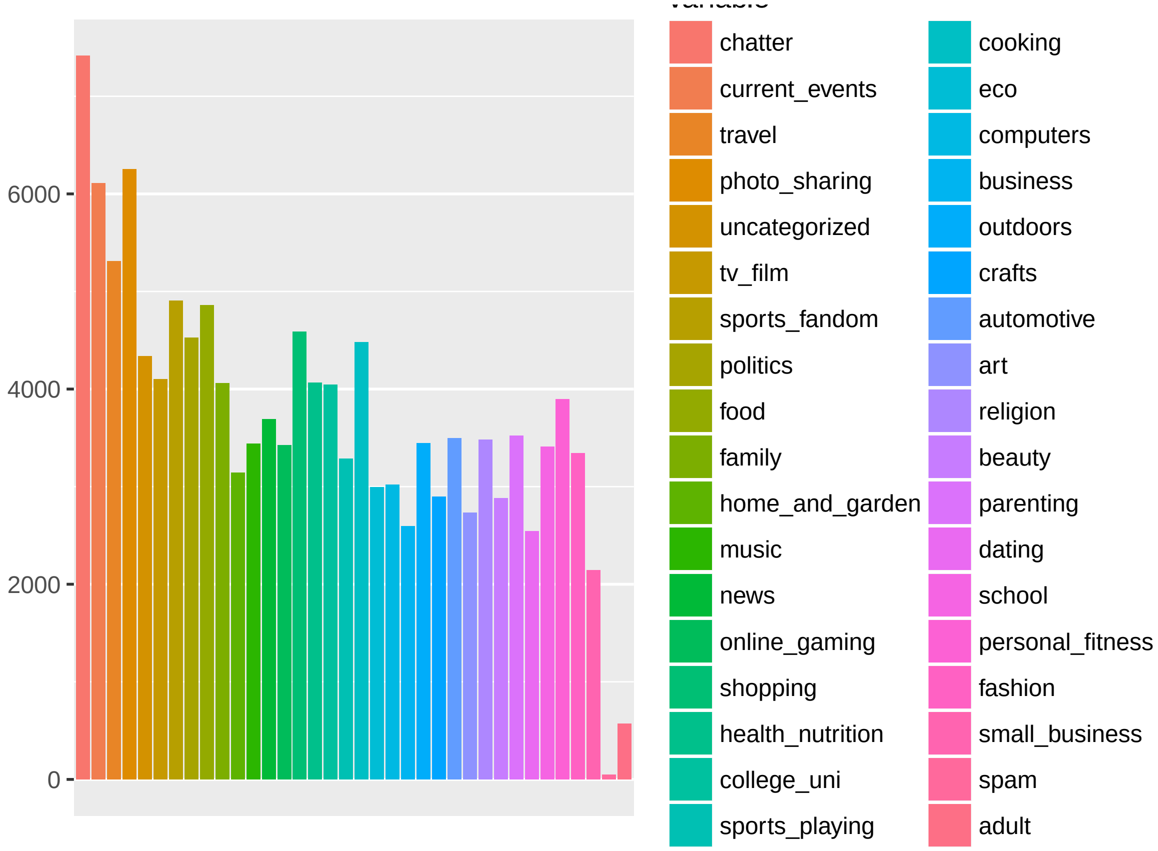
In order to accurately tailor its messages to customers' demand, the company needs to understand what its customers most care about, and capture the primary topics that appeal to the broadest population among its followers.

Let's plot the average frequency of each topic appearing in the tweets. Whenever a topic is mentioned by a tweeter, it will be counted as 1, regardless how many times it is occurs.

```
tweets_count <- sapply(tweets, function(v)ifelse(as.numeric(v)>0, as.numeric(1), 0))[, -1]
tweets_count <- as.data.frame(tweets_count)
# sapply(tweets_count, mean)
tweets_melt <- melt(tweets_count, value.name="value")
```

```
>>> No id variables; using all as measure variables
```

```r
ggplot(data=tweets_melt, aes(x=variable, y=value, fill=variable)) +
    geom_bar(stat="identity") +
    labs(x=" ", y=" ") +
    scale_x_discrete(breaks=0)
```



