Discriminant Analysis

Nadia Ahmad

2023-02-18

PCA Analysis on Travel Discrimination

Dataset explanation: 1. Dependent Variable: * Q1 = travel frequency 2. Independent Variables: * Q6__15 = Checkin experience rate * Q6__16 = Bag drop off experience rate
* Q6__17 = Security line experience rate * Q6__18 = Boarding airplane experience rate
* Q6__19 = Travel experience compared to other travelers rate * Q14 = Age Group * Q15 = Gender * Q16 = US citizenship * Q17 = Race
Library

```
library(readr)
library(tidyverse)
library(XML)
library(corrplot)
library(factoextra)
library(MASS)
library(mvtnorm)
library(MVN)
library(psych)
library(ggfortify)
library(ggpubr)
library(mvoutlier)
library(heplots)
library(e1071)
library(caret)
library(klaR)
library(candisc)
library(caTools)
library(DMwR2)
library(class)
```

Read the dataset

```
travel_df <- read_csv("Travel Study 2.14.23.csv")
head(travel_df)
```

```
## # A tibble: 6 x 89
##   Start~1 EndDate Status IPAdd~2 Progr~3 Durat~4 Finis~5 Recor~6 Respo~7 Recip~8
##   <chr>   <chr>   <chr>  <chr>   <chr>   <chr>   <chr>   <chr>   <chr>   <chr>
## 1 "Start~ "End D~ "Resp~ "IP Ad~ "Progr~ "Durat~ "Finis~ "Recor~ "Respo~ "Recip~
## 2 "{\"Im~ "{\"Im~ "{\"I~ "{\"Im~ "{\"Im~ "{\"Im~ "{\"Im~ "{\"Im~ "{\"Im~ "{\"Im~
## 3 "7/4/2~ "7/4/2~ "0"    "72.31~ "100"   "177"   "1"     "7/4/2~ "R_33D~  <NA>
```

```
## 4 "7/4/2~ "7/4/2~ "0"    "172.5~ "100"   "174"   "1"     "7/4/2~ "R_32K~  <NA>
## 5 "7/4/2~ "7/4/2~ "0"    "174.2~ "100"   "570"   "1"     "7/4/2~ "R_1K2~  <NA>
## 6 "7/4/2~ "7/4/2~ "0"    "72.18~ "100"   "256"   "1"     "7/4/2~ "R_2zB~  <NA>
## # ... with 79 more variables: RecipientFirstName <chr>, RecipientEmail <chr>,
## #   ExternalReference <chr>, LocationLatitude <chr>, LocationLongitude <chr>,
## #   DistributionChannel <chr>, UserLanguage <chr>, `Text / Graphic` <chr>,
## #   Q1 <chr>, Q2 <chr>, Q3 <chr>, Q4 <chr>, Q5 <chr>, Q6_15 <chr>, Q6_16 <chr>,
## #   Q6_17 <chr>, Q6_18 <chr>, Q6_19 <chr>, Q7_0_GROUP <chr>, Q7_1_GROUP <chr>,
## #   Q7_2_GROUP <chr>, Q7_0_1_RANK <chr>, Q7_0_2_RANK <chr>, Q7_0_3_RANK <chr>,
## #   Q7_0_4_RANK <chr>, Q7_0_5_RANK <chr>, Q7_0_6_RANK <chr>, ...
```

Dataset contains 230 rows and 89 columns which is still messy. Thus, we'll conduct some data preprocessing steps.

DATA PREPROCESSING

```
# First, drop two first rows. Next, filter only data that has 100 in progress
travel_df <- travel_df %>%
  slice(-c(1,2)) %>%
  filter(Progress == '100')

# Drop the first 11 columns since it contains the questionnaire status
travel_df_clean <- travel_df[-c(1:18)]

# Drop all column that contains _RANK in the end of the name
travel_df_clean <- travel_df_clean[!grepl("_RANK$", names(travel_df_clean))]

# Drop optional column named Q20 and column contains _TEXT in the end of name
travel_df_clean <- travel_df_clean[!grepl("_TEXT$", names(travel_df_clean))]
```

```
# Select used columns
travel_df_clean <- subset(travel_df_clean, select = c(Q1, Q6_15, Q6_16,
                                                      Q6_17, Q6_18, Q6_19,
                                                      Q14, Q15, Q16, Q17))
```

```
# CHECK MISSING VALUE----
# Count the missing values by column wise
print("Count of missing values by column wise")
```

```
## [1] "Count of missing values by column wise"
```

```
sapply(travel_df_clean, function(x) sum(is.na(x)))
```

```
##    Q1 Q6_15 Q6_16 Q6_17 Q6_18 Q6_19   Q14   Q15   Q16   Q17
##     0     4     8     6     6     8     1     1     1     1
```

```
# Missing value imputation
# Since our data contains 46 missing value, let's impute with mode
# Function to see mode
calc_mode <- function(x){

  # List the distinct / unique values
  distinct_values <- unique(na.omit(x))

  # Count the occurrence of each distinct value
  distinct_tabulate <- tabulate(match(x, distinct_values))
```

```r
  # Return the value with the highest occurrence
  distinct_values[which.max(distinct_tabulate)]
}

# Impute missing value----
travel_df_clean <- travel_df_clean %>%
  mutate(across(everything(), ~replace_na(.x, calc_mode(.x))))
```

```r
# CONVERT DATA TYPE----
# Convert all variables into integer
# Convert column 2 to 6 to numeric
travel_df_clean[,1:10] <- sapply(travel_df_clean[,1:10], as.integer)
travel_df_clean[,2:6] <- sapply(travel_df_clean[,2:6], as.numeric)
head(travel_df_clean)
```

```
## # A tibble: 6 x 10
##      Q1 Q6_15 Q6_16 Q6_17 Q6_18 Q6_19   Q14   Q15   Q16   Q17
##   <int> <dbl> <dbl> <dbl> <dbl> <dbl> <int> <int> <int> <int>
## 1     4    54    50    71    50    51     2     1     1     6
## 2     4    52    52    51    53    52     4     2     1     1
## 3     3    50    50    50    50    50     7     2     1     6
## 4     4    51    53    54    52    57     3     1     1     6
## 5     4    48   100   100   100   100     3     2     2     6
## 6     4    50    50    50    50    50     7     2     1     2
```

```r
# Rename column name
travel_df_clean <- travel_df_clean %>%
       rename(travel_frequency = 1, checkin_exp = 2,
              baggage_exp = 3, security_exp = 4, boarding_exp =5,
              travel_exp = 6, age = 7, gender =8, citizenship = 9,
              race = 10)
head(travel_df_clean)
```

```
## # A tibble: 6 x 10
##   travel_fr~1 check~2 bagga~3 secur~4 board~5 trave~6   age gender citiz~7  race
##         <int>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <int>  <int>   <int> <int>
## 1           4      54      50      71      50      51     2      1       1     6
## 2           4      52      52      51      53      52     4      2       1     1
## 3           3      50      50      50      50      50     7      2       1     6
## 4           4      51      53      54      52      57     3      1       1     6
## 5           4      48     100     100     100     100     3      2       2     6
## 6           4      50      50      50      50      50     7      2       1     2
## # ... with abbreviated variable names 1: travel_frequency, 2: checkin_exp,
## #   3: baggage_exp, 4: security_exp, 5: boarding_exp, 6: travel_exp,
## #   7: citizenship
```

