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)

merge\_data$rain<-as.numeric(merge\_data$rain)  
merge\_data$rain[is.na(merge\_data$rain)]<-0  
  
for ( i in 1:length(merge\_data$rain)){  
 if(merge\_data$rain[i]>0 & merge\_data$rain[i]<=0.30){  
 merge\_data$rain[i]=1  
 }  
}  
  
for ( i in 1:length(merge\_data$rain)){  
 if(merge\_data$rain[i]>=0.30 & merge\_data$rain[i]!=1){  
 merge\_data$rain[i]=2  
 }  
}  
  
merge\_data$rain = factor(merge\_data$rain,  
 levels = c(0,1,2),  
 labels = c(0,1,2))  
  
merge\_data$location = factor(merge\_data$location,  
 levels = c('Back Bay', 'Beacon Hill'),  
 labels = c(0,1))  
  
#install.packages("dummies")  
library(dummies)

## dummies-1.5.6 provided by Decision Patterns

# example data  
  
merge\_data <- cbind(merge\_data, dummy(merge\_data$rain, sep = "\_"))  
#names(merge\_data$merge\_data\_0)<-("rain\_0")  
#names(merge\_data$merge\_data\_1)<-("rain\_1")  
merge\_data<-merge\_data[-6]  
merge\_data<-merge\_data[-23]  
#View(merge\_data)

# Handling Missing values

#Extracting the numerical columns in a new dataframe "df"  
merge\_data$temp<-merge\_data[,c(2)] #renaming a column  
df<-merge\_data[,c(3,4,5,7,8,9,10,11,21,22,23,15)]  
#View(df)  
df$cab\_type<-factor(merge\_data$cab\_type)  
df<-na.omit(df)

# 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)  
View(df)

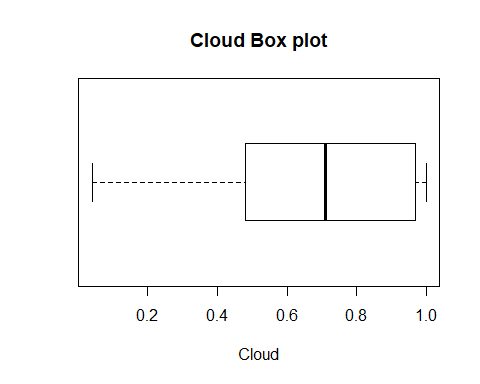
# Creating a Numeric dataframe

x<-df[,c(2,3,4,5,11,12)]  
head(x)

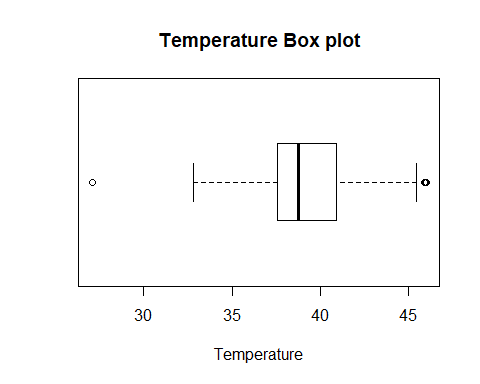
## clouds pressure humidity wind temp price  
## 3 0.86 1014.17 0.93 2.59 40.63 8.5  
## 4 0.86 1014.17 0.93 2.65 40.61 16.5  
## 6 0.95 1013.78 0.92 2.59 40.72 26.5  
## 7 0.95 1013.78 0.92 2.59 40.72 7.5  
## 12 0.92 1013.76 0.92 3.02 40.64 22.5  
## 19 1.00 1014.18 0.91 1.16 40.46 22.5

# BOXPLOT

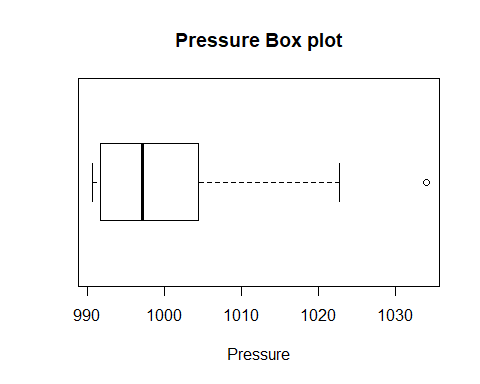
boxplot(x$clouds, main="Cloud Box plot",yaxt="n", xlab="Cloud", horizontal=TRUE)



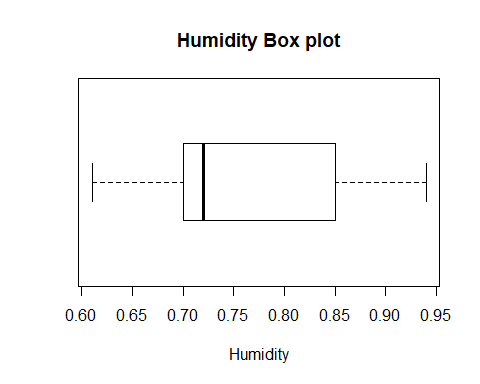
boxplot(x$temp, main="Temperature Box plot",yaxt="n", xlab="Temperature", horizontal=TRUE)



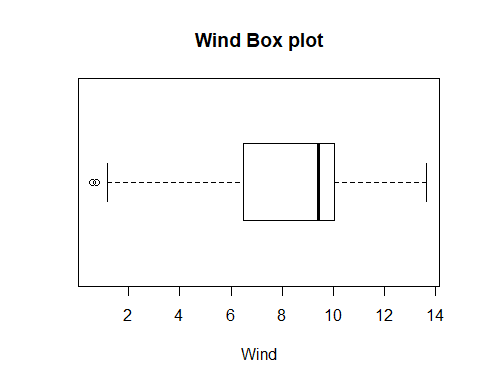
boxplot(x$pressure, main="Pressure Box plot",yaxt="n", xlab="Pressure", horizontal=TRUE)



boxplot(x$humidity, main="Humidity Box plot",yaxt="n", xlab="Humidity", horizontal=TRUE)



boxplot(x$wind, main="Wind Box plot",yaxt="n", xlab="Wind", horizontal=TRUE)



#boxplot(x$distance, main="Wind Box plot",yaxt="n", xlab="Wind", horizontal=TRUE)

#Q-Q Plot to check normality..

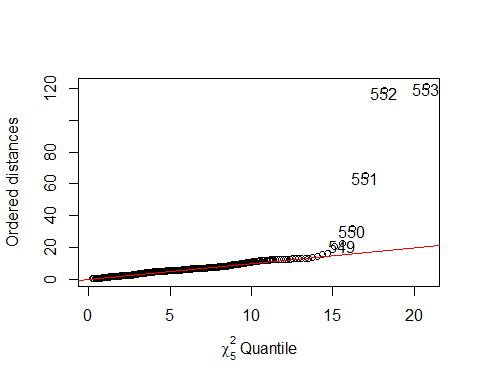
# Deviation from normality can be observed in our variables. Let’s check for multivariate analysis using chi-squre plot

## CORRELATION, COVARIANCE AND DISTANCE

#We are calculating for: clouds, pressure, rain, humidity, wind, distance, temp  
covariance<-cov(x) #variance-covariance matrix created  
correlation<-cor(x) #standardized  
#colmeans  
cm<-colMeans(x)  
distance<-dist(scale(x,center=FALSE))  
#Calculating di(generalized distance for all observations of our data)  
d <- apply(x, MARGIN = 1, function(x) + t(x - cm) %\*% solve(covariance) %\*% (x - cm))

## The sorted distance are now plotted against the appropriate quantiles of the chi-distribution

plot(qc <- qchisq((1:nrow(x) - 1/2) / nrow(x), df = 5), sd <- sort(d),xlab = expression(paste(chi[5]^2, " Quantile")),ylab = "Ordered distances")  
oups <- which(rank(abs(qc - sd), ties = "random") > nrow(x) - 5)  
text(qc[oups], sd[oups] - 1.5,oups)  
abline(a=0,b=1,col="red")



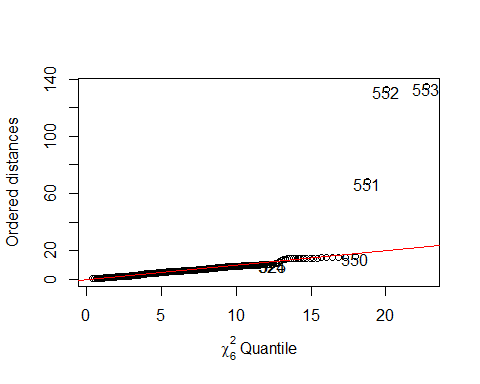
#Our observations seems to deviate from linearity after a certain point

# There is a complete deviation from Normality. We will apply the log transformation on our dataset.

#x\_new<-x+1  
#x\_new=log(x - (min(x) - 1))  
x\_new<-log(x+1)

covariance<-cov(x\_new) #variance-covariance matrix created  
#x\_new$clouds  
correlation<-cor(x\_new) #standardized  
#colmeans  
cm<-colMeans(x\_new)  
distance<-dist(scale(x\_new,center=FALSE))  
#Calculating di(generalized distance for all observations of our data)  
d <- apply(x\_new, MARGIN = 1, function(x\_new) + t(x\_new - cm) %\*% solve(covariance) %\*% (x\_new - cm))

plot(qc <- qchisq((1:nrow(x\_new) - 1/2) / nrow(x\_new), df = 6), sd <- sort(d),xlab = expression(paste(chi[6]^2, " Quantile")),ylab = "Ordered distances")  
oups <- which(rank(abs(qc - sd), ties = "random") > nrow(x) - 6)  
text(qc[oups], sd[oups] - 1.5,oups)  
abline(a=0,b=1,col="red")



# We have normalized the data..

## Pca || T-test || F-test

# Get the Correlations between the measurements

x\_new<-x\_new[-7]  
cor(x\_new)

## clouds pressure humidity wind temp  
## clouds 1.00000000 0.56597486 0.16258638 -0.08549042 0.73863888  
## pressure 0.56597486 1.00000000 0.64972406 -0.54652613 0.54005177  
## humidity 0.16258638 0.64972406 1.00000000 -0.59098133 0.12722028  
## wind -0.08549042 -0.54652613 -0.59098133 1.00000000 0.09654736  
## temp 0.73863888 0.54005177 0.12722028 0.09654736 1.00000000  
## price 0.06790078 0.08269453 0.06668679 -0.06122460 0.04504944  
## price  
## clouds 0.06790078  
## pressure 0.08269453  
## humidity 0.06668679  
## wind -0.06122460  
## temp 0.04504944  
## price 1.00000000

sapply(x\_new, sd, na.rm = TRUE)

## clouds pressure humidity wind temp price   
## 0.190348031 0.008331149 0.055919840 0.473834287 0.087131522 0.499397872

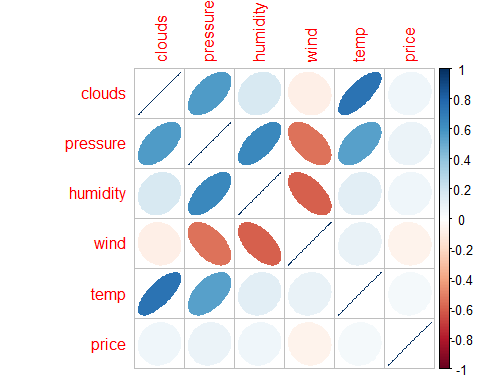
#There are not considerable differences between these standard deviations.. Still let's see the PCAs.

library(corrplot)

## Warning: package 'corrplot' was built under R version 3.5.3

## corrplot 0.84 loaded

corrplot(cor(x\_new), method="ellipse")



# 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\_new,scale=TRUE)  
#x\_pca$rotation  
summary(x\_pca)

## Importance of components:  
## PC1 PC2 PC3 PC4 PC5 PC6  
## Standard deviation 1.6264 1.2426 0.9934 0.64988 0.4887 0.40330  
## Proportion of Variance 0.4409 0.2573 0.1645 0.07039 0.0398 0.02711  
## Cumulative Proportion 0.4409 0.6982 0.8627 0.93309 0.9729 1.00000

#x\_pca$rotation

# Each of these explains a percent of total vaiation in the dataset. PC1 explains 27.8% of total variance, PC2 explains 26% of total variance.. we need to go to PC5 to get a ver accurate view on where it stands in relation to other samples as PC1-PC5 can explain 89.9% of the variance.

