Emotion Appraisal Performance (EAP) Profiles

MS

28/04/2021

Data Cleaning

Cleaning Trait level

```
load("data_Trait.Rda")
##### GET INFO
# Raw data, including Pilot -> age, gender etc
mean(data_Trait[data_Trait$Age<=15,]$Age, na.rm = T)
```

```
## [1] 12.88032
```

```
sd(data_Trait[data_Trait$Age<=15,]$Age, na.rm = T)
```

```
## [1] 0.7035422
```

```
table(data_Trait$Year)
```

```
##
##   1   2
## 169 214
```

```
table(data_Trait$Gender)
```

```
##
##   1   2
## 191 192
```

```
# Raw data, excluding Pilot
data_Trait <- data_Trait[!data_Trait$uid<137098,] # Excluding the pilot
######## START DATA CLEANING ############
###### 1. MAD procedure per construct
data_Trait <- data_Trait[!c(data_Trait$uid==137827|data_Trait$uid==137266|data_Trait$uid==137111|data_T
data_Trait <- data_Trait[!c(data_Trait$uid==137836),]  # Is excluded due to answering 1 on all question
# descriptives
mean(data_Trait[data_Trait$Age<=15,]$Age, na.rm = T)
```

```
## [1] 12.92082
```

```
sd(data_Trait[data_Trait$Age<=15,]$Age, na.rm = T)
```

```
## [1] 0.7047364
```

```
table(data_Trait$Year)
```

```
##
```

```
##   1   2
## 141 206
```

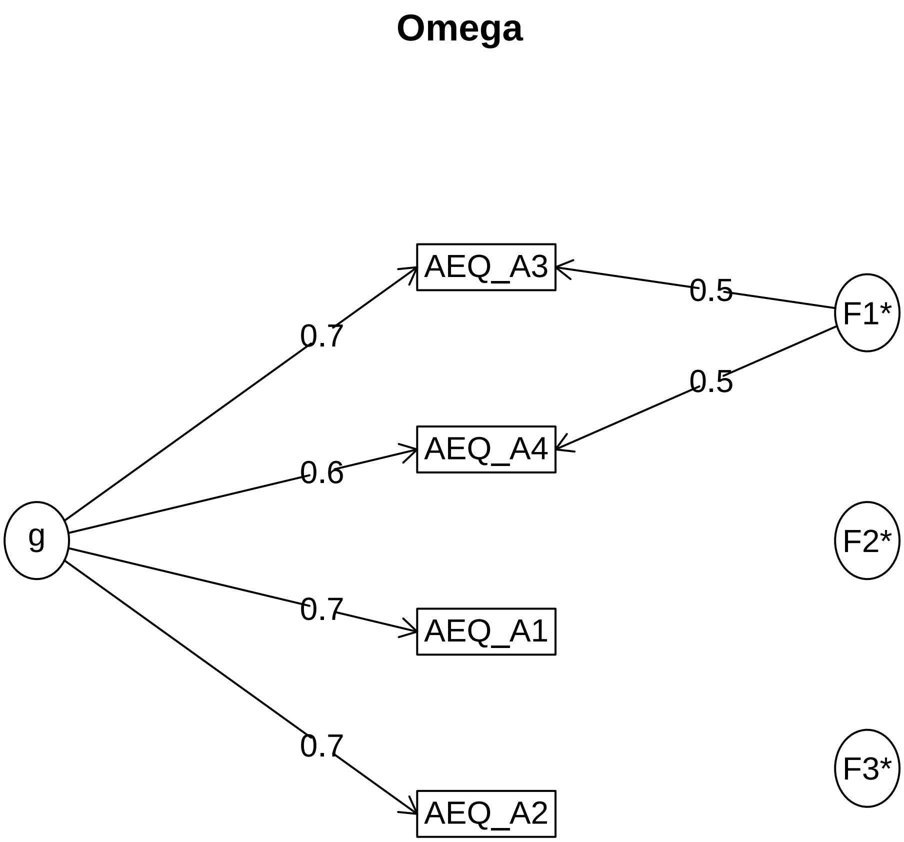
```r
table(data_Trait$Gender)
```

```
##
##   1   2
## 168 179
```

```r
# data set with outliers removed:
data_Trait_OutRm <- data_Trait
# Anger: AEQ_Ang
data_Trait_OutRm$AEQ_Ang_mean <- (data_Trait_OutRm$AEQ_A1+data_Trait_OutRm$AEQ_A2+data_Trait_OutRm$AEQ_
data_Trait$AEQ_Ang_mean <- data_Trait_OutRm$AEQ_Ang_mean
MAD <- mad(data_Trait_OutRm$AEQ_Ang_mean, na.rm = TRUE)
data_Trait_OutRm$AEQ_Ang_MADfin <- (data_Trait_OutRm$AEQ_Ang_mean-median(data_Trait_OutRm$AEQ_Ang_mean))
data_Trait_OutRm[abs(data_Trait_OutRm$AEQ_Ang_MADfin)>3,]$AEQ_Ang_mean <- NA # >= 4
# Anxiety: AMAS
data_Trait$AMAS_mean <- (data_Trait$AMAS1+data_Trait$AMAS2+data_Trait$AMAS3+data_Trait$AMAS4+data_Trait
                         data_Trait$AMAS7+data_Trait$AMAS8)/8
data_Trait_OutRm$AMAS_mean <- data_Trait$AMAS_mean
MAD <- mad(data_Trait_OutRm$AMAS_mean, na.rm = TRUE)
data_Trait_OutRm$AMAS_MADfin <- (data_Trait_OutRm$AMAS_mean-median(data_Trait_OutRm$AMAS_mean))/MAD
data_Trait_OutRm[abs(data_Trait_OutRm$AMAS_MADfin)>3,]$AMAS_mean <- NA # >= 3.5
# Boredom: AEQ_Bor
data_Trait_OutRm$AEQ_Bor_mean <- (data_Trait_OutRm$AEQ_B1+data_Trait_OutRm$AEQ_B2+data_Trait_OutRm$AEQ_
data_Trait$AEQ_Bor_mean <- data_Trait_OutRm$AEQ_Bor_mean
MAD <- mad(data_Trait_OutRm$AEQ_Bor_mean, na.rm = TRUE)
data_Trait_OutRm$AEQ_Bor_MADfin <- (data_Trait_OutRm$AEQ_Bor_mean-median(data_Trait_OutRm$AEQ_Bor_mean,
# Enjoyment: AEQ_Enj
data_Trait_OutRm$AEQ_Enj_mean <- (data_Trait_OutRm$AEQ_E1+data_Trait_OutRm$AEQ_E2+data_Trait_OutRm$AEQ_
data_Trait$AEQ_Enj_mean <- data_Trait_OutRm$AEQ_Enj_mean
MAD <- mad(data_Trait_OutRm$AEQ_Enj_mean, na.rm = TRUE)
data_Trait_OutRm$AEQ_Enj_MADfin <- (data_Trait_OutRm$AEQ_Enj_mean-median(data_Trait_OutRm$AEQ_Enj_mean))
data_Trait_OutRm[abs(data_Trait_OutRm$AEQ_Enj_MADfin)>3,]$AEQ_Enj_mean <- NA # >= 4.75
# Control: SELF-Efficacy
data_Trait_OutRm$SELFE_mean <- (data_Trait_OutRm$SelfE1+data_Trait_OutRm$SelfE2+
                                 data_Trait_OutRm$SelfE3+
                               data_Trait_OutRm$SelfE4+data_Trait_OutRm$SelfE5+
                               data_Trait_OutRm$SelfE6+
                                 data_Trait_OutRm$SelfE7+data_Trait_OutRm$SelfE8)/8
data_Trait$SELFE_mean <- data_Trait_OutRm$SELFE_mean
MAD <- mad(data_Trait_OutRm$SELFE_mean)
data_Trait_OutRm$SELFE_MADfin <- (data_Trait_OutRm$SELFE_mean-median(data_Trait_OutRm$SELFE_mean))/MAD
data_Trait_OutRm[abs(data_Trait_OutRm$SELFE_MADfin)>3,]$SELFE_mean <- NA # >= 2
# Control: SELF-Concept
data_Trait_OutRm$SELFC_mean <- (data_Trait_OutRm$SelfC1+data_Trait_OutRm$SelfC2+data_Trait_OutRm$SelfC3
                                 data_Trait_OutRm$SelfC5)/5
data_Trait$SELFC_mean <- data_Trait_OutRm$SELFC_mean
MAD <- mad(data_Trait_OutRm$SELFC_mean)
data_Trait_OutRm$SELFC_MADfin <- (data_Trait_OutRm$SELFC_mean-median(data_Trait_OutRm$SELFC_mean))/MAD
# Value: Instrumental
data_Trait_OutRm$VAL_mean <- (data_Trait_OutRm$Val1+data_Trait_OutRm$Val2+data_Trait_OutRm$Val3+data_Tr
data_Trait$VAL_mean <- data_Trait_OutRm$VAL_mean
MAD <- mad(data_Trait_OutRm$VAL_mean)
```

```r
data_Trait_OutRm$VAL_MADfin <- (data_Trait_OutRm$VAL_mean-median(data_Trait_OutRm$VAL_mean))/MAD # none
# Value: Intrinsic
data_Trait_OutRm$INT_mean <- (data_Trait_OutRm$Int1+data_Trait_OutRm$Int2+data_Trait_OutRm$Int3+data_Tra
data_Trait$INT_mean <- data_Trait_OutRm$INT_mean
MAD <- mad(data_Trait_OutRm$INT_mean)
data_Trait_OutRm$INT_MADfin <- (data_Trait_OutRm$INT_mean-median(data_Trait_OutRm$INT_mean))/MAD # none
# Performance: Grade
MAD <- mad(data_Trait_OutRm$Grade, na.rm = TRUE)
data_Trait_OutRm$Grade_MADfin <- (data_Trait_OutRm$Grade-median(data_Trait_OutRm$Grade, na.rm = TRUE))/M

###### 2. Check McDonald's omega, skeness, kurtosis
# Anger: AEQ_Ang
summary(omega(m = data_Trait[c(39:42)])) # omega
```




```
## Omega
## omega(m = data_Trait[c(39:42)])
## Alpha:                 0.81
## G.6:                   0.77
## Omega Hierarchical:    0.74
## Omega H asymptotic:    0.88
## Omega Total            0.84
##
## With eigenvalues of:
##     g   F1*   F2*   F3*
## 1.877 0.485 0.000 0.046
## The degrees of freedom for the model is -3  and the fit was  0
## The number of observations was  347  with Chi Square =  0  with prob <  NA
```

```
##
## The root mean square of the residuals is  0.01
## The df corrected root mean square of the residuals is  NA
## Explained Common Variance of the general factor =  0.78
##
##  Total, General and Subset omega for each subset
##                                                  g  F1* F2*  F3*
## Omega total for total scores and subscales    0.84 0.83  NA 0.65
## Omega general for total scores and subscales  0.74 0.54  NA 0.65
## Omega group for total scores and subscales    0.10 0.29  NA 0.00
```

```r
skew(as.numeric(data_Trait$AEQ_Ang_mean)) # scale level
```

```
## [1] 1.124085
```

```r
kurtosi(as.numeric(data_Trait$AEQ_Ang_mean))
```

```
## [1] 0.8448558
```

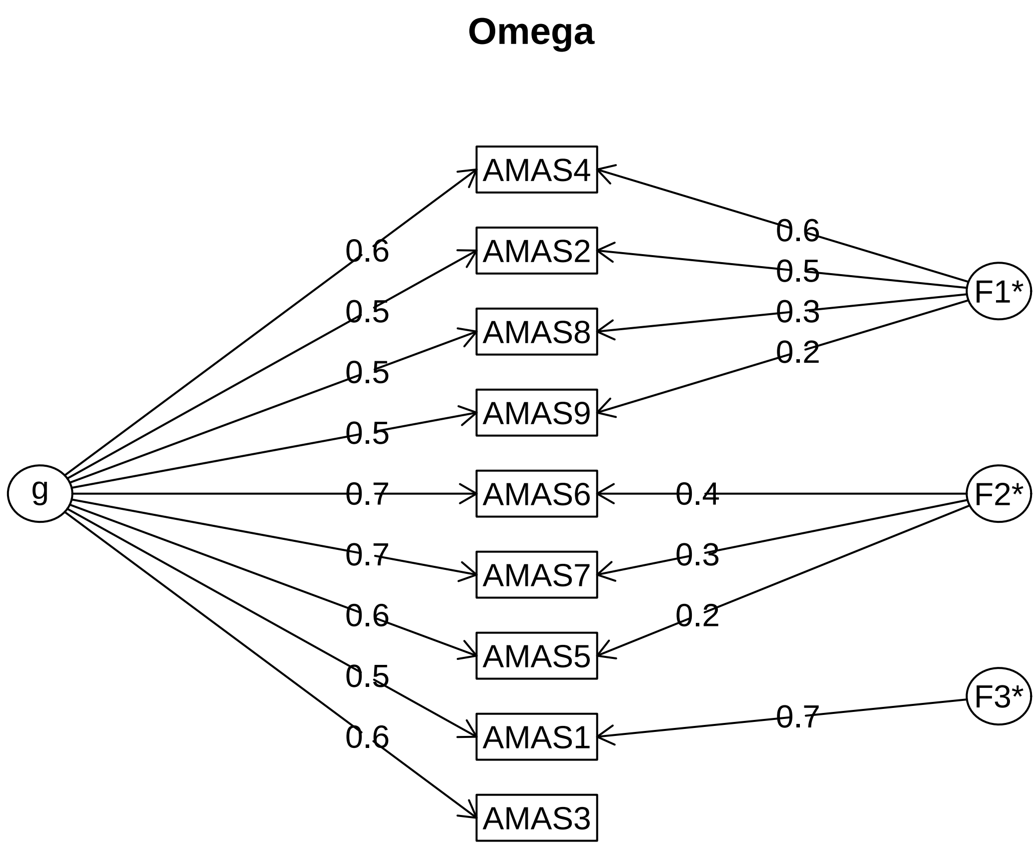
```r
lapply(data_Trait[c(39:42)],skew) # item level
```

```
## $AEQ_A1
## [1] 1.032784
##
## $AEQ_A2
## [1] 0.3079086
##
## $AEQ_A3
## [1] 1.382853
##
## $AEQ_A4
## [1] 1.442298
```

```r
lapply(data_Trait[c(39:42)],kurtosi)
```

```
## $AEQ_A1
## [1] 0.009196354
##
## $AEQ_A2
## [1] -0.8972436
##
## $AEQ_A3
## [1] 0.8873641
##
## $AEQ_A4
## [1] 1.195012
```

```r
# Anxiety: AMAS
summary(omega(m = data_Trait[c(4:12)])) # omega
```




```
## Omega
## omega(m = data_Trait[c(4:12)])
## Alpha:                 0.86
## G.6:                   0.86
## Omega Hierarchical:    0.72
## Omega H asymptotic:    0.81
## Omega Total            0.89
##
## With eigenvalues of:
##    g  F1*  F2*  F3*
## 3.04 0.75 0.36 0.54
## The degrees of freedom for the model is 12  and the fit was  0.07
## The number of observations was  347  with Chi Square =  22.56  with prob <  0.03
##
## The root mean square of the residuals is  0.02
## The df corrected root mean square of the residuals is  0.05
##
## RMSEA and the  0.9 confidence intervals are  0.05 0.015 0.082
## BIC =  -47.64Explained Common Variance of the general factor =  0.65
##
##  Total, General and Subset omega for each subset
##                                                  g  F1*  F2*  F3*
## Omega total for total scores and subscales    0.89 0.75 0.78 0.69
## Omega general for total scores and subscales  0.72 0.49 0.63 0.43
## Omega group for total scores and subscales    0.11 0.27 0.15 0.25
```

```r
skew(as.numeric(data_Trait$AMAS_mean)) # scale level
```

```
## [1] 1.120515
```

```r
kurtosi(as.numeric(data_Trait$AMAS_mean))
```

```
## [1] 1.094191
```

```r
lapply(data_Trait[c(4:12)],skew) # item level
```

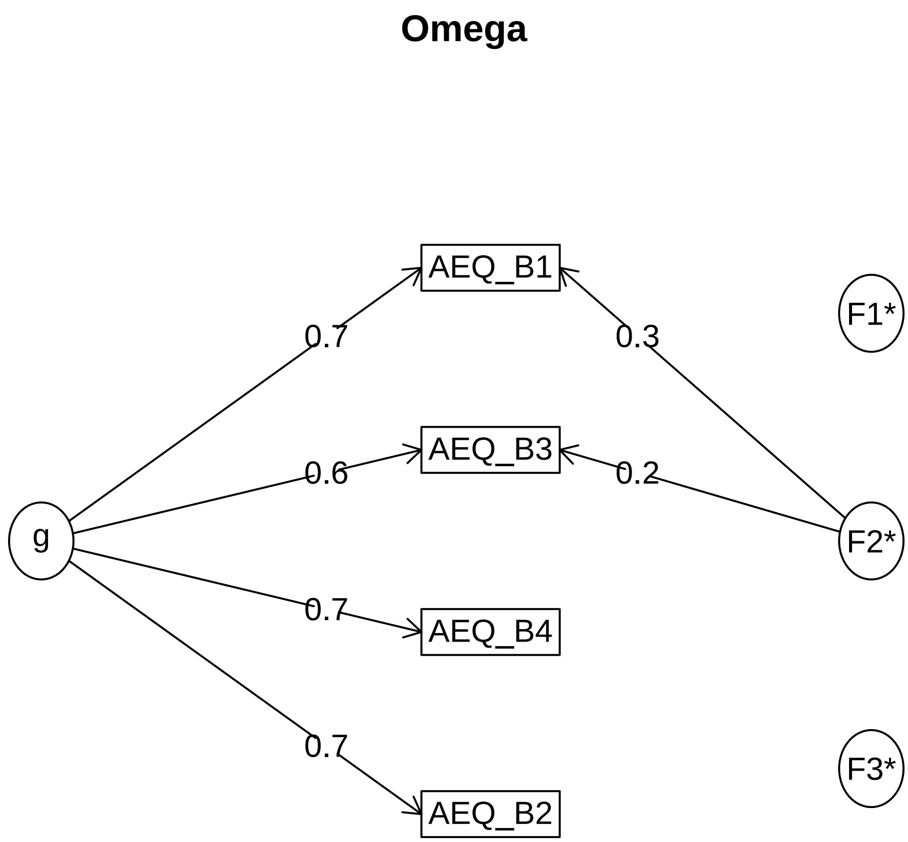
```
## $AMAS1
## [1] 2.484056
##
## $AMAS2
## [1] 0.1724814
##
## $AMAS3
## [1] 0.9320675
##
## $AMAS4
## [1] 0.3993813
##
## $AMAS5
## [1] 1.038723
##
## $AMAS6
## [1] 2.872552
##
## $AMAS7
## [1] 2.096665
##
## $AMAS8
## [1] 1.250945
##
## $AMAS9
## [1] 2.138801
```

```r
lapply(data_Trait[c(4:12)],kurtosi)
```

```
## $AMAS1
## [1] 6.261257
##
## $AMAS2
## [1] -1.123776
##
## $AMAS3
## [1] -0.2031096
##
## $AMAS4
## [1] -0.9018012
##
## $AMAS5
## [1] 0.4700348
##
## $AMAS6
## [1] 8.348623
```

```
##
## $AMAS7
## [1] 3.753721
##
## $AMAS8
## [1] 0.5653574
##
## $AMAS9
## [1] 4.432982
```

```r
# Boredom: AEQ_Bor
summary(omega(m = data_Trait[c(30:33)])) # omega
```




```
## Omega
## omega(m = data_Trait[c(30:33)])
## Alpha:                 0.78
## G.6:                   0.73
## Omega Hierarchical:    0.77
## Omega H asymptotic:    0.96
## Omega Total            0.8
##
## With eigenvalues of:
##      g    F1*    F2*    F3*
## 1.8805 0.0049 0.1235 0.0511
## The degrees of freedom for the model is -3  and the fit was  0
## The number of observations was  347  with Chi Square =  0  with prob <  NA
##
## The root mean square of the residuals is  0
```

```
## The df corrected root mean square of the residuals is  NA
## Explained Common Variance of the general factor =  0.91
##
##  Total, General and Subset omega for each subset
##                                                   g F1*  F2*  F3*
## Omega total for total scores and subscales    0.80  NA 0.62 0.72
## Omega general for total scores and subscales  0.77  NA 0.54 0.72
## Omega group for total scores and subscales    0.03  NA 0.09 0.00
```

```r
skew(as.numeric(data_Trait$AEQ_Bor_mean)) # scale level
```

```
## [1] 0.6414259
```

```r
kurtosi(as.numeric(data_Trait$AEQ_Bor_mean))
```

```
## [1] -0.06085742
```

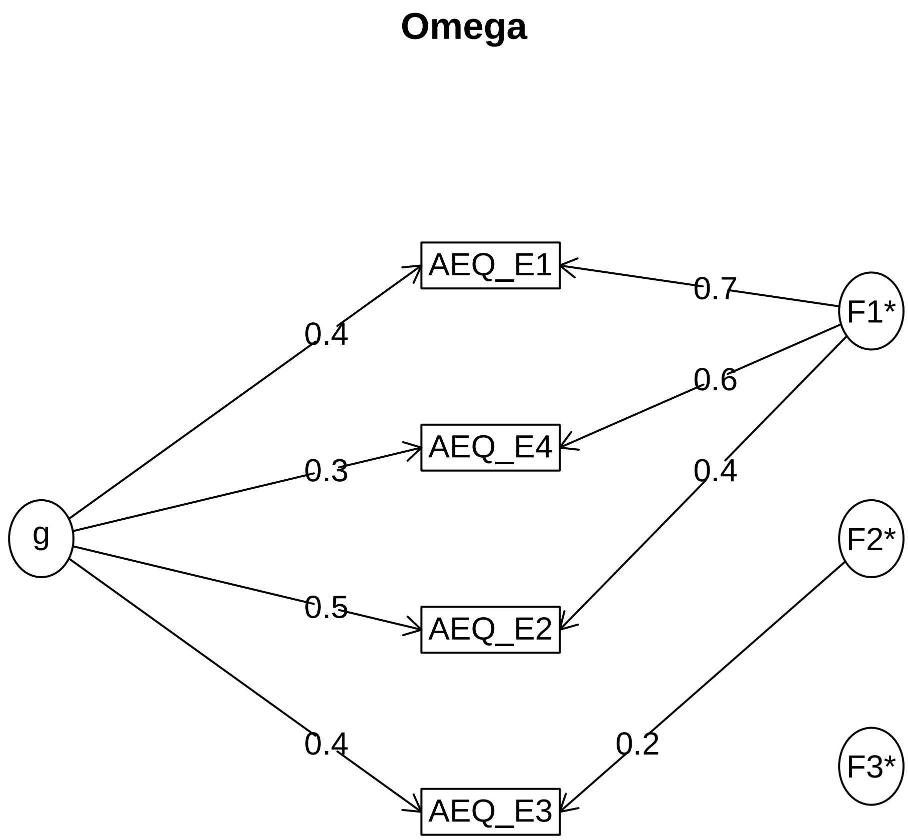
```r
lapply(data_Trait[c(30:33)],skew) # item level
```

```
## $AEQ_B1
## [1] 0.2534494
##
## $AEQ_B2
## [1] 1.428423
##
## $AEQ_B3
## [1] 0.5558892
##
## $AEQ_B4
## [1] 0.5374542
```

```r
lapply(data_Trait[c(30:33)],kurtosi)
```

```
## $AEQ_B1
## [1] -0.9245111
##
## $AEQ_B2
## [1] 1.520998
##
## $AEQ_B3
## [1] -0.6601932
##
## $AEQ_B4
## [1] -0.7668666
```

```r
# Enjoyment: AEQ_Enj
summary(omega(data_Trait[c(35:38)])) #   omega
```




```
## Omega
## omega(m = data_Trait[c(35:38)])
## Alpha:                 0.63
## G.6:                   0.6
## Omega Hierarchical:    0.31
## Omega H asymptotic:    0.44
## Omega Total            0.69
##
## With eigenvalues of:
##     g   F1*   F2*   F3*
## 0.610 1.000 0.054 0.010
## The degrees of freedom for the model is -3  and the fit was  0
## The number of observations was  347  with Chi Square =  0  with prob <  NA
##
## The root mean square of the residuals is  0
## The df corrected root mean square of the residuals is  NA
## Explained Common Variance of the general factor =  0.36
##
##  Total, General and Subset omega for each subset
##                                                  g  F1*  F2* F3*
## Omega total for total scores and subscales    0.69 0.74 0.17  NA
## Omega general for total scores and subscales  0.31 0.23 0.13  NA
## Omega group for total scores and subscales    0.39 0.50 0.04  NA
```

```
skew(as.numeric(data_Trait$AEQ_Enj_mean))
```

```
## [1] 0.4451874
```

```r
kurtosi(as.numeric(data_Trait$AEQ_Enj_mean))
```

```
## [1] 0.3085275
```

```r
lapply(data_Trait[c(35:38)],skew) # item level
```

```
## $AEQ_E1
## [1] 0.6456197
##
## $AEQ_E2
## [1] 0.338357
##
## $AEQ_E3
## [1] -1.506531
##
## $AEQ_E4
## [1] 1.226786
```

```r
lapply(data_Trait[c(35:38)],kurtosi)
```

```
## $AEQ_E1
## [1] -0.4814458
##
## $AEQ_E2
## [1] -0.3646058
##
## $AEQ_E3
## [1] 1.796091
##
## $AEQ_E4
## [1] 0.7916341
```

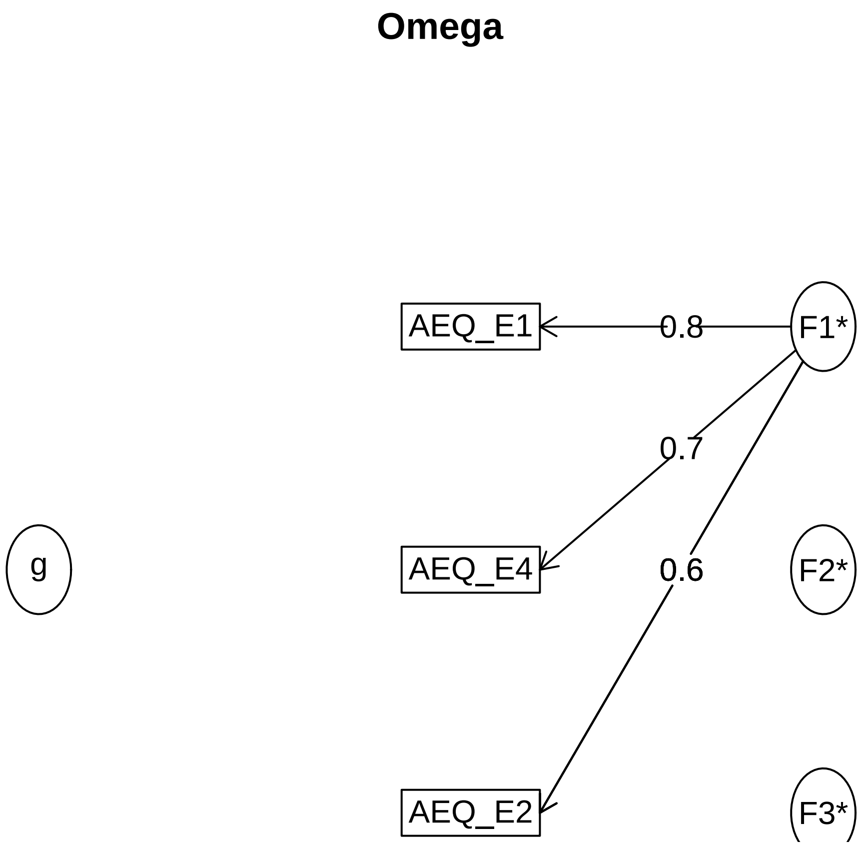
```r
### McDonald's omega low -> exclude item E3
# Enjoyment: AEQ_Enj_3 items
data_Trait$AEQ_Enj_3_mean <- (data_Trait$AEQ_E1+data_Trait$AEQ_E2+data_Trait$AEQ_E4)/3
data_Trait_OutRm$AEQ_Enj_3_mean <- data_Trait$AEQ_Enj_3_mean
# outliers:
MAD <- mad(data_Trait_OutRm$AEQ_Enj_3_mean)
data_Trait_OutRm$AEQ_Enj_3_MADfin <- (data_Trait_OutRm$AEQ_Enj_3_mean-median(data_Trait_OutRm$AEQ_Enj_3
data_Trait_OutRm[abs(data_Trait_OutRm$AEQ_Enj_3_MADfin)>3,]$AEQ_Enj_3_mean
```

```
## [1] 5 5
```

```r
data_Trait_OutRm[abs(data_Trait_OutRm$AEQ_Enj_3_MADfin)>3,]$uid
```

```
## [1] 137135 137296
```

```r
data_Trait_OutRm[abs(data_Trait_OutRm$AEQ_Enj_3_MADfin)>3,]$AEQ_Enj_3_mean <- NA
# internal consistency & distribution
summary(omega(data_Trait[c(35,36,38)])) # omega
```




```
## Omega
## omega(m = data_Trait[c(35, 36, 38)])
## Alpha:                 0.72
## G.6:                   0.63
## Omega Hierarchical:    0.03
## Omega H asymptotic:    0.05
## Omega Total            0.73
##
## With eigenvalues of:
##     g   F1*   F2*   F3*
## 0.067 1.342 0.027 0.000
## The degrees of freedom for the model is -3  and the fit was  0
## The number of observations was  347  with Chi Square =  0  with prob <  NA
##
## The root mean square of the residuals is  0
## The df corrected root mean square of the residuals is  NA
## Explained Common Variance of the general factor =  0.05
##
##  Total, General and Subset omega for each subset
##                                                  g  F1* F2* F3*
## Omega total for total scores and subscales    0.73 0.73  NA  NA
## Omega general for total scores and subscales  0.03 0.03  NA  NA
## Omega group for total scores and subscales    0.69 0.69  NA  NA
```

```r
skew(as.numeric(data_Trait$AEQ_Enj_3_mean)) # scale level
```

```
## [1] 0.6962254
```

```r
kurtosi(as.numeric(data_Trait$AEQ_Enj_3_mean))
```

```
## [1] 0.2131584
```

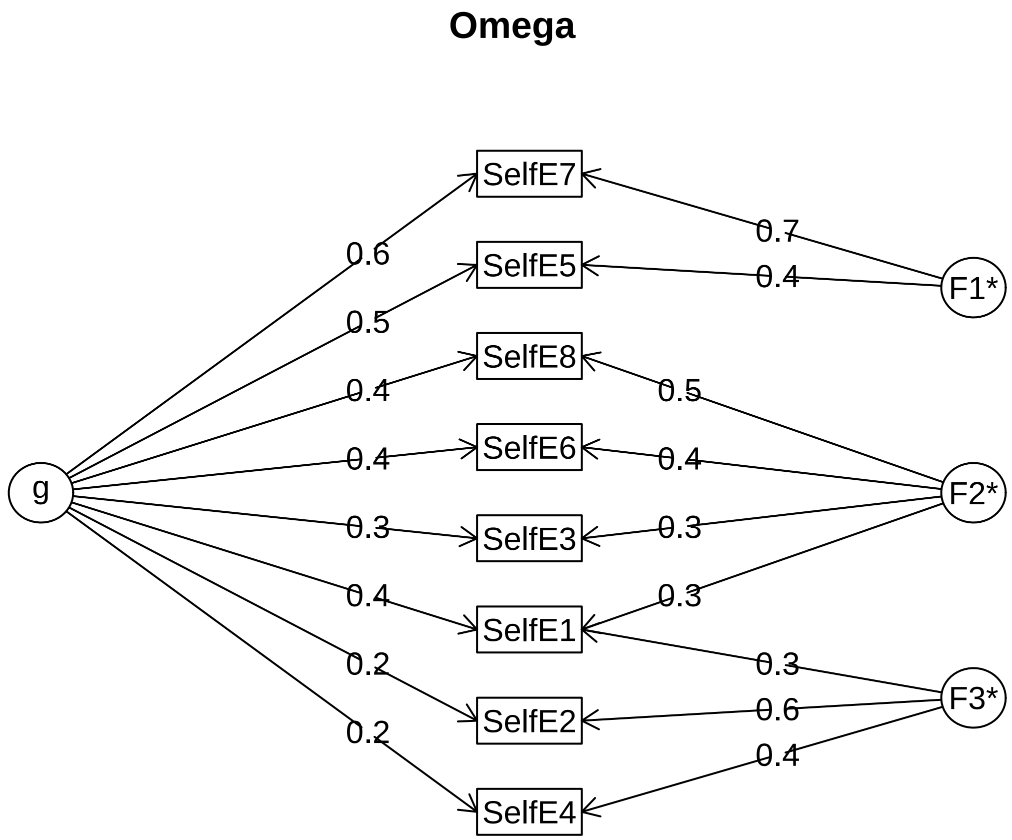
```r
lapply(data_Trait[c(35,36,38)],skew) # item level
```

```
## $AEQ_E1
## [1] 0.6456197
##
## $AEQ_E2
## [1] 0.338357
##
## $AEQ_E4
## [1] 1.226786
```

```r
lapply(data_Trait[c(35,36,38)],kurtosi)
```

```
## $AEQ_E1
## [1] -0.4814458
##
## $AEQ_E2
## [1] -0.3646058
##
## $AEQ_E4
## [1] 0.7916341
```

```r
# Control: Self-Efficacy
summary(omega(data_Trait[c(22:29)])) #   omega
```



```
## Omega
## omega(m = data_Trait[c(22:29)])
## Alpha:                 0.69
## G.6:                   0.69
## Omega Hierarchical:    0.44
## Omega H asymptotic:    0.57
## Omega Total            0.76
##
## With eigenvalues of:
##    g  F1*  F2*  F3*
## 1.23 0.74 0.67 0.66
## The degrees of freedom for the model is 7  and the fit was  0.01
## The number of observations was  347  with Chi Square =  2.69  with prob <  0.91
##
## The root mean square of the residuals is  0.01
## The df corrected root mean square of the residuals is  0.04
##
## RMSEA and the  0.9 confidence intervals are  0 0 0.025
## BIC =  -38.26Explained Common Variance of the general factor =  0.37
##
##  Total, General and Subset omega for each subset
##                                                  g  F1*  F2*  F3*
## Omega total for total scores and subscales    0.76 0.80 0.62 0.47
## Omega general for total scores and subscales  0.44 0.38 0.28 0.07
## Omega group for total scores and subscales    0.24 0.42 0.34 0.39
```

```
skew(as.numeric(data_Trait$SELFE_mean)) # scale level
```

```
## [1] -0.9677814
```

```
kurtosi(as.numeric(data_Trait$SELFE_mean))
```

```
## [1] 1.000474
```

```
lapply(data_Trait[c(22:29)],skew) # item level
```

