```
> #=========================
> #   UNIVARIATE OUTLIERS
> #=========================
>
> library(readxl)
>
> Data <- read_xlsx("C:/Users/Public/SupplementaryData_2-Dataset.xlsx", sheet = "Matrix", range = "A1:AN244", col_names = TRUE)
>
> zScoreAOC1 <- abs((Data$AOC1 - mean(Data$AOC1))/sd(Data$AOC1))
> which(zScoreAOC1 > 3)
integer(0)
> zScoreAOC2 <- abs((Data$AOC2 - mean(Data$AOC2))/sd(Data$AOC2))
> which(zScoreAOC2 > 3)
integer(0)
> zScoreAOC3 <- abs((Data$AOC3 - mean(Data$AOC3))/sd(Data$AOC3))
> which(zScoreAOC3 > 3)
integer(0)
> zScoreAOC4 <- abs((Data$AOC4 - mean(Data$AOC4))/sd(Data$AOC4))
> which(zScoreAOC4 > 3)
integer(0)
> zScoreAOC5 <- abs((Data$AOC5 - mean(Data$AOC5))/sd(Data$AOC5))
> which(zScoreAOC5 > 3)
integer(0)
> zScoreCOCR1 <- abs((Data$COCR1 - mean(Data$COCR1))/sd(Data$COCR1))
> which(zScoreCOCR1 > 3)
integer(0)
> zScoreCOCI2 <- abs((Data$COCI2 - mean(Data$COCI2))/sd(Data$COCI2))
> which(zScoreCOCI2 > 3)
integer(0)
> zScoreCOCS3 <- abs((Data$COCS3 - mean(Data$COCS3))/sd(Data$COCS3))
> which(zScoreCOCS3 > 3)
integer(0)
> zScoreCOCS4 <- abs((Data$COCS4 - mean(Data$COCS4))/sd(Data$COCS4))
> which(zScoreCOCS4 > 3)
integer(0)
> zScoreCOCS5 <- abs((Data$COCS5 - mean(Data$COCS5))/sd(Data$COCS5))
> which(zScoreCOCS5 > 3)
integer(0)
> zScoreCOCS6 <- abs((Data$COCS6 - mean(Data$COCS6))/sd(Data$COCS6))
> which(zScoreCOCS6 > 3)
integer(0)
> zScoreCOCI7 <- abs((Data$COCI7 - mean(Data$COCI7))/sd(Data$COCI7))
> which(zScoreCOCI7 > 3)
integer(0)
> zScoreCOCI8 <- abs((Data$COCI8 - mean(Data$COCI8))/sd(Data$COCI8))
> which(zScoreCOCI8 > 3)
integer(0)
> zScoreCOCP9 <- abs((Data$COCP9 - mean(Data$COCP9))/sd(Data$COCP9))
> which(zScoreCOCP9 > 3)
integer(0)
```

```
> zScoreCOCP10 <- abs((Data$COCP10 - mean(Data$COCP10))/sd(Data$COCP10))
> which(zScoreCOCP10 > 3)
integer(0)
> zScoreCOCR11 <- abs((Data$COCR11 - mean(Data$COCR11))/sd(Data$COCR11))
> which(zScoreCOCR11 > 3)
integer(0)
> zScoreCOCR12 <- abs((Data$COCR12 - mean(Data$COCR12))/sd(Data$COCR12))
> which(zScoreCOCR12 > 3)
integer(0)
> zScoreCOCP13 <- abs((Data$COCP13 - mean(Data$COCP13))/sd(Data$COCP13))
> which(zScoreCOCP13 > 3)
integer(0)
> zScoreCOCP14 <- abs((Data$COCP14 - mean(Data$COCP14))/sd(Data$COCP14))
> which(zScoreCOCP14 > 3)
integer(0)
> zScoreCOCP15 <- abs((Data$COCP15 - mean(Data$COCP15))/sd(Data$COCP15))
> which(zScoreCOCP15 > 3)
integer(0)
> zScoreTI1 <- abs((Data$TI1 - mean(Data$TI1))/sd(Data$TI1))
> which(zScoreTI1 > 3)
integer(0)
> zScoreTI2 <- abs((Data$TI2 - mean(Data$TI2))/sd(Data$TI2))
> which(zScoreTI2 > 3)
integer(0)
> zScoreTI3 <- abs((Data$TI3 - mean(Data$TI3))/sd(Data$TI3))
> which(zScoreTI3 > 3)
integer(0)
> zScorePOQ1 <- abs((Data$POQ1 - mean(Data$POQ1))/sd(Data$POQ1))
> which(zScorePOQ1 > 3)
integer(0)
> zScorePOQ2 <- abs((Data$POQ2 - mean(Data$POQ2))/sd(Data$POQ2))
> which(zScorePOQ2 > 3)
integer(0)
> zScorePOQ3 <- abs((Data$POQ3 - mean(Data$POQ3))/sd(Data$POQ3))
> which(zScorePOQ3 > 3)
integer(0)
> zScorePOQ4 <- abs((Data$POQ4 - mean(Data$POQ4))/sd(Data$POQ4))
> which(zScorePOQ4 > 3)
integer(0)
> zScorePOQ5 <- abs((Data$POQ5 - mean(Data$POQ5))/sd(Data$POQ5))
> which(zScorePOQ5 > 3)
integer(0)
> zScorePOQ6 <- abs((Data$POQ6 - mean(Data$POQ6))/sd(Data$POQ6))
> which(zScorePOQ6 > 3)
integer(0)
> zScorePOQ7 <- abs((Data$POQ7 - mean(Data$POQ7))/sd(Data$POQ7))
> which(zScorePOQ7 > 3)
integer(0)
> zScorePOQ8 <- abs((Data$POQ8 - mean(Data$POQ8))/sd(Data$POQ8))
> which(zScorePOQ8 > 3)
```

```
integer(0)
> zScorePOQ9 <- abs((Data$POQ9 - mean(Data$POQ9))/sd(Data$POQ9))
> which(zScorePOQ9 > 3)
integer(0)
>
> #============================
> #   MULTIVARIATE OUTLIERS
> #============================
>
> library(readxl)
> library(psych)
>
> Data <- read_xlsx("C:/Users/Public/SupplementaryData_2-Dataset.xlsx", sheet = "Matrix", range = "A1:AN244", col_names = TRUE)
>
> outliersMulti_AOC <- outlier(data.frame(Data$AOC1, Data$AOC2, Data$AOC3, Data$AOC4, Data$AOC5), plot = FALSE)
> outliersMulti_COCS <- outlier(data.frame(Data$COCS3, Data$COCS4, Data$COCS5, Data$COCS6), plot = FALSE)
> outliersMulti_COCI <- outlier(data.frame(Data$COCI2, Data$COCI7, Data$COCI8), plot = FALSE)
> outliersMulti_COCR <- outlier(data.frame(Data$COCR1, Data$COCR11, Data$COCR12), plot = FALSE)
> outliersMulti_COCP <- outlier(data.frame(Data$COCP9, Data$COCP10, Data$COCP13, Data$COCP14, Data$COCP15), plot = FALSE)
> outliersMulti_TI <- outlier(data.frame(Data$TI1, Data$TI2, Data$TI3), plot = FALSE)
> outliersMulti_POQN <- outlier(data.frame(Data$POQ2, Data$POQ4, Data$POQ5, Data$POQ8), plot = FALSE)
> outliersMulti_POQR <- outlier(data.frame(Data$POQ1, Data$POQ3, Data$POQ6, Data$POQ7, Data$POQ9), plot = FALSE)
> #p = 0,001
> outliersMulti <- c(
+ outliersMulti_AOC[which(outliersMulti_AOC > 20.515)],
