MVA_Project2.R

thindprateek

2020-10-08

```r
###### Assignment 4 - Applying Principal Component Analysis #############

#Getting working directory
getwd()
```

```
## [1] "/Users/thindprateek/Desktop/Multivariate Analysis"
```

```r
library(pander)
#Setting directory to load dataset
setwd("/Users/thindprateek/Desktop/Multivariate Analysis")

#Reading the data into a dataframe
df <- read.csv(file = 'US_Accidents_June20.csv')

# Printing first few columns of dataset for inference
#head(df)

## Setting random seed to shuflle data before splitting
set.seed(23)

#Checking number of rows
rows<-sample(nrow(df))

#Shuffling the data
mva<-df[rows,]

#Taking the required number of instances from the shuffled data to reduce any biases
mva<-mva[500000:1000000,]

#Checking the structure of the dataset
str(mva)
```

```
## 'data.frame':    500001 obs. of  49 variables:
##  $ ID                    : chr  "A-210675" "A-1806002" "A-1932273" "A-11427" ...
##  $ Source                : chr  "MapQuest" "MapQuest" "MapQuest" "MapQuest" ...
##  $ TMC                   : num  241 201 201 201 NA 201 245 201 NA 201 ...
##  $ Severity              : int  3 2 2 2 2 2 3 2 2 3 ...
##  $ Start_Time            : chr  "2016-10-05 18:56:21" "2018-06-04 07:56:44" "2018-04-04 16:38:49" "20
##  $ End_Time              : chr  "2016-10-05 19:41:07" "2018-06-04 08:26:29" "2018-04-04 17:23:33" "20
##  $ Start_Lat             : num  40.8 40.3 33.7 37.6 38.6 ...
##  $ Start_Lng             : num  -74.3 -75.7 -117.9 -122.1 -121.6 ...
```

```
##  $ End_Lat              : num  NA NA NA NA 38.6 ...
##  $ End_Lng              : num  NA NA NA NA -122 ...
##  $ Distance.mi.         : num  0 0 0 0.01 0 0.01 0 0 0 0 ...
##  $ Description          : chr  "Left lane blocked due to accident on I-280 Westbound at Exits 4A 4B 
##  $ Number               : num  NA 1535 NA NA NA ...
##  $ Street               : chr  "I-280 E" "Sell Rd" "Corona del Mar Fwy N" "W Jackson St" ...
##  $ Side                 : chr  "R" "L" "R" "R" ...
##  $ City                 : chr  "Roseland" "Pottstown" "Costa Mesa" "Hayward" ...
##  $ County               : chr  "Essex" "Montgomery" "Orange" "Alameda" ...
##  $ State                : chr  "NJ" "PA" "CA" "CA" ...
##  $ Zipcode              : chr  "07068" "19464" "92626" "94544" ...
##  $ Country              : chr  "US" "US" "US" "US" ...
##  $ Timezone             : chr  "US/Eastern" "US/Eastern" "US/Pacific" "US/Pacific" ...
##  $ Airport_Code         : chr  "KCDW" "KPTW" "KSNA" "KHWD" ...
##  $ Weather_Timestamp    : chr  "2016-10-05 18:53:00" "2018-06-04 07:54:00" "2018-04-04 16:53:00" "20
##  $ Temperature.F.       : num  60.1 54 62.1 54 65 ...
##  $ Wind_Chill.F.        : num  NA NA NA NA 65 NA NA NA 55 NA ...
##  $ Humidity...          : num  67 93 72 59 15 87 96 54 38 12 ...
##  $ Pressure.in.         : num  30.3 29.8 30 30.1 29.9 ...
##  $ Visibility.mi.       : num  10 8 9 7 10 10 0.8 10 10 10 ...
##  $ Wind_Direction       : chr  "Calm" "NE" "SW" "NE" ...
##  $ Wind_Speed.mph.      : num  NA 4.6 8.1 9.2 17 9.2 NA NA 10 NA ...
##  $ Precipitation.in.    : num  NA NA NA NA 0 0 0.11 0 0 NA ...
##  $ Weather_Condition    : chr  "Clear" "Scattered Clouds" "Partly Cloudy" "Partly Cloudy" ...
##  $ Amenity              : chr  "False" "False" "False" "False" ...
##  $ Bump                 : chr  "False" "False" "False" "False" ...
##  $ Crossing             : chr  "False" "False" "False" "False" ...
##  $ Give_Way             : chr  "False" "False" "False" "False" ...
##  $ Junction             : chr  "False" "False" "False" "False" ...
##  $ No_Exit              : chr  "False" "False" "False" "False" ...
##  $ Railway              : chr  "False" "False" "False" "False" ...
##  $ Roundabout           : chr  "False" "False" "False" "False" ...
##  $ Station              : chr  "False" "False" "False" "False" ...
##  $ Stop                 : chr  "False" "False" "False" "False" ...
##  $ Traffic_Calming      : chr  "False" "False" "False" "False" ...
##  $ Traffic_Signal       : chr  "False" "False" "False" "False" ...
##  $ Turning_Loop         : chr  "False" "False" "False" "False" ...
##  $ Sunrise_Sunset       : chr  "Night" "Day" "Day" "Day" ...
##  $ Civil_Twilight       : chr  "Day" "Day" "Day" "Day" ...
##  $ Nautical_Twilight    : chr  "Day" "Day" "Day" "Day" ...
##  $ Astronomical_Twilight: chr  "Day" "Day" "Day" "Day" ...
```

```r
# Checking the number of rows and columns in the current uncleaned dataset
ncol(mva)
```

```
## [1] 49
```

```r
nrow(mva)
```

```
## [1] 500001
```

```r
# Printing all the column names to find and filter the relevant and irrelevant attributes
names<-names(mva)
names
```

```
##  [1] "ID"                    "Source"                "TMC"
##  [4] "Severity"              "Start_Time"            "End_Time"
##  [7] "Start_Lat"             "Start_Lng"             "End_Lat"
## [10] "End_Lng"               "Distance.mi."          "Description"
## [13] "Number"                "Street"                "Side"
## [16] "City"                  "County"                "State"
## [19] "Zipcode"               "Country"               "Timezone"
## [22] "Airport_Code"          "Weather_Timestamp"     "Temperature.F."
## [25] "Wind_Chill.F."         "Humidity..."           "Pressure.in."
## [28] "Visibility.mi."        "Wind_Direction"        "Wind_Speed.mph."
## [31] "Precipitation.in."     "Weather_Condition"     "Amenity"
## [34] "Bump"                  "Crossing"              "Give_Way"
## [37] "Junction"              "No_Exit"               "Railway"
## [40] "Roundabout"            "Station"               "Stop"
## [43] "Traffic_Calming"       "Traffic_Signal"        "Turning_Loop"
## [46] "Sunrise_Sunset"        "Civil_Twilight"        "Nautical_Twilight"
## [49] "Astronomical_Twilight"
```

```r
## DATA CLEANING ##

#Dropping the surplus attributes which do not contribute to the analysis
mva <- mva[-c(1:3,7:10,13,14,19,21:23,33,47:49)]

#Checking for any null values in the present dataset
# is.na(mva[,])

#Checking which rows have all the values filled and complete
# complete.cases(mva)

#Making a new dataframe with only the rows that have complete information and all values filled
Mva<-na.omit(mva)
Mva<-Mva[!(is.na(Mva$Sunrise_Sunset) | Mva$Sunrise_Sunset==""), ]
#Mva<- Mva[complete.cases(Mva),]
#Verifying for missing values in the new dataframe
#complete.cases(Mva)
unique(Mva$Sunrise_Sunset)
```

```
## [1] "Night" "Day"
```

```r
#Checking the number of rows and columns in the new CLEANED dataframe
ncol(Mva)
```

```
## [1] 32
```

```r
nrow(Mva)
```

```
## [1] 182841
```

```r
# Creating new dataframe with only the numerical attributes to perform statistical functions
num<-Mva[,c(1,4,11:15,17,18)]

##### STEP 1 #######

#Getting the Correlations between the measurements
s<-cor(num)
pander(s)
```

