codingRMD_MNK

MNK

2024-11-11

Contents

1 Start

1.1 Installing and loading packages

```
if(!requireNamespace("rempsyc")) install.packages("rempsyc"); library(rempsyc)
if(!requireNamespace("flextable")) install.packages("flextable"); library(flextable)
if(!requireNamespace("broom")) install.packages("broom"); library(broom)
if(!requireNamespace("report")) install.packages("report"); library(report)
if(!requireNamespace("effectsize")) install.packages("effectsize"); library(effectsize)
if(!requireNamespace("tinytex")) install.packages("tinytex"); library(tinytex)
if(!requireNamespace("haven")) install.packages("haven"); library(haven)
if(!requireNamespace("dplyr")) install.packages("dplyr"); library(dplyr)
if(!requireNamespace("visdat")) install.packages("visdat"); library(visdat)
if(!requireNamespace("naniar")) install.packages("naniar"); library(naniar)
if(!requireNamespace("psych")) install.packages("psych"); library(psych)
if(!requireNamespace("mice")) install.packages("mice"); library(mice)
if(!requireNamespace("Hmisc")) install.packages("Hmisc"); library(Hmisc)
if(!requireNamespace("knitr")) install.packages("knitr"); library(knitr)
if(!requireNamespace("kableExtra")) install.packages("kableExtra"); library(kableExtra)
if(!requireNamespace("lavaan")) install.packages("lavaan"); library(lavaan)
if(!requireNamespace("semPlot")) install.packages("semPlot"); library(semPlot)
if(!requireNamespace("rstatix")) install.packages("rstatix"); library(rstatix)
```

1.2 Loading datafiles

1.2.1 EMC data

```
setwd("V:/Research/Dementie/Studenten/Studenten/Max/Databeheer")
dataEMC <- haven::read_sav("data_SCTQ_merged_23082024.sav") #load EMC collected data
dataEMC[dataEMC == 999] <- NA #set missing values to NA
dataEMC_copy <- dataEMC #make copy of dataset
```

1.2.2 UMCG data

```
setwd("V:/Research/Dementie/Studenten/Studenten/Max/Databeheer")
dataUMCG <- haven::read_sav("data_SC_UMCG_mnk_errorAdj.sav") #load UMCG collected data
dataUMCG[dataUMCG == 999] <- NA #set missing values to NA
dataUMCG_copy <- dataUMCG #make copy of dataset
```

2 Methods: Data preprocessing

2.1 Pre defined transformation

```r
#change direction; higher scores better perception
dataEMC$TAS20_fac1_tf <- 35 - dataEMC$TAS20_fac1_Identificeren_Gevoelens
dataUMCG$TAS20_fac1_tf <- 35 - dataUMCG$TAS20_fac1_Identificeren_Gevoelens
```

2.2 Synchronizing names and compute totals

```r
#adding total scores for FP subset to EMC data
dataEMC$FP_1_6_total <- dataEMC$FP_1t6_ToM + dataEMC$FP_1t6_empathy

#realising identical column names
colnames(dataUMCG)[which(colnames(dataUMCG) == "SET_UMCG_Totaal")] <- "SET_UMCG_total"

#calculating subscores of SET subset to EMC data
dataEMC$SET_UMCG_Cognitief_Totaal <- dataEMC$SET_UMCG_1 +dataEMC$SET_UMCG_2 + dataEMC$SET_UMCG_3
dataEMC$SET_UMCG_Affectief_Totaal <- dataEMC$SET_UMCG_4 +dataEMC$SET_UMCG_5 + dataEMC$SET_UMCG_6
```

2.3 Creating ID values for UMCG sample

```r
dataUMCG$ID <- paste0("UMCG", seq(1, nrow(dataUMCG)))
```

2.4 Data pooling

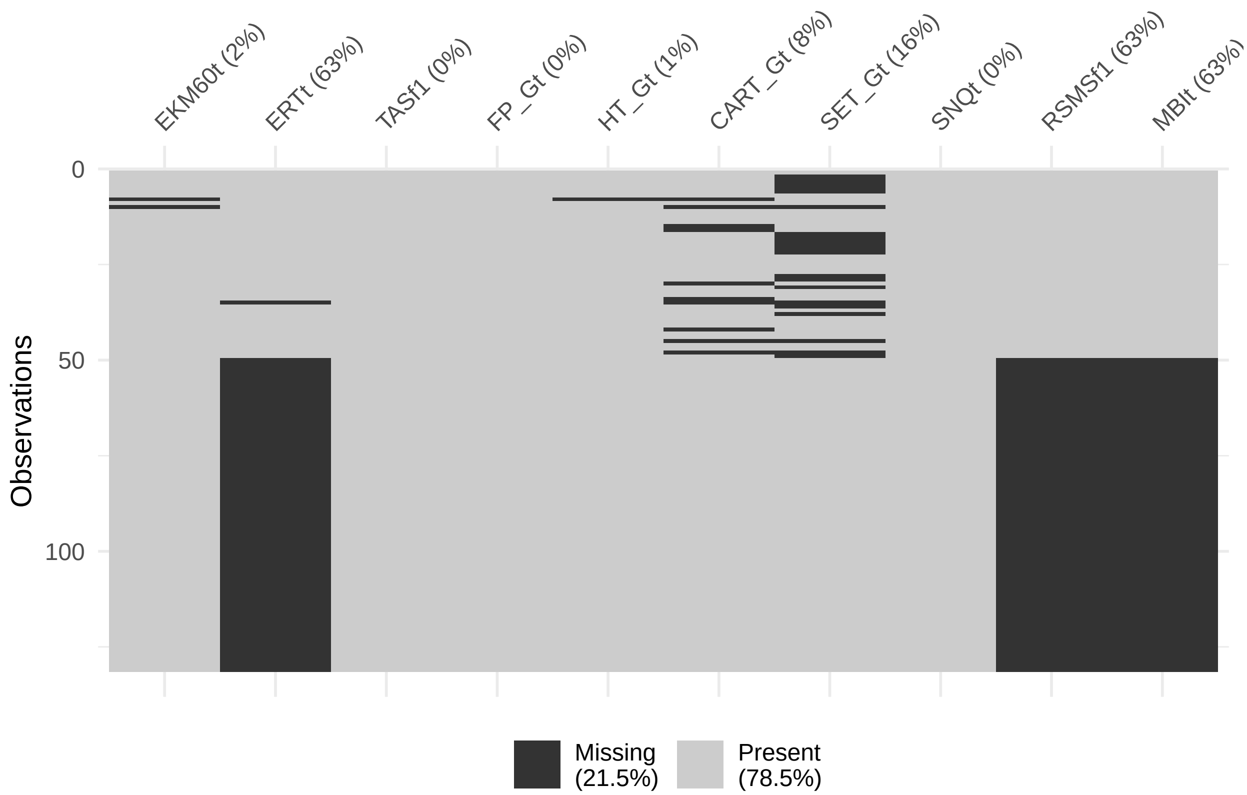
```r
data_pooled <- dplyr::bind_rows(dataEMC, dataUMCG) #pooling the data
data_pooled_copy <- data_pooled #make a copy of pooled dataset
```

2.5 Renaming and var name sets

```r
names_relevant_items <- c("Ekman60_total", "ERT_total", "TAS20_fac1_tf",
                          "FP_1_6_total", "hintingtask_total", "cartoons_total", "SET_UMCG_total",
                          "SNQ_NL_total", "RSMS_AMSP", "MBI_total")
names_rel_items_abbr <- c('EKM60t', 'ERTt', 'TASf1',
                          'FP_Gt', 'HT_Gt', 'CART_Gt', 'SET_Gt',
                          'SNQt', 'RSMSf1', 'MBIt')
names(data_pooled)[match(names_relevant_items, names(data_pooled))] <- names_rel_items_abbr #renaming
names_covariates <- c("age", "sex", "education_level")
```

2.6 Missing values part 1; visualisation

```r
vis_miss(data_pooled[,names_rel_items_abbr])
```



15% of data is missing for SET_Gt, all these missings are in the EMC data. For these people with missings, SET was conducted with another testing protocol (solely multiple choice question, no open questions). Below, we will impute SET_Gt scores for these people based on means in the EMC sample corrected for their scores on the MC questions.

2.6.1 Imputing the SET_Gt variable some EMC missings due to different testing protocol

```r
#imputing SET_Gt open question scores based on means in UMCG sample conditioned on
#correctness of corresponding MC question.
data_pooled[data_pooled$centerID == "EMC" & is.na(data_pooled$SET_open_2), "SET_open_2"] <-
  as.numeric(colMeans(data_pooled[data_pooled$centerID == "EMC", "SET_open_2"], na.rm = TRUE))
data_pooled[data_pooled$centerID == "EMC" & is.na(data_pooled$SET_open_3), "SET_open_3"] <-
  as.numeric(colMeans(data_pooled[data_pooled$centerID == "EMC", "SET_open_3"], na.rm = TRUE))
data_pooled[data_pooled$centerID == "EMC" & is.na(data_pooled$SET_open_6), "SET_open_6"] <-
  as.numeric(colMeans(data_pooled[data_pooled$centerID == "EMC", "SET_open_6"], na.rm = TRUE))
data_pooled[data_pooled$centerID == "EMC" & is.na(data_pooled$SET_open_14), "SET_open_14"] <-
  as.numeric(colMeans(data_pooled[data_pooled$centerID == "EMC", "SET_open_14"], na.rm = TRUE))
data_pooled[data_pooled$centerID == "EMC" & is.na(data_pooled$SET_open_16), "SET_open_16"] <-
  as.numeric(colMeans(data_pooled[data_pooled$centerID == "EMC", "SET_open_16"], na.rm = TRUE))
data_pooled[data_pooled$centerID == "EMC" & is.na(data_pooled$SET_open_17), "SET_open_17"] <-
  as.numeric(colMeans(data_pooled[data_pooled$centerID == "EMC", "SET_open_17"], na.rm = TRUE))
```

```r
data_pooled[data_pooled$centerID == "EMC", "SET_UMCG_1"] <-
  data_pooled[data_pooled$centerID == "EMC",]$SET_open_2 +
  data_pooled[data_pooled$centerID == "EMC",]$SET_mc_2
data_pooled[data_pooled$centerID == "EMC", "SET_UMCG_2"] <-
  data_pooled[data_pooled$centerID == "EMC",]$SET_open_3 +
  data_pooled[data_pooled$centerID == "EMC",]$SET_mc_3
data_pooled[data_pooled$centerID == "EMC", "SET_UMCG_3"] <-
  data_pooled[data_pooled$centerID == "EMC",]$SET_open_6 +
  data_pooled[data_pooled$centerID == "EMC",]$SET_mc_6
data_pooled[data_pooled$centerID == "EMC", "SET_UMCG_4"] <-
  data_pooled[data_pooled$centerID == "EMC",]$SET_open_14 +
  data_pooled[data_pooled$centerID == "EMC",]$SET_mc_14
data_pooled[data_pooled$centerID == "EMC", "SET_UMCG_5"] <-
  data_pooled[data_pooled$centerID == "EMC",]$SET_open_16 +
  data_pooled[data_pooled$centerID == "EMC",]$SET_mc_16
data_pooled[data_pooled$centerID == "EMC", "SET_UMCG_6"] <-
  data_pooled[data_pooled$centerID == "EMC",]$SET_open_17 +
  data_pooled[data_pooled$centerID == "EMC",]$SET_mc_17

data_pooled$SET_Gt <- rowSums(data_pooled[,c(which(names(data_pooled)=="SET_UMCG_1")
                                        : which(names(data_pooled)=="SET_UMCG_6"))])
```

In the above plot it is visible that we miss >=63% of data for four items (ERT, SET, RSMS, MBI). These items were not included in the data collection of UMCG, and therefor we assume that these missing values are missing completely at random (aka the characteristics of these participants did not influence whether this data is present/absent). Later on, data for the UMCG sample will be imputed for those four items using stochastic imputation (predicted mean matching).

The 8% missing values for the CART_Gt variable can probably be explained by lack of time (slower participants) because it was the last test in the EMC protocol. We will impute the scores using stochastic imputation (predicted mean matching).

Imputation will be continued after checks for outliers and non-normality.

3 Methods: Outlier and normality checks

3.1 Descriptives (before outlier deletion and/or tranformations)

3.1.1 EMC sample

```r
# Descriptives EMC before outlier deletion and/or tranformations
psych::describe(data_pooled[data_pooled$centerID == "EMC",
                            c(names_covariates,
                              "MoCA_total",
                              names_rel_items_abbr)]) %>%
  select(n, min, max, mean, median, sd, skew, kurtosis)
```

```
##                    n  min max  mean median    sd  skew kurtosis
## age               49 21.0  79 48.10  54.00 19.28 -0.12    -1.47
## sex               49  0.0   1  0.65   1.00  0.48 -0.62    -1.64
```

```
## education_level 49  2.0   7  5.71   6.00  1.12 -1.01     1.11
## MoCA_total      49 19.0  30 25.96  26.00  2.60 -0.72     0.10
## EKM60t          47 34.0  58 46.96  47.00  5.32 -0.47     0.13
## ERTt            48 34.0  76 56.85  57.50  9.67 -0.19    -0.65
## TASf1           49  5.0  28 20.08  21.00  5.37 -0.62     0.02
## FP_Gt           49 10.5  24 20.14  20.50  2.51 -1.07     2.30
## HT_Gt           48  6.0  12 11.21  11.50  1.11 -2.32     7.76
## CART_Gt         39  2.5  12  8.10   8.00  2.67 -0.30    -0.91
## SET_Gt          46  8.0  12 11.33  11.45  0.86 -1.91     3.88
## SNQt            49 11.0  21 19.06  19.00  1.63 -2.49     9.95
## RSMSf1          49 11.0  32 22.61  23.00  4.68 -0.51    -0.15
## MBIt            49 42.0  70 54.43  55.00  7.08  0.16    -0.69
```

3.1.2 UMCG sample

```r
# Descriptives UMCG before outlier deletion and/or tranformations
psych::describe(data_pooled[data_pooled$centerID == "UMCG",
                            c(names_covariates,
                              "MoCA_total",
                              names_rel_items_abbr)]) %>%
  select(n, min, max, mean, median, sd, skew, kurtosis)
```

```
## Warning in FUN(newX[, i], ...): no non-missing arguments to min; returning Inf
## Warning in FUN(newX[, i], ...): no non-missing arguments to min; returning Inf
## Warning in FUN(newX[, i], ...): no non-missing arguments to min; returning Inf
## Warning in FUN(newX[, i], ...): no non-missing arguments to min; returning Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning -Inf
## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning -Inf
## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning -Inf
## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning -Inf

##                  n min  max  mean median    sd  skew kurtosis
## age             82  18   66 28.71     22 14.07  1.36     0.16
## sex             82   0    1  0.63      1  0.48 -0.55    -1.72
## education_level 82   3    7  6.00      6  0.61 -1.30     6.07
## MoCA_total       0 Inf -Inf   NaN     NA    NA    NA       NA
## EKM60t          82  38   56 46.99     48  4.52 -0.26    -0.88
## ERTt             0 Inf -Inf   NaN     NA    NA    NA       NA
## TASf1           82   7   28 21.00     21  4.50 -0.80     0.58
## FP_Gt           82  13   24 20.17     21  2.51 -0.81     0.21
## HT_Gt           82   6   12 11.40     12  1.00 -2.66     9.62
## CART_Gt         82   0   12  8.51      9  2.51 -0.49    -0.05
## SET_Gt          82   8   12 11.33     12  0.94 -1.38     1.28
## SNQt            82  16   22 19.38     19  1.27 -0.40    -0.21
## RSMSf1           0 Inf -Inf   NaN     NA    NA    NA       NA
## MBIt             0 Inf -Inf   NaN     NA    NA    NA       NA
```

