FACTOR ANALYSIS

Humaun Farid Sohag

2025-08-13

FACTOR ANALYSIS

#Load Required Libraries

```
# Install the 'psych' package if not already installed
if(!require(psych)) install.packages("psych", dependencies = TRUE)
```

```
## Loading required package: psych
```

```
library(psych)
```

#Factor Analysis on the Iris Dataset

```
# Load iris dataset
data(iris)
# View the first few rows
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
# Scale numeric variables (first 4 columns)
iris_scaled <- scale(iris[, 1:4])
# Determine the number of factors using factor analysis
# Using 4 factors and varimax rotation
fa_iris <- fa(r = iris_scaled,
              nfactors = 4,
              rotate = "varimax")
```

Summarize results

```r
# Summarize results
summary(fa_iris)
```

```
##
## Factor analysis with Call: fa(r = iris_scaled, nfactors = 4, rotate = "varimax")
##
## Test of the hypothesis that 4 factors are sufficient.
## The degrees of freedom for the model is -4  and the objective function was  0
## The number of observations was  150  with Chi Square =  0  with prob <  NA
##
## The root mean square of the residuals (RMSA) is  0
## The df corrected root mean square of the residuals is  NA
##
## Tucker Lewis Index of factoring reliability =  1.009
```

```r
# View factor loadings
fa_iris$loadings
```

```
##
## Loadings:
##              MR1    MR2    MR3    MR4
## Sepal.Length  0.997
## Sepal.Width  -0.108  0.757
## Petal.Length  0.861 -0.413  0.288
## Petal.Width   0.801 -0.317  0.492
##
##                  MR1   MR2   MR3   MR4
## SS loadings    2.389 0.844 0.332 0.000
## Proportion Var 0.597 0.211 0.083 0.000
## Cumulative Var 0.597 0.808 0.891 0.891
```

```r
# Optional: Factor analysis on a subset
subset1 <- subset(iris[, 1:4], iris$Sepal.Length < mean(iris$Sepal.Length))
fa_subset <- fa(subset1, nfactors = 4, rotate = "varimax")
print(fa_subset)
```

```
## Factor Analysis using method =  minres
## Call: fa(r = subset1, nfactors = 4, rotate = "varimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                MR3   MR2   MR1 MR4   h2    u2 com
## Sepal.Length  0.05  0.90  0.09   0 0.82 0.178 1.0
## Sepal.Width  -0.92 -0.05 -0.09   0 0.85 0.150 1.0
## Petal.Length  0.68  0.66  0.30   0 1.00 0.005 2.4
## Petal.Width   0.62  0.62  0.45   0 0.97 0.031 2.8
##
##                        MR3  MR2  MR1  MR4
## SS loadings           1.69 1.64 0.31 0.00
## Proportion Var        0.42 0.41 0.08 0.00
## Cumulative Var        0.42 0.83 0.91 0.91
## Proportion Explained  0.46 0.45 0.08 0.00
## Cumulative Proportion 0.46 0.92 1.00 1.00
```

```
##
## Mean item complexity =  1.8
## Test of the hypothesis that 4 factors are sufficient.
##
## df null model =  6  with the objective function =  4.57 with Chi Square =  351.02
## df of  the model are -4  and the objective function was  0
##
## The root mean square of the residuals (RMSR) is  0
## The df corrected root mean square of the residuals is  NA
##
## The harmonic n.obs is  80 with the empirical chi square  0  with prob <  NA
## The total n.obs was  80  with Likelihood Chi Square =  0  with prob <  NA
##
## Tucker Lewis Index of factoring reliability =  1.018
## Fit based upon off diagonal values = 1
## Measures of factor score adequacy
##                                                    MR3  MR2  MR1 MR4
## Correlation of (regression) scores with factors   0.94 0.93 0.73   0
## Multiple R square of scores with factors          0.89 0.87 0.53   0
## Minimum correlation of possible factor scores     0.78 0.75 0.06  -1
```