Table 1: Table continues below

	Severity	Distance.mi.	Temperature.F.
Severity	1	0.1831	-0.02777
Distance.mi.	0.1831	1	-0.02183
Temperature.F.	-0.02777	-0.02183	1
Wind_Chill.F.	-0.03348	-0.02463	0.9938
Humidity...	0.05271	0.02179	-0.4314
Pressure.in.	-0.003833	-0.03924	0.03943
Visibility.mi.	-0.0259	-0.01786	0.3168
Wind_Speed.mph.	0.05236	0.02073	-0.0109
Precipitation.in.	0.01955	0.003446	-0.03309

Table 2: Table continues below

	Wind_Chill.F.	Humidity...	Pressure.in.
Severity	-0.03348	0.05271	-0.003833
Distance.mi.	-0.02463	0.02179	-0.03924
Temperature.F.	0.9938	-0.4314	0.03943
Wind_Chill.F.	1	-0.4143	0.04066
Humidity...	-0.4143	1	0.1989
Pressure.in.	0.04066	0.1989	1
Visibility.mi.	0.3258	-0.4307	-0.08966
Wind_Speed.mph.	-0.06539	-0.1482	-0.04924
Precipitation.in.	-0.0336	0.1038	0.01784

	Visibility.mi.	Wind_Speed.mph.	Precipitation.in.
Severity	-0.0259	0.05236	0.01955
Distance.mi.	-0.01786	0.02073	0.003446
Temperature.F.	0.3168	-0.0109	-0.03309
Wind_Chill.F.	0.3258	-0.06539	-0.0336
Humidity...	-0.4307	-0.1482	0.1038
Pressure.in.	-0.08966	-0.04924	0.01784
Visibility.mi.	1	-0.02255	-0.1466
Wind_Speed.mph.	-0.02255	1	0.03133
Precipitation.in.	-0.1466	0.03133	1

```r
sum(diag(s))
```

```
## [1] 9
```

```
# Looking at the correlations table, it can be inferred that the attributes have moderate to low correl
# Therefore PCA may not prove to be a much beneficial Method for this Dataset

##### STEP 2 #######

# Using prcomp to compute the principal components
mva_pca<-prcomp(num[,],scale=TRUE)
pander(mva_pca)
```

Table 4: Principal Components Analysis (continued below)

	PC1	PC2	PC3	PC4	PC5
Severity	0.04896	-0.2888	-0.6212	0.152	-0.1152
Distance.mi.	0.03264	-0.3314	-0.5717	0.2638	0.1465
Temperature.F.	-0.5729	0.2091	-0.2037	-0.08448	0.003953
Wind_Chill.F.	-0.5713	0.2335	-0.1992	-0.05017	0.03824
Humidity...	0.4364	0.3363	-0.1681	0.1079	0.01564
Pressure.in.	0.049	0.5743	-0.2505	0.05854	-0.5941
Visibility.mi.	-0.3752	-0.226	0.1868	0.2242	0.007102
Wind_Speed.mph.	-0.005816	-0.4294	-0.02732	-0.6053	-0.5937
Precipitation.in.	0.09148	0.1735	-0.2818	-0.6826	0.5079