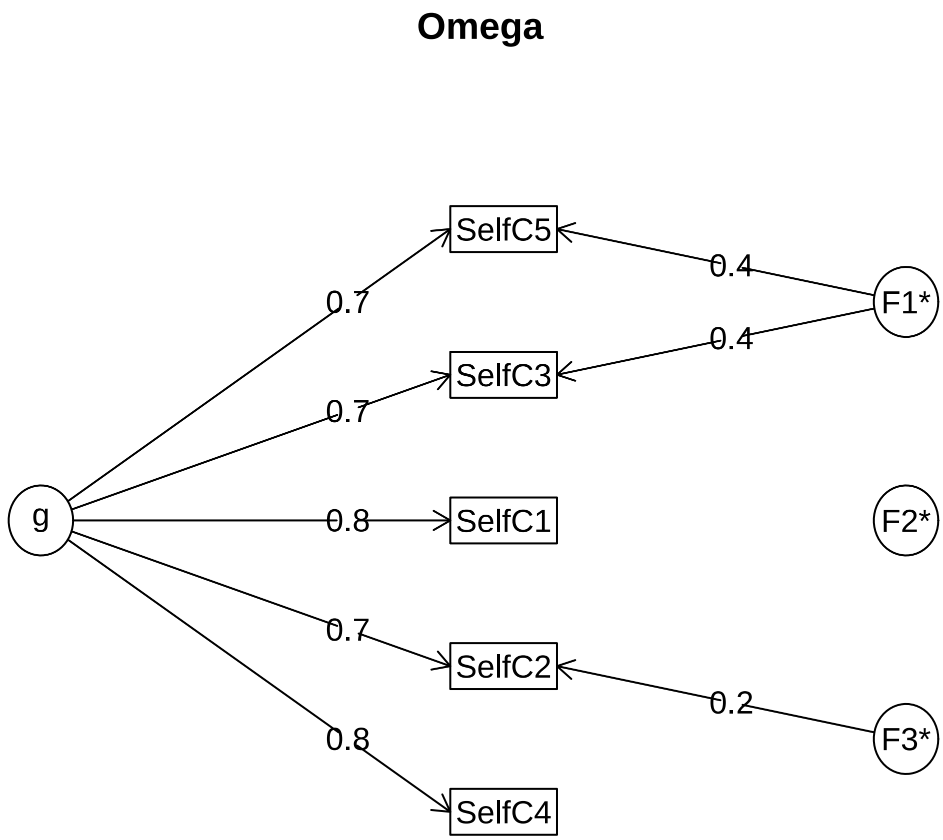
```
## $SelfE1
## [1] -0.4328752
##
## $SelfE2
## [1] -0.8796628
##
## $SelfE3
## [1] -0.382596
##
## $SelfE4
## [1] -0.60542
##
## $SelfE5
## [1] -1.021401
##
## $SelfE6
## [1] -0.1661444
##
## $SelfE7
## [1] -0.2207649
```

```
##
## $SelfE8
## [1] -0.170165
```

```r
lapply(data_Trait[c(22:29)],kurtosi)
```

```
## $SelfE1
## [1] -0.9757984
##
## $SelfE2
## [1] 0.6099122
##
## $SelfE3
## [1] -0.9322313
##
## $SelfE4
## [1] -0.2514012
##
## $SelfE5
## [1] 0.1895921
##
## $SelfE6
## [1] -0.8120543
##
## $SelfE7
## [1] -0.907005
##
## $SelfE8
## [1] -0.52688
```

```r
# Control: Self-Concept
summary(omega(data_Trait[c(17:21)])) #   omega
```




```
## Omega
## omega(m = data_Trait[c(17:21)])
## Alpha:                 0.88
## G.6:                   0.86
## Omega Hierarchical:    0.84
## Omega H asymptotic:    0.93
## Omega Total            0.9
##
## With eigenvalues of:
##    g  F1*  F2*  F3*
## 2.82 0.35 0.00 0.11
## The degrees of freedom for the model is -2  and the fit was  0
## The number of observations was  347  with Chi Square =  0  with prob <  NA
##
## The root mean square of the residuals is  0.01
## The df corrected root mean square of the residuals is  NA
## Explained Common Variance of the general factor =  0.86
##
##  Total, General and Subset omega for each subset
##                                                  g  F1* F2*  F3*
## Omega total for total scores and subscales    0.90 0.86  NA 0.72
## Omega general for total scores and subscales  0.84 0.77  NA 0.71
## Omega group for total scores and subscales    0.04 0.09  NA 0.00
```

```
skew(as.numeric(data_Trait$SELFC_mean)) # scale level
```

```
## [1] -0.1570266
```

```r
kurtosi(as.numeric(data_Trait$SELFC_mean))
```

```
## [1] -0.6232701
```

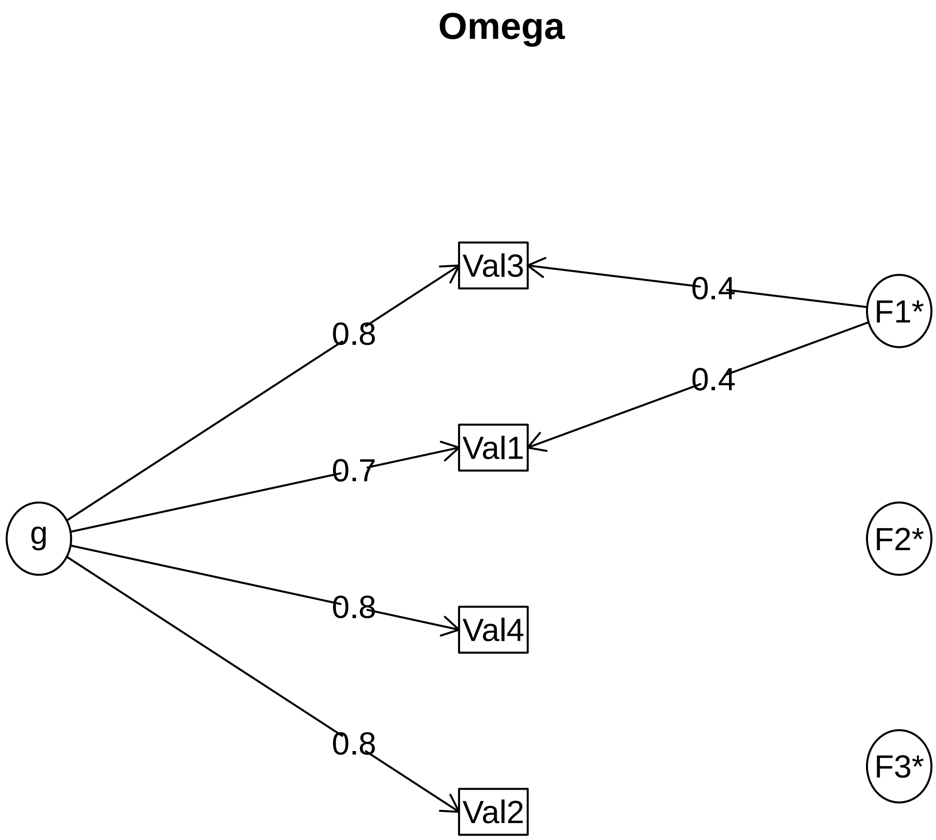
```r
lapply(data_Trait[c(17:21)],skew) # item level
```

```
## $SelfC1
## [1] -0.6403493
##
## $SelfC2
## [1] -0.4078936
##
## $SelfC3
## [1] -0.25179
##
## $SelfC4
## [1] 0.2275204
##
## $SelfC5
## [1] 0.1987716
```

```r
lapply(data_Trait[c(17:21)],kurtosi)
```

```
## $SelfC1
## [1] -0.2554144
##
## $SelfC2
## [1] -0.1207348
##
## $SelfC3
## [1] -0.6985789
##
## $SelfC4
## [1] -1.113323
##
## $SelfC5
## [1] -0.6491652
```

```r
# Value: Instrumental
summary(omega(data_Trait[c(43:46)])) #   omega
```




```
## Omega
## omega(m = data_Trait[c(43:46)])
## Alpha:                 0.86
## G.6:                   0.83
## Omega Hierarchical:    0.84
## Omega H asymptotic:    0.94
## Omega Total            0.89
##
## With eigenvalues of:
##      g    F1*    F2*    F3*
## 2.3593 0.2606 0.0950 0.0054
## The degrees of freedom for the model is -3  and the fit was  0
## The number of observations was  347  with Chi Square =  0  with prob <  NA
##
## The root mean square of the residuals is  0
## The df corrected root mean square of the residuals is  NA
## Explained Common Variance of the general factor =  0.87
##
##  Total, General and Subset omega for each subset
##                                                  g  F1*  F2* F3*
## Omega total for total scores and subscales    0.89 0.84 0.76  NA
## Omega general for total scores and subscales  0.84 0.69 0.76  NA
## Omega group for total scores and subscales    0.05 0.15 0.00  NA
```

```
skew(as.numeric(data_Trait$VAL_mean)) # scale level
```

```
## [1] -0.4605487
```

```r
kurtosi(as.numeric(data_Trait$VAL_mean))
```

```
## [1] 0.1206481
```

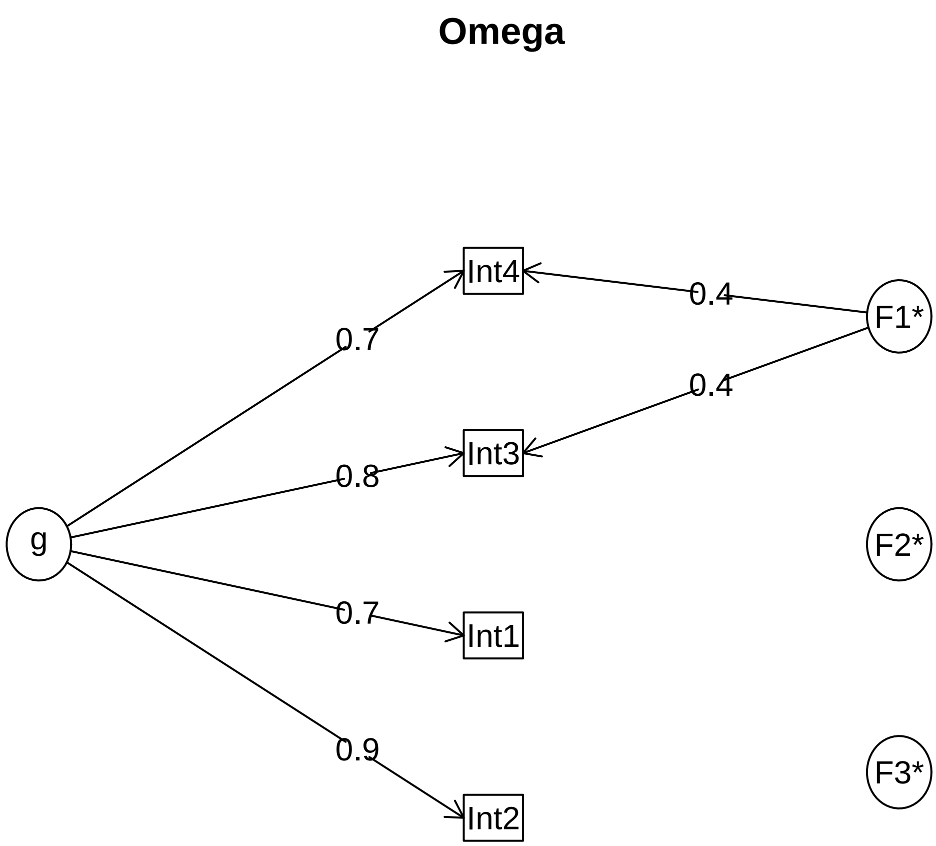
```r
lapply(data_Trait[c(43:46)],skew) # item level
```

```
## $Val1
## [1] -0.4316452
##
## $Val2
## [1] -0.6119715
##
## $Val3
## [1] -0.2406252
##
## $Val4
## [1] -0.4879892
```

```r
lapply(data_Trait[c(43:46)],kurtosi)
```

```
## $Val1
## [1] -0.6013761
##
## $Val2
## [1] 0.04918312
##
## $Val3
## [1] -0.7105032
##
## $Val4
## [1] -0.2295752
```

```r
# Value: Intrinsic
summary(omega(data_Trait[c(13:16)])) #   omega
```




```
## Omega
## omega(m = data_Trait[c(13:16)])
## Alpha:                 0.86
## G.6:                   0.84
## Omega Hierarchical:    0.83
## Omega H asymptotic:    0.93
## Omega Total            0.89
##
## With eigenvalues of:
##     g   F1*   F2*   F3*
## 2.359 0.333 0.000 0.077
## The degrees of freedom for the model is -3  and the fit was  0
## The number of observations was  347  with Chi Square =  0  with prob <  NA
##
## The root mean square of the residuals is  0
## The df corrected root mean square of the residuals is  NA
## Explained Common Variance of the general factor =  0.85
##
##  Total, General and Subset omega for each subset
##                                                  g  F1* F2*  F3*
## Omega total for total scores and subscales    0.89 0.84  NA 0.78
## Omega general for total scores and subscales  0.83 0.65  NA 0.78
## Omega group for total scores and subscales    0.06 0.20  NA 0.00
```

```
skew(as.numeric(data_Trait$INT_mean)) # scale level
```

```
## [1] 0.2781868
```

```
kurtosi(as.numeric(data_Trait$INT_mean))
```

```
## [1] -0.08620528
```

```
lapply(data_Trait[c(13:16)],skew) # item level
```

```
## $Int1
## [1] 1.126097
##
## $Int2
## [1] 0.4373667
##
## $Int3
## [1] 0.1811409
##
## $Int4
## [1] -0.4529891
```

```
lapply(data_Trait[c(13:16)],kurtosi)
```

```
## $Int1
## [1] 0.9923062
##
## $Int2
## [1] -0.4497411
##
## $Int3
## [1] -0.8055637
##
## $Int4
## [1] -0.4182785
```

```
# Performance: Grade
skew(as.numeric(data_Trait$Grade))
```

```
## [1] -0.1780777
```

```
kurtosi(as.numeric(data_Trait$Grade))
```

```
## [1] -0.1540494
```

```
# Save -> include all questionnaires, no change
data_Trait_OutRm <- data_Trait_OutRm[c(1,54,56,58,71,62,64,66,68,51)]
data_Trait_usable <- data_Trait
save(data_Trait_usable,data_Trait_OutRm, file = "data_Trait_usable.Rda")
```

Cleaning State level

No choice block

```
load("data_State_wide_both.Rda")
data_State <- data_State_wide_nC
# Raw data, excluding Pilot
data_State <- data_State[!data_State$uid<137098,] # Excluding the pilot
data_State <- data_State[!c(data_State$uid==137827|data_State$uid==137266|data_State$uid==137111|data_S
# New data_frame: all outliers removed:
```

```
data_State_OutRm <- data_State
###### 1. MAD procedure per construct
# Anger
data_State_OutRm$ang_mean <- (data_State_OutRm$ang1+data_State$ang2)/2
MAD <- mad(data_State_OutRm$ang_mean)
data_State_OutRm$ang_mean_MADfin <- (data_State_OutRm$ang_mean-median(data_State_OutRm$ang_mean))/MAD
sort(data_State_OutRm[abs(data_State_OutRm$ang_mean_MADfin)>3,]$ang_mean)  # >= .688
```

```
##  [1] 0.6888021 0.7015625 0.7054687 0.7171875 0.7210937 0.7281250 0.7335938
##  [8] 0.7390625 0.7437500 0.7519562 0.7554688 0.7783565 0.7862776 0.7880711
## [15] 0.8252315 0.8281250 0.8750000 0.8923611 0.8976563 0.9083333 0.9108796
## [22] 0.9114583 0.9199219 0.9437500 0.9461806 0.9491393 0.9593750 0.9635417
## [29] 0.9726562 0.9728009 0.9796875 0.9890951 0.9937500
```

```
data_State_OutRm[!is.na(data_State_OutRm$ang_mean)&abs(data_State_OutRm$ang_mean_MADfin)>3,]$ang_mean <-
# Anxiety
data_State_OutRm$anx_mean <- (data_State_OutRm$anx1+data_State_OutRm$anx2)/2
MAD <- mad(data_State_OutRm$anx_mean)
data_State_OutRm$anx_mean_MADfin <- (data_State_OutRm$anx_mean-median(data_State_OutRm$anx_mean))/MAD
sort(data_State_OutRm[abs(data_State_OutRm$anx_mean_MADfin)>3,]$anx_mean) # >= .43
```

```
##  [1] 0.4326389 0.4438477 0.4462891 0.4633789 0.4646991 0.4657360 0.4777778
##  [8] 0.4798611 0.4849537 0.4894206 0.5015649 0.5038071 0.5070313 0.5101562
## [15] 0.5166377 0.5242187 0.5391236 0.5476562 0.5481073 0.5546875 0.5561343
## [22] 0.5630787 0.5804688 0.6007813 0.6168981 0.6375000 0.6409722 0.6909722
## [29] 0.7117187 0.7375000 0.7796875 0.8000000 0.8054687 0.8257813 0.8680556
## [36] 0.8972081 0.9182943
```

```
data_State_OutRm[!is.na(data_State_OutRm$anx_mean)&abs(data_State_OutRm$anx_mean_MADfin)>3,]$anx_mean <-
# Boredom
data_State_OutRm$bor_mean <- (data_State_OutRm$bor1+data_State$bor2)/2
MAD <- mad(data_State_OutRm$bor_mean)
data_State_OutRm$bor_mean_MADfin <- (data_State_OutRm$bor_mean-median(data_State_OutRm$bor_mean))/MAD
sort(data_State_OutRm[abs(data_State_OutRm$bor_mean_MADfin)>3,]$bor_mean) # none
```

```
## numeric(0)
```

```
# Enjoyment
data_State_OutRm$enj_mean <- (data_State_OutRm$enj1+data_State_OutRm$enj2)/2
MAD <- mad(data_State_OutRm$enj_mean)
data_State_OutRm$enj_mean_MADfin <- (data_State_OutRm$enj_mean-median(data_State_OutRm$enj_mean))/MAD
data_State_OutRm[abs(data_State_OutRm$enj_mean_MADfin)>3,]$enj_mean # none
```

```
## numeric(0)
```

```
# Value
data_State_OutRm$val_mean <- (data_State_OutRm$val1+data_State_OutRm$val2)/2
MAD <- mad(data_State_OutRm$val_mean)
data_State_OutRm$val_mean_MADfin <- (data_State_OutRm$val_mean-median(data_State_OutRm$val_mean))/MAD
sort(data_State_OutRm[abs(data_State_OutRm$val_mean_MADfin)>3,]$val_mean) # none
```

```
## numeric(0)
```

```
# Control
MAD <- mad(data_State_OutRm$cont)
data_State_OutRm$cont_MADfin <- (data_State_OutRm$cont-median(data_State_OutRm$cont))/MAD
sort(data_State_OutRm[abs(data_State_OutRm$cont_MADfin)>3,]$cont) # none
```

```
## numeric(0)
```

```
# Performance: Accuracy
# Level1
MAD <- mad(data_State_OutRm$acc1, na.rm = TRUE)
data_State_OutRm$acc1_MADfin <- (data_State_OutRm$acc1-median(data_State_OutRm$acc1, na.rm = TRUE))/MAD
sort(data_State_OutRm[abs(data_State_OutRm$acc1_MADfin)>3,]$acc1) # >= .75
```

```
##  [1] 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.50 0.50 0.50
## [16] 0.50 0.50 0.50 0.50 0.50 0.50 0.50 0.50 0.50 0.50 0.75 0.75 0.75 0.75 0.75
## [31] 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
## [46] 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
## [61] 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
```

```
data_State_OutRm[!is.na(data_State_OutRm$acc1)&!is.na(abs(data_State_OutRm$acc1_MADfin)),]$acc1 <- NA
# Level2
MAD <- mad(data_State_OutRm$acc2, na.rm = TRUE)
data_State_OutRm$acc2_MADfin <- (data_State_OutRm$acc2-median(data_State_OutRm$acc2, na.rm = TRUE))/MAD
sort(data_State_OutRm[abs(data_State_OutRm$acc2_MADfin)>3,]$acc2) # >= .75
```

```
##  [1] 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.50 0.50 0.50 0.50 0.50 0.50 0.50 0.50
## [16] 0.50 0.50 0.50 0.50 0.50 0.50 0.50 0.50 0.50 0.50 0.75 0.75 0.75 0.75 0.75
## [31] 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
## [46] 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
## [61] 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
## [76] 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
```

```
data_State_OutRm[!is.na(data_State_OutRm$acc2)&!is.na(abs(data_State_OutRm$acc2_MADfin)),]$acc2 <- NA
# Level3
MAD <- mad(data_State_OutRm$acc3, na.rm = TRUE)
data_State_OutRm$acc3_MADfin <- (data_State_OutRm$acc3-median(data_State_OutRm$acc3, na.rm = TRUE))/MAD
sort(data_State_OutRm[abs(data_State_OutRm$acc3_MADfin)>3,]$acc3) # >= .75
```

```
##  [1] 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.50 0.50 0.50 0.50 0.50 0.50
## [16] 0.50 0.50 0.50 0.50 0.50 0.50 0.50 0.50 0.50 0.50 0.50 0.50 0.50 0.50 0.50
## [31] 0.50 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
## [46] 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
## [61] 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
## [76] 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75
```

```
data_State_OutRm[!is.na(data_State_OutRm$acc3)&!is.na(abs(data_State_OutRm$acc3_MADfin)),]$acc3 <- NA
# Performance: Initial reaction time correct
# Level1
MAD <- mad(data_State_OutRm$initial_rt_correct1, na.rm = TRUE)
data_State_OutRm$initial_rt_correct1_MADfin <- (data_State_OutRm$initial_rt_correct1-median(data_State_(
sort(data_State_OutRm[abs(data_State_OutRm$initial_rt_correct1_MADfin)>3,]$initial_rt_correct1)
```

```
##  [1] 36829.28 36992.91 37385.25 38252.38 39140.33 42685.36 43708.22 46033.87
##  [9] 49462.49 49850.64 54614.35 61353.55
```

```
data_State_OutRm[!is.na(data_State_OutRm$initial_rt_correct1)&abs(data_State_OutRm$initial_rt_correct1_M
# Level2
MAD <- mad(data_State_OutRm$initial_rt_correct2, na.rm = TRUE)
data_State_OutRm$initial_rt_correct2_MADfin <- (data_State_OutRm$initial_rt_correct2-median(data_State_(
sort(data_State_OutRm[abs(data_State_OutRm$initial_rt_correct2_MADfin)>3,]$initial_rt_correct2)
```

```
## [1] 45279.65 45532.36 49200.47 49673.27 52089.15 52276.60 52417.93 63457.14
## [9] 83852.14
```

```r
data_State_OutRm[!is.na(data_State_OutRm$initial_rt_correct2)&abs(data_State_OutRm$initial_rt_correct2_M
# Level3
MAD <- mad(data_State_OutRm$initial_rt_correct3, na.rm = TRUE)
data_State_OutRm$initial_rt_correct3_MADfin <- (data_State_OutRm$initial_rt_correct3-median(data_State_O
sort(data_State_OutRm[abs(data_State_OutRm$initial_rt_correct3_MADfin)>3,]$initial_rt_correct3)
```

```
##  [1] 49295.12 49612.11 50545.59 50567.88 50959.66 52497.49 57641.59 58804.97
##  [9] 59272.15 61230.45 62032.84 63090.81 63783.15 66826.00 68576.95 70315.85
```

```r
data_State_OutRm[!is.na(data_State_OutRm$initial_rt_correct3)&abs(data_State_OutRm$initial_rt_correct3_M
###### 2. Check skeness, kurtosis
# Anger
data_State$ang_mean <- (data_State$ang1+data_State$ang2)/2
skew(as.numeric(data_State$ang_mean))
```

```
## [1] 1.305168
```

```r
kurtosi(as.numeric(data_State$ang_mean))
```

```
## [1] 0.900484
```

```r
# Anxiety
data_State$anx_mean <- (data_State$anx1+data_State$anx2)/2
skew(as.numeric(data_State$anx_mean))
```

```
## [1] 1.726626
```

```r
kurtosi(as.numeric(data_State$anx_mean)) # >1.96!
```

```
## [1] 2.845533
```

```r
# Boredom
data_State$bor_mean <- (data_State$bor1+data_State$bor2)/2
skew(as.numeric(data_State$bor_mean))
```

```
## [1] -0.2254316
```

```r
kurtosi(as.numeric(data_State$bor_mean))
```

```
## [1] -0.9575266
```

```r
# Enjoyment
data_State$enj_mean <- (data_State$enj1+data_State$enj2)/2
skew(as.numeric(data_State$enj_mean))
```

```
## [1] 0.196676
```

```r
kurtosi(as.numeric(data_State$enj_mean))
```

```
## [1] -0.716943
```

```r
# Control
skew(as.numeric(data_State$cont))
```

```
## [1] -0.520988
```

```r
kurtosi(as.numeric(data_State$cont))
```

```
## [1] -0.2361692
```

```r
# Value
data_State$val_mean <- (data_State$val1+data_State$val2)/2
skew(as.numeric(data_State$val_mean))
```

```
## [1] 0.1255104
```

```r
kurtosi(as.numeric(data_State$val_mean))
```

```
## [1] -0.5280841
```

```r
# Performance: Accuracy -> check normality with outliers
data_State$acc_mean <- (data_State$acc1+data_State$acc2+data_State$acc3)/3
skew(as.numeric(data_State$acc_mean))
```

```
## [1] -2.399463
```

```r
kurtosi(as.numeric(data_State$acc_mean)) # not normally distributed
```

```
## [1] 6.282835
```

```r
# Performance: initial_rt_correct
data_State_OutRm$initial_rt_correct_mean <- (data_State_OutRm$initial_rt_correct1+data_State_OutRm$initi
skew(as.numeric(data_State_OutRm$initial_rt_correct_mean))
```

```
## [1] 0.3822013
```

```r
kurtosi(as.numeric(data_State_OutRm$initial_rt_correct_mean))
```

```
## [1] -0.2460094
```

```r
# SAVE
data_State_nC_usable <- cbind(data_State[c(1:2,20:23,17,24)],data_State_OutRm[c(4,6,8)])
data_State_nC_OutRm <- data_State_OutRm[c(1:2,20,22,24,26,17,28,4,6,8)]
save(data_State_nC_usable,data_State_nC_OutRm, file = "data_State_nC_usable.Rda" )
```

Choice block

```r
load("data_State_wide_both.Rda")
data_State <- data_State_wide_C
# Raw data, excluding Pilot
data_State <- data_State[!data_State$uid<137098,] # Excluding the pilot
data_State <- data_State[!c(data_State$uid==137827|data_State$uid==137266|data_State$uid==137111|data_St
data_State <- data_State[!c(data_State$uid==137134|data_State$uid==137342|data_State$uid==137475|data_St
# New data_frame: all outliers removed:
data_State_OutRm <- data_State
###### 1. MAD procedure per construct
# Anger
data_State_OutRm$ang_mean <- (data_State_OutRm$ang1+data_State$ang2)/2
MAD <- mad(data_State_OutRm$ang_mean)
data_State_OutRm$ang_mean_MADfin <- (data_State_OutRm$ang_mean-median(data_State_OutRm$ang_mean))/MAD
sort(data_State_OutRm[abs(data_State_OutRm$ang_mean_MADfin)>3,]$ang_mean)
```

```
##  [1] 0.6539062 0.6585938 0.6656250 0.6890625 0.6923611 0.7109375 0.7373817
##  [8] 0.7429687 0.7523438 0.7812500 0.7914063 0.7945312 0.8177083 0.8449074
## [15] 0.8673858 0.8737310 0.9577546 0.9625000 0.9679688 0.9694836 0.9701389
## [22] 0.9710937 0.9711914 0.9718750 0.9803241 0.9820313
```

```
data_State_OutRm[!is.na(data_State_OutRm$ang_mean)&abs(data_State_OutRm$ang_mean_MADfin)>3,]$ang_mean <
# Anxiety
data_State_OutRm$anx_mean <- (data_State_OutRm$anx1+data_State_OutRm$anx2)/2
MAD <- mad(data_State_OutRm$anx_mean)
data_State_OutRm$anx_mean_MADfin <- (data_State_OutRm$anx_mean-median(data_State_OutRm$anx_mean))/MAD
sort(data_State_OutRm[abs(data_State_OutRm$anx_mean_MADfin)>3,]$anx_mean)
```

```
##  [1] 0.3902616 0.3982747 0.4021909 0.4062500 0.4236111 0.4305556 0.4332682
##  [8] 0.4500000 0.4539063 0.4548611 0.4937500 0.4959310 0.4984351 0.5000000
## [15] 0.5046875 0.5093750 0.5098380 0.5196759 0.5214120 0.5226563 0.5243056
## [22] 0.5307571 0.5335937 0.5374349 0.5399306 0.5484375 0.5488281 0.5789062
## [29] 0.5814938 0.6166667 0.6375000 0.6375000 0.6562500 0.6757812 0.7273437
## [36] 0.7585937 0.7703125 0.8645833 0.9539062 0.9664352 0.9885787
```

```
data_State_OutRm[!is.na(data_State_OutRm$anx_mean)&abs(data_State_OutRm$anx_mean_MADfin)>3,]$anx_mean <
# Boredom
data_State_OutRm$bor_mean <- (data_State_OutRm$bor1+data_State$bor2)/2
MAD <- mad(data_State_OutRm$bor_mean)
data_State_OutRm$bor_mean_MADfin <- (data_State_OutRm$bor_mean-median(data_State_OutRm$bor_mean))/MAD
sort(data_State_OutRm[abs(data_State_OutRm$bor_mean_MADfin)>3,]$bor_mean) # none
```

```
## numeric(0)
```

```
# Enjoyment
data_State_OutRm$enj_mean <- (data_State_OutRm$enj1+data_State_OutRm$enj2)/2
MAD <- mad(data_State_OutRm$enj_mean)
data_State_OutRm$enj_mean_MADfin <- (data_State_OutRm$enj_mean-median(data_State_OutRm$enj_mean))/MAD
data_State_OutRm[abs(data_State_OutRm$enj_mean_MADfin)>3,]$enj_mean # none
```

```
## numeric(0)
```

```
# Value
data_State_OutRm$val_mean <- (data_State_OutRm$val1+data_State_OutRm$val2)/2
MAD <- mad(data_State_OutRm$val_mean)
data_State_OutRm$val_mean_MADfin <- (data_State_OutRm$val_mean-median(data_State_OutRm$val_mean))/MAD
sort(data_State_OutRm[abs(data_State_OutRm$val_mean_MADfin)>3,]$val_mean) # nonde
```

```
## numeric(0)
```

```
# Control
MAD <- mad(data_State_OutRm$cont)
data_State_OutRm$cont_MADfin <- (data_State_OutRm$cont-median(data_State_OutRm$cont))/MAD
sort(data_State_OutRm[abs(data_State_OutRm$cont_MADfin)>3,]$cont) # none
```

```
## numeric(0)
```

```
# Performance: Accuracy
# Level1
MAD <- mad(data_State_OutRm$acc1, na.rm = TRUE)
data_State_OutRm$acc1_MADfin <- (data_State_OutRm$acc1-median(data_State_OutRm$acc1, na.rm = TRUE))/MAD
sort(data_State_OutRm[abs(data_State_OutRm$acc1_MADfin)>3,]$acc1)
```

```
##  [1] 0.2500000 0.2857143 0.4285714 0.5000000 0.5000000 0.5000000 0.5000000
##  [8] 0.5714286 0.5714286 0.6000000 0.6666667 0.6666667 0.6666667 0.6666667
## [15] 0.6666667 0.7500000 0.7500000 0.7500000 0.7500000 0.7500000 0.8000000
## [22] 0.8000000 0.8000000 0.8000000 0.8000000 0.8000000 0.8333333 0.8333333
## [29] 0.8333333 0.8571429 0.8571429 0.8571429 0.8750000 0.8750000 0.8750000
## [36] 0.8750000
```

```r
data_State_OutRm[!is.na(data_State_OutRm$acc1)&!is.na(abs(data_State_OutRm$acc1_MADfin)),]$acc1 <- NA
# Level2
MAD <- mad(data_State_OutRm$acc2, na.rm = TRUE)
data_State_OutRm$acc2_MADfin <- (data_State_OutRm$acc2-median(data_State_OutRm$acc2, na.rm = TRUE))/MAD
sort(data_State_OutRm[abs(data_State_OutRm$acc2_MADfin)>3,]$acc2)
```

```
##  [1] 0.2000000 0.2500000 0.2500000 0.2500000 0.3333333 0.4000000 0.4000000
##  [8] 0.5000000 0.5000000 0.5000000 0.5000000 0.5000000 0.5000000 0.5000000
## [15] 0.5000000 0.5000000 0.6000000 0.6250000 0.6666667 0.7142857 0.7142857
## [22] 0.7500000 0.7500000 0.7500000 0.7500000 0.7500000 0.7500000 0.7500000
## [29] 0.7500000 0.7500000 0.7500000 0.7500000 0.7500000 0.7500000 0.7500000
## [36] 0.7500000 0.7500000 0.7500000 0.7500000 0.8000000 0.8000000 0.8000000
## [43] 0.8000000 0.8000000 0.8000000 0.8000000 0.8000000 0.8333333 0.8333333
## [50] 0.8333333 0.8333333 0.8333333 0.8333333 0.8571429 0.8750000
```

```r
data_State_OutRm[!is.na(data_State_OutRm$acc2)&!is.na(abs(data_State_OutRm$acc2_MADfin)),]$acc2 <- NA
# Level3
MAD <- mad(data_State_OutRm$acc3, na.rm = TRUE)
data_State_OutRm$acc3_MADfin <- (data_State_OutRm$acc3-median(data_State_OutRm$acc3, na.rm = TRUE))/MAD
sort(data_State_OutRm[abs(data_State_OutRm$acc3_MADfin)>3,]$acc3)
```

```
##  [1] 0.2500000 0.2500000 0.3333333 0.3333333 0.5000000 0.5000000 0.5000000
##  [8] 0.5000000 0.5000000 0.5000000 0.5000000 0.5000000 0.5000000 0.6000000
## [15] 0.6000000 0.6000000 0.6250000 0.6666667 0.6666667 0.6666667 0.6666667
## [22] 0.6666667 0.7142857 0.7142857 0.7142857 0.7142857 0.7500000 0.7500000
## [29] 0.7500000 0.7500000 0.8000000 0.8000000 0.8000000 0.8000000 0.8000000
## [36] 0.8333333 0.8333333 0.8333333 0.8333333 0.8333333 0.8333333 0.8571429
## [43] 0.8571429 0.8571429 0.8571429 0.8750000 0.8750000 0.8750000 0.8750000
## [50] 0.8750000 0.8750000 0.8750000 0.8750000 0.8750000 0.8750000 0.8750000
## [57] 0.8750000 0.8750000 0.8750000 0.8750000 0.8750000 0.8750000 0.8750000
## [64] 0.8750000 0.8750000 0.8750000 0.8750000 0.8750000
```

```r
data_State_OutRm[!is.na(data_State_OutRm$acc3)&!is.na(abs(data_State_OutRm$acc3_MADfin)),]$acc3 <- NA
# Performance: Initial reaction time correct
# Level1
MAD <- mad(data_State_OutRm$initial_rt_correct1, na.rm = TRUE)
data_State_OutRm$initial_rt_correct1_MADfin <- (data_State_OutRm$initial_rt_correct1-median(data_State_
sort(data_State_OutRm[abs(data_State_OutRm$initial_rt_correct1_MADfin)>3,]$initial_rt_correct1)
```

```
## [1] 36981.89 39622.41 44306.72 53169.11
```

```r
data_State_OutRm[!is.na(data_State_OutRm$initial_rt_correct1)&abs(data_State_OutRm$initial_rt_correct1_
# Level2
MAD <- mad(data_State_OutRm$initial_rt_correct2, na.rm = TRUE)
data_State_OutRm$initial_rt_correct2_MADfin <- (data_State_OutRm$initial_rt_correct2-median(data_State_
sort(data_State_OutRm[abs(data_State_OutRm$initial_rt_correct2_MADfin)>3,]$initial_rt_correct2)
```

```
##  [1] 33139.95 33594.82 33670.68 34564.61 34971.68 35235.53 35873.46 40739.66
##  [9] 43056.95 45108.70 48878.85 53411.14 60509.47 67211.30
```

```r
data_State_OutRm[!is.na(data_State_OutRm$initial_rt_correct2)&abs(data_State_OutRm$initial_rt_correct2_
# Level3
MAD <- mad(data_State_OutRm$initial_rt_correct3, na.rm = TRUE)
data_State_OutRm$initial_rt_correct3_MADfin <- (data_State_OutRm$initial_rt_correct3-median(data_State_
sort(data_State_OutRm[abs(data_State_OutRm$initial_rt_correct3_MADfin)>3,]$initial_rt_correct3)
```

```
##  [1] 52232.07 52870.22 53602.57 58164.86 59750.60 63352.00 63552.52 67905.51
```

```
##  [9] 69563.63 70356.45 84242.25
```

```r
data_State_OutRm[!is.na(data_State_OutRm$initial_rt_correct3)&abs(data_State_OutRm$initial_rt_correct3_
###### 2. Check skeness, kurtosis
# Anger
data_State$ang_mean <- (data_State$ang1+data_State$ang2)/2
skew(as.numeric(data_State$ang_mean))
```

```
## [1] 1.444163
```

```r
kurtosi(as.numeric(data_State$ang_mean))
```

```
## [1] 1.503188
```

```r
# Anxiety
data_State$anx_mean <- (data_State$anx1+data_State$anx2)/2
skew(as.numeric(data_State$anx_mean))
```

```
## [1] 1.899044
```

```r
kurtosi(as.numeric(data_State$anx_mean)) # >1.96!
```

```
## [1] 3.791565
```

```r
# Boredom
data_State$bor_mean <- (data_State$bor1+data_State$bor2)/2
skew(as.numeric(data_State$bor_mean))
```

```
## [1] -0.1286591
```

```r
kurtosi(as.numeric(data_State$bor_mean))
```

```
## [1] -1.010007
```

```r
# Enjoyment
data_State$enj_mean <- (data_State$enj1+data_State$enj2)/2
skew(as.numeric(data_State$enj_mean))
```

```
## [1] 0.1301605
```

```r
kurtosi(as.numeric(data_State$enj_mean))
```

```
## [1] -0.6877128
```

```r
# Control
skew(as.numeric(data_State$cont))
```

```
## [1] -0.5974813
```

```r
kurtosi(as.numeric(data_State$cont))
```

```
## [1] -0.1587232
```

```r
# Value
data_State$val_mean <- (data_State$val1+data_State$val2)/2
skew(as.numeric(data_State$val_mean))
```

```
## [1] 0.1474533
```

```r
kurtosi(as.numeric(data_State$val_mean))
```

```
## [1] -0.5711582
```

```r
# Performance: Accuracy -> check normality with outliers
data_State$acc_mean <- (data_State$acc1+data_State$acc2+data_State$acc3)/3
skew(as.numeric(data_State$acc_mean))
```

```
## [1] -1.872849
```

```r
kurtosi(as.numeric(data_State$acc_mean)) # not normally distributed
```

```
## [1] 3.072964
```

```r
# Performance: initial_rt_correct
data_State_OutRm$initial_rt_correct_mean <- (data_State_OutRm$initial_rt_correct1+data_State_OutRm$initi
skew(as.numeric(data_State_OutRm$initial_rt_correct_mean))
```

```
## [1] -0.561006
```

```r
kurtosi(as.numeric(data_State_OutRm$initial_rt_correct_mean))
```

```
## [1] 0.6740194
```

```r
# SAVE
data_State_C_usable <- cbind(data_State[c(1:2,20:23,17,24)],data_State_OutRm[c(4,6,8)])
data_State_C_OutRm <- data_State_OutRm[c(1:2,20,22,24,26,17,28,4,6,8)]
save(data_State_C_usable,data_State_C_OutRm, file = "data_State_C_usable.Rda" )
```