# 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

str(x\_pca)

## List of 5  
## $ sdev : num [1:6] 1.626 1.243 0.993 0.65 0.489 ...  
## $ rotation: num [1:6, 1:6] -0.442 -0.574 -0.43 0.348 -0.402 ...  
## ..- attr(\*, "dimnames")=List of 2  
## .. ..$ : chr [1:6] "clouds" "pressure" "humidity" "wind" ...  
## .. ..$ : chr [1:6] "PC1" "PC2" "PC3" "PC4" ...  
## $ center : Named num [1:6] 0.495 6.908 0.568 2.131 3.686 ...  
## ..- attr(\*, "names")= chr [1:6] "clouds" "pressure" "humidity" "wind" ...  
## $ scale : Named num [1:6] 0.19035 0.00833 0.05592 0.47383 0.08713 ...  
## ..- attr(\*, "names")= chr [1:6] "clouds" "pressure" "humidity" "wind" ...  
## $ x : num [1:553, 1:6] -2.76 -2.86 -3 -2.8 -2.85 ...  
## ..- attr(\*, "dimnames")=List of 2  
## .. ..$ : chr [1:553] "3" "4" "6" "7" ...  
## .. ..$ : chr [1:6] "PC1" "PC2" "PC3" "PC4" ...  
## - attr(\*, "class")= chr "prcomp"

eigen\_x <- x\_pca$sdev^2  
names(eigen\_x) <- paste("PC",1:6,sep="")  
eigen\_x

## PC1 PC2 PC3 PC4 PC5 PC6   
## 2.6452195 1.5440576 0.9869193 0.4223476 0.2388064 0.1626497

sumlambdas <- sum(eigen\_x)  
sumlambdas #total sample variance

## [1] 6

propvar <- eigen\_x/sumlambdas  
propvar

## PC1 PC2 PC3 PC4 PC5 PC6   
## 0.44086991 0.25734293 0.16448655 0.07039127 0.03980107 0.02710828

cumvar\_x <- cumsum(propvar)  
cumvar\_x

## PC1 PC2 PC3 PC4 PC5 PC6   
## 0.4408699 0.6982128 0.8626994 0.9330907 0.9728917 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  
## Eigenvalues 2.6452 1.5441 0.9869 0.4223 0.2388 0.1626  
## Prop. variance 0.4409 0.2573 0.1645 0.0704 0.0398 0.0271  
## Cum. prop. variance 0.4409 0.6982 0.8627 0.9331 0.9729 1.0000

# Sample scores stored in x\_pca$x

# We need to calculate the scores on each of these components for each individual in our sample.

#x\_pca$rotation  
xtyp\_pca <- cbind(data.frame(df$price),x\_pca$x)  
str(xtyp\_pca)

## 'data.frame': 553 obs. of 7 variables:  
## $ df.price: num 8.5 16.5 26.5 7.5 22.5 22.5 15.5 16.5 27.5 38.5 ...  
## $ PC1 : num -2.76 -2.86 -3 -2.8 -2.85 ...  
## $ PC2 : num -1.22 -1.23 -1.1 -1.04 -1.01 ...  
## $ PC3 : num -1.22372 -0.00427 0.89835 -1.44242 0.59623 ...  
## $ PC4 : num 0.172 0.134 0.293 0.326 0.122 ...  
## $ PC5 : num 0.14117 0.14754 0.02832 0.00016 0.02977 ...  
## $ PC6 : num -0.0514 -0.065 -0.0366 -0.0418 -0.1333 ...

#xtyp\_pca

# Merging price column

colnames(xtyp\_pca)[colnames(xtyp\_pca)=="df.price"] <- "price"  
str(xtyp\_pca)

## 'data.frame': 553 obs. of 7 variables:  
## $ price: num 8.5 16.5 26.5 7.5 22.5 22.5 15.5 16.5 27.5 38.5 ...  
## $ PC1 : num -2.76 -2.86 -3 -2.8 -2.85 ...  
## $ PC2 : num -1.22 -1.23 -1.1 -1.04 -1.01 ...  
## $ PC3 : num -1.22372 -0.00427 0.89835 -1.44242 0.59623 ...  
## $ PC4 : num 0.172 0.134 0.293 0.326 0.122 ...  
## $ PC5 : num 0.14117 0.14754 0.02832 0.00016 0.02977 ...  
## $ PC6 : num -0.0514 -0.065 -0.0366 -0.0418 -0.1333 ...

# Sample scores stoted. x\_pca$x

# 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 = -41.748, df = 1104, p-value < 2.2e-16  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -17.02935 -15.50049  
## sample estimates:  
## mean of x mean of y   
## -1.520487e-14 1.626492e+01

t.test(xtyp\_pca$PC2,xtyp\_pca$price,var.equal = TRUE)

##   
## Two Sample t-test  
##   
## data: xtyp\_pca$PC2 and xtyp\_pca$price  
## t = -42.025, df = 1104, p-value < 2.2e-16  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -17.02431 -15.50552  
## sample estimates:  
## mean of x mean of y   
## -2.815137e-15 1.626492e+01

t.test(xtyp\_pca$PC3,xtyp\_pca$price,var.equal = TRUE)

##   
## Two Sample t-test  
##   
## data: xtyp\_pca$PC3 and xtyp\_pca$price  
## t = -42.167, df = 1104, p-value < 2.2e-16  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -17.02176 -15.50808  
## sample estimates:  
## mean of x mean of y   
## -1.210148e-15 1.626492e+01

t.test(xtyp\_pca$PC4,xtyp\_pca$price,var.equal = TRUE)

##   
## Two Sample t-test  
##   
## data: xtyp\_pca$PC4 and xtyp\_pca$price  
## t = -42.313, df = 1104, p-value < 2.2e-16  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -17.01916 -15.51068  
## sample estimates:  
## mean of x mean of y   
## -1.244961e-15 1.626492e+01

t.test(xtyp\_pca$PC5,xtyp\_pca$price,var.equal = TRUE)

##   
## Two Sample t-test  
##   
## data: xtyp\_pca$PC5 and xtyp\_pca$price  
## t = -42.36, df = 1104, p-value < 2.2e-16  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -17.01831 -15.51153  
## sample estimates:  
## mean of x mean of y   
## 9.343221e-15 1.626492e+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 = -42.38, df = 1104, p-value < 2.2e-16  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -17.01796 -15.51188  
## sample estimates:  
## mean of x mean of y   
## -2.113320e-14 1.626492e+01

#F-Test #Testing Variation

# Variance Test- Test for variance

var.test(xtyp\_pca$PC1,xtyp\_pca$price)

##   
## F test to compare two variances  
##   
## data: xtyp\_pca$PC1 and xtyp\_pca$price  
## F = 0.03254, num df = 552, denom df = 552, p-value < 2.2e-16  
## alternative hypothesis: true ratio of variances is not equal to 1  
## 95 percent confidence interval:  
## 0.02753513 0.03845520  
## sample estimates:  
## ratio of variances   
## 0.03254027

var.test(xtyp\_pca$PC2,xtyp\_pca$price)

##   
## F test to compare two variances  
##   
## data: xtyp\_pca$PC2 and xtyp\_pca$price  
## F = 0.018994, num df = 552, denom df = 552, p-value < 2.2e-16  
## alternative hypothesis: true ratio of variances is not equal to 1  
## 95 percent confidence interval:  
## 0.01607270 0.02244693  
## sample estimates:  
## ratio of variances   
## 0.01899428

var.test(xtyp\_pca$PC3,xtyp\_pca$price)

##   
## F test to compare two variances  
##   
## data: xtyp\_pca$PC3 and xtyp\_pca$price  
## F = 0.012141, num df = 552, denom df = 552, p-value < 2.2e-16  
## alternative hypothesis: true ratio of variances is not equal to 1  
## 95 percent confidence interval:  
## 0.01027323 0.01434746  
## sample estimates:  
## ratio of variances   
## 0.01214062

var.test(xtyp\_pca$PC4,xtyp\_pca$price)

##   
## F test to compare two variances  
##   
## data: xtyp\_pca$PC4 and xtyp\_pca$price  
## F = 0.0051955, num df = 552, denom df = 552, p-value < 2.2e-16  
## alternative hypothesis: true ratio of variances is not equal to 1  
## 95 percent confidence interval:  
## 0.004396382 0.006139930  
## sample estimates:  
## ratio of variances   
## 0.005195525

var.test(xtyp\_pca$PC5,xtyp\_pca$price)

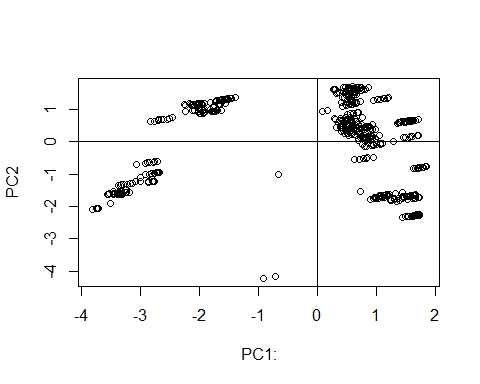
##   
## F test to compare two variances  
##   
## data: xtyp\_pca$PC5 and xtyp\_pca$price  
## F = 0.0029377, num df = 552, denom df = 552, p-value < 2.2e-16  
## alternative hypothesis: true ratio of variances is not equal to 1  
## 95 percent confidence interval:  
## 0.002485830 0.003471678  
## sample estimates:  
## ratio of variances   
## 0.002937686

var.test(xtyp\_pca$PC6,xtyp\_pca$price)

##   
## F test to compare two variances  
##   
## data: xtyp\_pca$PC6 and xtyp\_pca$price  
## F = 0.0020008, num df = 552, denom df = 552, p-value < 2.2e-16  
## alternative hypothesis: true ratio of variances is not equal to 1  
## 95 percent confidence interval:  
## 0.001693084 0.002364539  
## sample estimates:  
## ratio of variances   
## 0.002000841

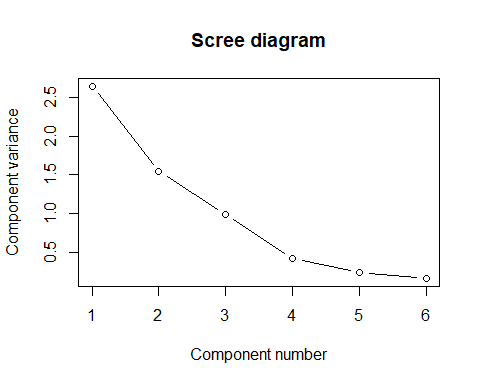
# Plotting the scores of Pricipal Component 1 and Principal component 2

plot(xtyp\_pca$PC1, xtyp\_pca$PC2,xlab="PC1:", ylab="PC2")  
abline(h=0)  
abline(v=0)

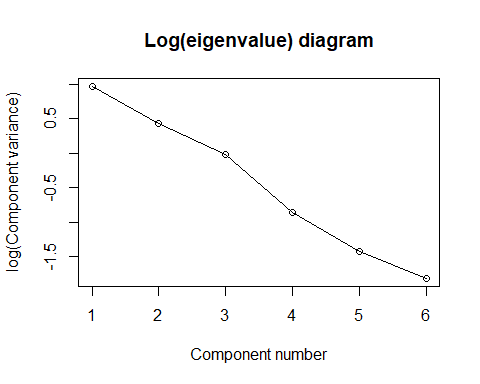


# Plotting the Variance of Principal Components

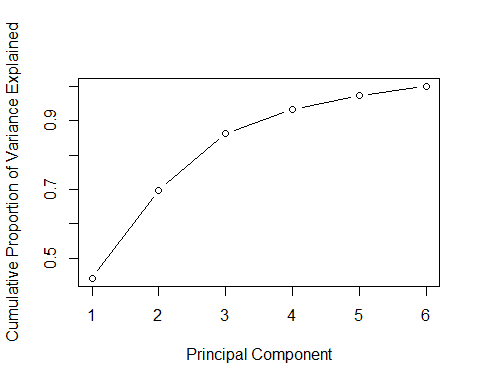
plot(eigen\_x, xlab = "Component number", ylab = "Component variance", type = "b", main = "Scree diagram")

 #Plotting the Log variance of COmponents

plot(log(eigen\_x), xlab = "Component number",ylab = "log(Component variance)", type="o",main = "Log(eigenvalue) diagram")

 #Cumulative scree plot

plot(cumsum(propvar), xlab = "Principal Component",  
 ylab = "Cumulative Proportion of Variance Explained",  
 type = "b")



# Variance of the principal components

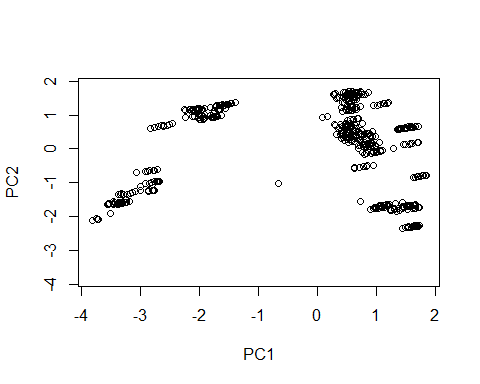
#View(x\_pca)  
diag(cov(x\_pca$x))

## PC1 PC2 PC3 PC4 PC5 PC6   
## 2.6452195 1.5440576 0.9869193 0.4223476 0.2388064 0.1626497

#x\_pca$x[,1]  
#x\_pca$x

# Plotting the scores

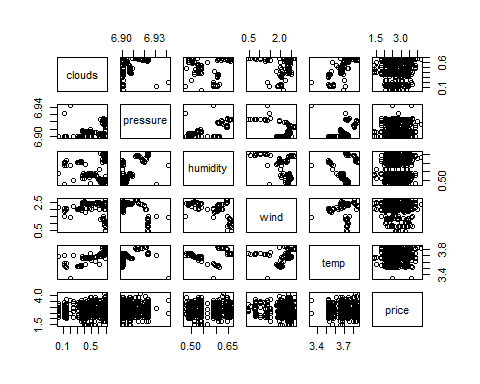
xlim <- range(x\_pca$x[,1])  
plot(x\_pca$x,xlim=xlim,ylim=xlim)



