Lyft-Uber-Price-Prediction

Nishtha Chaudhary

10 October 2019

IMPORTING DATASETS AND CLEANING THEM

Importing dataset cab_rides

```
cab_rides <- read.csv("C:/Users/nisht/Desktop/MITA/Fall/MVA/Final
Project/cab_rides.csv")
summary(cab_rides)
```

```
##     distance     cab_type        time_stamp
##  Min.   :0.020   Lyft:307408   Min.   :1.543e+12
##  1st Qu.:1.280   Uber:385663   1st Qu.:1.543e+12
##  Median :2.160                 Median :1.544e+12
##  Mean   :2.189                 Mean   :1.544e+12
##  3rd Qu.:2.920                 3rd Qu.:1.545e+12
##  Max.   :7.860                 Max.   :1.545e+12
##
##              destination                    source          price
##  Financial District: 58851   Financial District: 58857   Min.   : 2.50
##  Theatre District  : 57798   Theatre District  : 57813   1st Qu.: 9.00
##  Back Bay          : 57780   Back Bay          : 57792   Median :13.50
##  Boston University : 57764   Boston University : 57764   Mean   :16.55
##  Haymarket Square  : 57764   North End         : 57763   3rd Qu.:22.50
##  Fenway            : 57757   Fenway            : 57757   Max.   :97.50
##  (Other)           :345357   (Other)           :345325   NA's   :55095
##  surge_multiplier                                  id
##  Min.   :1.000    00005b8c-5647-4104-9ac6-94fa6a40f3c3:     1
##  1st Qu.:1.000    00006eeb-0183-40c1-8198-c441d3c8a734:     1
##  Median :1.000    00008b42-5ecc-4f66-b4b9-b22a331634e6:     1
##  Mean   :1.014    000094c0-00c4-43f1-ae1b-4693eec2a580:     1
##  3rd Qu.:1.000    0000a8b2-e4d3-4227-8374-af8a2366e475:     1
##  Max.   :3.000    0000b5d6-59be-4534-b371-8214334d94f0:     1
##                   (Other)                                :693065
##                                product_id            name
##  6d318bcc-22a3-4af6-bddd-b409bfce1546: 55096   Black SUV: 55096
##  6f72dfc5-27f1-42e8-84db-ccc7a75f6969: 55096   UberXL   : 55096
##  9a0e7b09-b92b-4c41-9779-2ad22b4d779d: 55096   WAV      : 55096
##  6c84fd89-3f11-4782-9b50-97c468b19529: 55095   Black    : 55095
##  8cf7e821-f0d3-49c6-8eba-e679c0ebcf6a: 55095   Taxi     : 55095
##  55c66225-fbe7-4fd5-9072-eab1ece5e23e: 55094   UberX    : 55094
##  (Other)                              :362499   (Other)  :362499
```

```
cab_data<-cab_rides
```

Creating a date_time column

```
cab_data$date_time<-as.POSIXct((cab_data$time_stamp/1000),origin = "1970-01-
01 00:53:20", tz="GMT")
```