Data Analysis

Trait Level

Factor analysis - Trait

```r
load("data_Trait_usable.Rda")
data_Factors <- data_Trait_usable[c(4:36,38:46)]
# original model: one factor per questionnaire
model0 <- ' Anger =~ AEQ_A1 + AEQ_A2 + AEQ_A3 + AEQ_A4
              Anxiety =~ AMAS1 + AMAS2 + AMAS3 + AMAS4 + AMAS5 + AMAS6 + AMAS7 + AMAS8 + AMAS9
              Boredom =~ AEQ_B1 + AEQ_B2 + AEQ_B3 + AEQ_B4 + AEQ_B5
              Enjoyment =~ AEQ_E1 + AEQ_E2 + AEQ_E4
              Self_Concept =~ SelfC1 + SelfC2 + SelfC3 + SelfC4 + SelfC5
              Self_Efficacy =~ SelfE1 + SelfE2 + SelfE3 + SelfE4 + SelfE5 + SelfE6 + SelfE7 + SelfE8
              IntrinsicVal =~ Int1 + Int2  + Int3 + Int4
              InstrumentalVal =~ Val1 + Val2 + Val3 + Val4'
fit_o <- cfa(model0,std.lv = TRUE, data = data_Factors, estimator = "MLM")
summary(fit_o, fit.measures = TRUE, standardized = TRUE)
```

```
## lavaan 0.6-8 ended normally after 40 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                       112
##
##   Number of observations                           347
##
## Model Test User Model:
```

```
##                                              Standard      Robust
##   Test Statistic                             1453.654    1259.422
##   Degrees of freedom                              791         791
##   P-value (Chi-square)                          0.000       0.000
##   Scaling correction factor                                 1.154
##        Satorra-Bentler correction
##
## Model Test Baseline Model:
##
##   Test statistic                             7074.558    5684.651
##   Degrees of freedom                              861         861
##   P-value                                       0.000       0.000
##   Scaling correction factor                                 1.245
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                   0.893       0.903
##   Tucker-Lewis Index (TLI)                      0.884       0.894
##
##   Robust Comparative Fit Index (CFI)                        0.910
##   Robust Tucker-Lewis Index (TLI)                           0.902
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)            -17380.000  -17380.000
##   Loglikelihood unrestricted model (H1)    -16653.173  -16653.173
##
##   Akaike (AIC)                              34984.001   34984.001
##   Bayesian (BIC)                            35415.125   35415.125
##   Sample-size adjusted Bayesian (BIC)       35059.827   35059.827
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                         0.049       0.041
##   90 Percent confidence interval - lower        0.045       0.037
##   90 Percent confidence interval - upper        0.053       0.045
##   P-value RMSEA <= 0.05                         0.635       1.000
##
##   Robust RMSEA                                              0.044
##   90 Percent confidence interval - lower                    0.040
##   90 Percent confidence interval - upper                    0.049
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                          0.056       0.056
##
## 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
```

```
##   Anger =~
##     AEQ_A1             0.797    0.071   11.171    0.000    0.797    0.665
##     AEQ_A2             0.762    0.057   13.261    0.000    0.762    0.612
##     AEQ_A3             1.027    0.065   15.723    0.000    1.027    0.857
##     AEQ_A4             0.853    0.071   12.070    0.000    0.853    0.745
##   Anxiety =~
##     AMAS1              0.371    0.055    6.753    0.000    0.371    0.522
##     AMAS2              0.849    0.053   16.035    0.000    0.849    0.651
##     AMAS3              0.799    0.060   13.223    0.000    0.799    0.654
##     AMAS4              0.923    0.053   17.374    0.000    0.923    0.741
##     AMAS5              0.622    0.069    9.076    0.000    0.622    0.595
##     AMAS6              0.518    0.065    8.002    0.000    0.518    0.687
##     AMAS7              0.642    0.075    8.559    0.000    0.642    0.674
##     AMAS8              0.621    0.075    8.331    0.000    0.621    0.543
##     AMAS9              0.415    0.061    6.798    0.000    0.415    0.587
##   Boredom =~
##     AEQ_B1             0.780    0.057   13.729    0.000    0.780    0.635
##     AEQ_B2             0.742    0.063   11.805    0.000    0.742    0.716
##     AEQ_B3             0.788    0.072   10.878    0.000    0.788    0.633
##     AEQ_B4             0.949    0.057   16.500    0.000    0.949    0.741
##     AEQ_B5             0.947    0.054   17.520    0.000    0.947    0.739
##   Enjoyment =~
##     AEQ_E1             0.879    0.052   16.886    0.000    0.879    0.772
##     AEQ_E2             0.677    0.059   11.458    0.000    0.677    0.619
##     AEQ_E4             0.675    0.063   10.738    0.000    0.675    0.647
##   Self_Concept =~
##     SelfC1             0.729    0.039   18.477    0.000    0.729    0.810
##     SelfC2             0.555    0.038   14.585    0.000    0.555    0.707
##     SelfC3             0.714    0.036   19.726    0.000    0.714    0.794
##     SelfC4             0.787    0.038   20.894    0.000    0.787    0.761
##     SelfC5             0.650    0.036   18.182    0.000    0.650    0.757
##   Self_Efficacy =~
##     SelfE1             0.418    0.054    7.812    0.000    0.418    0.438
##     SelfE2             0.158    0.058    2.710    0.007    0.158    0.214
##     SelfE3             0.325    0.056    5.783    0.000    0.325    0.346
##     SelfE4             0.207    0.053    3.919    0.000    0.207    0.254
##     SelfE5             0.598    0.048   12.547    0.000    0.598    0.701
##     SelfE6             0.379    0.054    7.027    0.000    0.379    0.418
##     SelfE7             0.684    0.045   15.208    0.000    0.684    0.719
##     SelfE8             0.386    0.048    8.060    0.000    0.386    0.474
##   IntrinsicVal =~
##     Int1               0.484    0.042   11.425    0.000    0.484    0.654
##     Int2               0.644    0.032   19.827    0.000    0.644    0.821
##     Int3               0.790    0.031   25.193    0.000    0.790    0.903
##     Int4               0.624    0.040   15.682    0.000    0.624    0.743
##   InstrumentalVal =~
##     Val1               0.715    0.041   17.596    0.000    0.715    0.789
##     Val2               0.587    0.048   12.191    0.000    0.587    0.727
##     Val3               0.741    0.038   19.438    0.000    0.741    0.847
##     Val4               0.629    0.044   14.140    0.000    0.629    0.753
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   Anger ~~
```

```
##     Anxiety          0.482    0.065    7.417    0.000    0.482    0.482
##     Boredom          0.743    0.036   20.831    0.000    0.743    0.743
##     Enjoyment       -0.558    0.054  -10.311    0.000   -0.558   -0.558
##     Self_Concept    -0.443    0.060   -7.352    0.000   -0.443   -0.443
##     Self_Efficacy   -0.466    0.058   -8.007    0.000   -0.466   -0.466
##     IntrinsicVal    -0.524    0.054   -9.635    0.000   -0.524   -0.524
##     InstrumentalVl  -0.294    0.069   -4.243    0.000   -0.294   -0.294
##   Anxiety ~~
##     Boredom          0.275    0.069    4.006    0.000    0.275    0.275
##     Enjoyment       -0.389    0.053   -7.389    0.000   -0.389   -0.389
##     Self_Concept    -0.716    0.038  -18.973    0.000   -0.716   -0.716
##     Self_Efficacy   -0.656    0.060  -11.013    0.000   -0.656   -0.656
##     IntrinsicVal    -0.354    0.055   -6.458    0.000   -0.354   -0.354
##     InstrumentalVl  -0.213    0.070   -3.050    0.002   -0.213   -0.213
##   Boredom ~~
##     Enjoyment       -0.512    0.053   -9.641    0.000   -0.512   -0.512
##     Self_Concept    -0.205    0.067   -3.076    0.002   -0.205   -0.205
##     Self_Efficacy   -0.279    0.068   -4.093    0.000   -0.279   -0.279
##     IntrinsicVal    -0.501    0.055   -9.059    0.000   -0.501   -0.501
##     InstrumentalVl  -0.224    0.071   -3.162    0.002   -0.224   -0.224
##   Enjoyment ~~
##     Self_Concept     0.652    0.044   14.734    0.000    0.652    0.652
##     Self_Efficacy    0.361    0.068    5.343    0.000    0.361    0.361
##     IntrinsicVal     0.813    0.041   19.928    0.000    0.813    0.813
##     InstrumentalVl   0.309    0.074    4.200    0.000    0.309    0.309
##   Self_Concept ~~
##     Self_Efficacy    0.597    0.061    9.809    0.000    0.597    0.597
##     IntrinsicVal     0.599    0.044   13.619    0.000    0.599    0.599
##     InstrumentalVl   0.356    0.063    5.657    0.000    0.356    0.356
##   Self_Efficacy ~~
##     IntrinsicVal     0.308    0.075    4.124    0.000    0.308    0.308
##     InstrumentalVl   0.171    0.070    2.438    0.015    0.171    0.171
##   IntrinsicVal ~~
##     InstrumentalVl   0.398    0.061    6.536    0.000    0.398    0.398
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .AEQ_A1           0.803    0.097    8.303    0.000    0.803    0.558
##    .AEQ_A2           0.968    0.079   12.251    0.000    0.968    0.625
##    .AEQ_A3           0.382    0.061    6.310    0.000    0.382    0.266
##    .AEQ_A4           0.583    0.094    6.237    0.000    0.583    0.445
##    .AMAS1            0.367    0.053    6.934    0.000    0.367    0.728
##    .AMAS2            0.980    0.082   12.009    0.000    0.980    0.576
##    .AMAS3            0.853    0.086    9.906    0.000    0.853    0.572
##    .AMAS4            0.699    0.070    9.967    0.000    0.699    0.451
##    .AMAS5            0.705    0.067   10.504    0.000    0.705    0.645
##    .AMAS6            0.299    0.047    6.303    0.000    0.299    0.528
##    .AMAS7            0.495    0.059    8.318    0.000    0.495    0.545
##    .AMAS8            0.921    0.083   11.067    0.000    0.921    0.705
##    .AMAS9            0.328    0.034    9.553    0.000    0.328    0.656
##    .AEQ_B1           0.903    0.075   11.972    0.000    0.903    0.597
##    .AEQ_B2           0.522    0.054    9.656    0.000    0.522    0.487
##    .AEQ_B3           0.928    0.099    9.416    0.000    0.928    0.599
##    .AEQ_B4           0.738    0.075    9.881    0.000    0.738    0.451
```

```
##    .AEQ_B5           0.744    0.073   10.194    0.000    0.744    0.453
##    .AEQ_E1           0.523    0.071    7.371    0.000    0.523    0.404
##    .AEQ_E2           0.738    0.083    8.873    0.000    0.738    0.617
##    .AEQ_E4           0.631    0.082    7.719    0.000    0.631    0.581
##    .SelfC1           0.278    0.037    7.464    0.000    0.278    0.343
##    .SelfC2           0.309    0.029   10.784    0.000    0.309    0.500
##    .SelfC3           0.299    0.031    9.701    0.000    0.299    0.370
##    .SelfC4           0.449    0.041   10.850    0.000    0.449    0.421
##    .SelfC5           0.314    0.030   10.561    0.000    0.314    0.427
##    .SelfE1           0.736    0.048   15.311    0.000    0.736    0.808
##    .SelfE2           0.517    0.037   13.845    0.000    0.517    0.954
##    .SelfE3           0.779    0.048   16.348    0.000    0.779    0.880
##    .SelfE4           0.621    0.042   14.863    0.000    0.621    0.935
##    .SelfE5           0.370    0.036   10.148    0.000    0.370    0.509
##    .SelfE6           0.678    0.049   13.979    0.000    0.678    0.825
##    .SelfE7           0.437    0.047    9.207    0.000    0.437    0.483
##    .SelfE8           0.514    0.042   12.363    0.000    0.514    0.776
##    .Int1             0.314    0.029   10.893    0.000    0.314    0.573
##    .Int2             0.201    0.022    8.963    0.000    0.201    0.326
##    .Int3             0.142    0.022    6.408    0.000    0.142    0.185
##    .Int4             0.317    0.029   10.742    0.000    0.317    0.448
##    .Val1             0.310    0.037    8.407    0.000    0.310    0.378
##    .Val2             0.307    0.041    7.517    0.000    0.307    0.472
##    .Val3             0.217    0.042    5.215    0.000    0.217    0.283
##    .Val4             0.302    0.042    7.135    0.000    0.302    0.433
##     Anger            1.000                               1.000    1.000
##     Anxiety          1.000                               1.000    1.000
##     Boredom          1.000                               1.000    1.000
##     Enjoyment        1.000                               1.000    1.000
##     Self_Concept     1.000                               1.000    1.000
##     Self_Efficacy    1.000                               1.000    1.000
##     IntrinsicVal     1.000                               1.000    1.000
##     InstrumentalVl   1.000                               1.000    1.000
```

```
parameterEstimates(fit_o, standardized = TRUE)
```

```
##                lhs op           rhs   est    se      z pvalue ci.lower
## 1            Anger =~        AEQ_A1 0.797 0.071 11.171  0.000    0.657
## 2            Anger =~        AEQ_A2 0.762 0.057 13.261  0.000    0.649
## 3            Anger =~        AEQ_A3 1.027 0.065 15.723  0.000    0.899
## 4            Anger =~        AEQ_A4 0.853 0.071 12.070  0.000    0.714
## 5          Anxiety =~         AMAS1 0.371 0.055  6.753  0.000    0.263
## 6          Anxiety =~         AMAS2 0.849 0.053 16.035  0.000    0.745
## 7          Anxiety =~         AMAS3 0.799 0.060 13.223  0.000    0.680
## 8          Anxiety =~         AMAS4 0.923 0.053 17.374  0.000    0.819
## 9          Anxiety =~         AMAS5 0.622 0.069  9.076  0.000    0.488
## 10         Anxiety =~         AMAS6 0.518 0.065  8.002  0.000    0.391
## 11         Anxiety =~         AMAS7 0.642 0.075  8.559  0.000    0.495
## 12         Anxiety =~         AMAS8 0.621 0.075  8.331  0.000    0.475
## 13         Anxiety =~         AMAS9 0.415 0.061  6.798  0.000    0.295
## 14         Boredom =~        AEQ_B1 0.780 0.057 13.729  0.000    0.669
## 15         Boredom =~        AEQ_B2 0.742 0.063 11.805  0.000    0.619
## 16         Boredom =~        AEQ_B3 0.788 0.072 10.878  0.000    0.646
## 17         Boredom =~        AEQ_B4 0.949 0.057 16.500  0.000    0.836
## 18         Boredom =~        AEQ_B5 0.947 0.054 17.520  0.000    0.841
```

```
## 19        Enjoyment =~          AEQ_E1  0.879 0.052  16.886  0.000     0.777
## 20        Enjoyment =~          AEQ_E2  0.677 0.059  11.458  0.000     0.562
## 21        Enjoyment =~          AEQ_E4  0.675 0.063  10.738  0.000     0.552
## 22     Self_Concept =~          SelfC1  0.729 0.039  18.477  0.000     0.652
## 23     Self_Concept =~          SelfC2  0.555 0.038  14.585  0.000     0.480
## 24     Self_Concept =~          SelfC3  0.714 0.036  19.726  0.000     0.643
## 25     Self_Concept =~          SelfC4  0.787 0.038  20.894  0.000     0.713
## 26     Self_Concept =~          SelfC5  0.650 0.036  18.182  0.000     0.580
## 27    Self_Efficacy =~          SelfE1  0.418 0.054   7.812  0.000     0.313
## 28    Self_Efficacy =~          SelfE2  0.158 0.058   2.710  0.007     0.044
## 29    Self_Efficacy =~          SelfE3  0.325 0.056   5.783  0.000     0.215
## 30    Self_Efficacy =~          SelfE4  0.207 0.053   3.919  0.000     0.103
## 31    Self_Efficacy =~          SelfE5  0.598 0.048  12.547  0.000     0.504
## 32    Self_Efficacy =~          SelfE6  0.379 0.054   7.027  0.000     0.273
## 33    Self_Efficacy =~          SelfE7  0.684 0.045  15.208  0.000     0.596
## 34    Self_Efficacy =~          SelfE8  0.386 0.048   8.060  0.000     0.292
## 35     IntrinsicVal =~            Int1  0.484 0.042  11.425  0.000     0.401
## 36     IntrinsicVal =~            Int2  0.644 0.032  19.827  0.000     0.580
## 37     IntrinsicVal =~            Int3  0.790 0.031  25.193  0.000     0.729
## 38     IntrinsicVal =~            Int4  0.624 0.040  15.682  0.000     0.546
## 39  InstrumentalVal =~            Val1  0.715 0.041  17.596  0.000     0.635
## 40  InstrumentalVal =~            Val2  0.587 0.048  12.191  0.000     0.492
## 41  InstrumentalVal =~            Val3  0.741 0.038  19.438  0.000     0.667
## 42  InstrumentalVal =~            Val4  0.629 0.044  14.140  0.000     0.541
## 43           AEQ_A1 ~~          AEQ_A1  0.803 0.097   8.303  0.000     0.613
## 44           AEQ_A2 ~~          AEQ_A2  0.968 0.079  12.251  0.000     0.813
## 45           AEQ_A3 ~~          AEQ_A3  0.382 0.061   6.310  0.000     0.264
## 46           AEQ_A4 ~~          AEQ_A4  0.583 0.094   6.237  0.000     0.400
## 47            AMAS1 ~~           AMAS1  0.367 0.053   6.934  0.000     0.263
## 48            AMAS2 ~~           AMAS2  0.980 0.082  12.009  0.000     0.820
## 49            AMAS3 ~~           AMAS3  0.853 0.086   9.906  0.000     0.684
## 50            AMAS4 ~~           AMAS4  0.699 0.070   9.967  0.000     0.562
## 51            AMAS5 ~~           AMAS5  0.705 0.067  10.504  0.000     0.573
## 52            AMAS6 ~~           AMAS6  0.299 0.047   6.303  0.000     0.206
## 53            AMAS7 ~~           AMAS7  0.495 0.059   8.318  0.000     0.378
## 54            AMAS8 ~~           AMAS8  0.921 0.083  11.067  0.000     0.758
## 55            AMAS9 ~~           AMAS9  0.328 0.034   9.553  0.000     0.261
## 56           AEQ_B1 ~~          AEQ_B1  0.903 0.075  11.972  0.000     0.755
## 57           AEQ_B2 ~~          AEQ_B2  0.522 0.054   9.656  0.000     0.416
## 58           AEQ_B3 ~~          AEQ_B3  0.928 0.099   9.416  0.000     0.735
## 59           AEQ_B4 ~~          AEQ_B4  0.738 0.075   9.881  0.000     0.592
## 60           AEQ_B5 ~~          AEQ_B5  0.744 0.073  10.194  0.000     0.601
## 61           AEQ_E1 ~~          AEQ_E1  0.523 0.071   7.371  0.000     0.384
## 62           AEQ_E2 ~~          AEQ_E2  0.738 0.083   8.873  0.000     0.575
## 63           AEQ_E4 ~~          AEQ_E4  0.631 0.082   7.719  0.000     0.471
## 64           SelfC1 ~~          SelfC1  0.278 0.037   7.464  0.000     0.205
## 65           SelfC2 ~~          SelfC2  0.309 0.029  10.784  0.000     0.252
## 66           SelfC3 ~~          SelfC3  0.299 0.031   9.701  0.000     0.238
## 67           SelfC4 ~~          SelfC4  0.449 0.041  10.850  0.000     0.368
## 68           SelfC5 ~~          SelfC5  0.314 0.030  10.561  0.000     0.256
## 69           SelfE1 ~~          SelfE1  0.736 0.048  15.311  0.000     0.641
## 70           SelfE2 ~~          SelfE2  0.517 0.037  13.845  0.000     0.443
## 71           SelfE3 ~~          SelfE3  0.779 0.048  16.348  0.000     0.685
## 72           SelfE4 ~~          SelfE4  0.621 0.042  14.863  0.000     0.539
```

```
## 73           SelfE5 ~~           SelfE5  0.370 0.036  10.148  0.000     0.299
## 74           SelfE6 ~~           SelfE6  0.678 0.049  13.979  0.000     0.583
## 75           SelfE7 ~~           SelfE7  0.437 0.047   9.207  0.000     0.344
## 76           SelfE8 ~~           SelfE8  0.514 0.042  12.363  0.000     0.433
## 77             Int1 ~~             Int1  0.314 0.029  10.893  0.000     0.257
## 78             Int2 ~~             Int2  0.201 0.022   8.963  0.000     0.157
## 79             Int3 ~~             Int3  0.142 0.022   6.408  0.000     0.099
## 80             Int4 ~~             Int4  0.317 0.029  10.742  0.000     0.259
## 81             Val1 ~~             Val1  0.310 0.037   8.407  0.000     0.238
## 82             Val2 ~~             Val2  0.307 0.041   7.517  0.000     0.227
## 83             Val3 ~~             Val3  0.217 0.042   5.215  0.000     0.135
## 84             Val4 ~~             Val4  0.302 0.042   7.135  0.000     0.219
## 85            Anger ~~            Anger  1.000 0.000      NA     NA     1.000
## 86          Anxiety ~~          Anxiety  1.000 0.000      NA     NA     1.000
## 87          Boredom ~~          Boredom  1.000 0.000      NA     NA     1.000
## 88        Enjoyment ~~        Enjoyment  1.000 0.000      NA     NA     1.000
## 89     Self_Concept ~~     Self_Concept  1.000 0.000      NA     NA     1.000
## 90    Self_Efficacy ~~    Self_Efficacy  1.000 0.000      NA     NA     1.000
## 91     IntrinsicVal ~~     IntrinsicVal  1.000 0.000      NA     NA     1.000
## 92  InstrumentalVal ~~ InstrumentalVal  1.000 0.000      NA     NA     1.000
## 93            Anger ~~          Anxiety  0.482 0.065   7.417  0.000     0.354
## 94            Anger ~~          Boredom  0.743 0.036  20.831  0.000     0.673
## 95            Anger ~~        Enjoyment -0.558 0.054 -10.311  0.000    -0.664
## 96            Anger ~~     Self_Concept -0.443 0.060  -7.352  0.000    -0.561
## 97            Anger ~~    Self_Efficacy -0.466 0.058  -8.007  0.000    -0.580
## 98            Anger ~~     IntrinsicVal -0.524 0.054  -9.635  0.000    -0.630
## 99            Anger ~~ InstrumentalVal -0.294 0.069  -4.243  0.000    -0.430
## 100         Anxiety ~~          Boredom  0.275 0.069   4.006  0.000     0.140
## 101         Anxiety ~~        Enjoyment -0.389 0.053  -7.389  0.000    -0.493
## 102         Anxiety ~~     Self_Concept -0.716 0.038 -18.973  0.000    -0.790
## 103         Anxiety ~~    Self_Efficacy -0.656 0.060 -11.013  0.000    -0.772
## 104         Anxiety ~~     IntrinsicVal -0.354 0.055  -6.458  0.000    -0.462
## 105         Anxiety ~~ InstrumentalVal -0.213 0.070  -3.050  0.002    -0.350
## 106         Boredom ~~        Enjoyment -0.512 0.053  -9.641  0.000    -0.616
## 107         Boredom ~~     Self_Concept -0.205 0.067  -3.076  0.002    -0.336
## 108         Boredom ~~    Self_Efficacy -0.279 0.068  -4.093  0.000    -0.413
## 109         Boredom ~~     IntrinsicVal -0.501 0.055  -9.059  0.000    -0.609
## 110         Boredom ~~ InstrumentalVal -0.224 0.071  -3.162  0.002    -0.363
## 111       Enjoyment ~~     Self_Concept  0.652 0.044  14.734  0.000     0.565
## 112       Enjoyment ~~    Self_Efficacy  0.361 0.068   5.343  0.000     0.229
## 113       Enjoyment ~~     IntrinsicVal  0.813 0.041  19.928  0.000     0.733
## 114       Enjoyment ~~ InstrumentalVal  0.309 0.074   4.200  0.000     0.165
## 115    Self_Concept ~~    Self_Efficacy  0.597 0.061   9.809  0.000     0.478
## 116    Self_Concept ~~     IntrinsicVal  0.599 0.044  13.619  0.000     0.513
## 117    Self_Concept ~~ InstrumentalVal  0.356 0.063   5.657  0.000     0.233
## 118   Self_Efficacy ~~     IntrinsicVal  0.308 0.075   4.124  0.000     0.162
## 119   Self_Efficacy ~~ InstrumentalVal  0.171 0.070   2.438  0.015     0.034
## 120    IntrinsicVal ~~ InstrumentalVal  0.398 0.061   6.536  0.000     0.279
##     ci.upper std.lv std.all std.nox
## 1      0.937  0.797   0.665   0.665
## 2      0.874  0.762   0.612   0.612
## 3      1.155  1.027   0.857   0.857
## 4      0.991  0.853   0.745   0.745
## 5      0.478  0.371   0.522   0.522
```

```
## 6       0.953  0.849   0.651   0.651
## 7       0.917  0.799   0.654   0.654
## 8       1.027  0.923   0.741   0.741
## 9       0.757  0.622   0.595   0.595
## 10      0.644  0.518   0.687   0.687
## 11      0.789  0.642   0.674   0.674
## 12      0.767  0.621   0.543   0.543
## 13      0.535  0.415   0.587   0.587
## 14      0.891  0.780   0.635   0.635
## 15      0.865  0.742   0.716   0.716
## 16      0.930  0.788   0.633   0.633
## 17      1.061  0.949   0.741   0.741
## 18      1.053  0.947   0.739   0.739
## 19      0.981  0.879   0.772   0.772
## 20      0.793  0.677   0.619   0.619
## 21      0.798  0.675   0.647   0.647
## 22      0.807  0.729   0.810   0.810
## 23      0.630  0.555   0.707   0.707
## 24      0.785  0.714   0.794   0.794
## 25      0.861  0.787   0.761   0.761
## 26      0.720  0.650   0.757   0.757
## 27      0.523  0.418   0.438   0.438
## 28      0.272  0.158   0.214   0.214
## 29      0.435  0.325   0.346   0.346
## 30      0.311  0.207   0.254   0.254
## 31      0.691  0.598   0.701   0.701
## 32      0.485  0.379   0.418   0.418
## 33      0.772  0.684   0.719   0.719
## 34      0.480  0.386   0.474   0.474
## 35      0.567  0.484   0.654   0.654
## 36      0.707  0.644   0.821   0.821
## 37      0.852  0.790   0.903   0.903
## 38      0.702  0.624   0.743   0.743
## 39      0.794  0.715   0.789   0.789
## 40      0.681  0.587   0.727   0.727
## 41      0.816  0.741   0.847   0.847
## 42      0.716  0.629   0.753   0.753
## 43      0.992  0.803   0.558   0.558
## 44      1.123  0.968   0.625   0.625
## 45      0.501  0.382   0.266   0.266
## 46      0.767  0.583   0.445   0.445
## 47      0.471  0.367   0.728   0.728
## 48      1.140  0.980   0.576   0.576
## 49      1.021  0.853   0.572   0.572
## 50      0.837  0.699   0.451   0.451
## 51      0.836  0.705   0.645   0.645
## 52      0.392  0.299   0.528   0.528
## 53      0.611  0.495   0.545   0.545
## 54      1.084  0.921   0.705   0.705
## 55      0.396  0.328   0.656   0.656
## 56      1.050  0.903   0.597   0.597
## 57      0.628  0.522   0.487   0.487
## 58      1.121  0.928   0.599   0.599
## 59      0.885  0.738   0.451   0.451
```

```
## 60     0.887  0.744   0.453   0.453
## 61     0.662  0.523   0.404   0.404
## 62     0.901  0.738   0.617   0.617
## 63     0.791  0.631   0.581   0.581
## 64     0.351  0.278   0.343   0.343
## 65     0.365  0.309   0.500   0.500
## 66     0.359  0.299   0.370   0.370
## 67     0.531  0.449   0.421   0.421
## 68     0.373  0.314   0.427   0.427
## 69     0.830  0.736   0.808   0.808
## 70     0.590  0.517   0.954   0.954
## 71     0.872  0.779   0.880   0.880
## 72     0.703  0.621   0.935   0.935
## 73     0.442  0.370   0.509   0.509
## 74     0.773  0.678   0.825   0.825
## 75     0.530  0.437   0.483   0.483
## 76     0.596  0.514   0.776   0.776
## 77     0.370  0.314   0.573   0.573
## 78     0.245  0.201   0.326   0.326
## 79     0.186  0.142   0.185   0.185
## 80     0.375  0.317   0.448   0.448
## 81     0.382  0.310   0.378   0.378
## 82     0.387  0.307   0.472   0.472
## 83     0.298  0.217   0.283   0.283
## 84     0.385  0.302   0.433   0.433
## 85     1.000  1.000   1.000   1.000
## 86     1.000  1.000   1.000   1.000
## 87     1.000  1.000   1.000   1.000
## 88     1.000  1.000   1.000   1.000
## 89     1.000  1.000   1.000   1.000
## 90     1.000  1.000   1.000   1.000
## 91     1.000  1.000   1.000   1.000
## 92     1.000  1.000   1.000   1.000
## 93     0.609  0.482   0.482   0.482
## 94     0.813  0.743   0.743   0.743
## 95    -0.452 -0.558  -0.558  -0.558
## 96    -0.325 -0.443  -0.443  -0.443
## 97    -0.352 -0.466  -0.466  -0.466
## 98    -0.417 -0.524  -0.524  -0.524
## 99    -0.158 -0.294  -0.294  -0.294
## 100    0.409  0.275   0.275   0.275
## 101   -0.286 -0.389  -0.389  -0.389
## 102   -0.642 -0.716  -0.716  -0.716
## 103   -0.539 -0.656  -0.656  -0.656
## 104   -0.247 -0.354  -0.354  -0.354
## 105   -0.076 -0.213  -0.213  -0.213
## 106   -0.408 -0.512  -0.512  -0.512
## 107   -0.074 -0.205  -0.205  -0.205
## 108   -0.145 -0.279  -0.279  -0.279
## 109   -0.393 -0.501  -0.501  -0.501
## 110   -0.085 -0.224  -0.224  -0.224
## 111    0.739  0.652   0.652   0.652
## 112    0.493  0.361   0.361   0.361
## 113    0.893  0.813   0.813   0.813
```

```
## 114    0.453  0.309   0.309   0.309
## 115    0.717  0.597   0.597   0.597
## 116    0.686  0.599   0.599   0.599
## 117    0.479  0.356   0.356   0.356
## 118    0.454  0.308   0.308   0.308
## 119    0.309  0.171   0.171   0.171
## 120    0.517  0.398   0.398   0.398
```

```
bla <- parameterEstimates(fit_o, standardized = TRUE)

model1   <- ' Anger =~ AEQ_A1 + AEQ_A2 + AEQ_A3 + AEQ_A4
              Anxiety =~ AMAS1 + AMAS2 + AMAS3 + AMAS4 + AMAS5 + AMAS6 + AMAS7 + AMAS8 + AMAS9
              Boredom =~ AEQ_B1 + AEQ_B2 + AEQ_B3 + AEQ_B4 + AEQ_B5
              Enjoyment =~ AEQ_E1 + AEQ_E2 + AEQ_E4
              Control =~ SelfE1 + SelfE2 + SelfE3 + SelfE4 + SelfE5 + SelfE6 + SelfE7 + SelfE8 + SelfC1
              SelfC4 + SelfC5
              Value =~ Val1 + Val2 + Val3 + Val4 + Int1 + Int2  + Int3 + Int4'
fit_m1 <- cfa(model1, data = data_Factors, std.lv = TRUE, estimator = "MLM")
summary(fit_m1, fit.measures = TRUE, standardized = TRUE)
```

```
## lavaan 0.6-8 ended normally after 40 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        99
##
##   Number of observations                           347
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                              2140.292    1819.456
##   Degrees of freedom                               804         804
##   P-value (Chi-square)                           0.000       0.000
##   Scaling correction factor                                  1.176
##        Satorra-Bentler correction
##
## Model Test Baseline Model:
##
##   Test statistic                              7074.558    5684.651
##   Degrees of freedom                               861         861
##   P-value                                        0.000       0.000
##   Scaling correction factor                                  1.245
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.785       0.789
##   Tucker-Lewis Index (TLI)                       0.770       0.775
##
##   Robust Comparative Fit Index (CFI)                         0.801
##   Robust Tucker-Lewis Index (TLI)                            0.787
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -17723.319  -17723.319
##   Loglikelihood unrestricted model (H1)     -16653.173  -16653.173
```

```
##
##   Akaike (AIC)                              35644.638   35644.638
##   Bayesian (BIC)                            36025.722   36025.722
##   Sample-size adjusted Bayesian (BIC)       35711.663   35711.663
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                         0.069       0.060
##   90 Percent confidence interval - lower        0.066       0.057
##   90 Percent confidence interval - upper        0.073       0.064
##   P-value RMSEA <= 0.05                         0.000       0.000
##
##   Robust RMSEA                                              0.065
##   90 Percent confidence interval - lower                    0.061
##   90 Percent confidence interval - upper                    0.069
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                          0.073       0.073
##
## 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
##   Anger =~
##     AEQ_A1            0.801    0.071   11.221    0.000    0.801    0.668
##     AEQ_A2            0.763    0.057   13.286    0.000    0.763    0.614
##     AEQ_A3            1.024    0.066   15.614    0.000    1.024    0.854
##     AEQ_A4            0.852    0.071   12.033    0.000    0.852    0.744
##   Anxiety =~
##     AMAS1             0.370    0.055    6.773    0.000    0.370    0.520
##     AMAS2             0.852    0.053   16.111    0.000    0.852    0.653
##     AMAS3             0.797    0.061   13.148    0.000    0.797    0.653
##     AMAS4             0.928    0.053   17.571    0.000    0.928    0.745
##     AMAS5             0.624    0.069    9.106    0.000    0.624    0.597
##     AMAS6             0.517    0.065    7.965    0.000    0.517    0.687
##     AMAS7             0.637    0.075    8.483    0.000    0.637    0.669
##     AMAS8             0.620    0.075    8.322    0.000    0.620    0.543
##     AMAS9             0.414    0.061    6.756    0.000    0.414    0.585
##   Boredom =~
##     AEQ_B1            0.781    0.057   13.740    0.000    0.781    0.636
##     AEQ_B2            0.739    0.063   11.766    0.000    0.739    0.714
##     AEQ_B3            0.788    0.073   10.856    0.000    0.788    0.633
##     AEQ_B4            0.949    0.058   16.428    0.000    0.949    0.741
##     AEQ_B5            0.949    0.054   17.508    0.000    0.949    0.741
##   Enjoyment =~
##     AEQ_E1            0.876    0.052   16.944    0.000    0.876    0.770
##     AEQ_E2            0.687    0.059   11.652    0.000    0.687    0.628
##     AEQ_E4            0.667    0.064   10.488    0.000    0.667    0.640
##   Control =~
```

```
##     SelfE1            0.306    0.050    6.153    0.000    0.306    0.320
##     SelfE2            0.049    0.047    1.034    0.301    0.049    0.067
##     SelfE3            0.276    0.050    5.481    0.000    0.276    0.293
##     SelfE4            0.098    0.047    2.104    0.035    0.098    0.120
##     SelfE5            0.360    0.045    8.050    0.000    0.360    0.422
##     SelfE6            0.331    0.050    6.550    0.000    0.331    0.365
##     SelfE7            0.485    0.047   10.345    0.000    0.485    0.510
##     SelfE8            0.336    0.043    7.751    0.000    0.336    0.412
##     SelfC1            0.720    0.039   18.332    0.000    0.720    0.800
##     SelfC2            0.550    0.038   14.590    0.000    0.550    0.701
##     SelfC3            0.702    0.036   19.315    0.000    0.702    0.781
##     SelfC4            0.769    0.038   20.322    0.000    0.769    0.744
##     SelfC5            0.647    0.035   18.365    0.000    0.647    0.754
##   Value =~
##     Val1              0.367    0.052    6.999    0.000    0.367    0.405
##     Val2              0.319    0.049    6.461    0.000    0.319    0.395
##     Val3              0.364    0.051    7.109    0.000    0.364    0.416
##     Val4              0.316    0.052    6.108    0.000    0.316    0.378
##     Int1              0.479    0.042   11.496    0.000    0.479    0.647
##     Int2              0.628    0.033   19.253    0.000    0.628    0.801
##     Int3              0.772    0.032   24.110    0.000    0.772    0.882
##     Int4              0.633    0.039   16.364    0.000    0.633    0.753
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   Anger ~~