EXPLORATORY DATA ANALYSIS

1. Summary Statistics

```r
mvn(travel_df_clean, univariatePlot = "qq")
```

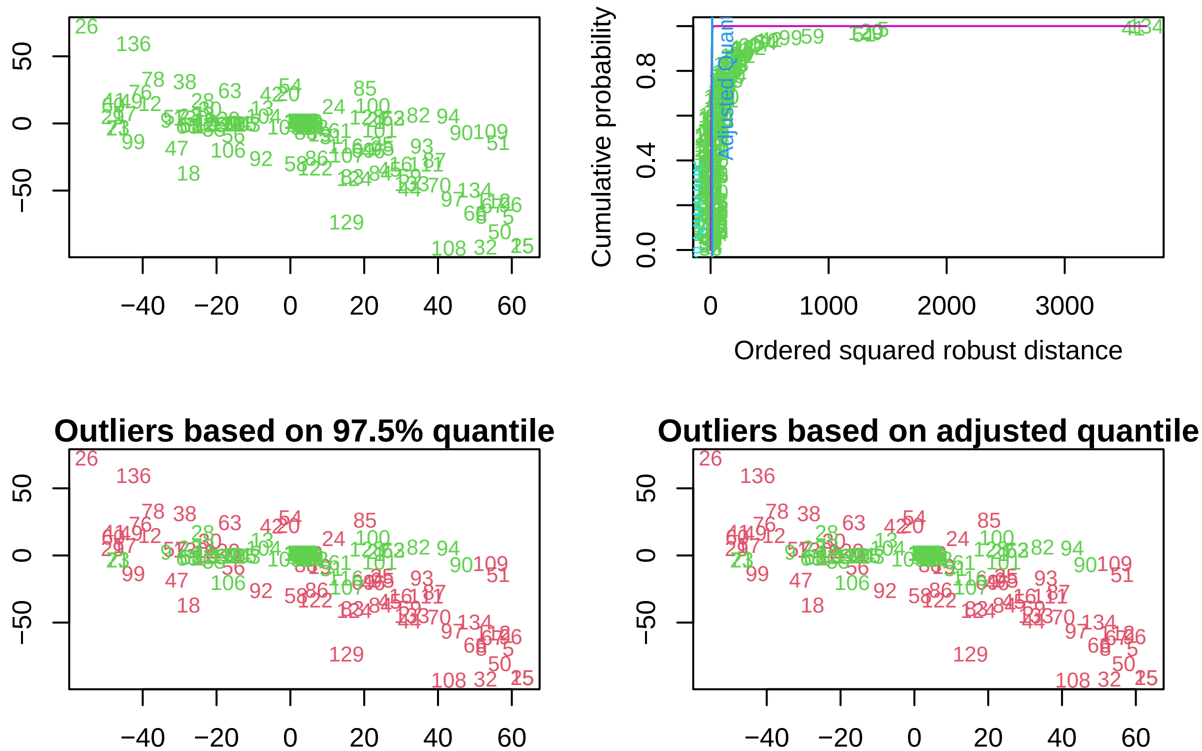
```
## $multivariateNormality
##            Test       HZ p value MVN
## 1 Henze-Zirkler 2.053604       0  NO
```

```
##
## $univariateNormality
##               Test         Variable Statistic   p value Normality
## 1  Anderson-Darling travel_frequency   14.4161  <0.001      NO
## 2  Anderson-Darling   checkin_exp       8.1987  <0.001      NO
## 3  Anderson-Darling   baggage_exp      11.4048  <0.001      NO
## 4  Anderson-Darling   security_exp      1.7367   2e-04      NO
## 5  Anderson-Darling   boarding_exp      9.7366  <0.001      NO
## 6  Anderson-Darling    travel_exp      12.4107  <0.001      NO
## 7  Anderson-Darling       age           5.5692  <0.001      NO
## 8  Anderson-Darling      gender        17.1722  <0.001      NO
## 9  Anderson-Darling   citizenship      32.1842  <0.001      NO
## 10 Anderson-Darling       race         12.9843  <0.001      NO
##
## $Descriptives
##                    n      Mean    Std.Dev Median Min Max 25th  75th
## travel_frequency 136  3.294118  0.6338319      3   2   4    3  4.00
## checkin_exp      136 54.448529 19.7258319     50   0 100   50 60.00
## baggage_exp      136 56.272059 16.8709555     50   0 100   50 60.00
## security_exp     136 50.345588 26.2431307     50   0 100   30 70.00
## boarding_exp     136 55.691176 18.2227024     50   0 100   50 60.25
## travel_exp       136 56.250000 17.1911907     50   5 100   50 60.00
## age              136  3.852941  1.6712356      4   2   7    2  5.00
## gender           136  1.566176  0.6520873      2   1   5    1  2.00
## citizenship      136  1.352941  1.0578063      1   1   7    1  1.00
## race             136  5.102941  2.4533557      6   1  11    2  6.00
##                         Skew   Kurtosis
## travel_frequency -0.326729731 -0.7155349
## checkin_exp       0.053050339  1.0952834
## baggage_exp       0.739915663  1.4220413
## security_exp     -0.008747504 -0.6426500
## boarding_exp      0.372732355  1.4279178
## travel_exp        0.658496427  1.7247166
## age               0.382441118 -1.1282366
## gender            1.983934918  8.6788556
## citizenship       4.485652051 20.9228723
## race              0.013401814 -0.5532205
```

2. Detecting Outliers
We'll look at outlier in first 6 columns in variable.

```
aq.plot(travel_df_clean[,1:6])
```

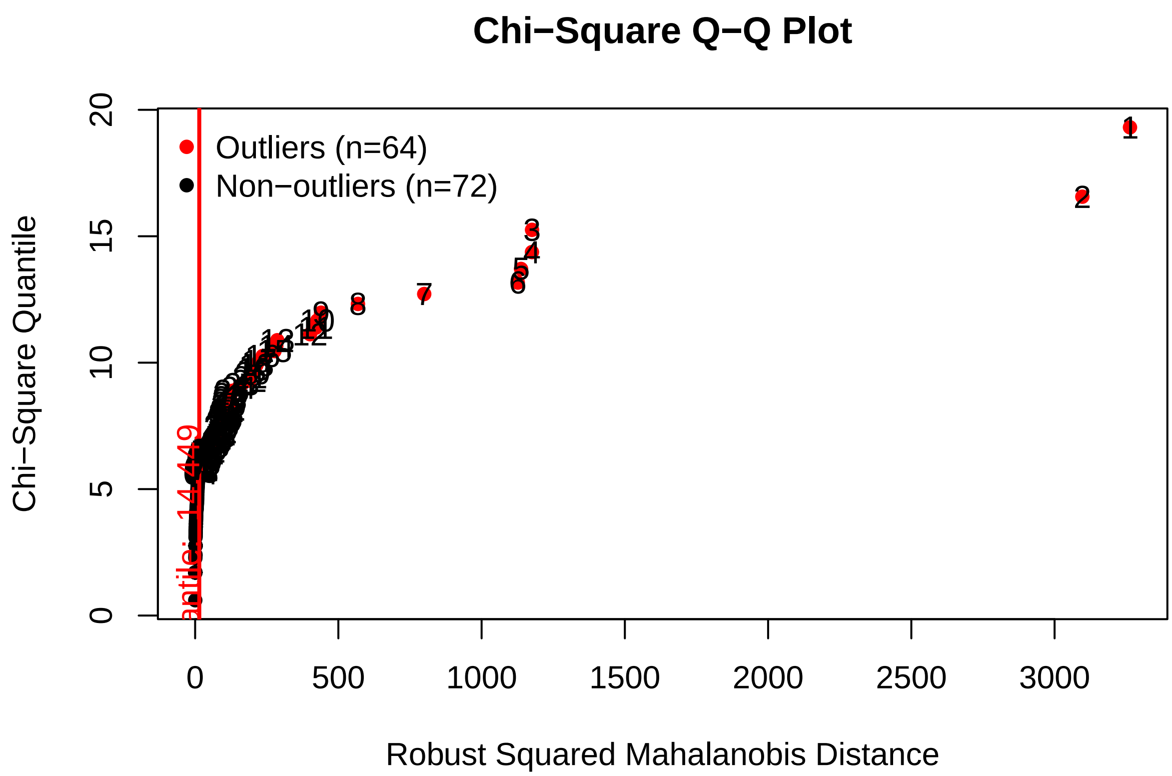
```
## Projection to the first and second robust principal components.
## Proportion of total variation (explained variance): 0.7883824
```




```
## $outliers
##   [1] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE
##  [13] FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE
##  [25]  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE
##  [37] FALSE  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE
##  [49]  TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [61] FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE FALSE
##  [73] FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE
##  [85]  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE
##  [97]  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
## [109]  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
## [121] FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE
## [133]  TRUE  TRUE FALSE  TRUE
```