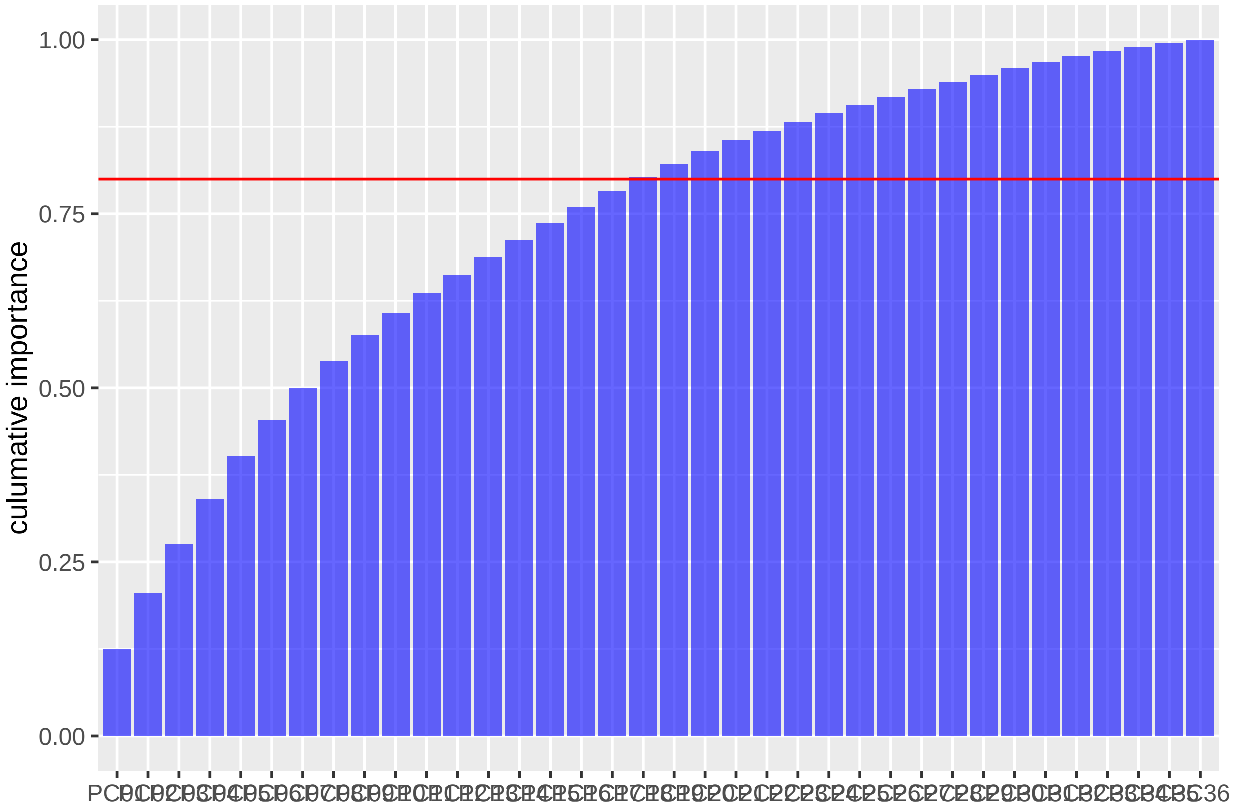
From the plot it can be seen that apart from `chatter`, `current_events`, `travel`, `photo_sharing`, `sports_fandom` and `food` are the most common topics, and therefore including information relevant to these topics in the company's message will generally appeal to most audience, which is a safe way to broadcast the brand.

However, only sending hot topics may not necessarily bring the most benefit possible to the company, since such information is hardly differentiated in the context of social media, and could deteriorate into cliché.

Therefore, it is essential to also occasionally include user-group-specific elements, which are particularly attractive to certain customer cohorts. This process in the core is to capture the variance of different tweeters' favourite topics, and Principal Component Analysis serves such purposes by extracting underlying defining characteristics.

Let's plot the PCs of the tweet topics.