##     Anxiety           0.483    0.065    7.438    0.000    0.483    0.483
##     Boredom           0.743    0.036   20.892    0.000    0.743    0.743
##     Enjoyment        -0.560    0.054  -10.353    0.000   -0.560   -0.560
##     Control          -0.471    0.054   -8.776    0.000   -0.471   -0.471
##     Value            -0.537    0.054  -10.038    0.000   -0.537   -0.537
##   Anxiety ~~
##     Boredom           0.274    0.069    3.995    0.000    0.274    0.274
##     Enjoyment        -0.392    0.053   -7.428    0.000   -0.392   -0.392
##     Control          -0.754    0.034  -22.284    0.000   -0.754   -0.754
##     Value            -0.365    0.056   -6.504    0.000   -0.365   -0.365
##   Boredom ~~
##     Enjoyment        -0.515    0.053   -9.709    0.000   -0.515   -0.515
##     Control          -0.229    0.065   -3.496    0.000   -0.229   -0.229
##     Value            -0.505    0.055   -9.135    0.000   -0.505   -0.505
##   Enjoyment ~~
##     Control           0.639    0.045   14.158    0.000    0.639    0.639
##     Value             0.806    0.040   20.087    0.000    0.806    0.806
##   Control ~~
##     Value             0.600    0.045   13.412    0.000    0.600    0.600
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .AEQ_A1            0.797    0.097    8.239    0.000    0.797    0.554
##    .AEQ_A2            0.965    0.079   12.276    0.000    0.965    0.624
##    .AEQ_A3            0.388    0.063    6.190    0.000    0.388    0.270
##    .AEQ_A4            0.585    0.093    6.264    0.000    0.585    0.446
##    .AMAS1             0.368    0.053    6.957    0.000    0.368    0.729
##    .AMAS2             0.975    0.082   11.916    0.000    0.975    0.573
```

```
##    .AMAS3              0.855    0.086    9.977    0.000    0.855    0.574
##    .AMAS4              0.689    0.070    9.864    0.000    0.689    0.444
##    .AMAS5              0.703    0.067   10.550    0.000    0.703    0.644
##    .AMAS6              0.300    0.047    6.376    0.000    0.300    0.529
##    .AMAS7              0.501    0.060    8.376    0.000    0.501    0.552
##    .AMAS8              0.922    0.083   11.074    0.000    0.922    0.705
##    .AMAS9              0.329    0.034    9.561    0.000    0.329    0.658
##    .AEQ_B1             0.901    0.076   11.901    0.000    0.901    0.596
##    .AEQ_B2             0.525    0.054    9.668    0.000    0.525    0.490
##    .AEQ_B3             0.928    0.098    9.427    0.000    0.928    0.599
##    .AEQ_B4             0.738    0.075    9.811    0.000    0.738    0.451
##    .AEQ_B5             0.740    0.073   10.116    0.000    0.740    0.451
##    .AEQ_E1             0.528    0.070    7.596    0.000    0.528    0.408
##    .AEQ_E2             0.724    0.083    8.723    0.000    0.724    0.605
##    .AEQ_E4             0.641    0.084    7.622    0.000    0.641    0.590
##    .SelfE1             0.817    0.046   17.719    0.000    0.817    0.897
##    .SelfE2             0.539    0.045   11.878    0.000    0.539    0.996
##    .SelfE3             0.808    0.048   16.963    0.000    0.808    0.914
##    .SelfE4             0.654    0.046   14.354    0.000    0.654    0.986
##    .SelfE5             0.598    0.053   11.243    0.000    0.598    0.822
##    .SelfE6             0.712    0.047   15.285    0.000    0.712    0.867
##    .SelfE7             0.670    0.048   14.002    0.000    0.670    0.740
##    .SelfE8             0.550    0.043   12.789    0.000    0.550    0.830
##    .SelfC1             0.291    0.037    7.861    0.000    0.291    0.360
##    .SelfC2             0.314    0.028   11.102    0.000    0.314    0.509
##    .SelfC3             0.315    0.031   10.239    0.000    0.315    0.389
##    .SelfC4             0.477    0.042   11.316    0.000    0.477    0.447
##    .SelfC5             0.318    0.029   10.926    0.000    0.318    0.431
##    .Val1               0.686    0.050   13.631    0.000    0.686    0.836
##    .Val2               0.550    0.045   12.325    0.000    0.550    0.844
##    .Val3               0.634    0.049   12.936    0.000    0.634    0.827
##    .Val4               0.597    0.047   12.591    0.000    0.597    0.857
##    .Int1               0.318    0.029   10.870    0.000    0.318    0.581
##    .Int2               0.221    0.024    9.195    0.000    0.221    0.359
##    .Int3               0.170    0.022    7.769    0.000    0.170    0.222
##    .Int4               0.306    0.026   11.543    0.000    0.306    0.433
##     Anger              1.000                               1.000    1.000
##     Anxiety            1.000                               1.000    1.000
##     Boredom            1.000                               1.000    1.000
##     Enjoyment          1.000                               1.000    1.000
##     Control            1.000                               1.000    1.000
##     Value              1.000                               1.000    1.000
```

```
model2   <- 'Negative =~ AEQ_A1 + AEQ_A2 + AEQ_A3 + AEQ_A4 + AMAS1 + AMAS2 + AMAS3 + AMAS4 + AMAS5 + AM
                          AMAS7 + AMAS8 + AMAS9 + AEQ_B1 + AEQ_B2 + AEQ_B3 + AEQ_B4 + AEQ_B5
             Positive =~ AEQ_E1 + AEQ_E2 + AEQ_E4 + SelfE1 + SelfE2 + SelfE3 + SelfE4 + SelfE5 + SelfE6
                          SelfE7 + SelfE8 + SelfC1 + SelfC2 + SelfC3 + SelfC4 + SelfC5 + Val1 + Val2 + V
                          Val4 + Int1 + Int2  + Int3 + Int4'
fit_m2 <- cfa(model2, data = data_Factors, std.lv = TRUE, estimator = "MLM")
summary(fit_m2, fit.measures = TRUE, standardized = TRUE)
```

```
## lavaan 0.6-8 ended normally after 27 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
```

```
##   Number of model parameters                        85
##
##   Number of observations                           347
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                              3514.899    2946.487
##   Degrees of freedom                               818         818
##   P-value (Chi-square)                           0.000       0.000
##   Scaling correction factor                                  1.193
##        Satorra-Bentler correction
##
## Model Test Baseline Model:
##
##   Test statistic                              7074.558    5684.651
##   Degrees of freedom                               861         861
##   P-value                                        0.000       0.000
##   Scaling correction factor                                  1.245
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.566       0.559
##   Tucker-Lewis Index (TLI)                       0.543       0.536
##
##   Robust Comparative Fit Index (CFI)                         0.577
##   Robust Tucker-Lewis Index (TLI)                            0.555
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -18410.623  -18410.623
##   Loglikelihood unrestricted model (H1)     -16653.173  -16653.173
##
##   Akaike (AIC)                               36991.246   36991.246
##   Bayesian (BIC)                             37318.439   37318.439
##   Sample-size adjusted Bayesian (BIC)        37048.793   37048.793
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.097       0.087
##   90 Percent confidence interval - lower         0.094       0.084
##   90 Percent confidence interval - upper         0.101       0.090
##   P-value RMSEA <= 0.05                          0.000       0.000
##
##   Robust RMSEA                                               0.095
##   90 Percent confidence interval - lower                     0.091
##   90 Percent confidence interval - upper                     0.098
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.104       0.104
##
## 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
##   Negative =~
##     AEQ_A1            0.755    0.062   12.113    0.000    0.755    0.629
##     AEQ_A2            0.725    0.057   12.714    0.000    0.725    0.583
##     AEQ_A3            0.811    0.068   11.968    0.000    0.811    0.677
##     AEQ_A4            0.712    0.067   10.695    0.000    0.712    0.622
##     AMAS1             0.326    0.050    6.481    0.000    0.326    0.458
##     AMAS2             0.610    0.065    9.377    0.000    0.610    0.468
##     AMAS3             0.684    0.066   10.384    0.000    0.684    0.561
##     AMAS4             0.736    0.061   12.066    0.000    0.736    0.591
##     AMAS5             0.474    0.073    6.464    0.000    0.474    0.454
##     AMAS6             0.463    0.066    7.038    0.000    0.463    0.615
##     AMAS7             0.551    0.072    7.676    0.000    0.551    0.578
##     AMAS8             0.464    0.078    5.949    0.000    0.464    0.406
##     AMAS9             0.342    0.060    5.708    0.000    0.342    0.484
##     AEQ_B1            0.546    0.065    8.351    0.000    0.546    0.444
##     AEQ_B2            0.529    0.060    8.783    0.000    0.529    0.511
##     AEQ_B3            0.652    0.069    9.488    0.000    0.652    0.524
##     AEQ_B4            0.653    0.068    9.628    0.000    0.653    0.511
##     AEQ_B5            0.619    0.064    9.723    0.000    0.619    0.484
##   Positive =~
##     AEQ_E1            0.742    0.050   14.730    0.000    0.742    0.652
##     AEQ_E2            0.592    0.057   10.443    0.000    0.592    0.541
##     AEQ_E4            0.513    0.063    8.165    0.000    0.513    0.492
##     SelfE1            0.278    0.050    5.565    0.000    0.278    0.291
##     SelfE2            0.021    0.046    0.450    0.653    0.021    0.028
##     SelfE3            0.218    0.049    4.426    0.000    0.218    0.231
##     SelfE4            0.071    0.045    1.560    0.119    0.071    0.087
##     SelfE5            0.316    0.044    7.158    0.000    0.316    0.370
##     SelfE6            0.262    0.050    5.265    0.000    0.262    0.289
##     SelfE7            0.426    0.047    9.072    0.000    0.426    0.448
##     SelfE8            0.296    0.044    6.711    0.000    0.296    0.364
##     SelfC1            0.681    0.039   17.501    0.000    0.681    0.756
##     SelfC2            0.483    0.039   12.320    0.000    0.483    0.615
##     SelfC3            0.615    0.039   15.820    0.000    0.615    0.684
##     SelfC4            0.718    0.040   17.894    0.000    0.718    0.694
##     SelfC5            0.590    0.039   15.122    0.000    0.590    0.688
##     Val1              0.348    0.054    6.464    0.000    0.348    0.384
##     Val2              0.300    0.050    5.972    0.000    0.300    0.372
##     Val3              0.358    0.052    6.940    0.000    0.358    0.409
##     Val4              0.308    0.050    6.120    0.000    0.308    0.369
##     Int1              0.398    0.041    9.685    0.000    0.398    0.538
##     Int2              0.493    0.036   13.723    0.000    0.493    0.628
##     Int3              0.661    0.034   19.678    0.000    0.661    0.755
##     Int4              0.562    0.039   14.350    0.000    0.562    0.668
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   Negative ~~
##     Positive         -0.704    0.036  -19.788    0.000   -0.704   -0.704
```

```
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .AEQ_A1            0.869    0.083   10.472    0.000    0.869    0.604
##    .AEQ_A2            1.022    0.080   12.860    0.000    1.022    0.661
##    .AEQ_A3            0.780    0.081    9.584    0.000    0.780    0.542
##    .AEQ_A4            0.804    0.095    8.451    0.000    0.804    0.613
##    .AMAS1             0.399    0.062    6.428    0.000    0.399    0.790
##    .AMAS2             1.328    0.090   14.804    0.000    1.328    0.781
##    .AMAS3             1.022    0.089   11.486    0.000    1.022    0.686
##    .AMAS4             1.010    0.076   13.351    0.000    1.010    0.651
##    .AMAS5             0.867    0.069   12.633    0.000    0.867    0.794
##    .AMAS6             0.353    0.051    6.910    0.000    0.353    0.622
##    .AMAS7             0.604    0.070    8.643    0.000    0.604    0.666
##    .AMAS8             1.091    0.088   12.441    0.000    1.091    0.835
##    .AMAS9             0.383    0.041    9.332    0.000    0.383    0.766
##    .AEQ_B1            1.213    0.083   14.684    0.000    1.213    0.803
##    .AEQ_B2            0.792    0.081    9.750    0.000    0.792    0.739
##    .AEQ_B3            1.124    0.090   12.426    0.000    1.124    0.726
##    .AEQ_B4            1.211    0.091   13.360    0.000    1.211    0.739
##    .AEQ_B5            1.257    0.090   13.973    0.000    1.257    0.766
##    .AEQ_E1            0.745    0.067   11.109    0.000    0.745    0.575
##    .AEQ_E2            0.846    0.076   11.116    0.000    0.846    0.707
##    .AEQ_E4            0.823    0.086    9.542    0.000    0.823    0.758
##    .SelfE1            0.833    0.047   17.567    0.000    0.833    0.915
##    .SelfE2            0.541    0.046   11.647    0.000    0.541    0.999
##    .SelfE3            0.837    0.048   17.335    0.000    0.837    0.946
##    .SelfE4            0.659    0.046   14.314    0.000    0.659    0.992
##    .SelfE5            0.628    0.053   11.752    0.000    0.628    0.863
##    .SelfE6            0.753    0.047   16.104    0.000    0.753    0.917
##    .SelfE7            0.723    0.049   14.839    0.000    0.723    0.799
##    .SelfE8            0.575    0.045   12.873    0.000    0.575    0.868
##    .SelfC1            0.347    0.032   10.710    0.000    0.347    0.428
##    .SelfC2            0.384    0.032   11.938    0.000    0.384    0.622
##    .SelfC3            0.430    0.033   12.981    0.000    0.430    0.532
##    .SelfC4            0.554    0.046   11.982    0.000    0.554    0.518
##    .SelfC5            0.388    0.033   11.764    0.000    0.388    0.527
##    .Val1              0.700    0.051   13.818    0.000    0.700    0.853
##    .Val2              0.561    0.045   12.542    0.000    0.561    0.862
##    .Val3              0.638    0.049   12.918    0.000    0.638    0.833
##    .Val4              0.602    0.046   13.128    0.000    0.602    0.864
##    .Int1              0.389    0.036   10.817    0.000    0.389    0.711
##    .Int2              0.372    0.031   12.002    0.000    0.372    0.605
##    .Int3              0.330    0.028   11.964    0.000    0.330    0.430
##    .Int4              0.391    0.031   12.810    0.000    0.391    0.554
##     Negative          1.000                               1.000    1.000
##     Positive          1.000                               1.000    1.000
```

```r
# Without SE & Intrinsic Value
model3   <- ' Anger =~ AEQ_A1 + AEQ_A2 + AEQ_A3 + AEQ_A4
              Anxiety =~ AMAS1 + AMAS2 + AMAS3 + AMAS4 + AMAS5 + AMAS6 + AMAS7 + AMAS8 + AMAS9
              Boredom =~ AEQ_B1 + AEQ_B2 + AEQ_B3 + AEQ_B4 + AEQ_B5
              Enjoyment =~ AEQ_E1 + AEQ_E2 + AEQ_E4
              Self_Concept =~ SelfC1 + SelfC2 + SelfC3 + SelfC4 + SelfC5
```

```r
              InstrumentalVal =~ Val1 + Val2 + Val3 + Val4'
fit_m3 <- cfa(model3, data = data_Factors, std.lv = TRUE, estimator = "MLM")
summary(fit_m3, fit.measures = TRUE, standardized = TRUE)
```

```
## lavaan 0.6-8 ended normally after 29 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        75
##
##   Number of observations                           347
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                               786.080     654.614
##   Degrees of freedom                               390         390
##   P-value (Chi-square)                           0.000       0.000
##   Scaling correction factor                                  1.201
##        Satorra-Bentler correction
##
## Model Test Baseline Model:
##
##   Test statistic                              5010.862    3781.559
##   Degrees of freedom                               435         435
##   P-value                                        0.000       0.000
##   Scaling correction factor                                  1.325
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.913       0.921
##   Tucker-Lewis Index (TLI)                       0.903       0.912
##
##   Robust Comparative Fit Index (CFI)                         0.928
##   Robust Tucker-Lewis Index (TLI)                            0.920
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -12857.814  -12857.814
##   Loglikelihood unrestricted model (H1)     -12464.774  -12464.774
##
##   Akaike (AIC)                               25865.628   25865.628
##   Bayesian (BIC)                             26154.328   26154.328
##   Sample-size adjusted Bayesian (BIC)        25916.405   25916.405
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.054       0.044
##   90 Percent confidence interval - lower         0.049       0.039
##   90 Percent confidence interval - upper         0.060       0.050
##   P-value RMSEA <= 0.05                          0.107       0.963
##
##   Robust RMSEA                                               0.048
##   90 Percent confidence interval - lower                     0.042
##   90 Percent confidence interval - upper                     0.055
```

```
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                          0.055       0.055
##
## 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
##   Anger =~
##     AEQ_A1              0.799    0.071   11.190    0.000    0.799    0.667
##     AEQ_A2              0.766    0.057   13.364    0.000    0.766    0.616
##     AEQ_A3              1.021    0.066   15.511    0.000    1.021    0.852
##     AEQ_A4              0.854    0.071   12.085    0.000    0.854    0.746
##   Anxiety =~
##     AMAS1               0.371    0.055    6.778    0.000    0.371    0.522
##     AMAS2               0.847    0.053   15.840    0.000    0.847    0.649
##     AMAS3               0.801    0.061   13.143    0.000    0.801    0.656
##     AMAS4               0.926    0.053   17.572    0.000    0.926    0.743
##     AMAS5               0.626    0.069    9.089    0.000    0.626    0.599
##     AMAS6               0.521    0.065    7.979    0.000    0.521    0.692
##     AMAS7               0.640    0.076    8.470    0.000    0.640    0.672
##     AMAS8               0.613    0.075    8.161    0.000    0.613    0.537
##     AMAS9               0.411    0.062    6.666    0.000    0.411    0.581
##   Boredom =~
##     AEQ_B1              0.784    0.057   13.830    0.000    0.784    0.638
##     AEQ_B2              0.740    0.062   11.860    0.000    0.740    0.715
##     AEQ_B3              0.780    0.073   10.725    0.000    0.780    0.627
##     AEQ_B4              0.952    0.058   16.456    0.000    0.952    0.744
##     AEQ_B5              0.949    0.054   17.456    0.000    0.949    0.741
##   Enjoyment =~
##     AEQ_E1              0.881    0.055   16.090    0.000    0.881    0.774
##     AEQ_E2              0.694    0.059   11.721    0.000    0.694    0.635
##     AEQ_E4              0.655    0.064   10.283    0.000    0.655    0.628
##   Self_Concept =~
##     SelfC1              0.728    0.040   18.282    0.000    0.728    0.809
##     SelfC2              0.560    0.038   14.701    0.000    0.560    0.713
##     SelfC3              0.715    0.036   19.783    0.000    0.715    0.795
##     SelfC4              0.785    0.038   20.865    0.000    0.785    0.759
##     SelfC5              0.647    0.035   18.234    0.000    0.647    0.754
##   InstrumentalVal =~
##     Val1                0.713    0.041   17.510    0.000    0.713    0.787
##     Val2                0.586    0.048   12.145    0.000    0.586    0.726
##     Val3                0.742    0.038   19.535    0.000    0.742    0.848
##     Val4                0.630    0.044   14.295    0.000    0.630    0.754
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   Anger ~~
##     Anxiety            0.485    0.065    7.464    0.000    0.485    0.485
```

```
##     Boredom           0.743    0.036   20.845    0.000    0.743    0.743
##     Enjoyment        -0.562    0.054  -10.477    0.000   -0.562   -0.562
##     Self_Concept     -0.445    0.060   -7.382    0.000   -0.445   -0.445
##     InstrumentalVl   -0.295    0.069   -4.256    0.000   -0.295   -0.295
##   Anxiety ~~
##     Boredom           0.274    0.069    4.001    0.000    0.274    0.274
##     Enjoyment        -0.393    0.053   -7.478    0.000   -0.393   -0.393
##     Self_Concept     -0.717    0.038  -18.989    0.000   -0.717   -0.717
##     InstrumentalVl   -0.214    0.070   -3.056    0.002   -0.214   -0.214
##   Boredom ~~
##     Enjoyment        -0.513    0.053   -9.713    0.000   -0.513   -0.513
##     Self_Concept     -0.205    0.067   -3.071    0.002   -0.205   -0.205
##     InstrumentalVl   -0.224    0.071   -3.155    0.002   -0.224   -0.224
##   Enjoyment ~~
##     Self_Concept      0.654    0.044   14.703    0.000    0.654    0.654
##     InstrumentalVl    0.311    0.074    4.223    0.000    0.311    0.311
##   Self_Concept ~~
##     InstrumentalVl    0.355    0.063    5.640    0.000    0.355    0.355
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .AEQ_A1            0.799    0.097    8.240    0.000    0.799    0.556
##    .AEQ_A2            0.960    0.079   12.212    0.000    0.960    0.620
##    .AEQ_A3            0.395    0.063    6.259    0.000    0.395    0.275
##    .AEQ_A4            0.581    0.094    6.209    0.000    0.581    0.443
##    .AMAS1             0.367    0.053    6.989    0.000    0.367    0.728
##    .AMAS2             0.984    0.083   11.898    0.000    0.984    0.578
##    .AMAS3             0.850    0.086    9.932    0.000    0.850    0.570
##    .AMAS4             0.694    0.070    9.953    0.000    0.694    0.447
##    .AMAS5             0.700    0.067   10.470    0.000    0.700    0.641
##    .AMAS6             0.296    0.046    6.398    0.000    0.296    0.521
##    .AMAS7             0.498    0.060    8.355    0.000    0.498    0.549
##    .AMAS8             0.930    0.084   11.090    0.000    0.930    0.712
##    .AMAS9             0.331    0.035    9.517    0.000    0.331    0.662
##    .AEQ_B1            0.896    0.076   11.780    0.000    0.896    0.593
##    .AEQ_B2            0.524    0.055    9.606    0.000    0.524    0.489
##    .AEQ_B3            0.940    0.098    9.564    0.000    0.940    0.607
##    .AEQ_B4            0.732    0.077    9.561    0.000    0.732    0.447
##    .AEQ_B5            0.741    0.073   10.079    0.000    0.741    0.451
##    .AEQ_E1            0.520    0.072    7.198    0.000    0.520    0.401
##    .AEQ_E2            0.715    0.083    8.620    0.000    0.715    0.597
##    .AEQ_E4            0.657    0.078    8.400    0.000    0.657    0.605
##    .SelfC1            0.280    0.038    7.272    0.000    0.280    0.345
##    .SelfC2            0.303    0.029   10.585    0.000    0.303    0.491
##    .SelfC3            0.297    0.032    9.403    0.000    0.297    0.367
##    .SelfC4            0.453    0.041   10.950    0.000    0.453    0.424
##    .SelfC5            0.318    0.030   10.600    0.000    0.318    0.432
##    .Val1              0.312    0.037    8.427    0.000    0.312    0.380
##    .Val2              0.308    0.041    7.492    0.000    0.308    0.473
##    .Val3              0.215    0.041    5.229    0.000    0.215    0.281
##    .Val4              0.301    0.041    7.268    0.000    0.301    0.431
##     Anger             1.000                               1.000    1.000
##     Anxiety           1.000                               1.000    1.000
##     Boredom           1.000                               1.000    1.000
```

```
##     Enjoyment         1.000                                1.000    1.000
##     Self_Concept      1.000                                1.000    1.000
##     InstrumentalVl    1.000                                1.000    1.000
```

```r
parameterEstimates(fit_m3, standardized = TRUE)
```

```
##                lhs op            rhs   est    se       z pvalue ci.lower
## 1            Anger =~         AEQ_A1 0.799 0.071  11.190  0.000    0.659
## 2            Anger =~         AEQ_A2 0.766 0.057  13.364  0.000    0.654
## 3            Anger =~         AEQ_A3 1.021 0.066  15.511  0.000    0.892
## 4            Anger =~         AEQ_A4 0.854 0.071  12.085  0.000    0.716
## 5          Anxiety =~          AMAS1 0.371 0.055   6.778  0.000    0.264
## 6          Anxiety =~          AMAS2 0.847 0.053  15.840  0.000    0.742
## 7          Anxiety =~          AMAS3 0.801 0.061  13.143  0.000    0.681
## 8          Anxiety =~          AMAS4 0.926 0.053  17.572  0.000    0.823
## 9          Anxiety =~          AMAS5 0.626 0.069   9.089  0.000    0.491
## 10         Anxiety =~          AMAS6 0.521 0.065   7.979  0.000    0.393
## 11         Anxiety =~          AMAS7 0.640 0.076   8.470  0.000    0.492
## 12         Anxiety =~          AMAS8 0.613 0.075   8.161  0.000    0.466
## 13         Anxiety =~          AMAS9 0.411 0.062   6.666  0.000    0.290
## 14         Boredom =~         AEQ_B1 0.784 0.057  13.830  0.000    0.673
## 15         Boredom =~         AEQ_B2 0.740 0.062  11.860  0.000    0.618
## 16         Boredom =~         AEQ_B3 0.780 0.073  10.725  0.000    0.638
## 17         Boredom =~         AEQ_B4 0.952 0.058  16.456  0.000    0.839
## 18         Boredom =~         AEQ_B5 0.949 0.054  17.456  0.000    0.842
## 19       Enjoyment =~         AEQ_E1 0.881 0.055  16.090  0.000    0.773
## 20       Enjoyment =~         AEQ_E2 0.694 0.059  11.721  0.000    0.578
## 21       Enjoyment =~         AEQ_E4 0.655 0.064  10.283  0.000    0.530
## 22    Self_Concept =~         SelfC1 0.728 0.040  18.282  0.000    0.650
## 23    Self_Concept =~         SelfC2 0.560 0.038  14.701  0.000    0.485
## 24    Self_Concept =~         SelfC3 0.715 0.036  19.783  0.000    0.644
## 25    Self_Concept =~         SelfC4 0.785 0.038  20.865  0.000    0.711
## 26    Self_Concept =~         SelfC5 0.647 0.035  18.234  0.000    0.577
## 27 InstrumentalVal =~           Val1 0.713 0.041  17.510  0.000    0.634
## 28 InstrumentalVal =~           Val2 0.586 0.048  12.145  0.000    0.491
## 29 InstrumentalVal =~           Val3 0.742 0.038  19.535  0.000    0.668
## 30 InstrumentalVal =~           Val4 0.630 0.044  14.295  0.000    0.543
## 31          AEQ_A1 ~~         AEQ_A1 0.799 0.097   8.240  0.000    0.609
## 32          AEQ_A2 ~~         AEQ_A2 0.960 0.079  12.212  0.000    0.806
## 33          AEQ_A3 ~~         AEQ_A3 0.395 0.063   6.259  0.000    0.271
## 34          AEQ_A4 ~~         AEQ_A4 0.581 0.094   6.209  0.000    0.397
## 35           AMAS1 ~~          AMAS1 0.367 0.053   6.989  0.000    0.264
## 36           AMAS2 ~~          AMAS2 0.984 0.083  11.898  0.000    0.822
## 37           AMAS3 ~~          AMAS3 0.850 0.086   9.932  0.000    0.682
## 38           AMAS4 ~~          AMAS4 0.694 0.070   9.953  0.000    0.557
## 39           AMAS5 ~~          AMAS5 0.700 0.067  10.470  0.000    0.569
## 40           AMAS6 ~~          AMAS6 0.296 0.046   6.398  0.000    0.205
## 41           AMAS7 ~~          AMAS7 0.498 0.060   8.355  0.000    0.381
## 42           AMAS8 ~~          AMAS8 0.930 0.084  11.090  0.000    0.766
## 43           AMAS9 ~~          AMAS9 0.331 0.035   9.517  0.000    0.263
## 44          AEQ_B1 ~~         AEQ_B1 0.896 0.076  11.780  0.000    0.747
## 45          AEQ_B2 ~~         AEQ_B2 0.524 0.055   9.606  0.000    0.417
## 46          AEQ_B3 ~~         AEQ_B3 0.940 0.098   9.564  0.000    0.748
## 47          AEQ_B4 ~~         AEQ_B4 0.732 0.077   9.561  0.000    0.582
## 48          AEQ_B5 ~~         AEQ_B5 0.741 0.073  10.079  0.000    0.597
```

```
## 49          AEQ_E1 ~~          AEQ_E1  0.520 0.072   7.198  0.000    0.379
## 50          AEQ_E2 ~~          AEQ_E2  0.715 0.083   8.620  0.000    0.552
## 51          AEQ_E4 ~~          AEQ_E4  0.657 0.078   8.400  0.000    0.504
## 52          SelfC1 ~~          SelfC1  0.280 0.038   7.272  0.000    0.204
## 53          SelfC2 ~~          SelfC2  0.303 0.029  10.585  0.000    0.247
## 54          SelfC3 ~~          SelfC3  0.297 0.032   9.403  0.000    0.235
## 55          SelfC4 ~~          SelfC4  0.453 0.041  10.950  0.000    0.372
## 56          SelfC5 ~~          SelfC5  0.318 0.030  10.600  0.000    0.259
## 57            Val1 ~~            Val1  0.312 0.037   8.427  0.000    0.239
## 58            Val2 ~~            Val2  0.308 0.041   7.492  0.000    0.227
## 59            Val3 ~~            Val3  0.215 0.041   5.229  0.000    0.135
## 60            Val4 ~~            Val4  0.301 0.041   7.268  0.000    0.220
## 61           Anger ~~           Anger  1.000 0.000      NA     NA    1.000
## 62         Anxiety ~~         Anxiety  1.000 0.000      NA     NA    1.000
## 63         Boredom ~~         Boredom  1.000 0.000      NA     NA    1.000
## 64       Enjoyment ~~       Enjoyment  1.000 0.000      NA     NA    1.000
## 65    Self_Concept ~~    Self_Concept  1.000 0.000      NA     NA    1.000
## 66 InstrumentalVal ~~ InstrumentalVal  1.000 0.000      NA     NA    1.000
## 67           Anger ~~         Anxiety  0.485 0.065   7.464  0.000    0.357
## 68           Anger ~~         Boredom  0.743 0.036  20.845  0.000    0.673
## 69           Anger ~~       Enjoyment -0.562 0.054 -10.477  0.000   -0.667
## 70           Anger ~~    Self_Concept -0.445 0.060  -7.382  0.000   -0.563
## 71           Anger ~~ InstrumentalVal -0.295 0.069  -4.256  0.000   -0.431
## 72         Anxiety ~~         Boredom  0.274 0.069   4.001  0.000    0.140
## 73         Anxiety ~~       Enjoyment -0.393 0.053  -7.478  0.000   -0.496
## 74         Anxiety ~~    Self_Concept -0.717 0.038 -18.989  0.000   -0.791
## 75         Anxiety ~~ InstrumentalVal -0.214 0.070  -3.056  0.002   -0.351
## 76         Boredom ~~       Enjoyment -0.513 0.053  -9.713  0.000   -0.617
## 77         Boredom ~~    Self_Concept -0.205 0.067  -3.071  0.002   -0.336
## 78         Boredom ~~ InstrumentalVal -0.224 0.071  -3.155  0.002   -0.363
## 79       Enjoyment ~~    Self_Concept  0.654 0.044  14.703  0.000    0.567
## 80       Enjoyment ~~ InstrumentalVal  0.311 0.074   4.223  0.000    0.167
## 81    Self_Concept ~~ InstrumentalVal  0.355 0.063   5.640  0.000    0.232
##    ci.upper std.lv std.all std.nox
## 1     0.939  0.799   0.667   0.667
## 2     0.879  0.766   0.616   0.616
## 3     1.150  1.021   0.852   0.852
## 4     0.993  0.854   0.746   0.746
## 5     0.478  0.371   0.522   0.522
## 6     0.951  0.847   0.649   0.649
## 7     0.920  0.801   0.656   0.656
## 8     1.029  0.926   0.743   0.743
## 9     0.761  0.626   0.599   0.599
## 10    0.649  0.521   0.692   0.692
## 11    0.788  0.640   0.672   0.672
## 12    0.761  0.613   0.537   0.537
## 13    0.532  0.411   0.581   0.581
## 14    0.896  0.784   0.638   0.638
## 15    0.862  0.740   0.715   0.715
## 16    0.923  0.780   0.627   0.627
## 17    1.065  0.952   0.744   0.744
## 18    1.055  0.949   0.741   0.741
## 19    0.988  0.881   0.774   0.774
## 20    0.810  0.694   0.635   0.635
```

```
## 21    0.780  0.655   0.628   0.628
## 22    0.806  0.728   0.809   0.809
## 23    0.635  0.560   0.713   0.713
## 24    0.786  0.715   0.795   0.795
## 25    0.859  0.785   0.759   0.759
## 26    0.717  0.647   0.754   0.754
## 27    0.793  0.713   0.787   0.787
## 28    0.681  0.586   0.726   0.726
## 29    0.817  0.742   0.848   0.848
## 30    0.716  0.630   0.754   0.754
## 31    0.989  0.799   0.556   0.556
## 32    1.115  0.960   0.620   0.620
## 33    0.519  0.395   0.275   0.275
## 34    0.764  0.581   0.443   0.443
## 35    0.470  0.367   0.728   0.728
## 36    1.146  0.984   0.578   0.578
## 37    1.017  0.850   0.570   0.570
## 38    0.830  0.694   0.447   0.447
## 39    0.831  0.700   0.641   0.641
## 40    0.386  0.296   0.521   0.521
## 41    0.615  0.498   0.549   0.549
## 42    1.095  0.930   0.712   0.712
## 43    0.400  0.331   0.662   0.662
## 44    1.045  0.896   0.593   0.593
## 45    0.631  0.524   0.489   0.489
## 46    1.133  0.940   0.607   0.607
## 47    0.882  0.732   0.447   0.447
## 48    0.885  0.741   0.451   0.451
## 49    0.662  0.520   0.401   0.401
## 50    0.878  0.715   0.597   0.597
## 51    0.811  0.657   0.605   0.605
## 52    0.355  0.280   0.345   0.345
## 53    0.359  0.303   0.491   0.491
## 54    0.359  0.297   0.367   0.367
## 55    0.534  0.453   0.424   0.424
## 56    0.377  0.318   0.432   0.432
## 57    0.384  0.312   0.380   0.380
## 58    0.388  0.308   0.473   0.473
## 59    0.296  0.215   0.281   0.281
## 60    0.382  0.301   0.431   0.431
## 61    1.000  1.000   1.000   1.000
## 62    1.000  1.000   1.000   1.000
## 63    1.000  1.000   1.000   1.000
## 64    1.000  1.000   1.000   1.000
## 65    1.000  1.000   1.000   1.000
## 66    1.000  1.000   1.000   1.000
## 67    0.612  0.485   0.485   0.485
## 68    0.813  0.743   0.743   0.743
## 69   -0.457 -0.562  -0.562  -0.562
## 70   -0.327 -0.445  -0.445  -0.445
## 71   -0.159 -0.295  -0.295  -0.295
## 72    0.409  0.274   0.274   0.274
## 73   -0.290 -0.393  -0.393  -0.393
## 74   -0.643 -0.717  -0.717  -0.717
```

```
## 75   -0.077 -0.214  -0.214  -0.214
## 76   -0.410 -0.513  -0.513  -0.513
## 77   -0.074 -0.205  -0.205  -0.205
## 78   -0.085 -0.224  -0.224  -0.224
## 79    0.741  0.654   0.654   0.654
## 80    0.455  0.311   0.311   0.311
## 81    0.479  0.355   0.355   0.355
```