Factor Analysis on the mtcars Dataset

```r
# Load mtcars dataset
data(mtcars)
# Perform factor analysis using 'factanal' with 3 factors and varimax rotation
factor_analysis_mtcars <- factanal(x = mtcars,
                                   factors = 3,
                                   rotation = "varimax")
# Print results
print(factor_analysis_mtcars)
```

```
##
## Call:
## factanal(x = mtcars, factors = 3, rotation = "varimax")
##
## Uniquenesses:
##   mpg   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb
## 0.135 0.055 0.090 0.127 0.290 0.060 0.051 0.223 0.208 0.125 0.158
##
## Loadings:
##      Factor1 Factor2 Factor3
## mpg   0.643  -0.478  -0.473
## cyl  -0.618   0.703   0.261
## disp -0.719   0.537   0.323
## hp   -0.291   0.725   0.513
## drat  0.804  -0.241
## wt   -0.778   0.248   0.524
## qsec -0.177  -0.946  -0.151
## vs    0.295  -0.805  -0.204
## am    0.880
## gear  0.908           0.224
```

```
## carb  0.114   0.559   0.719
##
##                Factor1 Factor2 Factor3
## SS loadings      4.380   3.520   1.578
## Proportion Var   0.398   0.320   0.143
## Cumulative Var   0.398   0.718   0.862
##
## Test of the hypothesis that 3 factors are sufficient.
## The chi square statistic is 30.53 on 25 degrees of freedom.
## The p-value is 0.205
```

- Notes: factanal() arguments:
- x : numeric data matrix or data frame
- factors : number of factors to extract
- rotation : rotation method (e.g., "varimax", "promax")
- scores : type of factor scores ("none", "regression", "Bartlett")
- covmat : covariance matrix if x is not provided

MULTIPLE FACTOR ANALYSIS (MFA) IN R

Step 1: Install and load required packages

```r
if(!require(FactoMineR)) install.packages("FactoMineR", dependencies = TRUE)
library(FactoMineR)

if(!require(factoextra)) install.packages("factoextra", dependencies = TRUE)
library(factoextra)
```

Step 2: Load the dataset

```r
data("iris")
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Step 3: Define groups for MFA

```r
iris_data <- iris[, 1:4]          # Only numeric columns
group_definitions <- c(2, 2)      # Sepal: 2 variables, Petal: 2 variables
```

Step 4: Perform Multiple Factor Analysis

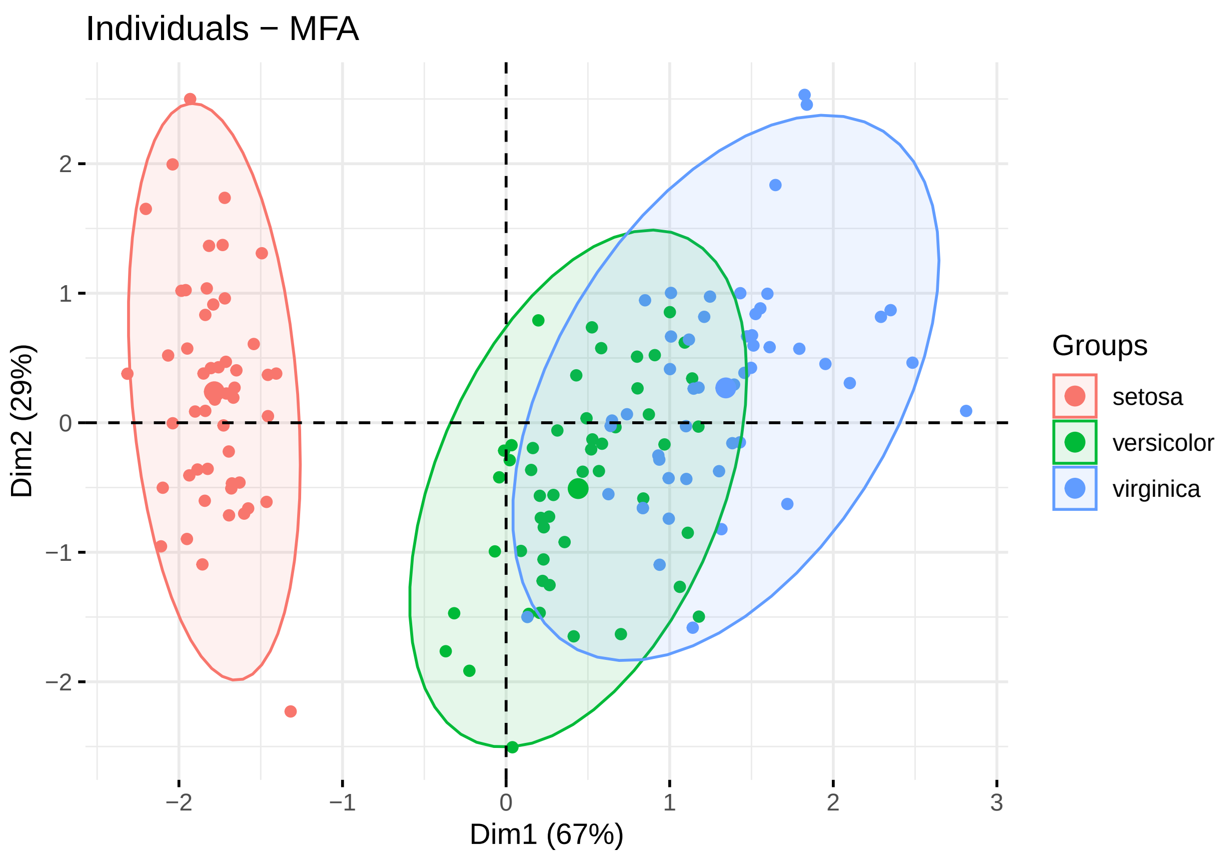
```r
res_mfa <- MFA(
  iris_data,
  group = group_definitions,
  type = c("s", "s"),              # Quantitative variables
  name.group = c("Sepal", "Petal"),
  graph = FALSE
)