Importing dataset weather

```
weather <- read.csv("C:/Users/nisht/Desktop/MITA/Fall/MVA/Final
Project/weather.xls")
summary(weather)
```

```
##     ï..temp                     location        clouds
##  Min.   :19.62   Back Bay           : 523   Min.   :0.0000
##  1st Qu.:36.08   Beacon Hill        : 523   1st Qu.:0.4400
##  Median :40.13   Boston University  : 523   Median :0.7800
##  Mean   :39.09   Fenway             : 523   Mean   :0.6778
##  3rd Qu.:42.83   Financial District: 523   3rd Qu.:0.9700
##  Max.   :55.41   Haymarket Square   : 523   Max.   :1.0000
##                  (Other)            :3138
##     pressure           rain          time_stamp           humidity
##  Min.   : 988.2   Min.   :0.000   Min.   :1.543e+09   Min.   :0.450
##  1st Qu.: 997.7   1st Qu.:0.005   1st Qu.:1.543e+09   1st Qu.:0.670
##  Median :1007.7   Median :0.015   Median :1.544e+09   Median :0.760
##  Mean   :1008.4   Mean   :0.058   Mean   :1.544e+09   Mean   :0.764
##  3rd Qu.:1018.5   3rd Qu.:0.061   3rd Qu.:1.545e+09   3rd Qu.:0.890
##  Max.   :1035.1   Max.   :0.781   Max.   :1.545e+09   Max.   :0.990
##                   NA's   :5382
##       wind
##  Min.   : 0.290
##  1st Qu.: 3.518
##  Median : 6.570
##  Mean   : 6.803
##  3rd Qu.: 9.920
##  Max.   :18.180
##
```

```
str(weather)
```

```
## 'data.frame':    6276 obs. of  8 variables:
##  $ ï..temp   : num  42.4 42.4 42.5 42.1 43.1 ...
##  $ location  : Factor w/ 12 levels "Back Bay","Beacon Hill",..: 1 2 3 4 5
6 7 8 9 10 ...
##  $ clouds    : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ pressure  : num  1012 1012 1012 1012 1012 ...
##  $ rain      : num  0.1228 0.1846 0.1089 0.0969 0.1786 ...
##  $ time_stamp: int  1545003901 1545003901 1545003901 1545003901 1545003901
1545003901 1545003901 1545003901 1545003901 1545003901 ...
```

```
##  $ humidity  : num  0.77 0.76 0.76 0.77 0.75 0.77 0.77 0.77 0.78 0.75 ...
##  $ wind      : num  11.2 11.3 11.1 11.1 11.5 ...

weather_data<-weather
```

creating a date_time column in weather_data

```
weather_data$date_time<-as.POSIXct(weather_data$time_stamp,origin = "1970-01-
01 00:53:20", tz="GMT")
str(weather_data)

## 'data.frame':    6276 obs. of  9 variables:
##  $ ï..temp   : num  42.4 42.4 42.5 42.1 43.1 ...
##  $ location  : Factor w/ 12 levels "Back Bay","Beacon Hill",..: 1 2 3 4 5
6 7 8 9 10 ...
##  $ clouds    : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ pressure  : num  1012 1012 1012 1012 1012 ...
##  $ rain      : num  0.1228 0.1846 0.1089 0.0969 0.1786 ...
##  $ time_stamp: int  1545003901 1545003901 1545003901 1545003901 1545003901
1545003901 1545003901 1545003901 1545003901 1545003901 ...
##  $ humidity  : num  0.77 0.76 0.76 0.77 0.75 0.77 0.77 0.77 0.78 0.75 ...
##  $ wind      : num  11.2 11.3 11.1 11.1 11.5 ...
##  $ date_time : POSIXct, format: "2018-12-17 00:38:21" "2018-12-17
00:38:21" ...
```

merge the datasets to reflect the same time for a location

```
cab_data$merge_date<-paste(cab_data$source,"-",as.Date(cab_data$date_time),"-
",format(cab_data$date_time,"%H:%M:%S"))
weather_data$merge_date<-paste(weather_data$location,"-
",as.Date(weather_data$date_time),"-
",format(weather_data$date_time,"%H:%M:%S"))

#making those values as characters
weather_data$merge_date<-as.character(weather_data$merge_date)
cab_data$merge_date<-as.character(cab_data$merge_date)
```

verify that merge_date has unique values.

```
weather_data<-subset(weather_data,!duplicated(weather_data$merge_date))
isTRUE(duplicated(weather_data$merge_date))

## [1] FALSE
```