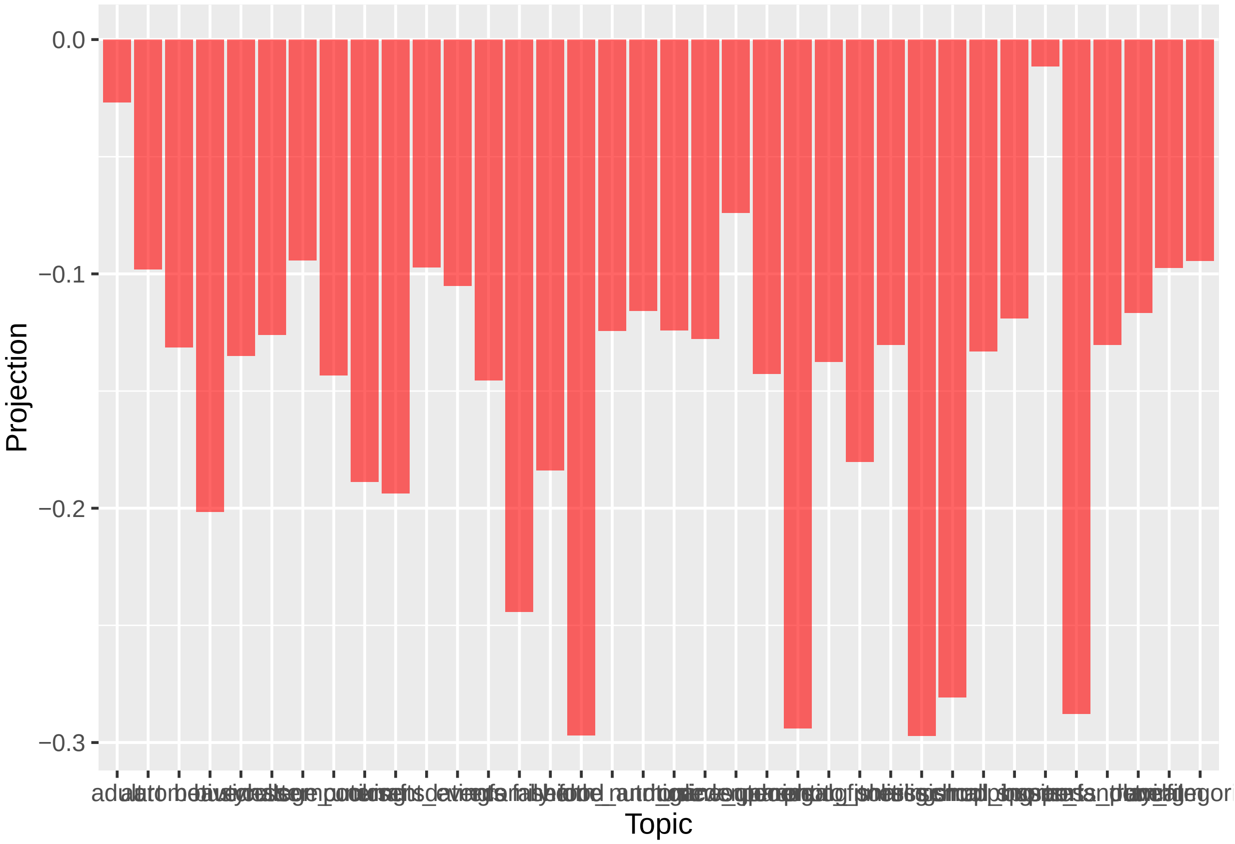
```r
tweets_pc <- prcomp(tweets[, -1], center=TRUE, scale=TRUE)
tweets_pc_imp <- melt(summary(tweets_pc)$importance[3,], value.name="Cumulative_importance")
x_index <- paste("PC0", c(1:9), sep="")
x_index <- c(x_index, paste("PC", c(10:36), sep=""))
ggplot(data=tweets_pc_imp, aes(x=x_index, y=Cumulative_importance)) +
    geom_bar(stat="identity", fill="blue", alpha=0.6) +
    geom_hline(yintercept=0.8, colour="red") +
    labs(x=" ", y="culumative importance")
```



To reduce the dimension and extract primary information of the topics, we only select the most significant PCs whose cumulative importance is around 80%, and therefore only #1-18 PC are considered.

Then let's have a look at what these PCs represent. Take PC1 as an example.

```r
pc <- tweets_pc$rotation
pc1 <- pc[,1]
pc1_df <- data.frame(names(pc1), pc1)
row.names(pc1_df) <- NULL
colnames(pc1_df) <- c("Topic", "Projection")
ggplot(data=pc1_df, aes(x=Topic, y=Projection)) +
    geom_bar(stat="identity", fill="red", alpha=0.6)
```



The barplot shows the projection of PC1 on each topics, which is equivalent to the weight of each topic contained in PC1. From the projection pattern it is observed `food`, `parenting`, `religion`, `sports_fandom` and `school` are most represented in PC1. Repeating the above process across PC2 to PC18 will give more combinations, with which the company could tailor its messages to specific consumer groups. However, this should be done in caution, since PC in nature captures most variable features. Variance in our context here could also indicate controversy in the topic, which some users try to avoid while others find noteworthy. Unconscious involvement in such issues may risk the business.