	PC6	PC7	PC8	PC9
Severity	0.6966	-0.04714	0.06667	-0.002319
Distance.mi.	-0.6755	0.1218	0.009067	-0.0009642
Temperature.F.	-0.03761	-0.2485	-0.1301	0.7067
Wind_Chill.F.	-0.02867	-0.2381	-0.1379	-0.7064
Humidity...	0.00185	-0.1741	-0.7911	0.01183
Pressure.in.	-0.09186	0.4356	0.2246	-0.002437
Visibility.mi.	0.1432	0.6972	-0.465	0.01049
Wind_Speed.mph.	-0.1487	-0.07566	-0.2585	-0.03668
Precipitation.in.	0.07209	0.3907	0.01534	0.001204

```
pander(summary(mva_pca)$importance)
```

Table 6: Table continues below

	PC1	PC2	PC3	PC4	PC5
Standard deviation	1.591	1.118	1.086	1.023	0.9682
Proportion of Variance	0.2814	0.1388	0.1311	0.1162	0.1042
Cumulative Proportion	0.2814	0.4202	0.5514	0.6676	0.7717

	PC6	PC7	PC8	PC9
Standard deviation	0.9008	0.8639	0.7015	0.06792
Proportion of Variance	0.09016	0.08292	0.05468	0.00051

	PC6	PC7	PC8	PC9
Cumulative Proportion	0.8619	0.9448	0.9995	1

```
# sample scores stored in mva_pca$x
# singular values(square roots of eigenvalues) stored in mva_pca$sdev
# loadings (eigenvectors) are stored in mva_pca$rotation
# variable means stored in mva_pca$center
# variable standard deviations stored in mva_pca$scale
# A table containing eigenvalues and %'s accounted, follows
# Eigenvalues are sdev^2

pander(eigen_mva<-mva_pca$sdev^2)
```

2.533, *1.249*, *1.18*, *1.046*, *0.9374*, *0.8115*, *0.7463*, *0.4921* and *0.004613*

```
names(eigen_mva)<-paste("PC",1:9,sep="")
pander(eigen_mva)
```

PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9
2.533	1.249	1.18	1.046	0.9374	0.8115	0.7463	0.4921	0.004613

```
sum_lambda<-sum(eigen_mva)
pander(sum_lambda)
```

9

```
prop_var<-eigen_mva/sum_lambda
pander(prop_var)
```

PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9
0.2814	0.1388	0.1312	0.1162	0.1042	0.09016	0.08292	0.05468	0.0005126

```
cum_var_mva<-cumsum(prop_var)
pander(cum_var_mva)
```

PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9
0.2814	0.4202	0.5514	0.6676	0.7717	0.8619	0.9448	0.9995	1

```
mat_lambda<-rbind(eigen_mva,prop_var,cum_var_mva)
rownames(mat_lambda)<-c("Eigenvalues","Prop. variance","Cum. prop. variance")
pander(round(mat_lambda,4))
```

Table 11: Table continues below

	PC1	PC2	PC3	PC4	PC5	PC6
Eigenvalues	2.533	1.25	1.18	1.046	0.9374	0.8115
Prop. variance	0.2814	0.1388	0.1312	0.1162	0.1042	0.0902
Cum. prop. variance	0.2814	0.4202	0.5514	0.6676	0.7717	0.8619

	PC7	PC8	PC9
Eigenvalues	0.7463	0.4921	0.0046
Prop. variance	0.0829	0.0547	5e-04
Cum. prop. variance	0.9448	0.9995	1

```
pander(summary(mva_pca))
```

Table 13: Principal Components Analysis (continued below)

	PC1	PC2	PC3	PC4	PC5
Severity	0.04896	-0.2888	-0.6212	0.152	-0.1152
Distance.mi.	0.03264	-0.3314	-0.5717	0.2638	0.1465
Temperature.F.	-0.5729	0.2091	-0.2037	-0.08448	0.003953
Wind_Chill.F.	-0.5713	0.2335	-0.1992	-0.05017	0.03824
Humidity...	0.4364	0.3363	-0.1681	0.1079	0.01564
Pressure.in.	0.049	0.5743	-0.2505	0.05854	-0.5941
Visibility.mi.	-0.3752	-0.226	0.1868	0.2242	0.007102
Wind_Speed.mph.	-0.005816	-0.4294	-0.02732	-0.6053	-0.5937
Precipitation.in.	0.09148	0.1735	-0.2818	-0.6826	0.5079

	PC6	PC7	PC8	PC9
Severity	0.6966	-0.04714	0.06667	-0.002319
Distance.mi.	-0.6755	0.1218	0.009067	-0.0009642
Temperature.F.	-0.03761	-0.2485	-0.1301	0.7067
Wind_Chill.F.	-0.02867	-0.2381	-0.1379	-0.7064
Humidity...	0.00185	-0.1741	-0.7911	0.01183
Pressure.in.	-0.09186	0.4356	0.2246	-0.002437
Visibility.mi.	0.1432	0.6972	-0.465	0.01049
Wind_Speed.mph.	-0.1487	-0.07566	-0.2585	-0.03668
Precipitation.in.	0.07209	0.3907	0.01534	0.001204

Table 15: Table continues below

	PC1	PC2	PC3	PC4	PC5
Standard deviation	1.591	1.118	1.086	1.023	0.9682
Proportion of Variance	0.2814	0.1388	0.1311	0.1162	0.1042
Cumulative Proportion	0.2814	0.4202	0.5514	0.6676	0.7717

	PC6	PC7	PC8	PC9
Standard deviation	0.9008	0.8639	0.7015	0.06792
Proportion of Variance	0.09016	0.08292	0.05468	0.00051
Cumulative Proportion	0.8619	0.9448	0.9995	1

```
pander(mva_pca$rotation)
```