Merging both the dataframes.

```
merge_data<-merge(x=weather_data, y=cab_data,by='merge_date', all.x=TRUE)
str(merge_data)
```

```
## 'data.frame':    9306 obs. of  21 variables:
##  $ merge_date      : chr  "Back Bay - 2018-11-26 - 04:34:05" "Back Bay -
2018-11-26 - 05:34:13" "Back Bay - 2018-11-26 - 05:34:58" "Back Bay - 2018-
11-26 - 05:36:38" ...
##  $ ï..temp         : num  41 40.6 40.6 40.6 40.6 ...
##  $ location        : Factor w/ 12 levels "Back Bay","Beacon Hill",..: 1 1
1 1 1 1 1 1 1 1 ...
##  $ clouds          : num  0.87 0.86 0.86 0.86 0.86 0.95 0.95 0.94 0.93
0.93 ...
##  $ pressure        : num  1014 1014 1014 1014 1014 ...
##  $ rain            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ time_stamp.x    : int  1543203645 1543207253 1543207298 1543207398
1543207398 1543207777 1543207777 1543208142 1543208578 1543209183 ...
##  $ humidity        : num  0.92 0.93 0.93 0.93 0.93 0.92 0.92 0.92 0.92
0.92 ...
##  $ wind            : num  1.46 2.57 2.59 2.65 2.65 2.59 2.59 2.83 3 3.01
...
##  $ date_time.x     : POSIXct, format: "2018-11-26 04:34:05" "2018-11-26
05:34:13" ...
##  $ distance        : num  NA NA 1.44 1.36 1.22 1.34 1.1 NA NA NA ...
##  $ cab_type        : Factor w/ 2 levels "Lyft","Uber": NA NA 2 1 2 2 2 NA
NA NA ...
##  $ time_stamp.y    : num  NA NA 1.54e+12 1.54e+12 1.54e+12 ...
##  $ destination     : Factor w/ 12 levels "Back Bay","Beacon Hill",..: NA
NA 3 10 9 4 9 NA NA NA ...
##  $ source          : Factor w/ 12 levels "Back Bay","Beacon Hill",..: NA
NA 1 1 1 1 1 NA NA NA ...
##  $ price           : num  NA NA 8.5 16.5 NA 26.5 7.5 NA NA NA ...
##  $ surge_multiplier: num  NA NA 1 1 1 1 1 NA NA NA ...
##  $ id              : Factor w/ 693071 levels "00005b8c-5647-4104-9ac6-
94fa6a40f3c3",..: NA NA 548701 610037 513190 566219 94420 NA NA NA ...
##  $ product_id      : Factor w/ 13 levels "55c66225-fbe7-4fd5-9072-
eab1ece5e23e",..: NA NA 7 10 5 3 1 NA NA NA ...
##  $ name            : Factor w/ 13 levels "Black","Black SUV",..: NA NA 13
4 9 2 11 NA NA NA ...
##  $ date_time.y     : POSIXct, format: NA NA ...
```