3.1.3 Pooled

```r
# Descriptives pooled before outlier deletion and/or tranformations
psych::describe(data_pooled[,
                            c(names_covariates,
                              "MoCA_total",
                              names_rel_items_abbr)]) %>%
  select(n, min, max, mean, median, sd, skew, kurtosis)
```

```
##                   n  min max  mean median    sd  skew kurtosis
## age             131 18.0  79 35.96   24.0 18.69  0.73    -1.02
## sex             131  0.0   1  0.64    1.0  0.48 -0.58    -1.67
## education_level 131  2.0   7  5.89    6.0  0.84 -1.48     4.09
## MoCA_total       49 19.0  30 25.96   26.0  2.60 -0.72     0.10
## EKM60t          129 34.0  58 46.98   48.0  4.81 -0.37    -0.23
## ERTt             48 34.0  76 56.85   57.5  9.67 -0.19    -0.65
## TASf1           131  5.0  28 20.66   21.0  4.84 -0.77     0.47
## FP_Gt           131 10.5  24 20.16   20.5  2.50 -0.91     1.05
## HT_Gt           130  6.0  12 11.33   12.0  1.04 -2.55     8.99
## CART_Gt         121  0.0  12  8.38    9.0  2.56 -0.44    -0.33
## SET_Gt          128  8.0  12 11.33   12.0  0.91 -1.56     2.13
## SNQt            131 11.0  22 19.26   19.0  1.42 -1.67     7.31
## RSMSf1           49 11.0  32 22.61   23.0  4.68 -0.51    -0.15
## MBIt             49 42.0  70 54.43   55.0  7.08  0.16    -0.69
```

3.2 Outliers

3.2.1 EMC sample

The following code will check whether there are participants with an absolute scaled mean indicator (ASMI) Z-score above 3 solely based on EMC sample.

```r
#calculating Z-scores
dataEMC_subset_scaled <- cbind(data_pooled[data_pooled$centerID == "EMC",]
                               [, c("ID", names_covariates, "MoCA_total")],
                               scale(data_pooled[data_pooled$centerID == "EMC",]
                                     [,names_rel_items_abbr]))
#check if MoCA score is below 26
moca_below26_EMC <- dataEMC_subset_scaled$MoCA_total < 26


#calculating mean Z-score of all items per participant
meanitemSS_EMC <- rowMeans(dataEMC_subset_scaled[,c(
                          'EKM60t', 'ERTt', 'TASf1',
                          'FP_Gt', 'CART_Gt', 'SET_Gt',
                          'SNQt', 'RSMSf1', 'MBIt')], na.rm = TRUE)

#scaling the vector of mean Z-scores of all items per participant
SSmeanitemSS_EMC <- scale(meanitemSS_EMC)

# #check if ASMI Z-score > 3
outlier_EMC <- dataEMC_subset_scaled[which(abs(SSmeanitemSS_EMC)>3), "ID"]
length(outlier_EMC)
```

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

The detection method identified 'r length(outlier_EMC)' outlier within the EMC sample.

3.2.2 UMCG sample

The following code will check whether there are participants with an absolute scaled mean indicator (ASMI) Z-score above 3 solely based on UMCG sample.

```
#calculating Z-scores
dataUMCG_subset_scaled <- cbind(data_pooled[data_pooled$centerID == "UMCG",]
                                [, c("ID", names_covariates, "MoCA_total")],
                               scale(data_pooled[data_pooled$centerID == "UMCG",]
                                     [,names_rel_items_abbr]))
#check if MoCA score is below 26
moca_below26_UMCG <- dataUMCG_subset_scaled$MoCA_total < 26


#calculating mean Z-score of all items per participant
meanitemSS_UMCG <- rowMeans(dataUMCG_subset_scaled[,c(
                          'EKM60t', 'ERTt', 'TASf1',
                          'FP_Gt', 'CART_Gt', 'SET_Gt',
                          'SNQt', 'RSMSf1', 'MBIt')], na.rm = TRUE)

#scaling the vector of mean Z-scores of all items per participant
SSmeanitemSS_UMCG <- scale(meanitemSS_UMCG)

# #check if ASMI Z-score > 3
outlier_UMCG <- dataUMCG_subset_scaled[which(abs(SSmeanitemSS_UMCG)>3), "ID"]
length(outlier_UMCG)
```

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

The detection method identified 'r length(outlier_UMCG)' outlier within the EMC sample.

3.2.3 Pooled sample

The following code will check whether there are participants with an absolute scaled mean indicator (ASMI) Z-score above 3 based on the pooled.

```
#calculating Z-scores
datapld_subset_scaled <- cbind(data_pooled[, c("ID",
                                                  names_covariates,
                                                  "MoCA_total")],
                               scale(data_pooled[,names_rel_items_abbr]))
#check if MoCA score is below 26
moca_below26_pld <- datapld_subset_scaled$MoCA_total < 26


#calculating mean Z-score of all items per participant
meanitemSS_pld <- rowMeans(datapld_subset_scaled[,c(
```

```r
  'EKM60t', 'ERTt', 'TASf1',
  'FP_Gt', 'CART_Gt', 'SET_Gt',
  'SNQt', 'RSMSf1', 'MBIt')], na.rm = TRUE)

#scaling the vector of mean Z-scores of all items per participant
SSmeanitemSS_pld <- scale(meanitemSS_pld)

# #check if ASMI Z-score > 3
outlier_pld <- datapld_subset_scaled[which(abs(SSmeanitemSS_pld)>3), "ID"]
length(outlier_pld)
```

```
## [1] 2
```

The detection method identified ‘r length(outlier_pld)’ outliers within the pooled sample.

The following code will check whether the ‘r length(outlier_pld)’ ourliers detected in the pooled sample correspond to the outliers deteced seperately in the EMC sample and UMCG sample

```r
c(outlier_EMC, outlier_UMCG) == outlier_pld
```

```
## [1] TRUE TRUE
```

It does. These two outliers are considered as ‘not belonging to the intended population of interest’ and will be deleted from the analysis dataset.

```r
#deleting the outlier from the pooled-dataset
data_pooled <- data_pooled[-(which(data_pooled$ID==outlier_EMC)),]
data_pooled <- data_pooled[-(which(data_pooled$ID==outlier_UMCG)),]
```

3.3 Normality

3.3.1 Normality - EMC sample

```r
psych::describe(data_pooled[data_pooled$centerID == "EMC",
                            c(names_covariates, "MoCA_total",
                              names_rel_items_abbr)]) %>%
  select(n, min, max, mean, median, sd, skew, kurtosis)
```

```
##                  n  min max  mean median    sd  skew kurtosis
## age             48 21.0  79 47.60  52.50 19.16 -0.09    -1.46
## sex             48  0.0   1  0.65   1.00  0.48 -0.59    -1.68
## education_level 48  2.0   7  5.75   6.00  1.10 -1.10     1.48
## MoCA_total      48 19.0  30 26.10  26.50  2.42 -0.59    -0.08
## EKM60t          46 34.0  58 47.24  47.50  5.02 -0.37     0.15
## ERTt            47 34.0  76 57.21  58.00  9.44 -0.20    -0.55
## TASf1           48  5.0  28 20.08  21.00  5.43 -0.62    -0.04
## FP_Gt           48 10.5  24 20.23  20.50  2.46 -1.16     2.83
## HT_Gt           47  6.0  12 11.26  12.00  1.07 -2.57     9.65
## CART_Gt         39  2.5  12  8.10   8.00  2.67 -0.30    -0.91
## SET_Gt          45  9.0  12 11.40  11.45  0.71 -1.52     2.17
```

```
## SNQt            48 16.0  21 19.23  19.00  1.13 -0.44    -0.28
## RSMSf1          48 11.0  32 22.75  23.00  4.62 -0.57     0.02
## MBIt            48 42.0  70 54.29  54.50  7.09  0.21    -0.66
```

```
normalityCheck_EMC <- psych::describe(data_pooled[data_pooled$centerID == "EMC",
                                                  c(names_covariates, "MoCA_total",
                                                   names_rel_items_abbr)]) %>%
  select(skew, kurtosis)

any(abs(normalityCheck_EMC$skew)>3)
```

```
## [1] FALSE
```

```
any(abs(normalityCheck_EMC$kurtosis)>10)
```

```
## [1] FALSE
```

No absolute skew>3 and no absolute kurtosis>10 is EMC sample after deleting 1 outlier.

3.3.2 Normality - UMCG sample

```
psych::describe(data_pooled[data_pooled$centerID == "UMCG",
                            c(names_covariates, "MoCA_total",
                             names_rel_items_abbr)]) %>%
  select(n, min, max, mean, median, sd, skew, kurtosis)
```

```
##                  n min  max  mean median    sd  skew kurtosis
## age             81  18   66 28.70     22 14.16  1.35     0.12
## sex             81   0    1  0.63      1  0.49 -0.53    -1.74
## education_level 81   3    7  6.01      6  0.60 -1.36     6.61
## MoCA_total       0 Inf -Inf   NaN     NA    NA    NA       NA
## EKM60t          81  38   56 47.06     48  4.50 -0.29    -0.83
## ERTt             0 Inf -Inf   NaN     NA    NA    NA       NA
## TASf1           81   7   28 21.11     21  4.42 -0.83     0.79
## FP_Gt           81  14   24 20.26     21  2.40 -0.72     0.03
## HT_Gt           81   6   12 11.40     12  1.01 -2.64     9.49
## CART_Gt         81   0   12  8.55      9  2.49 -0.53     0.05
## SET_Gt          81   8   12 11.35     12  0.94 -1.44     1.50
## SNQt            81  16   22 19.41     19  1.25 -0.41    -0.12
## RSMSf1           0 Inf -Inf   NaN     NA    NA    NA       NA
## MBIt             0 Inf -Inf   NaN     NA    NA    NA       NA
```

```
normalityCheck_UMCG <- psych::describe(data_pooled[data_pooled$centerID == "UMCG",
                                                   c(names_covariates, "MoCA_total",
                                                    names_rel_items_abbr)]) %>%
  select(skew, kurtosis)

any(abs(normalityCheck_UMCG$skew)>3, na.rm = TRUE)
```

```
## [1] FALSE
```

```r
any(abs(normalityCheck_UMCG$kurtosis)>10, na.rm = TRUE)
```

```
## [1] FALSE
```

No absolute skew>3 and no absolute kurtosis>10 is EMC sample after deleting 1 outlier.

3.3.3 Normality - pooled

```r
psych::describe(data_pooled[, c(names_covariates, "MoCA_total", names_rel_items_abbr)]) %>%
  select(n, min, max, mean, median, sd, skew, kurtosis)
```

```
##                   n  min max  mean median    sd  skew kurtosis
## age             129 18.0  79 35.74   24.0 18.56  0.74    -1.01
## sex             129  0.0   1  0.64    1.0  0.48 -0.56    -1.70
## education_level 129  2.0   7  5.91    6.0  0.83 -1.55     4.63
## MoCA_total       48 19.0  30 26.10   26.5  2.42 -0.59    -0.08
## EKM60t          127 34.0  58 47.13   48.0  4.68 -0.32    -0.30
## ERTt             47 34.0  76 57.21   58.0  9.44 -0.20    -0.55
## TASf1           129  5.0  28 20.73   21.0  4.82 -0.79     0.56
## FP_Gt           129 10.5  24 20.25   20.5  2.41 -0.90     1.23
## HT_Gt           128  6.0  12 11.34   12.0  1.03 -2.64     9.79
## CART_Gt         120  0.0  12  8.40    9.0  2.55 -0.46    -0.28
## SET_Gt          126  8.0  12 11.37   12.0  0.86 -1.53     1.99
## SNQt            129 16.0  22 19.34   19.0  1.21 -0.41    -0.11
## RSMSf1           48 11.0  32 22.75   23.0  4.62 -0.57     0.02
## MBIt             48 42.0  70 54.29   54.5  7.09  0.21    -0.66
```

```r
normalityCheck_pld <- psych::describe(data_pooled[, c(names_covariates, "MoCA_total",
                                                       names_rel_items_abbr)]) %>%
  select(skew, kurtosis)

any(abs(normalityCheck_pld$skew)>3, na.rm = TRUE)
```

```
## [1] FALSE
```

```r
any(abs(normalityCheck_pld$kurtosis)>10, na.rm = TRUE)
```

```
## [1] FALSE
```

No absolute skew>3 and no absolute kurtosis>10 is pooled sample after deleting 2 outliers.

3.4 Descriptives (after outlier removal)

```r
# Descriptives pooled before outlier deletion and/or tranformations
psych::describe(data_pooled[,
                            c(names_covariates,
                              "MoCA_total",
                              names_rel_items_abbr)]) %>%
  select(n, min, max, mean, median, sd, skew, kurtosis)
```

```
##                     n  min max  mean median    sd  skew kurtosis
## age               129 18.0  79 35.74   24.0 18.56  0.74    -1.01
## sex               129  0.0   1  0.64    1.0  0.48 -0.56    -1.70
## education_level 129  2.0   7  5.91    6.0  0.83 -1.55     4.63
## MoCA_total         48 19.0  30 26.10   26.5  2.42 -0.59    -0.08
## EKM60t            127 34.0  58 47.13   48.0  4.68 -0.32    -0.30
## ERTt               47 34.0  76 57.21   58.0  9.44 -0.20    -0.55
## TASf1             129  5.0  28 20.73   21.0  4.82 -0.79     0.56
## FP_Gt             129 10.5  24 20.25   20.5  2.41 -0.90     1.23
## HT_Gt             128  6.0  12 11.34   12.0  1.03 -2.64     9.79
## CART_Gt           120  0.0  12  8.40    9.0  2.55 -0.46    -0.28
## SET_Gt            126  8.0  12 11.37   12.0  0.86 -1.53     1.99
## SNQt              129 16.0  22 19.34   19.0  1.21 -0.41    -0.11
## RSMSf1             48 11.0  32 22.75   23.0  4.62 -0.57     0.02
## MBIt               48 42.0  70 54.29   54.5  7.09  0.21    -0.66
```