+ outliersMulti_COCS[which(outliersMulti_COCS > 18.467)],
+ outliersMulti_COCI[which(outliersMulti_COCI > 16.266)],
+ outliersMulti_COCR[which(outliersMulti_COCR > 16.266)],
+ outliersMulti_COCP[which(outliersMulti_COCP > 20.515)],
+ outliersMulti_TI[which(outliersMulti_TI > 16.266)],
+ outliersMulti_POQN[which(outliersMulti_POQN > 18.467)],
+ outliersMulti_POQR[which(outliersMulti_POQR > 20.515)]
+ )
> outliersMulti
       1       13      140      152      158       99       35       88      209      179      197
26.13716 25.82238 39.93720 16.40575 18.76010 27.65739 21.95876 21.82650 22.60985 21.07530 20.67126
>
> #============================
> #   UNIVARIATE NORMALITY
> #============================
>
> library(readxl)
> library(dgof)
>
> Data <- read_xlsx("C:/Users/Public/SupplementaryData_2-Dataset.xlsx", sheet = "Matrix", range = "A1:AN244", col_names = TRUE)
>
> ks.test(Data$AOC1, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test
```

```
data:  Data$AOC1
D = 0.84968, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$AOC1, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
> ks.test(Data$AOC2, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test

data:  Data$AOC2
D = 0.89906, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$AOC2, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
> ks.test(Data$AOC3, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test

data:  Data$AOC3
D = 0.84145, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$AOC3, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
> ks.test(Data$AOC4, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test

data:  Data$AOC4
D = 0.9361, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$AOC4, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
> ks.test(Data$AOC5, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test

data:  Data$AOC5
D = 0.85379, p-value < 2.2e-16
```

```
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$AOC5, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
> ks.test(Data$COCR1, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test

data:  Data$COCR1
D = 0.84134, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$COCR1, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
> ks.test(Data$COCI2, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test

data:  Data$COCI2
D = 0.84134, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$COCI2, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
> ks.test(Data$COCS3, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test

data:  Data$COCS3
D = 0.89083, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$COCS3, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
> ks.test(Data$COCS4, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test

data:  Data$COCS4
D = 0.84134, p-value < 2.2e-16
alternative hypothesis: two-sided
```

```
Warning message:
In ks.test(Data$COCS4, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
> ks.test(Data$COCS5, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test

data:  Data$COCS5
D = 0.84134, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$COCS5, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
> ks.test(Data$COCS6, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test

data:  Data$COCS6
D = 0.84134, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$COCS6, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
> ks.test(Data$COCI7, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test

data:  Data$COCI7
D = 0.84134, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$COCI7, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
> ks.test(Data$COCI8, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test

data:  Data$COCI8
D = 0.84134, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$COCI8, "pnorm", alternative = "two.sided") :
```

```
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
> ks.test(Data$COCP9, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test

data:  Data$COCP9
D = 0.84134, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$COCP9, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
> ks.test(Data$COCP10, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test

data:  Data$COCP10
D = 0.84134, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$COCP10, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
> ks.test(Data$COCR11, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test

data:  Data$COCR11
D = 0.84134, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$COCR11, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
> ks.test(Data$COCR12, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test

data:  Data$COCR12