# View MFA summary
summary(res_mfa)
```

```
##
## Call:
## MFA(base = iris_data, group = group_definitions, type = c("s",
##      "s"), name.group = c("Sepal", "Petal"), graph = FALSE)
##
##
## Eigenvalues
##                        Dim.1   Dim.2   Dim.3   Dim.4
## Variance               1.882   0.815   0.101   0.011
## % of var.             67.006  29.026   3.579   0.389
## Cumulative % of var.  67.006  96.032  99.611 100.000
##
## Groups
##                  Dim.1    ctr   cos2    Dim.2    ctr   cos2    Dim.3    ctr
## Sepal        |  0.932 49.551  0.536 |  0.810 99.336  0.404 |  0.047 46.595
## Petal        |  0.949 50.449  0.901 |  0.005  0.664  0.000 |  0.054 53.405
##                cos2
## Sepal         0.001 |
## Petal         0.003 |
##
## Individuals (the 10 first)
##                  Dim.1    ctr   cos2    Dim.2    ctr   cos2    Dim.3    ctr
## 1            | -1.804  1.153  0.943 |  0.422  0.146  0.052 | -0.132  0.116
## 2            | -1.577  0.881  0.834 | -0.661  0.358  0.147 | -0.230  0.350
## 3            | -1.886  1.261  0.965 | -0.361  0.107  0.035 |  0.005  0.000
## 4            | -1.842  1.202  0.902 | -0.603  0.297  0.097 |  0.052  0.018
## 5            | -1.949  1.346  0.920 |  0.573  0.268  0.080 | -0.009  0.001
## 6            | -1.733  1.064  0.615 |  1.372  1.540  0.385 |  0.003  0.000
## 7            | -2.038  1.471  0.985 | -0.004  0.000  0.000 |  0.252  0.420
## 8            | -1.781  1.123  0.987 |  0.179  0.026  0.010 | -0.096  0.061
## 9            | -1.856  1.221  0.741 | -1.094  0.978  0.257 |  0.090  0.054
## 10           | -1.676  0.995  0.911 | -0.468  0.179  0.071 | -0.233  0.361
##                cos2
## 1             0.005 |
## 2             0.018 |
```

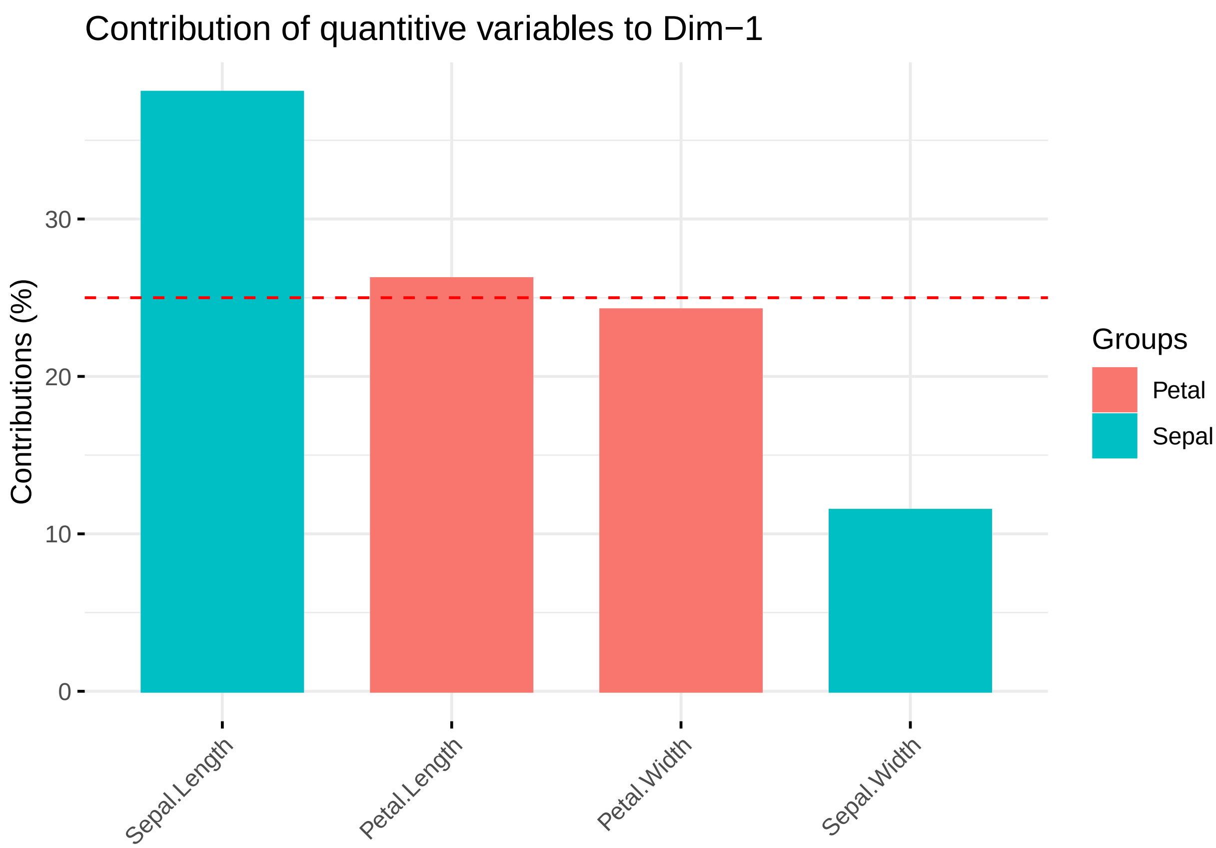
```
## 3             0.000 |
## 4             0.001 |
## 5             0.000 |
## 6             0.000 |
## 7             0.015 |
## 8             0.003 |
## 9             0.002 |
## 10            0.018 |
##
## Continuous variables
##                 Dim.1    ctr   cos2    Dim.2    ctr   cos2    Dim.3    ctr
## Sepal.Length |  0.895 38.053  0.800 |  0.391 16.752  0.153 | -0.216 41.516
## Sepal.Width  | -0.492 11.497  0.242 |  0.867 82.584  0.752 |  0.076  5.078
## Petal.Length |  0.984 26.213  0.968 |  0.052  0.168  0.003 |  0.122  7.520
## Petal.Width  |  0.946 24.236  0.895 |  0.089  0.495  0.008 |  0.301 45.885
##                cos2
## Sepal.Length  0.047 |
## Sepal.Width   0.006 |
## Petal.Length  0.015 |
## Petal.Width   0.091 |
```

Step 5: Visualize MFA results