Profiles - Trait

```
load("data_Trait_usable.Rda")
data <- data_Trait_usable

# Cluster analysis
data_Trait_stan <- na.omit(data[c(1,54,55,56,62,59,60,51)])
data_Trait_stan[c(2:8)] <- standardize(data_Trait_stan[c(2:8)]) # select variables
data_Trait_mclust <- Mclust(data_Trait_stan[c(2:8)])
summary(data_Trait_mclust, parameters = TRUE)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VEE (ellipsoidal, equal shape and orientation) model with 2 components:
##
##  log-likelihood   n df       BIC       ICL
##       -2860.812 338 44 -5977.839 -6048.808
##
## Clustering table:
##   1   2
##  88 250
##
## Mixing probabilities:
##         1         2
## 0.2984269 0.7015731
##
## Means:
##                     [,1]       [,2]
## AEQ_Ang_mean    0.9950131 -0.4232470
## AMAS_mean       0.5913289 -0.2515325
## AEQ_Bor_mean    0.6991896 -0.2974130
## AEQ_Enj_3_mean -0.3318939  0.1411771
## SELFC_mean     -0.4636369  0.1972164
## VAL_mean       -0.4627793  0.1968516
## Grade          -0.5452792  0.2319444
##
## Variances:
## [,,1]
##                AEQ_Ang_mean   AMAS_mean AEQ_Bor_mean AEQ_Enj_3_mean  SELFC_mean
## AEQ_Ang_mean      0.8418684  0.26456088   0.49634740     -0.4832592 -0.31384108
## AMAS_mean         0.2645609  1.26572792   0.09183274     -0.3431815 -0.81544707
## AEQ_Bor_mean      0.4963474  0.09183274   1.28060142     -0.5513275 -0.07441792
## AEQ_Enj_3_mean   -0.4832592 -0.34318151  -0.55132755      1.6473428  0.75703676
```

```
## SELFC_mean       -0.3138411 -0.81544707  -0.07441792      0.7570368  1.52626930
## VAL_mean         -0.1196211 -0.07905677  -0.10061194      0.3397873  0.37201796
## Grade            -0.1645984 -0.52392961  -0.06115722      0.4767422  1.01859014
##                    VAL_mean       Grade
## AEQ_Ang_mean   -0.11962109 -0.16459843
## AMAS_mean      -0.07905677 -0.52392961
## AEQ_Bor_mean   -0.10061194 -0.06115722
## AEQ_Enj_3_mean  0.33978725  0.47674220
## SELFC_mean      0.37201796  1.01859014
## VAL_mean        1.49704283  0.16650387
## Grade           0.16650387  1.52497888
## [,,2]
##                AEQ_Ang_mean   AMAS_mean AEQ_Bor_mean AEQ_Enj_3_mean  SELFC_mean
## AEQ_Ang_mean     0.38773446  0.12184728   0.22859985     -0.2225719 -0.14454397
## AMAS_mean        0.12184728  0.58294900   0.04229488     -0.1580571 -0.37556574
## AEQ_Bor_mean     0.22859985  0.04229488   0.58979920     -0.2539217 -0.03427423
## AEQ_Enj_3_mean  -0.22257189 -0.15805713  -0.25392174      0.7587072  0.34866404
## SELFC_mean      -0.14454397 -0.37556574  -0.03427423      0.3486640  0.70294503
## VAL_mean        -0.05509319 -0.03641072  -0.04633826      0.1564939  0.17133816
## Grade           -0.07580815 -0.24130323  -0.02816683      0.2195704  0.46912617
##                    VAL_mean       Grade
## AEQ_Ang_mean   -0.05509319 -0.07580815
## AMAS_mean      -0.03641072 -0.24130323
## AEQ_Bor_mean   -0.04633826 -0.02816683
## AEQ_Enj_3_mean  0.15649385  0.21957040
## SELFC_mean      0.17133816  0.46912617
## VAL_mean        0.68948436  0.07668573
## Grade           0.07668573  0.70235071
```

```
mclustBIC(data_Trait_stan[c(2:8)])
```

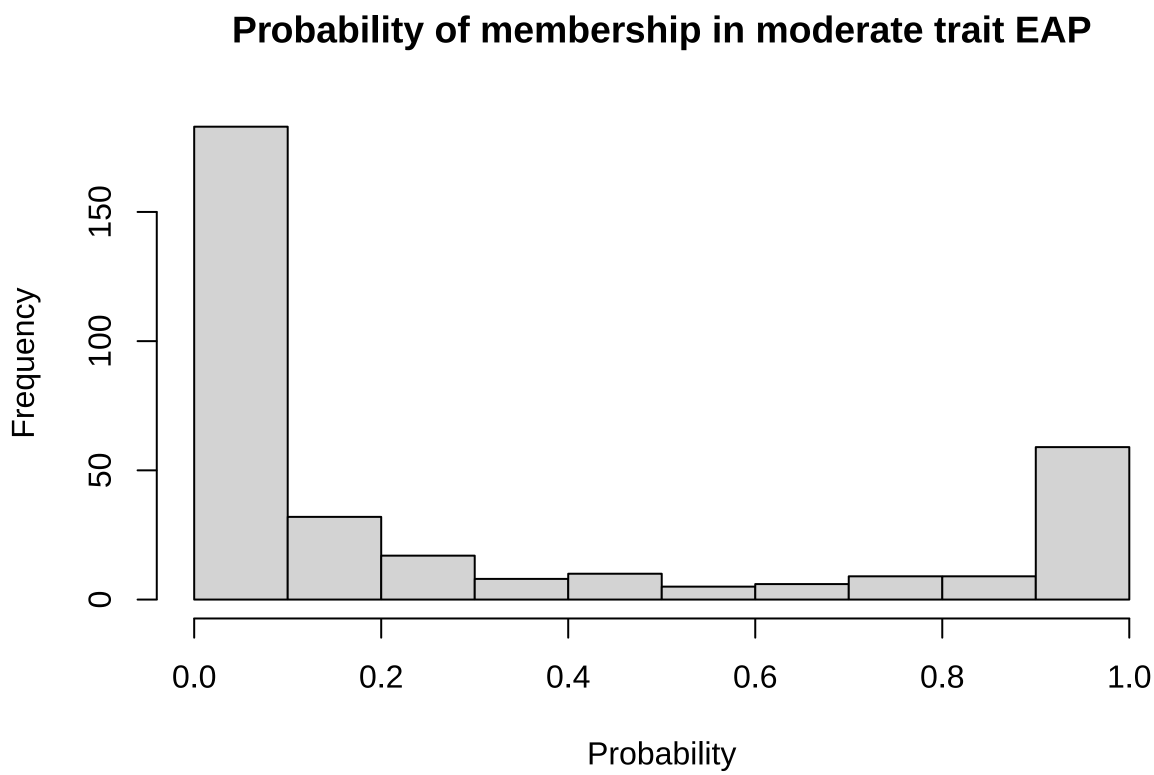
```
## Bayesian Information Criterion (BIC):
##         EII       VII       EEI       VEI       EVI       VVI       EEE
## 1 -6753.991 -6753.991 -6788.929 -6788.929 -6788.929 -6788.929 -6058.374
## 2 -6370.952 -6331.461 -6382.207 -6339.321 -6293.541 -6270.088 -6072.731
## 3 -6271.108 -6225.705 -6255.025 -6209.054 -6191.301 -6153.677 -6027.398
## 4 -6233.993 -6195.780 -6202.238 -6151.395 -6143.078 -6094.230 -6043.176
## 5 -6228.223 -6190.893 -6175.478 -6120.765 -6160.959 -6123.288 -6049.518
## 6 -6242.346 -6209.952 -6179.008 -6118.320 -6148.763 -6174.039 -6069.722
## 7 -6256.388 -6217.150 -6156.812 -6102.558        NA -6174.125 -6090.984
## 8 -6249.052 -6237.358 -6166.422 -6139.650 -6209.976        NA -6132.376
## 9 -6286.348 -6234.781 -6191.166 -6159.469 -6268.472        NA -6169.684
##         VEE       EVE       VVE       EEV       VEV       EVV       VVV
## 1 -6058.374 -6058.374 -6058.374 -6058.374 -6058.374 -6058.374 -6058.374
## 2 -5977.839 -6041.406 -5995.239 -6028.903 -6006.580 -6056.144 -6033.098
## 3 -6008.084 -6042.040 -5992.550 -6123.373 -6094.187 -6183.504 -6142.130
## 4 -6026.993 -6073.993 -6041.828 -6169.594 -6153.582 -6245.575 -6236.108
## 5 -6054.824 -6053.788 -6097.353 -6280.726 -6263.964 -6385.054 -6382.565
## 6 -6058.204 -6165.388 -6137.461 -6395.101 -6395.711 -6543.937 -6517.757
## 7 -6107.817 -6193.503 -6173.779 -6428.692 -6517.632 -6638.190 -6655.088
## 8 -6111.967        NA -6210.299 -6548.370 -6652.109 -6762.588 -6715.775
## 9 -6151.058 -6234.348 -6278.016 -6676.765 -6745.607 -6902.958 -6867.542
##
## Top 3 models based on the BIC criterion:
##     VEE,2     VVE,3     VVE,2
```

```
## -5977.839 -5992.550 -5995.239
```

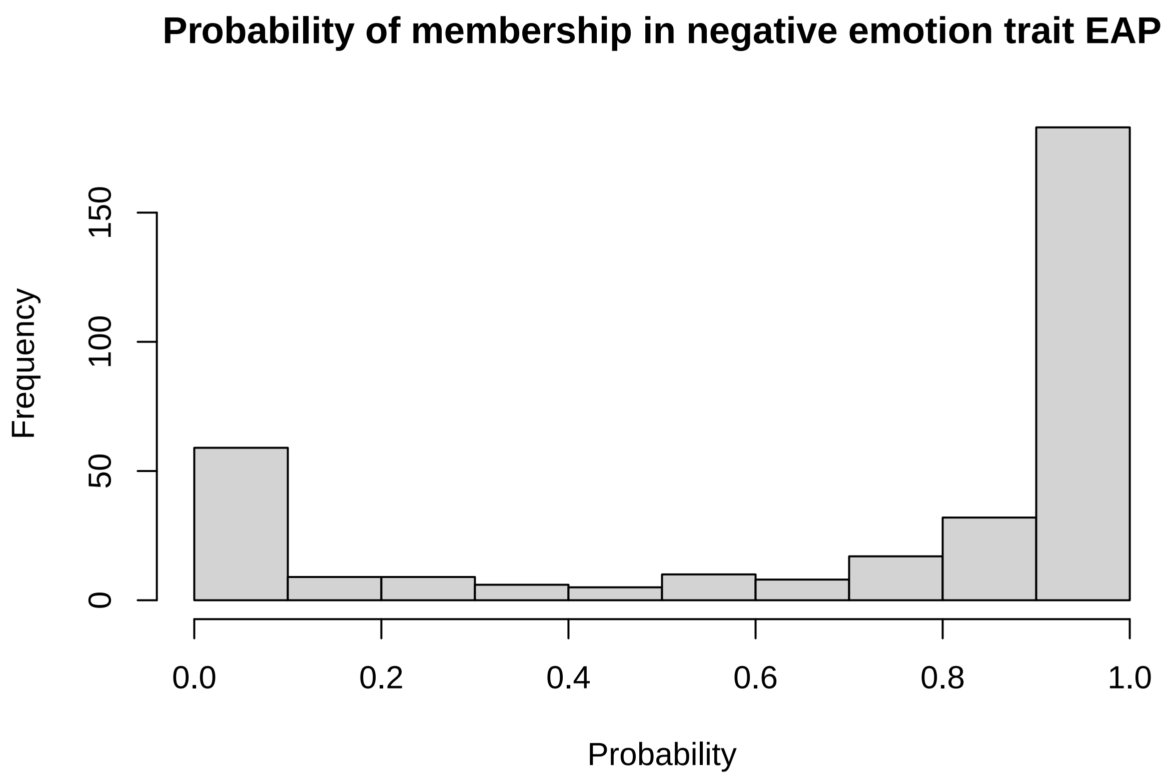
```r
data_Trait_mclust$loglik # Log likelihood
```

```
## [1] -2860.812
```

```r
# combine data frames
data_Trait_stan <- cbind(data_Trait_stan,data_Trait_mclust$classification,
                         data_Trait_mclust$z) # for barplot
names(data_Trait_stan) <- c(names(data_Trait_stan[-c(9:11)]),"Trait", "prob1","prob2")
data_Trait <- merge(data,data_Trait_stan[c(1,c(9:11))], by = "uid", all.x = TRUE) # for
# Distribution of membership probability
hist(data_Trait$prob1, main = "Probability of membership in moderate trait EAP",
     xlab = "Probability")
```




```r
hist(data_Trait$prob2, main = "Probability of membership in negative emotion trait EAP",
     xlab = "Probability")
```




```
######## Means and SDs
P1 <- na.omit(data_Trait[data_Trait$Trait==1,c(54,55,56,62,59,60,51)])
P2 <- na.omit(data_Trait[data_Trait$Trait==2,c(54,55,56,62,59,60,51)])
apply(P1, FUN=mean, 2)
```

```
##    AEQ_Ang_mean      AMAS_mean    AEQ_Bor_mean AEQ_Enj_3_mean      SELFC_mean
##        3.213068       2.450284       3.159091       1.867424       2.179545
##        VAL_mean          Grade
##        2.471591       6.122386
```

```
apply(P1, FUN=sd, 2)
```

```
##    AEQ_Ang_mean      AMAS_mean    AEQ_Bor_mean AEQ_Enj_3_mean      SELFC_mean
##       0.9734550      0.9784936      1.0390797      0.9820090       0.8425685
##        VAL_mean          Grade
##       0.8445496      1.2075119
```

```
apply(P2, FUN=mean, 2)
```

```
##    AEQ_Ang_mean      AMAS_mean    AEQ_Bor_mean AEQ_Enj_3_mean      SELFC_mean
##        1.656000       1.753500       2.158400       2.321333       2.732000
##        VAL_mean          Grade
##        3.026000       7.190520
```

```
apply(P2, FUN=sd, 2)
```

```
##    AEQ_Ang_mean      AMAS_mean    AEQ_Bor_mean AEQ_Enj_3_mean      SELFC_mean
##       0.5120672      0.5174420      0.7280037      0.8106487       0.6442953
##        VAL_mean          Grade
##       0.6207781      1.1163992
```

```
# Test differences between profiles
summary(aov(AEQ_Ang_mean ~ Trait, data = data_Trait))
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)    
## Trait         1  157.8  157.81   358.9 <2e-16 ***
## Residuals   336  147.7    0.44                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 9 observations deleted due to missingness
```

```
summary(aov(AMAS_mean ~ Trait, data = data_Trait))
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)    
## Trait         1   31.6  31.601    70.8 1.15e-15 ***
## Residuals   336  150.0   0.446                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 9 observations deleted due to missingness
```

```
summary(aov(AEQ_Bor_mean ~ Trait, data = data_Trait))
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)    
## Trait         1  65.18   65.18   96.95 <2e-16 ***
## Residuals   336 225.90    0.67                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 9 observations deleted due to missingness
```

```
summary(aov(AEQ_Enj_3_mean ~ Trait, data = data_Trait))
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)    
## Trait         1  13.41  13.410    18.2 2.58e-05 ***
## Residuals   336 247.53   0.737                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 9 observations deleted due to missingness
```

```
summary(aov(SELFC_mean ~ Trait, data = data_Trait))
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)    
## Trait         1  19.87  19.865   40.42 6.67e-10 ***
## Residuals   336 165.13   0.491                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 9 observations deleted due to missingness
```

```
summary(aov(VAL_mean ~ Trait, data = data_Trait))
```

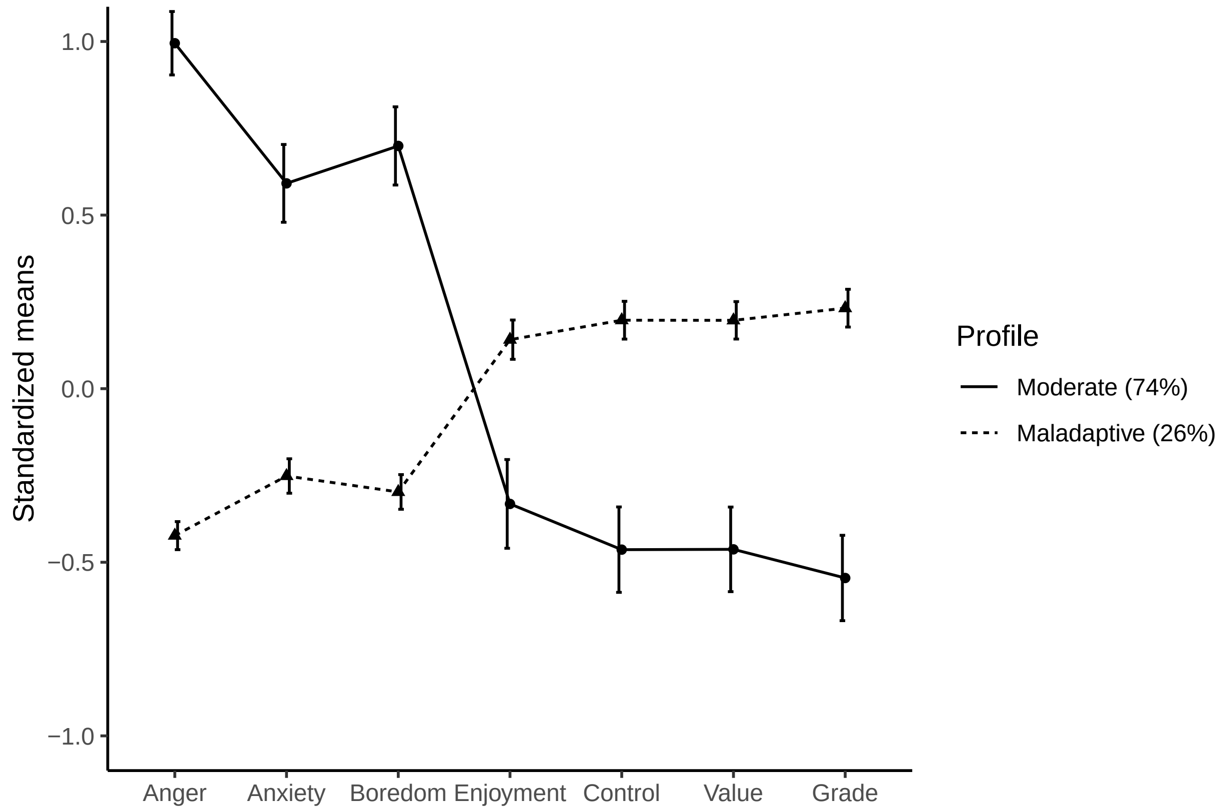
```
##              Df Sum Sq Mean Sq F value   Pr(>F)    
## Trait         1  20.01   20.01   42.54 2.54e-10 ***
## Residuals   336 158.01    0.47                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 9 observations deleted due to missingness
```

```
summary(aov(Grade ~ Trait, data = data_Trait))
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)
```

```
## Trait         1   74.3   74.26   57.07 4.02e-13 ***
## Residuals   336  437.2    1.30
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 9 observations deleted due to missingness
```

```
############ Line plot - Figure 2
# Save estimated means & variances
means <- as.data.frame(data_Trait_mclust$parameters$mean)
means$Var <- c( "Anger", "Anxiety", "Boredom", "Enjoyment","Control",
                        "Value", "Grade")
names(means) <- c("1", "2")
data_Trait_table_long <- gather(means, Profile, measurement, 1:2, factor_key=TRUE)
names(data_Trait_table_long) <- c("Var","Profile","Measurement")
data_Trait_table_long$order <- c(rep(c(1:7),2))
data_Trait_table_long <- data_Trait_table_long[order(data_Trait_table_long$order),]
data_Trait_table_long$Profile <- as.factor(data_Trait_table_long$Profile)
data_Trait_table_long$Var <- as.factor(data_Trait_table_long$Var)
data_Trait_table_long$Var <- factor(data_Trait_table_long$Var, levels = c("Anger", "Anxiety", "Boredom"
                                                                          "Enjoyment","Control"
                                                                          "Value", "Grade"))
stan_mean <- data_Trait_mclust$parameters$mean
stan_var1 <- diag(data_Trait_mclust$parameters$variance$sigma[,,1])
stan_var2 <- diag(data_Trait_mclust$parameters$variance$sigma[,,2])
SE_var1 <- sqrt(stan_var1)/sqrt(data_Trait_mclust$parameters$pro[1]*nrow(data_Trait_stan))
SE_var2 <- sqrt(stan_var2)/sqrt(data_Trait_mclust$parameters$pro[2]*nrow(data_Trait_stan))
SE_var1$Profile <- 1
SE_var2$Profile <- 2
SE_var1 <- unlist(SE_var1)
SE_var2 <- unlist(SE_var2)
SE_both <- data.frame(rbind(SE_var1,SE_var2))
data_Trait_SE_long <- gather(SE_both, variable, measurement, AEQ_Ang_mean:Grade,
                                factor_key=TRUE)
data_Trait_table_long <- cbind(data_Trait_table_long[c(1:3)], data_Trait_SE_long[3])
names(data_Trait_table_long) <- c("Var", "Profile", "mean","se")
levels(data_Trait_table_long$Profile)[levels(data_Trait_table_long$Profile)=="1"] <- "Moderate (74%)"
levels(data_Trait_table_long$Profile)[levels(data_Trait_table_long$Profile)=="2"] <- "Maladaptive (26%)
# FIGURE 2
pd <- position_dodge(0.1)
ggplot(data = data_Trait_table_long, mapping = ggplot2::aes(x=Var,y=mean,group=Profile)) +
  geom_errorbar(aes(ymin=mean-se, ymax=mean+se), colour="black", width=.1, position=pd) +
  geom_line(aes(linetype = Profile))+
  geom_point(aes(shape = Profile)) +
  ylab("Standardized means") + xlab("") +
  scale_shape_discrete(name = "Profile", breaks = c("Negative", "Positive"), labels = c("Negative", "Po
  theme_classic() + coord_cartesian( ylim = c(-1,1))
```




```
# Cluster analysis with Outliers removed
dataOutRm_Trait_stan <- na.omit(data_Trait_OutRm)
dataOutRm_Trait_stan[c(2:5,7:8,10)] <- standardize(dataOutRm_Trait_stan[c(2:5,7:8,10)]) # select variab
dataOutRm_Trait_mclust <- Mclust(dataOutRm_Trait_stan[c(2:5,7:8,10)])
summary(dataOutRm_Trait_mclust, parameters = TRUE)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VVE (ellipsoidal, equal orientation) model with 2 components:
##
##  log-likelihood   n df       BIC       ICL
##       -2575.357 295 50 -5435.063 -5503.312
##
## Clustering table:
##   1   2
## 170 125
##
## Mixing probabilities:
##         1         2
## 0.5788133 0.4211867
##
## Means:
##                        [,1]       [,2]
## AEQ_Ang_mean    0.5163433 -0.7095818
## AMAS_mean       0.2398712 -0.3296416
## AEQ_Bor_mean    0.5025437 -0.6906178
```

```
## AEQ_Enj_3_mean -0.4193118  0.5762368
## SELFC_mean     -0.2327980  0.3199214
## VAL_mean       -0.1876821  0.2579210
## Grade          -0.1123611  0.1544116
##
## Variances:
## [,,1]
##                AEQ_Ang_mean   AMAS_mean AEQ_Bor_mean AEQ_Enj_3_mean SELFC_mean
## AEQ_Ang_mean     0.80758200  0.07332047   0.21667567    -0.12816586 -0.1247062
## AMAS_mean        0.07332047  1.07726660  -0.08221244    -0.09404029 -0.4995552
## AEQ_Bor_mean     0.21667567 -0.08221244   0.97927585    -0.17603046  0.1217794
## AEQ_Enj_3_mean  -0.12816586 -0.09404029  -0.17603046     0.61369699  0.2884912
## SELFC_mean      -0.12470623 -0.49955523   0.12177942     0.28849115  1.0132355
## VAL_mean        -0.08398817 -0.10454192  -0.00977680     0.13977837  0.2604114
## Grade           -0.16287919 -0.26186629  -0.12842590     0.30948688  0.6746238
##                   VAL_mean      Grade
## AEQ_Ang_mean   -0.08398817 -0.1628792
## AMAS_mean      -0.10454192 -0.2618663
## AEQ_Bor_mean   -0.00977680 -0.1284259
## AEQ_Enj_3_mean  0.13977837  0.3094869
## SELFC_mean      0.26041142  0.6746238
## VAL_mean        1.02374523  0.1784259
## Grade           0.17842589  1.1297012
## [,,2]
##                AEQ_Ang_mean   AMAS_mean AEQ_Bor_mean AEQ_Enj_3_mean
## AEQ_Ang_mean     0.27229776  0.15559367  0.109722814    -0.07991172
## AMAS_mean        0.15559367  0.62963765  0.023124992    -0.14171077
## AEQ_Bor_mean     0.10972281  0.02312499  0.334520991     0.03961407
## AEQ_Enj_3_mean  -0.07991172 -0.14171077  0.039614067     0.81545683
## SELFC_mean      -0.15466188 -0.43078991 -0.004980236     0.32281404
## VAL_mean        -0.06874501 -0.04901988  0.045613068     0.06852544
## Grade           -0.17271310 -0.38629534 -0.105854334     0.05341040
##                  SELFC_mean    VAL_mean      Grade
## AEQ_Ang_mean   -0.154661879 -0.06874501 -0.1727131
## AMAS_mean      -0.430789909 -0.04901988 -0.3862953
## AEQ_Bor_mean   -0.004980236  0.04561307 -0.1058543
## AEQ_Enj_3_mean  0.322814041  0.06852544  0.0534104
## SELFC_mean      0.804092206  0.24006926  0.5717426
## VAL_mean        0.240069262  0.86597723  0.1974921
## Grade           0.571742640  0.19749211  0.9219488
```

```
dataOutRm_Trait_mclust$BIC
```

```
## Bayesian Information Criterion (BIC):
##         EII       VII       EEI       VEI       EVI       VVI       EEE
## 1 -5898.700 -5898.700 -5932.822 -5932.822 -5932.822 -5932.822 -5453.340
## 2 -5651.233 -5632.737 -5665.812 -5646.136 -5661.497 -5637.055 -5446.688
## 3 -5631.751 -5611.857 -5624.266 -5602.488 -5617.239 -5591.215 -5471.372
## 4 -5586.232 -5583.614 -5581.954 -5571.025 -5579.335 -5564.491 -5466.560
## 5 -5594.744 -5579.770 -5573.849 -5546.287 -5606.264 -5592.847 -5493.262
## 6 -5616.544 -5592.837 -5595.741 -5565.823 -5660.854 -5629.482 -5521.311
## 7 -5631.576 -5615.756 -5593.924 -5558.155 -5635.049 -5673.673 -5549.162
## 8 -5651.275 -5637.356 -5569.245 -5578.027 -5677.943        NA -5599.890
## 9 -5656.753 -5658.921 -5597.380 -5616.747 -5730.921        NA -5608.773
##         VEE       EVE       VVE       EEV       VEV       EVV       VVV
```

```
## 1 -5453.340 -5453.340 -5453.340 -5453.340 -5453.340 -5453.340 -5453.340
## 2 -5445.691 -5465.607 -5435.063 -5496.850 -5497.382 -5542.774 -5515.052
## 3 -5474.956 -5502.235 -5481.443 -5617.091 -5593.540 -5671.501 -5650.973
## 4 -5482.519 -5513.134        NA -5696.144 -5712.278 -5791.268 -5776.988
## 5 -5505.451 -5563.083        NA -5813.864 -5809.641 -5940.153        NA
## 6 -5541.993 -5652.115 -5603.795 -5924.356 -5943.518 -6082.329 -5992.207
## 7 -5579.176 -5722.336 -5658.849 -6051.464 -6062.052 -6155.488 -6251.270
## 8 -5618.453 -5759.563 -5715.784 -6136.064 -6165.837 -6343.440 -6323.714
## 9 -5647.735 -5809.019 -5761.298 -6211.441 -6258.863 -6383.974 -6461.098
##
## Top 3 models based on the BIC criterion:
##     VVE,2     VEE,2     EEE,2
## -5435.063 -5445.691 -5446.688
```

```r
dataOutRm_Trait_mclust$loglik # Log likelihood
```

```
## [1] -2575.357
```

```r
round(dataOutRm_Trait_mclust$parameters$mean,5)
```

```
##                    [,1]     [,2]
## AEQ_Ang_mean    0.51634 -0.70958
## AMAS_mean       0.23987 -0.32964
## AEQ_Bor_mean    0.50254 -0.69062
## AEQ_Enj_3_mean -0.41931  0.57624
## SELFC_mean     -0.23280  0.31992
## VAL_mean       -0.18768  0.25792
## Grade          -0.11236  0.15441
```

```r
# combine data frames
dataOutRm_Trait_stan <- cbind(dataOutRm_Trait_stan,dataOutRm_Trait_mclust$classification) #
names(dataOutRm_Trait_stan) <- c(names(dataOutRm_Trait_stan[-c(11)]),"Trait_Out")
dataOutRm_Trait <- merge(data_Trait,dataOutRm_Trait_stan[c(1,c(11))], by = "uid", all.x = TRUE) # for
# table of both
table(dataOutRm_Trait$Trait, dataOutRm_Trait$Trait_Out)
```

```
##
##       1   2
##   1  47   5
##   2 123 120
```

```r
chisq.test(dataOutRm_Trait$Trait, dataOutRm_Trait$Trait_Out)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  dataOutRm_Trait$Trait and dataOutRm_Trait$Trait_Out
## X-squared = 26.137, df = 1, p-value = 3.181e-07
```

```r
CramerV(dataOutRm_Trait$Trait, dataOutRm_Trait$Trait_Out, conf.level = TRUE)
```

```
##  Cramer V    lwr.ci    upr.ci
## 0.3066562 0.0000000 1.0000000
```

```r
# save
data_Trait_profile <- data_Trait
save(data_Trait_profile, file = "data_Trait_profile.Rda")
```

Profiles - State

```r
load("data_State_nC_usable.Rda")
data <- data_State_nC_usable
########
#data_State
data_State_stan_nC_L2 <- na.omit(data[c(1,3:8,10)])
data_State_stan_nC_L2[c(2:8)] <- standardize(data_State_stan_nC_L2[c(2:8)])
data_State_mclust_nC_L2 <- Mclust(data_State_stan_nC_L2[c(2:8)])
summary(data_State_mclust_nC_L2, parameters = TRUE) # chosen model
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VVE (ellipsoidal, equal orientation) model with 4 components:
##
##  log-likelihood   n df       BIC       ICL
##       -2627.611 332 80 -5719.633 -5795.924
##
## Clustering table:
##   1   2   3   4
## 103 123  48  58
##
## Mixing probabilities:
##         1         2         3         4
## 0.2964167 0.3717609 0.1600481 0.1717743
##
## Means:
##                          [,1]        [,2]       [,3]       [,4]
## ang_mean           -0.7981744 -0.21576590  1.4233495  0.5181284
## anx_mean           -0.5781341  0.21832257  1.1602984 -0.5559548
## bor_mean           -0.5781298 -0.17114377  0.3059742  1.0829404
## enj_mean            0.7199075  0.28852284 -0.8173857 -1.1051309
## cont                0.2721173  0.01639334 -0.4167264 -0.1167708
## val_mean            0.4785532  0.32829701 -0.4423517 -1.1241589
## initial_rt_correct2 -0.4338361  0.10357038  0.4214228  0.1318295
##
## Variances:
## [,,1]
##                         ang_mean     anx_mean     bor_mean     enj_mean
## ang_mean             0.020551520  0.016923948 -0.020211675  0.009498774
## anx_mean             0.016923948  0.108029312 -0.072602713  0.009175055
## bor_mean            -0.020211675 -0.072602713  0.824139121 -0.498861393
## enj_mean             0.009498774  0.009175055 -0.498861393  0.640247047
## cont                -0.028713560 -0.072894915  0.058408905  0.002674475
## val_mean             0.005922408  0.009083437 -0.633820560  0.553246269
## initial_rt_correct2 -0.023245659  0.016369598 -0.005389494 -0.019640661
##                             cont     val_mean initial_rt_correct2
## ang_mean            -0.028713560  0.005922408        -0.023245659
## anx_mean            -0.072894915  0.009083437         0.016369598
## bor_mean             0.058408905 -0.633820560        -0.005389494
## enj_mean             0.002674475  0.553246269        -0.019640661
## cont                 0.822862820  0.061012795        -0.091595808
```

```
## val_mean             0.061012795  0.933871551        -0.016584279
## initial_rt_correct2 -0.091595808 -0.016584279         0.417371410
## [,,2]
##                        ang_mean    anx_mean    bor_mean      enj_mean
## ang_mean             0.20275820  0.07995233  0.01284629  0.0151652470
## anx_mean             0.07995233  0.66610547  0.09376102  0.0545673619
## bor_mean             0.01284629  0.09376102  0.53048966 -0.1941904318
## enj_mean             0.01516525  0.05456736 -0.19419043  0.4551397029
## cont                -0.02125471 -0.01060129  0.04754914 -0.0002228069
## val_mean             0.00850356  0.05586288 -0.25468015  0.2166005821
## initial_rt_correct2 -0.06108730  0.01174487 -0.01748405 -0.0445451464
##                              cont    val_mean initial_rt_correct2
## ang_mean            -0.0212547097  0.00850356         -0.06108730
## anx_mean            -0.0106012890  0.05586288          0.01174487
## bor_mean             0.0475491422 -0.25468015         -0.01748405
## enj_mean            -0.0002228069  0.21660058         -0.04454515
## cont                 0.6641451491  0.04797030          0.08041960
## val_mean             0.0479703050  0.56469016          0.01977202
## initial_rt_correct2  0.0804195960  0.01977202          1.03861902
## [,,3]
##                        ang_mean   anx_mean    bor_mean     enj_mean
## ang_mean             1.009855471 0.19397382  0.05928517  0.029327290
## anx_mean             0.193973820 2.15170489  0.36954340  0.188849145
## bor_mean             0.059285172 0.36954340  1.02517697 -0.171985202
## enj_mean             0.029327290 0.18884915 -0.17198520  0.546044936
## cont                -0.001376785 0.03123744  0.10471320 -0.006742055
## val_mean             0.028677031 0.15250224 -0.18423621  0.457938952
## initial_rt_correct2  0.002085596 0.01408363 -0.01922084 -0.044189030
##                             cont    val_mean initial_rt_correct2
## ang_mean            -0.001376785  0.02867703         0.002085596
## anx_mean             0.031237441  0.15250224         0.014083628
## bor_mean             0.104713196 -0.18423621        -0.019220841
## enj_mean            -0.006742055  0.45793895        -0.044189030
## cont                 1.396039710  0.09776671        -0.086771257
## val_mean             0.097766710  0.82086909         0.010510229
## initial_rt_correct2 -0.086771257  0.01051023         1.009983539
## [,,4]
##                        ang_mean     anx_mean     bor_mean     enj_mean
## ang_mean             0.951024473 -0.137880524 -0.002758093 -0.003494396
## anx_mean            -0.137880524  0.133964717 -0.037748703 -0.000549176
## bor_mean            -0.002758093 -0.037748703  0.251543628 -0.068591870
## enj_mean            -0.003494396 -0.000549176 -0.068591870  0.159327262
## cont                -0.035461388 -0.140188482  0.130999340 -0.010267343
## val_mean            -0.003422259 -0.019147654 -0.069182477  0.111471300
## initial_rt_correct2 -0.021508471  0.022941809 -0.046245525 -0.067441900
##                            cont     val_mean initial_rt_correct2
## ang_mean            -0.03546139 -0.003422259         -0.02150847
## anx_mean            -0.14018848 -0.019147654          0.02294181
## bor_mean             0.13099934 -0.069182477         -0.04624553
## enj_mean            -0.01026734  0.111471300         -0.06744190
## cont                 1.40085150  0.127418411         -0.02021262
## val_mean             0.12741841  0.236224904          0.02629212
## initial_rt_correct2 -0.02021262  0.026292124          1.33054091
```