D = 0.84134, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$COCR12, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
```

```
> ks.test(Data$COCP13, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test

data:  Data$COCP13
D = 0.84134, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$COCP13, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
> ks.test(Data$COCP14, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test

data:  Data$COCP14
D = 0.84134, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$COCP14, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
> ks.test(Data$COCP15, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test

data:  Data$COCP15
D = 0.84134, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$COCP15, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
> ks.test(Data$TI1, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test

data:  Data$TI1
D = 0.84134, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$TI1, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
> ks.test(Data$TI2, "pnorm", alternative = "two.sided")
```

```
        One-sample Kolmogorov-Smirnov test

data:  Data$TI2
D = 0.84134, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$TI2, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
> ks.test(Data$TI3, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test

data:  Data$TI3
D = 0.84134, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$TI3, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
> ks.test(Data$POQ1, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test

data:  Data$POQ1
D = 0.84134, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$POQ1, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
> ks.test(Data$POQ2, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test

data:  Data$POQ2
D = 0.84134, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$POQ2, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
> ks.test(Data$POQ3, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test
```

```
data:  Data$POQ3
D = 0.84134, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$POQ3, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
> ks.test(Data$POQ4, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test

data:  Data$POQ4
D = 0.84134, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$POQ4, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
> ks.test(Data$POQ5, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test

data:  Data$POQ5
D = 0.84134, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$POQ5, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
> ks.test(Data$POQ6, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test

data:  Data$POQ6
D = 0.84134, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$POQ6, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
> ks.test(Data$POQ7, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test

data:  Data$POQ7
D = 0.84134, p-value < 2.2e-16
```

```
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$POQ7, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
> ks.test(Data$POQ8, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test

data:  Data$POQ8
D = 0.84134, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$POQ8, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
> ks.test(Data$POQ9, "pnorm", alternative = "two.sided")

        One-sample Kolmogorov-Smirnov test

data:  Data$POQ9
D = 0.84134, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(Data$POQ9, "pnorm", alternative = "two.sided") :
  default ks.test() cannot compute correct p-values with ties;
 see help page for one-sample Kolmogorov test for discrete distributions.
>
> #============================
> #   MULTIVARIATE NORMALITY
> #============================
>
> library(readxl)
> library(psych)
>
> Data <- read_xlsx("C:/Users/Public/SupplementaryData_2-Dataset.xlsx", sheet = "Matrix", range = "A1:AN244", col_names = TRUE)
>
> mardia(data.frame(Data$AOC1, Data$AOC2, Data$AOC3, Data$AOC4, Data$AOC5), na.rm = FALSE, plot = FALSE)
Call: mardia(x = data.frame(Data$AOC1, Data$AOC2, Data$AOC3, Data$AOC4,
    Data$AOC5), na.rm = FALSE, plot = FALSE)

Mardia tests of multivariate skew and kurtosis
Use describe(x) the to get univariate tests
n.obs = 243   num.vars =  5
b1p =  3.73   skew =  151.23  with probability  <=  2.1e-16
 small sample skew =  153.73  with probability <=  8e-17
b2p =  47.32   kurtosis =  11.48  with probability <=  0> mardia(data.frame(Data$COCS3, Data$COCS4, Data$COCS5, Data$COCS6), na.rm = FALSE, plot = FALSE)
```

```
Call: mardia(x = data.frame(Data$COCS3, Data$COCS4, Data$COCS5, Data$COCS6),
    na.rm = FALSE, plot = FALSE)

Mardia tests of multivariate skew and kurtosis