3. QQ-Plot Since we got error in integer variable, `system is exactly singular: U[2,2] = 0`, thus we'll do chi-square quantile plot in numeric (var 1 to 6) only.

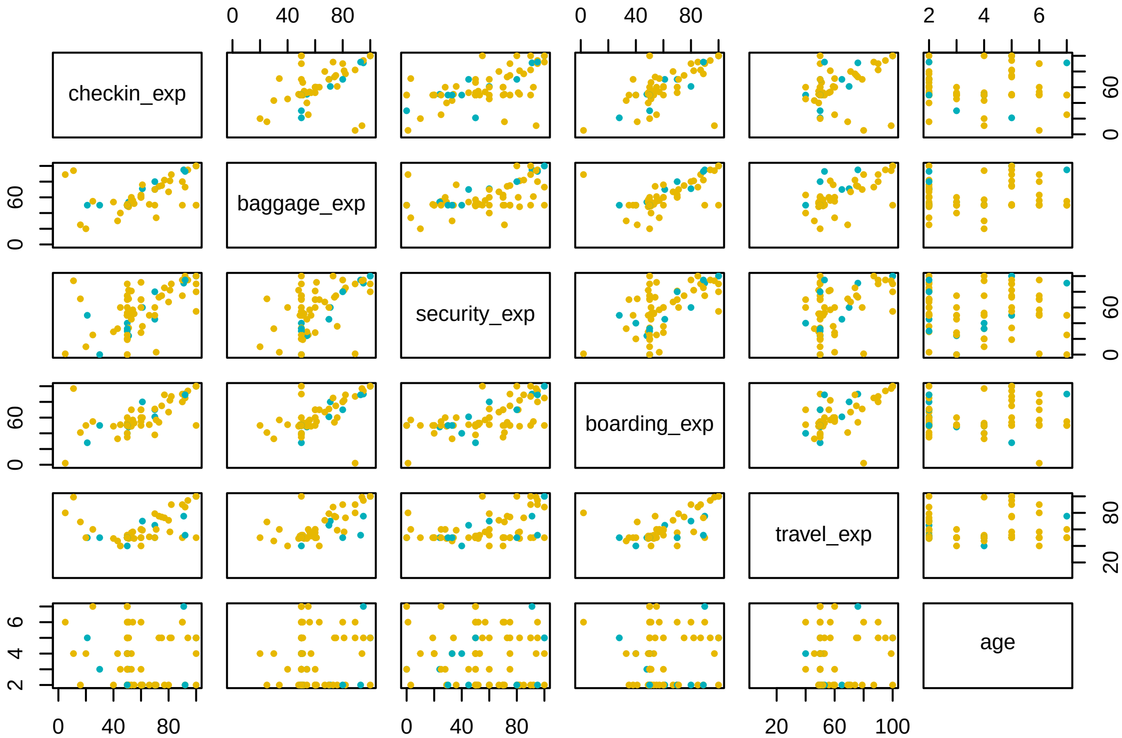
```
mvn(travel_df_clean[,1:6], mvnTest = "hz", multivariatePlot = "scatter",
    multivariateOutlierMethod="quan")
```



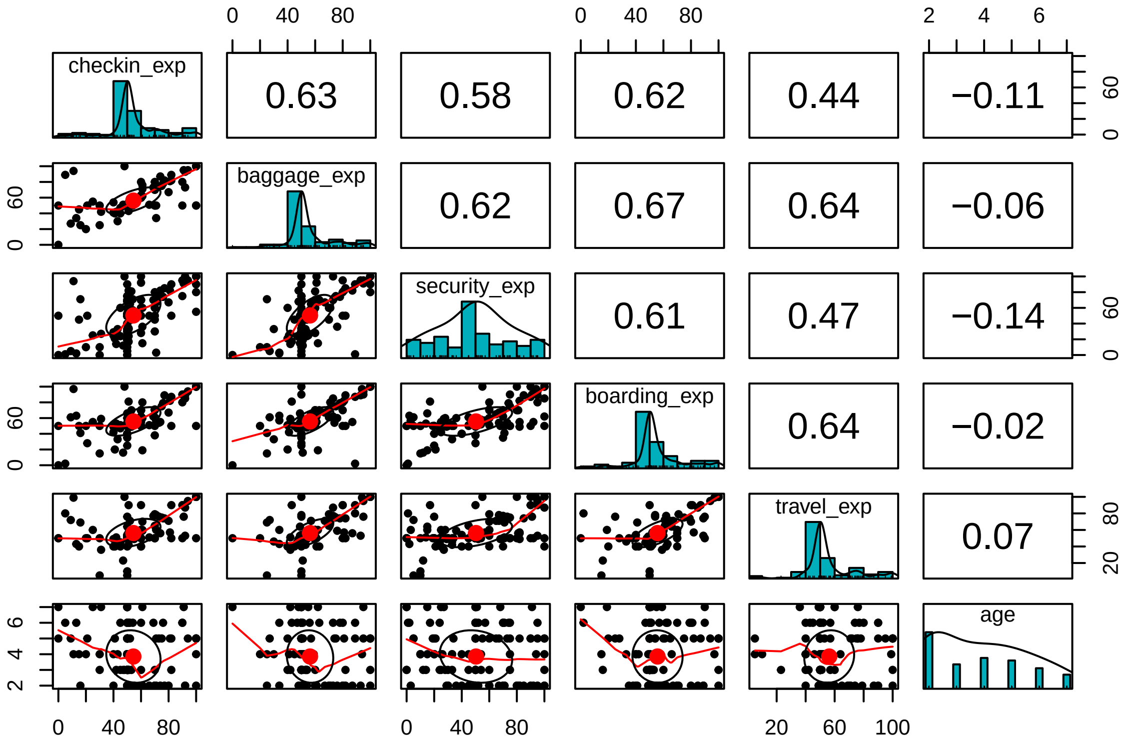

```
## $multivariateNormality
##            Test       HZ p value MVN
## 1 Henze-Zirkler 6.820019       0  NO
##
## $univariateNormality
##               Test          Variable Statistic   p value Normality
## 1 Anderson-Darling travel_frequency   14.4161  <0.001      NO
## 2 Anderson-Darling   checkin_exp       8.1987  <0.001      NO
## 3 Anderson-Darling   baggage_exp      11.4048  <0.001      NO
## 4 Anderson-Darling   security_exp      1.7367   2e-04      NO
## 5 Anderson-Darling   boarding_exp      9.7366  <0.001      NO
## 6 Anderson-Darling    travel_exp      12.4107  <0.001      NO
##
## $Descriptives
##                    n      Mean    Std.Dev Median Min Max 25th  75th
## travel_frequency 136  3.294118  0.6338319      3   2   4    3  4.00
## checkin_exp      136 54.448529 19.7258319     50   0 100   50 60.00
## baggage_exp      136 56.272059 16.8709555     50   0 100   50 60.00
## security_exp     136 50.345588 26.2431307     50   0 100   30 70.00
## boarding_exp     136 55.691176 18.2227024     50   0 100   50 60.25
## travel_exp       136 56.250000 17.1911907     50   5 100   50 60.00
##                          Skew   Kurtosis
## travel_frequency -0.326729731 -0.7155349
## checkin_exp       0.053050339  1.0952834
## baggage_exp       0.739915663  1.4220413
## security_exp     -0.008747504 -0.6426500
## boarding_exp      0.372732355  1.4279178
## travel_exp        0.658496427  1.7247166
```