Table 17: Table continues below

	PC1	PC2	PC3	PC4	PC5
Severity	0.04896	-0.2888	-0.6212	0.152	-0.1152
Distance.mi.	0.03264	-0.3314	-0.5717	0.2638	0.1465
Temperature.F.	-0.5729	0.2091	-0.2037	-0.08448	0.003953
Wind_Chill.F.	-0.5713	0.2335	-0.1992	-0.05017	0.03824
Humidity...	0.4364	0.3363	-0.1681	0.1079	0.01564
Pressure.in.	0.049	0.5743	-0.2505	0.05854	-0.5941
Visibility.mi.	-0.3752	-0.226	0.1868	0.2242	0.007102
Wind_Speed.mph.	-0.005816	-0.4294	-0.02732	-0.6053	-0.5937
Precipitation.in.	0.09148	0.1735	-0.2818	-0.6826	0.5079

	PC6	PC7	PC8	PC9
Severity	0.6966	-0.04714	0.06667	-0.002319
Distance.mi.	-0.6755	0.1218	0.009067	-0.0009642
Temperature.F.	-0.03761	-0.2485	-0.1301	0.7067
Wind_Chill.F.	-0.02867	-0.2381	-0.1379	-0.7064
Humidity...	0.00185	-0.1741	-0.7911	0.01183
Pressure.in.	-0.09186	0.4356	0.2246	-0.002437
Visibility.mi.	0.1432	0.6972	-0.465	0.01049
Wind_Speed.mph.	-0.1487	-0.07566	-0.2585	-0.03668
Precipitation.in.	0.07209	0.3907	0.01534	0.001204

```
# Sample scores stored in mva_pca$x
#pander(mva_pca$x)

attach(Mva)
#Identifying the scores by their time of day occurence ( performed wrt to one categorical variable as a
#will be performed with all later)
mvatyp_pca<-cbind(data.frame(Sunrise_Sunset),mva_pca$x)
#pander(mvatyp_pca)
#names(mva)

# Means of scores for all the PC's classified by Sunrise_Sunset
tab_meansPC<-aggregate(mvatyp_pca[,2:10],by=list(Sunrise_Sunset=Mva$Sunrise_Sunset),mean)
pander(tab_meansPC)
```

Table 19: Table continues below

Sunrise_Sunset	PC1	PC2	PC3	PC4	PC5	PC6
Day	-0.2595	-0.02209	-0.01892	-0.09007	-0.04513	-0.0325
Night	0.736	0.06264	0.05368	0.2555	0.128	0.09218

PC7	PC8	PC9
-0.04505	0.02237	0.001413
0.1278	-0.06346	-0.004008

```
tab_meansPC <- tab_meansPC[rev(order(tab_meansPC$Sunrise_Sunset)),]
pander(tab_meansPC)
```

Table 21: Table continues below

	Sunrise_Sunset	PC1	PC2	PC3	PC4	PC5
2	Night	0.736	0.06264	0.05368	0.2555	0.128
1	Day	-0.2595	-0.02209	-0.01892	-0.09007	-0.04513

	PC6	PC7	PC8	PC9
2	0.09218	0.1278	-0.06346	-0.004008
1	-0.0325	-0.04505	0.02237	0.001413

```
tab_fmeans <- t(tab_meansPC[,-1])
pander(tab_fmeans)
```

	2	1
PC1	0.736	-0.2595
PC2	0.06264	-0.02209
PC3	0.05368	-0.01892
PC4	0.2555	-0.09007
PC5	0.128	-0.04513
PC6	0.09218	-0.0325
PC7	0.1278	-0.04505
PC8	-0.06346	0.02237
PC9	-0.004008	0.001413

```
colnames(tab_fmeans) <- t(as.vector(tab_meansPC[1]))
pander(tab_fmeans)
```

	Night	Day
PC1	0.736	-0.2595
PC2	0.06264	-0.02209
PC3	0.05368	-0.01892

	Night	Day
PC4	0.2555	-0.09007
PC5	0.128	-0.04513
PC6	0.09218	-0.0325
PC7	0.1278	-0.04505
PC8	-0.06346	0.02237
PC9	-0.004008	0.001413

```
# The third column here represents no specified value of Day or Night, it will be taken care of later a

# Standard deviations of scores for all the PC's classified by Sunrise_Sunset

tab_sdsPC <- aggregate(mvatyp_pca[,2:10],by=list(Sunrise_Sunset=Mva$Sunrise_Sunset),sd)
tab_fsds <- t(tab_sdsPC[,-1])

colnames(tab_fsds) <- t(as.vector(tab_sdsPC[1]))
pander(tab_fsds)
```