Handling Missing values

```
#Filling NA values in price
merge_data$rain[is.na(merge_data$rain)]<-0

#Extracting the numerical columns in a new dataframe "df"
merge_data$temp<-merge_data[,c(2)] #renaming a column
df<-merge_data[,c(4,5,6,8,9,10,11,17,22,16)]

#Data preparation
#Dealing with missing values
summary(merge_data)
```

```
##   merge_date           ï..temp                         location
##  Length:9306        Min.   :19.62   Haymarket Square       : 843
##  Class :character   1st Qu.:36.74   North Station          : 801
##  Mode  :character   Median :39.73   Theatre District       : 800
##                     Mean   :39.12   Northeastern University: 788
##                     3rd Qu.:41.86   North End              : 772
##                     Max.   :55.41   Fenway                 : 771
##                                     (Other)                :4531
##      clouds          pressure           rain          time_stamp.x
##  Min.   :0.0000   Min.   : 988.2   Min.   :0.00000   Min.   :1.543e+09
##  1st Qu.:0.4500   1st Qu.: 992.2   1st Qu.:0.00000   1st Qu.:1.543e+09
##  Median :0.7700   Median :1002.2   Median :0.00000   Median :1.543e+09
##  Mean   :0.6799   Mean   :1005.2   Mean   :0.01197   Mean   :1.544e+09
##  3rd Qu.:0.9700   3rd Qu.:1014.4   3rd Qu.:0.00000   3rd Qu.:1.544e+09
##  Max.   :1.0000   Max.   :1035.1   Max.   :0.78070   Max.   :1.545e+09
##
##     humidity           wind         date_time.x
##  Min.   :0.4500   Min.   : 0.290   Min.   :2018-11-26 04:34:04
##  1st Qu.:0.6700   1st Qu.: 4.183   1st Qu.:2018-11-28 01:38:42
##  Median :0.7500   Median : 7.490   Median :2018-11-28 23:55:29
##  Mean   :0.7623   Mean   : 7.212   Mean   :2018-12-01 23:49:51
##  3rd Qu.:0.8800   3rd Qu.: 9.990   3rd Qu.:2018-12-02 09:31:14
##  Max.   :0.9900   Max.   :18.180   Max.   :2018-12-18 19:38:22
##
##     distance     cab_type     time_stamp.y                    destination
##  Min.   :0.020   Lyft:1732   Min.   :1.543e+12   Fenway            : 344
##  1st Qu.:1.250   Uber:2134   1st Qu.:1.543e+12   Financial District: 342
##  Median :2.140   NA's:5440   Median :1.543e+12   Back Bay          : 337
##  Mean   :2.168               Mean   :1.543e+12   Beacon Hill       : 335
##  3rd Qu.:2.947               3rd Qu.:1.543e+12   South Station     : 334
##  Max.   :7.460               Max.   :1.545e+12   (Other)           :2174
##  NA's   :5440                NA's   :5440        NA's              :5440
##                       source         price       surge_multiplier
##  Haymarket Square       : 392   Min.   : 2.50   Min.   :1.000
##  North Station          : 351   1st Qu.: 9.00   1st Qu.:1.000
##  Theatre District       : 344   Median :13.50   Median :1.000
##  Northeastern University: 329   Mean   :16.67   Mean   :1.018
##  North End              : 316   3rd Qu.:22.50   3rd Qu.:1.000
##  (Other)                :2134   Max.   :92.00   Max.   :2.000
##  NA's                   :5440   NA's   :5758    NA's   :5440
##                                     id
##  000baa63-5e1c-4f9d-891c-e4e78e830199:   1
##  002b15bc-b433-44a4-8174-b8ac95caebf8:   1
##  00423464-fb1b-4e96-9154-b55a00854181:   1
##  00552d6f-c5fa-4006-962a-4613097afabe:   1
##  005ca94d-9dad-4b34-a8ce-82a6de9058b4:   1
##  (Other)                              :3861
##  NA's                                 :5440
##                                product_id          name
##  8cf7e821-f0d3-49c6-8eba-e679c0ebcf6a: 318   Taxi     : 318
```

```
##  6d318bcc-22a3-4af6-bddd-b409bfce1546: 308   Black SUV: 308
##  6c84fd89-3f11-4782-9b50-97c468b19529: 307   Black    : 307
##  6f72dfc5-27f1-42e8-84db-ccc7a75f6969: 306   UberPool : 306
##  997acbb5-e102-41e1-b155-9df7de0a73f2: 306   UberXL   : 306
##  (Other)                               :2321   (Other)  :2321
##  NA's                                  :5440   NA's     :5440
##   date_time.y                        temp
##  Min.   :2018-11-26 04:34:06   Min.   :19.62
##  1st Qu.:2018-11-27 03:08:42   1st Qu.:36.74
##  Median :2018-11-28 14:25:28   Median :39.73
##  Mean   :2018-11-28 08:15:46   Mean   :39.12
##  3rd Qu.:2018-11-29 00:42:54   3rd Qu.:41.86
##  Max.   :2018-12-16 20:38:27   Max.   :55.41
##  NA's   :5440
```

```
summary(df)
```

```
##      clouds          pressure           rain            humidity
##  Min.   :0.0000   Min.   : 988.2   Min.   :0.00000   Min.   :0.4500
##  1st Qu.:0.4500   1st Qu.: 992.2   1st Qu.:0.00000   1st Qu.:0.6700
##  Median :0.7700   Median :1002.2   Median :0.00000   Median :0.7500
##  Mean   :0.6799   Mean   :1005.2   Mean   :0.01197   Mean   :0.7623
##  3rd Qu.:0.9700   3rd Qu.:1014.4   3rd Qu.:0.00000   3rd Qu.:0.8800
##  Max.   :1.0000   Max.   :1035.1   Max.   :0.78070   Max.   :0.9900
##
##       wind         date_time.x                     distance
##  Min.   : 0.290   Min.   :2018-11-26 04:34:04   Min.   :0.020
##  1st Qu.: 4.183   1st Qu.:2018-11-28 01:38:42   1st Qu.:1.250
##  Median : 7.490   Median :2018-11-28 23:55:29   Median :2.140
##  Mean   : 7.212   Mean   :2018-12-01 23:49:51   Mean   :2.168
##  3rd Qu.: 9.990   3rd Qu.:2018-12-02 09:31:14   3rd Qu.:2.947
##  Max.   :18.180   Max.   :2018-12-18 19:38:22   Max.   :7.460
##                                                 NA's   :5440
##  surge_multiplier      temp           price
##  Min.   :1.000    Min.   :19.62   Min.   : 2.50
##  1st Qu.:1.000    1st Qu.:36.74   1st Qu.: 9.00
##  Median :1.000    Median :39.73   Median :13.50
##  Mean   :1.018    Mean   :39.12   Mean   :16.67
##  3rd Qu.:1.000    3rd Qu.:41.86   3rd Qu.:22.50
##  Max.   :2.000    Max.   :55.41   Max.   :92.00
##  NA's   :5440                     NA's   :5758
```

```
merge_data$surge_multiplier = ifelse(is.na(merge_data$surge_multiplier),
                                     ave(merge_data$surge_multiplier , FUN =
function(x) mean(x, na.rm = TRUE)),
                                     merge_data$surge_multiplier)

merge_data$price = ifelse(is.na(merge_data$price),
                          ave(merge_data$price , FUN = function(x) mean(x,
na.rm = TRUE)),
```

```
                          merge_data$price)

df$distance = ifelse(is.na(df$distance),
                     ave(df$distance , FUN = function(x) mean(x, na.rm =
TRUE)),
                     df$distance)

df$surge_multiplier = ifelse(is.na(df$surge_multiplier),
                             ave(df$surge_multiplier , FUN = function(x)
mean(x, na.rm = TRUE)),
                             df$surge_multiplier)

df$price = ifelse(is.na(df$price),
                  ave(df$price , FUN = function(x) mean(x, na.rm = TRUE)),
                  df$price)
```