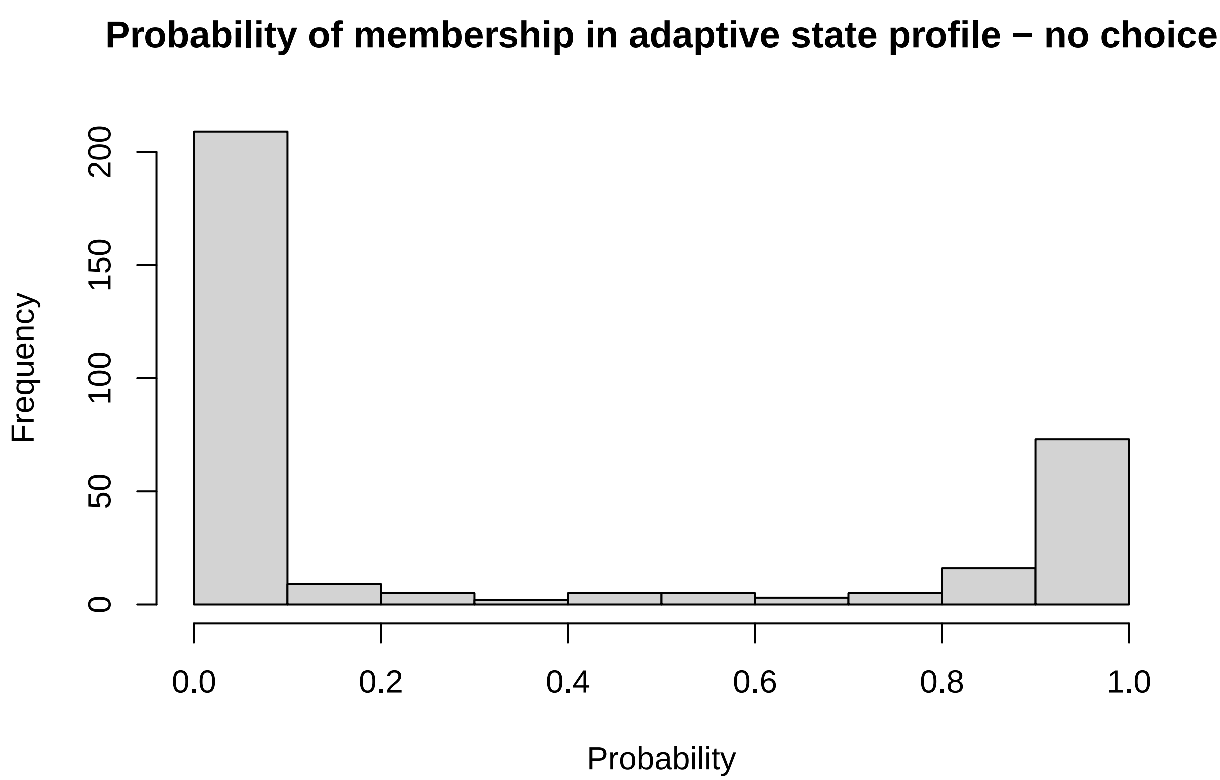
```r
mclustBIC(data_State_stan_nC_L2[c(2:8)]) # BIC fit of top three models
```

```
## Bayesian Information Criterion (BIC):
##         EII       VII       EEI       VEI       EVI       VVI       EEE
## 1 -6634.657 -6634.657 -6669.488 -6669.488 -6669.488 -6669.488 -6031.991
## 2 -6338.279 -6253.071 -6257.319 -6209.056 -6164.415 -6064.094 -5943.074
## 3 -6242.712 -6141.835 -6138.482 -6071.342 -6004.256 -5881.224 -5962.746
## 4 -6221.109 -6053.906 -6074.971 -5959.401 -5875.535 -5813.896 -5895.729
## 5 -6184.362 -6034.291 -6011.046 -5884.119 -5871.455 -5771.021 -5924.863
## 6 -6142.983 -5997.737 -6051.655 -5881.674 -5878.757 -5770.699 -5965.080
## 7 -6148.134 -5980.746 -6020.907 -5878.908 -5888.680 -5775.756 -5986.374
## 8 -6169.060 -5993.775 -6051.719 -5915.852 -5872.824 -5780.692 -5988.636
## 9 -6108.482 -5987.108 -5967.644 -5856.360 -5901.169 -5802.278 -5915.968
##         VEE       EVE       VVE       EEV       VEV       EVV       VVV
## 1 -6031.991 -6031.991 -6031.991 -6031.991 -6031.991 -6031.991 -6031.991
## 2 -5902.755 -5933.912 -5762.271 -5916.870 -5807.868 -5948.162 -5790.140
## 3 -5919.341 -5869.463 -5744.150 -5854.457 -5834.074 -5960.149 -5825.659
## 4 -5875.050 -5779.037 -5719.633 -5884.113 -5895.611 -5965.652 -5901.841
## 5 -5843.812 -5784.982 -5755.400 -5939.182 -5953.672 -6048.886 -6036.067
## 6 -5878.388 -5903.265 -5931.868 -6015.591 -6046.346 -6150.866 -6191.474
## 7 -5907.073 -5872.307 -5785.080 -6134.808 -6145.781 -6285.667 -6313.733
## 8 -5918.049 -5888.060 -5964.716 -6234.812 -6284.085 -6440.531 -6459.962
## 9 -5847.391 -5911.899 -5990.789 -6339.057 -6351.823 -6590.710 -6592.528
##
## Top 3 models based on the BIC criterion:
##     VVE,4     VVE,3     VVE,5
## -5719.633 -5744.150 -5755.400
```

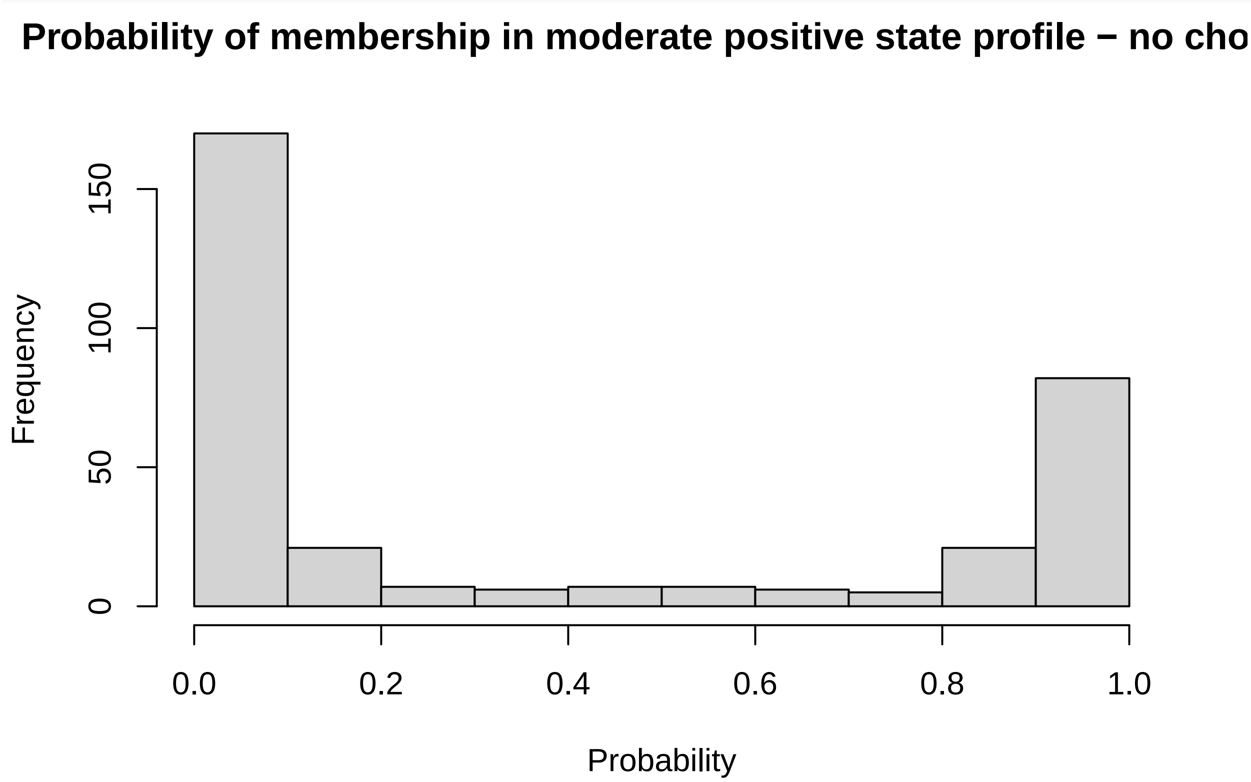
```r
data_State_mclust_nC_L2$loglik # Log likelihood
```

```
## [1] -2627.611
```

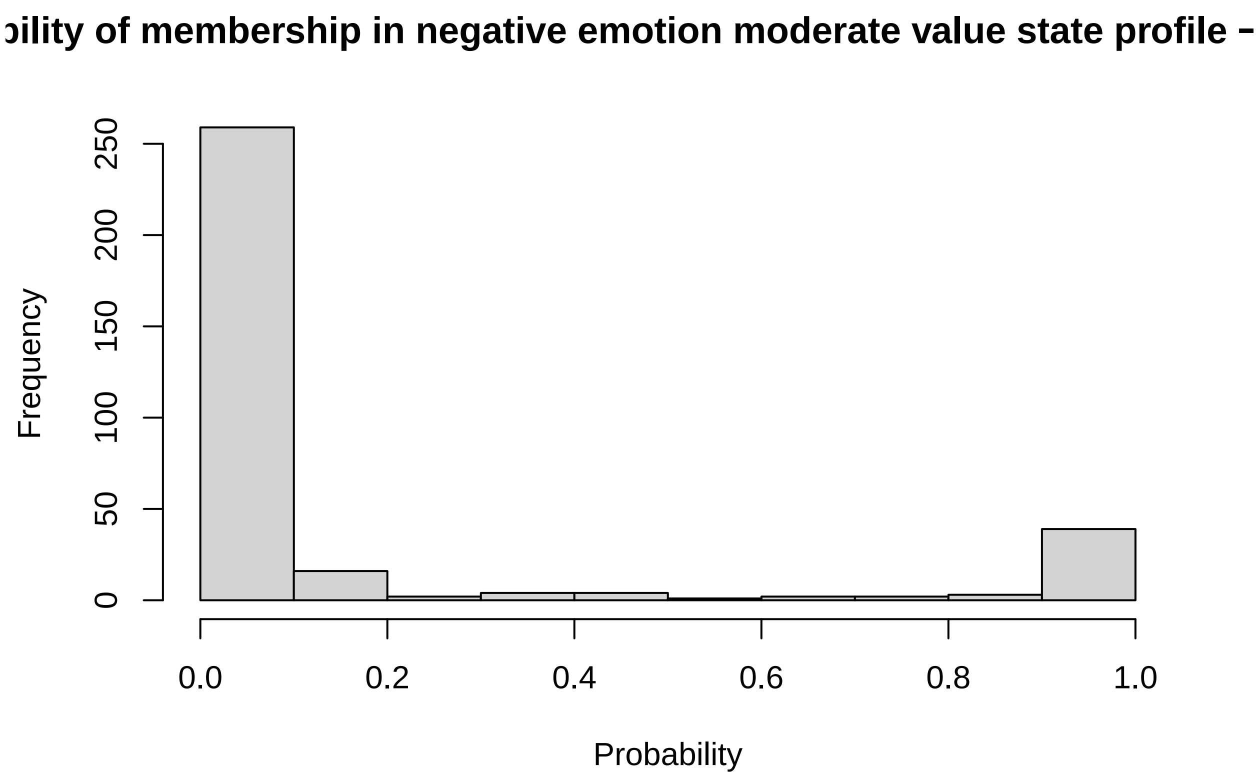
```r
# SAVE
data_State_stan_nC_L2 <- cbind(data_State_stan_nC_L2,data_State_mclust_nC_L2$classification, data_State_
names(data_State_stan_nC_L2) <- c(names(data_State_stan_nC_L2[-c(9:13)]),"State_nC_L2","Stat_L2_p1","St
                                  "Stat_L2_p4")
data_State_nC_L2 <- merge(data,data_State_stan_nC_L2[c(1,c(9:13))], by = "uid", all.x = TRUE) # for tab
# Distribution of membership probability
hist(data_State_nC_L2$Stat_L2_p1, main = "Probability of membership in adaptive state profile - no choi
     xlab = "Probability")
```



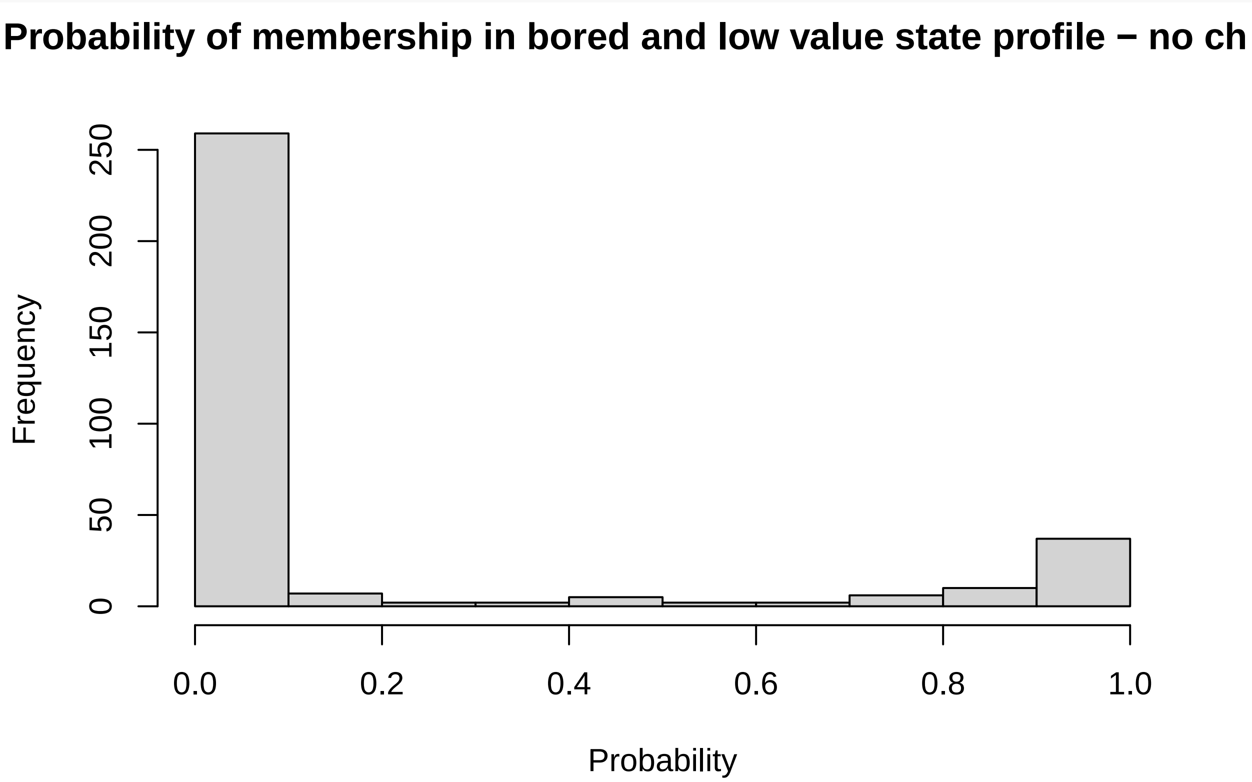

```
hist(data_State_nC_L2$Stat_L2_p2, main = "Probability of membership in moderate positive state profile -
     xlab = "Probability")
```




```
hist(data_State_nC_L2$Stat_L2_p3, main = "Probability of membership in negative emotion moderate value s
     xlab = "Probability")
```




```r
hist(data_State_nC_L2$Stat_L2_p4, main = "Probability of membership in bored and low value state profil
     xlab = "Probability")
```




```r
######## Means and SDs
Ang_mean <- aggregate(ang_mean ~ State_nC_L2, mean, data = data_State_nC_L2)
```

```r
Anx_mean <- aggregate(anx_mean ~ State_nC_L2, mean, data = data_State_nC_L2)
Bor_mean <- aggregate(bor_mean ~ State_nC_L2, mean, data = data_State_nC_L2)
Enj_mean <- aggregate(enj_mean ~ State_nC_L2, mean, data = data_State_nC_L2)
Con_mean <- aggregate(cont ~ State_nC_L2, mean, data = data_State_nC_L2)
Val_mean <- aggregate(val_mean ~ State_nC_L2, mean, data = data_State_nC_L2)
rt_mean <- aggregate(initial_rt_correct2 ~ State_nC_L2, mean, data = data_State_nC_L2)
data_State_table <- cbind(Ang_mean,Anx_mean[,2],Bor_mean[,2],Enj_mean[,2],Con_mean[,2],Val_mean[,2],rt_m
names(data_State_table) <- c("Profile", "Anger", "Anxiety", "Boredom", "Enjoyment", "Control","Value","
data_State_table_long <- gather(data_State_table, variable, measurement, Anger:rt, factor_key=TRUE)
########  SDs
Ang_sd <- aggregate(ang_mean ~ State_nC_L2, sd, data = data_State_nC_L2)
Anx_sd <- aggregate(anx_mean ~ State_nC_L2, sd, data = data_State_nC_L2)
Bor_sd <- aggregate(bor_mean ~ State_nC_L2, sd, data = data_State_nC_L2)
Enj_sd <- aggregate(enj_mean ~ State_nC_L2, sd, data = data_State_nC_L2)
Con_sd <- aggregate(cont ~ State_nC_L2, sd, data = data_State_nC_L2)
Val_sd <- aggregate(val_mean ~ State_nC_L2, sd, data = data_State_nC_L2)
rt_sd <- aggregate(initial_rt_correct2 ~ State_nC_L2, sd, data = data_State_nC_L2)
data_State_table_sd <- cbind(Ang_sd,Anx_sd[,2],Bor_sd[,2],Enj_sd[,2],Con_sd[,2],Val_sd[,2],rt_sd[,2])
names(data_State_table_sd) <- c("Profile", "Anger", "Anxiety", "Boredom", "Enjoyment", "Control","Value
data_State_table_sd_long <- gather(data_State_table_sd, variable, measurement, Anger:rt,
                                    factor_key=TRUE)
data_State_nC_L2$State_nC_L2 <- as.factor(data_State_nC_L2$State_nC_L2)
# Test differences between profiles
a_ang <- aov(ang_mean ~ State_nC_L2, data = data_State_nC_L2)
summary(a_ang)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## State_nC_L2   3 12.317   4.106     173 <2e-16 ***
## Residuals   328  7.785   0.024
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 15 observations deleted due to missingness
```

```r
TukeyHSD(a_ang)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = ang_mean ~ State_nC_L2, data = data_State_nC_L2)
##
## $State_nC_L2
##           diff         lwr        upr p adj
## 2-1  0.1523103  0.09917537  0.2054452     0
## 3-1  0.5774421  0.50791624  0.6469679     0
## 4-1  0.3364414  0.27113179  0.4017510     0
## 3-2  0.4251318  0.35742663  0.4928369     0
## 4-2  0.1841311  0.12076320  0.2474991     0
## 4-3 -0.2410006 -0.31862806 -0.1633732     0
```

```r
a_anx <- aov(anx_mean ~ State_nC_L2, data = data_State_nC_L2)
summary(a_anx)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## State_nC_L2   3  4.370  1.4566   75.61 <2e-16 ***
## Residuals   328  6.319  0.0193
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 15 observations deleted due to missingness
```

```r
TukeyHSD(a_anx)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = anx_mean ~ State_nC_L2, data = data_State_nC_L2)
##
## $State_nC_L2
##             diff         lwr         upr     p adj
## 2-1  0.156207011  0.10833590  0.20407812 0.0000000
## 3-1  0.324809223  0.26217097  0.38744748 0.0000000
## 4-1  0.003923423 -0.05491632  0.06276316 0.9981828
## 3-2  0.168602212  0.10760425  0.22960017 0.0000000
## 4-2 -0.152283588 -0.20937400 -0.09519318 0.0000000
## 4-3 -0.320885800 -0.39082308 -0.25094852 0.0000000
```

```r
a_bor <- aov(bor_mean ~ State_nC_L2, data = data_State_nC_L2)
summary(a_bor)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## State_nC_L2   3  7.965  2.6549   54.36 <2e-16 ***
## Residuals   328 16.020  0.0488
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 15 observations deleted due to missingness
```

```r
TukeyHSD(a_bor)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = bor_mean ~ State_nC_L2, data = data_State_nC_L2)
##
## $State_nC_L2
##          diff        lwr       upr     p adj
## 2-1 0.1111106 0.03488940 0.1873318 0.0011299
## 3-1 0.2334109 0.13367723 0.3331446 0.0000000
## 4-1 0.4474144 0.35372873 0.5411000 0.0000000
## 3-2 0.1223003 0.02517836 0.2194223 0.0069087
## 4-2 0.3363038 0.24540347 0.4272041 0.0000000
## 4-3 0.2140034 0.10264814 0.3253588 0.0000067
```

```r
a_enj <- aov(enj_mean ~ State_nC_L2, data = data_State_nC_L2)
summary(a_enj)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## State_nC_L2   3  8.886  2.9619   112.7 <2e-16 ***
## Residuals   328  8.622  0.0263
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 15 observations deleted due to missingness
```

```r
TukeyHSD(a_enj)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = enj_mean ~ State_nC_L2, data = data_State_nC_L2)
##
## $State_nC_L2
##            diff        lwr         upr     p adj
## 2-1 -0.09323915 -0.1491567 -0.03732155 0.0001290
## 3-1 -0.35737815 -0.4305451 -0.28421124 0.0000000
## 4-1 -0.41700923 -0.4857391 -0.34827932 0.0000000
## 3-2 -0.26413901 -0.3353899 -0.19288810 0.0000000
## 4-2 -0.32377009 -0.3904566 -0.25708354 0.0000000
## 4-3 -0.05963108 -0.1413239  0.02206172 0.2365193
```

```r
a_con <- aov(cont ~ State_nC_L2, data = data_State_nC_L2)
summary(a_con)
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)
## State_nC_L2   3  0.921 0.30695   5.556 0.000993 ***
## Residuals   328 18.122 0.05525
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 15 observations deleted due to missingness
```

```r
TukeyHSD(a_con)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = cont ~ State_nC_L2, data = data_State_nC_L2)
##
## $State_nC_L2
##            diff         lwr           upr     p adj
## 2-1 -0.05764158 -0.13870846  0.0234252880 0.2582961
## 3-1 -0.16167437 -0.26774853 -0.0556002079 0.0005842
## 4-1 -0.09029532 -0.18993692  0.0093462830 0.0912010
## 3-2 -0.10403279 -0.20732920 -0.0007363713 0.0476444
## 4-2 -0.03265373 -0.12933294  0.0640254782 0.8192585
## 4-3  0.07137905 -0.04705557  0.1898136750 0.4052232
```

```r
a_val <- aov(val_mean ~ State_nC_L2, data = data_State_nC_L2)
summary(a_val)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## State_nC_L2   3  6.017  2.0057   62.78 <2e-16 ***
## Residuals   328 10.479  0.0319
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 15 observations deleted due to missingness
```

```r
TukeyHSD(a_val)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
```

```
## Fit: aov(formula = val_mean ~ State_nC_L2, data = data_State_nC_L2)
##
## $State_nC_L2
##            diff         lwr         upr     p adj
## 2-1 -0.03053682 -0.09218341  0.03110977 0.5768910
## 3-1 -0.20777738 -0.28844054 -0.12711422 0.0000000
## 4-1 -0.35732941 -0.43310099 -0.28155783 0.0000000
## 3-2 -0.17724056 -0.25579141 -0.09868971 0.0000001
## 4-2 -0.32679259 -0.40031145 -0.25327374 0.0000000
## 4-3 -0.14955203 -0.23961459 -0.05948946 0.0001391
```

```
a_rt <- aov(initial_rt_correct2 ~ State_nC_L2, data = data_State_nC_L2)
summary(a_rt)
```

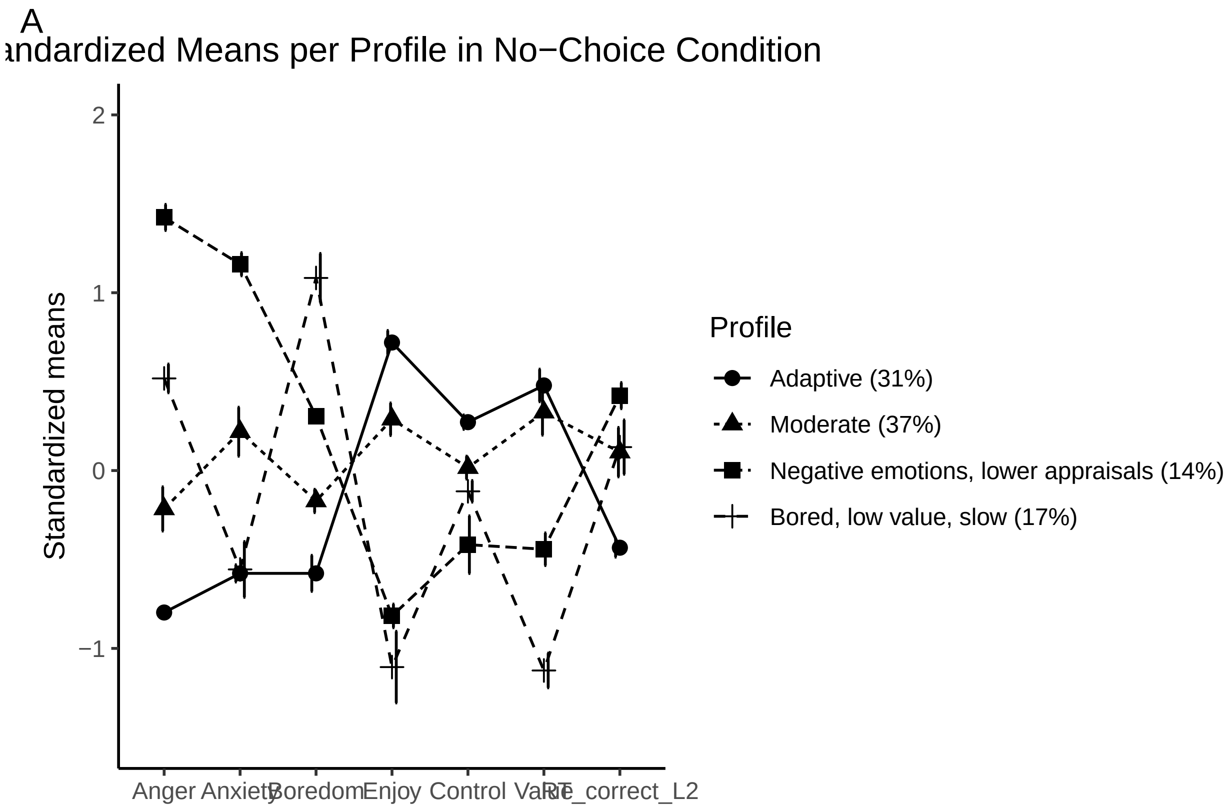
```
##              Df    Sum Sq   Mean Sq F value   Pr(>F)
## State_nC_L2   3 2.407e+09 802484573   12.15 1.46e-07 ***
## Residuals   328 2.166e+10  66025072
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 15 observations deleted due to missingness
```

```
TukeyHSD(a_rt)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = initial_rt_correct2 ~ State_nC_L2, data = data_State_nC_L2)
##
## $State_nC_L2
##           diff         lwr       upr     p adj
## 2-1  4286.9479  1484.51198  7089.384 0.0005522
## 3-1  7942.5419  4275.61811 11609.466 0.0000003
## 4-1  4973.2152  1528.66145  8417.769 0.0012934
## 3-2  3655.5940    84.69537  7226.493 0.0425241
## 4-2   686.2673 -2655.87834  4028.413 0.9516904
## 4-3 -2969.3267 -7063.54459  1124.891 0.2418117
```

```
############ LINE PLOT
### Save means and variances for Figure 3a
means <- as.data.frame(data_State_mclust_nC_L2$parameters$mean)
means$Var <- c( "Anger", "Anxiety", "Boredom", "Enjoy",
                        "Control", "Value", "RT_correct_L2")
names(means) <- c("1", "2", "3", "4", "Var")
data_State_nC_table_long <- gather(means, Profile, mean, 1:4, factor_key=TRUE)
data_State_nC_table_long$order <- c(rep(c(1:4),7))
data_State_nC_table_long <- data_State_nC_table_long[order(data_State_nC_table_long$order),]
data_State_nC_table_long$Profile <- as.factor(data_State_nC_table_long$Profile)
data_State_nC_table_long$Var <- as.factor(data_State_nC_table_long$Var)
data_State_nC_table_long$Var <- factor(data_State_nC_table_long$Var, levels = c("Anger", "Anxiety", "Bo
                                                                                "Control","Value",  "RT
stan_var1 <- diag(data_State_mclust_nC_L2$parameters$variance$sigma[,,1])
stan_var2 <- diag(data_State_mclust_nC_L2$parameters$variance$sigma[,,2])
stan_var3 <- diag(data_State_mclust_nC_L2$parameters$variance$sigma[,,3])
stan_var4 <- diag(data_State_mclust_nC_L2$parameters$variance$sigma[,,4])
SE_var1 <- sqrt(stan_var1)/sqrt(data_State_mclust_nC_L2$parameters$pro[1]*nrow(data_State_stan_nC_L2))
SE_var2 <- sqrt(stan_var2)/sqrt(data_State_mclust_nC_L2$parameters$pro[2]*nrow(data_State_stan_nC_L2))
```

```
SE_var3 <- sqrt(stan_var3)/sqrt(data_State_mclust_nC_L2$parameters$pro[3]*nrow(data_State_stan_nC_L2))
SE_var4 <- sqrt(stan_var4)/sqrt(data_State_mclust_nC_L2$parameters$pro[4]*nrow(data_State_stan_nC_L2))
SE_var1$Profile <- 1
SE_var2$Profile <- 2
SE_var3$Profile <- 3
SE_var4$Profile <- 4
SE_var1 <- unlist(SE_var1)
SE_var2 <- unlist(SE_var2)
SE_var3 <- unlist(SE_var3)
SE_var4 <- unlist(SE_var4)
SE_both <- data.frame(rbind(SE_var1,SE_var2,SE_var3,SE_var4))
data_State_nC_SE_long <- gather(SE_both, variable, measurement, ang_mean:initial_rt_correct2,
                                factor_key=TRUE)
data_State_nC_table_long <- cbind(data_State_nC_table_long[c(1:3)], data_State_nC_SE_long[3])
names(data_State_nC_table_long) <- c("Var", "Profile", "mean","se")
levels(data_State_nC_table_long$Profile)[levels(data_State_nC_table_long$Profile)=="1"] <- "Adaptive (3
levels(data_State_nC_table_long$Profile)[levels(data_State_nC_table_long$Profile)=="2"] <- "Moderate (3
levels(data_State_nC_table_long$Profile)[levels(data_State_nC_table_long$Profile)=="3"] <- "Negative emo
levels(data_State_nC_table_long$Profile)[levels(data_State_nC_table_long$Profile)=="4"] <- "Bored, low 
# FIGURE 3A
pd <- position_dodge(0.15)
ggplot(data = data_State_nC_table_long, mapping = ggplot2::aes(x=Var,y=mean,group=Profile)) +
  geom_errorbar(aes(ymin=mean-se, ymax=mean+se), colour="black", width=.1, position=pd) +
  geom_line(aes(linetype = Profile))+
  geom_point(aes(shape = Profile), size = 2.5) +labs(title = "Standardized Means per Profile in No-Choic
  ylab("Standardized means") + xlab("") +
  scale_shape_discrete(name = "Profile") +
  theme_classic() + theme(plot.title = element_text(hjust = 0.5)) + coord_cartesian(ylim = c(-1.5,2))
```




```r
# Outliers removed
dataOutRm_State_stan <- na.omit(data_State_nC_OutRm[c(1,3:8,10)])
dataOutRm_State_stan[c(2:8)] <- standardize(dataOutRm_State_stan[c(2:8)])
dataOutRm_State_mclust <- Mclust(dataOutRm_State_stan[c(2:8)])
summary(dataOutRm_State_mclust, parameters = TRUE)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VVE (ellipsoidal, equal orientation) model with 3 components:
##
##  log-likelihood   n df       BIC       ICL
##       -2321.371 280 65 -5009.003 -5063.347
##
## Clustering table:
##   1   2   3
##  47  99 134
##
## Mixing probabilities:
##         1         2         3
## 0.1748243 0.3685036 0.4566721
##
## Means:
##                           [,1]        [,2]       [,3]
## ang_mean             0.8642604  0.46976950 -0.7099306
## anx_mean            -0.6293238  0.98955619 -0.5575860
## bor_mean             0.8044244  0.11003802 -0.3967449
```

```
## enj_mean            -1.1988881 -0.06573503  0.5120049
## cont                -0.1245472 -0.22135548  0.2262984
## val_mean            -1.0762616  0.06954685  0.3558974
## initial_rt_correct2  0.2656795  0.05864093 -0.1490273
##
## Variances:
## [,,1]
##                         ang_mean     anx_mean    bor_mean     enj_mean
## ang_mean             1.183514609 -0.112452513  0.02527065 -0.005560751
## anx_mean            -0.112452513  0.123559798 -0.06404322 -0.004215915
## bor_mean             0.025270648 -0.064043223  0.74841531 -0.035659671
## enj_mean            -0.005560751 -0.004215915 -0.03565967  0.363825042
## cont                -0.031347046 -0.097024350  0.28001912 -0.009862777
## val_mean             0.069745743 -0.006744931 -0.06934534  0.313383035
## initial_rt_correct2  0.029888289  0.056898514 -0.07250940 -0.106655606
##                             cont     val_mean initial_rt_correct2
## ang_mean            -0.031347046  0.069745743          0.02988829
## anx_mean            -0.097024350 -0.006744931          0.05689851
## bor_mean             0.280019123 -0.069345338         -0.07250940
## enj_mean            -0.009862777  0.313383035         -0.10665561
## cont                 1.454000911  0.048048133         -0.06052732
## val_mean             0.048048133  0.436406442          0.04344940
## initial_rt_correct2 -0.060527322  0.043449397          1.58616079
## [,,2]
##                         ang_mean     anx_mean     bor_mean     enj_mean
## ang_mean             0.793830781  0.007214006  0.054754520 -0.011126405
## anx_mean             0.007214006  0.887790282  0.006309498 -0.003727873
## bor_mean             0.054754520  0.006309498  0.675817245 -0.342526561
## enj_mean            -0.011126405 -0.003727873 -0.342526561  0.617664136
## cont                -0.029515616 -0.005504536  0.073786932  0.118986142
## val_mean             0.020130432  0.031726560 -0.399894666  0.355965437
## initial_rt_correct2  0.012620258  0.011383206 -0.082769128 -0.048490306
##                             cont     val_mean initial_rt_correct2
## ang_mean            -0.029515616  0.020130432         0.012620258
## anx_mean            -0.005504536  0.031726560         0.011383206
## bor_mean             0.073786932 -0.399894666        -0.082769128
## enj_mean             0.118986142  0.355965437        -0.048490306
## cont                 0.834966782  0.145981804        -0.125540889
## val_mean             0.145981804  0.747436988         0.007452372
## initial_rt_correct2 -0.125540889  0.007452372         0.928216085
## [,,3]
##                        ang_mean    anx_mean    bor_mean    enj_mean        cont
## ang_mean             0.09176292  0.01063625  0.02431380 -0.05262513 -0.02757810
## anx_mean             0.01063625  0.17602397 -0.04326143  0.02244225 -0.03597204
## bor_mean             0.02431380 -0.04326143  0.86866425 -0.50391629 -0.03039088
## enj_mean            -0.05262513  0.02244225 -0.50391629  0.74049705  0.17319868
## cont                -0.02757810 -0.03597204 -0.03039088  0.17319868  0.77556304
## val_mean            -0.06581576  0.02606532 -0.60537268  0.61121964  0.21545279
## initial_rt_correct2  0.04526549  0.02789844 -0.06204312 -0.04323548 -0.09943941
##                        val_mean initial_rt_correct2
## ang_mean            -0.06581576          0.04526549
## anx_mean             0.02606532          0.02789844
## bor_mean            -0.60537268         -0.06204312
## enj_mean             0.61121964         -0.04323548
```

```
## cont                 0.21545279         -0.09943941
## val_mean             0.94442725          0.01591850
## initial_rt_correct2  0.01591850          0.81873130
```

```
mclustBIC(dataOutRm_State_stan[c(2:8)]) # BIC fit of top three models
```

```
## Bayesian Information Criterion (BIC):
##         EII       VII       EEI       VEI       EVI       VVI       EEE
## 1 -5600.305 -5600.305 -5634.114 -5634.114 -5634.114 -5634.114 -5168.797
## 2 -5374.697 -5317.699 -5319.744 -5288.189 -5292.746 -5239.886 -5163.071
## 3 -5292.546 -5270.336 -5206.409 -5191.551 -5144.277 -5113.116 -5112.241
## 4 -5280.409 -5251.903 -5169.473 -5164.804 -5125.975 -5083.276 -5087.685
## 5 -5263.015 -5213.880 -5122.793 -5080.002 -5168.993 -5103.462 -5095.856
## 6 -5263.177 -5164.955 -5145.940 -5069.918 -5198.791 -5096.282 -5154.570
## 7 -5237.656 -5154.769 -5134.259 -5082.567 -5187.452 -5098.998 -5143.490
## 8 -5251.917 -5152.188 -5155.306 -5129.738 -5263.242 -5156.124 -5190.436
## 9 -5250.237 -5188.968 -5156.117 -5133.802 -5289.770 -5219.664 -5206.559
##         VEE       EVE       VVE       EEV       VEV       EVV       VVV
## 1 -5168.797 -5168.797 -5168.797 -5168.797 -5168.797 -5168.797 -5168.797
## 2 -5111.803 -5092.274 -5061.127 -5102.166 -5073.843 -5131.785 -5094.641
## 3 -5095.782 -5033.686 -5009.003 -5139.202 -5165.943 -5194.023 -5165.644
## 4 -5083.498 -5066.348 -5021.509 -5233.363 -5201.707 -5312.896 -5260.145
## 5 -5096.023 -5116.520 -5058.817 -5335.126 -5302.331 -5428.378 -5390.186
## 6 -5091.161 -5156.398 -5117.952 -5461.047 -5388.082 -5575.253 -5528.155
## 7 -5113.653 -5194.441 -5161.472 -5510.724 -5461.801 -5641.641 -5582.766
## 8 -5138.202 -5230.499 -5226.634 -5591.318 -5614.561 -5770.898 -5729.067
## 9 -5164.233 -5294.445 -5226.605 -5689.523 -5739.475 -5896.782 -5898.112
##
## Top 3 models based on the BIC criterion:
##     VVE,3     VVE,4     EVE,3
## -5009.003 -5021.509 -5033.686
```

```
dataOutRm_State_mclust$loglik # Log likelihood
```

```
## [1] -2321.371
```

```
# SAVE
dataOutRm_State_stan <- cbind(dataOutRm_State_stan,dataOutRm_State_mclust$classification) # for barplot
names(dataOutRm_State_stan) <- c(names(dataOutRm_State_stan[-9]),"StateOutRm")
dataOutRm_State <- merge(data_State_nC_L2,dataOutRm_State_stan[c(1,9)], by = "uid", all.x = TRUE) # for
table(dataOutRm_State$State_nC_L2, dataOutRm_State$StateOutRm) # table original x OutRemoved
```

```
##
##      1  2  3
##   1  2  2 99
##   2  4 77 28
##   3  5 13  0
##   4 36  7  7
```

```
chisq.test(dataOutRm_State$State_nC_L2, dataOutRm_State$StateOutRm)
```

```
##
##  Pearson's Chi-squared test
##
## data:  dataOutRm_State$State_nC_L2 and dataOutRm_State$StateOutRm
## X-squared = 282.99, df = 6, p-value < 2.2e-16
```

```r
CramerV(dataOutRm_State$State_nC_L2, dataOutRm_State$StateOutRm, conf.level = TRUE)
```

```
##  Cramer V    lwr.ci    upr.ci
## 0.7108664 0.3554344 1.0000000
```

```r
###### SAVE
data_State_nC_L2_profile <- data_State_nC_L2
save(data_State_nC_L2_profile, file = "data_State_nC_L2_profile.Rda")
```

```r
load("data_State_C_usable.Rda")
data <- data_State_C_usable
data <- data[!data$uid==137150,]
########
#data_State
data_State_stan_C_L2 <- na.omit(data[c(1,3:9)])
data_State_stan_C_L2[c(2:8)] <- standardize(data_State_stan_C_L2[c(2:8)])
data_State_C_L2_mclust <- Mclust(data_State_stan_C_L2[c(2:8)])
summary(data_State_C_L2_mclust, parameters = TRUE)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VVE (ellipsoidal, equal orientation) model with 4 components:
##
##  log-likelihood   n df      BIC       ICL
##       -2579.239 328 80 -5621.92 -5676.467
##
## Clustering table:
##   1   2   3   4
## 154  74  23  77
##
## Mixing probabilities:
##          1          2          3          4
## 0.46589904 0.23999765 0.07971668 0.21438663
##
## Means:
##                           [,1]        [,2]        [,3]        [,4]
## ang_mean            -0.6440240  0.16735078  1.94939183  0.48737806
## anx_mean            -0.4567146  0.90406811  1.47253986 -0.56709364
## bor_mean            -0.5112146 -0.28161477  0.76608937  1.14135469
## enj_mean             0.4919813  0.07699239 -0.87155539 -0.83127433
## cont                 0.1850385 -0.09110516 -0.68336005 -0.04603382
## val_mean             0.4622079  0.29046236 -0.92072119 -0.98726149
## initial_rt_correct2 -0.2671398  0.33986300 -0.02717533  0.21018181
##
## Variances:
## [,,1]
##                         ang_mean     anx_mean    bor_mean     enj_mean
## ang_mean             0.061198299  0.001188805  0.01409284 -0.041304865
## anx_mean             0.001188805  0.103285328  0.00732290 -0.003654684
## bor_mean             0.014092836  0.007322900  0.53871756 -0.233062087
## enj_mean            -0.041304865 -0.003654684 -0.23306209  0.825215596
## cont                -0.013931605 -0.011939949 -0.11568174  0.251053718
## val_mean            -0.044033250  0.007951873 -0.22947862  0.592405690
```

```
## initial_rt_correct2  0.016093782 -0.012998641  0.04546311 -0.117616203
##                             cont     val_mean initial_rt_correct2
## ang_mean            -0.01393160 -0.044033250          0.01609378
## anx_mean            -0.01193995  0.007951873         -0.01299864
## bor_mean            -0.11568174 -0.229478620          0.04546311
## enj_mean             0.25105372  0.592405690         -0.11761620
## cont                 0.68445259  0.254460214         -0.08889714
## val_mean             0.25446021  0.783158547         -0.10632624
## initial_rt_correct2 -0.08889714 -0.106326240          0.68313436
## [,,2]
##                         ang_mean     anx_mean    bor_mean     enj_mean
## ang_mean             0.292922854  0.023652883  0.00784318 -0.019769883
## anx_mean             0.023652883  0.836233868 -0.01093225  0.034454664
## bor_mean             0.007843180 -0.010932248  0.38044492 -0.143510235
## enj_mean            -0.019769883  0.034454664 -0.14351023  0.521571104
## cont                -0.004078208  0.008873936 -0.03157836  0.081179366
## val_mean            -0.016849746 -0.045125433 -0.13996550  0.378345503
## initial_rt_correct2  0.009800759 -0.001217439 -0.02740754  0.002432326
##                             cont    val_mean initial_rt_correct2
## ang_mean            -0.004078208 -0.01684975         0.009800759
## anx_mean             0.008873936 -0.04512543        -0.001217439
## bor_mean            -0.031578357 -0.13996550        -0.027407538
## enj_mean             0.081179366  0.37834550         0.002432326
## cont                 0.646345777  0.09125684        -0.069839027
## val_mean             0.091256836  0.49416924         0.013566590
## initial_rt_correct2 -0.069839027  0.01356659         0.960044535
## [,,3]
##                         ang_mean    anx_mean      bor_mean      enj_mean
## ang_mean             0.584683424  0.08728133 -1.732398e-03 -6.640498e-03
## anx_mean             0.087281327  2.61972626 -4.096748e-02  1.049300e-01
## bor_mean            -0.001732398 -0.04096748  8.905351e-01 -2.462404e-06
## enj_mean            -0.006640498  0.10492999 -2.462404e-06  7.336981e-01
## cont                 0.023088611  0.02554387  1.173806e-01 -2.264985e-01
## val_mean            -0.014763203 -0.12316185  3.940113e-03  1.365652e-01
## initial_rt_correct2  0.009599409  0.03678393  3.725320e-03 -1.111035e-02
##                             cont     val_mean initial_rt_correct2
## ang_mean             0.02308861 -0.014763203         0.009599409
## anx_mean             0.02554387 -0.123161848         0.036783934
## bor_mean             0.11738057  0.003940113         0.003725320
## enj_mean            -0.22649846  0.136565171        -0.011110348
## cont                 1.62295283 -0.185580340         0.156842677
## val_mean            -0.18558034  0.708339060        -0.003419369
## initial_rt_correct2  0.15684268 -0.003419369         0.990710927
## [,,4]
##                         ang_mean     anx_mean     bor_mean     enj_mean
## ang_mean             1.345138301 -0.052969936  0.013838423  0.003514537
## anx_mean            -0.052969936  0.073321116 -0.001700905 -0.013376860
## bor_mean             0.013838423 -0.001700905  0.155271218 -0.194406594
## enj_mean             0.003514537 -0.013376860 -0.194406594  0.588785447
## cont                 0.007647279 -0.035780871  0.109135225 -0.167269418
## val_mean             0.036111023  0.018020631 -0.179981919  0.200727576
## initial_rt_correct2 -0.003536501 -0.038563798 -0.099112744  0.087281746
##                             cont    val_mean initial_rt_correct2
## ang_mean             0.007647279  0.03611102        -0.003536501
```

```
## anx_mean            -0.035780871  0.01802063        -0.038563798
## bor_mean             0.109135225 -0.17998192        -0.099112744
## enj_mean            -0.167269418  0.20072758         0.087281746
## cont                 1.357020477 -0.12690920         0.022128459
## val_mean            -0.126909197  0.56890364         0.107852079
## initial_rt_correct2  0.022128459  0.10785208         1.611738658
```

```
mclustBIC(data_State_stan_C_L2[c(2:8)]) # BIC fit of top three models
```

```
## Bayesian Information Criterion (BIC):
##         EII       VII       EEI       VEI       EVI       VVI       EEE
## 1 -6555.099 -6555.099 -6589.857 -6589.857 -6589.857 -6589.857 -6011.926
## 2 -6276.522 -6164.768 -6217.080 -6150.891 -6165.191 -5993.155 -6052.090
## 3 -6191.475 -6123.345 -6092.210 -6041.397 -5903.295 -5782.826 -5850.248
## 4 -6160.927 -5999.522 -6008.314 -5934.581 -5807.646 -5746.411 -5871.328
## 5 -6110.113 -5961.622 -5969.294 -5827.227 -5760.421 -5690.773 -5846.739
## 6 -6059.915 -5920.737 -5842.490 -5773.861 -5746.557 -5709.817 -5920.547
## 7 -6040.661 -5914.748 -5839.355 -5745.967 -5767.813 -5720.164 -5878.658
## 8 -6069.190 -5913.787 -5872.382 -5770.600 -5789.570 -5740.575 -5892.686
## 9 -6084.134 -5914.979 -5898.346 -5766.753 -5731.461 -5765.602 -5975.107
##         VEE       EVE       VVE       EEV       VEV       EVV       VVV
## 1 -6011.926 -6011.926 -6011.926 -6011.926 -6011.926 -6011.926 -6011.926
## 2 -5893.574 -5840.487 -5682.611 -5811.703 -5720.587 -5827.038 -5708.485
## 3 -5874.392 -5752.958 -5651.811 -5809.116 -5728.995 -5823.592 -5728.451
## 4 -5811.724 -5677.773 -5621.920 -5785.605 -5844.204 -5821.710 -5865.955
## 5 -5829.264 -5681.043 -5650.478 -5869.740 -5908.243 -5959.992 -5986.006
## 6 -5803.999 -5690.136 -5627.645 -5996.859 -5988.847 -6121.234 -6046.521
## 7 -5796.010        NA -5650.582 -6083.927 -6070.930 -6201.208 -6131.380
## 8 -5824.666 -5887.402 -5722.871 -6119.270 -6168.100 -6230.970 -6287.250
## 9 -5797.887 -5890.932 -5787.448 -6212.541 -6266.214 -6369.594 -6430.040
##
## Top 3 models based on the BIC criterion:
##     VVE,4     VVE,6     VVE,5
## -5621.920 -5627.645 -5650.478
```

```
data_State_C_L2_mclust$loglik # Log likelihood
```