```r
# Individuals colored by species with ellipses
fviz_mfa_ind(
  res_mfa,
  label = "var",
  habillage = iris$Species,
  addEllipses = TRUE,
  ellipse.level = 0.95
)
```




```r
# Contributions of quantitative variables to Dimension 1
fviz_contrib(res_mfa, choice = "quanti.var", axes = 1)
```



CONFIRMATORY FACTOR ANALYSIS (CFA)

```
# Step 1: Install and load required package
if(!require(lavaan)) install.packages("lavaan", dependencies = TRUE)
library(lavaan)

# Step 2: Load and check the structure of the dataset
data("HolzingerSwineford1939")  # Load dataset from lavaan package
head(HolzingerSwineford1939)    # View first few rows
```

```
##   id sex ageyr agemo  school grade       x1   x2    x3       x4   x5        x6
## 1  1   1    13     1 Pasteur     7 3.333333 7.75 0.375 2.333333 5.75 1.2857143
## 2  2   2    13     7 Pasteur     7 5.333333 5.25 2.125 1.666667 3.00 1.2857143
## 3  3   2    13     1 Pasteur     7 4.500000 5.25 1.875 1.000000 1.75 0.4285714
## 4  4   1    13     2 Pasteur     7 5.333333 7.75 3.000 2.666667 4.50 2.4285714
## 5  5   2    12     2 Pasteur     7 4.833333 4.75 0.875 2.666667 4.00 2.5714286
## 6  6   2    14     1 Pasteur     7 5.333333 5.00 2.250 1.000000 3.00 0.8571429
##         x7   x8       x9
## 1 3.391304 5.75 6.361111
## 2 3.782609 6.25 7.916667
## 3 3.260870 3.90 4.416667
## 4 3.000000 5.30 4.861111
## 5 3.695652 6.30 5.916667
```

```
## 6 4.347826 6.65 7.500000
```

```r
str(HolzingerSwineford1939)     # Check structure
```

```
## 'data.frame':    301 obs. of  15 variables:
##  $ id    : int  1 2 3 4 5 6 7 8 9 11 ...
##  $ sex   : int  1 2 2 1 2 2 1 2 2 2 ...
##  $ ageyr : int  13 13 13 13 12 14 12 12 13 12 ...
##  $ agemo : int  1 7 1 2 2 1 1 2 0 5 ...
##  $ school: Factor w/ 2 levels "Grant-White",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ grade : int  7 7 7 7 7 7 7 7 7 7 ...
##  $ x1    : num  3.33 5.33 4.5 5.33 4.83 ...
##  $ x2    : num  7.75 5.25 5.25 7.75 4.75 5 6 6.25 5.75 5.25 ...
##  $ x3    : num  0.375 2.125 1.875 3 0.875 ...
##  $ x4    : num  2.33 1.67 1 2.67 2.67 ...
##  $ x5    : num  5.75 3 1.75 4.5 4 3 6 4.25 5.75 5 ...
##  $ x6    : num  1.286 1.286 0.429 2.429 2.571 ...
##  $ x7    : num  3.39 3.78 3.26 3 3.7 ...
##  $ x8    : num  5.75 6.25 3.9 5.3 6.3 6.65 6.2 5.15 4.65 4.55 ...
##  $ x9    : num  6.36 7.92 4.42 4.86 5.92 ...
```

```r
# Step 3: Specify the CFA model
# Three latent variables:
#   visual   -> x1, x2, x3
#   textual  -> x4, x5, x6
#   speed    -> x7, x8, x9
model <- '
  visual  =~ x1 + x2 + x3
  textual =~ x4 + x5 + x6
  speed   =~ x7 + x8 + x9
'

# Step 4: Run CFA and check results
cfa_result <- cfa(model, data = HolzingerSwineford1939)

# Display summary with fit measures and standardized estimates
summary(cfa_result, fit.measures = TRUE, standardized = TRUE)
```

```
## lavaan 0.6-19 ended normally after 35 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        21
##
##   Number of observations                           301
##
## Model Test User Model:
##
##   Test statistic                                85.306
##   Degrees of freedom                                24
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
```