4 Methods: Sample comparisons

Statistical comparisons – Welch's t-tests, Chi-square test of independence, and the Mann-Whitney U test – of the EMC sample versus the UMCG sample based on the observed data (before imputation).

```
#age
scAge <- t.test(formula = age ~ centerID, data = data_pooled); scAge
```

```
##
##  Welch Two Sample t-test
##
## data:  age by centerID
## t = 5.9392, df = 77.565, p-value = 7.644e-08
## alternative hypothesis: true difference in means between group EMC and group UMCG is not equal to 0
## 95 percent confidence interval:
##  12.56440 25.23652
## sample estimates:
##  mean in group EMC mean in group UMCG
##           47.60417           28.70370
```

```
cohens_d(formula = age ~ centerID, data = data_pooled, var.equal = FALSE)
```

```
## # A tibble: 1 x 7
##   .y.   group1 group2 effsize    n1    n2 magnitude
## * <chr> <chr>  <chr>    <dbl> <int> <int> <ord>
## 1 age   EMC    UMCG      1.12    48    81 large
```

```
#sex
table(data_pooled$sex, data_pooled$centerID)
```

```
##
##     EMC UMCG
##   0  17   30
##   1  31   51
```

```r
scSex <- chisq.test(table(data_pooled$sex, data_pooled$centerID)); scSex
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(data_pooled$sex, data_pooled$centerID)
## X-squared = 2.4514e-31, df = 1, p-value = 1
```

```r
sqrt(scSex$statistic[[1]] / (nrow(data_pooled)*5))
```

```
## [1] 1.949511e-17
```

```r
#education level
scEduc <- rstatix:: wilcox_test(data_pooled, education_level~centerID)
rstatix:: wilcox_effsize(data_pooled, education_level~centerID)
```

```
## # A tibble: 1 x 7
##   .y.             group1 group2 effsize    n1    n2 magnitude
## * <chr>           <chr>  <chr>    <dbl> <int> <int> <ord>
## 1 education_level EMC    UMCG     0.102    48    81 small
```

```r
#Ekman
scEKM <- t.test(formula = EKM60t ~ centerID, data = data_pooled); scEKM
```

```
##
##  Welch Two Sample t-test
##
## data:  EKM60t by centerID
## t = 0.1987, df = 85.501, p-value = 0.843
## alternative hypothesis: true difference in means between group EMC and group UMCG is not equal to 0
## 95 percent confidence interval:
##  -1.597633  1.952437
## sample estimates:
##  mean in group EMC mean in group UMCG
##           47.23913           47.06173
```

```r
cohens_d(formula = EKM60t ~ centerID, data = data_pooled, var.equal = FALSE)
```

```
## # A tibble: 1 x 7
##   .y.    group1 group2 effsize    n1    n2 magnitude
## * <chr>  <chr>  <chr>    <dbl> <int> <int> <ord>
## 1 EKM60t EMC    UMCG    0.0372    46    81 negligible
```

```r
#TAS
scTAS<- t.test(formula = TASf1 ~ centerID, data = data_pooled); scTAS
```

```
##
##  Welch Two Sample t-test
##
```

```
## data:  TASf1 by centerID
## t = -1.1121, df = 83.573, p-value = 0.2693
## alternative hypothesis: true difference in means between group EMC and group UMCG is not equal to 0
## 95 percent confidence interval:
##  -2.8657047  0.8101491
## sample estimates:
##  mean in group EMC mean in group UMCG
##           20.08333           21.11111
```

```
cohens_d(formula = TASf1 ~ centerID, data = data_pooled, var.equal = FALSE)
```

```
## # A tibble: 1 x 7
##   .y.   group1 group2 effsize    n1    n2 magnitude
## * <chr> <chr>  <chr>    <dbl> <int> <int> <ord>
## 1 TASf1 EMC    UMCG    -0.208    48    81 small
```

```
#FP
scFP<- t.test(formula = FP_Gt ~ centerID, data = data_pooled); scFP
```

```
##
##  Welch Two Sample t-test
##
## data:  FP_Gt by centerID
## t = -0.06772, df = 96.615, p-value = 0.9461
## alternative hypothesis: true difference in means between group EMC and group UMCG is not equal to 0
## 95 percent confidence interval:
##  -0.9120807  0.8518955
## sample estimates:
##  mean in group EMC mean in group UMCG
##           20.22917           20.25926
```

```
cohens_d(formula = FP_Gt ~ centerID, data = data_pooled, var.equal = FALSE)
```

```
## # A tibble: 1 x 7
##   .y.   group1 group2 effsize    n1    n2 magnitude
## * <chr> <chr>  <chr>    <dbl> <int> <int> <ord>
## 1 FP_Gt EMC    UMCG   -0.0124    48    81 negligible
```

```
#CART
scCART<- t.test(formula = CART_Gt ~ centerID, data = data_pooled); scCART
```

```
##
##  Welch Two Sample t-test
##
## data:  CART_Gt by centerID
## t = -0.87672, df = 70.697, p-value = 0.3836
## alternative hypothesis: true difference in means between group EMC and group UMCG is not equal to 0
## 95 percent confidence interval:
##  -1.4631069  0.5694697
## sample estimates:
##  mean in group EMC mean in group UMCG
##           8.102564           8.549383
```

```r
cohens_d(formula = CART_Gt ~ centerID, data = data_pooled, var.equal = FALSE)
```

```
## # A tibble: 1 x 7
##   .y.     group1 group2 effsize    n1    n2 magnitude
## * <chr>   <chr>  <chr>    <dbl> <int> <int> <ord>
## 1 CART_Gt EMC    UMCG    -0.173    39    81 negligible
```

```r
#SET
scSET<- t.test(formula = SET_Gt ~ centerID, data = data_pooled); scSET
```

```
##
##  Welch Two Sample t-test
##
## data:  SET_Gt by centerID
## t = 0.39442, df = 112.48, p-value = 0.694
## alternative hypothesis: true difference in means between group EMC and group UMCG is not equal to 0
## 95 percent confidence interval:
##  -0.2356628  0.3528113
## sample estimates:
##  mean in group EMC mean in group UMCG
##           11.40425           11.34568
```

```r
cohens_d(formula = SET_Gt ~ centerID, data = data_pooled, var.equal = FALSE)
```

```
## # A tibble: 1 x 7
##   .y.    group1 group2 effsize    n1    n2 magnitude
## * <chr>  <chr>  <chr>    <dbl> <int> <int> <ord>
## 1 SET_Gt EMC    UMCG    0.0704    45    81 negligible
```

```r
#SNQ
scSNQ<- t.test(formula = SNQt ~ centerID, data = data_pooled); scSNQ
```

```
##
##  Welch Two Sample t-test
##
## data:  SNQt by centerID
## t = -0.82939, df = 106.73, p-value = 0.4087
## alternative hypothesis: true difference in means between group EMC and group UMCG is not equal to 0
## 95 percent confidence interval:
##  -0.6042762  0.2477947
## sample estimates:
##  mean in group EMC mean in group UMCG
##           19.22917           19.40741
```

```r
cohens_d(formula = SNQt ~ centerID, data = data_pooled, var.equal = FALSE)
```

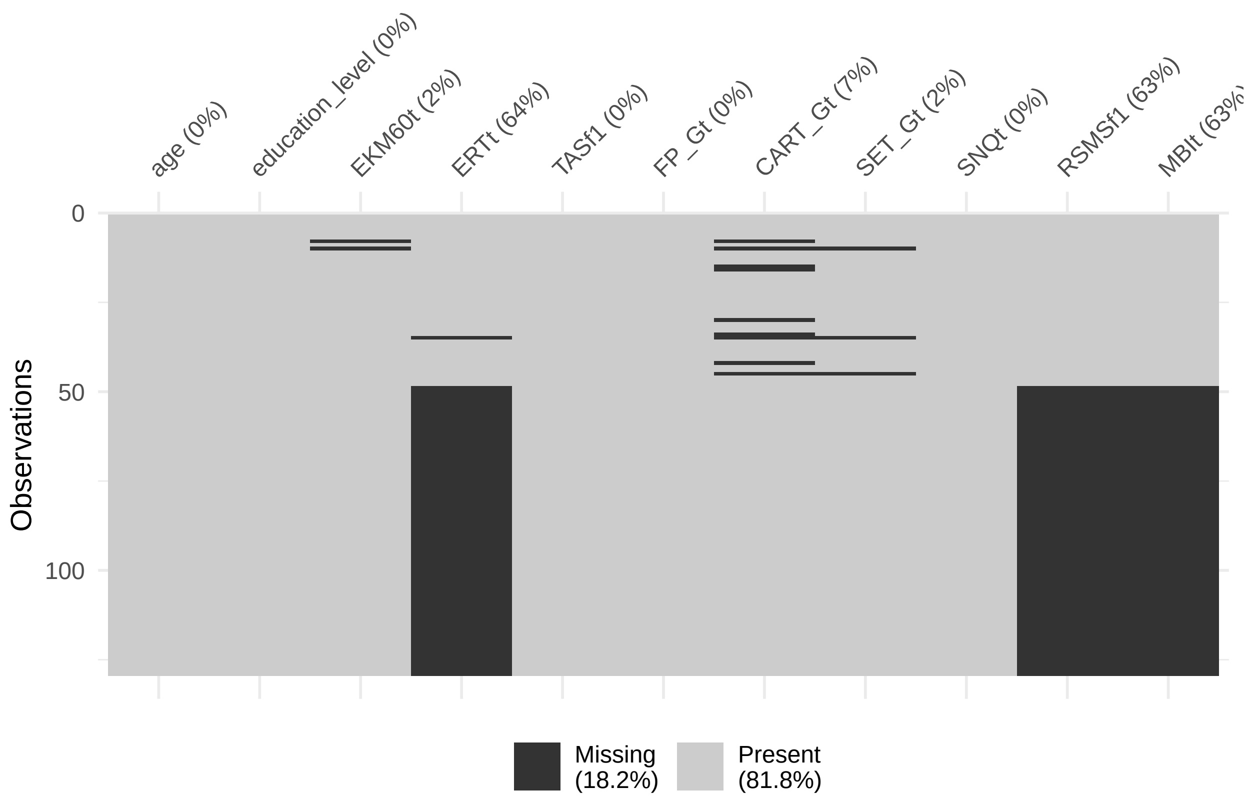
```
## # A tibble: 1 x 7
##   .y.   group1 group2 effsize    n1    n2 magnitude
## * <chr> <chr>  <chr>    <dbl> <int> <int> <ord>
## 1 SNQt  EMC    UMCG    -0.149    48    81 negligible
```

5 Methods: Stochastic imputation

5.1 Missing data

At this point, we have the following missing values.

```
vis_miss(data_pooled[,c("age", "education_level",
                        'EKM60t', 'ERTt', 'TASf1',
                        'FP_Gt', 'CART_Gt', 'SET_Gt',
                        'SNQt', 'RSMSf1', 'MBIt')])
```



5.2 Correlations

And the following zero order paired correlation (respectively: corr, N, p-values).

```
Hmisc::rcorr(as.matrix(data_pooled[, c("age", "education_level",
                                       'EKM60t', 'ERTt', 'TASf1',
                                       'FP_Gt', 'CART_Gt', 'SET_Gt',
                                       'SNQt', 'RSMSf1', 'MBIt')]),
             type = "spearman")
```

```
##                    age education_level EKM60t  ERTt TASf1 FP_Gt CART_Gt SET_Gt
## age               1.00           -0.09  -0.02 -0.47  0.06 -0.14    0.08  -0.19
```

```
## education_level -0.09           1.00   0.23  0.46  0.10  0.18    0.36   0.17
## EKM60t          -0.02           0.23   1.00  0.59  0.14  0.26    0.19   0.05
## ERTt            -0.47           0.46   0.59  1.00  0.16  0.38    0.21   0.27
## TASf1            0.06           0.10   0.14  0.16  1.00  0.06    0.01   0.04
## FP_Gt           -0.14           0.18   0.26  0.38  0.06  1.00    0.18   0.17
## CART_Gt          0.08           0.36   0.19  0.21  0.01  0.18    1.00   0.13
## SET_Gt          -0.19           0.17   0.05  0.27  0.04  0.17    0.13   1.00
## SNQt            -0.01           0.19   0.13  0.08  0.08  0.24    0.07   0.06
## RSMSf1          -0.08          -0.06   0.23  0.13  0.27  0.09    0.23  -0.06
## MBIt             0.30           0.00   0.19  0.13  0.20 -0.01    0.30  -0.03
##                  SNQt RSMSf1  MBIt
## age             -0.01  -0.08  0.30
## education_level  0.19  -0.06  0.00
## EKM60t           0.13   0.23  0.19
## ERTt             0.08   0.13  0.13
## TASf1            0.08   0.27  0.20
## FP_Gt            0.24   0.09 -0.01
## CART_Gt          0.07   0.23  0.30
## SET_Gt           0.06  -0.06 -0.03
## SNQt             1.00  -0.23  0.10
## RSMSf1          -0.23   1.00 -0.09
## MBIt             0.10  -0.09  1.00
##
## n
##                 age education_level EKM60t ERTt TASf1 FP_Gt CART_Gt SET_Gt SNQt
## age             129             129    127   47   129   129     120    126  129
## education_level 129             129    127   47   129   129     120    126  129
## EKM60t          127             127    127   45   127   127     120    125  127
## ERTt             47              47     45   47    47    47      39     45   47
## TASf1           129             129    127   47   129   129     120    126  129
## FP_Gt           129             129    127   47   129   129     120    126  129
## CART_Gt         120             120    120   39   120   120     120    120  120
## SET_Gt          126             126    125   45   126   126     120    126  126
## SNQt            129             129    127   47   129   129     120    126  129
## RSMSf1           48              48     46   47    48    48      39     45   48
## MBIt             48              48     46   47    48    48      39     45   48
##                 RSMSf1 MBIt
## age                 48   48
## education_level     48   48
## EKM60t              46   46
## ERTt                47   47
## TASf1               48   48
## FP_Gt               48   48
## CART_Gt             39   39
## SET_Gt              45   45
## SNQt                48   48
## RSMSf1              48   48
## MBIt                48   48
##
## P
##                 age    education_level EKM60t ERTt   TASf1  FP_Gt  CART_Gt
## age                    0.2964          0.8498 0.0008 0.5149 0.1108 0.3629
## education_level 0.2964                 0.0108 0.0011 0.2542 0.0358 0.0000
## EKM60t          0.8498 0.0108                 0.0000 0.1110 0.0035 0.0347
```

```
## ERTt            0.0008 0.0011          0.0000        0.2912 0.0093 0.1951
## TASf1           0.5149 0.2542          0.1110 0.2912        0.4645 0.9166
## FP_Gt           0.1108 0.0358          0.0035 0.0093 0.4645        0.0544
## CART_Gt         0.3629 0.0000          0.0347 0.1951 0.9166 0.0544
## SET_Gt          0.0340 0.0645          0.6146 0.0738 0.6720 0.0605 0.1560
## SNQt            0.9125 0.0280          0.1441 0.5789 0.3492 0.0073 0.4617
## RSMSf1          0.5954 0.7063          0.1205 0.3964 0.0635 0.5606 0.1509
## MBIt            0.0353 0.9846          0.2060 0.3663 0.1678 0.9730 0.0651
##                 SET_Gt SNQt   RSMSf1 MBIt
## age             0.0340 0.9125 0.5954 0.0353
## education_level 0.0645 0.0280 0.7063 0.9846
## EKM60t          0.6146 0.1441 0.1205 0.2060
## ERTt            0.0738 0.5789 0.3964 0.3663
## TASf1           0.6720 0.3492 0.0635 0.1678
## FP_Gt           0.0605 0.0073 0.5606 0.9730
## CART_Gt         0.1560 0.4617 0.1509 0.0651
## SET_Gt                 0.5168 0.7084 0.8448
## SNQt            0.5168        0.1213 0.5163
## RSMSf1          0.7084 0.1213        0.5422
## MBIt            0.8448 0.5163 0.5422
```

5.3 Imputation predictor selection

5.3.1 Predictor selection ERTt

```
matcor <- Hmisc:: rcorr(as.matrix(data_pooled[, c("age", "education_level",
                                          'EKM60t', 'ERTt', 'TASf1',
                                          'FP_Gt', 'CART_Gt', 'SET_Gt',
                                          'SNQt', 'RSMSf1', 'MBIt')]),
                        type = "spearman")

#check whether sex is also a significant predictor
sexERTt <- lm(ERTt ~ sex, data = data_pooled)
summary(sexERTt)$coefficients["sex", "Pr(>|t|)"] #Not significant; p = 0.218
```

```
## [1] 0.2181282
```

```
#selecting predictors based on significant correlation
sigpred_ERTt <- matcor$r["ERTt",matcor$P["ERTt",]<.05]; sigpred_ERTt
```

```
##             age education_level          EKM60t            <NA>           FP_Gt
##      -0.4733335       0.4620705       0.5866877              NA       0.3753067
```

5.3.2 Predictor selection RSMSf1

Lets check the correlations of RSMSf1 with other relevant variables.

```
#Check whether sex is also a significant predictor
sexRSMSf1 <- lm(RSMSf1 ~ sex, data = data_pooled)
summary(sexRSMSf1)$coefficients["sex", "Pr(>|t|)"] #Not significant, p = 0.0867
```

```
## [1] 0.08671293
```

```r
#Check numerical correlations
sigpred_RSMSf1 <- matcor$r["RSMSf1",matcor$P["RSMSf1",]<.05];
sigpred_RSMSf1 #less than three predictors
```

```
## [1] NA
```

```r
# increment alpha with .05
sigpred_RSMSf1 <- matcor$r["RSMSf1",matcor$P["RSMSf1",]<.1]
#adds sex to the predictors
sigpred_RSMSf1$sex <- summary(sexRSMSf1)$coefficients["sex", "Pr(>|t|)"]
```

```
## Warning in sigpred_RSMSf1$sex <- summary(sexRSMSf1)$coefficients["sex", :
## Coercing LHS to a list
```

```r
sigpred_RSMSf1 #less than three predictors
```

```
## $TASf1
## [1] 0.2700051
##
## $<NA>
## [1] NA
##
## $sex
## [1] 0.08671293
```

```r
# increment alpha with .05
sigpred_RSMSf1 <- matcor$r["RSMSf1",matcor$P["RSMSf1",]<.15]
#adds sex to the predictors
sigpred_RSMSf1$sex <- summary(sexRSMSf1)$coefficients["sex", "Pr(>|t|)"]
```

```
## Warning in sigpred_RSMSf1$sex <- summary(sexRSMSf1)$coefficients["sex", :
## Coercing LHS to a list
```

```r
sigpred_RSMSf1 # 4 predictors
```

```
## $EKM60t
## [1] 0.2321768
##
## $TASf1
## [1] 0.2700051
##
## $SNQt
## [1] -0.2266534
##
## $<NA>
## [1] NA
##
## $sex
## [1] 0.08671293
```