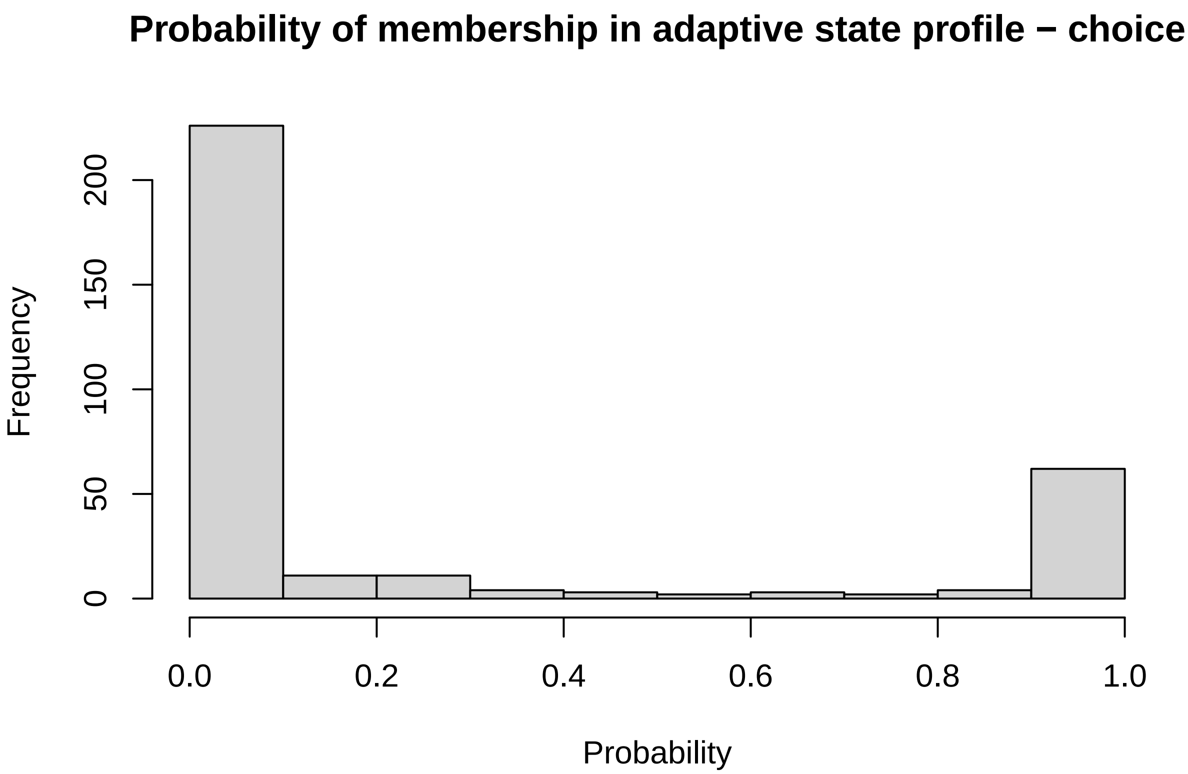
```
## [1] -2579.239
```

```
mclustICL(data_State_stan_C_L2[c(2:8)])
```

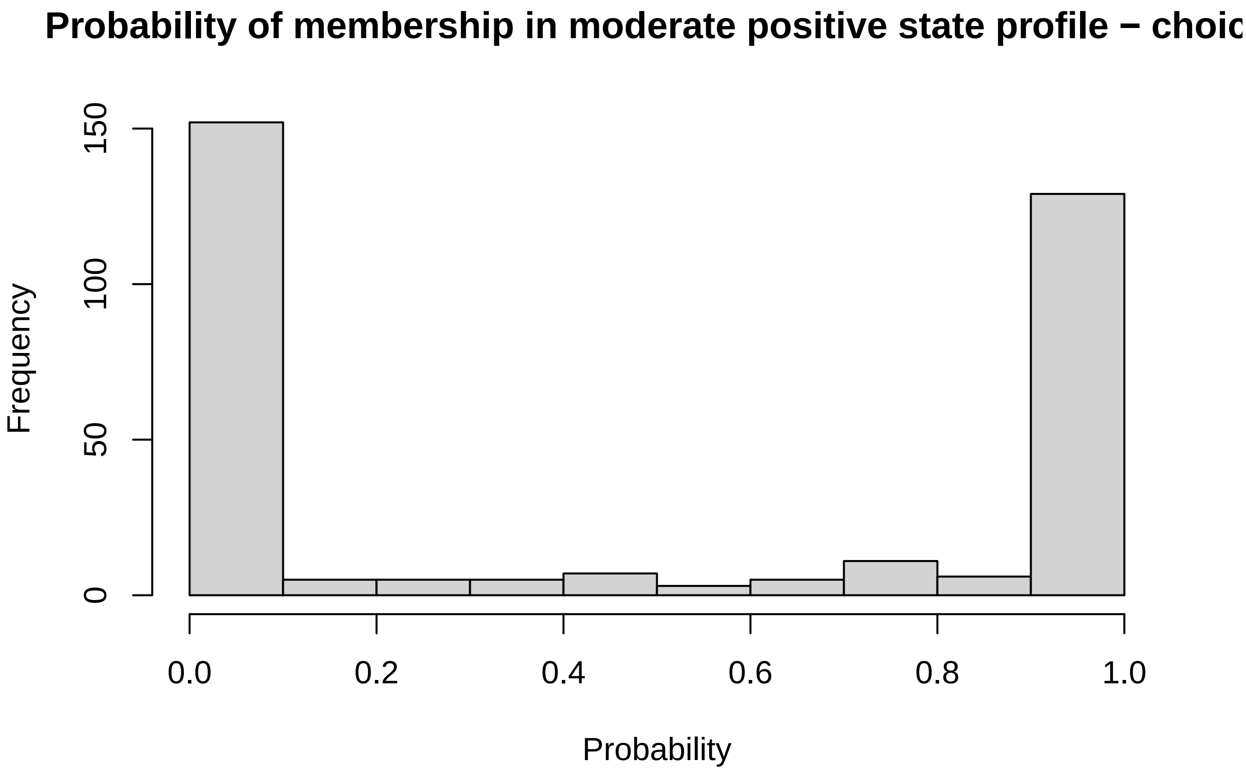
```
## Integrated Complete-data Likelihood (ICL) criterion:
##         EII       VII       EEI       VEI       EVI       VVI       EEE
## 1 -6555.099 -6555.099 -6589.857 -6589.857 -6589.857 -6589.857 -6011.926
## 2 -6338.824 -6212.957 -6261.381 -6198.143 -6215.301 -6029.744 -6196.222
## 3 -6264.884 -6212.414 -6158.423 -6084.507 -5937.824 -5825.004 -5874.642
## 4 -6264.858 -6092.913 -6082.390 -6000.878 -5857.494 -5817.882 -5938.217
## 5 -6219.404 -6069.393 -6044.175 -5896.076 -5824.518 -5760.942 -5945.248
## 6 -6157.010 -6020.648 -5914.401 -5842.866 -5818.525 -5783.877 -6041.653
## 7 -6145.613 -6015.159 -5913.974 -5807.070 -5856.651 -5795.678 -5991.417
## 8 -6194.993 -6027.128 -5997.373 -5861.016 -5891.349 -5814.877 -6000.359
## 9 -6251.914 -6022.042 -6016.891 -5865.876 -5798.751 -5832.814 -6092.585
##         VEE       EVE       VVE       EEV       VEV       EVV       VVV
## 1 -6011.926 -6011.926 -6011.926 -6011.926 -6011.926 -6011.926 -6011.926
## 2 -5961.180 -5867.870 -5716.591 -5843.416 -5750.720 -5859.706 -5740.238
## 3 -5965.623 -5794.428 -5692.561 -5846.776 -5778.300 -5863.721 -5767.352
```

```
## 4 -5923.819 -5719.262 -5676.467 -5825.144 -5897.487 -5855.620 -5903.510
## 5 -5929.134 -5725.170 -5706.076 -5931.985 -5968.204 -6010.895 -6022.677
## 6 -5900.514 -5760.313 -5716.543 -6061.291 -6053.667 -6176.440 -6072.951
## 7 -5905.688        NA -5720.654 -6145.612 -6125.976 -6254.255 -6170.747
## 8 -5931.364 -6034.131 -5803.618 -6176.543 -6216.611 -6271.662 -6336.143
## 9 -5906.190 -6029.199 -5855.799 -6271.555 -6304.546 -6409.492 -6469.052
##
## Top 3 models based on the ICL criterion:
##     VVE,4     VVE,3     VVE,5
## -5676.467 -5692.561 -5706.076
```

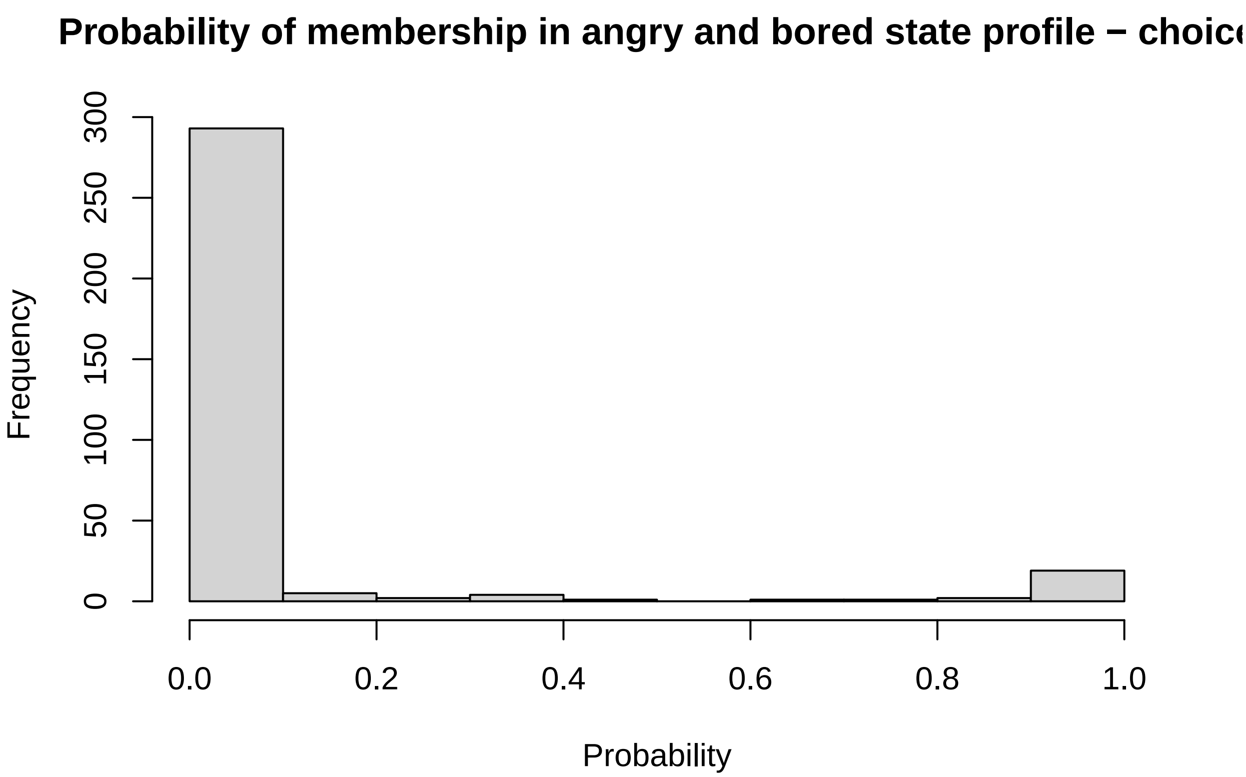
```
# SAVE
data_State_stan_C_L2 <- cbind(data_State_stan_C_L2,data_State_C_L2_mclust$classification, data_State_C_L
names(data_State_stan_C_L2) <- c(names(data_State_stan_C_L2[-c(9:13)]),"State_C_L2","Stat_L2_p1","Stat_L
data_State_C_L2 <- merge(data,data_State_stan_C_L2[c(1,c(9:13))], by = "uid", all.x = TRUE) # for table
data_State_C_L2$State_C_L2 <- as.factor(data_State_C_L2$State_C_L2)
#data_State_C_L2$State_C_L2 <- revalue(data_State_C_L2$State_C_L2, c("2"="1", "1"="2")) # same numberin
# Distribution of membership probability
hist(data_State_C_L2$Stat_L2_p2, main = "Probability of membership in adaptive state profile - choice",
     xlab = "Probability")
```



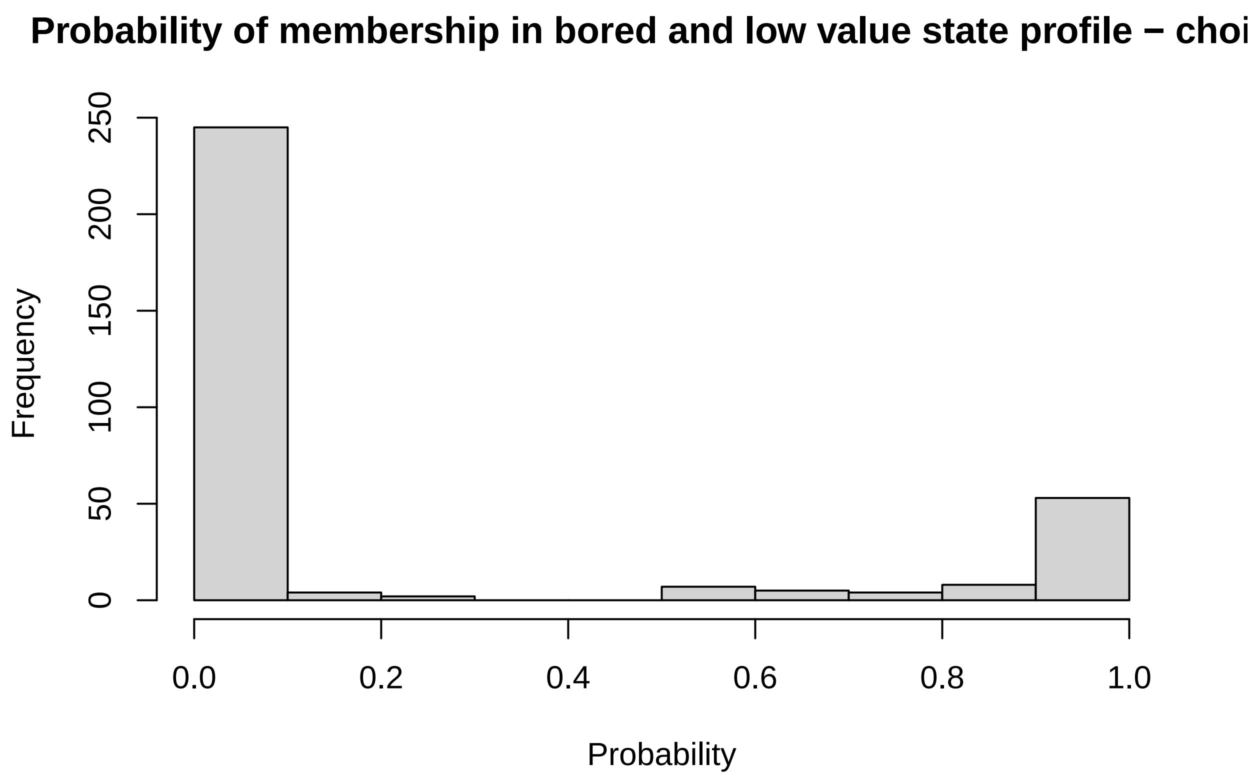

```
hist(data_State_C_L2$Stat_L2_p1, main = "Probability of membership in moderate positive state profile -
     xlab = "Probability")
```




```r
hist(data_State_C_L2$Stat_L2_p3, main = "Probability of membership in angry and bored state profile - cl
     xlab = "Probability")
```




```r
hist(data_State_C_L2$Stat_L2_p4, main = "Probability of membership in bored and low value state profile
     xlab = "Probability")
```




```
######## Means and SDs
Ang_mean <- aggregate(ang_mean ~ State_C_L2, mean, data = data_State_C_L2)
Anx_mean <- aggregate(anx_mean ~ State_C_L2, mean, data = data_State_C_L2)
Bor_mean <- aggregate(bor_mean ~ State_C_L2, mean, data = data_State_C_L2)
Enj_mean <- aggregate(enj_mean ~ State_C_L2, mean, data = data_State_C_L2)
Con_mean <- aggregate(cont ~ State_C_L2, mean, data = data_State_C_L2)
Val_mean <- aggregate(val_mean ~ State_C_L2, mean, data = data_State_C_L2)
rt_mean <- aggregate(initial_rt_correct2 ~ State_C_L2, mean, data = data_State_C_L2)
########  SDs
Ang_sd <- aggregate(ang_mean ~ State_C_L2, sd, data = data_State_C_L2)
Anx_sd <- aggregate(anx_mean ~ State_C_L2, sd, data = data_State_C_L2)
Bor_sd <- aggregate(bor_mean ~ State_C_L2, sd, data = data_State_C_L2)
Enj_sd <- aggregate(enj_mean ~ State_C_L2, sd, data = data_State_C_L2)
Con_sd <- aggregate(cont ~ State_C_L2, sd, data = data_State_C_L2)
Val_sd <- aggregate(val_mean ~ State_C_L2, sd, data = data_State_C_L2)
rt_sd <- aggregate(initial_rt_correct2 ~ State_C_L2, sd, data = data_State_C_L2)

# Test differences between profiles
a_ang <- aov(ang_mean ~ State_C_L2, data = data_State_C_L2)
summary(a_ang)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## State_C_L2    3  8.658  2.8860   133.2 <2e-16 ***
## Residuals   324  7.019  0.0217
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 19 observations deleted due to missingness
```

```r
TukeyHSD(a_ang)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = ang_mean ~ State_C_L2, data = data_State_C_L2)
##
## $State_C_L2
##            diff         lwr        upr     p adj
## 2-1  0.19249174  0.13872811  0.2462554 0.0000000
## 3-1  0.59298701  0.50801815  0.6779559 0.0000000
## 4-1  0.23644467  0.18339317  0.2894962 0.0000000
## 3-2  0.40049527  0.30975428  0.4912363 0.0000000
## 4-2  0.04395293 -0.01792342  0.1058293 0.2590996
## 4-3 -0.35654234 -0.44686322 -0.2662215 0.0000000
```

```r
a_anx <- aov(anx_mean ~ State_C_L2, data = data_State_C_L2)
summary(a_anx)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## State_C_L2    3  5.070  1.6900   149.9 <2e-16 ***
## Residuals   324  3.653  0.0113
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 19 observations deleted due to missingness
```

```r
TukeyHSD(a_anx)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = anx_mean ~ State_C_L2, data = data_State_C_L2)
##
## $State_C_L2
##            diff         lwr        upr     p adj
## 2-1  0.23484693  0.19606105  0.2736328 0.0000000
## 3-1  0.34040910  0.27911131  0.4017069 0.0000000
## 4-1 -0.02003724 -0.05830937  0.0182349 0.5304985
## 3-2  0.10556217  0.04010028  0.1710241 0.0002336
## 4-2 -0.25488417 -0.29952268 -0.2102457 0.0000000
## 4-3 -0.36044634 -0.42560515 -0.2952875 0.0000000
```

```r
a_bor <- aov(bor_mean ~ State_C_L2, data = data_State_C_L2)
summary(a_bor)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## State_C_L2    3  12.53   4.177   109.2 <2e-16 ***
## Residuals   324  12.39   0.038
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 19 observations deleted due to missingness
```

```r
TukeyHSD(a_bor)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
```

```
## Fit: aov(formula = bor_mean ~ State_C_L2, data = data_State_C_L2)
##
## $State_C_L2
##           diff          lwr       upr     p adj
## 2-1 0.07299565  0.001572011 0.1444193 0.0430298
## 3-1 0.35884367  0.245964679 0.4717227 0.0000000
## 4-1 0.46465613  0.394178538 0.5351337 0.0000000
## 3-2 0.28584802  0.165300893 0.4063952 0.0000000
## 4-2 0.39166048  0.309459300 0.4738617 0.0000000
## 4-3 0.10581245 -0.014176565 0.2258015 0.1054821
```

```r
a_con <- aov(cont ~ State_C_L2, data = data_State_C_L2)
summary(a_con)
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)
## State_C_L2    3  1.006  0.3353   6.273 0.000378 ***
## Residuals   324 17.317  0.0534
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 19 observations deleted due to missingness
```

```r
TukeyHSD(a_con)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = cont ~ State_C_L2, data = data_State_C_L2)
##
## $State_C_L2
##             diff          lwr          upr     p adj
## 2-1 -0.066614317 -0.151060722  0.017832088 0.1765798
## 3-1 -0.211757388 -0.345217756 -0.078297020 0.0003072
## 4-1 -0.061096525 -0.144424385  0.022231336 0.2328583
## 3-2 -0.145143071 -0.287669713 -0.002616429 0.0441256
## 4-2  0.005517792 -0.091671230  0.102706815 0.9988746
## 4-3  0.150660863  0.008794093  0.292527634 0.0324760
```

```r
a_val <- aov(val_mean ~ State_C_L2, data = data_State_C_L2)
summary(a_val)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## State_C_L2    3  6.766  2.2555   76.49 <2e-16 ***
## Residuals   324  9.554  0.0295
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 19 observations deleted due to missingness
```

```r
TukeyHSD(a_val)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = val_mean ~ State_C_L2, data = data_State_C_L2)
##
## $State_C_L2
##              diff        lwr         upr     p adj
## 2-1 -0.0479170542 -0.1106412  0.01480712 0.2004513
```

```
## 3-1 -0.3243335260 -0.4234638 -0.22520329 0.0000000
## 4-1 -0.3250579896 -0.3869513 -0.26316463 0.0000000
## 3-2 -0.2764164718 -0.3822809 -0.17055209 0.0000000
## 4-2 -0.2771409355 -0.3493299 -0.20495193 0.0000000
## 4-3 -0.0007244637 -0.1060987  0.10464979 0.9999980
```

```
a_rt <- aov(initial_rt_correct2 ~ State_C_L2, data = data_State_C_L2)
summary(a_rt)
```