```
##
##   Test statistic                               918.852
##   Degrees of freedom                                36
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.931
##   Tucker-Lewis Index (TLI)                       0.896
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -3737.745
##   Loglikelihood unrestricted model (H1)      -3695.092
##
##   Akaike (AIC)                                7517.490
##   Bayesian (BIC)                              7595.339
##   Sample-size adjusted Bayesian (SABIC)       7528.739
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.092
##   90 Percent confidence interval - lower         0.071
##   90 Percent confidence interval - upper         0.114
##   P-value H_0: RMSEA <= 0.050                    0.001
##   P-value H_0: RMSEA >= 0.080                    0.840
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.065
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   visual =~
##     x1                1.000                               0.900    0.772
##     x2                0.554    0.100    5.554    0.000    0.498    0.424
##     x3                0.729    0.109    6.685    0.000    0.656    0.581
##   textual =~
##     x4                1.000                               0.990    0.852
##     x5                1.113    0.065   17.014    0.000    1.102    0.855
##     x6                0.926    0.055   16.703    0.000    0.917    0.838
##   speed =~
##     x7                1.000                               0.619    0.570
##     x8                1.180    0.165    7.152    0.000    0.731    0.723
##     x9                1.082    0.151    7.155    0.000    0.670    0.665
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
```

```
##   visual ~~
##     textual           0.408    0.074    5.552    0.000    0.459    0.459
##     speed             0.262    0.056    4.660    0.000    0.471    0.471
##   textual ~~
##     speed             0.173    0.049    3.518    0.000    0.283    0.283
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .x1                0.549    0.114    4.833    0.000    0.549    0.404
##    .x2                1.134    0.102   11.146    0.000    1.134    0.821
##    .x3                0.844    0.091    9.317    0.000    0.844    0.662
##    .x4                0.371    0.048    7.779    0.000    0.371    0.275
##    .x5                0.446    0.058    7.642    0.000    0.446    0.269
##    .x6                0.356    0.043    8.277    0.000    0.356    0.298
##    .x7                0.799    0.081    9.823    0.000    0.799    0.676
##    .x8                0.488    0.074    6.573    0.000    0.488    0.477
##    .x9                0.566    0.071    8.003    0.000    0.566    0.558
##     visual            0.809    0.145    5.564    0.000    1.000    1.000
##     textual           0.979    0.112    8.737    0.000    1.000    1.000
##     speed             0.384    0.086    4.451    0.000    1.000    1.000
```

Exploratory Factor Analysis (EFA)

```r
# Step 1: Install and Load Required Packages
if(!require(psych)) install.packages("psych", dependencies = TRUE)
library(psych)

if(!require(factoextra)) install.packages("factoextra", dependencies = TRUE)
library(factoextra)

if(!require(lavaan)) install.packages("lavaan", dependencies = TRUE)
library(lavaan)
```

Step 2: Load and Inspect the Dataset