Use describe(x) the to get univariate tests
n.obs = 243   num.vars =  4
b1p =  1.48   skew =  59.98  with probability  <=  7.2e-06
 small sample skew =  61.02  with probability <=  4.9e-06
b2p =  23.46   kurtosis =  -0.6  with probability <=  0.55> mardia(data.frame(Data$COCI2, Data$COCI7, Data$COCI8), na.rm = FALSE, plot = FALSE)
Call: mardia(x = data.frame(Data$COCI2, Data$COCI7, Data$COCI8), na.rm = FALSE,
    plot = FALSE)

Mardia tests of multivariate skew and kurtosis
Use describe(x) the to get univariate tests
n.obs = 243   num.vars =  3
b1p =  0.65   skew =  26.2  with probability  <=  0.0035
 small sample skew =  26.69  with probability <=  0.0029
b2p =  15.87   kurtosis =  1.24  with probability <=  0.21> mardia(data.frame(Data$COCR1, Data$COCR11, Data$COCR12), na.rm = FALSE, plot = FALSE)
Call: mardia(x = data.frame(Data$COCR1, Data$COCR11, Data$COCR12),
    na.rm = FALSE, plot = FALSE)

Mardia tests of multivariate skew and kurtosis
Use describe(x) the to get univariate tests
n.obs = 243   num.vars =  3
b1p =  0.33   skew =  13.56  with probability  <=  0.19
 small sample skew =  13.81  with probability <=  0.18
b2p =  13.98   kurtosis =  -1.46  with probability <=  0.15> mardia(data.frame(Data$COCP9, Data$COCP10, Data$COCP13, Data$COCP14, Data$COCP15), na.rm = FALSE, plot = FALSE)
Call: mardia(x = data.frame(Data$COCP9, Data$COCP10, Data$COCP13, Data$COCP14,
    Data$COCP15), na.rm = FALSE, plot = FALSE)

Mardia tests of multivariate skew and kurtosis
Use describe(x) the to get univariate tests
n.obs = 243   num.vars =  5
b1p =  3.28   skew =  132.64  with probability  <=  2.8e-13
 small sample skew =  134.83  with probability <=  1.2e-13
b2p =  42.88   kurtosis =  7.34  with probability <=  2.1e-13> mardia(data.frame(Data$TI1, Data$TI2, Data$TI3), na.rm = FALSE, plot = FALSE)
Call: mardia(x = data.frame(Data$TI1, Data$TI2, Data$TI3), na.rm = FALSE,
    plot = FALSE)

Mardia tests of multivariate skew and kurtosis
Use describe(x) the to get univariate tests
n.obs = 243   num.vars =  3
b1p =  0.64   skew =  25.84  with probability  <=  0.004
 small sample skew =  26.32  with probability <=  0.0033
b2p =  20.01   kurtosis =  7.14  with probability <=  9.6e-13> mardia(data.frame(Data$POQ2, Data$POQ4, Data$POQ5, Data$POQ8), na.rm = FALSE, plot = FALSE)
Call: mardia(x = data.frame(Data$POQ2, Data$POQ4, Data$POQ5, Data$POQ8),
    na.rm = FALSE, plot = FALSE)

Mardia tests of multivariate skew and kurtosis
Use describe(x) the to get univariate tests
```

```
n.obs = 243   num.vars =  4
b1p =  0.77   skew =  31.37  with probability  <=  0.051
 small sample skew =  31.91  with probability <=  0.044
b2p =  24.24   kurtosis =  0.27  with probability <=  0.79> mardia(data.frame(Data$POQ1, Data$POQ3, Data$POQ6, Data$POQ7, Data$POQ9), na.rm = FALSE, plot = FALSE)
Call: mardia(x = data.frame(Data$POQ1, Data$POQ3, Data$POQ6, Data$POQ7,
    Data$POQ9), na.rm = FALSE, plot = FALSE)

Mardia tests of multivariate skew and kurtosis
Use describe(x) the to get univariate tests
n.obs = 243   num.vars =  5
b1p =  2.38   skew =  96.58  with probability  <=  1.1e-07
 small sample skew =  98.17  with probability <=  6.6e-08
b2p =  39.88   kurtosis =  4.55  with probability <=  5.4e-06>
>
> #==================================
> #   CONFIRMATORY FACTOR ANALYSIS
> #==================================
>
> library(readxl)
> library(psych)
> library(lavaan)
> library(semTools)
> library(GPArotation)
> library(arm)
>
> Data <- read_xlsx("C:/Users/Public/SupplementaryData_2-Dataset.xlsx", sheet = "Matrix", range = "A1:AN244", col_names = TRUE)
>
> Model <-'
+ POQN =~ POQ2 + POQ4 + POQ5 + POQ8
+ POQR =~ POQ1 + POQ3 + POQ6 + POQ7 + POQ9
+ POQ =~ NA*POQN + equal("POQ =~ POQN")*POQR
+ POQ ~~ 1*POQ
+ AOC =~ AOC1 + AOC2 + AOC3 + AOC4 + AOC5
+ TI =~ TI1 + TI2 + TI3
+ COCS =~ COCS3 + COCS4 + COCS5 + COCS6
+ COCI =~ COCI2 + COCI7 + COCI8
+ COCR =~ COCR1 + COCR11 + COCR12
+ COCP =~ COCP9 + COCP10 + COCP13 + COCP14 + COCP15
+ '
> fitModel <- cfa(model = Model, data = Data, estimator = "MLM", se = "robust.sem", test = "Satorra.Bentler")
> summary(fitModel, fit.measures = TRUE, standardized = TRUE, rsq = TRUE)
lavaan 0.6-9 ended normally after 59 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        87
  Number of equality constraints                     1

  Number of observations                           243
```

```
Model Test User Model:
                                              Standard      Robust
  Test Statistic                               739.303     639.850
  Degrees of freedom                               442         442
  P-value (Chi-square)                           0.000       0.000
  Scaling correction factor                                  1.155