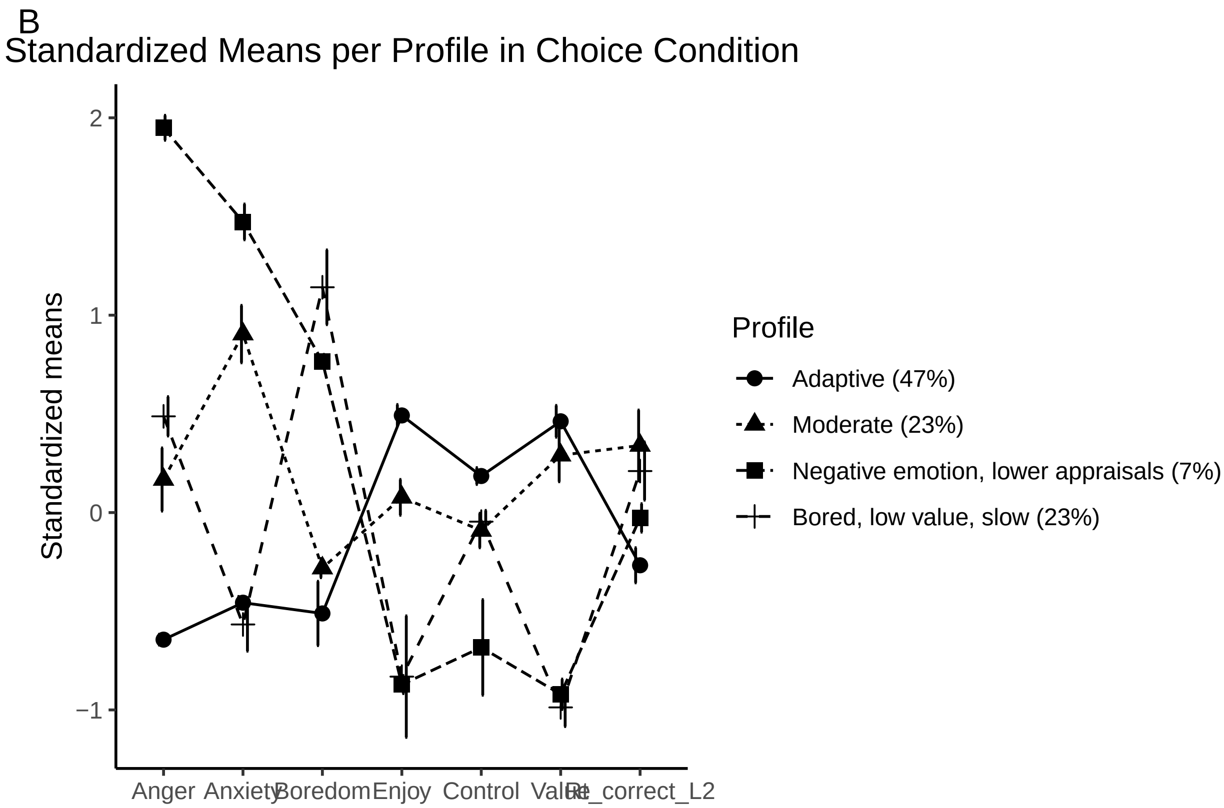
```
##              Df    Sum Sq   Mean Sq F value   Pr(>F)
## State_C_L2    3 8.934e+08 297815932   7.572 6.56e-05 ***
## Residuals   324 1.274e+10  39333494
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 19 observations deleted due to missingness
```

```
TukeyHSD(a_rt)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = initial_rt_correct2 ~ State_C_L2, data = data_State_C_L2)
##
## $State_C_L2
##           diff        lwr      upr     p adj
## 2-1  3821.3518  1530.4936 6112.210 0.0001283
## 3-1  1814.2974 -1806.2094 5434.804 0.5673497
## 4-1  2923.5330   663.0187 5184.047 0.0051560
## 3-2 -2007.0544 -5873.5107 1859.402 0.5377922
## 4-2  -897.8188 -3534.3581 1738.721 0.8155584
## 4-3  1109.2357 -2739.3196 4957.791 0.8790456
```

```
### Save means and variances for Figure
means <- as.data.frame(data_State_C_L2_mclust$parameters$mean)
means$Var <- c( "Anger", "Anxiety", "Boredom", "Enjoy",
                        "Control","Value", "Rt_correct_L2")
names(means) <- c("1", "2", "3", "4","Var") # same as in no choice
data_State_C_table_long <- gather(means, Profile, mean, 1:4, factor_key=TRUE)
data_State_C_table_long$order <- c(rep(c(1:4),7))
data_State_C_table_long <- data_State_C_table_long[order(data_State_C_table_long$order),]
data_State_C_table_long$Profile <- as.factor(data_State_C_table_long$Profile)
data_State_C_table_long$Var <- as.factor(data_State_C_table_long$Var)
data_State_C_table_long$Var <- factor(data_State_C_table_long$Var, levels = c("Anger", "Anxiety", "Bored
                                                                                 "Control","Value","Rt_c
stan_var1 <- diag(data_State_C_L2_mclust$parameters$variance$sigma[,,1])
stan_var2 <- diag(data_State_C_L2_mclust$parameters$variance$sigma[,,2])
stan_var3 <- diag(data_State_C_L2_mclust$parameters$variance$sigma[,,3])
stan_var4 <- diag(data_State_C_L2_mclust$parameters$variance$sigma[,,4])
SE_var1 <- sqrt(stan_var1)/sqrt(data_State_C_L2_mclust$parameters$pro[2]*nrow(data_State_C_L2))
SE_var2 <- sqrt(stan_var2)/sqrt(data_State_C_L2_mclust$parameters$pro[1]*nrow(data_State_C_L2))
SE_var3 <- sqrt(stan_var3)/sqrt(data_State_C_L2_mclust$parameters$pro[3]*nrow(data_State_C_L2))
SE_var4 <- sqrt(stan_var4)/sqrt(data_State_C_L2_mclust$parameters$pro[4]*nrow(data_State_C_L2))
SE_var1$Profile <- 1
SE_var2$Profile <- 2
SE_var3$Profile <- 3
SE_var4$Profile <- 4
```

```
SE_var1 <- unlist(SE_var1)
SE_var2 <- unlist(SE_var2)
SE_var3 <- unlist(SE_var3)
SE_var4 <- unlist(SE_var4)
SE_both <- data.frame(rbind(SE_var1,SE_var2,SE_var3,SE_var4))
data_State_C_SE_long <- gather(SE_both, variable, measurement, ang_mean:initial_rt_correct2,
                                factor_key=TRUE)
data_State_C_table_long <- cbind(data_State_C_table_long[c(1:3)], data_State_C_SE_long[3])
names(data_State_C_table_long) <- c("Var", "Profile", "mean","se")
levels(data_State_C_table_long$Profile)[levels(data_State_C_table_long$Profile)=="1"] <- "Adaptive (47%
levels(data_State_C_table_long$Profile)[levels(data_State_C_table_long$Profile)=="2"] <- "Moderate (23%
levels(data_State_C_table_long$Profile)[levels(data_State_C_table_long$Profile)=="3"] <- "Negative emot
levels(data_State_C_table_long$Profile)[levels(data_State_C_table_long$Profile)=="4"] <- "Bored, low va
# FIGURE 3 b
pd <- position_dodge(0.15)
ggplot(data = data_State_C_table_long, mapping = ggplot2::aes(x=Var,y=mean,group=Profile)) +
  geom_errorbar(aes(ymin=mean-se, ymax=mean+se), colour="black", width=.1, position=pd) +
  geom_line(aes(linetype = Profile))+
  geom_point(aes(shape = Profile), size = 2.5) + labs(title = "Standardized Means per Profile in Choice
  ylab("Standardized means") + xlab("") +
  scale_shape_discrete(name = "Profile") +
  theme_classic() + theme(plot.title = element_text(hjust = 0.5))
```




```
# Outliers removed
dataOutRm <- data_State_C_OutRm
dataOutRm_State_stan <- na.omit(dataOutRm[c(1,3:9)])
dataOutRm_State_stan[c(2:8)] <- standardize(dataOutRm_State_stan[c(2:8)])
dataOutRm_State_mclust <- Mclust(dataOutRm_State_stan[c(2:8)])
```

```r
summary(dataOutRm_State_mclust, parameters = TRUE)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust EVE (ellipsoidal, equal volume and orientation) model with 3 components:
##
##  log-likelihood   n df       BIC      ICL
##       -2365.505 285 63 -5087.116 -5122.13
##
## Clustering table:
##   1   2   3
##  56 193  36
##
## Mixing probabilities:
##         1         2         3
## 0.2213493 0.6545593 0.1240914
##
## Means:
##                          [,1]        [,2]        [,3]
## ang_mean             0.70547190 -0.55951245  1.69293300
## anx_mean             1.36514988 -0.36049783 -0.53353979
## bor_mean            -0.04413342 -0.17799803  1.01763038
## enj_mean            -0.07980347  0.20969676 -0.96376170
## cont                -0.08718063  0.11077386 -0.42880247
## val_mean             0.15852547  0.13437060 -0.99155166
## initial_rt_correct2  0.26661754 -0.08445295 -0.03010787
##
## Variances:
## [,,1]
##                         ang_mean     anx_mean     bor_mean     enj_mean
## ang_mean             0.670272499  0.144441808  0.061257297  0.004442331
## anx_mean             0.144441808  1.105370147 -0.015595956  0.023331666
## bor_mean             0.061257297 -0.015595956  0.555196116 -0.297517648
## enj_mean             0.004442331  0.023331666 -0.297517648  0.524721574
## cont                -0.026917970 -0.006113031 -0.008756194  0.111020319
## val_mean             0.006101613 -0.022368155 -0.349534655  0.406175259
## initial_rt_correct2 -0.005934031 -0.024760877  0.006937830 -0.054193845
##                             cont     val_mean initial_rt_correct2
## ang_mean            -0.026917970  0.006101613        -0.005934031
## anx_mean            -0.006113031 -0.022368155        -0.024760877
## bor_mean            -0.008756194 -0.349534655         0.006937830
## enj_mean             0.111020319  0.406175259        -0.054193845
## cont                 0.633133693  0.106499617        -0.215445484
## val_mean             0.106499617  0.521192378        -0.054802752
## initial_rt_correct2 -0.215445484 -0.054802752         0.697273847
## [,,2]
##                          ang_mean     anx_mean    bor_mean    enj_mean
## ang_mean             0.1438698891  0.059097263  0.03213316 -0.07396856
## anx_mean             0.0590972629  0.323331614 -0.03004312  0.04671054
## bor_mean             0.0321331598 -0.030043125  1.00893018 -0.63135701
## enj_mean            -0.0739685570  0.046710537 -0.63135701  1.04119399
## cont                -0.0009492828  0.014613191 -0.13537608  0.31119998
```

```
## val_mean           -0.0754330616  0.044844044 -0.69263156  0.77183641
## initial_rt_correct2  0.0431298322  0.009755699  0.20280328 -0.18748240
##                           cont    val_mean initial_rt_correct2
## ang_mean           -0.0009492828 -0.07543306         0.043129832
## anx_mean            0.0146131906  0.04484404         0.009755699
## bor_mean           -0.1353760817 -0.69263156         0.202803285
## enj_mean            0.3111999791  0.77183641        -0.187482404
## cont                1.0175955207  0.30167342        -0.056063912
## val_mean            0.3016734156  1.06192867        -0.206355050
## initial_rt_correct2 -0.0560639123 -0.20635505         0.982791096
## [,,3]
##                         ang_mean     anx_mean     bor_mean    enj_mean
## ang_mean            0.305400122 -0.046428489 -0.005684847 -0.01792823
## anx_mean           -0.046428489  0.162707970 -0.013592064  0.01476008
## bor_mean           -0.005684847 -0.013592064  0.634729387 -0.04264857
## enj_mean           -0.017928228  0.014760078 -0.042648567  0.46487058
## cont                0.028816036 -0.005383903  0.329124513 -0.04243171
## val_mean           -0.006942514  0.017303517 -0.123486516  0.16141204
## initial_rt_correct2  0.038526368  0.063719978 -0.125648813  0.14255941
##                         cont     val_mean initial_rt_correct2
## ang_mean            0.028816036 -0.006942514          0.03852637
## anx_mean           -0.005383903  0.017303517          0.06371998
## bor_mean            0.329124513 -0.123486516         -0.12564881
## enj_mean           -0.042431713  0.161412044          0.14255941
## cont                1.452478200 -0.080781020         -0.06877025
## val_mean           -0.080781020  0.416742528          0.12970468
## initial_rt_correct2 -0.068770246  0.129704683          1.56736931
```

```
dataOutRm_State_mclust$loglik
```

```
## [1] -2365.505
```

```
dataOutRm_State_mclust$BIC
```

```
## Bayesian Information Criterion (BIC):
##         EII       VII       EEI       VEI       EVI       VVI       EEE
## 1 -5699.772 -5699.772 -5733.687 -5733.687 -5733.687 -5733.687 -5275.573
## 2 -5487.123 -5407.925 -5448.877 -5400.994 -5411.215 -5349.189 -5176.886
## 3 -5395.922 -5334.528 -5349.673 -5306.639 -5373.198 -5284.264 -5206.857
## 4 -5348.694 -5269.124 -5241.036 -5201.921 -5192.632 -5187.984 -5146.861
## 5 -5313.843 -5256.869 -5174.970 -5162.687 -5163.198 -5149.219 -5165.149
## 6 -5306.766 -5263.910 -5206.617 -5168.937 -5182.716 -5175.781 -5195.012
## 7 -5244.816 -5222.949 -5198.188 -5149.595 -5187.177 -5159.463 -5196.249
## 8 -5253.620 -5234.692 -5182.594 -5150.572 -5201.884 -5201.519 -5188.070
## 9 -5282.592 -5245.589 -5200.913 -5180.578 -5218.407 -5249.403 -5208.832
##         VEE       EVE       VVE       EEV       VEV       EVV       VVV
## 1 -5275.573 -5275.573 -5275.573 -5275.573 -5275.573 -5275.573 -5275.573
## 2 -5202.783 -5179.641 -5112.905 -5188.030 -5143.969 -5208.335 -5162.013
## 3 -5147.203 -5087.116 -5146.405 -5200.121 -5210.420 -5199.815 -5280.343
## 4 -5115.728 -5110.912 -5099.847 -5274.362 -5253.910 -5358.318 -5364.215
## 5 -5132.143 -5168.010 -5173.675 -5380.188 -5375.559 -5476.447 -5490.215
## 6 -5159.430 -5207.613 -5201.005 -5477.224 -5489.579 -5622.863 -5617.705
## 7 -5170.177 -5181.065 -5198.446 -5547.366 -5562.211 -5709.150 -5720.798
## 8 -5204.211 -5253.548 -5234.630 -5629.265 -5667.827 -5796.932 -5847.560
## 9 -5232.953 -5305.367 -5282.211 -5718.287 -5744.736 -5950.455 -5953.779
```

```
##
## Top 3 models based on the BIC criterion:
##     EVE,3     VVE,4     EVE,4
## -5087.116 -5099.847 -5110.912
```

```
# Outliers removed
dataOutRm <- data_State_C_OutRm
dataOutRm_State_stan <- na.omit(dataOutRm[c(1,3:9)])
dataOutRm_State_stan[c(2:8)] <- standardize(dataOutRm_State_stan[c(2:8)])
dataOutRm_State_mclust <- Mclust(dataOutRm_State_stan[c(2:8)])
summary(dataOutRm_State_mclust, parameters = TRUE)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust EVE (ellipsoidal, equal volume and orientation) model with 3 components:
##
##  log-likelihood   n df       BIC      ICL
##       -2365.505 285 63 -5087.116 -5122.13
##
## Clustering table:
##   1   2   3
##  56 193  36
##
## Mixing probabilities:
##         1         2         3
## 0.2213493 0.6545593 0.1240914
##
## Means:
##                            [,1]        [,2]        [,3]
## ang_mean             0.70547190 -0.55951245  1.69293300
## anx_mean             1.36514988 -0.36049783 -0.53353979
## bor_mean            -0.04413342 -0.17799803  1.01763038
## enj_mean            -0.07980347  0.20969676 -0.96376170
## cont                -0.08718063  0.11077386 -0.42880247
## val_mean             0.15852547  0.13437060 -0.99155166
## initial_rt_correct2  0.26661754 -0.08445295 -0.03010787
##
## Variances:
## [,,1]
##                         ang_mean     anx_mean     bor_mean     enj_mean
## ang_mean             0.670272499  0.144441808  0.061257297  0.004442331
## anx_mean             0.144441808  1.105370147 -0.015595956  0.023331666
## bor_mean             0.061257297 -0.015595956  0.555196116 -0.297517648
## enj_mean             0.004442331  0.023331666 -0.297517648  0.524721574
## cont                -0.026917970 -0.006113031 -0.008756194  0.111020319
## val_mean             0.006101613 -0.022368155 -0.349534655  0.406175259
## initial_rt_correct2 -0.005934031 -0.024760877  0.006937830 -0.054193845
##                             cont     val_mean initial_rt_correct2
## ang_mean            -0.026917970  0.006101613        -0.005934031
## anx_mean            -0.006113031 -0.022368155        -0.024760877
## bor_mean            -0.008756194 -0.349534655         0.006937830
## enj_mean             0.111020319  0.406175259        -0.054193845
## cont                 0.633133693  0.106499617        -0.215445484
```

```
## val_mean             0.106499617  0.521192378        -0.054802752
## initial_rt_correct2 -0.215445484 -0.054802752         0.697273847
## [,,2]
##                         ang_mean     anx_mean    bor_mean    enj_mean
## ang_mean             0.1438698891  0.059097263  0.03213316 -0.07396856
## anx_mean             0.0590972629  0.323331614 -0.03004312  0.04671054
## bor_mean             0.0321331598 -0.030043125  1.00893018 -0.63135701
## enj_mean            -0.0739685570  0.046710537 -0.63135701  1.04119399
## cont                -0.0009492828  0.014613191 -0.13537608  0.31119998
## val_mean            -0.0754330616  0.044844044 -0.69263156  0.77183641
## initial_rt_correct2  0.0431298322  0.009755699  0.20280328 -0.18748240
##                             cont    val_mean initial_rt_correct2
## ang_mean            -0.0009492828 -0.07543306         0.043129832
## anx_mean             0.0146131906  0.04484404         0.009755699
## bor_mean            -0.1353760817 -0.69263156         0.202803285
## enj_mean             0.3111999791  0.77183641        -0.187482404
## cont                 1.0175955207  0.30167342        -0.056063912
## val_mean             0.3016734156  1.06192867        -0.206355050
## initial_rt_correct2 -0.0560639123 -0.20635505         0.982791096
## [,,3]
##                        ang_mean     anx_mean     bor_mean    enj_mean
## ang_mean             0.305400122 -0.046428489 -0.005684847 -0.01792823
## anx_mean            -0.046428489  0.162707970 -0.013592064  0.01476008
## bor_mean            -0.005684847 -0.013592064  0.634729387 -0.04264857
## enj_mean            -0.017928228  0.014760078 -0.042648567  0.46487058
## cont                 0.028816036 -0.005383903  0.329124513 -0.04243171
## val_mean            -0.006942514  0.017303517 -0.123486516  0.16141204
## initial_rt_correct2  0.038526368  0.063719978 -0.125648813  0.14255941
##                            cont     val_mean initial_rt_correct2
## ang_mean             0.028816036 -0.006942514          0.03852637
## anx_mean            -0.005383903  0.017303517          0.06371998
## bor_mean             0.329124513 -0.123486516         -0.12564881
## enj_mean            -0.042431713  0.161412044          0.14255941
## cont                 1.452478200 -0.080781020         -0.06877025
## val_mean            -0.080781020  0.416742528          0.12970468
## initial_rt_correct2 -0.068770246  0.129704683          1.56736931
```

```
dataOutRm_State_mclust$loglik
```

```
## [1] -2365.505
```

```
dataOutRm_State_mclust$BIC
```

```
## Bayesian Information Criterion (BIC):
##         EII       VII       EEI       VEI       EVI       VVI       EEE
## 1 -5699.772 -5699.772 -5733.687 -5733.687 -5733.687 -5733.687 -5275.573
## 2 -5487.123 -5407.925 -5448.877 -5400.994 -5411.215 -5349.189 -5176.886
## 3 -5395.922 -5334.528 -5349.673 -5306.639 -5373.198 -5284.264 -5206.857
## 4 -5348.694 -5269.124 -5241.036 -5201.921 -5192.632 -5187.984 -5146.861
## 5 -5313.843 -5256.869 -5174.970 -5162.687 -5163.198 -5149.219 -5165.149
## 6 -5306.766 -5263.910 -5206.617 -5168.937 -5182.716 -5175.781 -5195.012
## 7 -5244.816 -5222.949 -5198.188 -5149.595 -5187.177 -5159.463 -5196.249
## 8 -5253.620 -5234.692 -5182.594 -5150.572 -5201.884 -5201.519 -5188.070
## 9 -5282.592 -5245.589 -5200.913 -5180.578 -5218.407 -5249.403 -5208.832
##         VEE       EVE       VVE       EEV       VEV       EVV       VVV
```

```
## 1 -5275.573 -5275.573 -5275.573 -5275.573 -5275.573 -5275.573 -5275.573
## 2 -5202.783 -5179.641 -5112.905 -5188.030 -5143.969 -5208.335 -5162.013
## 3 -5147.203 -5087.116 -5146.405 -5200.121 -5210.420 -5199.815 -5280.343
## 4 -5115.728 -5110.912 -5099.847 -5274.362 -5253.910 -5358.318 -5364.215
## 5 -5132.143 -5168.010 -5173.675 -5380.188 -5375.559 -5476.447 -5490.215
## 6 -5159.430 -5207.613 -5201.005 -5477.224 -5489.579 -5622.863 -5617.705
## 7 -5170.177 -5181.065 -5198.446 -5547.366 -5562.211 -5709.150 -5720.798
## 8 -5204.211 -5253.548 -5234.630 -5629.265 -5667.827 -5796.932 -5847.560
## 9 -5232.953 -5305.367 -5282.211 -5718.287 -5744.736 -5950.455 -5953.779
##
## Top 3 models based on the BIC criterion:
##     EVE,3     VVE,4     EVE,4
## -5087.116 -5099.847 -5110.912
```

```
# SAVE
dataOutRm_State_stan <- cbind(dataOutRm_State_stan,dataOutRm_State_mclust$classification) # for barplot
names(dataOutRm_State_stan) <- c(names(dataOutRm_State_stan[-9]),"StateOutRm")
dataOutRm_State <- merge(data_State_C_L2,dataOutRm_State_stan[c(1,9)], by = "uid", all.x = TRUE) # for
table(dataOutRm_State$State_C_L2, dataOutRm_State$StateOutRm) # table original x OutRemoved
```

```
##
##       1   2   3
##   1   0 154   0
##   2  49   3   2
##   3   4   0   3
##   4   2  36  31
```

```
chisq.test(dataOutRm_State$State_C_L2, dataOutRm_State$StateOutRm)
```

```
##
##  Pearson's Chi-squared test
##
## data:  dataOutRm_State$State_C_L2 and dataOutRm_State$StateOutRm
## X-squared = 332.79, df = 6, p-value < 2.2e-16
```

```
CramerV(dataOutRm_State$State_C_L2, dataOutRm_State$StateOutRm, conf.level = TRUE)
```

```
##  Cramer V    lwr.ci    upr.ci
## 0.7654372 0.3827197 1.0000000
```

```
###### SAVE
data_State_C_L2_profile <- data_State_C_L2
save(data_State_C_L2_profile, file = "data_State_C_L2_profile.Rda")
```

Descriptives

```
# TRAIT level
load("data_Trait_profile.Rda")
data_T <- data_Trait_profile[!is.na(data_Trait_profile$Trait),]
# Means and sd's of Trait measures
round(apply(data_T[c(54,55,56,62,59,60,51)], 2, mean),2) # Mean
```

```
##   AEQ_Ang_mean      AMAS_mean   AEQ_Bor_mean AEQ_Enj_3_mean     SELFC_mean
##           2.06           1.93           2.42           2.20           2.59
##       VAL_mean          Grade
##           2.88           6.91
```

```r
round(apply(data_T[c(54,55,56,62,59,60,51)], 2, sd),2) # SD
```

```
##   AEQ_Ang_mean      AMAS_mean   AEQ_Bor_mean AEQ_Enj_3_mean     SELFC_mean
##           0.95           0.73           0.93           0.88           0.74
##       VAL_mean          Grade
##           0.73           1.23
```

```r
# DATA STATE LEVEL - no choice block
load("data_State_nC_L2_profile.Rda")
data_S_nC <- data_State_nC_L2_profile[!is.na(data_State_nC_L2_profile$State_nC_L2),]
# means and sds
round(apply(data_S_nC[c(3:8,10)], 2, mean),2) # Mean
```

```
##            ang_mean            anx_mean            bor_mean            enj_mean
##                0.26                0.17                0.57                0.40
##                cont            val_mean initial_rt_correct2
##                0.67                0.40            18412.76
```

```r
round(apply(data_S_nC[c(3:8,10)], 2, sd),2) # SD
```

```
##            ang_mean            anx_mean            bor_mean            enj_mean
##                0.25                0.18                0.27                0.23
##                cont            val_mean initial_rt_correct2
##                0.24                0.22             8526.43
```

```r
# DATA STATE LEVEL - choice block
load("data_State_C_L2_profile.Rda")
data_S_C <- data_State_C_L2_profile[!is.na(data_State_C_L2_profile$State_C_L2),]
# Means and sds
round(apply(data_S_C[c(3:9)], 2, mean),2) # Mean
```

```
##            ang_mean            anx_mean            bor_mean            enj_mean
##                0.22                0.15                0.57                0.41
##                cont            val_mean initial_rt_correct2
##                0.69                0.40            13916.81
```

```r
round(apply(data_S_C[c(3:9)], 2, sd),2) # SD
```

```
##            ang_mean            anx_mean            bor_mean            enj_mean
##                0.22                0.16                0.28                0.23
##                cont            val_mean initial_rt_correct2
##                0.24                0.22             6457.93
```

```r
# CORRELATION TABLE -> Table S3
data_S3_up <- merge(data_T,data_S_nC, by = "uid")
data_Cor_S3_up <- na.omit(data_S3_up[c(54:56,62,59:60,51,66:72,74)])
cor_mtrx_S3_up <- rcorr(as.matrix(data_Cor_S3_up))
cor_mtr_val_S3_up <- round(cor_mtrx_S3_up$r,2)
cor_mtr_p.val_S3_up <- round(cor_mtrx_S3_up$P,3)
data_S3_mid <- merge(data_T,data_S_C, by = "uid")
data_Cor_S3_mid <- na.omit(data_S3_mid[c(54:56,62,59:60,51,66:72,73)])
cor_mtrx_S3_mid <- rcorr(as.matrix(data_Cor_S3_mid))
cor_mtr_val_S3_mid <- round(cor_mtrx_S3_mid$r,2)
cor_mtr_p.val_S3_mid <- round(cor_mtrx_S3_mid$P,3)
data_S3_down <- merge(data_S_nC,data_S_C, by = "uid")
data_Cor_S3_down <- na.omit(data_S3_down[c(3:8,10,18:24)])
cor_mtrx_S3_down <- rcorr(as.matrix(data_Cor_S3_down))
```

```r
cor_mtr_val_S3_down <- round(cor_mtrx_S3_down$r,2)
cor_mtr_p.val_S3_down <- round(cor_mtrx_S3_down$P,3)
######## t test state differences
t.test(data_S3_down$ang_mean.x,data_S3_down$ang_mean.y, paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  data_S3_down$ang_mean.x and data_S3_down$ang_mean.y
## t = 2.2942, df = 315, p-value = 0.02244
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.003293091 0.042958853
## sample estimates:
## mean of the differences
##              0.02312597
```

```r
t.test(data_S3_down$anx_mean.x,data_S3_down$anx_mean.y, paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  data_S3_down$anx_mean.x and data_S3_down$anx_mean.y
## t = 1.3691, df = 315, p-value = 0.1719
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.004304686  0.024001429
## sample estimates:
## mean of the differences
##             0.009848371
```

```r
t.test(data_S3_down$bor_mean.x,data_S3_down$bor_mean.y, paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  data_S3_down$bor_mean.x and data_S3_down$bor_mean.y
## t = -0.13328, df = 315, p-value = 0.8941
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.02428599  0.02120444
## sample estimates:
## mean of the differences
##            -0.001540776
```

```r
t.test(data_S3_down$enj_mean.x,data_S3_down$enj_mean.y, paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  data_S3_down$enj_mean.x and data_S3_down$enj_mean.y
## t = -1.4602, df = 315, p-value = 0.1452
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.029432711  0.004355745
## sample estimates:
```

```
## mean of the differences
##             -0.01253848
```

```r
t.test(data_S3_down$cont.x,data_S3_down$cont.y, paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  data_S3_down$cont.x and data_S3_down$cont.y
## t = -1.5398, df = 315, p-value = 0.1246
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.038947660  0.004750236
## sample estimates:
## mean of the differences
##             -0.01709871
```

```r
t.test(data_S3_down$val_mean.x,data_S3_down$val_mean.y, paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  data_S3_down$val_mean.x and data_S3_down$val_mean.y
## t = 0.42792, df = 315, p-value = 0.669
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.01408580  0.02191598
## sample estimates:
## mean of the differences
##             0.003915088
```

```r
t.test(data_S3_down$initial_rt_correct2.x,data_S3_down$initial_rt_correct2.y, paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  data_S3_down$initial_rt_correct2.x and data_S3_down$initial_rt_correct2.y
## t = 8.7445, df = 315, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  3290.814 5201.612
## sample estimates:
## mean of the differences
##                4246.213
```

Compare profile memberships: Chi-square & Mosaic Plot

```r
load("data_Trait_profile.Rda")
load("data_State_nC_L2_profile.Rda")
load("data_State_C_L2_profile.Rda")
# merge
data_all1 <- merge(data_Trait_profile, data_State_nC_L2_profile, by = "uid")
data_all <- merge(data_all1, data_State_C_L2_profile, by = "uid")
######
# chi-square tests
```

```
# Trait
chisq.test(data_all$Trait,data_all$Gender) # Sex

##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  data_all$Trait and data_all$Gender
## X-squared = 4.5372, df = 1, p-value = 0.03317

CramerV(data_all$Trait,data_all$Gender, conf.level = TRUE)

##  Cramer V    lwr.ci    upr.ci
## 0.1226104 0.0000000 1.0000000

table(data_all$Trait,data_all$Gender)

##
##        1   2
##   1   51  37
##   2  110 140

# State & Sex
# no choice
chisq.test(data_all$State_nC_L2,data_all$Gender)

##
##  Pearson's Chi-squared test
##
## data:  data_all$State_nC_L2 and data_all$Gender
## X-squared = 8.8257, df = 3, p-value = 0.0317

CramerV(data_all$State_nC_L2,data_all$Gender, conf.level = TRUE)

##  Cramer V    lwr.ci    upr.ci
## 0.1630443 0.0000000 1.0000000

table(data_all$State_nC_L2,data_all$Gender)

##
##       1  2
##   1 47 56
##   2 51 72
##   3 26 22
##   4 37 21

# choice
chisq.test(data_all$State_C_L2,data_all$Gender)

##
##  Pearson's Chi-squared test
##
## data:  data_all$State_C_L2 and data_all$Gender
## X-squared = 9.1738, df = 3, p-value = 0.02707

CramerV(data_all$State_C_L2,data_all$Gender, conf.level = TRUE)

##  Cramer V    lwr.ci    upr.ci
## 0.1672392 0.0000000 1.0000000
```

```r
table(data_all$State_C_L2,data_all$Gender)
```

```
##
##      1  2
##   1 71 83
##   2 26 48
##   3 14  9
##   4 44 33
```

```r
# State & School year
# no choice
chisq.test(data_all$State_nC_L2,data_all$Year)
```

```
##
##  Pearson's Chi-squared test
##
## data:  data_all$State_nC_L2 and data_all$Year
## X-squared = 12.567, df = 3, p-value = 0.005672
```

```r
CramerV(data_all$State_nC_L2,data_all$Year, conf.level = TRUE)
```

```
##  Cramer V    lwr.ci    upr.ci
## 0.1945595 0.0000000 1.0000000
```

```r
table(data_all$State_nC_L2,data_all$Year)
```

```
##
##      1  2
##   1 30 73
##   2 47 76
##   3 26 22
##   4 30 28
```

```r
# choice
chisq.test(data_all$State_C_L2,data_all$Year)
```

```
##
##  Pearson's Chi-squared test
##
## data:  data_all$State_C_L2 and data_all$Year
## X-squared = 2.4242, df = 3, p-value = 0.4892
```

```r
CramerV(data_all$State_C_L2,data_all$Year, conf.level = TRUE)
```

```
##   Cramer V     lwr.ci     upr.ci
## 0.08596937 0.00000000 1.00000000
```

```r
## Both State blocks
chisq.test(data_all$State_nC_L2,data_all$State_C_L2)
```

```
##
##  Pearson's Chi-squared test
##
## data:  data_all$State_nC_L2 and data_all$State_C_L2
## X-squared = 190.93, df = 9, p-value < 2.2e-16
```

```r
CramerV(data_all$State_nC_L2,data_all$State_C_L2, conf.level = TRUE)
```

```
##  Cramer V    lwr.ci    upr.ci
```

```
## 0.4487809 0.2243916 1.0000000
```

```r
table(data_all$State_nC_L2,data_all$State_C_L2)
```

```
##
##       1  2  3  4
##    1 79  6  0 17
##    2 54 47  1 18
##    3  6 15 16  4
##    4 10  5  5 33
```

```r
#Plot
dat <- as.data.frame(cbind(data_all$State_C_L2, data_all$State_nC_L2))
names(dat) <- c("State_C_L2", "State_nC_L2")
## FIGURE 4
ggplot(data = dat) +
          geom_mosaic(aes(x = product(State_nC_L2, State_C_L2),
                      fill=State_nC_L2), na.rm=TRUE) +
          labs(x = "Choice", y = "No choice",
               title='Profile Membership Across State Conditions') +
          scale_fill_manual(name = "Profile",
                     values = c("darkgreen","darkolivegreen3","orange","FireBrick"),
                     labels = c("Adaptive","Moderate","NegEmoLowerApp","BoredLowValueSlow")) +
          theme_classic() + theme(plot.title = element_text(hjust = 0.5)) +
          theme(axis.text.x = element_text(face="bold", color="#993333",
                           size=14, angle=45),
          axis.text.y = element_text(face="bold", color="#993333",
                           size=14, angle=45))
```

```
# State and Trait
# no Choice block
chisq.test(data_all$State_nC_L2,data_all$Trait)
```

```
##
##  Pearson's Chi-squared test
##
## data:  data_all$State_nC_L2 and data_all$Trait
## X-squared = 21.212, df = 3, p-value = 9.512e-05
```

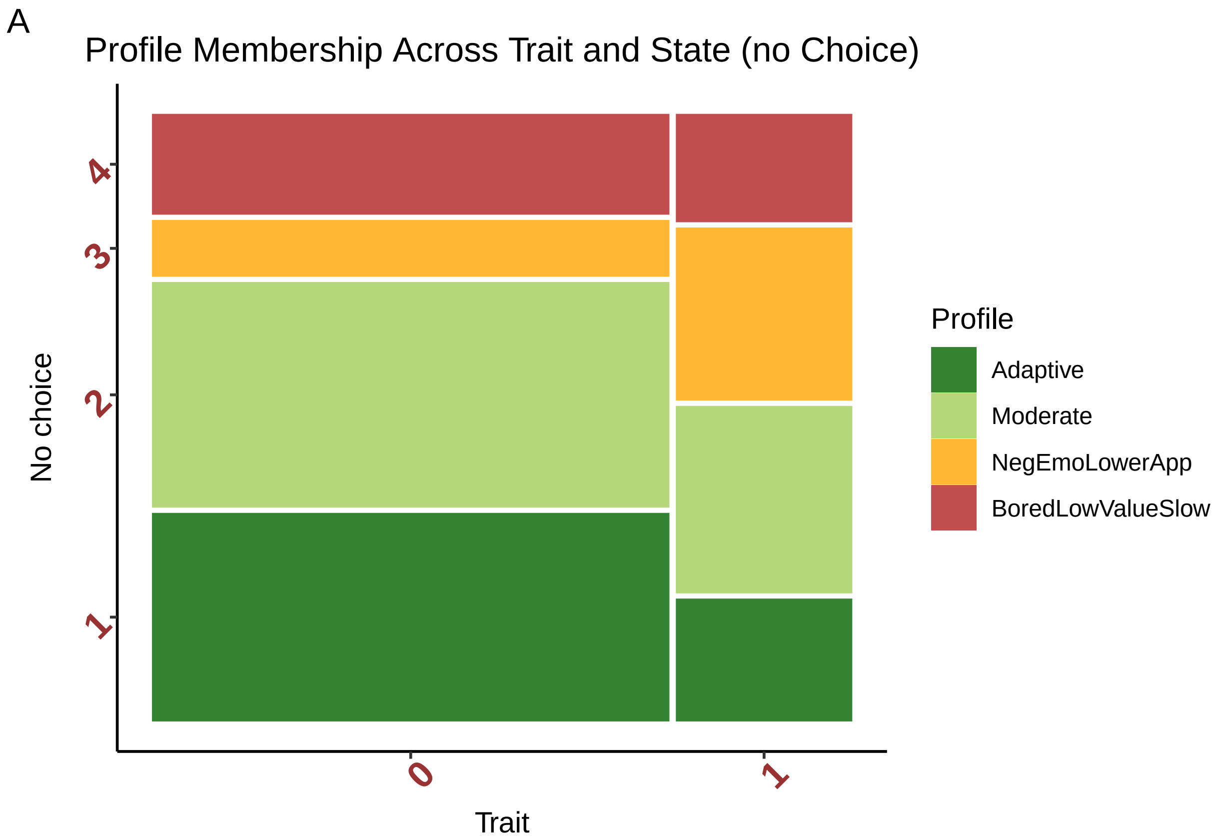
```
CramerV(data_all$State_nC_L2,data_all$Trait, conf.level = TRUE)
```

```
##  Cramer V    lwr.ci    upr.ci
## 0.2562654 0.0000000 1.0000000
```

```
table(data_all$State_nC_L2,data_all$Trait)
```

```
##
##       1  2
##   1 17 85
##   2 26 92
##   3 24 23
##   4 15 41
```

```
#Plot
dat <- as.data.frame(cbind(data_all$Trait, data_all$State_nC_L2))
names(dat) <- c("Trait", "State_nC_L2")
dat[!is.na(dat$Trait)&dat$Trait==2,]$Trait <- 0
# FIGURE 5a
ggplot(data = dat) +
          geom_mosaic(aes(x = product(State_nC_L2, Trait),
                      fill=State_nC_L2), na.rm=TRUE) +
          labs(x = "Trait", y = "No choice",
               title='Profile Membership Across Trait and State (no Choice)',
               tag = "A") +
          scale_fill_manual(name = "Profile",
                     values = c("darkgreen","darkolivegreen3","orange","FireBrick"),
                     labels = c("Adaptive","Moderate","NegEmoLowerApp","BoredLowValueSlow")) +
          theme_classic() + theme(plot.title = element_text(hjust = 0.5)) +
          theme(axis.text.x = element_text(face="bold", color="#993333",
                           size=14, angle=45),
          axis.text.y = element_text(face="bold", color="#993333",
                           size=14, angle=45))
```




```
# Choice block
chisq.test(data_all$State_C_L2,data_all$Trait)
```

```
##
##  Pearson's Chi-squared test
##
## data:  data_all$State_C_L2 and data_all$Trait
## X-squared = 32.951, df = 3, p-value = 3.297e-07
```

```
CramerV(data_all$State_C_L2,data_all$Trait, conf.level = TRUE)
```

```
##  Cramer V    lwr.ci    upr.ci
## 0.3213972 0.0000000 1.0000000
```

```
table(data_all$State_C_L2,data_all$Trait)
```

```
##
##        1   2
##    1  18 134
##    2  23  46
##    3  13   9
##    4  25  51
```

```
dat <- as.data.frame(cbind(data_all$Trait, data_all$State_C_L2))
names(dat) <- c("Trait", "State_C_L2")
dat[!is.na(dat$Trait)&dat$Trait==2,]$Trait <- 0
ggplot(data = dat) +
          geom_mosaic(aes(x = product(State_C_L2, Trait),
                      fill=State_C_L2), na.rm=TRUE) +
```

```
        labs(x = "Trait", y = "Choice",
             title='Profile Membership Across Trait and State (Choice)',
             tag = "B") +
        scale_fill_manual(name = "Profile",
                    values = c("darkgreen","darkolivegreen3","orange","FireBrick"),
                    labels = c("Adaptive","Moderate","NegEmoLowerApp","BoredLowValueSlow")) +
        theme_classic() + theme(plot.title = element_text(hjust = 0.5)) +
        theme(axis.text.x = element_text(face="bold", color="#993333",
                          size=14, angle=45),
        axis.text.y = element_text(face="bold", color="#993333",
                          size=14, angle=45))
```