	Day	Night
PC1	1.584	1.366
PC2	1.115	1.124
PC3	1.076	1.114
PC4	1.007	1.023
PC5	0.9733	0.9419
PC6	0.898	0.9023
PC7	0.8585	0.8663
PC8	0.7099	0.6732
PC9	0.06658	0.07143

```
t.test(PC1~Mva$Sunrise_Sunset,data=mvatyp_pca)
```

```
##
##  Welch Two Sample t-test
##
## data:  PC1 by Mva$Sunrise_Sunset
## t = -131.06, df = 95987, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.0104161 -0.9806401
## sample estimates:
##   mean in group Day mean in group Night
##          -0.2595089           0.7360192
```

```
t.test(PC2~Mva$Sunrise_Sunset,data=mvatyp_pca)
```

```
##
##  Welch Two Sample t-test
##
## data:  PC2 by Mva$Sunrise_Sunset
```

```
## t = -14.184, df = 82966, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.09643839 -0.07302155
## sample estimates:
##   mean in group Day mean in group Night
##         -0.02208695          0.06264302
```

```r
t.test(PC3~Mva$Sunrise_Sunset,data=mvatyp_pca)
```

```
##
##  Welch Two Sample t-test
##
## data:  PC3 by Mva$Sunrise_Sunset
## t = -12.343, df = 81087, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.08412842 -0.06107154
## sample estimates:
##   mean in group Day mean in group Night
##         -0.01892497          0.05367501
```

```r
t.test(PC4~Mva$Sunrise_Sunset,data=mvatyp_pca)
```

```
##
##  Welch Two Sample t-test
##
## data:  PC4 by Mva$Sunrise_Sunset
## t = -63.665, df = 82414, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.3561837 -0.3349077
## sample estimates:
##   mean in group Day mean in group Night
##         -0.09007498          0.25547073
```

```r
t.test(PC5~Mva$Sunrise_Sunset,data=mvatyp_pca)
```

```
##
##  Welch Two Sample t-test
##
## data:  PC5 by Mva$Sunrise_Sunset
## t = -34.2, df = 86003, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1830388 -0.1631961
## sample estimates:
##   mean in group Day mean in group Night
##         -0.04512732          0.12799013
```

```r
t.test(PC6~Mva$Sunrise_Sunset,data=mvatyp_pca)
```

```
## 
##  Welch Two Sample t-test
## 
## data:  PC6 by Mva$Sunrise_Sunset
## t = -25.969, df = 83183, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1340878 -0.1152682
## sample estimates:
##   mean in group Day mean in group Night 
##         -0.03250038          0.09217761
```

```r
t.test(PC7~Mva$Sunrise_Sunset,data=mvatyp_pca)
```

```
## 
##  Welch Two Sample t-test
## 
## data:  PC7 by Mva$Sunrise_Sunset
## t = -37.536, df = 82877, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1818450 -0.1637967
## sample estimates:
##   mean in group Day mean in group Night 
##          -0.0450500           0.1277708
```

```r
t.test(PC8~Mva$Sunrise_Sunset,data=mvatyp_pca)
```

```
## 
##  Welch Two Sample t-test
## 
## data:  PC8 by Mva$Sunrise_Sunset
## t = 23.593, df = 87617, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.07870404 0.09296544
## sample estimates:
##   mean in group Day mean in group Night 
##          0.02237493         -0.06345980
```

```r
t.test(PC9~Mva$Sunrise_Sunset,data=mvatyp_pca)
```

```
## 
##  Welch Two Sample t-test
## 
## data:  PC9 by Mva$Sunrise_Sunset
## t = 14.498, df = 78733, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

```
##  0.004688522 0.006154374
## sample estimates:
##   mean in group Day mean in group Night
##         0.001413234        -0.004008214
```

```r
# F ratio tests

var.test(PC1~Mva$Sunrise_Sunset,data=mvatyp_pca)
```

```
##
##  F test to compare two variances
##
## data:  PC1 by Mva$Sunrise_Sunset
## F = 1.3454, num df = 135178, denom df = 47661, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.325621 1.365353
## sample estimates:
## ratio of variances
##           1.345376
```

```r
var.test(PC2~Mva$Sunrise_Sunset,data=mvatyp_pca)
```

```
##
##  F test to compare two variances
##
## data:  PC2 by Mva$Sunrise_Sunset
## F = 0.98447, num df = 135178, denom df = 47661, p-value = 0.0374
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.9700143 0.9990883
## sample estimates:
## ratio of variances
##            0.98447
```

```r
var.test(PC3~Mva$Sunrise_Sunset,data=mvatyp_pca)
```

```
##
##  F test to compare two variances
##
## data:  PC3 by Mva$Sunrise_Sunset
## F = 0.93312, num df = 135178, denom df = 47661, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.9194196 0.9469770
## sample estimates:
## ratio of variances
##          0.9331213
```

```r
var.test(PC4~Mva$Sunrise_Sunset,data=mvatyp_pca)
```

```
##
##  F test to compare two variances
##
## data:  PC4 by Mva$Sunrise_Sunset
## F = 0.96938, num df = 135178, denom df = 47661, p-value = 3.479e-05
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.9551432 0.9837714
## sample estimates:
## ratio of variances
##          0.9693773
```

```
var.test(PC5~Mva$Sunrise_Sunset,data=mvatyp_pca)
```

```
##
##  F test to compare two variances
##
## data:  PC5 by Mva$Sunrise_Sunset
## F = 1.0677, num df = 135178, denom df = 47661, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.052024 1.083556
## sample estimates:
## ratio of variances
##           1.067702
```

```
var.test(PC6~Mva$Sunrise_Sunset,data=mvatyp_pca)
```

```
##
##  F test to compare two variances
##
## data:  PC6 by Mva$Sunrise_Sunset
## F = 0.99041, num df = 135178, denom df = 47661, p-value = 0.2001
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.9758677 1.0051170
## sample estimates:
## ratio of variances
##          0.9904106
```

```
var.test(PC7~Mva$Sunrise_Sunset,data=mvatyp_pca)
```

```
##
##  F test to compare two variances
##
## data:  PC7 by Mva$Sunrise_Sunset
## F = 0.98203, num df = 135178, denom df = 47661, p-value = 0.01588
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.9676106 0.9966125
## sample estimates:
## ratio of variances
##          0.9820304
```

```r
var.test(PC8~Mva$Sunrise_Sunset,data=mvatyp_pca)
```

```
##
##  F test to compare two variances
##
## data:  PC8 by Mva$Sunrise_Sunset
## F = 1.1121, num df = 135178, denom df = 47661, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.095789 1.128632
## sample estimates:
## ratio of variances
##           1.112119
```