4. PAIRS PLOT

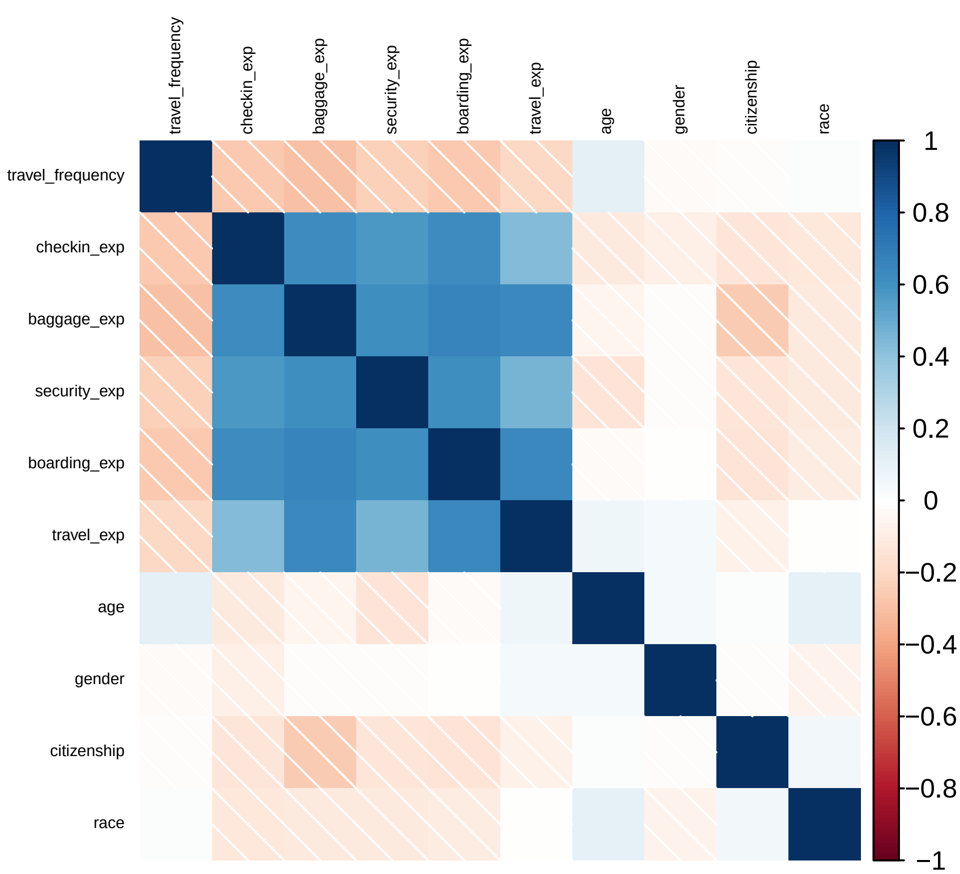
```
my_cols <- c( "#FC4E07","#00AFBB", "#E7B800")
pairs(travel_df_clean[,2:7], pch = 19,  cex = 0.5,
      col = my_cols[travel_df_clean$travel_frequency])
```




```
pairs.panels(travel_df_clean[,2:7],
             method = "pearson", # correlation method
             hist.col = "#00AFBB",
             density = TRUE,  # show density plots
             ellipses = TRUE # show correlation ellipses
             )
```




```r
#
corrplot(cor(travel_df_clean), method = "shade",  tl.col = "black",
         title = "", tl.cex = 0.5)
```



DISCRIMINANT ANALYSIS : Box's M Test

```r
res <- boxM(travel_df_clean[, 2:10], travel_df_clean$travel_frequency)
res
```

```
##
##  Box's M-test for Homogeneity of Covariance Matrices
##
## data:  travel_df_clean[, 2:10]
## Chi-Sq (approx.) = 171.12, df = 90, p-value = 5.497e-07
```

```r
summary(res)
```

```
## Summary for Box's M-test of Equality of Covariance Matrices
##
## Chi-Sq:   171.1248
## df:   90
## p-value: 5.497e-07
##
## log of Covariance determinants:
##        2        3        4   pooled
## 18.27477 29.23418 26.88740 28.88102
##
## Eigenvalues:
##              2            3           4       pooled
## 1 2.789809e+03 1184.2516982 991.7000715 1246.9425150
## 2 2.006836e+02  305.0701065 265.0773167  256.2821763
## 3 1.073063e+02  211.8006628 191.3870988  190.7017421
## 4 3.718787e+01   89.7146485 138.4337134  106.0105372
## 5 1.048402e+01   53.6342078  54.0857507   81.5675913
## 6 2.608863e+00    6.4090639   4.8602248    5.8535852
## 7 1.554174e+00    2.6017449   2.5141851    2.6364432
## 8 1.148517e-01    1.4368029   0.5192163    1.0304644
## 9 7.923229e-03    0.5632593   0.1989103    0.4164921
##
## Statistics based on eigenvalues:
##                      2            3            4       pooled
## product   8.642335e+07 4.968680e+12 4.753894e+11 3.490328e+12
## sum       3.149756e+03 1.855482e+03 1.648776e+03 1.891442e+03
## precision 7.348526e-03 3.278267e-01 1.317075e-01 2.529585e-01
## max       2.789809e+03 1.184252e+03 9.917001e+02 1.246943e+03
```