#x\_pca$rotation[,1]  
#x\_pca$rotation

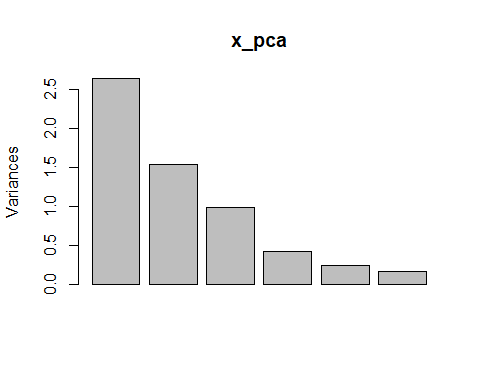
# Scatter plot matrix of the actual data

plot(x\_new)



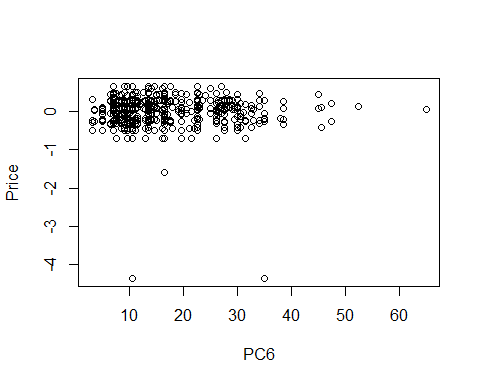
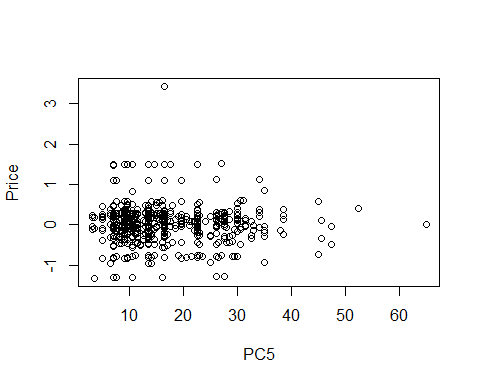
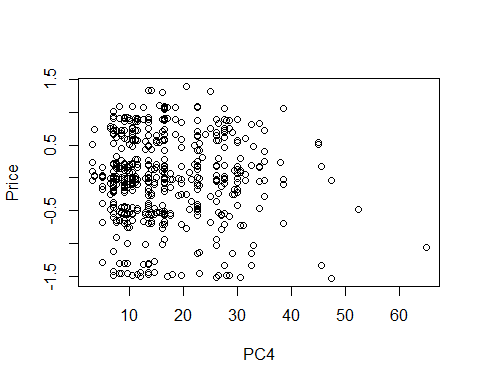
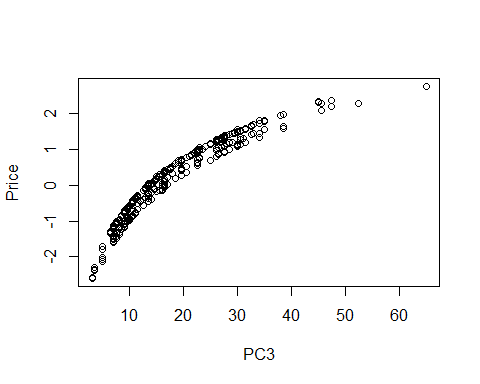
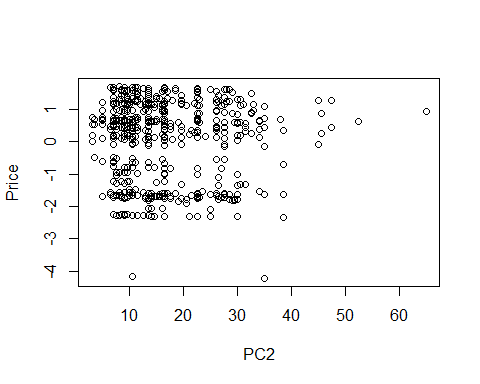
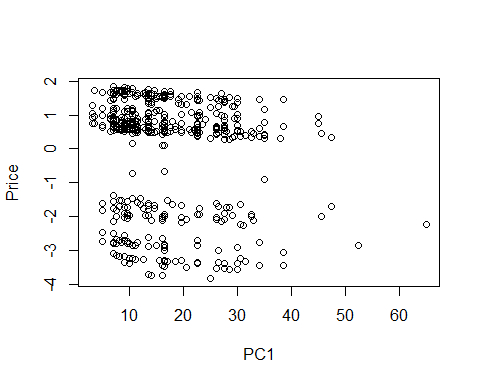
# Variance plot for each component. We can see that all components play a dominant role.

plot(x\_pca)

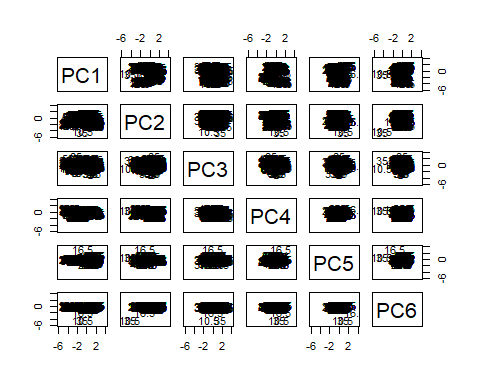
 #Taking first 4 components

xtyp\_pca<-xtyp\_pca[1:5]

#get the original value of the data based on PCA  
center <- x\_pca$center  
scale <- x\_pca$scale  
new\_x <- as.matrix(x\_new)  
#drop(scale(new\_x,center=center, scale=scale)%\*%x\_pca$rotation[,1])  
#predict(x\_pca)[,1]  
#The aboved two gives us the same thing. predict is a good function to know.  
  
x\_new$price<-df$price  
out <- sapply(1:6, function(i){plot(x\_new$price,x\_pca$x[,i],xlab=paste("PC",i,sep=""),  
 ylab="Price")})



pairs(x\_pca$x[,1:6], ylim = c(-6,4),xlim = c(-6,4),panel=function(x,y,...){text(x,y,x\_new$price)})



## Factor Analysis

library(psych)

## Warning: package 'psych' was built under R version 3.5.3

#install.packages("psych", lib="/Library/Frameworks/R.framework/Versions/3.5/Resources/library")  
fit.pc <- principal(x\_new, nfactors=4, rotate="varimax")  
fit.pc

## Principal Components Analysis  
## Call: principal(r = x\_new, nfactors = 4, rotate = "varimax")  
## Standardized loadings (pattern matrix) based upon correlation matrix  
## RC1 RC4 RC2 RC3 h2 u2 com  
## clouds 0.93 0.00 -0.15 0.04 0.89 0.11026 1.1  
## pressure 0.60 0.59 -0.41 0.04 0.88 0.11847 2.8  
## humidity 0.07 0.93 -0.28 0.04 0.96 0.04317 1.2  
## wind 0.02 -0.36 0.92 -0.02 0.97 0.02998 1.3  
## temp 0.91 0.16 0.20 0.01 0.90 0.09918 1.2  
## price 0.04 0.03 -0.02 1.00 1.00 0.00011 1.0  
##   
## RC1 RC4 RC2 RC3  
## SS loadings 2.07 1.37 1.15 1.00  
## Proportion Var 0.34 0.23 0.19 0.17  
## Cumulative Var 0.34 0.57 0.77 0.93  
## Proportion Explained 0.37 0.25 0.21 0.18  
## Cumulative Proportion 0.37 0.61 0.82 1.00  
##   
## Mean item complexity = 1.4  
## Test of the hypothesis that 4 components are sufficient.  
##   
## The root mean square of the residuals (RMSR) is 0.04   
## with the empirical chi square 24.65 with prob < NA   
##   
## Fit based upon off diagonal values = 0.99

round(fit.pc$values, 3)

## [1] 2.647 1.544 0.985 0.422 0.239 0.163

fit.pc$loadings

##   
## Loadings:  
## RC1 RC4 RC2 RC3   
## clouds 0.931 -0.148   
## pressure 0.601 0.592 -0.411   
## humidity 0.933 -0.283   
## wind -0.357 0.918   
## temp 0.914 0.158 0.202   
## price 0.998  
##   
## RC1 RC4 RC2 RC3  
## SS loadings 2.068 1.374 1.154 1.002  
## Proportion Var 0.345 0.229 0.192 0.167  
## Cumulative Var 0.345 0.574 0.766 0.933

# The first 4 factors have an Eigenvalue >1 and which explains almost 88% of the variance. We can effectively reduce dimensionality from 6 to 4 while only losing about 11% of the variance.

# Communalities

fit.pc$communality

## clouds pressure humidity wind temp price   
## 0.8897446 0.8815262 0.9568320 0.9700165 0.9008220 0.9998854

sum(fit.pc$communality)

## [1] 5.598827

# The variance in clouds accounted by all fators is 0.87, This is the extent to which an item correlates with all other items. All comunalities are high, which means that the extracted components represent the variables well. If they are low , you may need to extract another component.