```r
var.test(PC9~Mva$Sunrise_Sunset,data=mvatyp_pca)
```

```
##
##  F test to compare two variances
##
## data:  PC9 by Mva$Sunrise_Sunset
## F = 0.86894, num df = 135178, denom df = 47661, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.8561807 0.8818427
## sample estimates:
## ratio of variances
##          0.8689399
```

```r
# Levene's tests (one-sided)

library(car)
```

```
## Loading required package: carData
```

```r
(LT_PC1 <- leveneTest(PC1~Mva$Sunrise_Sunset,data=mvatyp_pca))
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##           Df F value    Pr(>F)
## group      1  960.23 < 2.2e-16 ***
##       182839
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
(LT_PC2 <- leveneTest(PC2~Mva$Sunrise_Sunset,data=mvatyp_pca))
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##           Df F value   Pr(>F)
## group      1  38.237 6.28e-10 ***
##       182839
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(LT_PC3 <- leveneTest(PC3~Mva$Sunrise_Sunset,data=mvatyp_pca))
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##           Df F value    Pr(>F)
## group      1  109.82 < 2.2e-16 ***
##       182839
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(LT_PC4 <- leveneTest(PC4~Mva$Sunrise_Sunset,data=mvatyp_pca))
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##           Df F value Pr(>F)
## group      1   1e-04 0.9911
##       182839
```

```
(LT_PC5 <- leveneTest(PC5~Mva$Sunrise_Sunset,data=mvatyp_pca))
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##           Df F value    Pr(>F)
## group      1   30.86 2.777e-08 ***
##       182839
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(LT_PC6 <- leveneTest(PC6~Mva$Sunrise_Sunset,data=mvatyp_pca))
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##           Df F value    Pr(>F)
## group      1  11.636 0.0006468 ***
##       182839
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(LT_PC7 <- leveneTest(PC7~Mva$Sunrise_Sunset,data=mvatyp_pca))
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##           Df F value    Pr(>F)
## group      1  159.92 < 2.2e-16 ***
##       182839
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(LT_PC8 <- leveneTest(PC8~Mva$Sunrise_Sunset,data=mvatyp_pca))
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##           Df F value    Pr(>F)
## group      1  42.863 5.887e-11 ***
##       182839
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(LT_PC9 <- leveneTest(PC9~Mva$Sunrise_Sunset,data=mvatyp_pca))
```

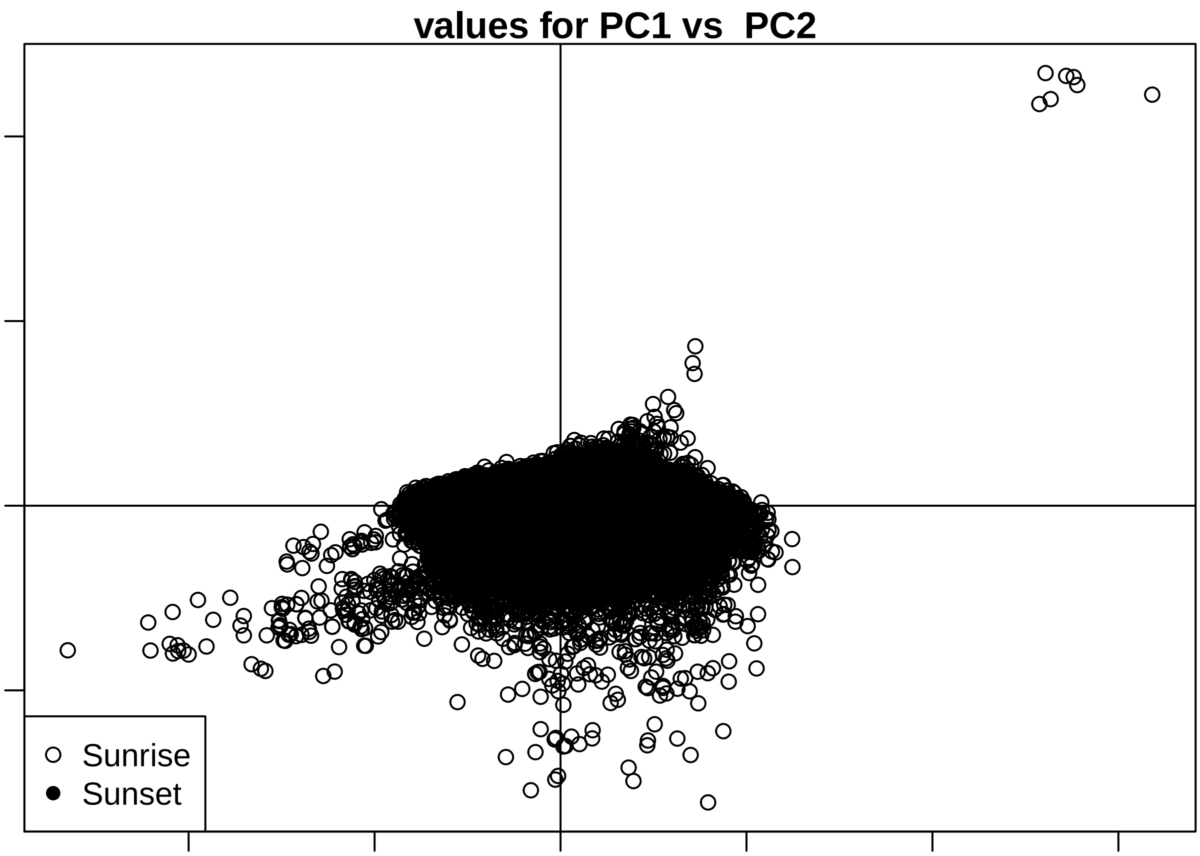
```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##           Df F value    Pr(>F)
## group      1  178.89 < 2.2e-16 ***
##       182839
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