5.3.3 Predictor selection MBIt

Lets check the correlations of MBIt with other relevant variables.

```
#Check whether sex is also a significant predictor
sexMBIt <- lm(MBIt ~ sex, data = data_pooled)
summary(sexMBIt)$coefficients["sex", "Pr(>|t|)"] #Not significant, p = 0.3339
```

```
## [1] 0.3339226
```

```
#Check numerical correlations
sigpred_MBIt <- matcor$r["MBIt",matcor$P["MBIt",]<.05];
sigpred_MBIt # only one sign predictor
```

```
##       age      <NA>
## 0.3045774        NA
```

```
# increment alpha with .05
sigpred_MBIt <- matcor$r["MBIt",matcor$P["MBIt",]<.1];
sigpred_MBIt # only 2 predictors
```

```
##       age   CART_Gt      <NA>
## 0.3045774 0.2983429        NA
```

```
# increment alpha with .05
sigpred_MBIt <- matcor$r["MBIt",matcor$P["MBIt",]<.15];
sigpred_MBIt # only 2 predictors
```

```
##       age   CART_Gt      <NA>
## 0.3045774 0.2983429        NA
```

```
# increment alpha with .05
sigpred_MBIt <- matcor$r["MBIt",matcor$P["MBIt",]<.2];
sigpred_MBIt # 3 predictors
```

```
##       age     TASf1   CART_Gt      <NA>
## 0.3045774 0.2023403 0.2983429        NA
```

5.4 Imputation of ERTt, RSMSf, MBIt

```
data_pooled_subset <- data_pooled[,c("ID", "centerID", "sex", "age", "education_level",
                                      'EKM60t', 'ERTt', 'TASf1',
                                      'FP_Gt', 'CART_Gt', 'SET_Gt',
                                      'SNQt', 'RSMSf1', 'MBIt')]

predictor_matrix <- make.predictorMatrix(data_pooled_subset)
predictor_matrix[] <- 0
```

```r
#assign the significant predictors to the prediction matrix
predictor_matrix["ERTt", ] <- colnames(predictor_matrix) %in%
  names(sigpred_ERTt)[!is.na(names(sigpred_ERTt))]
predictor_matrix["RSMSf1", ] <- colnames(predictor_matrix) %in%
  names(sigpred_RSMSf1)[!is.na(names(sigpred_RSMSf1))]
predictor_matrix["MBIt", ] <- colnames(predictor_matrix) %in%
  names(sigpred_MBIt)[!is.na(names(sigpred_MBIt))]



#assign imputation medhods
imputation_methods <- make.method(data_pooled_subset)
imputation_methods[] <- ""  # Set all methods to "" initially
# Use "pmm" for ERTt, ESMf1, and MBIt
imputation_methods[c("ERTt", "RSMSf1", "MBIt")] <- "pmm"

#imputation
data_pooled_subset_imp_s1 <- mice(data_pooled_subset,
                               method = imputation_methods,
                               predictorMatrix = predictor_matrix,
                               m = 1,
                               seed = 2)
```

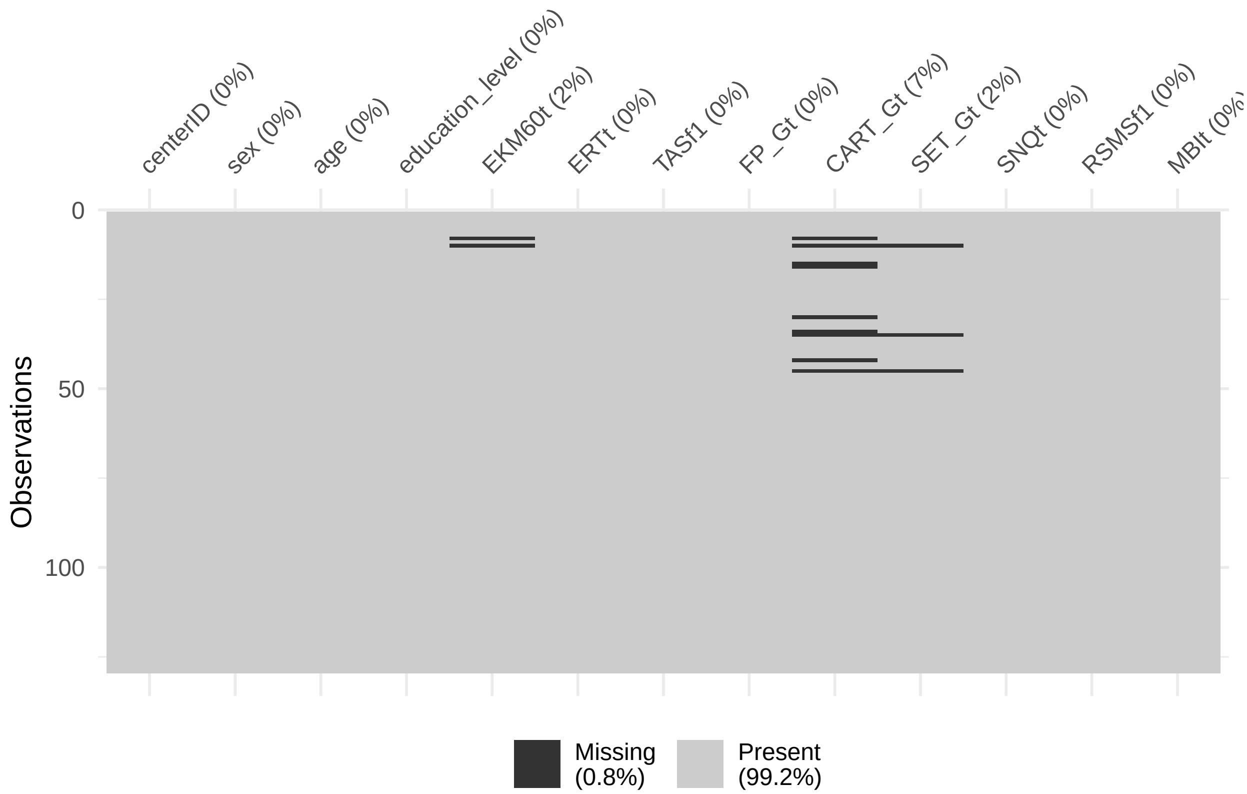
```
##
##  iter imp variable
##   1   1  ERTt  RSMSf1  MBIt
##   2   1  ERTt  RSMSf1  MBIt
##   3   1  ERTt  RSMSf1  MBIt
##   4   1  ERTt  RSMSf1  MBIt
##   5   1  ERTt  RSMSf1  MBIt
```

```r
data_pooled_subset_imp <-  complete(data_pooled_subset_imp_s1, 1)
```

5.5 Mid evaluation of missing data

At this point, we have the remaining missing data:

```r
vis_miss(data_pooled_subset_imp[,-1])
```



We will impute the remaining missing data, again with a stochastic approach based on relations in the available data using predicted mean matching.

5.6 Imputation predictor selection part 2

5.6.1 Predictor selection CART_Gt

Selecting related predictors for imputing CART_Gt.

```
matcor2 <- Hmisc:: rcorr(as.matrix(data_pooled_subset_imp[, c("age", "education_level",
                                        'EKM60t', 'ERTt', 'TASf1',
                                        'FP_Gt', 'CART_Gt', 'SET_Gt',
                                        'SNQt', 'RSMSf1', 'MBIt')]),
                         type = "spearman")

#Check whether sex is also a significant predictor
sexCART_Gt <- lm(CART_Gt ~ sex, data = data_pooled_subset_imp)
summary(sexCART_Gt)$coefficients["sex", "Pr(>|t|)"] #Significant
```

```
## [1] 0.0003290575
```

```
sigpred_CART_Gt <- matcor2$r["CART_Gt",matcor2$P["CART_Gt",]<.05]
# add sex since it is significant
sigpred_CART_Gt$sex <- summary(sexCART_Gt)$coefficients["sex", "Pr(>|t|)"]
```

```
## Warning in sigpred_CART_Gt$sex <- summary(sexCART_Gt)$coefficients["sex", :
## Coercing LHS to a list
```

```r
sigpred_CART_Gt #significant correlations of CART_Gt
```

```
## $education_level
## [1] 0.3649806
##
## $EKM60t
## [1] 0.1930374
##
## $<NA>
## [1] NA
##
## $RSMSf1
## [1] 0.187745
##
## $MBIt
## [1] 0.2487448
##
## $sex
## [1] 0.0003290575
```

5.6.2 Predictor selection SET_Gt

Selecting related predictors for imputing SET_Gt.

```r
#Check whether sex is also a significant predictor
sexSET_Gt <- lm(SET_Gt ~ sex, data = data_pooled_subset_imp)
summary(sexSET_Gt)$coefficients["sex", "Pr(>|t|)"] #Not significant
```

```
## [1] 0.1178248
```

```r
#< 3 predictors selected
sigpred_SET_Gt <- matcor2$r["SET_Gt",matcor2$P["SET_Gt",]<.05]; sigpred_SET_Gt
```

```
##        age       <NA>
## -0.1890881         NA
```

```r
# increment alpha with .05
#Selected predictors
sigpred_SET_Gt <- matcor2$r["SET_Gt",matcor2$P["SET_Gt",]<.1]; sigpred_SET_Gt
```

```
##             age education_level            ERTt           FP_Gt            <NA>
##      -0.1890881       0.1651829       0.1639694       0.1676832              NA
```

5.6.3 Predictor selection EKM60t

Lets check the correlations of EKM60t with other relevant variables.

```r
#Check whether sex is also a significant predictor
sexEKM60t <- lm(EKM60t ~ sex, data = data_pooled_subset_imp)
summary(sexEKM60t)$coefficients["sex", "Pr(>|t|)"] #Not significant
```

```
## [1] 0.2253531
```

```r
#significant correlations of EKM60t
sigpred_EKM60t <- matcor2$r["EKM60t",matcor2$P["EKM60t",]<.05]; sigpred_EKM60t
```

```
## education_level            <NA>           ERTt          FP_Gt        CART_Gt
##       0.2253768              NA      0.5872751      0.2574957      0.1930374
##          RSMSf1
##       0.2370015
```

5.6.4 Predictor selection ERTt (second round)

Lets check the correlations of ERTt with other relevant variables.

```r
#Check whether sex is also a significant predictor
sexERTt_2nd <- lm(ERTt ~ sex, data = data_pooled_subset_imp)
summary(sexERTt_2nd)$coefficients["sex", "Pr(>|t|)"] #Not significant
```

```
## [1] 0.5954151
```

```r
#significant correlations of ERTt
sigpred_ERTt_2nd <- matcor2$r["ERTt",matcor2$P["ERTt",]<.05]; sigpred_ERTt_2nd
```

```
##             age education_level          EKM60t            <NA>           FP_Gt
##      -0.3284992       0.2868658       0.5872751              NA       0.2750760
```

5.7 Imputation of CART_Gt, SET_Gt, EKM60t, ERTt (2nd imputation round)

```r
data_pooled_subset_imp2 <- data_pooled_subset_imp

predictor_matrix2 <- make.predictorMatrix(data_pooled_subset_imp)
predictor_matrix2[] <- 0

#assign the significant predictors to the prediction matrix
predictor_matrix2["CART_Gt", ] <-
  colnames(predictor_matrix2) %in% names(sigpred_CART_Gt)[!is.na(names(sigpred_CART_Gt))]
predictor_matrix2["SET_Gt", ] <-
  colnames(predictor_matrix2) %in% names(sigpred_SET_Gt)[!is.na(names(sigpred_SET_Gt))]
predictor_matrix2["EKM60t", ] <-
  colnames(predictor_matrix2) %in% names(sigpred_EKM60t)[!is.na(names(sigpred_EKM60t))]
predictor_matrix2["ERTt", ] <-
  colnames(predictor_matrix2) %in% names(sigpred_ERTt_2nd)[!is.na(names(sigpred_ERTt_2nd))]
```

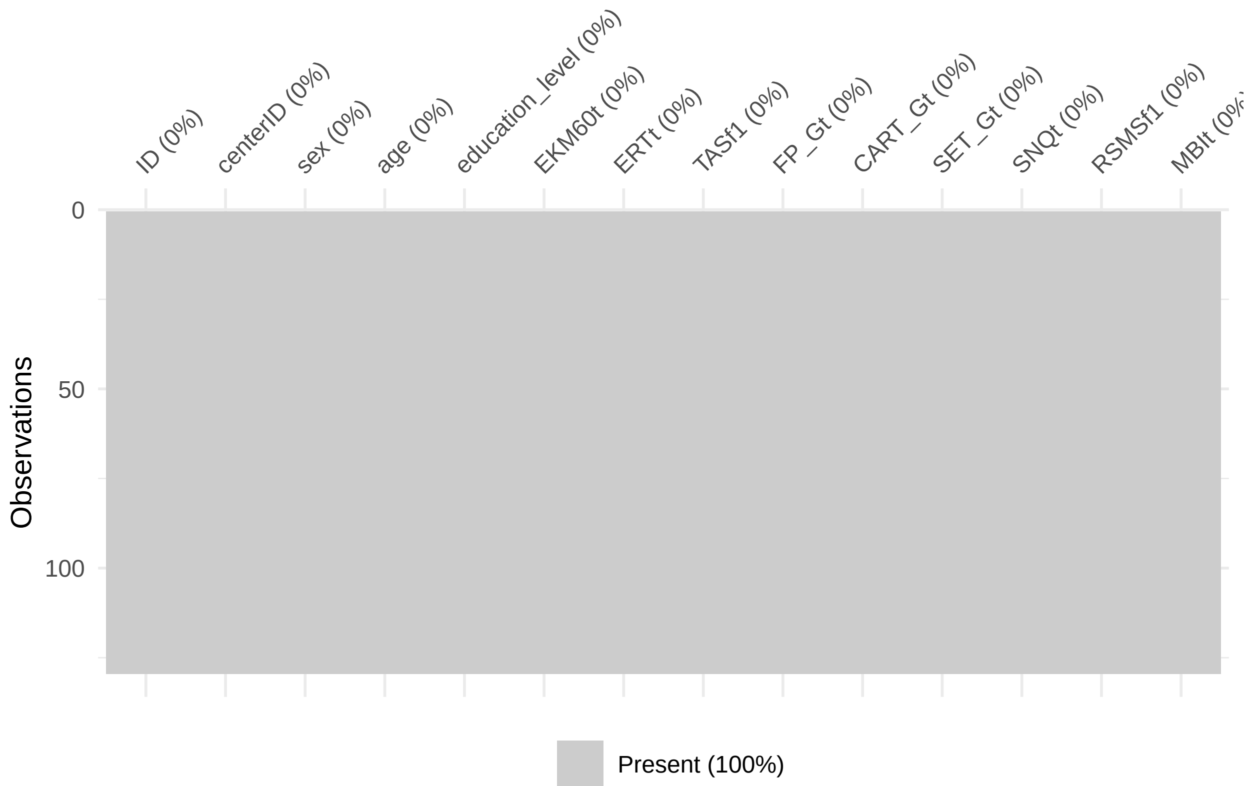
```r
#assign imputation methods
imputation_methods2 <- make.method(data_pooled_subset_imp)
imputation_methods2[] <- ""  # Set all methods to "" initially
imputation_methods2[c("CART_Gt", "SET_Gt", "EKM60t", "ERTt")] <- "pmm"  # Use "pmm" for ERTt

#imputation
data_pooled_subset_imp2_s1 <- mice(data_pooled_subset_imp,
                              method = imputation_methods2,
                              predictorMatrix = predictor_matrix2,
                              m = 1,
                              seed = 2)
```

```
##
##  iter imp variable
##   1   1  EKM60t  CART_Gt  SET_Gt
##   2   1  EKM60t  CART_Gt  SET_Gt
##   3   1  EKM60t  CART_Gt  SET_Gt
##   4   1  EKM60t  CART_Gt  SET_Gt
##   5   1  EKM60t  CART_Gt  SET_Gt
```

```r
data_pooled_subset_imp2 <- complete(data_pooled_subset_imp2_s1, 1)
```