```r
data(mtcars)
head(mtcars)    # View first few rows
```

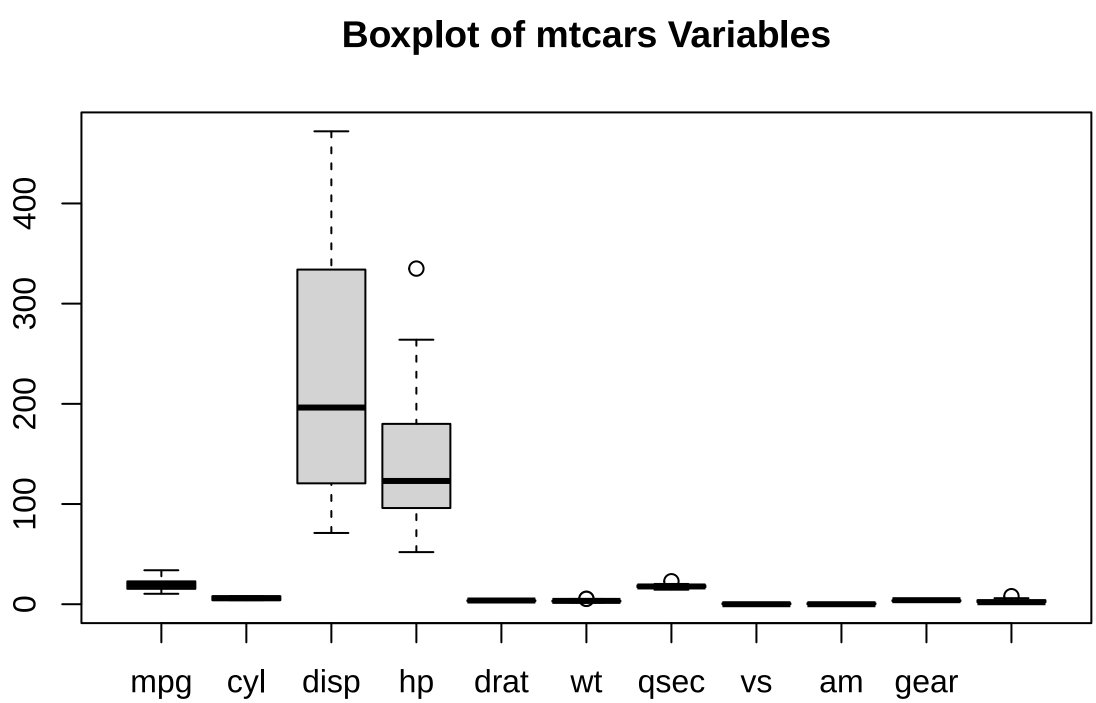
```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

Step 3: Perform Exploratory Data Analysis (EDA)

```
# Check for missing values
sum(is.na(mtcars))
```

```
## [1] 0
```

```
# Check for outliers
boxplot(mtcars, main = "Boxplot of mtcars Variables")
```



Step 4: Conduct EFA

```
# Perform EFA with 2 factors and Varimax rotation
efa_result <- fa(r = mtcars, nfactors = 2, rotate = "varimax")

# Step 5: View EFA Results
print(efa_result)
```

```
## Factor Analysis using method =  minres
## Call: fa(r = mtcars, nfactors = 2, rotate = "varimax")
```

```
## Standardized loadings (pattern matrix) based upon correlation matrix
##        MR1   MR2   h2    u2 com
## mpg   0.68 -0.63 0.85 0.147 2.0
## cyl  -0.63  0.73 0.94 0.064 2.0
## disp -0.73  0.61 0.90 0.102 1.9
## hp   -0.32  0.88 0.88 0.124 1.3
## drat  0.81 -0.22 0.71 0.292 1.1
## wt   -0.78  0.45 0.82 0.179 1.6
## qsec -0.15 -0.87 0.78 0.216 1.1
## vs    0.30 -0.79 0.71 0.292 1.3
## am    0.90  0.07 0.82 0.183 1.0
## gear  0.88  0.15 0.80 0.200 1.1
## carb  0.05  0.81 0.66 0.342 1.0
##
##                        MR1  MR2
## SS loadings           4.46 4.39
## Proportion Var        0.41 0.40
## Cumulative Var        0.41 0.81
## Proportion Explained  0.50 0.50
## Cumulative Proportion 0.50 1.00
##
## Mean item complexity =  1.4
## Test of the hypothesis that 2 factors are sufficient.
##
## df null model =  55  with the objective function =  15.4 with Chi Square =  408.01
## df of  the model are 34  and the objective function was  2.76
##
## The root mean square of the residuals (RMSR) is  0.04
## The df corrected root mean square of the residuals is  0.06
##
## The harmonic n.obs is  32 with the empirical chi square  6.87  with prob <  1
## The total n.obs was  32  with Likelihood Chi Square =  69.56  with prob <  0.00031
##
## Tucker Lewis Index of factoring reliability =  0.827
## RMSEA index =  0.178  and the 90 % confidence intervals are  0.121 0.245
## BIC =  -48.28
## Fit based upon off diagonal values = 0.99
## Measures of factor score adequacy
##                                                    MR1  MR2
## Correlation of (regression) scores with factors   0.98 0.98
## Multiple R square of scores with factors          0.95 0.96
## Minimum correlation of possible factor scores     0.91 0.92
```