```
###IMPORT DATASET
# EFA.data.csv
# EFA.data <- read.csv("~/OneDrive - The University of Texas at Tyler/Teaching GD/PSYC 6341/PPT/9 EFA/EFA data.csv")

View(EFA.data)

summary(EFA.data)
dim(EFA.data)

## Missing Data
any(is.na(EFA.data))

nomiss=na.omit(EFA.data)
dim(nomiss)

## Outliers


df = ncol(nomiss)
cutoff = qchisq(1-.001, ncol(nomiss)) ## Cutoff score
mahal = mahalanobis(nomiss,
                    colMeans(nomiss),
                    cov(nomiss) # Calculate Mahalanobis distance
)
cutoff ## Print cutoff score
ncol(nomiss) ##df
summary(mahal < cutoff) # Remember FALSE is bad.



##exclude outliers
noout = subset(nomiss, mahal < cutoff)

## Check
dim(noout)



# Assumptions

## Correlation Matrix
```

~~~~
# Use correlation matrix rather than raw data file

corfact = cor(noout)
round(corfact,3)    # round correlations to three decimal places

#Check that the determinant for the correlation matrix is positive.
det(corfact)

# Check the determinant for the covariance matrix is postive
det(cov(noout))

# Prints out the p-values for correlations in matrix
library(Hmisc)
rcorr(as.matrix(noout))



## Normality

# Run a fake regression
random = rchisq(nrow(noout), df)
fake = lm(random~., data = data.frame(noout))
standardized = rstudent(fake)
fitted = scale(fake$fitted.values)

#install.packages("lindia")
library(lindia)
# Histogram
gg_reshist(fake, bins = 10)
```

## Linearity


qqnorm(standardized)



## Homogeneity of Variance-Covariance Matrices
## Homogeneity of Variance-Covariance Matrices PLOT
plot(fitted, standardized)
abline(0,0)
abline(v = 0)
~~~~

```
# install.packages("rela")
library(rela)
paf.corfact = paf(as.matrix(noout), eigcrit=1, convcrit=.001)
# There's a lot of output, so I used $KMO
summary(paf.corfact)$KMO


summary(paf.corfact)$MSA
any(summary(paf.corfact)$MSA <.5) #Check

# Test significance of the Bartlett test
library(psych)
bt = cortest.bartlett(cor(noout), n = 185)
bt

bt.p.value  = ifelse(round(bt$p.value,3) == 0, "<.001", paste0("= ", round(bt$p.value,3)))


# paralel analysis
nofactors = fa.parallel(noout, n.obs=185, fm="ml", fa="fa")
nofactors



## Factor Analysis
fa.noout = fa(r = noout, nfactors = 3, rotate = "promax", fm = "ml")
# Scree Plot
plot( fa.noout$values, type = 'b', xlim=c(1,10),
     main = "Scree Plot", xlab="Number of Factors",
     ylab="Eigenvalues")

## Factor Analysis
fa.noout = fa(r = noout, nfactors = 3, rotate = "promax", fm = "ml")
fa.noout


fa.noout = fa(noout[,-c(24)], nfactors = 3, rotate = "promax", fm = "ml")
# This print option will hide anything less than .300
# ALSO sort
print(fa.noout$loadings,cutoff = 0.3, sort = TRUE)

##reliability
# I excluded item 24
factor1 = c(1, 3, 7, 8, 10:12, 14, 16, 18, 20:23, 25, 29, 31, 32)
```

```
factor2 = c(2, 5, 13, 19, 26, 28, 30)
factor3 = c(4, 6, 9, 15, 17, 27)
psych::alpha(noout[ , factor1])
psych::alpha(noout[ , factor2])
# ITEM 6,8,17,27 are reverse coded.
psych::alpha(noout[ , factor3], check.keys= TRUE)

 ## Factor Descriptives


##create new factor scores
noout$f1 = apply(noout[ , factor1], 1, mean) ##creates average scores
noout$f2 = apply(noout[ , factor2], 1, mean) ##creates average scores
noout$f3 = apply(noout[ , factor3], 1, mean) ##creates average scores

summary(noout)

sd(noout$f1)
sd(noout$f2)
sd(noout$f3)

# Factor Correlations
# noout[,c(33:35)] #last columns are the factors
library(Hmisc)
rcorr(as.matrix(noout[,c(33:35)]))
```