5.7.1 Check after imputation part 2

```r
vis_miss(data_pooled_subset_imp2) #missing data check
```



No missing values anymore

6 Methods: Final dataset

6.1 Characteristics

```
data_final <- data_pooled_subset_imp2
data_final <- merge(data_final, data_pooled[,c('ID', 'TQ_total')], by = "ID", all.x = TRUE)
colnames(data_final)[colnames(data_final)=="TQ_total"] <- "TQ"

colnames(data_final) <- c("ID", "centerID", "sex", "age", "education_level",
                          "EKM", "ERT", "TASf1",
                          "FP", "CART", "SET",
                          "SNQ", "RSMSf1", "MBI",
                          "TQ")

descriptives_data_final <- psych::describe(data_final[,-1]) %>%
  select(n, min, max, mean, median, sd, skew, kurtosis); descriptives_data_final
```

```
##                    n  min max  mean median    sd  skew kurtosis
## centerID*        129  1.0   2  1.63    2.0  0.49 -0.52    -1.74
## sex              129  0.0   1  0.64    1.0  0.48 -0.56    -1.70
## age              129 18.0  79 35.74   24.0 18.56  0.74    -1.01
```

```
## education_level 129  2.0   7  5.91    6.0  0.83 -1.55     4.63
## EKM             129 34.0  58 47.12   48.0  4.67 -0.32    -0.33
## ERT             129 34.0  76 59.64   61.0  8.87 -0.42    -0.36
## TASf1           129  5.0  28 20.73   21.0  4.82 -0.79     0.56
## FP              129 10.5  24 20.25   20.5  2.41 -0.90     1.23
## CART            129  0.0  12  8.35    8.5  2.52 -0.42    -0.31
## SET             129  8.0  12 11.35   12.0  0.88 -1.51     1.84
## SNQ             129 16.0  22 19.34   19.0  1.21 -0.41    -0.11
## RSMSf1          129 11.0  32 23.21   24.0  4.86 -0.70     0.37
## MBI             129 42.0  70 51.88   52.0  7.03  0.34    -0.76
## TQ               48 10.0  18 14.65   15.0  2.23 -0.20    -1.00
```

```r
corr_data_final <- Hmisc:: rcorr(as.matrix(data_final[,-c(1:2)]), type = "spearman")
corr_data_final
```

```
##                   sex   age education_level   EKM   ERT TASf1    FP  CART   SET
## sex              1.00 -0.07           -0.17  0.12  0.06 -0.03 -0.10 -0.35 -0.17
## age             -0.07  1.00           -0.09 -0.01 -0.33  0.06 -0.14  0.02 -0.21
## education_level -0.17 -0.09            1.00  0.21  0.29  0.10  0.18  0.40  0.20
## EKM              0.12 -0.01            0.21  1.00  0.59  0.15  0.24  0.20  0.06
## ERT              0.06 -0.33            0.29  0.59  1.00  0.07  0.28  0.14  0.17
## TASf1           -0.03  0.06            0.10  0.15  0.07  1.00  0.06  0.02  0.03
## FP              -0.10 -0.14            0.18  0.24  0.28  0.06  1.00  0.22  0.19
## CART            -0.35  0.02            0.40  0.20  0.14  0.02  0.22  1.00  0.17
## SET             -0.17 -0.21            0.20  0.06  0.17  0.03  0.19  0.17  1.00
## SNQ              0.03 -0.01            0.19  0.11  0.10  0.08  0.24  0.10  0.07
## RSMSf1          -0.34  0.03            0.11  0.24  0.15  0.07  0.09  0.19  0.06
## MBI              0.02  0.43            0.03  0.03 -0.08  0.16 -0.06  0.19 -0.01
## TQ               0.12 -0.08            0.30  0.42  0.30  0.07  0.27  0.18  0.17
##                   SNQ RSMSf1   MBI    TQ
## sex              0.03  -0.34  0.02  0.12
## age             -0.01   0.03  0.43 -0.08
## education_level  0.19   0.11  0.03  0.30
## EKM              0.11   0.24  0.03  0.42
## ERT              0.10   0.15 -0.08  0.30
## TASf1            0.08   0.07  0.16  0.07
## FP               0.24   0.09 -0.06  0.27
## CART             0.10   0.19  0.19  0.18
## SET              0.07   0.06 -0.01  0.17
## SNQ              1.00  -0.17 -0.02 -0.08
## RSMSf1          -0.17   1.00  0.01  0.17
## MBI             -0.02   0.01  1.00 -0.08
## TQ              -0.08   0.17 -0.08  1.00
##
## n
##                 sex age education_level EKM ERT TASf1  FP CART SET SNQ RSMSf1
## sex             129 129             129 129 129   129 129  129 129 129    129
## age             129 129             129 129 129   129 129  129 129 129    129
## education_level 129 129             129 129 129   129 129  129 129 129    129
## EKM             129 129             129 129 129   129 129  129 129 129    129
## ERT             129 129             129 129 129   129 129  129 129 129    129
## TASf1           129 129             129 129 129   129 129  129 129 129    129
## FP              129 129             129 129 129   129 129  129 129 129    129
## CART            129 129             129 129 129   129 129  129 129 129    129
```

```
## SET             129 129           129 129 129   129 129   129 129 129    129
## SNQ             129 129           129 129 129   129 129   129 129 129    129
## RSMSf1          129 129           129 129 129   129 129   129 129 129    129
## MBI             129 129           129 129 129   129 129   129 129 129    129
## TQ               48  48            48  48  48    48  48    48  48  48     48
##                 MBI TQ
## sex             129 48
## age             129 48
## education_level 129 48
## EKM             129 48
## ERT             129 48
## TASf1           129 48
## FP              129 48
## CART            129 48
## SET             129 48
## SNQ             129 48
## RSMSf1          129 48
## MBI             129 48
## TQ               48 48
##
## P
##                 sex    age    education_level EKM    ERT    TASf1  FP
## sex                    0.4536 0.0550          0.1789 0.5163 0.7604 0.2429
## age             0.4536        0.2964          0.8972 0.0001 0.5149 0.1108
## education_level 0.0550 0.2964                 0.0188 0.0010 0.2542 0.0358
## EKM             0.1789 0.8972 0.0188                 0.0000 0.0836 0.0059
## ERT             0.5163 0.0001 0.0010          0.0000        0.4495 0.0016
## TASf1           0.7604 0.5149 0.2542          0.0836 0.4495        0.4645
## FP              0.2429 0.1108 0.0358          0.0059 0.0016 0.4645
## CART            0.0000 0.7981 0.0000          0.0254 0.1095 0.8046 0.0141
## SET             0.0603 0.0185 0.0240          0.4845 0.0529 0.7024 0.0356
## SNQ             0.7112 0.9125 0.0280          0.2184 0.2523 0.3492 0.0073
## RSMSf1          0.0000 0.7135 0.2174          0.0055 0.0920 0.4428 0.3364
## MBI             0.8016 0.0000 0.7219          0.6962 0.3393 0.0753 0.5015
## TQ              0.4198 0.5811 0.0383          0.0030 0.0395 0.6353 0.0641
##                 CART   SET    SNQ    RSMSf1 MBI    TQ
## sex             0.0000 0.0603 0.7112 0.0000 0.8016 0.4198
## age             0.7981 0.0185 0.9125 0.7135 0.0000 0.5811
## education_level 0.0000 0.0240 0.0280 0.2174 0.7219 0.0383
## EKM             0.0254 0.4845 0.2184 0.0055 0.6962 0.0030
## ERT             0.1095 0.0529 0.2523 0.0920 0.3393 0.0395
## TASf1           0.8046 0.7024 0.3492 0.4428 0.0753 0.6353
## FP              0.0141 0.0356 0.0073 0.3364 0.5015 0.0641
## CART                   0.0577 0.2384 0.0354 0.0340 0.2223
## SET             0.0577        0.4363 0.4961 0.8714 0.2501
## SNQ             0.2384 0.4363        0.0513 0.7897 0.5986
## RSMSf1          0.0354 0.4961 0.0513        0.8673 0.2444
## MBI             0.0340 0.8714 0.7897 0.8673        0.5859
## TQ              0.2223 0.2501 0.5986 0.2444 0.5859
```

```r
corr_data_final_r <- round(as.data.frame(corr_data_final$r), 3); corr_data_final_r
```

```
##                    sex    age education_level    EKM    ERT  TASf1     FP
## sex              1.000 -0.067          -0.169  0.119  0.058 -0.027 -0.104
```

```
## age             -0.067  1.000          -0.093 -0.011 -0.328  0.058 -0.141
## education_level -0.169 -0.093           1.000  0.207  0.287  0.101  0.185
## EKM              0.119 -0.011           0.207  1.000  0.593  0.153  0.241
## ERT              0.058 -0.328           0.287  0.593  1.000  0.067  0.275
## TASf1           -0.027  0.058           0.101  0.153  0.067  1.000  0.065
## FP              -0.104 -0.141           0.185  0.241  0.275  0.065  1.000
## CART            -0.346  0.023           0.404  0.197  0.142  0.022  0.216
## SET             -0.166 -0.207           0.199  0.062  0.171  0.034  0.185
## SNQ              0.033 -0.010           0.193  0.109  0.102  0.083  0.235
## RSMSf1          -0.342  0.033           0.109  0.243  0.149  0.068  0.085
## MBI              0.022  0.426           0.032  0.035 -0.085  0.157 -0.060
## TQ               0.119 -0.082           0.300  0.419  0.298  0.070  0.269
##                   CART    SET    SNQ RSMSf1    MBI     TQ
## sex             -0.346 -0.166  0.033 -0.342  0.022  0.119
## age              0.023 -0.207 -0.010  0.033  0.426 -0.082
## education_level  0.404  0.199  0.193  0.109  0.032  0.300
## EKM              0.197  0.062  0.109  0.243  0.035  0.419
## ERT              0.142  0.171  0.102  0.149 -0.085  0.298
## TASf1            0.022  0.034  0.083  0.068  0.157  0.070
## FP               0.216  0.185  0.235  0.085 -0.060  0.269
## CART             1.000  0.168  0.105  0.185  0.187  0.179
## SET              0.168  1.000  0.069  0.060 -0.014  0.169
## SNQ              0.105  0.069  1.000 -0.172 -0.024 -0.078
## RSMSf1           0.185  0.060 -0.172  1.000  0.015  0.171
## MBI              0.187 -0.014 -0.024  0.015  1.000 -0.081
## TQ               0.179  0.169 -0.078  0.171 -0.081  1.000
```

```
corr_data_final_P <- round(as.data.frame(corr_data_final$P), 3); corr_data_final_P
```

```
##                   sex   age education_level   EKM   ERT TASf1    FP  CART   SET
## sex                NA 0.454           0.055 0.179 0.516 0.760 0.243 0.000 0.060
## age             0.454    NA           0.296 0.897 0.000 0.515 0.111 0.798 0.019
## education_level 0.055 0.296              NA 0.019 0.001 0.254 0.036 0.000 0.024
## EKM             0.179 0.897           0.019    NA 0.000 0.084 0.006 0.025 0.485
## ERT             0.516 0.000           0.001 0.000    NA 0.449 0.002 0.110 0.053
## TASf1           0.760 0.515           0.254 0.084 0.449    NA 0.465 0.805 0.702
## FP              0.243 0.111           0.036 0.006 0.002 0.465    NA 0.014 0.036
## CART            0.000 0.798           0.000 0.025 0.110 0.805 0.014    NA 0.058
## SET             0.060 0.019           0.024 0.485 0.053 0.702 0.036 0.058    NA
## SNQ             0.711 0.913           0.028 0.218 0.252 0.349 0.007 0.238 0.436
## RSMSf1          0.000 0.714           0.217 0.005 0.092 0.443 0.336 0.035 0.496
## MBI             0.802 0.000           0.722 0.696 0.339 0.075 0.501 0.034 0.871
## TQ              0.420 0.581           0.038 0.003 0.040 0.635 0.064 0.222 0.250
##                   SNQ RSMSf1   MBI    TQ
## sex             0.711  0.000 0.802 0.420
## age             0.913  0.714 0.000 0.581
## education_level 0.028  0.217 0.722 0.038
## EKM             0.218  0.005 0.696 0.003
## ERT             0.252  0.092 0.339 0.040
## TASf1           0.349  0.443 0.075 0.635
## FP              0.007  0.336 0.501 0.064
## CART            0.238  0.035 0.034 0.222
## SET             0.436  0.496 0.871 0.250
## SNQ                NA  0.051 0.790 0.599
```

```
## RSMSf1          0.051     NA 0.867 0.244
## MBI             0.790  0.867    NA 0.586
## TQ              0.599  0.244 0.586    NA
```

```
#knitr::kable(corr_data_final_r, format = "latex", booktabs = TRUE, caption = "Correlation matrix final
#knitr::kable(corr_data_final_P, format = "latex", booktabs = TRUE, caption = "P-values corresponding t
```

6.2 Z-scores

```
names_items_analysis <- c("EKM", "ERT", "TASf1", "CART", "FP", "SET", "SNQ", "MBI", "RSMSf1")
data_final_Z <- cbind(data_final[, c("ID", "centerID", names_covariates)],
                      round(scale(data_final[, c(names_items_analysis, 'TQ')]), 4))
```

7 Methods: Analyses

7.1 CFA models

```
#full 3 factor model
m1_f3fm <- '
# Defining the factors (latent variables)
f1_P =~ EKM + ERT + TASf1
f2_U =~ SET + FP + CART
f3_BR =~ RSMSf1 + SNQ + MBI

# Allow factors to be correlated
f1_P ~~ f2_U
f1_P ~~ f3_BR
f2_U ~~ f3_BR
'

#two factor model: f1 = f2
m2_2fm_1eq2 <- '
# Defining the factors (latent variables)
f1_P =~ EKM + ERT + TASf1
f2_U =~ SET + FP + CART
f3_BR =~ RSMSf1 + SNQ + MBI

# Allow factors to be correlated
f1_P ~~ 1*f2_U  # Fix correlation between f1_P and f2_U at 1
f1_P ~~ f3_BR
f2_U ~~ f3_BR
'

#model 3: two factor model: f1 = f3 (CART)
m3_2fm_1eq3 <- '
# Defining the factors (latent variables)
f1_P =~ EKM + ERT + TASf1
f2_U =~ SET + FP + CART
```

```
f3_BR =~ RSMSf1 + SNQ + MBI

# Allow factors to be correlated
f1_P ~~ f2_U
f1_P ~~ 1*f3_BR # Fix correlation between f1_P and f3_BR at 1
f2_U ~~ f3_BR
'

#model 4: two factor model: f2 = f3 (CART)
m4_2fm_2eq3 <- '
# Defining the factors (latent variables)
f1_P =~ EKM + ERT + TASf1
f2_U =~ SET + FP + CART
f3_BR =~ RSMSf1 + SNQ + MBI

# Allow factors to be correlated
f1_P ~~ f2_U
f1_P ~~ f3_BR
f2_U ~~ 1*f3_BR # Fix correlation between f2_U and f3_BR at 1
'

#model 5: one fact0r model: f1 = f2 = f3 (CART)
m5_1fm <- '
# Defining the factors (latent variables)
f1_P =~ EKM + ERT + TASf1
f2_U =~ SET + FP + CART
f3_BR =~ RSMSf1 + SNQ + MBI

# Fixing all interfactor correlations at 1
f1_P ~~ 1*f2_U
f1_P ~~ 1*f3_BR
f2_U ~~ 1*f3_BR
'

#model 6: 3 one facter model: independent factors (CART)
m6_3x1fm <- '
# Defining the factors (latent variables)
f1_P =~ EKM + ERT + TASf1
f2_U =~ SET + FP + CART
f3_BR =~ RSMSf1 + SNQ + MBI

# Fixing all interfactor correlations at 0
f1_P ~~ 0*f2_U
f1_P ~~ 0*f3_BR
f2_U ~~ 0*f3_BR
'

# #model 6.3alt: 3 factor model; BR independent
# m6_3fm_BRindep <- '
# # Defining the factors (latent variables)
# f1_P =~ EKM + ERT + TASf1
# f2_U =~ SET + FP + CART
# f3_BR =~ RSMSf1 + SNQ + MBI
```

```
# # Fixing 2 interfactor correlations at 0
# f1_P ~~ f2_U
# f1_P ~~ 0*f3_BR
# f2_U ~~ 0*f3_BR
# '
```