Checking for null values

```
any(is.na(df))

## [1] FALSE
```

Adding date and time column in the df data set

```
df$day<-weekdays(df$date_time)
df$time<-format(df$date_time.x,"%H:%M:%S")
df$date_time<-as.Date(df$date_time.x)
merge_data$day=weekdays(merge_data$date_time.x)
```

CORRELATION, COVARIANCE AND DISTANCE

```
#We are calculating for: clouds, pressure, rain, humidity, wind, distance,
surge_multiplier, temp, price
covariance<-cov(df[,c(1,2,3,4,5,7,8,9,10)]) #variamce-covariance matrix
created
correlation<-cor(df[,c(1,2,3,4,5,7,8,9,10)]) #standardized
#colmeans
cm<-colMeans(df[,c(1,2,3,4,5,7,8,9,10)])
distance<-dist(scale(df[,c(1,2,3,4,5,7,8,9,10)],center=FALSE))
#Calculating di(generalized distance for all observations of our data)
#before that first extract all numeric variable in a dataframe
x<-df[,c(1,2,3,4,5,7,8,9,10)]
d <- apply(x, MARGIN = 1, function(x) + t(x - cm) %*% solve(covariance) %*%
(x - cm))
```

Pca || T-test || F-test

```
#Keeping only the independent variables
x<-x[,c(-9)]
summary(x)
```

```
##      clouds          pressure           rain            humidity
##  Min.   :0.0000   Min.   : 988.2   Min.   :0.00000   Min.   :0.4500
##  1st Qu.:0.4500   1st Qu.: 992.2   1st Qu.:0.00000   1st Qu.:0.6700
##  Median :0.7700   Median :1002.2   Median :0.00000   Median :0.7500
##  Mean   :0.6799   Mean   :1005.2   Mean   :0.01197   Mean   :0.7623
##  3rd Qu.:0.9700   3rd Qu.:1014.4   3rd Qu.:0.00000   3rd Qu.:0.8800
##  Max.   :1.0000   Max.   :1035.1   Max.   :0.78070   Max.   :0.9900
##       wind           distance     surge_multiplier      temp
##  Min.   : 0.290   Min.   :0.020   Min.   :1.000    Min.   :19.62
##  1st Qu.: 4.183   1st Qu.:2.168   1st Qu.:1.000    1st Qu.:36.74
##  Median : 7.490   Median :2.168   Median :1.018    Median :39.73
##  Mean   : 7.212   Mean   :2.168   Mean   :1.018    Mean   :39.12
##  3rd Qu.: 9.990   3rd Qu.:2.168   3rd Qu.:1.018    3rd Qu.:41.86
##  Max.   :18.180   Max.   :7.460   Max.   :2.000    Max.   :55.41
```