Since the dataset didn't achive the equal covariance assumption, we need to transform the dataset.

```r
# Box cox transformation
ind <- travel_df_clean[,-1]
ind <- sqrt(ind)
df_new <- cbind(travel_df_clean$travel_frequency, ind)
head(df_new)
```

```
##   travel_df_clean$travel_frequency checkin_exp baggage_exp security_exp
## 1                                4    7.348469    7.071068     8.426150
## 2                                4    7.211103    7.211103     7.141428
```

```
## 3                               3    7.071068    7.071068     7.071068
## 4                               4    7.141428    7.280110     7.348469
## 5                               4    6.928203   10.000000    10.000000
## 6                               4    7.071068    7.071068     7.071068
##   boarding_exp travel_exp      age   gender citizenship     race
## 1     7.071068   7.141428 1.414214 1.000000    1.000000 2.449490
## 2     7.280110   7.211103 2.000000 1.414214    1.000000 1.000000
## 3     7.071068   7.071068 2.645751 1.414214    1.000000 2.449490
## 4     7.211103   7.549834 1.732051 1.000000    1.000000 2.449490
## 5    10.000000  10.000000 1.732051 1.414214    1.414214 2.449490
## 6     7.071068   7.071068 2.645751 1.414214    1.000000 1.414214
```

```r
# Rename dependent variable
df_new <- df_new %>%
       rename(travel_frequency = 1)
```

```r
# Test Box's M again
res2 <- boxM(df_new[, 2:10], df_new$travel_frequency)
res2
```

```
##
##  Box's M-test for Homogeneity of Covariance Matrices
##
## data:  df_new[, 2:10]
## Chi-Sq (approx.) = 183.93, df = 90, p-value = 2.039e-08
```

```r
summary(res2)
```

```
## Summary for Box's M-test of Equality of Covariance Matrices
##
## Chi-Sq:   183.934
## df:   90
## p-value: 2.039e-08
##
## log of Covariance determinants:
##          2          3          4     pooled
## -16.798212  -6.924692  -7.284162  -6.286623
##
## Eigenvalues:
##              2          3          4     pooled
## 1 13.279489505 6.53094201 8.37630929 7.77236985
## 2  1.948864692 1.88663543 2.21911124 1.78896453
## 3  0.695580663 1.16747483 1.39673455 1.31711897
## 4  0.371598461 0.55155820 1.22084811 0.70565371
## 5  0.120836397 0.37459286 0.35837003 0.54030182
## 6  0.073275399 0.25443962 0.28446667 0.33661708
## 7  0.047120051 0.17007796 0.16122557 0.17069483
## 8  0.013708598 0.11445292 0.04395433 0.08139106
## 9  0.001324006 0.06679171 0.02997248 0.05699184
##
## Statistics based on eigenvalues:
##                      2            3             4       pooled
## product   5.065580e-08 9.832057e-04   0.000686323   0.001861033
```

```
## sum       1.655180e+01 1.111697e+01 14.090992262 12.770103704
## precision 1.141432e-03 2.529000e-02  0.014138640  0.023053854
## max       1.327949e+01 6.530942e+00  8.376309291  7.772369853
```

```
# Convert the travel_frequency to factor
df_new$travel_frequency <- as.factor(df_new$travel_frequency)
```

DISCRIMINANT ANALYSIS: Linear LDA

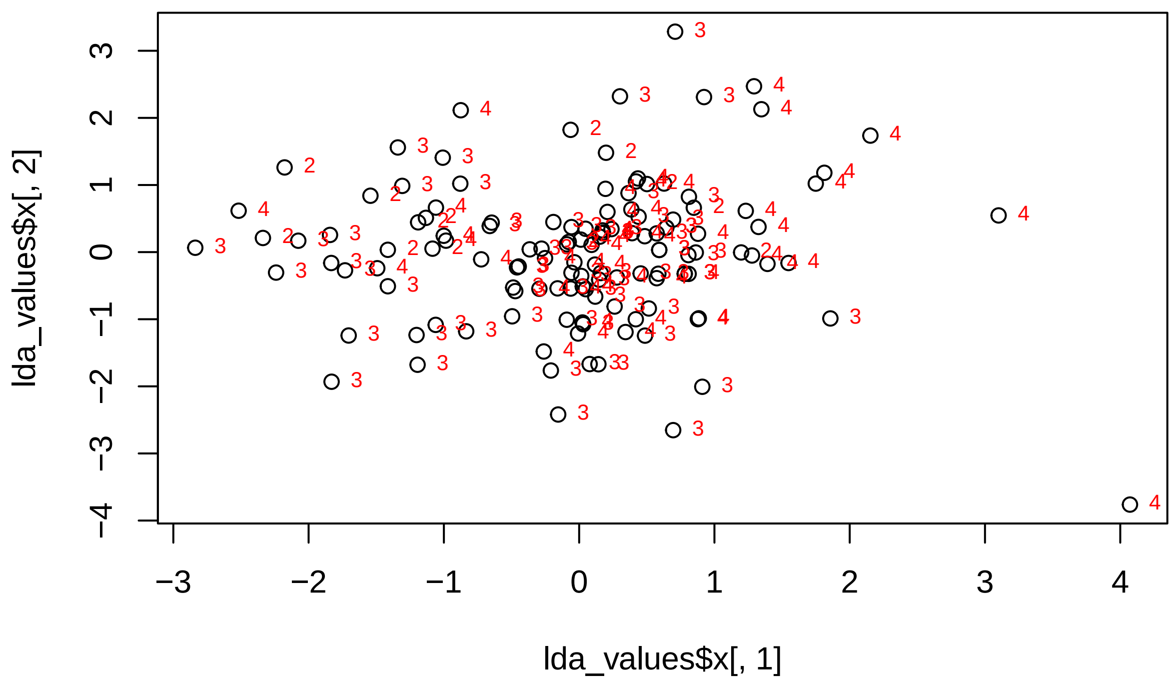
```
lda_model <- lda(travel_frequency ~., data = df_new)
lda_model
```

```
## Call:
## lda(travel_frequency ~ ., data = df_new)
##
## Prior probabilities of groups:
##          2          3          4
## 0.09558824 0.51470588 0.38970588
##
## Group means:
##   checkin_exp baggage_exp security_exp boarding_exp travel_exp      age
## 2    7.852995    8.294713     7.092386     7.984569   7.782902 1.837625
## 3    7.437929    7.487044     7.111951     7.499173   7.586359 1.883587
## 4    6.734485    7.073375     6.123599     6.954934   7.067300 1.978919
##     gender citizenship     race
## 2 1.191175    1.063725 2.270467
## 3 1.248340    1.147458 2.124357
## 4 1.211014    1.101390 2.231751
##
## Coefficients of linear discriminants:
##                      LD1         LD2
## checkin_exp  -0.20069507 -0.20650029
## baggage_exp  -0.40145970  0.89934925
## security_exp  0.01262291 -0.28836682
## boarding_exp -0.11011986  0.11282853
## travel_exp   -0.26989173 -0.33685062
## age           0.67987633 -0.01198983
## gender       -0.55932396 -1.43407268
## citizenship  -1.09422297 -0.56563158
## race         -0.10025985  0.64101934
##
## Proportion of trace:
##    LD1    LD2
## 0.6549 0.3451
```

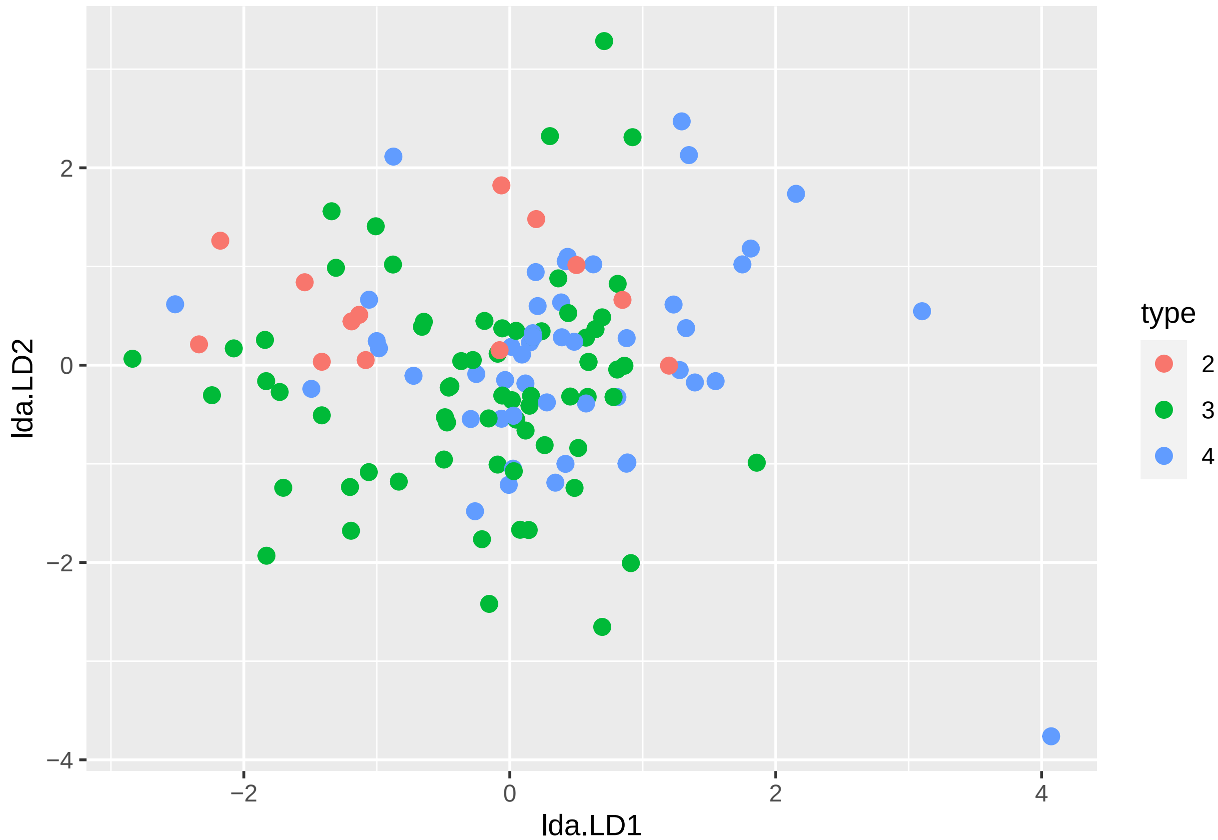
The percentage separation achieved by each discriminant function is 73.8% and 26.2% respectively.