7.2 CFA EMC data (N=49)

```
#model 1; EMC data
fit_m1_EMC <- cfa(model = m1_f3fm, data = data_final_Z[data_final_Z$centerID == "EMC",])
```

```
## Warning: lavaan->lav_object_post_check():
##    some estimated lv variances are negative
```

```
summ_m1_EMC <- summary(fit_m1_EMC, standardized = TRUE, fit.measures = TRUE)
fm_m1_EMC <- fitMeasures(fit_m1_EMC, c("npar", "chisq", "df", "pvalue", "aic",
                                       "srmr", "cfi", "ifi", "nfi", "rmsea", "bic"))
```

```
#model 2; EMC data
fit_m2_EMC <- cfa(model = m2_2fm_1eq2, data = data_final_Z[data_final_Z$centerID == "EMC",])
```

```
## Warning: lavaan->lav_start_check_cov():
##    starting values imply a correlation larger than 1; variables involved are:
##    f1_P f2_U
```

```
## Warning: lavaan->lav_object_post_check():
##    some estimated lv variances are negative
```

```
summ_m2_EMC <- summary(fit_m2_EMC, standardized = TRUE, fit.measures = TRUE)
fm_m2_EMC <- fitMeasures(fit_m2_EMC, c("npar", "chisq", "df", "pvalue", "aic",
                                       "srmr", "cfi", "ifi", "nfi", "rmsea", "bic"))
```

```
#model 3; EMC data
fit_m3_EMC <- cfa(model = m3_2fm_1eq3, data = data_final_Z[data_final_Z$centerID == "EMC",])
```

```
## Warning: lavaan->lav_start_check_cov():
##    starting values imply a correlation larger than 1; variables involved are:
##    f1_P f3_BR
```

```
## Warning: lavaan->lav_object_post_check():
##    some estimated lv variances are negative
```

```
summ_m3_EMC <- summary(fit_m3_EMC, standardized = TRUE, fit.measures = TRUE)
fm_m3_EMC <- fitMeasures(fit_m3_EMC, c("npar", "chisq", "df", "pvalue", "aic",
                                       "srmr", "cfi", "ifi", "nfi", "rmsea", "bic"))
```

```r
#model 4; EMC data
fit_m4_EMC <- cfa(model = m4_2fm_2eq3, data = data_final_Z[data_final_Z$centerID == "EMC",])
```

```
## Warning: lavaan->lav_start_check_cov():
##    starting values imply a correlation larger than 1; variables involved are:
##    f2_U f3_BR

## Warning: lavaan->lav_object_post_check():
##    some estimated lv variances are negative
```

```r
summ_m4_EMC <- summary(fit_m4_EMC, standardized = TRUE, fit.measures = TRUE)
fm_m4_EMC <- fitMeasures(fit_m4_EMC, c("npar", "chisq", "df", "pvalue", "aic",
                                       "srmr", "cfi", "ifi", "nfi", "rmsea", "bic"))
```

```r
#model 5; EMC data
fit_m5_EMC <- cfa(model = m5_1fm, data = data_final_Z[data_final_Z$centerID == "EMC",])
```

```
## Warning: lavaan->lav_start_check_cov():
##    starting values imply a correlation larger than 1; variables involved are:
##    f1_P f2_U

## Warning: lavaan->lav_start_check_cov():
##    starting values imply a correlation larger than 1; variables involved are:
##    f1_P f3_BR

## Warning: lavaan->lav_object_post_check():
##    some estimated lv variances are negative
```

```r
summ_m5_EMC <- summary(fit_m5_EMC, standardized = TRUE, fit.measures = TRUE)
fm_m5_EMC <- fitMeasures(fit_m5_EMC, c("npar", "chisq", "df", "pvalue", "aic",
                                       "srmr", "cfi", "ifi", "nfi", "rmsea", "bic"))
```

```r
#model 6; EMC data
fit_m6_EMC <- cfa(model = m6_3x1fm, data = data_final_Z[data_final_Z$centerID == "EMC",])
```

```
## Warning: lavaan->lav_object_post_check():
##    some estimated ov variances are negative
```

```r
summ_m6_EMC <- summary(fit_m6_EMC, standardized = TRUE, fit.measures = TRUE)
fm_m6_EMC <- fitMeasures(fit_m6_EMC, c("npar", "chisq", "df", "pvalue", "aic",
                                       "srmr", "cfi", "ifi", "nfi", "rmsea", "bic"))
```

7.3 CFA pooled data (N=128)

```r
#model 1; pooled data
fit_m1_pld <- cfa(model = m1_f3fm, data = data_final_Z, std.lv = FALSE)
```

```
## Warning: lavaan->lav_object_post_check():
##    some estimated lv variances are negative
```

```r
summ_m1_pld <- summary(fit_m1_pld, standardized = TRUE, fit.measures = TRUE)
fm_m1_pld <- fitMeasures(fit_m1_pld, c("npar", "chisq", "df", "pvalue", "aic",
                                       "srmr", "cfi", "ifi", "nfi", "rmsea", "bic"))
```

```r
#model 2; pooled data
fit_m2_pld <- cfa(model = m2_2fm_1eq2, data = data_final_Z)
```

```
## Warning: lavaan->lav_start_check_cov():
##    starting values imply a correlation larger than 1; variables involved are:
##    f1_P f2_U

## Warning: lavaan->lav_object_post_check():
##    some estimated lv variances are negative
```

```r
summ_m2_pld <- summary(fit_m2_pld, standardized = TRUE, fit.measures = TRUE)
fm_m2_pld <- fitMeasures(fit_m2_pld, c("npar", "chisq", "df", "pvalue", "aic",
                                       "srmr", "cfi", "ifi", "nfi", "rmsea", "bic"))
```

```r
#model 3; pooled data
fit_m3_pld <- cfa(model = m3_2fm_1eq3, data = data_final_Z)
```

```
## Warning: lavaan->lav_start_check_cov():
##    starting values imply a correlation larger than 1; variables involved are:
##    f1_P f3_BR

## Warning: lavaan->lav_object_post_check():
##    some estimated lv variances are negative
```

```r
summ_m3_pld <- summary(fit_m3_pld, standardized = TRUE, fit.measures = TRUE)
fm_m3_pld <- fitMeasures(fit_m3_pld, c("npar", "chisq", "df", "pvalue", "aic",
                                       "srmr", "cfi", "ifi", "nfi", "rmsea", "bic"))
```

```r
#model 4; pooled data
fit_m4_pld <- cfa(model = m4_2fm_2eq3, data = data_final_Z)
```

```
## Warning: lavaan->lav_start_check_cov():
##    starting values imply a correlation larger than 1; variables involved are:
##    f2_U f3_BR

## Warning: lavaan->lav_object_post_check():
##    some estimated lv variances are negative
```

```r
summ_m4_pld <- summary(fit_m4_pld, standardized = TRUE, fit.measures = TRUE)
fm_m4_pld <- fitMeasures(fit_m4_pld, c("npar", "chisq", "df", "pvalue", "aic",
                                       "srmr", "cfi", "ifi", "nfi", "rmsea", "bic"))
```

```r
#model 5; pooled data
fit_m5_pld <- cfa(model = m5_1fm, data = data_final_Z)
```

```
## Warning: lavaan->lav_start_check_cov():
##    starting values imply a correlation larger than 1; variables involved are:
##    f1_P f2_U

## Warning: lavaan->lav_start_check_cov():
##    starting values imply a correlation larger than 1; variables involved are:
##    f1_P f3_BR

## Warning: lavaan->lav_object_post_check():
##    some estimated lv variances are negative
```

```
summ_m5_pld <- summary(fit_m5_pld, standardized = TRUE, fit.measures = TRUE)
fm_m5_pld <- fitMeasures(fit_m5_pld, c("npar", "chisq", "df", "pvalue", "aic",
                                       "srmr", "cfi", "ifi", "nfi", "rmsea", "bic"))
```

```
#model 6; pooled data
fit_m6_pld <- cfa(model = m6_3x1fm, data = data_final_Z)
summ_m6_pld <- summary(fit_m6_pld, standardized = TRUE, fit.measures = TRUE)
fm_m6_pld <- fitMeasures(fit_m6_pld, c("npar", "chisq", "df", "pvalue", "aic",
                                       "srmr", "cfi", "ifi", "nfi", "rmsea", "bic"))
```

7.4 SEM models

```
#3 paths model
#full 3 factor model
sem1 <- '
# Defining the factors (latent variables)
f1_P =~ EKM + ERT + TASf1
f2_U =~ SET + FP + CART
f3_BR =~ RSMSf1 + SNQ + MBI

#regression
TQ ~ f1_P + f2_U + f3_BR

# Allow factors to be correlated
f1_P ~~ f2_U
f1_P ~~ f3_BR
f2_U ~~ f3_BR
'

#2 path models
#path P->TQ=0
sem2 <- '
# Defining the factors (latent variables)
f1_P =~ EKM + ERT + TASf1
f2_U =~ SET + FP + CART
f3_BR =~ RSMSf1 + SNQ + MBI

#regression
TQ ~ 0*f1_P + f2_U + f3_BR
```

```
# Allow factors to be correlated
f1_P ~~ f2_U
f1_P ~~ f3_BR
f2_U ~~ f3_BR
'

#path U->TQ=0
sem3 <- '
# Defining the factors (latent variables)
f1_P =~ EKM + ERT + TASf1
f2_U =~ SET + FP + CART
f3_BR =~ RSMSf1 + SNQ + MBI

#regression
TQ ~ f1_P + 0*f2_U + f3_BR

# Allow factors to be correlated
f1_P ~~ f2_U
f1_P ~~ f3_BR
f2_U ~~ f3_BR
'

#path BR->TQ=0
sem4 <- '
# Defining the factors (latent variables)
f1_P =~ EKM + ERT + TASf1
f2_U =~ SET + FP + CART
f3_BR =~ RSMSf1 + SNQ + MBI

#regression
TQ ~ f1_P + f2_U + 0*f3_BR

# Allow factors to be correlated
f1_P ~~ f2_U
f1_P ~~ f3_BR
f2_U ~~ f3_BR
'


#1 path models
#only path P
sem5 <- '
# Defining the factors (latent variables)
f1_P =~ EKM + ERT + TASf1
f2_U =~ SET + FP + CART
f3_BR =~ RSMSf1 + SNQ + MBI

#regression
TQ ~ f1_P + 0*f2_U + 0*f3_BR

# Allow factors to be correlated
f1_P ~~ f2_U
f1_P ~~ f3_BR
```

```
f2_U ~~ f3_BR
'

#only path U
sem6 <- '
# Defining the factors (latent variables)
f1_P =~ EKM + ERT + TASf1
f2_U =~ SET + FP + CART
f3_BR =~ RSMSf1 + SNQ + MBI

#regression
TQ ~ 0*f1_P + f2_U + 0*f3_BR

# Allow factors to be correlated
f1_P ~~ f2_U
f1_P ~~ f3_BR
f2_U ~~ f3_BR
'

#only path BR
sem7 <- '
# Defining the factors (latent variables)
f1_P =~ EKM + ERT + TASf1
f2_U =~ SET + FP + CART
f3_BR =~ RSMSf1 + SNQ + MBI

#regression
TQ ~ 0*f1_P + 0*f2_U + f3_BR

# Allow factors to be correlated
f1_P ~~ f2_U
f1_P ~~ f3_BR
f2_U ~~ f3_BR
'


#no path model (baseline)
sem8 <- '
# Defining the factors (latent variables)
f1_P =~ EKM + ERT + TASf1
f2_U =~ SET + FP + CART
f3_BR =~ RSMSf1 + SNQ + MBI

#regression
TQ ~ 0*f1_P + 0*f2_U + 0*f3_BR

# Allow factors to be correlated
f1_P ~~ f2_U
f1_P ~~ f3_BR
f2_U ~~ f3_BR
'
```

7.5 SEM EMC data (N=47)

```
#3 paths model
fit_sem1 <- sem(model = sem1, data = data_final_Z[data_final_Z$centerID == 'EMC',])
```

```
## Warning: lavaan->lav_object_post_check():
##    some estimated lv variances are negative
```

```
summ_sem1 <- summary(fit_sem1, standardized = TRUE, fit.measures = TRUE)
fm_sem1 <- fitMeasures(fit_sem1, c("npar", "chisq", "df", "pvalue", "aic",
                                   "srmr", "cfi", "ifi", "nfi", "rmsea", "bic"))
stcoef_sem1 <- standardizedSolution(fit_sem1)[10:12,1:7]

#2 path models
fit_sem2 <- sem(model = sem2, data = data_final_Z[data_final_Z$centerID == 'EMC',])
```

```
## Warning: lavaan->lav_object_post_check():
##    some estimated lv variances are negative
```

```
summ_sem2 <- summary(fit_sem2, standardized = TRUE, fit.measures = TRUE)
fm_sem2 <- fitMeasures(fit_sem2, c("npar", "chisq", "df", "pvalue", "aic",
                                   "srmr", "cfi", "ifi", "nfi", "rmsea", "bic"))
stcoef_sem2 <- standardizedSolution(fit_sem2)[10:12,1:7]


fit_sem3 <- sem(model = sem3, data = data_final_Z[data_final_Z$centerID == 'EMC',])
```

```
## Warning: lavaan->lav_object_post_check():
##    some estimated lv variances are negative
```

```
summ_sem3 <- summary(fit_sem3, standardized = TRUE, fit.measures = TRUE)
fm_sem3 <- fitMeasures(fit_sem3, c("npar", "chisq", "df", "pvalue", "aic",
                                   "srmr", "cfi", "ifi", "nfi", "rmsea", "bic"))
stcoef_sem3 <- standardizedSolution(fit_sem3)[10:12,1:7]


fit_sem4 <- sem(model = sem4, data = data_final_Z[data_final_Z$centerID == 'EMC',])
```

```
## Warning: lavaan->lav_object_post_check():
##    some estimated lv variances are negative
```

```
summ_sem4 <- summary(fit_sem4, standardized = TRUE, fit.measures = TRUE)
fm_sem4 <- fitMeasures(fit_sem4, c("npar", "chisq", "df", "pvalue", "aic",
                                   "srmr", "cfi", "ifi", "nfi", "rmsea", "bic"))
stcoef_sem4 <- standardizedSolution(fit_sem4)[10:12,1:7]


#1 path models
fit_sem5 <- sem(model = sem5, data = data_final_Z[data_final_Z$centerID == 'EMC',])
```

```
## Warning: lavaan->lav_object_post_check():
##    some estimated lv variances are negative
```

```r
summ_sem5 <- summary(fit_sem5, standardized = TRUE, fit.measures = TRUE)
fm_sem5 <- fitMeasures(fit_sem5, c("npar", "chisq", "df", "pvalue", "aic",
                                   "srmr", "cfi", "ifi", "nfi", "rmsea", "bic"))
stcoef_sem5 <- standardizedSolution(fit_sem5)[10:12,1:7]


fit_sem6 <- sem(model = sem6, data = data_final_Z[data_final_Z$centerID == 'EMC',])
```

```
## Warning: lavaan->lav_object_post_check():
##    some estimated lv variances are negative
```

```r
summ_sem6 <- summary(fit_sem6, standardized = TRUE, fit.measures = TRUE)
fm_sem6 <- fitMeasures(fit_sem6, c("npar", "chisq", "df", "pvalue", "aic",
                                   "srmr", "cfi", "ifi", "nfi", "rmsea", "bic"))
stcoef_sem6 <- standardizedSolution(fit_sem6)[10:12,1:7]


fit_sem7 <- sem(model = sem7, data = data_final_Z[data_final_Z$centerID == 'EMC',])
```

```
## Warning: lavaan->lav_object_post_check():
##    some estimated lv variances are negative
```

```r
summ_sem7 <- summary(fit_sem7, standardized = TRUE, fit.measures = TRUE)
fm_sem7 <- fitMeasures(fit_sem7, c("npar", "chisq", "df", "pvalue", "aic",
                                   "srmr", "cfi", "ifi", "nfi", "rmsea", "bic"))
stcoef_sem7 <- standardizedSolution(fit_sem7)[10:12,1:7]


#no path model
fit_sem8 <- sem(model = sem8, data = data_final_Z[data_final_Z$centerID == 'EMC',])
```

```
## Warning: lavaan->lav_object_post_check():
##    some estimated lv variances are negative
```

```r
summ_sem8 <- summary(fit_sem8, standardized = TRUE, fit.measures = TRUE)
fm_sem8 <- fitMeasures(fit_sem8, c("npar", "chisq", "df", "pvalue", "aic",
                                   "srmr", "cfi", "ifi", "nfi", "rmsea", "bic"))
stcoef_sem8 <- standardizedSolution(fit_sem8)[10:12,1:7]
```