Get the Correlations between the measurements

```
cor(x)
```

```
##                        clouds     pressure         rain     humidity
## clouds             1.000000000  0.049915595  0.179471808  0.416612604
## pressure           0.049915595  1.000000000 -0.003445035  0.058937875
## rain               0.179471808 -0.003445035  1.000000000  0.200335181
## humidity           0.416612604  0.058937875  0.200335181  1.000000000
## wind               0.029005927 -0.551280893  0.227242203 -0.364423309
## distance           0.016025970  0.003597521 -0.013858769  0.001519042
## surge_multiplier -0.001346841  0.015364812 -0.026535193  0.014156441
## temp               0.536578398 -0.149398121  0.162681890  0.333597659
##                            wind      distance surge_multiplier          temp
## clouds             0.029005927  0.0160259703     -0.001346841  0.5365783977
## pressure          -0.551280893  0.0035975207      0.015364812 -0.1493981210
## rain               0.227242203 -0.0138587690     -0.026535193  0.1626818904
## humidity          -0.364423309  0.0015190420      0.014156441  0.3335976588
## wind               1.000000000 -0.0029100842     -0.013216398  0.1213224650
## distance          -0.002910084  1.0000000000      0.040994672  0.0008661221
## surge_multiplier -0.013216398  0.0409946724      1.000000000  0.0029074916
## temp               0.121322465  0.0008661221      0.002907492  1.0000000000
```

Using prcomp to compute the principal components (eigenvalues and eigenvectors).

With scale=TRUE, variable means are set to zero, and variances set to one

```
x_pca <- prcomp(x,scale=TRUE)
x_pca
```

```
## Standard deviations (1, .., p=8):
## [1] 1.4019027 1.3075001 1.0226282 0.9797210 0.9631086 0.8396878 0.6678857
## [8] 0.4906786
##
## Rotation (n x k) = (8 x 8):
##                             PC1         PC2         PC3           PC4
## clouds            -0.5829822912 -0.01080413  0.03947368 -0.0002121081
## pressure           0.0038238073  0.60490429 -0.07696821  0.0646697039
## rain              -0.2996783226 -0.19653723 -0.18988372  0.1223888901
## humidity          -0.5169636088  0.26724759 -0.01070961 -0.0207508644
## wind               0.0359114310 -0.70125409  0.01682833  0.0024191121
## distance          -0.0068148311  0.01214100  0.68630853  0.7250242539
## surge_multiplier -0.0009048287  0.03884838  0.69362452 -0.6721562220
## temp              -0.5492919673 -0.17467688  0.06268686 -0.0542967590
##                            PC5          PC6         PC7          PC8
## clouds             0.14377838  0.403338563  0.63179366  0.275570786
## pressure          -0.30690124  0.564884403 -0.07452904 -0.452915624
## rain              -0.85332411 -0.099368057 -0.12346467  0.259329103
## humidity           0.02618876 -0.637214467  0.14021695 -0.484141650
## wind              -0.10256408  0.205160677  0.19393266 -0.645324614
## distance          -0.04328157 -0.029575693 -0.01941253 -0.001315112
## surge_multiplier -0.25546563  0.004962752  0.00396301  0.016673406
## temp               0.28042138  0.243550965 -0.72275546 -0.023661553
```

```
summary(x_pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4    PC5     PC6     PC7
## Standard deviation     1.4019 1.3075 1.0226 0.9797 0.9631 0.83969 0.66789
## Proportion of Variance 0.2457 0.2137 0.1307 0.1200 0.1159 0.08813 0.05576
## Cumulative Proportion  0.2457 0.4594 0.5901 0.7101 0.8260 0.91415 0.96990
##                           PC8
## Standard deviation     0.4907
## Proportion of Variance 0.0301
## Cumulative Proportion  1.0000
```

```
x_pca$rotation
```

```
##                             PC1         PC2         PC3           PC4
## clouds            -0.5829822912 -0.01080413  0.03947368 -0.0002121081
## pressure           0.0038238073  0.60490429 -0.07696821  0.0646697039
## rain              -0.2996783226 -0.19653723 -0.18988372  0.1223888901
## humidity          -0.5169636088  0.26724759 -0.01070961 -0.0207508644
## wind               0.0359114310 -0.70125409  0.01682833  0.0024191121
## distance          -0.0068148311  0.01214100  0.68630853  0.7250242539
## surge_multiplier -0.0009048287  0.03884838  0.69362452 -0.6721562220
## temp              -0.5492919673 -0.17467688  0.06268686 -0.0542967590
##                            PC5          PC6         PC7          PC8
## clouds             0.14377838  0.403338563  0.63179366  0.275570786
## pressure          -0.30690124  0.564884403 -0.07452904 -0.452915624
## rain              -0.85332411 -0.099368057 -0.12346467  0.259329103
```

```
## humidity          0.02618876 -0.637214467  0.14021695 -0.484141650
## wind             -0.10256408  0.205160677  0.19393266 -0.645324614
## distance         -0.04328157 -0.029575693 -0.01941253 -0.001315112
## surge_multiplier -0.25546563  0.004962752  0.00396301  0.016673406
## temp              0.28042138  0.243550965 -0.72275546 -0.023661553
```