Scatter plot for discriminant function

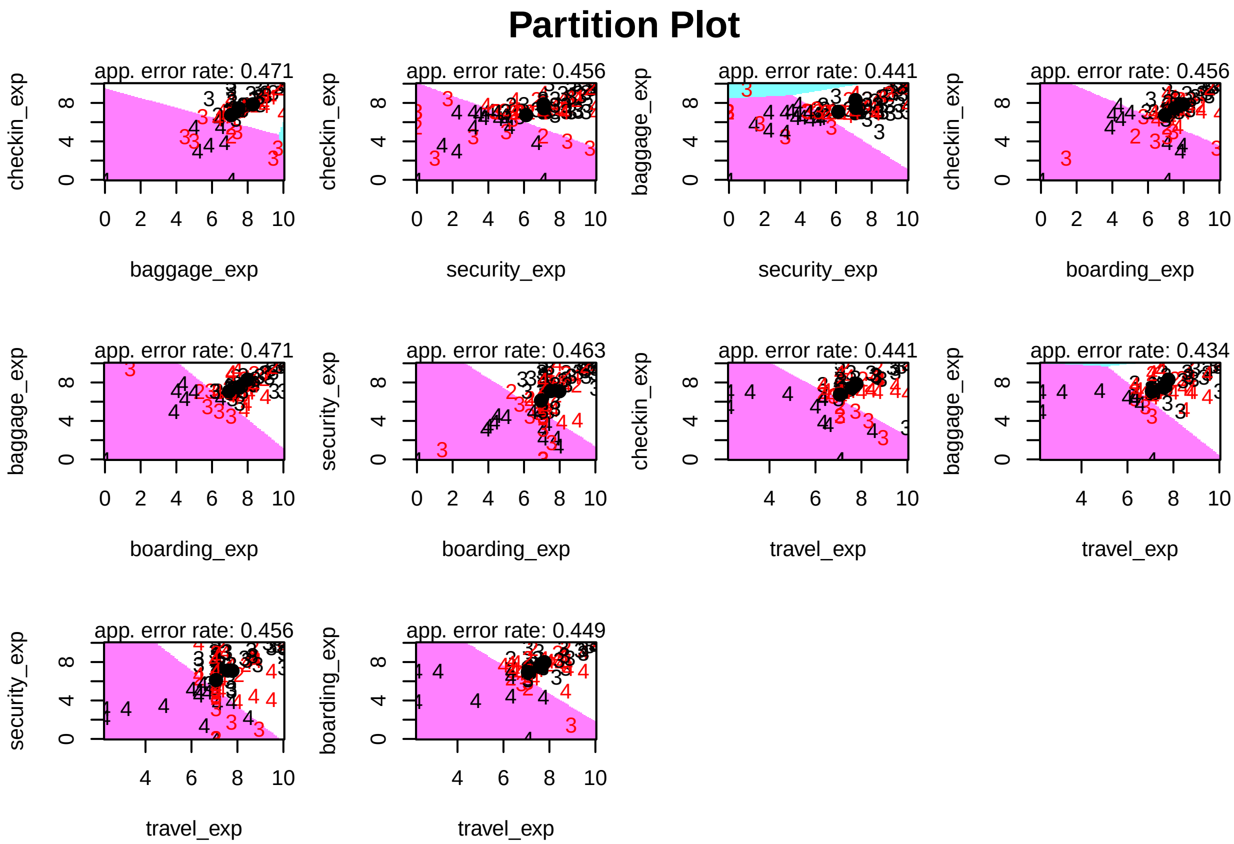
```
lda_values <- predict(lda_model)
plot(lda_values$x[,1], lda_values$x[,2])
text(lda_values$x[,1], lda_values$x[,2], df_new$travel_frequency, cex = 0.7, pos = 4, col = "red")
```



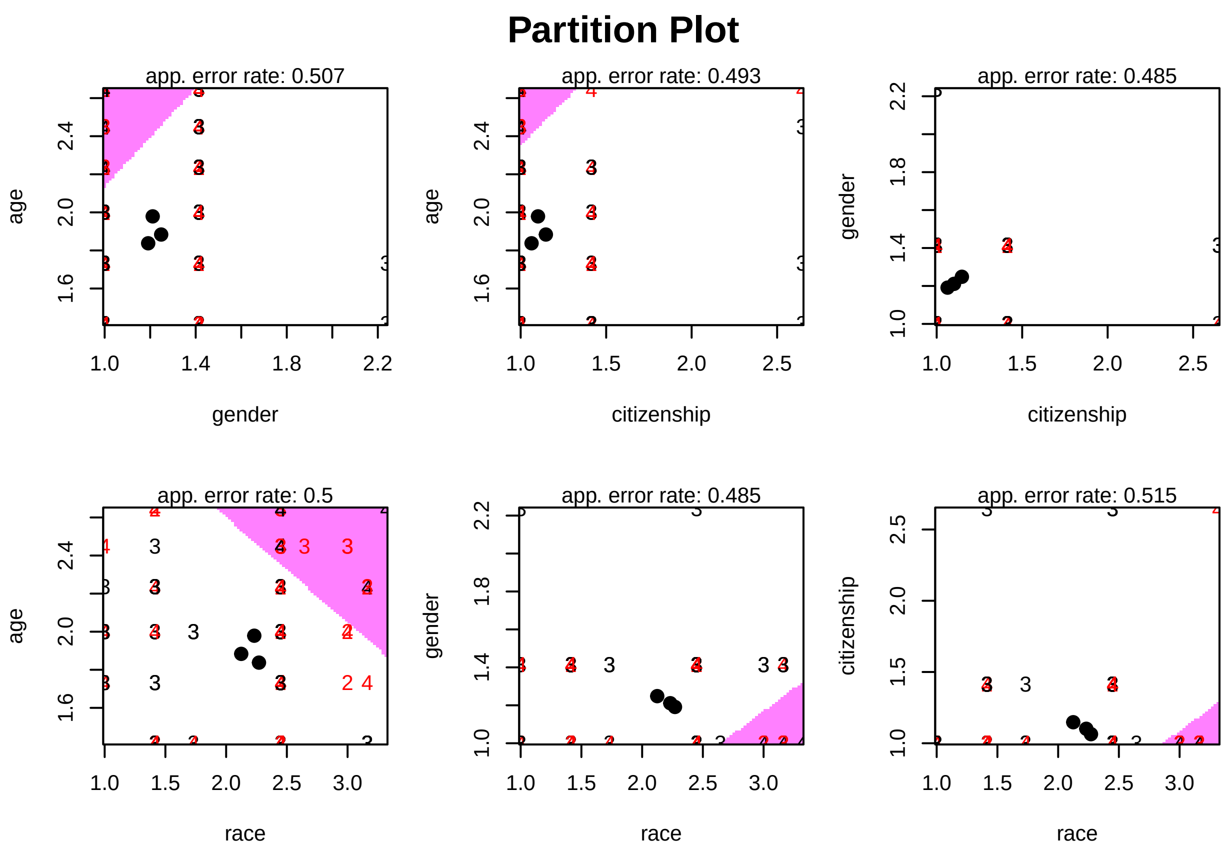

```
newdata <- data.frame(type = df_new[,1], lda = lda_values$x)
ggplot(newdata) + geom_point(aes(lda.LD1, lda.LD2, colour = type), size = 2.5)
```




```
# Partition Plot
partimat(travel_frequency~checkin_exp+baggage_exp+security_exp+boarding_exp+travel_exp,data=df_new,metho
```




```
partimat(travel_frequency~age+gender+citizenship+race,data=df_new,method="lda")
```