8 Results

8.1 CFA EMC data

8.1.1 Fit indices

```r
results_CFA_EMC <- data.frame(round(rbind(fm_m1_EMC, fm_m2_EMC, fm_m3_EMC,
                                          fm_m4_EMC, fm_m5_EMC, fm_m6_EMC
                                          ), 3))
modelNames_CFA <- c("Full three-factor",
                    "Two-factor: P = U",
                    "Two-factor: P = BR",
                    "Two-factor: U = BR",
                    "One-factor: P = U = BR",
                    "Independent three factor")
rownames(results_CFA_EMC) <- NULL
results_CFA_EMC <- cbind(Model = modelNames_CFA, results_CFA_EMC)
results_CFA_EMC
```

```
##                      Model npar  chisq df pvalue      aic  srmr   cfi   ifi
## 1        Full three-factor   21 26.107 24  0.348 1199.566 0.080 0.961 0.968
## 2        Two-factor: P = U   20 40.181 25  0.028 1211.639 0.325 0.718 0.766
## 3       Two-factor: P = BR   20 36.784 25  0.061 1208.242 0.266 0.781 0.818
## 4       Two-factor: U = BR   20 47.378 25  0.004 1218.836 0.409 0.584 0.655
## 5   One-factor: P = U = BR   18 47.650 27  0.008 1215.108 0.442 0.616 0.671
## 6 Independent three factor   18 46.143 27  0.012 1213.601 0.169 0.644 0.695
##     nfi rmsea      bic
## 1 0.709 0.043 1238.861
## 2 0.553 0.112 1249.063
## 3 0.590 0.099 1245.666
## 4 0.472 0.137 1256.260
## 5 0.469 0.126 1248.790
## 6 0.486 0.122 1247.283
```

8.1.2 Model comparisson

```r
anova(fit_m1_EMC, fit_m2_EMC)
```

```
##
## Chi-Squared Difference Test
##
##            Df    AIC    BIC  Chisq Chisq diff   RMSEA Df diff Pr(>Chisq)
## fit_m1_EMC 24 1199.6 1238.9 26.108
## fit_m2_EMC 25 1211.6 1249.1 40.181     14.074 0.52189       1  0.0001758 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(fit_m1_EMC, fit_m3_EMC)
```

```
##
## Chi-Squared Difference Test
##
##            Df    AIC    BIC  Chisq Chisq diff   RMSEA Df diff Pr(>Chisq)
## fit_m1_EMC 24 1199.6 1238.9 26.108
## fit_m3_EMC 25 1208.2 1245.7 36.784     10.676 0.44899       1   0.001085 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(fit_m1_EMC, fit_m4_EMC)
```

```
##
## Chi-Squared Difference Test
##
##            Df    AIC    BIC  Chisq Chisq diff   RMSEA Df diff Pr(>Chisq)
## fit_m1_EMC 24 1199.6 1238.9 26.108
## fit_m4_EMC 25 1218.8 1256.3 47.378      21.27 0.64984       1  3.989e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(fit_m1_EMC, fit_m5_EMC)
```

```
##
## Chi-Squared Difference Test
##
##            Df    AIC    BIC  Chisq Chisq diff   RMSEA Df diff Pr(>Chisq)
## fit_m1_EMC 24 1199.6 1238.9 26.108
## fit_m5_EMC 27 1215.1 1248.8 47.650     21.543 0.35885       3   8.12e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#anova(fit_m1_EMC, fit_m6_EMC)
```

8.1.3 Favoured model

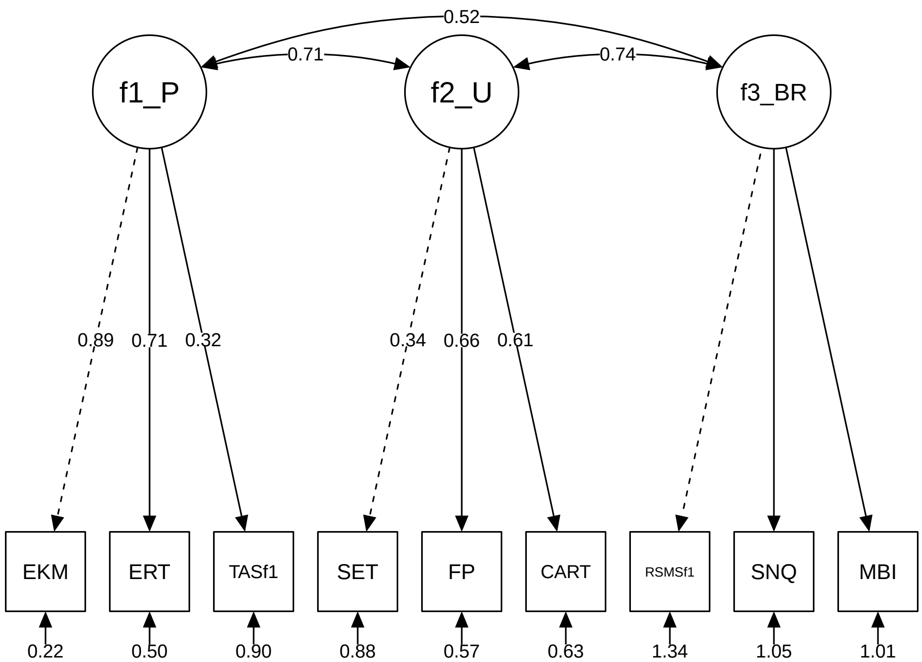
```r
stcoefs_fav_CFA_EMC <- standardizedSolution(fit_m1_EMC)
nice_table(stcoefs_fav_CFA_EMC)
```

```
## Warning: fonts used in `flextable` are ignored because the `pdflatex` engine is
## used and not `xelatex` or `lualatex`. You can avoid this warning by using the
## `set_flextable_defaults(fonts_ignore=TRUE)` command or use a compatible engine
## by defining `latex_engine: xelatex` in the YAML header of the R Markdown
## document.
```

lhs	op	rhs	est.std	se	z	p	ci.lower	ci.upper
f1_P	=~	EKM	0.89	0.10	8.73	< .001***	0.69	1.08
f1_P	=~	ERT	0.71	0.11	6.68	< .001***	0.50	0.92
f1_P	=~	TASf1	0.32	0.14	2.21	.027*	0.04	0.60
f2_U	=~	SET	0.34	0.16	2.21	.027*	0.04	0.65
f2_U	=~	FP	0.66	0.13	5.08	< .001***	0.40	0.91
f2_U	=~	CART	0.61	0.13	4.64	< .001***	0.35	0.86
f3_BR	=~	RSMSf1						
f3_BR	=~	SNQ						

lhs	op	rhs	est.std	se	z	p	ci.lower	ci.upper
f3_BR	=~	MBI						
f1_P	~~	f2_U	0.71	0.15	4.63	$<$ .001***	0.41	1.02
f1_P	~~	f3_BR	0.52	0.44	1.19	.235	-0.34	1.39
f2_U	~~	f3_BR	0.74	0.59	1.25	.210	-0.42	1.89
EKM	~~	EKM	0.22	0.18	1.21	.227	-0.14	0.57
ERT	~~	ERT	0.50	0.15	3.30	.001***	0.20	0.79
TASf1	~~	TASf1	0.90	0.09	9.77	$<$ .001***	0.72	1.08
SET	~~	SET	0.88	0.11	8.24	$<$ .001***	0.67	1.09
FP	~~	FP	0.57	0.17	3.38	.001***	0.24	0.90
CART	~~	CART	0.63	0.16	3.98	$<$ .001***	0.32	0.94
RSMSf1	~~	RSMSf1	1.34	0.53	2.52	.012*	0.30	2.38
SNQ	~~	SNQ	1.05	0.07	16.03	$<$ .001***	0.92	1.18
MBI	~~	MBI	1.01	0.03	34.63	$<$ .001***	0.96	1.07
f1_P	~~	f1_P	1.00	0.00			1.00	1.00
f2_U	~~	f2_U	1.00	0.00			1.00	1.00
f3_BR	~~	f3_BR						

```r
#print(nice_table(stcoefs_fav_CFA_EMC), preview = "docx") #Word format output
semPaths(fit_m1_EMC,
         whatLabels = "std",  # Display standardized estimates
         layout = "tree2",      # Choose a layout, "tree" is one option
         style = "lisrel",     # Style for the diagram
         nCharNodes = 0,       # Full variable names
         sizeMan = 7,          # Size of manifest variables
         sizeLat = 10,         # Size of latent variables
         edge.label.cex = 0.75, # Adjust the size of the edge labels
         label.cex = 0.8,      # Adjust the size of the node labels
         residuals = TRUE,     # Show residuals
         intercepts = FALSE,   # Hide intercepts
         thresholds = FALSE,
         fade = FALSE,
         edge.color = "black"
         )
```



8.2 CFA pooled data

8.2.1 Fit indices

```
results_CFA_pld <- data.frame(round(rbind(fm_m1_pld, fm_m2_pld, fm_m3_pld,
                                          fm_m4_pld, fm_m5_pld, fm_m6_pld), 3))
modelNames_CFA <- c("Full three-factor",
                    "Two-factor: P = U",
                    "Two-factor: P = BR",
                    "Two-factor: U = BR",
                    "One-factor: P = U = BR",
                    "Independent three factor")
rownames(results_CFA_pld) <- NULL
results_CFA_pld <- cbind(Model = modelNames_CFA, results_CFA_pld)
results_CFA_pld
```

```
##                      Model npar  chisq df pvalue      aic  srmr   cfi   ifi
## 1        Full three-factor   21 23.099 24  0.514 3203.035 0.052 1.000 1.007
## 2        Two-factor: P = U   20 62.850 25  0.000 3240.786 0.305 0.661 0.692
## 3       Two-factor: P = BR   20 56.667 25  0.000 3234.604 0.249 0.717 0.742
## 4       Two-factor: U = BR   20 71.317 25  0.000 3249.254 0.303 0.586 0.623
## 5   One-factor: P = U = BR   18 77.097 27  0.000 3251.034 0.395 0.552 0.585
## 6 Independent three factor   18 63.122 27  0.000 3237.059 0.126 0.677 0.701
```

```
##     nfi rmsea      bic
## 1 0.844 0.000 3263.091
## 2 0.575 0.108 3297.983
## 3 0.617 0.099 3291.800
## 4 0.517 0.120 3306.450
## 5 0.478 0.120 3302.510
## 6 0.573 0.102 3288.536
```

8.2.2 Model comparisson

```r
anova(fit_m1_pld, fit_m2_pld)
```

```
##
## Chi-Squared Difference Test
##
##            Df    AIC    BIC  Chisq Chisq diff   RMSEA Df diff Pr(>Chisq)
## fit_m1_pld 24 3203.0 3263.1 23.099
## fit_m2_pld 25 3240.8 3298.0 62.850     39.751 0.54809       1  2.885e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(fit_m1_pld, fit_m3_pld)
```

```
##
## Chi-Squared Difference Test
##
##            Df    AIC    BIC  Chisq Chisq diff   RMSEA Df diff Pr(>Chisq)
## fit_m1_pld 24 3203.0 3263.1 23.099
## fit_m3_pld 25 3234.6 3291.8 56.667     33.569 0.50246       1   6.88e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(fit_m1_pld, fit_m4_pld)
```

```
##
## Chi-Squared Difference Test
##
##            Df    AIC    BIC  Chisq Chisq diff   RMSEA Df diff Pr(>Chisq)
## fit_m1_pld 24 3203.0 3263.1 23.099
## fit_m4_pld 25 3249.3 3306.5 71.317     48.219 0.60501       1  3.812e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(fit_m1_pld, fit_m5_pld)
```

```
##
## Chi-Squared Difference Test
##
##            Df  AIC    BIC  Chisq Chisq diff   RMSEA Df diff Pr(>Chisq)
```

```
## fit_m1_pld 24 3203 3263.1 23.099
## fit_m5_pld 27 3251 3302.5 77.097     53.998 0.36301       3  1.123e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(fit_m1_pld, fit_m6_pld)
```

```
##
## Chi-Squared Difference Test
##
##            Df    AIC    BIC  Chisq Chisq diff  RMSEA Df diff Pr(>Chisq)
## fit_m1_pld 24 3203.0 3263.1 23.099
## fit_m6_pld 27 3237.1 3288.5 63.122     40.024 0.3093       3  1.053e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# anova(fit_m2_pld, fit_m5_pld)
# anova(fit_m3_pld, fit_m5_pld)
# anova(fit_m4_pld, fit_m5_pld)
```

8.2.3 Favoured model

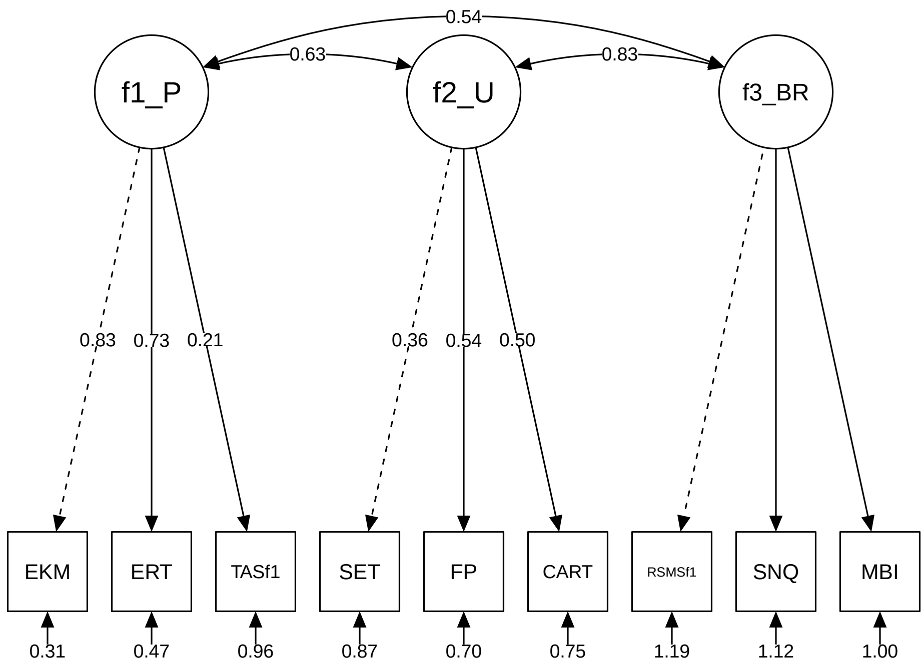
```r
stcoefs_fav_CFA_pld <- standardizedSolution(fit_m1_pld)
stcoefs_fav_CFA_pld_uns <- parameterEstimates(fit_m1_pld, standardized = TRUE)
nice_table(stcoefs_fav_CFA_pld)
```

```
## Warning: fonts used in `flextable` are ignored because the `pdflatex` engine is
## used and not `xelatex` or `lualatex`. You can avoid this warning by using the
## `set_flextable_defaults(fonts_ignore=TRUE)` command or use a compatible engine
## by defining `latex_engine: xelatex` in the YAML header of the R Markdown
## document.
```

lhs	op	rhs	est.std	se	z	p	ci.lower	ci.upper
f1_P	=~	EKM	0.83	0.08	10.20	$<$ .001***	0.67	0.99
f1_P	=~	ERT	0.73	0.08	9.16	$<$ .001***	0.57	0.88
f1_P	=~	TASf1	0.21	0.10	2.17	.030*	0.02	0.40
f2_U	=~	SET	0.36	0.10	3.52	$<$ .001***	0.16	0.57
f2_U	=~	FP	0.54	0.10	5.45	$<$ .001***	0.35	0.74
f2_U	=~	CART	0.50	0.10	5.05	$<$ .001***	0.31	0.70
f3_BR	=~	RSMSf1						
f3_BR	=~	SNQ						
f3_BR	=~	MBI						
f1_P	~~	f2_U	0.63	0.12	5.00	$<$ .001***	0.38	0.87

lhs	op	rhs	est.std	se	z	p	ci.lower	ci.upper
f1_P	~~	f3_BR	0.54	0.25	2.16	.031*	0.05	1.02
f2_U	~~	f3_BR	0.83	0.35	2.35	.019*	0.14	1.52
EKM	~~	EKM	0.31	0.14	2.24	.025*	0.04	0.57
ERT	~~	ERT	0.47	0.12	4.11	$<$ .001***	0.25	0.70
TASf1	~~	TASf1	0.96	0.04	23.99	$<$ .001***	0.88	1.03
SET	~~	SET	0.87	0.08	11.54	$<$ .001***	0.72	1.02
FP	~~	FP	0.70	0.11	6.51	$<$ .001***	0.49	0.92
CART	~~	CART	0.75	0.10	7.51	$<$ .001***	0.55	0.94
RSMSf1	~~	RSMSf1	1.19	0.15	8.18	$<$ .001***	0.90	1.47
SNQ	~~	SNQ	1.12	0.09	12.30	$<$ .001***	0.95	1.30
MBI	~~	MBI	1.00	0.01	117.24	$<$ .001***	0.99	1.02
f1_P	~~	f1_P	1.00	0.00			1.00	1.00
f2_U	~~	f2_U	1.00	0.00			1.00	1.00
f3_BR	~~	f3_BR						

```
#print(nice_table(stcoefs_fav_CFA_pld), preview = "docx") #Word format output
semPaths(fit_m1_pld,
         whatLabels = "std",  # Display standardized estimates
         layout = "tree2",     # Choose a layout, "tree" is one option
         style = "lisrel",     # Style for the diagram
         nCharNodes = 0,       # Full variable names
         sizeMan = 7,          # Size of manifest variables
         sizeLat = 10,         # Size of latent variables
         edge.label.cex = 0.75, # Adjust the size of the edge labels
         label.cex = 0.8,      # Adjust the size of the node labels
         residuals = TRUE,     # Show residuals
         intercepts = FALSE,   # Hide intercepts
         thresholds = FALSE,
         fade = FALSE,
         edge.color = "black"
         )
```