       Satorra-Bentler correction

Model Test Baseline Model:

  Test statistic                              4825.695    4629.301
  Degrees of freedom                               496         496
  P-value                                        0.000       0.000
  Scaling correction factor                                  1.042

User Model versus Baseline Model:

  Comparative Fit Index (CFI)                    0.931       0.952
  Tucker-Lewis Index (TLI)                       0.923       0.946

  Robust Comparative Fit Index (CFI)                         0.947
  Robust Tucker-Lewis Index (TLI)                            0.940

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)             -11887.984  -11887.984
  Loglikelihood unrestricted model (H1)     -11518.333  -11518.333

  Akaike (AIC)                               23947.969   23947.969
  Bayesian (BIC)                             24248.372   24248.372
  Sample-size adjusted Bayesian (BIC)        23975.764   23975.764

Root Mean Square Error of Approximation:

  RMSEA                                          0.053       0.043
  90 Percent confidence interval - lower         0.046       0.036
  90 Percent confidence interval - upper         0.059       0.050
  P-value RMSEA <= 0.05                          0.254       0.960

  Robust RMSEA                                               0.046
  90 Percent confidence interval - lower                     0.038
  90 Percent confidence interval - upper                     0.054

Standardized Root Mean Square Residual:

  SRMR                                           0.050       0.050

Parameter Estimates:

  Standard errors                            Robust.sem
```

```
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  POQN =~
    POQ2              1.000                               1.195    0.524
    POQ4              1.554    0.171    9.068    0.000    1.856    0.797
    POQ5              1.039    0.142    7.312    0.000    1.242    0.563
    POQ8              1.411    0.155    9.087    0.000    1.685    0.751
  POQR =~
    POQ1              1.000                               1.324    0.567
    POQ3              1.195    0.131    9.150    0.000    1.582    0.668
    POQ6              1.260    0.139    9.077    0.000    1.668    0.748
    POQ7              1.428    0.150    9.545    0.000    1.890    0.792
    POQ9              1.366    0.144    9.478    0.000    1.808    0.800
  POQ =~
    POQN    (.10.)    1.145    0.112   10.271    0.000    0.959    0.959
    POQR    (POQ=)    1.145    0.112   10.271    0.000    0.865    0.865
  AOC =~
    AOC1              1.000                               0.846    0.724
    AOC2              0.998    0.072   13.874    0.000    0.844    0.848
    AOC3              1.097    0.078   14.042    0.000    0.927    0.872
    AOC4              0.862    0.082   10.481    0.000    0.729    0.687
    AOC5              1.154    0.086   13.411    0.000    0.976    0.874
  TI =~
    TI1               1.000                               1.240    0.949
    TI2               1.043    0.038   27.129    0.000    1.292    0.906
    TI3               1.023    0.037   27.543    0.000    1.268    0.922
  COCS =~
    COCS3             1.000                               0.601    0.483
    COCS4             1.386    0.193    7.179    0.000    0.834    0.704
    COCS5             0.924    0.156    5.918    0.000    0.556    0.478
    COCS6             1.216    0.171    7.101    0.000    0.732    0.606
  COCI =~
    COCI2             1.000                               0.914    0.670
    COCI7             1.201    0.104   11.602    0.000    1.098    0.837
    COCI8             1.106    0.095   11.639    0.000    1.012    0.819
  COCR =~
    COCR1             1.000                               0.877    0.650
    COCR11            0.991    0.098   10.158    0.000    0.869    0.653
    COCR12            1.306    0.108   12.066    0.000    1.145    0.864
  COCP =~
    COCP9             1.000                               0.897    0.762
    COCP10            0.894    0.072   12.346    0.000    0.802    0.707
    COCP13            0.957    0.077   12.350    0.000    0.858    0.707
    COCP14            0.816    0.067   12.253    0.000    0.732    0.699
    COCP15            1.011    0.083   12.138    0.000    0.907    0.730

Covariances:
```

	Estimate	Std.Err	z-value	P(>|z|)	Std.lv	Std.all
POQ ~~						
AOC	-0.284	0.059	-4.785	0.000	-0.336	-0.336
TI	0.562	0.085	6.653	0.000	0.454	0.454
COCS	-0.188	0.057	-3.317	0.001	-0.313	-0.313
COCI	-0.189	0.074	-2.556	0.011	-0.207	-0.207
COCR	-0.299	0.071	-4.229	0.000	-0.341	-0.341
COCP	-0.248	0.072	-3.432	0.001	-0.276	-0.276
AOC ~~						