sample scores stored in x_pca$x # singular values (square roots of eigenvalues) stored in x_pca$sdev

loadings (eigenvectors) are stored in x_pca$rotation # variable means stored in x_pca$center

variable standard deviations stored in x_pca$scale

A table containing eigenvalues and %'s accounted, follows

Eigenvalues are sdev^2

```
(eigen_x <- x_pca$sdev^2)
```

```
## [1] 1.9653313 1.7095565 1.0457684 0.9598533 0.9275782 0.7050756 0.4460713
## [8] 0.2407655
```

```
names(eigen_x) <- paste("PC",1:8,sep="")
eigen_x
```

```
##       PC1       PC2       PC3       PC4       PC5       PC6       PC7
## 1.9653313 1.7095565 1.0457684 0.9598533 0.9275782 0.7050756 0.4460713
##       PC8
## 0.2407655
```

```
sumlambdas <- sum(eigen_x)
sumlambdas #total sample variance
```

```
## [1] 8
```

```
propvar <- eigen_x/sumlambdas
propvar
```

```
##        PC1        PC2        PC3        PC4        PC5        PC6
## 0.24566641 0.21369456 0.13072105 0.11998166 0.11594728 0.08813444
##        PC7        PC8
## 0.05575891 0.03009569
```

```
cumvar_x <- cumsum(propvar)
cumvar_x
```

```
##       PC1       PC2       PC3       PC4       PC5       PC6       PC7
## 0.2456664 0.4593610 0.5900820 0.7100637 0.8260110 0.9141454 0.9699043
##       PC8
## 1.0000000
```

```
matlambdas <- rbind(eigen_x,propvar,cumvar_x)
rownames(matlambdas) <- c("Eigenvalues","Prop. variance","Cum. prop.
variance")
round(matlambdas,4)
```

```
##                         PC1    PC2    PC3    PC4    PC5    PC6    PC7
## Eigenvalues          1.9653 1.7096 1.0458 0.9599 0.9276 0.7051 0.4461
## Prop. variance       0.2457 0.2137 0.1307 0.1200 0.1159 0.0881 0.0558
## Cum. prop. variance 0.2457 0.4594 0.5901 0.7101 0.8260 0.9141 0.9699
##                         PC8
## Eigenvalues          0.2408
## Prop. variance       0.0301
## Cum. prop. variance 1.0000
```

Sample scores stored in x_pca$x

```
#x_pca$x
xtyp_pca <- cbind(data.frame(df$price),x_pca$x)
#xtyp_pca
```

Merging price column

```
colnames(xtyp_pca)[colnames(xtyp_pca)=="df.price"] <- "price"
```

T-Test– We see that true difference in all the means is different from zero.

```
t.test(xtyp_pca$PC1,xtyp_pca$price,var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  xtyp_pca$PC1 and xtyp_pca$price
## t = -264.78, df = 18610, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -16.79719 -16.55033
## sample estimates:
##     mean of x     mean of y
## -6.108660e-17  1.667376e+01
```

```r
t.test(xtyp_pca$PC2,xtyp_pca$price,var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  xtyp_pca$PC2 and xtyp_pca$price
## t = -265.71, df = 18610, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -16.79676 -16.55076
## sample estimates:
##     mean of x     mean of y
## -5.651631e-16  1.667376e+01
```

```r
t.test(xtyp_pca$PC3,xtyp_pca$price,var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  xtyp_pca$PC3 and xtyp_pca$price
## t = -268.14, df = 18610, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -16.79564 -16.55188
## sample estimates:
##     mean of x     mean of y
## -2.753177e-16  1.667376e+01
```

```r
t.test(xtyp_pca$PC4,xtyp_pca$price,var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  xtyp_pca$PC4 and xtyp_pca$price
## t = -268.47, df = 18610, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -16.79550 -16.55202
## sample estimates:
##    mean of x    mean of y
## 1.508799e-16 1.667376e+01
```

```r
t.test(xtyp_pca$PC5,xtyp_pca$price,var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  xtyp_pca$PC5 and xtyp_pca$price
## t = -268.59, df = 18610, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