Prediction Accuracy

```
#df_new$travel_frequency <- as.factor(df_new$travel_frequency)
lda_predict <- train(travel_frequency ~ ., method = "lda", data = df_new)
confusionMatrix(df_new$travel_frequency, predict(lda_predict, df_new))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  2  3  4
##          2  0  8  5
##          3  0 53 17
##          4  1 31 21
##
## Overall Statistics
##
##                Accuracy : 0.5441
##                  95% CI : (0.4566, 0.6297)
##     No Information Rate : 0.6765
##     P-Value [Acc > NIR] : 0.999526
##
##                   Kappa : 0.1364
##
##  Mcnemar's Test P-Value : 0.002043
##
## Statistics by Class:
##
```

```
##                      Class: 2 Class: 3 Class: 4
## Sensitivity          0.000000   0.5761   0.4884
## Specificity          0.903704   0.6136   0.6559
## Pos Pred Value       0.000000   0.7571   0.3962
## Neg Pred Value       0.991870   0.4091   0.7349
## Prevalence           0.007353   0.6765   0.3162
## Detection Rate       0.000000   0.3897   0.1544
## Detection Prevalence 0.095588   0.5147   0.3897
## Balanced Accuracy    0.451852   0.5949   0.5721
```

We can only achieve 54.41% accuracy from our linear discriminant analysis model.

Quadratic Discriminant Analysis

```r
qda_model <- qda(travel_frequency ~., data = df_new)
qda_model
```

```
## Call:
## qda(travel_frequency ~ ., data = df_new)
##
## Prior probabilities of groups:
##          2          3          4
## 0.09558824 0.51470588 0.38970588
##
## Group means:
##   checkin_exp baggage_exp security_exp boarding_exp travel_exp      age
## 2    7.852995    8.294713     7.092386     7.984569   7.782902 1.837625
## 3    7.437929    7.487044     7.111951     7.499173   7.586359 1.883587
## 4    6.734485    7.073375     6.123599     6.954934   7.067300 1.978919
##     gender citizenship     race
## 2 1.191175    1.063725 2.270467
## 3 1.248340    1.147458 2.124357
## 4 1.211014    1.101390 2.231751
```

Accuracy for QDA

```r
qda_predict <- train(travel_frequency ~ ., method = "qda", data = df_new)
confusionMatrix(df_new$travel_frequency, predict(qda_predict, df_new))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  2  3  4
##          2 12  0  1
##          3  5 57  8
##          4  2 31 20
##
## Overall Statistics
##
##                Accuracy : 0.6544
##                  95% CI : (0.5681, 0.7338)
##     No Information Rate : 0.6471
##     P-Value [Acc > NIR] : 0.4677614
##
```

```
##                   Kappa : 0.3942
##
##  Mcnemar's Test P-Value : 0.0002871
##
## Statistics by Class:
##
##                      Class: 2 Class: 3 Class: 4
## Sensitivity           0.63158   0.6477   0.6897
## Specificity           0.99145   0.7292   0.6916
## Pos Pred Value        0.92308   0.8143   0.3774
## Neg Pred Value        0.94309   0.5303   0.8916
## Prevalence            0.13971   0.6471   0.2132
## Detection Rate        0.08824   0.4191   0.1471
## Detection Prevalence  0.09559   0.5147   0.3897
## Balanced Accuracy     0.81152   0.6884   0.6906
```

It looks like our QDA model has better accuracy, which is 65.44% comparing to LDA model.

STEP WISE LDA

```
# Wilk stepwise
greedy.wilks(travel_frequency~.,data=df_new)
```

```
## Formula containing included variables:
##
## travel_frequency ~ baggage_exp + security_exp
## <environment: 0x14225ea48>
##
##
## Values calculated in each step of the selection procedure:
##
##           vars Wilks.lambda F.statistics.overall p.value.overall
## 1  baggage_exp    0.9168516             6.030823     0.003110902
## 2 security_exp    0.8900001             3.959857     0.003869904
##   F.statistics.diff p.value.diff
## 1          6.030823  0.003110902
## 2          1.991236  0.140576186
```

Only two independents variables that have significant affect on travel_frequency.

WILK TEST

```
dependent <- df_new$travel_frequency
independent <- as.matrix(df_new[,-1])
manova1<-manova(independent ~ dependent)
wilks.test<-summary(manova1,test="Wilks")
wilks.test
```

```
##            Df   Wilks approx F num Df den Df Pr(>F)
## dependent   2 0.82708    1.383     18    250 0.1399
```

```
## Residuals 133
```

Wilk lambda explained how well the independent variable contributes to the model. The scale ranges from 0 to 1, where 0 means total discrimination, and 1 means no discrimination. Since our Wilk is close to 1, we can't say the variables used in this model can't explained the discriminant very well.

CANONICAL DISCRIMINANT ANALYSIS

```r
# Canonical Discriminant Analysis
cda <- candisc(manova1)
print(cda)
```

```
##
## Canonical Discriminant Analysis for dependent:
##
##     CanRsq Eigenvalue Difference Percent Cumulative
## 1 0.115827   0.131001   0.061977  65.492     65.492
## 2 0.064568   0.069024   0.061977  34.508    100.000
##
## Test of H0: The canonical correlations in the
## current row and all that follow are zero
##
##   LR test stat approx F numDF denDF Pr(> F)
## 1      0.82708   1.3830    18   250  0.1399
## 2      0.93543   1.0871     8   126  0.3764
```

```r
cda$coeffs.std
```