TI	-0.550	0.077	-7.157	0.000	-0.524	-0.524
COCS	0.146	0.043	3.384	0.001	0.286	0.286
COCI	0.177	0.064	2.789	0.005	0.229	0.229
COCR	0.235	0.058	4.066	0.000	0.317	0.317
COCP	0.252	0.065	3.882	0.000	0.332	0.332
TI ~~						
COCS	-0.411	0.076	-5.437	0.000	-0.551	-0.551
COCI	-0.480	0.091	-5.290	0.000	-0.424	-0.424
COCR	-0.622	0.092	-6.747	0.000	-0.572	-0.572
COCP	-0.624	0.084	-7.398	0.000	-0.562	-0.562
COCS ~~						
COCI	0.464	0.085	5.433	0.000	0.844	0.844
COCR	0.475	0.084	5.621	0.000	0.900	0.900
COCP	0.486	0.079	6.129	0.000	0.900	0.900
COCI ~~						
COCR	0.602	0.088	6.810	0.000	0.750	0.750
COCP	0.685	0.091	7.518	0.000	0.836	0.836
COCR ~~						
COCP	0.708	0.089	7.915	0.000	0.901	0.901

Variances:

	Estimate	Std.Err	z-value	P(>|z|)	Std.lv	Std.all
POQ	1.000				1.000	1.000
.POQ2	3.772	0.335	11.261	0.000	3.772	0.726
.POQ4	1.980	0.354	5.593	0.000	1.980	0.365
.POQ5	3.316	0.322	10.285	0.000	3.316	0.683
.POQ8	2.199	0.283	7.778	0.000	2.199	0.436
.POQ1	3.704	0.312	11.886	0.000	3.704	0.679
.POQ3	3.109	0.374	8.317	0.000	3.109	0.554
.POQ6	2.188	0.328	6.668	0.000	2.188	0.440
.POQ7	2.126	0.304	7.002	0.000	2.126	0.373
.POQ9	1.840	0.272	6.761	0.000	1.840	0.360
.AOC1	0.651	0.091	7.122	0.000	0.651	0.476
.AOC2	0.278	0.036	7.777	0.000	0.278	0.280
.AOC3	0.270	0.041	6.605	0.000	0.270	0.239
.AOC4	0.594	0.055	10.798	0.000	0.594	0.528
.AOC5	0.294	0.063	4.675	0.000	0.294	0.236
.TI1	0.170	0.045	3.797	0.000	0.170	0.100
.TI2	0.365	0.063	5.826	0.000	0.365	0.180
.TI3	0.285	0.055	5.150	0.000	0.285	0.151
.COCS3	1.187	0.107	11.099	0.000	1.187	0.766

.COCS4	0.707	0.093	7.559	0.000	0.707	0.504
.COCS5	1.045	0.085	12.284	0.000	1.045	0.772
.COCS6	0.922	0.097	9.544	0.000	0.922	0.633
.COCI2	1.024	0.122	8.409	0.000	1.024	0.551
.COCI7	0.516	0.084	6.159	0.000	0.516	0.300
.COCI8	0.502	0.076	6.578	0.000	0.502	0.329
.COCR1	1.048	0.111	9.432	0.000	1.048	0.577
.COCR11	1.014	0.100	10.106	0.000	1.014	0.573
.COCR12	0.444	0.077	5.744	0.000	0.444	0.253
.COCP9	0.580	0.067	8.612	0.000	0.580	0.419
.COCP10	0.642	0.079	8.136	0.000	0.642	0.500
.COCP13	0.736	0.094	7.856	0.000	0.736	0.500
.COCP14	0.562	0.064	8.818	0.000	0.562	0.512
.COCP15	0.719	0.081	8.856	0.000	0.719	0.467
.POQN	0.115	0.157	0.733	0.464	0.081	0.081
.POQR	0.441	0.181	2.440	0.015	0.252	0.252
AOC	0.715	0.105	6.782	0.000	1.000	1.000
TI	1.537	0.118	13.058	0.000	1.000	1.000
COCS	0.362	0.096	3.756	0.000	1.000	1.000
COCI	0.836	0.131	6.359	0.000	1.000	1.000
COCR	0.769	0.127	6.064	0.000	1.000	1.000
COCP	0.804	0.106	7.589	0.000	1.000	1.000

R-Square:

	Estimate
POQ2	0.274
POQ4	0.635
POQ5	0.317
POQ8	0.564
POQ1	0.321
POQ3	0.446
POQ6	0.560
POQ7	0.627
POQ9	0.640
AOC1	0.524
AOC2	0.720
AOC3	0.761
AOC4	0.472
AOC5	0.764
TI1	0.900
TI2	0.820
TI3	0.849
COCS3	0.234
COCS4	0.496
COCS5	0.228
COCS6	0.367
COCI2	0.449
COCI7	0.700
COCI8	0.671
COCR1	0.423

```
    COCR11            0.427
    COCR12            0.747
    COCP9             0.581
    COCP10            0.500
    COCP13            0.500
    COCP14            0.488
    COCP15            0.533
    POQN              0.919
    POQR              0.748

> reliability(fitModel)
            POQN      POQR       AOC        TI      COCS      COCI      COCR      COCP
alpha  0.7458151 0.8358243 0.8965537 0.9457958 0.6604264 0.8093835 0.7673307 0.8434909
omega  0.7602920 0.8407141 0.8995494 0.9462018 0.6574960 0.8174859 0.7692312 0.8446022
omega2 0.7602920 0.8407141 0.8995494 0.9462018 0.6574960 0.8174859 0.7692312 0.8446022
omega3 0.7699548 0.8410820 0.9017311 0.9466617 0.6492080 0.8244498 0.7640912 0.8442763
avevar 0.4509831 0.5170399 0.6438856 0.8543198 0.3298998 0.6002095 0.5307014 0.5223407
> reliabilityL2(fitModel, "POQ")
       omegaL1        omegaL2 partialOmegaL1
     0.7893499      0.9041912      0.8726918
>
> #==================================
> #   STRUCTURAL EQUATION MODELING
> #==================================
>
> library(readxl)
> library(lavaan)
> library(semTools)
> library(car)
>
> Data <- read_xlsx("C:/Users/Public/SupplementaryData_2-Dataset.xlsx", sheet = "Matrix", range = "A1:AN244", col_names = TRUE)
>
> Model <- '
+ POQN =~ POQ2 + POQ4 + POQ5 + POQ8
+ POQR =~ POQ1 + POQ3 + POQ6 + POQ7 + POQ9
+ POQ =~ NA*POQN + equal("POQ =~ POQN")*POQR
+ POQ ~~ 1*POQ
+ AOC =~ AOC1 + AOC2 + AOC3 + AOC4 + AOC5
+ TI =~ TI1 + TI2 + TI3
+ TI ~~ 1*TI
+ COCS =~ COCS3 + COCS4 + COCS5 + COCS6
+ COCI =~ COCI2 + COCI7 + COCI8
+ COCR =~ COCR1 + COCR11 + COCR12
+ COCP =~ COCP9 + COCP10 + COCP13 + COCP14 + COCP15
+ TI ~ c*POQ
+ AOC ~ a*POQ
+ COCS ~ d*POQ
+ COCI ~ f*POQ
+ COCR ~ h*POQ
+ COCP ~ j*POQ
```

```
+ TI ~ b*AOC
+ TI ~ e*COCS
+ TI ~ g*COCI
+ TI ~ i*COCR
+ TI ~ k*COCP
+ ab := a*b
+ de := d*e
+ fg := f*g
+ hi := h*i
+ jk := j*k
+ total := c + (a*b) + (d*e) + (f*g) + (h*i) + (j*k)
+ '
>