8.3 SEM EMC data

8.3.1 Fit indices

```
results_SEM <- data.frame(round(rbind(fm_sem1, fm_sem2, fm_sem3, fm_sem4,
                                      fm_sem5, fm_sem6, fm_sem7, fm_sem8), 3))
modelNames_SEM <- c("3p", "2p: U & BR", "2p: P & BR", "2p: P & U",
                    "1p: P", "1p: U","1p: BR", "0p")
rownames(results_SEM) <- NULL
results_SEM <- cbind(Model = modelNames_SEM, results_SEM)
#results_SEM
results_SEM[,c("Model", "npar", "df", "chisq", "pvalue", "aic", "srmr", "cfi", "ifi")]
```

```
##        Model npar df  chisq pvalue      aic  srmr   cfi   ifi
## 1         3p   25 30 29.976  0.467 1332.234 0.081 1.000 1.000
## 2 2p: U & BR   24 31 31.796  0.427 1332.053 0.080 0.987 0.989
## 3 2p: P & BR   24 31 29.977  0.519 1330.234 0.081 1.000 1.014
## 4  2p: P & U   24 31 30.384  0.498 1330.641 0.081 1.000 1.008
## 5      1p: P   23 32 30.390  0.548 1328.647 0.081 1.000 1.022
## 6      1p: U   23 32 32.932  0.421 1331.189 0.082 0.984 0.987
## 7     1p: BR   23 32 32.233  0.455 1330.490 0.085 0.996 0.997
## 8         0p   22 33 40.516  0.173 1336.773 0.119 0.873 0.894
```

```r
#print(nice_table(results_SEM[,c("Model", "npar", "df", "chisq", "pvalue", "aic", "srmr", "cfi", "ifi").
```

8.3.2 Model comparisson

```r
anova(fit_sem1, fit_sem2) #not significant
```

```
##
## Chi-Squared Difference Test
##
##           Df    AIC  BIC  Chisq Chisq diff  RMSEA Df diff Pr(>Chisq)
## fit_sem1 30 1332.2 1379 29.976
## fit_sem2 31 1332.0 1377 31.796     1.8199 0.1307       1     0.1773
```

```r
anova(fit_sem1, fit_sem3) #not significant
```

```
##
## Chi-Squared Difference Test
##
##           Df    AIC    BIC  Chisq Chisq diff RMSEA Df diff Pr(>Chisq)
## fit_sem1 30 1332.2 1379.0 29.976
## fit_sem3 31 1330.2 1375.1 29.977 0.00021022     0       1     0.9884
```

```r
anova(fit_sem1, fit_sem4) #not significant
```

```
##
## Chi-Squared Difference Test
##
##           Df    AIC    BIC  Chisq Chisq diff RMSEA Df diff Pr(>Chisq)
## fit_sem1 30 1332.2 1379.0 29.976
## fit_sem4 31 1330.6 1375.5 30.384    0.40759     0       1     0.5232
```

```r
#So: models 2-4 are not fitting sign than model 1

anova(fit_sem1, fit_sem5) #not significant
```

```
##
## Chi-Squared Difference Test
##
##           Df    AIC    BIC  Chisq Chisq diff RMSEA Df diff Pr(>Chisq)
## fit_sem1 30 1332.2 1379.0 29.976
## fit_sem5 32 1328.7 1371.7 30.390    0.41326     0       2     0.8133
```

```r
anova(fit_sem1, fit_sem6) #not significant
```

```
##
## Chi-Squared Difference Test
##
##           Df    AIC    BIC  Chisq Chisq diff    RMSEA Df diff Pr(>Chisq)
## fit_sem1 30 1332.2 1379.0 29.976
## fit_sem6 32 1331.2 1374.2 32.932     2.9556 0.099768       2     0.2281
```

```r
anova(fit_sem1, fit_sem7) #not significant
```

```
##
## Chi-Squared Difference Test
##
##           Df    AIC    BIC  Chisq Chisq diff    RMSEA Df diff Pr(>Chisq)
## fit_sem1 30 1332.2 1379.0 29.976
## fit_sem7 32 1330.5 1373.5 32.233     2.2566 0.051699       2     0.3236
```

```r
#So: models 5-7 are not fitting sign than model 1

anova(fit_sem2, fit_sem6) #not significant
```

```
##
## Chi-Squared Difference Test
##
##           Df    AIC    BIC  Chisq Chisq diff    RMSEA Df diff Pr(>Chisq)
## fit_sem2 31 1332.0 1377.0 31.796
## fit_sem6 32 1331.2 1374.2 32.932     1.1357 0.053162       1     0.2866
```

```r
anova(fit_sem2, fit_sem7) #not significant
```

```
##
## Chi-Squared Difference Test
##
##           Df    AIC    BIC  Chisq Chisq diff RMSEA Df diff Pr(>Chisq)
## fit_sem2 31 1332.0 1377.0 31.796
## fit_sem7 32 1330.5 1373.5 32.233    0.43669     0       1     0.5087
```

```r
#So: models 6-7 are not fitting sign than model 2

anova(fit_sem3, fit_sem5) #not significant
```

```
##
## Chi-Squared Difference Test
##
##           Df    AIC    BIC  Chisq Chisq diff RMSEA Df diff Pr(>Chisq)
## fit_sem3 31 1330.2 1375.1 29.977
## fit_sem5 32 1328.7 1371.7 30.390    0.41305     0       1     0.5204
```

```r
anova(fit_sem3, fit_sem7) #not significant
```

```
##
## Chi-Squared Difference Test
##
##           Df    AIC    BIC  Chisq Chisq diff   RMSEA Df diff Pr(>Chisq)
## fit_sem3 31 1330.2 1375.1 29.977
## fit_sem7 32 1330.5 1373.5 32.233     2.2564 0.16179       1     0.1331
```

```r
#So: models 5 & 7 are not fitting sign than model 3

anova(fit_sem4, fit_sem5) #not significant
```

```
##
## Chi-Squared Difference Test
##
##          Df    AIC    BIC  Chisq Chisq diff RMSEA Df diff Pr(>Chisq)
## fit_sem4 31 1330.6 1375.5 30.384
## fit_sem5 32 1328.7 1371.7 30.390  0.0056785     0       1     0.9399
```

```r
anova(fit_sem4, fit_sem6) #not significant
```

```
##
## Chi-Squared Difference Test
##
##          Df    AIC    BIC  Chisq Chisq diff   RMSEA Df diff Pr(>Chisq)
## fit_sem4 31 1330.6 1375.5 30.384
## fit_sem6 32 1331.2 1374.2 32.932      2.548 0.17958       1     0.1104
```

```r
#So: models 5-6 are not fitting sign than model 4

anova(fit_sem5, fit_sem8) #significant
```

```
##
## Chi-Squared Difference Test
##
##          Df    AIC    BIC  Chisq Chisq diff   RMSEA Df diff Pr(>Chisq)
## fit_sem5 32 1328.7 1371.7 30.390
## fit_sem8 33 1336.8 1377.9 40.516     10.126 0.43603       1   0.001462 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(fit_sem6, fit_sem8) #significant
```

```
##
## Chi-Squared Difference Test
##
##          Df    AIC    BIC  Chisq Chisq diff   RMSEA Df diff Pr(>Chisq)
## fit_sem6 32 1331.2 1374.2 32.932
## fit_sem8 33 1336.8 1377.9 40.516     7.5838 0.37035       1    0.00589 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(fit_sem7, fit_sem8) #significant
```

```
##
## Chi-Squared Difference Test
##
##          Df    AIC    BIC  Chisq Chisq diff   RMSEA Df diff Pr(>Chisq)
```

```
## fit_sem7 32 1330.5 1373.5 32.233
## fit_sem8 33 1336.8 1377.9 40.516     8.2827 0.38952       1   0.004002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#So, model 8 fits significantly worse than models 5-7

#Out of models 5-7, model 5 is in favour based on the fit indices
```

8.3.3 Favoured model

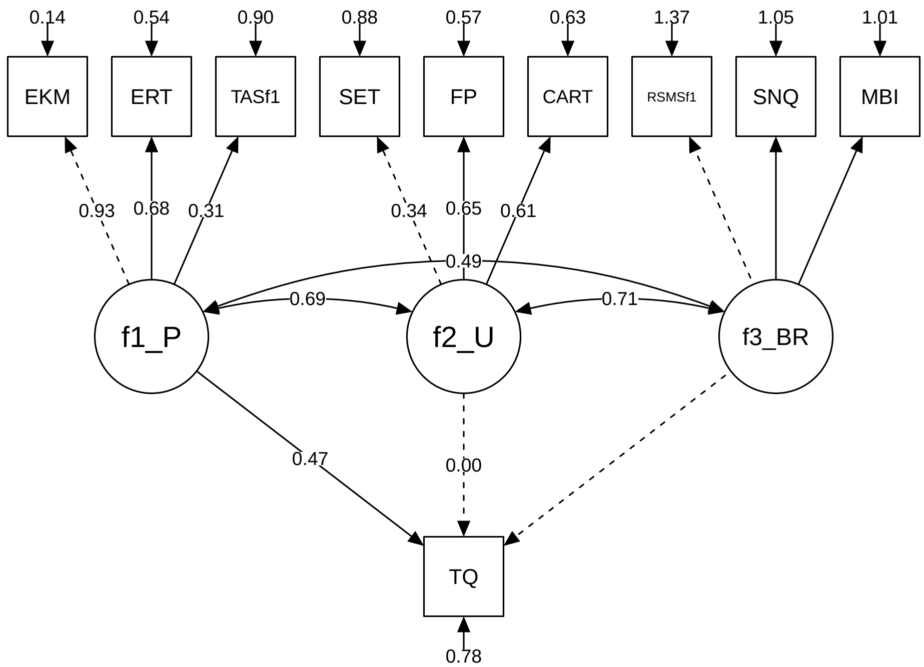
```
stcoefs_fav_SEM <- standardizedSolution(fit_sem5)
#parameterestimates(fit_sem5, standardized = TRUE)[, -10] #Unstandardized and standardized
nice_table(stcoefs_fav_SEM)
```

```
## Warning: fonts used in 'flextable' are ignored because the 'pdflatex' engine is
## used and not 'xelatex' or 'lualatex'. You can avoid this warning by using the
## 'set_flextable_defaults(fonts_ignore=TRUE)' command or use a compatible engine
## by defining 'latex_engine: xelatex' in the YAML header of the R Markdown
## document.
```

lhs	op	rhs	est.std	se	*z*	*p*	ci.lower	ci.upper
f1_P	=~	EKM	0.93	0.09	10.87	< .001***	0.76	1.09
f1_P	=~	ERT	0.68	0.10	6.76	< .001***	0.48	0.88
f1_P	=~	TASf1	0.31	0.14	2.24	.025*	0.04	0.59
f2_U	=~	SET	0.34	0.16	2.22	.027*	0.04	0.65
f2_U	=~	FP	0.65	0.13	5.05	< .001***	0.40	0.91
f2_U	=~	CART	0.61	0.13	4.66	< .001***	0.35	0.87
f3_BR	=~	RSMSf1						
f3_BR	=~	SNQ						
f3_BR	=~	MBI						
TQ	~	f1_P	0.47	0.13	3.74	< .001***	0.22	0.71
TQ	~	f2_U	0.00	0.00			0.00	0.00
TQ	~	f3_BR						
f1_P	~~	f2_U	0.69	0.15	4.58	< .001***	0.39	0.98
f1_P	~~	f3_BR	0.49	0.42	1.15	.250	-0.34	1.31
f2_U	~~	f3_BR	0.71	0.59	1.22	.224	-0.44	1.86
EKM	~~	EKM	0.14	0.16	0.92	.359	-0.16	0.45
ERT	~~	ERT	0.54	0.14	3.97	< .001***	0.27	0.81
TASf1	~~	TASf1	0.90	0.09	10.18	< .001***	0.73	1.07
SET	~~	SET	0.88	0.11	8.21	< .001***	0.67	1.09

lhs	op	rhs	est.std	se	z	p	ci.lower	ci.upper
FP	~~	FP	0.57	0.17	3.41	.001***	0.24	0.90
CART	~~	CART	0.63	0.16	3.94	< .001***	0.32	0.94
RSMSf1	~~	RSMSf1	1.37	0.61	2.26	.024*	0.18	2.56
SNQ	~~	SNQ	1.05	0.06	16.66	< .001***	0.92	1.17
MBI	~~	MBI	1.01	0.03	36.93	< .001***	0.96	1.07
TQ	~~	TQ	0.78	0.12	6.65	< .001***	0.55	1.01
f1_P	~~	f1_P	1.00	0.00			1.00	1.00
f2_U	~~	f2_U	1.00	0.00			1.00	1.00
f3_BR	~~	f3_BR						

```
#print(nice_table(stcoefs_fav_SEM), preview = "docx") #Word format output
semPaths(fit_sem5,
         whatLabels = "std",  # Display standardized estimates
         layout = "tree2",      # Choose a layout, "tree" is one option
         style = "lisrel",     # Style for the diagram
         nCharNodes = 0,       # Full variable names
         sizeMan = 7,          # Size of manifest variables
         sizeLat = 10,         # Size of latent variables
         edge.label.cex = 0.75, # Adjust the size of the edge labels
         label.cex = 0.8,      # Adjust the size of the node labels
         residuals = TRUE,     # Show residuals
         intercepts = FALSE,   # Hide intercepts
         thresholds = FALSE,
         fade = FALSE,
         edge.color = "black"
         )
```


0.14
0.54
0.90
0.88
0.57
0.63
1.37
1.05
1.01
EKM
ERT
TASf1
SET
FP
CART
RSMSf1
SNQ
MBI
0.93
0.68
0.31
0.34
0.65
0.61
0.49
0.69
0.71
f1_P
f2_U
f3_BR
0.47
0.00
TQ
0.78