# Rotated factor scores, Notice the columns ordering: RC1, RC3, RC2 and RC4  
fit.pc$scores

## RC1 RC4 RC2 RC3  
## 3 0.619089220 1.32061241 -1.479355625 -0.958518169  
## 4 0.590941419 1.30915291 -1.430982396 -0.067642265  
## 6 0.709488130 1.06974017 -1.549040428 1.046816282  
## 7 0.772255638 1.12287429 -1.582772087 -1.067715746  
## 12 0.674527040 1.18935514 -1.291654864 0.604741073  
## 19 0.767537852 0.57782494 -2.774865419 0.593506606  
## 20 0.790662723 0.59740067 -2.787292872 -0.185531509  
## 22 0.788768777 0.59827196 -2.774485276 -0.074228005  
## 24 0.752817381 0.57399486 -2.734972041 1.150178044  
## 25 0.716478298 0.54323300 -2.715443186 2.374380797  
## 26 0.808977783 0.62153592 -2.765152999 -0.741771665  
## 27 0.769335147 0.58797752 -2.743848793 0.593722247  
## 28 0.789156465 0.60475672 -2.754500896 -0.074024709  
## 37 0.868450657 0.95109656 -2.119531512 -1.133666263  
## 41 0.818444816 0.82954131 -2.483065594 -0.691137276  
## 42 0.742463096 0.76522105 -2.442232533 1.868559389  
## 43 0.821748369 0.83233784 -2.484840944 -0.802428436  
## 48 0.787366704 0.81918367 -2.569283695 0.975875059  
## 49 0.840223553 0.86392820 -2.597689303 -0.804783491  
## 50 0.843527106 0.86672473 -2.599464653 -0.916074650  
## 51 0.820402235 0.84714900 -2.587037200 -0.137036535  
## 52 0.820402235 0.84714900 -2.587037200 -0.137036535  
## 54 0.779107822 0.81219234 -2.564845319 1.254102957  
## 55 0.780052779 0.68722004 -3.324064343 -0.145727701  
## 56 0.776775431 0.97020556 -2.060687677 1.426734913  
## 60 0.684555594 1.20585924 -1.627812219 -0.065569467  
## 62 0.930265047 1.29951893 -1.054147999 -1.129706691  
## 63 0.826203126 1.21142815 -0.998224459 2.375964829  
## 64 0.918702612 1.28973107 -1.047934272 -0.740187633  
## 75 1.301449854 1.06978427 0.204160433 3.928010287  
## 76 1.445154412 1.19143345 0.126932687 -0.913155145  
## 77 1.362565585 1.12152013 0.171316449 1.869123839  
## 78 1.400556445 1.15368025 0.150899919 0.589275506  
## 79 1.430288423 1.17884905 0.134921764 -0.412344928  
## 80 1.509710885 1.09872462 0.610901492 -1.128102612  
## 81 1.425132383 1.06317980 0.778350002 0.265208112  
## 82 1.387141523 1.03101967 0.798766532 1.545056445  
## 85 1.390141002 1.04877934 0.790625278 -0.889774478  
## 86 1.365364354 1.02780534 0.803940407 -0.055090783  
## 87 1.375275013 1.03619494 0.798614355 -0.388964261  
## 88 1.329025270 0.99704348 0.823469262 1.169111970  
## 89 1.403355214 1.05996547 0.783523876 -1.334939115  
## 90 1.355453695 1.01941574 0.809266458 0.278782695  
## 96 1.214128418 1.44296600 1.052510252 -0.834296876  
## 97 1.136494922 1.37724747 1.094230988 1.781045368  
## 98 1.192655323 1.42478853 1.064050030 -0.110904341  
## 99 1.030781224 1.28775842 1.151042202 5.342362467  
## 100 1.214128418 1.44296600 1.052510252 -0.834296876  
## 106 0.955876493 1.65305872 1.435289424 1.734631984  
## 107 0.972394258 1.66704138 1.426412672 1.178176187  
## 108 1.018644001 1.70619284 1.401557766 -0.379900044  
## 109 0.912930303 1.61670379 1.458368981 3.181417055  
## 111 1.035270515 1.72554258 1.407740925 -0.991679994  
## 112 1.030315186 1.72134778 1.410403951 -0.824743255  
## 113 1.031966962 1.72274604 1.409516276 -0.880388835  
## 114 1.020404527 1.71295818 1.415730002 -0.490869777  
## 115 1.017100974 1.71016165 1.417505353 -0.379578618  
## 116 1.045181175 1.73393217 1.402414874 -1.325553472  
## 117 1.030315186 1.72134778 1.410403951 -0.824743255  
## 118 0.954333465 1.65702752 1.451237012 1.734953410  
## 120 1.038388090 1.51949428 1.188269400 0.675121026  
## 125 1.058802001 1.71079292 1.688519170 -0.264666459  
## 126 1.030721800 1.68702239 1.703609649 0.681308396  
## 127 1.083578649 1.73176692 1.675204042 -1.099350154  
## 129 1.067060884 1.71778425 1.684080794 -0.542894357  
## 130 1.058802001 1.71079292 1.688519170 -0.264666459  
## 131 1.062105554 1.71358945 1.686743820 -0.375957618  
## 132 1.062019935 1.64356379 1.877393208 0.018359561  
## 133 0.961261566 1.55826954 1.931541397 3.412739921  
## 134 1.083493030 1.66174126 1.865853430 -0.705032975  
## 135 1.078537700 1.65754646 1.868516456 -0.538096236  
## 136 1.095055465 1.67152912 1.859639703 -1.094552033  
## 137 1.032287957 1.61839500 1.893371362 1.019979995  
## 138 1.017421968 1.60581060 1.901360439 1.520790212  
## 184 -1.307202957 0.28260754 0.669618996 -0.623972624  
## 185 -1.295640521 0.29239540 0.663405269 -1.013491682  
## 186 -1.320417169 0.27142141 0.676720397 -0.178807987  
## 188 -1.295640521 0.29239540 0.663405269 -1.013491682  
## 197 -1.739110780 0.57271197 0.193547498 0.737394839  
## 198 -1.702771696 0.60347383 0.174018643 -0.486807914  
## 199 -1.686253931 0.61745649 0.165141891 -1.043263710  
## 200 -1.752324992 0.56152583 0.200648900 1.182559477  
## 201 -1.719289462 0.58949116 0.182895396 0.069647883  
## 203 -1.717637685 0.59088943 0.182007720 0.014002304  
## 205 -1.735997738 0.61965148 0.208782894 -0.208402343  
## 207 -1.742604844 0.61405841 0.212333595 0.014179975  
## 208 -1.717828197 0.63503241 0.199018467 -0.820503720  
## 209 -1.778943928 0.58329655 0.231862451 1.238382728  
## 210 -1.716176420 0.63643067 0.198130792 -0.876149299  
## 211 -1.747560174 0.60986361 0.214996621 0.181116714  
## 212 -1.742604844 0.61405841 0.212333595 0.014179975  
## 213 -1.716176420 0.63643067 0.198130792 -0.876149299  
## 215 -1.778943928 0.58329655 0.231862451 1.238382728  
## 216 -1.722783526 0.63083761 0.201681493 -0.653566981  
## 217 -1.717828197 0.63503241 0.199018467 -0.820503720  
## 226 -1.152931561 -0.11306710 -0.835888339 0.854421760  
## 228 -1.454335162 0.31095966 -0.380584946 0.804476315  
## 229 -1.413040748 0.34591632 -0.402776827 -0.586663177  
## 230 -1.426254961 0.33473019 -0.395675425 -0.141498539  
## 232 -1.452683385 0.31235792 -0.381472621 0.748830735  
## 233 -1.472708581 0.30530009 -0.356321581 1.250272866  
## 235 -1.408289296 0.35983249 -0.390940915 -0.919904741  
## 236 -1.408289296 0.35983249 -0.390940915 -0.919904741  
## 238 -1.507237140 0.52927944 -0.075050578 0.638707532  
## 244 -2.070710413 0.76131246 -0.135781176 -0.477273168  
## 245 -2.080621072 0.75292287 -0.130455124 -0.143399690  
## 249 -2.058155476 0.75027788 -0.128251330 -0.475932015  
## 250 -2.043289487 0.76286228 -0.136240407 -0.976742232  
## 251 -2.121349335 0.95827483 0.080087606 -0.359772601  
## 252 -2.207241715 0.88556498 0.126246718 2.533797542  
## 253 -2.104831570 0.97225750 0.071210853 -0.916228398  
## 255 -2.104831570 0.97225750 0.071210853 -0.916228398  
## 256 -1.645133392 0.35724964 -0.192122616 1.203806214  
## 257 -1.580714108 0.41178203 -0.226741950 -0.966371393  
## 259 -1.521293609 0.39107076 -0.110402088 -0.185565328  
## 260 -1.519641832 0.39246903 -0.111289763 -0.241210907  
## 261 -1.521293609 0.39107076 -0.110402088 -0.185565328  
## 262 -1.496516961 0.41204476 -0.123717217 -1.020249023  
## 263 -1.590668223 0.33234357 -0.073119728 2.151549019  
## 264 -1.557632693 0.36030890 -0.090873233 1.038637425  
## 265 -1.557632693 0.36030890 -0.090873233 1.038637425  
## 268 -0.414471223 -1.01069821 -0.347531861 3.252048649  
## 269 -0.302150419 -0.91561609 -0.407893777 -0.531850769  
## 270 -0.315590727 -1.08778376 -0.632060340 0.070708029  
## 271 -0.305680068 -1.07939416 -0.637386391 -0.263165449  
## 272 -0.376706459 -1.13951962 -0.599216356 2.129594477  
## 273 -0.351929811 -1.11854562 -0.612531485 1.294910782  
## 274 -0.295769409 -1.07100456 -0.642712443 -0.597038927  
## 275 -0.284206973 -1.06121670 -0.648926169 -0.986557984  
## 276 -0.284206973 -1.06121670 -0.648926169 -0.986557984  
## 277 -0.295769409 -1.07100456 -0.642712443 -0.597038927  
## 279 -0.370709039 -0.84536588 -0.295040410 -0.603177019  
## 280 -0.380619698 -0.85375548 -0.289714359 -0.269303541  
## 281 -0.400441016 -0.87053467 -0.279062256 0.398443415  
## 282 -0.426869440 -0.89290694 -0.264859452 1.288772690  
## 283 -0.345932391 -0.82439188 -0.308355539 -1.437860714  
## 284 -0.281072815 -0.76392213 0.646821442 -0.541061971  
## 285 -0.335581441 -0.81006492 0.676114725 1.295242159  
## 286 -0.274465709 -0.75832906 0.643270741 -0.763644289  
## 287 -0.319063675 -0.79608226 0.667237972 0.738786362  
## 289 -0.303176434 -0.76078424 0.659192442 -0.096860834  
## 290 -0.278399786 -0.73981024 0.645877313 -0.931544529  
## 291 -0.329604858 -0.78315650 0.673395246 0.793468441  
## 292 -0.278399786 -0.73981024 0.645877313 -0.931544529  
## 294 -0.308543842 -0.68069862 0.716830383 -0.269431841  
## 295 -0.565262871 -0.82516342 0.284621623 0.054914535  
## 296 -0.623075050 -0.87410274 0.315690256 2.002509824  
## 297 -0.555352212 -0.81677382 0.279295571 -0.278958943  
## 298 -0.533879117 -0.79859635 0.267755793 -1.002351479  
## 299 -0.596646625 -0.85173048 0.301487452 1.112180549  
## 300 -0.535530894 -0.79999462 0.268643469 -0.946705899  
## 303 0.362882555 -1.46162368 0.299179596 -0.059370987  
## 305 0.338105907 -1.48259768 0.312494725 0.775312708  
## 306 0.364534331 -1.46022542 0.298291921 -0.115016566  
## 307 0.265427740 -1.54412140 0.351552435 3.223718214  
## 310 0.873772876 -1.73679927 -0.086312820 -1.007970860  
## 311 0.835782015 -1.76895940 -0.065896290 0.271877473  
## 312 0.855603334 -1.75218020 -0.076548393 -0.395869483  
## 313 0.852299781 -1.75497674 -0.074773042 -0.284578324  
## 314 0.860558663 -1.74798541 -0.079211418 -0.562806222  
## 315 0.858906887 -1.74938367 -0.078323743 -0.507160643  
## 316 0.852299781 -1.75497674 -0.074773042 -0.284578324  
## 318 0.822567803 -1.78014553 -0.058794888 0.717042110  
## 319 0.804398261 -1.79552646 -0.049030461 1.329143487  
## 321 0.778951735 -1.60629962 0.112292622 -1.062950499  
## 325 0.936514705 -1.69993211 -0.090353154 -0.895622470  
## 326 0.906091293 -1.71896413 -0.070136046 0.050595095  
## 327 0.841875065 -1.45515752 0.021800255 -0.783420813  
## 328 0.833616182 -1.46214885 0.026238631 -0.505192914  
## 329 0.797277098 -1.49291071 0.045767486 0.719009839  
## 330 0.833616182 -1.46214885 0.026238631 -0.505192914  
## 336 0.689836654 -1.36814000 0.066312880 -0.119570714  
## 337 0.694791984 -1.36394520 0.063649854 -0.286507453  
## 338 0.665060006 -1.38911399 0.079628008 0.715112981  
## 339 0.704702643 -1.35555560 0.058323803 -0.620380931  
## 340 0.648542241 -1.40309666 0.088504761 1.271568778  
## 341 0.743912817 -1.62506517 -0.126149161 -0.286219050  
## 342 0.758778806 -1.61248077 -0.134138238 -0.787029267  
## 343 0.743912817 -1.62506517 -0.126149161 -0.286219050  
## 344 0.724091499 -1.64184437 -0.115497059 0.381527906  
## 345 0.700966628 -1.66142010 -0.103069605 1.160566021  
## 350 0.873875635 -1.69726465 -0.169268591 -0.723951368  
## 351 0.849098988 -1.71823864 -0.155953462 0.110732327  
## 352 0.860661423 -1.70845078 -0.162167189 -0.278786730  
## 353 0.875527412 -1.69586638 -0.170156266 -0.779596947  
## 354 0.882134518 -1.69027332 -0.173706967 -1.002179266  
## 357 0.597883623 -1.25728076 0.059740797 0.053183288  
## 358 0.566499869 -1.28384783 0.076606627 1.110449302  
## 359 0.616053165 -1.24189983 0.049976370 -0.558918089  
## 360 0.596270716 -1.22658935 0.072139342 -0.502910114  
## 363 0.449598726 -1.22535253 0.180956487 1.891318659  
## 364 0.485937810 -1.19459067 0.161427632 0.667115906  
## 365 0.514018011 -1.17082014 0.146337153 -0.278858948  
## 366 0.525580446 -1.16103228 0.140123426 -0.668378006  
## 367 0.525580446 -1.16103228 0.140123426 -0.668378006  
## 369 0.300711436 -0.74080994 0.316901002 -0.780498771  
## 373 0.296286132 -0.71302491 0.140120169 -0.508577186  
## 374 0.218652635 -0.77874343 0.181840905 2.106765058  
## 375 0.295335034 -0.70827447 0.145746076 -0.563934470  
## 376 0.239984913 -0.72578599 0.181135475 0.716045042  
## 377 0.200531200 -0.64183723 0.221586781 -0.118729995  
## 378 0.226959625 -0.61946497 0.207383977 -1.009059270  
## 379 0.195575871 -0.64603203 0.224249807 0.048206744  
## 380 0.225307848 -0.62086323 0.208271653 -0.953413690  
## 381 0.215397189 -0.62925283 0.213597704 -0.619540212  
## 382 0.215397189 -0.62925283 0.213597704 -0.619540212  
## 383 0.240173837 -0.60827884 0.200282575 -1.454223907  
## 384 0.215397189 -0.62925283 0.213597704 -0.619540212  
## 386 0.212093636 -0.63204936 0.215373054 -0.508249053  
## 388 0.164192117 -0.67259909 0.241115636 1.105472758  
## 389 0.137763692 -0.69497135 0.255318440 1.995802033  
## 390 0.217048966 -0.62785457 0.212710029 -0.675185792  
## 391 0.050675101 -0.52476749 0.351658713 1.716692728  
## 392 0.120049715 -0.46604030 0.314376354 -0.620421618  
## 393 0.121701491 -0.46464204 0.313488678 -0.676067198  
## 394 0.096924843 -0.48561603 0.326803807 0.158616497  
## 398 0.071527486 -0.31693536 0.347908553 -1.399917455  
## 399 0.059965050 -0.32672323 0.354122280 -1.010398397  
## 400 0.038491955 -0.34490069 0.365662058 -0.287005862  
## 401 0.048402614 -0.33651109 0.360336007 -0.620879340  
## 402 0.053357944 -0.33231629 0.357672981 -0.787816079  
## 403 -0.032534436 -0.40502615 0.403832093 2.105754064  
## 404 -0.073828849 -0.43998281 0.426023974 3.496893556  
## 405 -0.002802458 -0.37985735 0.387853939 1.104133630  
## 406 0.066572156 -0.32113016 0.350571579 -1.232980716  
## 407 0.028581296 -0.35329029 0.370988109 0.046867616  
## 410 -0.007757788 -0.38405215 0.390516965 1.271070369  
## 411 0.059965050 -0.32672323 0.354122280 -1.010398397  
## 412 0.048402614 -0.33651109 0.360336007 -0.620879340  
## 413 0.038491955 -0.34490069 0.365662058 -0.287005862  
## 414 0.028581296 -0.35329029 0.370988109 0.046867616  
## 415 0.018670637 -0.36167989 0.376314161 0.380741094  
## 416 -0.020972000 -0.39523828 0.397618366 1.716235007  
## 418 0.011379161 -0.43539974 0.301932941 -1.454677689  
## 419 -0.064602559 -0.49971999 0.342766001 1.105018976  
## 420 -0.087058841 -0.48462066 0.370162822 1.272041938  
## 421 -0.060622203 -0.38908472 0.403682743 -0.731039114  
## 422 -0.077905680 -0.37087067 0.414674696 -0.842159619  
## 423 -0.084512786 -0.37646374 0.418225397 -0.619577300  
## 424 -0.086164562 -0.37786200 0.419113073 -0.563931720  
## 425 -0.087816339 -0.37926027 0.420000748 -0.508286141  
## 426 -0.135717858 -0.41981000 0.445743330 1.105435670  
## 427 -0.072950350 -0.36667587 0.412011671 -1.009096358  
## 429 -0.077905680 -0.37087067 0.414674696 -0.842159619  
## 430 -0.162146282 -0.44218226 0.459946133 1.995764945  
## 431 -0.071298573 -0.36527761 0.411123995 -1.064741937  
## 433 -0.074602126 -0.36807414 0.412899346 -0.953450778  
## 435 -0.236694783 -0.40867657 0.508324694 2.496877271  
## 443 -0.167937750 -0.64484913 0.125762557 3.280292241  
## 444 -0.091956030 -0.58052887 0.084929497 0.720595576  
## 445 -0.124991560 -0.60849420 0.102683001 1.833507169  
## 446 -0.091956030 -0.58052887 0.084929497 0.720595576  
## 447 -0.062224052 -0.55536007 0.068951342 -0.281024858  
## 448 -0.052313393 -0.54697047 0.063625291 -0.614898336  
## 449 -0.108473795 -0.59451153 0.093806249 1.277051373  
## 450 -0.190093147 -0.44526568 0.199978363 1.553845742  
## 453 -0.315245032 -0.27863505 0.356904209 0.720054209  
## 454 -0.264039959 -0.23528879 0.329386276 -1.004958761  
## 455 -0.295423713 -0.26185586 0.346252106 0.052307253  
## 456 -0.295423713 -0.26185586 0.346252106 0.052307253  
## 457 -0.275602395 -0.24507666 0.335600003 -0.615439703  
## 462 -0.278125991 -0.23403608 0.407074366 -0.730500429  
## 466 -0.061905125 -0.32468146 0.325217134 0.605764688  
## 467 -0.045387360 -0.31069879 0.316340382 0.049308891  
## 468 -0.094940655 -0.35264679 0.342970639 1.718676282  
## 469 -0.033824924 -0.30091093 0.310126655 -0.340210166  
## 471 0.603271611 -0.99448440 -0.177492491 -0.003008406  
## 640 -1.808384506 0.99841391 -2.945754003 -0.618783541  
## 641 -1.889321556 0.92989886 -2.902257917 2.107849863  
## 749 0.614458041 1.31544279 -1.484623525 -0.847199160  
## 750 0.614458041 1.31544279 -1.484623525 -0.847199160  
## 751 0.596288499 1.30006186 -1.474859098 -0.235097783  
## 752 0.612806264 1.31404452 -1.483735850 -0.791553580  
## 758 0.733149990 1.21098645 -1.340473420 -0.897794872  
## 759 0.736453543 1.21378298 -1.342248771 -1.009086031  
## 760 0.731498213 1.20958818 -1.339585745 -0.842149292  
## 761 0.708373342 1.19001245 -1.327158292 -0.063111177  
## 762 0.663775375 1.15225926 -1.303191060 1.439319474  
## 763 0.738105319 1.21518125 -1.343136446 -1.064731611  
## 764 0.724891107 1.20399512 -1.336035044 -0.619566974  
## 770 0.755522704 0.41894702 -3.281082823 0.366554816  
## 773 0.781334938 0.59876066 -2.751921074 0.148647617  
## 774 0.768120726 0.58757453 -2.744819672 0.593812254  
## 775 0.786290268 0.60295546 -2.754584099 -0.018289123  
## 777 0.753201545 0.58042067 -2.715169340 1.150379492  
## 779 0.802754841 0.62236866 -2.741799597 -0.518987898  
## 780 0.817620829 0.63495306 -2.749788674 -1.019798115  
## 781 0.789540628 0.61118253 -2.734698195 -0.073823261  
## 783 0.789731408 0.61437366 -2.724863947 -0.073723220  
## 792 0.770342480 0.89478510 -2.062975736 1.593229159  
## 801 0.815968381 0.83864168 -2.607521117 -0.081410576  
## 802 0.779629298 0.80787982 -2.587992262 1.142792176  
## 803 0.771370415 0.80088849 -2.583553886 1.421020075  
## 805 0.815968381 0.83864168 -2.607521117 -0.081410576  
## 806 0.784160572 0.68400814 -3.359140167 -0.423970125  
## 808 0.782508795 0.68260988 -3.358252492 -0.368324545  
## 809 0.746169712 0.65184802 -3.338723637 0.855878208  
## 810 0.781258270 0.98007933 -2.065023453 1.037829454  
## 811 0.839070449 1.02901866 -2.096092086 -0.909765834  
## 813 0.844025778 1.03321346 -2.098755112 -1.076702573  
## 814 0.804383142 0.99965506 -2.077450906 0.258791339  
## 815 0.844025778 1.03321346 -2.098755112 -1.076702573  
## 817 0.656229280 1.20020795 -1.707788539 1.489061755  
## 821 0.896898327 1.27845221 -1.019805130 -0.072068899  
## 822 0.934889187 1.31061234 -1.040221660 -1.351917232  
## 823 0.915067868 1.29383314 -1.029569557 -0.684170276  
## 824 0.928282081 1.30501927 -1.036670959 -1.129334913  
## 825 0.906808986 1.28684181 -1.025131181 -0.405942377  
## 833 1.478777552 1.16371795 0.094383013 -1.359601206  
## 834 1.450697351 1.13994742 0.109473492 -0.413626352  
## 835 1.472170446 1.15812489 0.097933714 -1.137018888  
## 836 1.442438469 1.13295609 0.113911868 -0.135398453  
## 837 1.420965374 1.11477863 0.125451646 0.587994082  
## 842 1.443006674 1.10398279 0.754936068 -1.069860137  
## 843 1.438051345 1.09978799 0.757599093 -0.902923398  
## 844 1.421533579 1.08580532 0.766475846 -0.346467602  
## 845 1.414926473 1.08021226 0.770026547 -0.123885283  
## 846 1.436399568 1.09838972 0.758486769 -0.847277819  
## 848 1.367024954 1.03966253 0.795769128 1.489836528  
## 851 1.331496682 1.11546296 0.856023597 0.388792392  
## 852 1.346362671 1.12804736 0.848034519 -0.112017825  
## 853 1.367835766 1.14622482 0.836494741 -0.835410361  
## 854 1.366183989 1.14482655 0.837382417 -0.779764781  
## 856 1.310023587 1.09728549 0.867563375 1.112184928  
## 857 1.136265556 1.29224115 0.997308725 1.001683870  
## 858 1.197381288 1.34397701 0.964464742 -1.057202578  
## 859 1.197381288 1.34397701 0.964464742 -1.057202578  
## 860 1.197381288 1.34397701 0.964464742 -1.057202578  
## 861 1.167649311 1.31880821 0.980442896 -0.055582144  
## 865 1.160150128 1.39813557 1.083541933 1.001980495  
## 866 1.203096317 1.43449050 1.060462376 -0.444804577  
## 876 1.019251795 1.50660010 1.208583675 1.342921744  
## 877 1.007689359 1.49681224 1.214797402 1.732440802  
## 878 1.091929962 1.56812383 1.169525965 -1.105483762  
## 879 1.040724890 1.52477757 1.197043897 0.619529208  
## 882 1.079683790 1.59808357 1.432382748 -0.267495257  
## 884 1.116330355 1.69250169 1.787121071 -0.930890560  
## 885 1.056866400 1.64216410 1.819077379 1.072350309  
## 886 1.113026802 1.68970516 1.788896421 -0.819599400  
## 887 1.084946601 1.66593463 1.803986900 0.126375454  
## 888 1.099812590 1.67851903 1.795997823 -0.374434763  
## 889 1.117982132 1.69389996 1.786233395 -0.986536139  
## 890 1.106419696 1.68411209 1.792447122 -0.597017082  
## 891 1.079991272 1.66173983 1.806649926 0.293312193  
## 892 1.050259294 1.63657103 1.822628080 1.294932627  
## 893 1.106419696 1.68411209 1.792447122 -0.597017082  
## 894 1.051911071 1.63796930 1.821740405 1.239287048  
## 941 -1.334160498 0.26478257 0.689457483 0.155474587  
## 942 -1.311035626 0.28435830 0.677030030 -0.623563529  
## 943 -1.299473191 0.29414616 0.670816303 -1.013082586  
## 944 -1.365544252 0.23821550 0.706323313 1.212740601  
## 945 -1.330856945 0.26757910 0.687682133 0.044183427  
## 946 -1.350678263 0.25079990 0.698334236 0.711930383  
## 947 -1.306080297 0.28855310 0.674367004 -0.790500268  
## 948 -1.307732073 0.28715483 0.675254680 -0.734854688  
## 949 -1.320946285 0.27596870 0.682356081 -0.289690051  
## 950 -1.307732073 0.28715483 0.675254680 -0.734854688  
## 959 -1.741618242 0.61201200 0.203173837 -0.041504588  
## 960 -1.739966465 0.61341027 0.202286162 -0.097150167  
## 961 -1.733359359 0.61900333 0.198735461 -0.319732486  
## 962 -1.781260878 0.57845360 0.224478042 1.293989324  
## 963 -1.746098649 0.61246653 0.215280373 0.070006730  
## 964 -1.808537708 0.64895914 0.227740380 -0.317681602  
## 965 -1.823403697 0.63637474 0.235729457 0.183128615  
## 966 -1.859742780 0.60561288 0.255258312 1.407331368  
## 967 -1.795323496 0.66014527 0.220638978 -0.762846239  
## 969 -1.793671719 0.66154354 0.219751303 -0.818491819  
## 970 -1.820100144 0.63917127 0.233954107 0.071837456  
## 971 -1.811841261 0.64616261 0.229515731 -0.206390443  
## 972 -1.796975272 0.65874700 0.221526653 -0.707200660  
## 982 -1.433376628 0.44407172 -0.375616393 -0.699200878  
## 983 -1.477721246 0.40825900 -0.350540080 0.747724571  
## 985 -1.433123280 0.44601220 -0.374507311 -0.754706080  
## 986 -1.419909067 0.45719833 -0.381608713 -1.199870717  
## 987 -1.447989268 0.43342780 -0.366518234 -0.253895863  
## 988 -1.457899927 0.42503820 -0.361192183 0.079977615  
## 990 -1.426516173 0.45160526 -0.378058012 -0.977288399  
## 991 -1.434775056 0.44461393 -0.373619636 -0.699060500  
## 992 -1.500846117 0.38868327 -0.338112626 1.526762687  
## 993 -1.502497894 0.38728501 -0.337224951 1.582408266  
## 994 -1.422769297 0.35067440 -0.385833670 -0.585650605  
## 995 -1.411206861 0.36046226 -0.392047396 -0.975169663  
## 997 -1.525166122 0.51721649 -0.063630722 1.139815602  
## 998 -2.108714298 0.61970199 -0.170608074 1.580128316  
## 999 -2.083937650 0.64067599 -0.183923203 0.745444621  
## 1000 -2.032732578 0.68402225 -0.211441135 -0.979568349  
## 1001 -2.054205672 0.66584479 -0.199901357 -0.256175814  
## 1005 -2.073250314 0.76216991 -0.142228122 -0.588131242  
## 1006 -2.112892950 0.72861152 -0.120923916 0.747362671  
## 1009 -2.056191175 0.75408467 -0.135153918 -0.698164487  
## 1010 -2.094182035 0.72192454 -0.114737388 0.581683845  
## 1011 -2.054539398 0.75548293 -0.136041594 -0.753810067  
## 1012 -2.052887621 0.75688120 -0.136929269 -0.809455647  
## 1013 -2.114385074 0.96817137 0.076377134 -0.748876489  
## 1014 -2.111081521 0.97096790 0.074601784 -0.860167649  
## 1015 -2.119340403 0.96397657 0.079040160 -0.581939750  
## 1016 -2.167241923 0.92342684 0.104782742 1.031782060  
## 1017 -2.139161722 0.94719737 0.089692263 0.085807206  
## 1018 -2.130902839 0.95418870 0.085253887 -0.192420692  
## 1020 -2.119340403 0.96397657 0.079040160 -0.581939750  
## 1021 -1.614942600 0.42044572 -0.211194655 -0.743241790  
## 1023 -1.652933460 0.38828559 -0.190778124 0.536606543  
## 1024 -1.614942600 0.42044572 -0.211194655 -0.743241790  
## 1025 -1.681013661 0.36451506 -0.175687645 1.482581397  
## 1026 -1.682665437 0.36311679 -0.174799970 1.538226977  
## 1027 -1.608335494 0.42603878 -0.214745356 -0.965824108  
## 1028 -1.631460365 0.40646305 -0.202317902 -0.186785993  
## 1030 -1.628105897 0.38472791 -0.182062721 0.425538187  
## 1031 -1.671996768 0.44898130 -0.192219910 -0.242687674  
## 1032 -1.690166309 0.43360036 -0.182455482 0.369413703  
## 1033 -0.462475058 -0.82183713 -0.837070934 0.031177687  
## 1034 -0.444305516 -0.80645620 -0.846835361 -0.580923689  
## 1035 -0.421180645 -0.78688047 -0.859262815 -1.359961805  
## 1036 -0.434394857 -0.79806660 -0.852161413 -0.914797167  
## 1037 -0.495510588 -0.84980246 -0.819317429 1.144089281  
## 1038 -0.500465918 -0.85399726 -0.816654403 1.311026020  
## 1039 -0.432743080 -0.79666833 -0.853049088 -0.970442747  
## 1040 -0.432743080 -0.79666833 -0.853049088 -0.970442747  
## 1042 -0.444305516 -0.80645620 -0.846835361 -0.580923689  
## 1047 -0.193785042 -1.15434304 -0.399554658 0.895327516  
## 1048 -0.335334406 -0.77612914 0.663332451 0.349859005  
## 1049 -0.302298875 -0.74816381 0.645578946 -0.763052588  
## 1050 -0.295691769 -0.74257074 0.642028245 -0.985634907  
## 1051 -0.284661798 -0.74128358 0.639022559 -0.986597838  
## 1052 -0.273099362 -0.73149571 0.632808832 -1.376116895  
## 1053 -0.278054692 -0.73569051 0.635471858 -1.209180156  
## 1054 -0.316045552 -0.76785064 0.655888388 0.070668176  
## 1055 -0.291268904 -0.74687664 0.642573260 -0.764015519  
## 1056 -0.306134893 -0.75946104 0.650562337 -0.263205302  
## 1057 -0.352384635 -0.79861250 0.675417243 1.294870929  
## 1058 -0.316045552 -0.76785064 0.655888388 0.070668176  
## 1059 -0.306134893 -0.75946104 0.650562337 -0.263205302  
## 1060 -0.360643518 -0.80560383 0.679855620 1.573098827  
## 1061 -0.304028547 -0.67179442 0.700925932 -0.769469178  
## 1062 -0.318894536 -0.68437882 0.708915009 -0.268658961  
## 1063 -0.373403162 -0.73052161 0.738208292 1.567645169  
## 1064 -0.318894536 -0.68437882 0.708915009 -0.268658961  
## 1066 -0.584221566 -0.83831038 0.278401681 0.389409249  
## 1067 -0.574310906 -0.82992078 0.273075630 0.055535771  
## 1070 0.317366968 -1.52210950 0.297456158 1.555449512  
## 1071 0.381786252 -1.46757711 0.262836824 -0.614728095  
## 1073 0.401401711 -1.47315087 0.263778129 -0.725741002  
## 1074 0.401401711 -1.47315087 0.263778129 -0.725741002  
## 1075 0.383232169 -1.48853181 0.273542557 -0.113639626  
## 1077 0.859823532 -1.74439676 -0.086174886 -0.784813894  
## 1078 0.866430638 -1.73880369 -0.089725587 -1.007396213  
## 1079 0.835046884 -1.76537076 -0.072859757 0.049869801  
## 1080 0.815225566 -1.78214995 -0.062207654 0.717616757  
## 1082 0.747519836 -1.50048364 0.154387159 -1.008319614  
## 1083 0.709528976 -1.53264377 0.174803690 0.271528719  
## 1084 0.742564507 -1.50467844 0.157050185 -0.841382875  
## 1085 0.735957401 -1.51027151 0.160600886 -0.618800556  
## 1086 0.716136082 -1.52705070 0.171252989 0.048946400  
## 1087 0.696314764 -1.54382990 0.181905091 0.716693356  
## 1088 0.716136082 -1.52705070 0.171252989 0.048946400  
## 1089 0.684752328 -1.55361777 0.188118818 1.106212414  
## 1094 0.918846411 -1.70515712 -0.077580420 -0.505527965  
## 1095 0.819869192 -1.47206477 0.043351369 -0.004367083  
## 1097 0.836386957 -1.45808211 0.034474617 -0.560822879  
## 1098 0.776923002 -1.50841970 0.066430925 1.442417989  
## 1099 0.844645840 -1.45109077 0.030036241 -0.839050778  
## 1101 0.931833376 -1.52918096 0.003387219 -0.620825098  
## 1102 0.875672974 -1.57672202 0.033568176 1.271124611  
## 1103 0.945047588 -1.51799482 -0.003714183 -1.065989735  
## 1104 0.872369421 -1.57951855 0.035343527 1.382415771  
## 1105 0.883931857 -1.56973068 0.029129800 0.992896713  
## 1107 0.919137329 -1.74303356 -0.305839140 -1.013282749  
## 1109 0.710851619 -1.34996533 0.057101528 -0.842894522  
## 1110 0.745410746 -1.62209672 -0.135543617 -0.508441952  
## 1112 0.543785115 -1.17372192 0.137295274 -1.007026755  
## 1114 0.834330278 -1.59946398 -0.116262767 -0.280034167  
## 1115 0.771562770 -1.65259810 -0.082531108 1.834497860  
## 1119 0.599843868 -1.22835744 0.053183206 -0.725411681  
## 1120 0.568854306 -1.22117781 0.081612806 -0.335418310  
## 1121 0.575461412 -1.21558475 0.078062106 -0.558000629  
## 1123 0.485067473 -1.06573092 0.190991640 -0.224333285  
## 1124 0.455335495 -1.09089972 0.206969794 0.777287149  
## 1126 0.503237015 -1.05034999 0.181227213 -0.836434662  
## 1127 0.456987272 -1.08950145 0.206082119 0.721641569  
## 1128 0.476808590 -1.07272225 0.195430016 0.053894613  
## 1129 0.503237015 -1.05034999 0.181227213 -0.836434662  
## 1130 0.488371026 -1.06293439 0.189216290 -0.335624445  
## 1131 0.415692859 -1.12445811 0.228274000 2.112781061  
## 1132 0.488371026 -1.06293439 0.189216290 -0.335624445  
## 1133 0.456987272 -1.08950145 0.206082119 0.721641569  
## 1136 0.252112132 -0.74691071 0.348469704 0.054896260  
## 1137 0.276888780 -0.72593671 0.335154575 -0.779787435  
## 1138 0.232290814 -0.76368991 0.359121806 0.722643216  
## 1139 0.207514166 -0.78466390 0.372436935 1.557326911  
## 1141 0.264849287 -0.70449833 0.158707322 -0.285238831  
## 1142 0.183912237 -0.77301339 0.202203409 2.441394573  
## 1143 0.279735060 -0.68774902 0.155424570 -0.841496794  
## 1144 0.279735060 -0.68774902 0.155424570 -0.841496794  
## 1146 0.196774884 -0.51823898 0.252078118 -0.898840992  
## 1147 0.175301790 -0.53641644 0.263617896 -0.175448456  
## 1148 0.195123108 -0.51963724 0.252965794 -0.843195413  
## 1149 0.176953566 -0.53501817 0.262730221 -0.231094036  
## 1150 0.195123108 -0.51963724 0.252965794 -0.843195413  
## 1151 0.125748494 -0.57836443 0.290248153 1.493918934  
## 1152 0.198426661 -0.51684071 0.251190443 -0.954486572  
## 1153 0.122444941 -0.58116097 0.292023504 1.605210093  
## 1154 0.206685544 -0.50984938 0.246752067 -1.232714470  
## 1155 0.093701338 -0.48397561 0.324049460 0.047860576  
## 1156 0.125085092 -0.45740855 0.307183631 -1.009405438  
## 1159 -0.079028413 -0.22216214 0.487514228 -0.786577077  
## 1160 -0.107108614 -0.24593267 0.502604707 0.159397777  
## 1161 -0.083983743 -0.22635694 0.490177254 -0.619640338  
## 1163 0.002968692 -0.34448703 0.382922187 0.158554453  
## 1165 -0.029769376 -0.46460903 0.323635891 -0.285543123  
## 1169 -0.144731835 -0.32665395 0.457956987 -0.785726491  
## 1170 -0.169508483 -0.34762795 0.471272116 0.048957205  
## 1171 -0.169508483 -0.34762795 0.471272116 0.048957205  
## 1172 -0.189329802 -0.36440715 0.481924219 0.716704161  
## 1174 -0.185889056 -0.29774097 0.484716994 -0.729970722  
## 1175 -0.175978397 -0.28935138 0.479390943 -1.063844200  
## 1176 -0.192496162 -0.30333404 0.488267695 -0.507388404  
## 1183 -0.095714164 -0.58049606 0.082159157 0.665381245  
## 1184 -0.148317272 -0.43394660 0.173804047 0.552751471  
## 1185 -0.110326412 -0.40178647 0.153387517 -0.727096861  
## 1187 -0.108674636 -0.40038821 0.152499842 -0.782742441  
## 1188 -0.179701026 -0.46051367 0.190669877 1.610017485  
## 1189 -0.131799507 -0.41996394 0.164927295 -0.003704325  
## 1192 -0.316929605 -0.27778261 0.362149737 0.720259788  
## 1193 -0.252510320 -0.22325022 0.327530403 -1.449917819  
## 1194 -0.278938744 -0.24562248 0.341733207 -0.559588545  
## 1195 -0.265724532 -0.23443635 0.334631805 -1.004753182  
## 1196 -0.297108286 -0.26100341 0.351497634 0.052512832  
## 1197 -0.333447370 -0.29176528 0.371026489 1.276715585  
## 1198 -0.307018945 -0.26939301 0.356823685 0.386386310  
## 1199 -0.287197627 -0.25261381 0.346171583 -0.281360646  
## 1200 -0.278938744 -0.24562248 0.341733207 -0.559588545  
## 1201 -0.272331638 -0.24002942 0.338182506 -0.782170863  
## 1202 -0.290501180 -0.25541035 0.347946933 -0.170069487  
## 1203 -0.373890003 -0.10088073 0.459022789 0.051089605  
## 1204 -0.373890003 -0.10088073 0.459022789 0.051089605  
## 1205 -0.363979344 -0.09249114 0.453696737 -0.282783873  
## 1206 -0.335899143 -0.06872061 0.438606258 -1.228758728  
## 1207 -0.342506249 -0.07431367 0.442156959 -1.006176409  
## 1208 -0.373890003 -0.10088073 0.459022789 0.051089605  
## 1209 -0.349113355 -0.07990674 0.445707660 -0.783594090  
## 1210 -0.354068685 -0.08410154 0.448370686 -0.616657351  
## 1211 -0.418487969 -0.13863393 0.482990020 1.553520256  
## 1212 -0.340635435 -0.32214631 0.268633897 0.216590758  
## 1213 -0.309251681 -0.29557925 0.251768068 -0.840675255  
## 1214 -0.324117670 -0.30816365 0.259757145 -0.339865038  
## 1221 -0.013185909 -0.28314886 0.308279986 -0.952422985  
## 1222 -0.019793015 -0.28874193 0.311830687 -0.729840666  
## 1223 -0.052828546 -0.31670726 0.329584192 0.383070927  
## 1225 -0.074301641 -0.33488472 0.341123970 1.106463463  
## 1227 0.618565351 -0.98092959 -0.191144511 -0.614913889  
## 1228 0.598744033 -0.99770879 -0.180492408 0.052833067  
## 1229 0.618565351 -0.98092959 -0.191144511 -0.614913889  
## 1294 -0.047693247 -0.51428413 -1.777021493 -0.034107902