> fit <- sem(model = Model, data = Data, estimator = "MLM", se = "robust.sem", test = "satorra.bentler")
> summary(fit, fit.measures = TRUE, standardized = TRUE, rsq = TRUE)
lavaan 0.6-9 ended normally after 85 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        76
  Number of equality constraints                     1

  Number of observations                           243

Model Test User Model:
                                            Standard      Robust
  Test Statistic                             908.993     802.003
  Degrees of freedom                             453         453
  P-value (Chi-square)                         0.000       0.000
  Scaling correction factor                                1.133
       Satorra-Bentler correction

Model Test Baseline Model:

  Test statistic                            4825.695    4629.301
  Degrees of freedom                             496         496
  P-value                                      0.000       0.000
  Scaling correction factor                                1.042

User Model versus Baseline Model:

  Comparative Fit Index (CFI)                  0.895       0.916
  Tucker-Lewis Index (TLI)                     0.885       0.908

  Robust Comparative Fit Index (CFI)                       0.908
  Robust Tucker-Lewis Index (TLI)                          0.899

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)            -11972.829  -11972.829
```

```
  Loglikelihood unrestricted model (H1)     -11518.333  -11518.333

  Akaike (AIC)                               24095.659   24095.659
  Bayesian (BIC)                             24357.638   24357.638
  Sample-size adjusted Bayesian (BIC)        24119.899   24119.899

Root Mean Square Error of Approximation:

  RMSEA                                          0.064       0.056
  90 Percent confidence interval - lower         0.058       0.050
  90 Percent confidence interval - upper         0.070       0.062
  P-value RMSEA <= 0.05                          0.000       0.043

  Robust RMSEA                                               0.060
  90 Percent confidence interval - lower                     0.053
  90 Percent confidence interval - upper                     0.067

Standardized Root Mean Square Residual:

  SRMR                                           0.101       0.101

Parameter Estimates:

  Standard errors                            Robust.sem
  Information                                  Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  POQN =~
    POQ2              1.000                               1.129    0.498
    POQ4              1.669    0.233    7.148    0.000    1.884    0.809
    POQ5              1.042    0.166    6.289    0.000    1.177    0.534
    POQ8              1.521    0.209    7.292    0.000    1.718    0.766
  POQR =~
    POQ1              1.000                               1.229    0.523
    POQ3              1.208    0.150    8.077    0.000    1.485    0.626
    POQ6              1.298    0.165    7.861    0.000    1.595    0.715
    POQ7              1.629    0.189    8.597    0.000    2.002    0.838
    POQ9              1.542    0.180    8.585    0.000    1.896    0.838
  POQ =~
    POQN     (.10.)   0.377    0.090    4.191    0.000    0.334    0.334
    POQR     (POQ=)   0.377    0.090    4.191    0.000    0.306    0.306
  AOC =~
    AOC1              1.000                               0.847    0.724
    AOC2              0.997    0.072   13.894    0.000    0.844    0.848
    AOC3              1.097    0.078   14.024    0.000    0.929    0.873
    AOC4              0.860    0.082   10.473    0.000    0.728    0.687
    AOC5              1.152    0.086   13.393    0.000    0.976    0.874
  TI =~
```

```
    TI1               1.000                             1.317    0.957
    TI2               1.022    0.035   29.242    0.000    1.346    0.911
    TI3               1.002    0.034   29.610    0.000    1.320    0.926
  COCS =~
    COCS3             1.000                             0.593    0.477
    COCS4             1.404    0.196    7.144    0.000    0.833    0.703
    COCS5             0.960    0.161    5.950    0.000    0.569    0.489
    COCS6             1.227    0.174    7.044    0.000    0.728    0.603
  COCI =~
    COCI2             1.000                             0.910    0.667
    COCI7             1.217    0.107   11.348    0.000    1.108    0.844
    COCI8             1.105    0.098   11.303    0.000    1.005    0.814
  COCR =~
    COCR1             1.000                             0.872    0.647
    COCR11            0.990    0.099   10.023    0.000    0.864    0.649
    COCR12            1.321    0.111   11.907    0.000    1.152    0.870
  COCP =~