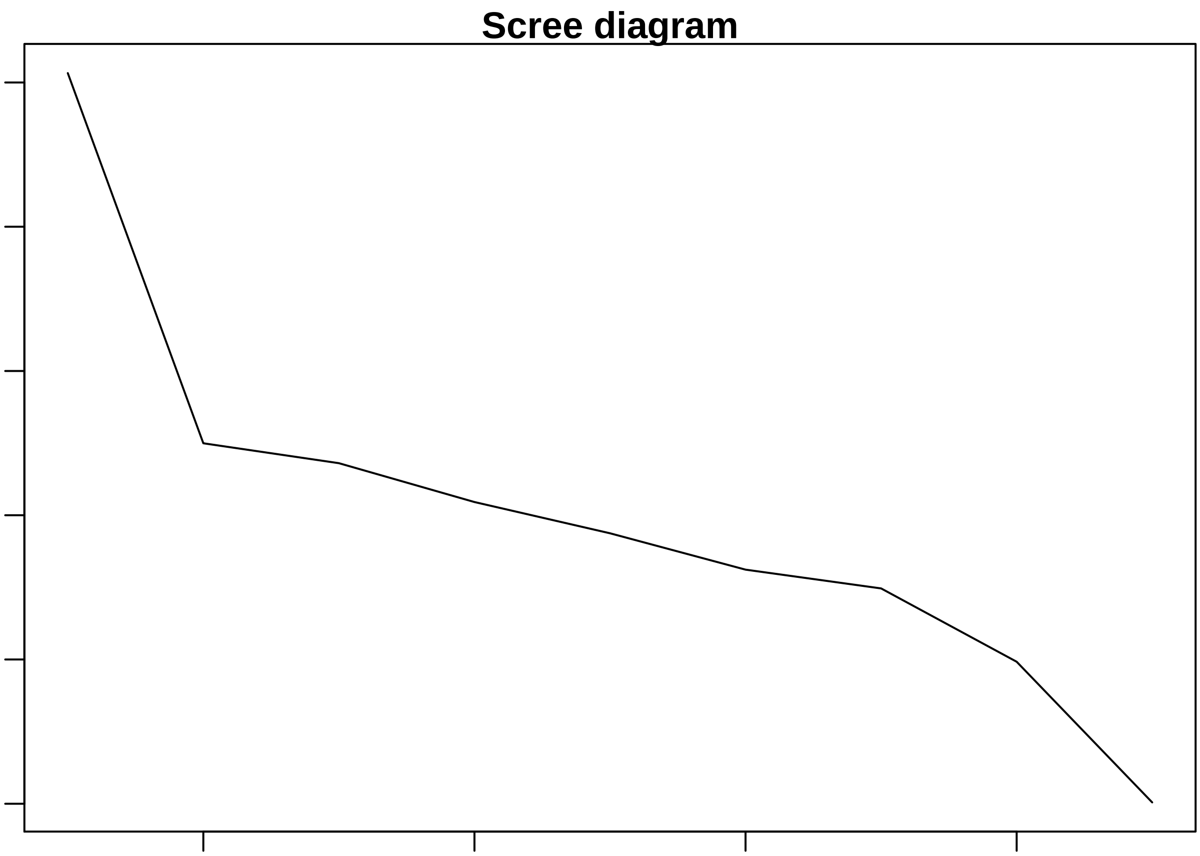
```
# Plotting the scores for the first and second components
par("mar")
```

```
## [1] 5.1 4.1 4.1 2.1
```

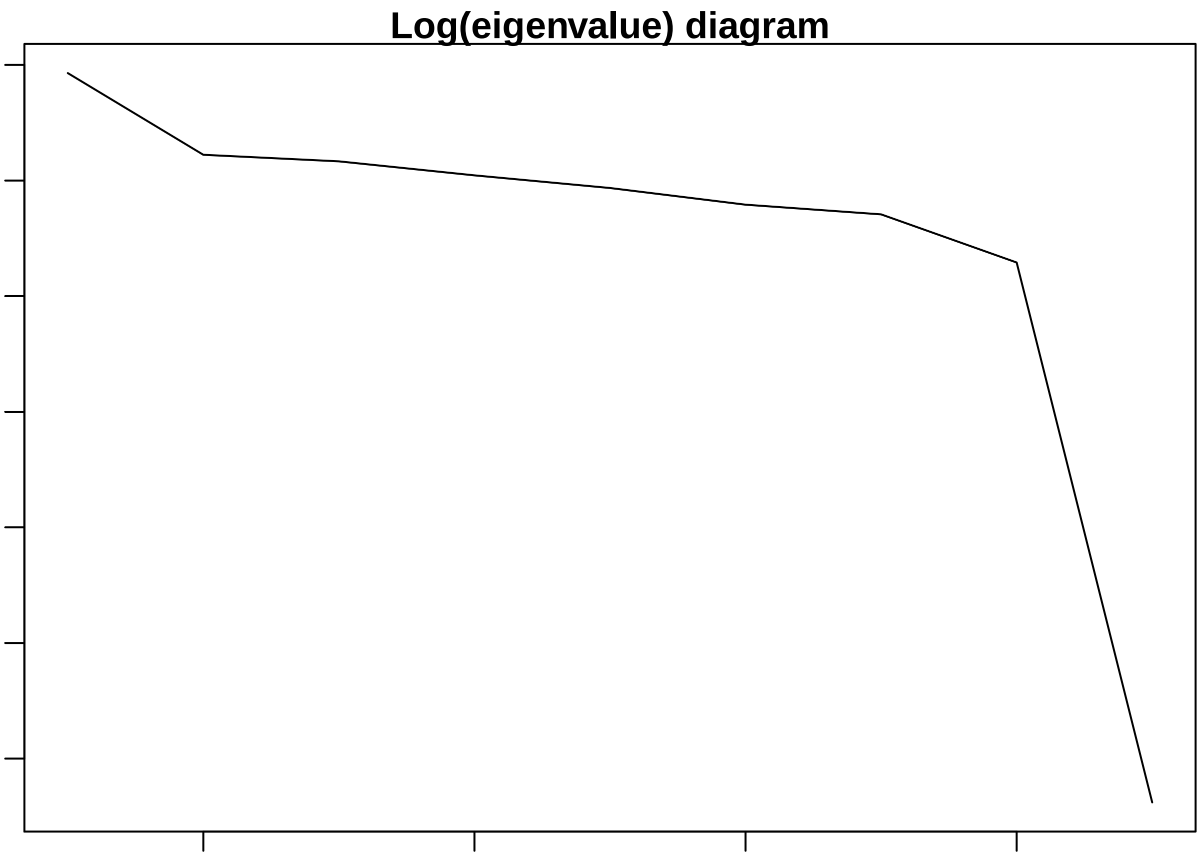
```
par(mar=c(1,1,1,1))
plot(mvatyp_pca$PC1, mvatyp_pca$PC2,xlab="PC1", ylab="PC2", main=" values for PC1 vs  PC2")
abline(h=0)
abline(v=0)
legend("bottomleft", legend=c("Sunrise","Sunset"), pch=c(1,16))
```




```
plot(eigen_mva, xlab = "Component number", ylab = "Component variance", type = "l", main = "Scree diagra
```