fit.pc$scores<-data.frame(fit.pc$scores)  
fit.pc$scores[1]

## RC1  
## 3 0.619089220  
## 4 0.590941419  
## 6 0.709488130  
## 7 0.772255638  
## 12 0.674527040  
## 19 0.767537852  
## 20 0.790662723  
## 22 0.788768777  
## 24 0.752817381  
## 25 0.716478298  
## 26 0.808977783  
## 27 0.769335147  
## 28 0.789156465  
## 37 0.868450657  
## 41 0.818444816  
## 42 0.742463096  
## 43 0.821748369  
## 48 0.787366704  
## 49 0.840223553  
## 50 0.843527106  
## 51 0.820402235  
## 52 0.820402235  
## 54 0.779107822  
## 55 0.780052779  
## 56 0.776775431  
## 60 0.684555594  
## 62 0.930265047  
## 63 0.826203126  
## 64 0.918702612  
## 75 1.301449854  
## 76 1.445154412  
## 77 1.362565585  
## 78 1.400556445  
## 79 1.430288423  
## 80 1.509710885  
## 81 1.425132383  
## 82 1.387141523  
## 85 1.390141002  
## 86 1.365364354  
## 87 1.375275013  
## 88 1.329025270  
## 89 1.403355214  
## 90 1.355453695  
## 96 1.214128418  
## 97 1.136494922  
## 98 1.192655323  
## 99 1.030781224  
## 100 1.214128418  
## 106 0.955876493  
## 107 0.972394258  
## 108 1.018644001  
## 109 0.912930303  
## 111 1.035270515  
## 112 1.030315186  
## 113 1.031966962  
## 114 1.020404527  
## 115 1.017100974  
## 116 1.045181175  
## 117 1.030315186  
## 118 0.954333465  
## 120 1.038388090  
## 125 1.058802001  
## 126 1.030721800  
## 127 1.083578649  
## 129 1.067060884  
## 130 1.058802001  
## 131 1.062105554  
## 132 1.062019935  
## 133 0.961261566  
## 134 1.083493030  
## 135 1.078537700  
## 136 1.095055465  
## 137 1.032287957  
## 138 1.017421968  
## 184 -1.307202957  
## 185 -1.295640521  
## 186 -1.320417169  
## 188 -1.295640521  
## 197 -1.739110780  
## 198 -1.702771696  
## 199 -1.686253931  
## 200 -1.752324992  
## 201 -1.719289462  
## 203 -1.717637685  
## 205 -1.735997738  
## 207 -1.742604844  
## 208 -1.717828197  
## 209 -1.778943928  
## 210 -1.716176420  
## 211 -1.747560174  
## 212 -1.742604844  
## 213 -1.716176420  
## 215 -1.778943928  
## 216 -1.722783526  
## 217 -1.717828197  
## 226 -1.152931561  
## 228 -1.454335162  
## 229 -1.413040748  
## 230 -1.426254961  
## 232 -1.452683385  
## 233 -1.472708581  
## 235 -1.408289296  
## 236 -1.408289296  
## 238 -1.507237140  
## 244 -2.070710413  
## 245 -2.080621072  
## 249 -2.058155476  
## 250 -2.043289487  
## 251 -2.121349335  
## 252 -2.207241715  
## 253 -2.104831570  
## 255 -2.104831570  
## 256 -1.645133392  
## 257 -1.580714108  
## 259 -1.521293609  
## 260 -1.519641832  
## 261 -1.521293609  
## 262 -1.496516961  
## 263 -1.590668223  
## 264 -1.557632693  
## 265 -1.557632693  
## 268 -0.414471223  
## 269 -0.302150419  
## 270 -0.315590727  
## 271 -0.305680068  
## 272 -0.376706459  
## 273 -0.351929811  
## 274 -0.295769409  
## 275 -0.284206973  
## 276 -0.284206973  
## 277 -0.295769409  
## 279 -0.370709039  
## 280 -0.380619698  
## 281 -0.400441016  
## 282 -0.426869440  
## 283 -0.345932391  
## 284 -0.281072815  
## 285 -0.335581441  
## 286 -0.274465709  
## 287 -0.319063675  
## 289 -0.303176434  
## 290 -0.278399786  
## 291 -0.329604858  
## 292 -0.278399786  
## 294 -0.308543842  
## 295 -0.565262871  
## 296 -0.623075050  
## 297 -0.555352212  
## 298 -0.533879117  
## 299 -0.596646625  
## 300 -0.535530894  
## 303 0.362882555  
## 305 0.338105907  
## 306 0.364534331  
## 307 0.265427740  
## 310 0.873772876  
## 311 0.835782015  
## 312 0.855603334  
## 313 0.852299781  
## 314 0.860558663  
## 315 0.858906887  
## 316 0.852299781  
## 318 0.822567803  
## 319 0.804398261  
## 321 0.778951735  
## 325 0.936514705  
## 326 0.906091293  
## 327 0.841875065  
## 328 0.833616182  
## 329 0.797277098  
## 330 0.833616182  
## 336 0.689836654  
## 337 0.694791984  
## 338 0.665060006  
## 339 0.704702643  
## 340 0.648542241  
## 341 0.743912817  
## 342 0.758778806  
## 343 0.743912817  
## 344 0.724091499  
## 345 0.700966628  
## 350 0.873875635  
## 351 0.849098988  
## 352 0.860661423  
## 353 0.875527412  
## 354 0.882134518  
## 357 0.597883623  
## 358 0.566499869  
## 359 0.616053165  
## 360 0.596270716  
## 363 0.449598726  
## 364 0.485937810  
## 365 0.514018011  
## 366 0.525580446  
## 367 0.525580446  
## 369 0.300711436  
## 373 0.296286132  
## 374 0.218652635  
## 375 0.295335034  
## 376 0.239984913  
## 377 0.200531200  
## 378 0.226959625  
## 379 0.195575871  
## 380 0.225307848  
## 381 0.215397189  
## 382 0.215397189  
## 383 0.240173837  
## 384 0.215397189  
## 386 0.212093636  
## 388 0.164192117  
## 389 0.137763692  
## 390 0.217048966  
## 391 0.050675101  
## 392 0.120049715  
## 393 0.121701491  
## 394 0.096924843  
## 398 0.071527486  
## 399 0.059965050  
## 400 0.038491955  
## 401 0.048402614  
## 402 0.053357944  
## 403 -0.032534436  
## 404 -0.073828849  
## 405 -0.002802458  
## 406 0.066572156  
## 407 0.028581296  
## 410 -0.007757788  
## 411 0.059965050  
## 412 0.048402614  
## 413 0.038491955  
## 414 0.028581296  
## 415 0.018670637  
## 416 -0.020972000  
## 418 0.011379161  
## 419 -0.064602559  
## 420 -0.087058841  
## 421 -0.060622203  
## 422 -0.077905680  
## 423 -0.084512786  
## 424 -0.086164562  
## 425 -0.087816339  
## 426 -0.135717858  
## 427 -0.072950350  
## 429 -0.077905680  
## 430 -0.162146282  
## 431 -0.071298573  
## 433 -0.074602126  
## 435 -0.236694783  
## 443 -0.167937750  
## 444 -0.091956030  
## 445 -0.124991560  
## 446 -0.091956030  
## 447 -0.062224052  
## 448 -0.052313393  
## 449 -0.108473795  
## 450 -0.190093147  
## 453 -0.315245032  
## 454 -0.264039959  
## 455 -0.295423713  
## 456 -0.295423713  
## 457 -0.275602395  
## 462 -0.278125991  
## 466 -0.061905125  
## 467 -0.045387360  
## 468 -0.094940655  
## 469 -0.033824924  
## 471 0.603271611  
## 640 -1.808384506  
## 641 -1.889321556  
## 749 0.614458041  
## 750 0.614458041  
## 751 0.596288499  
## 752 0.612806264  
## 758 0.733149990  
## 759 0.736453543  
## 760 0.731498213  
## 761 0.708373342  
## 762 0.663775375  
## 763 0.738105319  
## 764 0.724891107  
## 770 0.755522704  
## 773 0.781334938  
## 774 0.768120726  
## 775 0.786290268  
## 777 0.753201545  
## 779 0.802754841  
## 780 0.817620829  
## 781 0.789540628  
## 783 0.789731408  
## 792 0.770342480  
## 801 0.815968381  
## 802 0.779629298  
## 803 0.771370415  
## 805 0.815968381  
## 806 0.784160572  
## 808 0.782508795  
## 809 0.746169712  
## 810 0.781258270  
## 811 0.839070449  
## 813 0.844025778  
## 814 0.804383142  
## 815 0.844025778  
## 817 0.656229280  
## 821 0.896898327  
## 822 0.934889187  
## 823 0.915067868  
## 824 0.928282081  
## 825 0.906808986  
## 833 1.478777552  
## 834 1.450697351  
## 835 1.472170446  
## 836 1.442438469  
## 837 1.420965374  
## 842 1.443006674  
## 843 1.438051345  
## 844 1.421533579  
## 845 1.414926473  
## 846 1.436399568  
## 848 1.367024954  
## 851 1.331496682  
## 852 1.346362671  
## 853 1.367835766  
## 854 1.366183989  
## 856 1.310023587  
## 857 1.136265556  
## 858 1.197381288  
## 859 1.197381288  
## 860 1.197381288  
## 861 1.167649311  
## 865 1.160150128  
## 866 1.203096317  
## 876 1.019251795  
## 877 1.007689359  
## 878 1.091929962  
## 879 1.040724890  
## 882 1.079683790  
## 884 1.116330355  
## 885 1.056866400  
## 886 1.113026802  
## 887 1.084946601  
## 888 1.099812590  
## 889 1.117982132  
## 890 1.106419696  
## 891 1.079991272  
## 892 1.050259294  
## 893 1.106419696  
## 894 1.051911071  
## 941 -1.334160498  
## 942 -1.311035626  
## 943 -1.299473191  
## 944 -1.365544252  
## 945 -1.330856945  
## 946 -1.350678263  
## 947 -1.306080297  
## 948 -1.307732073  
## 949 -1.320946285  
## 950 -1.307732073  
## 959 -1.741618242  
## 960 -1.739966465  
## 961 -1.733359359  
## 962 -1.781260878  
## 963 -1.746098649  
## 964 -1.808537708  
## 965 -1.823403697  
## 966 -1.859742780  
## 967 -1.795323496  
## 969 -1.793671719  
## 970 -1.820100144  
## 971 -1.811841261  
## 972 -1.796975272  
## 982 -1.433376628  
## 983 -1.477721246  
## 985 -1.433123280  
## 986 -1.419909067  
## 987 -1.447989268  
## 988 -1.457899927  
## 990 -1.426516173  
## 991 -1.434775056  
## 992 -1.500846117  
## 993 -1.502497894  
## 994 -1.422769297  
## 995 -1.411206861  
## 997 -1.525166122  
## 998 -2.108714298  
## 999 -2.083937650  
## 1000 -2.032732578  
## 1001 -2.054205672  
## 1005 -2.073250314  
## 1006 -2.112892950  
## 1009 -2.056191175  
## 1010 -2.094182035  
## 1011 -2.054539398  
## 1012 -2.052887621  
## 1013 -2.114385074  
## 1014 -2.111081521  
## 1015 -2.119340403  
## 1016 -2.167241923  
## 1017 -2.139161722  
## 1018 -2.130902839  
## 1020 -2.119340403  
## 1021 -1.614942600  
## 1023 -1.652933460  
## 1024 -1.614942600  
## 1025 -1.681013661  
## 1026 -1.682665437  
## 1027 -1.608335494  
## 1028 -1.631460365  
## 1030 -1.628105897  
## 1031 -1.671996768  
## 1032 -1.690166309  
## 1033 -0.462475058  
## 1034 -0.444305516  
## 1035 -0.421180645  
## 1036 -0.434394857  
## 1037 -0.495510588  
## 1038 -0.500465918  
## 1039 -0.432743080  
## 1040 -0.432743080  
## 1042 -0.444305516  
## 1047 -0.193785042  
## 1048 -0.335334406  
## 1049 -0.302298875  
## 1050 -0.295691769  
## 1051 -0.284661798  
## 1052 -0.273099362  
## 1053 -0.278054692  
## 1054 -0.316045552  
## 1055 -0.291268904  
## 1056 -0.306134893  
## 1057 -0.352384635  
## 1058 -0.316045552  
## 1059 -0.306134893  
## 1060 -0.360643518  
## 1061 -0.304028547  
## 1062 -0.318894536  
## 1063 -0.373403162  
## 1064 -0.318894536  
## 1066 -0.584221566  
## 1067 -0.574310906  
## 1070 0.317366968  
## 1071 0.381786252  
## 1073 0.401401711  
## 1074 0.401401711  
## 1075 0.383232169  
## 1077 0.859823532  
## 1078 0.866430638  
## 1079 0.835046884  
## 1080 0.815225566  
## 1082 0.747519836  
## 1083 0.709528976  
## 1084 0.742564507  
## 1085 0.735957401  
## 1086 0.716136082  
## 1087 0.696314764  
## 1088 0.716136082  
## 1089 0.684752328  
## 1094 0.918846411  
## 1095 0.819869192  
## 1097 0.836386957  
## 1098 0.776923002  
## 1099 0.844645840  
## 1101 0.931833376  
## 1102 0.875672974  
## 1103 0.945047588  
## 1104 0.872369421  
## 1105 0.883931857  
## 1107 0.919137329  
## 1109 0.710851619  
## 1110 0.745410746  
## 1112 0.543785115  
## 1114 0.834330278  
## 1115 0.771562770  
## 1119 0.599843868  
## 1120 0.568854306  
## 1121 0.575461412  
## 1123 0.485067473  
## 1124 0.455335495  
## 1126 0.503237015  
## 1127 0.456987272  
## 1128 0.476808590  
## 1129 0.503237015  
## 1130 0.488371026  
## 1131 0.415692859  
## 1132 0.488371026  
## 1133 0.456987272  
## 1136 0.252112132  
## 1137 0.276888780  
## 1138 0.232290814  
## 1139 0.207514166  
## 1141 0.264849287  
## 1142 0.183912237  
## 1143 0.279735060  
## 1144 0.279735060  
## 1146 0.196774884  
## 1147 0.175301790  
## 1148 0.195123108  
## 1149 0.176953566  
## 1150 0.195123108  
## 1151 0.125748494  
## 1152 0.198426661  
## 1153 0.122444941  
## 1154 0.206685544  
## 1155 0.093701338  
## 1156 0.125085092  
## 1159 -0.079028413  
## 1160 -0.107108614  
## 1161 -0.083983743  
## 1163 0.002968692  
## 1165 -0.029769376  
## 1169 -0.144731835  
## 1170 -0.169508483  
## 1171 -0.169508483  
## 1172 -0.189329802  
## 1174 -0.185889056  
## 1175 -0.175978397  
## 1176 -0.192496162  
## 1183 -0.095714164  
## 1184 -0.148317272  
## 1185 -0.110326412  
## 1187 -0.108674636  
## 1188 -0.179701026  
## 1189 -0.131799507  
## 1192 -0.316929605  
## 1193 -0.252510320  
## 1194 -0.278938744  
## 1195 -0.265724532  
## 1196 -0.297108286  
## 1197 -0.333447370  
## 1198 -0.307018945  
## 1199 -0.287197627  
## 1200 -0.278938744  
## 1201 -0.272331638  
## 1202 -0.290501180  
## 1203 -0.373890003  
## 1204 -0.373890003  
## 1205 -0.363979344  
## 1206 -0.335899143  
## 1207 -0.342506249  
## 1208 -0.373890003  
## 1209 -0.349113355  
## 1210 -0.354068685  
## 1211 -0.418487969  
## 1212 -0.340635435  
## 1213 -0.309251681  
## 1214 -0.324117670  
## 1221 -0.013185909  
## 1222 -0.019793015  
## 1223 -0.052828546  
## 1225 -0.074301641  
## 1227 0.618565351  
## 1228 0.598744033  
## 1229 0.618565351  
## 1294 -0.047693247