    COCP9             1.000                             0.893    0.759
    COCP10            0.900    0.074   12.166    0.000    0.804    0.709
    COCP13            0.971    0.079   12.345    0.000    0.867    0.715
    COCP14            0.825    0.068   12.178    0.000    0.736    0.703
    COCP15            1.003    0.084   11.916    0.000    0.895    0.721

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  TI ~
    POQ        (c)    0.677    2.476    0.274    0.784    0.514    0.514
  AOC ~
    POQ        (a)   -0.285    0.064   -4.433    0.000   -0.336   -0.336
  COCS ~
    POQ        (d)   -0.567    0.077   -7.330    0.000   -0.956   -0.956
  COCI ~
    POQ        (f)   -0.765    0.074  -10.402    0.000   -0.841   -0.841
  COCR ~
    POQ        (h)   -0.810    0.071  -11.456    0.000   -0.929   -0.929
  COCP ~
    POQ        (j)   -0.859    0.061  -14.152    0.000   -0.962   -0.962
  TI ~
    AOC        (b)   -0.525    0.102   -5.146    0.000   -0.338   -0.338
    COCS       (e)   -0.010    1.602   -0.006    0.995   -0.004   -0.004
    COCI       (g)    0.220    0.285    0.773    0.439    0.152    0.152
    COCR       (i)   -0.161    0.663   -0.243    0.808   -0.107   -0.107
    COCP       (k)    0.067    1.147    0.058    0.954    0.045    0.045

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
    POQ               1.000                             1.000    1.000
   .TI                1.000                             0.577    0.577
   .POQ2              3.862    0.351   10.999    0.000    3.862    0.752
   .POQ4              1.870    0.459    4.070    0.000    1.870    0.345
```

.POQ5	3.471	0.342	10.149	0.000	3.471	0.715
.POQ8	2.084	0.347	6.007	0.000	2.084	0.414
.POQ1	4.011	0.324	12.365	0.000	4.011	0.726
.POQ3	3.417	0.389	8.794	0.000	3.417	0.608
.POQ6	2.438	0.349	6.990	0.000	2.438	0.489
.POQ7	1.705	0.305	5.583	0.000	1.705	0.299
.POQ9	1.530	0.249	6.148	0.000	1.530	0.299
.AOC1	0.649	0.092	7.092	0.000	0.649	0.475
.AOC2	0.278	0.036	7.779	0.000	0.278	0.281
.AOC3	0.268	0.041	6.537	0.000	0.268	0.237
.AOC4	0.594	0.055	10.812	0.000	0.594	0.528
.AOC5	0.295	0.064	4.628	0.000	0.295	0.237
.TI1	0.158	0.047	3.394	0.001	0.158	0.084
.TI2	0.370	0.064	5.810	0.000	0.370	0.170
.TI3	0.291	0.056	5.247	0.000	0.291	0.143
.COCS3	1.197	0.105	11.352	0.000	1.197	0.773
.COCS4	0.708	0.095	7.491	0.000	0.708	0.505
.COCS5	1.030	0.085	12.140	0.000	1.030	0.761
.COCS6	0.928	0.099	9.404	0.000	0.928	0.637
.COCI2	1.032	0.123	8.414	0.000	1.032	0.555
.COCI7	0.495	0.085	5.842	0.000	0.495	0.287
.COCI8	0.515	0.080	6.467	0.000	0.515	0.337
.COCR1	1.056	0.112	9.454	0.000	1.056	0.581
.COCR11	1.022	0.101	10.138	0.000	1.022	0.578
.COCR12	0.427	0.077	5.539	0.000	0.427	0.243
.COCP9	0.586	0.069	8.497	0.000	0.586	0.424
.COCP10	0.639	0.080	7.940	0.000	0.639	0.497
.COCP13	0.720	0.094	7.666	0.000	0.720	0.489
.COCP14	0.555	0.064	8.742	0.000	0.555	0.506
.COCP15	0.740	0.083	8.958	0.000	0.740	0.480
.POQN	1.133	0.291	3.894	0.000	0.889	0.889
.POQR	1.369	0.308	4.448	0.000	0.906	0.906
.AOC	0.636	0.103	6.169	0.000	0.887	0.887
.COCS	0.030	0.031	0.972	0.331	0.085	0.085
.COCI	0.243	0.054	4.500	0.000	0.293	0.293
.COCR	0.104	0.051	2.063	0.039	0.137	0.137
.COCP	0.060	0.036	1.683	0.092	0.075	0.075

R-Square:

	Estimate
TI	0.423
POQ2	0.248
POQ4	0.655
POQ5	0.285
POQ8	0.586
POQ1	0.274
POQ3	0.392
POQ6	0.511
POQ7	0.701
POQ9	0.701

```
    AOC1              0.525
    AOC2              0.719
    AOC3              0.763
    AOC4              0.472
    AOC5              0.763
    TI1               0.916
    TI2               0.830
    TI3               0.857
    COCS3             0.227
    COCS4             0.495
    COCS5             0.239
    COCS6             0.363
    COCI2             0.445
    COCI7             0.713
    COCI8             0.663
    COCR1             0.419
    COCR11            0.422
    COCR12            0.757
    COCP9             0.576
    COCP10            0.503
    COCP13            0.511
    COCP14            0.494
    COCP15            0.520
    POQN              0.111
    POQR              0.094
    AOC               0.113
    COCS              0.915
    COCI              0.707
    COCR              0.863
    COCP              0.925

Defined Parameters:
                  Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
    ab               0.150    0.044    3.391    0.001    0.114    0.114
    de               0.006    0.909    0.006    0.995    0.004    0.004
    fg              -0.168    0.217   -0.778    0.436   -0.128   -0.128
    hi               0.131    0.537    0.243    0.808    0.099    0.099
    jk              -0.057    0.985   -0.058    0.954   -0.043   -0.043
    total            0.737    0.089    8.302    0.000    0.560    0.560

>
```