```
##  -16.79544 -16.55208
## sample estimates:
##    mean of x    mean of y
## 4.551674e-16 1.667376e+01
```

```
t.test(xtyp_pca$PC6,xtyp_pca$price,var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  xtyp_pca$PC6 and xtyp_pca$price
## t = -269.42, df = 18610, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -16.79506 -16.55246
## sample estimates:
##     mean of x     mean of y
## -1.442536e-16  1.667376e+01
```

```
t.test(xtyp_pca$PC7,xtyp_pca$price,var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  xtyp_pca$PC7 and xtyp_pca$price
## t = -270.41, df = 18610, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -16.79462 -16.55290
## sample estimates:
##     mean of x     mean of y
## -4.621068e-16  1.667376e+01
```

```
t.test(xtyp_pca$PC8,xtyp_pca$price,var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  xtyp_pca$PC8 and xtyp_pca$price
## t = -271.2, df = 18610, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -16.79427 -16.55325
## sample estimates:
##    mean of x    mean of y
## 4.976572e-16 1.667376e+01
```

F-Test #Testing Variation

```
var.test(xtyp_pca$PC1,xtyp_pca$price)
```

```
##
##  F test to compare two variances
##
## data:  xtyp_pca$PC1 and xtyp_pca$price
## F = 0.056254, num df = 9305, denom df = 9305, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.05401366 0.05858716
## sample estimates:
## ratio of variances
##         0.05625395
```

```
var.test(xtyp_pca$PC2,xtyp_pca$price)
```

```
##
##  F test to compare two variances
##
## data:  xtyp_pca$PC2 and xtyp_pca$price
## F = 0.048933, num df = 9305, denom df = 9305, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.04698414 0.05096243
## sample estimates:
## ratio of variances
##         0.04893287
```

```
var.test(xtyp_pca$PC3,xtyp_pca$price)
```

```
##
##  F test to compare two variances
##
## data:  xtyp_pca$PC3 and xtyp_pca$price
## F = 0.029933, num df = 9305, denom df = 9305, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.0287411 0.0311747
## sample estimates:
## ratio of variances
##         0.02993317
```

```
var.test(xtyp_pca$PC4,xtyp_pca$price)
```

```
##
##  F test to compare two variances
##
## data:  xtyp_pca$PC4 and xtyp_pca$price
## F = 0.027474, num df = 9305, denom df = 9305, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.02637987 0.02861354
## sample estimates:
```

```
## ratio of variances
##         0.02747402

var.test(xtyp_pca$PC5,xtyp_pca$price)

##
##  F test to compare two variances
##
## data:  xtyp_pca$PC5 and xtyp_pca$price
## F = 0.02655, num df = 9305, denom df = 9305, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.02549285 0.02765141
## sample estimates:
## ratio of variances
##          0.0265502

var.test(xtyp_pca$PC6,xtyp_pca$price)

##
##  F test to compare two variances
##
## data:  xtyp_pca$PC6 and xtyp_pca$price
## F = 0.020181, num df = 9305, denom df = 9305, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.01937776 0.02101853
## sample estimates:
## ratio of variances
##         0.02018148

var.test(xtyp_pca$PC7,xtyp_pca$price)

##
##  F test to compare two variances
##
## data:  xtyp_pca$PC7 and xtyp_pca$price
## F = 0.012768, num df = 9305, denom df = 9305, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.01225948 0.01329753
## sample estimates:
## ratio of variances
##         0.01276796

var.test(xtyp_pca$PC8,xtyp_pca$price)

##
##  F test to compare two variances
##
## data:  xtyp_pca$PC8 and xtyp_pca$price
## F = 0.0068915, num df = 9305, denom df = 9305, p-value < 2.2e-16
```

```
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.006617014 0.007177297
## sample estimates:
## ratio of variances
##        0.006891464
```