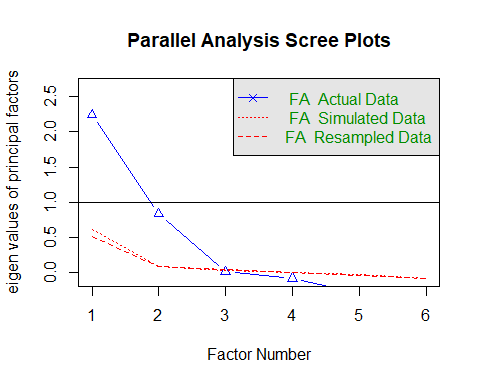
# See factor recommendation

fa.parallel(x\_new, fm='minres', fa='fa')

## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate =  
## rotate, : A loading greater than abs(1) was detected. Examine the loadings  
## carefully.

## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs  
## = np.obs, : The estimated weights for the factor scores are probably  
## incorrect. Try a different factor extraction method.

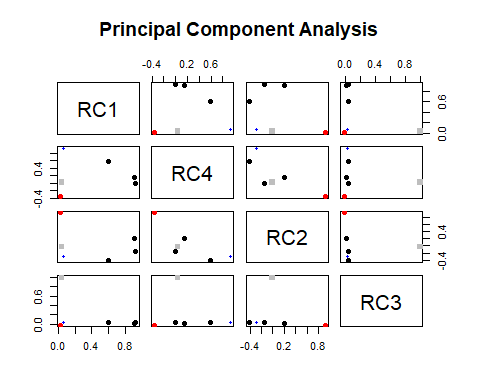
## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate =  
## rotate, : An ultra-Heywood case was detected. Examine the results carefully



## Parallel analysis suggests that the number of factors = 2 and the number of components = NA

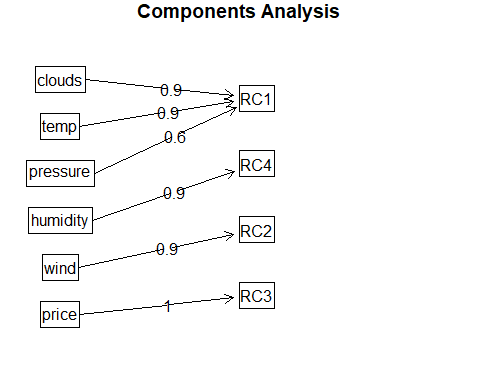
# Blue line shows the eigen values of actual data and the two red lines show simulated and resampled data. Here factors between 2-4 will be a good choice..

fa.plot(fit.pc) # See Correlations within Factors



# Visualize the relationship

fa.diagram(fit.pc)

 #Red dotted line means Wind marginally falls under the RC1 bucket.

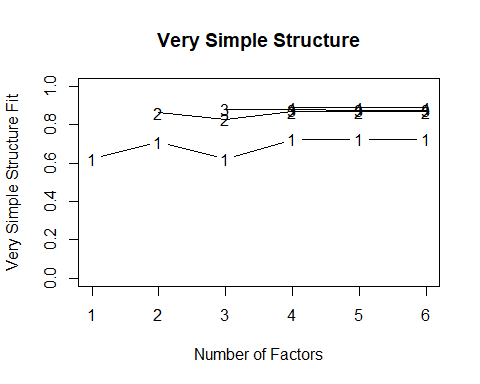
# See Factor recommendations for a simple structure

vss(x\_new)

## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate =  
## rotate, : A loading greater than abs(1) was detected. Examine the loadings  
## carefully.

## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs  
## = np.obs, : The estimated weights for the factor scores are probably  
## incorrect. Try a different factor extraction method.

## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate =  
## rotate, : An ultra-Heywood case was detected. Examine the results carefully



##   
## Very Simple Structure  
## Call: vss(x = x\_new)  
## Although the VSS complexity 1 shows 5 factors, it is probably more reasonable to think about 2 factors  
## VSS complexity 2 achieves a maximimum of 0.87 with 6 factors  
##   
## The Velicer MAP achieves a minimum of NA with 2 factors   
## BIC achieves a minimum of NA with 2 factors  
## Sample Size adjusted BIC achieves a minimum of NA with 2 factors  
##   
## Statistics by number of factors   
## vss1 vss2 map dof chisq prob sqresid fit RMSEA BIC SABIC complex  
## 1 0.62 0.00 0.16 9 6.0e+02 2.2e-123 4.0 0.62 0.346 543 571.6 1.0  
## 2 0.71 0.86 0.13 4 2.0e+01 4.3e-04 1.5 0.86 0.086 -5 7.7 1.3  
## 3 0.62 0.83 0.23 0 5.2e-01 NA 1.3 0.88 NA NA NA 1.6  
## 4 0.72 0.87 0.45 -3 1.9e-09 NA 1.2 0.89 NA NA NA 1.3  
## 5 0.72 0.87 1.00 -5 4.8e-13 NA 1.2 0.89 NA NA NA 1.3  
## 6 0.72 0.87 NA -6 4.8e-13 NA 1.2 0.89 NA NA NA 1.3  
## eChisq SRMR eCRMS eBIC  
## 1 5.7e+02 1.8e-01 0.239 510  
## 2 4.3e+00 1.6e-02 0.031 -21  
## 3 3.1e-01 4.3e-03 NA NA  
## 4 2.0e-10 1.1e-07 NA NA  
## 5 1.4e-13 2.9e-09 NA NA  
## 6 1.4e-13 2.9e-09 NA NA

# Regression analysis using the factors scores as the independent variable:

# Let’s combine the dependent variable and the factor scores into a dataset and label them.

cab<-cbind(fit.pc$scores[1], df[,c(1,4,5,7,8,9,10,12)])  
cab<-data.frame(cab)  
#View(cab)  
names(cab)[names(cab) == "RC1"] <- "Temp"  
names(cab)[names(cab) == "merge\_data\_0"] <- "No rain"  
names(cab)[names(cab) == "merge\_data\_1"] <- "Medium rain"  
#Labelling the data  
#names(cab)<-c("Price","Wind\_Pressure","Temperature","Humidity","Distance")  
head(cab)

## Temp location humidity wind distance cab\_type  
## 3 0.6190892 0 0.93 2.59 1.44 Uber  
## 4 0.5909414 0 0.93 2.65 1.36 Lyft  
## 6 0.7094881 0 0.92 2.59 1.34 Uber  
## 7 0.7722556 0 0.92 2.59 1.10 Uber  
## 12 0.6745270 0 0.92 3.02 2.28 Lyft  
## 19 0.7675379 0 0.91 1.16 2.64 Lyft  
## C..Users.nisht.Desktop.MITA.Fall.MVA.Final.Project.Uber.Lyft.price.predict.Multiple.Regression.Rmd\_0  
## 3 1  
## 4 1  
## 6 1  
## 7 1  
## 12 1  
## 19 1  
## C..Users.nisht.Desktop.MITA.Fall.MVA.Final.Project.Uber.Lyft.price.predict.Multiple.Regression.Rmd\_1  
## 3 0  
## 4 0  
## 6 0  
## 7 0  
## 12 0  
## 19 0  
## price  
## 3 8.5  
## 4 16.5  
## 6 26.5  
## 7 7.5  
## 12 22.5  
## 19 22.5

# Let’s split the dataset into training and testing dataset. (80:20)

set.seed(101)  
Atrain<-sample(nrow(cab),nrow(cab)\*0.80)  
cab\_train<-cab[Atrain,]  
cab\_test<-cab[-Atrain,]  
dim(cab\_train)

## [1] 442 9

dim(cab\_test)

## [1] 111 9

# Performing multiple regression (Taking alpha=0.1)

fit3 <- lm(price~., data=cab\_train)  
#show the results  
summary(fit3)

##   
## Call:  
## lm(formula = price ~ ., data = cab\_train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -17.988 -6.608 -1.678 4.753 47.263   
##   
## Coefficients:  
## Estimate  
## (Intercept) 0.1097  
## Temp 0.4161  
## location1 -1.8046  
## humidity 11.9681  
## wind 0.1391  
## distance 2.8853  
## cab\_typeUber -0.9246  
## C..Users.nisht.Desktop.MITA.Fall.MVA.Final.Project.Uber.Lyft.price.predict.Multiple.Regression.Rmd\_0 1.7434  
## C..Users.nisht.Desktop.MITA.Fall.MVA.Final.Project.Uber.Lyft.price.predict.Multiple.Regression.Rmd\_1 1.4068  
## Std. Error  
## (Intercept) 9.5568  
## Temp 0.4256  
## location1 0.8214  
## humidity 7.3182  
## wind 0.2432  
## distance 0.5527  
## cab\_typeUber 0.8436  
## C..Users.nisht.Desktop.MITA.Fall.MVA.Final.Project.Uber.Lyft.price.predict.Multiple.Regression.Rmd\_0 3.1061  
## C..Users.nisht.Desktop.MITA.Fall.MVA.Final.Project.Uber.Lyft.price.predict.Multiple.Regression.Rmd\_1 2.7055  
## t value  
## (Intercept) 0.011  
## Temp 0.978  
## location1 -2.197  
## humidity 1.635  
## wind 0.572  
## distance 5.221  
## cab\_typeUber -1.096  
## C..Users.nisht.Desktop.MITA.Fall.MVA.Final.Project.Uber.Lyft.price.predict.Multiple.Regression.Rmd\_0 0.561  
## C..Users.nisht.Desktop.MITA.Fall.MVA.Final.Project.Uber.Lyft.price.predict.Multiple.Regression.Rmd\_1 0.520  
## Pr(>|t|)  
## (Intercept) 0.9908  
## Temp 0.3288  
## location1 0.0286  
## humidity 0.1027  
## wind 0.5676  
## distance 2.77e-07  
## cab\_typeUber 0.2737  
## C..Users.nisht.Desktop.MITA.Fall.MVA.Final.Project.Uber.Lyft.price.predict.Multiple.Regression.Rmd\_0 0.5749  
## C..Users.nisht.Desktop.MITA.Fall.MVA.Final.Project.Uber.Lyft.price.predict.Multiple.Regression.Rmd\_1 0.6033  
##   
## (Intercept)   
## Temp   
## location1 \*   
## humidity   
## wind   
## distance \*\*\*  
## cab\_typeUber   
## C..Users.nisht.Desktop.MITA.Fall.MVA.Final.Project.Uber.Lyft.price.predict.Multiple.Regression.Rmd\_0   
## C..Users.nisht.Desktop.MITA.Fall.MVA.Final.Project.Uber.Lyft.price.predict.Multiple.Regression.Rmd\_1   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 8.556 on 433 degrees of freedom  
## Multiple R-squared: 0.07858, Adjusted R-squared: 0.06156   
## F-statistic: 4.616 on 8 and 433 DF, p-value: 2.009e-05