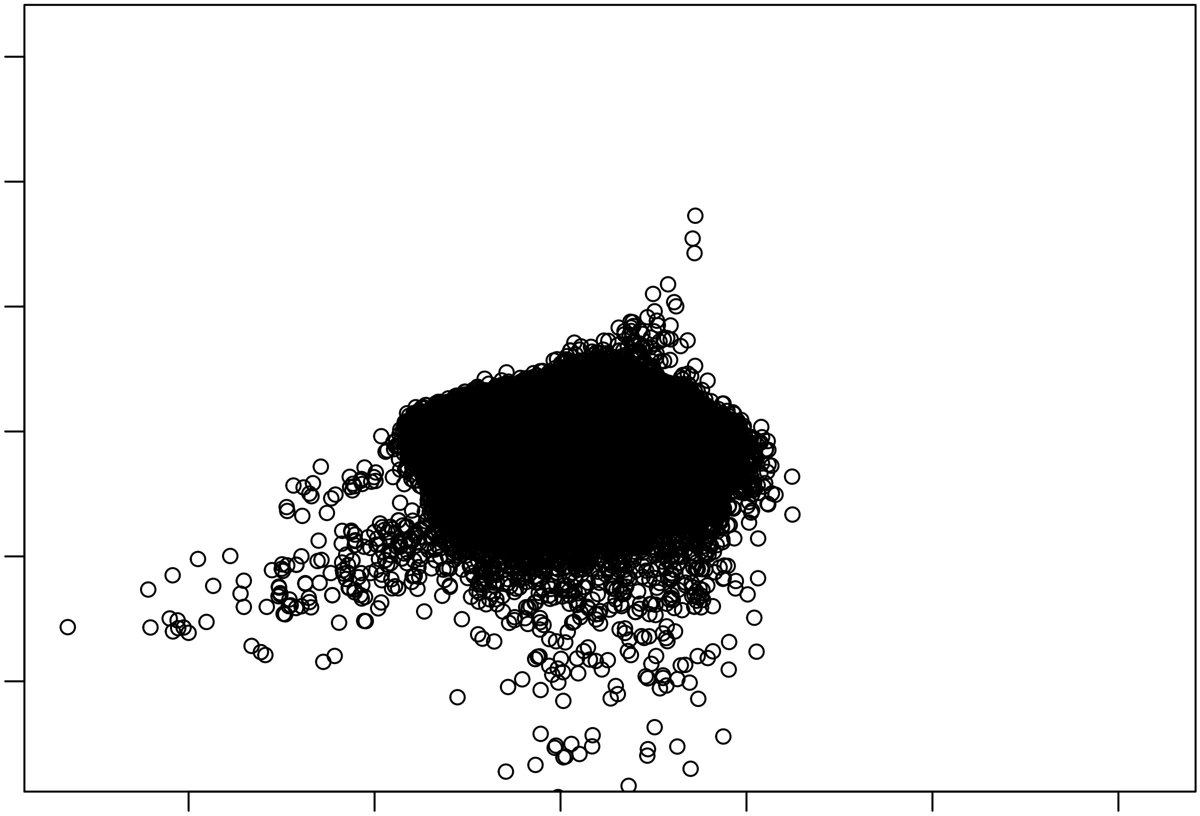

```
plot(log(eigen_mva), xlab = "Component number",ylab = "log(Component variance)", type="l",main = "Log(ei
```




```
pander(diag(cov(mva_pca$x)))
```

PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9
2.533	1.249	1.18	1.046	0.9374	0.8115	0.7463	0.4921	0.004613

```
xlim <- range(mva_pca$x[,1])
plot(mva_pca$x,xlim=xlim,ylim=xlim)
```



```
# So after conducting the principal component analysis , we can now interpret the results for dimension
# we can see from the graphs and the scree plots which are the principal components of our dataset amd
```