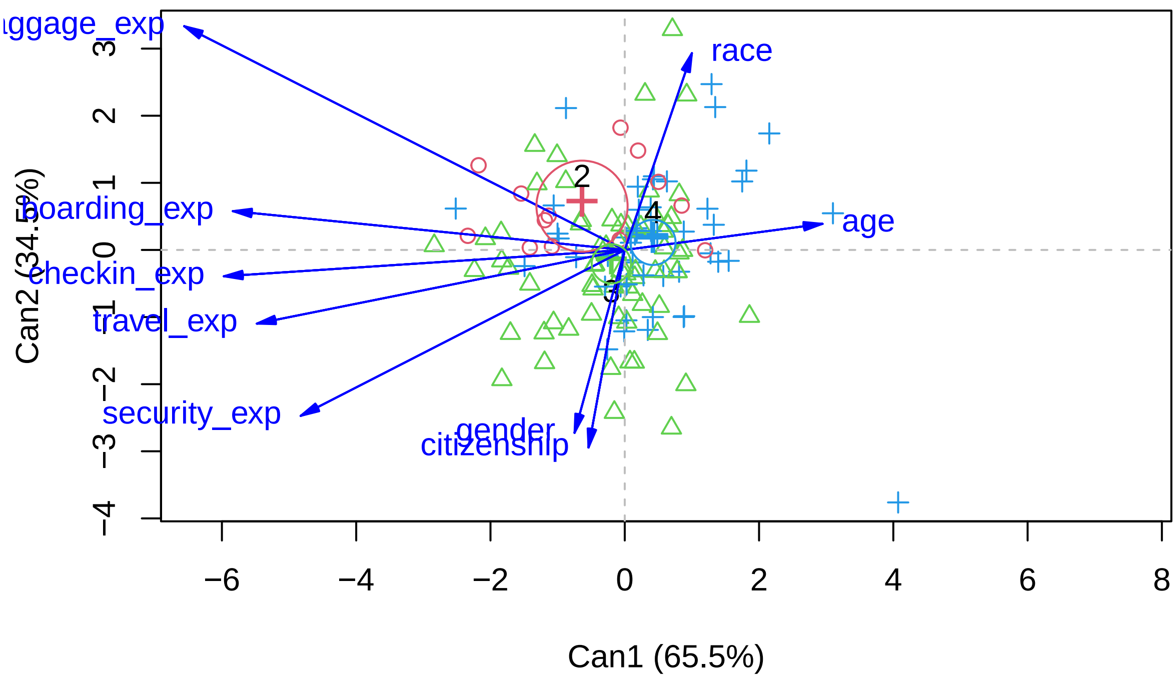
```
##                     Can1         Can2
## checkin_exp  -0.31468032 -0.323782641
## baggage_exp  -0.47082783  1.054747588
## security_exp  0.02822710 -0.644840209
## boarding_exp -0.14943060  0.153106218
## travel_exp   -0.31974200 -0.399068503
## age           0.28996863 -0.005113687
## gender       -0.13494808 -0.345998694
## citizenship  -0.33988187 -0.175693552
## race         -0.05966683  0.381484625
```

```r
cda$structure
```

```
##                     Can1        Can2
## checkin_exp  -0.72078632 -0.04732982
## baggage_exp  -0.79217106  0.40254418
## security_exp -0.58278689 -0.29846977
## boarding_exp -0.70456517  0.06981342
## travel_exp   -0.66153212 -0.13253366
## age           0.35561290  0.04700353
## gender       -0.09026694 -0.32887897
## citizenship  -0.06516545 -0.35550263
## race          0.12070332  0.35394728
```

```r
plot(cda)
```

```
## Vector scale factor set to 8.283
```




```r
# Using only significance variable
dependent <- df_new$travel_frequency
independent2 <- as.matrix(df_new[,3:4])
manova2<-manova(independent2 ~ dependent)
wilks.test2<-summary(manova2,test="Wilks")
wilks.test2
```

```
##             Df Wilks approx F num Df den Df  Pr(>F)
## dependent    2  0.89   3.9599      4    264 0.00387 **
## Residuals 133
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

CANONICAL DISCRIMINANT ANALYSIS FOR SIGNIFICANT VARIABLE

```r
cda2 <- candisc(manova2)
print(cda2)
```

```
##
## Canonical Discriminant Analysis for dependent:
##
##     CanRsq Eigenvalue Difference Percent Cumulative
## 1 0.083186   0.090733   0.060605  75.072     75.072
## 2 0.029247   0.030128   0.060605  24.928    100.000
##
## Test of H0: The canonical correlations in the
## current row and all that follow are zero
```

```
##
##   LR test stat approx F numDF denDF Pr(> F)
## 1      0.89000   3.9599     4   264 0.00387 **
## 2      0.97075   4.0071     1   133 0.04734 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
cda2$coeffs.std
```

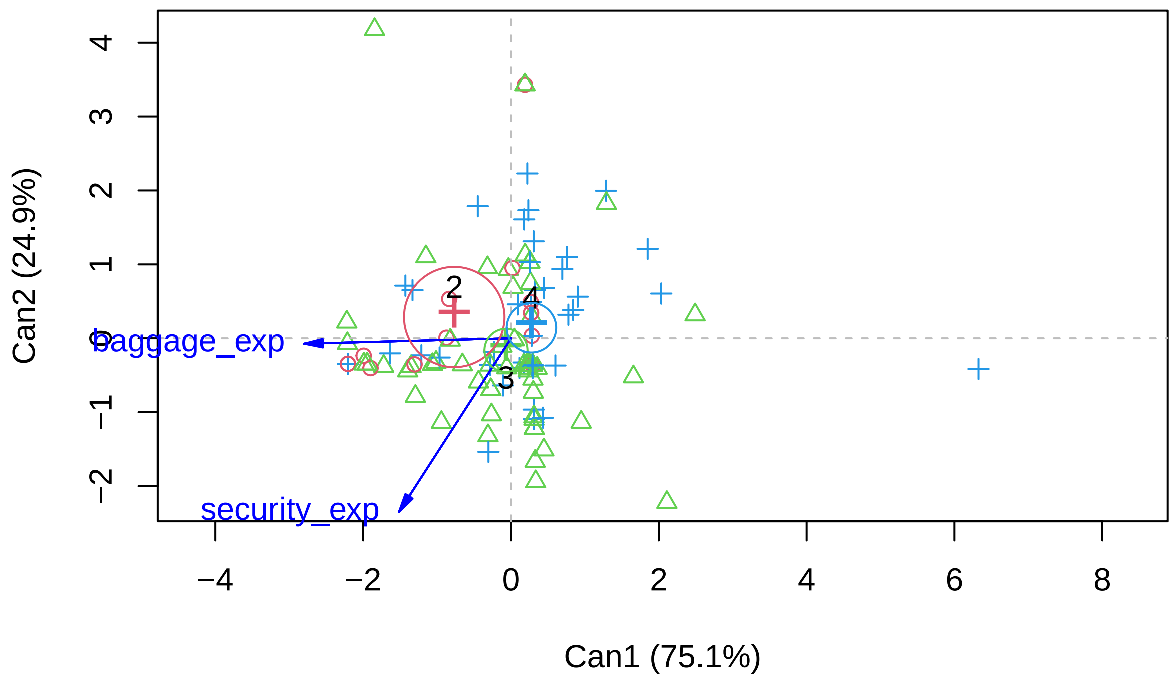
```
##                     Can1       Can2
## baggage_exp  -1.01762267  0.6377981
## security_exp  0.03248253 -1.2005362
```

```
cda2$structure
```

```
##                    Can1        Can2
## baggage_exp  -0.9996545 -0.02628518
## security_exp -0.5419874 -0.84038662
```

```
plot(cda2)
```

```
## Vector scale factor set to 2.799
```



KNN

```
set.seed(42)
sample <- sample(c(TRUE, FALSE), nrow(df_new), replace=TRUE, prob=c(0.75,0.25))
train  <- df_new[sample, ]
test   <- df_new[!sample, ]
```

```r
knn <- knn(train = train, test = test, cl= train$travel_frequency, k=3)
cm <- table(test$travel_frequency, knn)
cm
```

```
##    knn
##      2  3  4
##   2  1  2  0
##   3  0 16  4
##   4  0  3 10
```

```r
# Calculate out of Sample error
misClassError <- mean(knn != test$travel_frequency)
print(paste('Accuracy =', 1-misClassError))
```

```
## [1] "Accuracy = 0.75"
```