recalculating

Paula Wu (pw2551)

11/9/2022

Right now I haven't figured out how to map from the factor analysis results back to individual level values.

Let's do M2 first, retrieve the below variables

word list immediate (B3TWLITU), digits backward (B3TDBS), category fluency (B3TCTFLU), number series (B3TNSTOT), backward counting (B3TBKTOT), word list delayed (B3TWLDTU), mean of switch and non switch trial latencies (B3TSMXBB - SGST)

```
m2p3 = read_tsv("./data/ICPSR_25281/DS0001/25281-0001-Data.tsv")
```

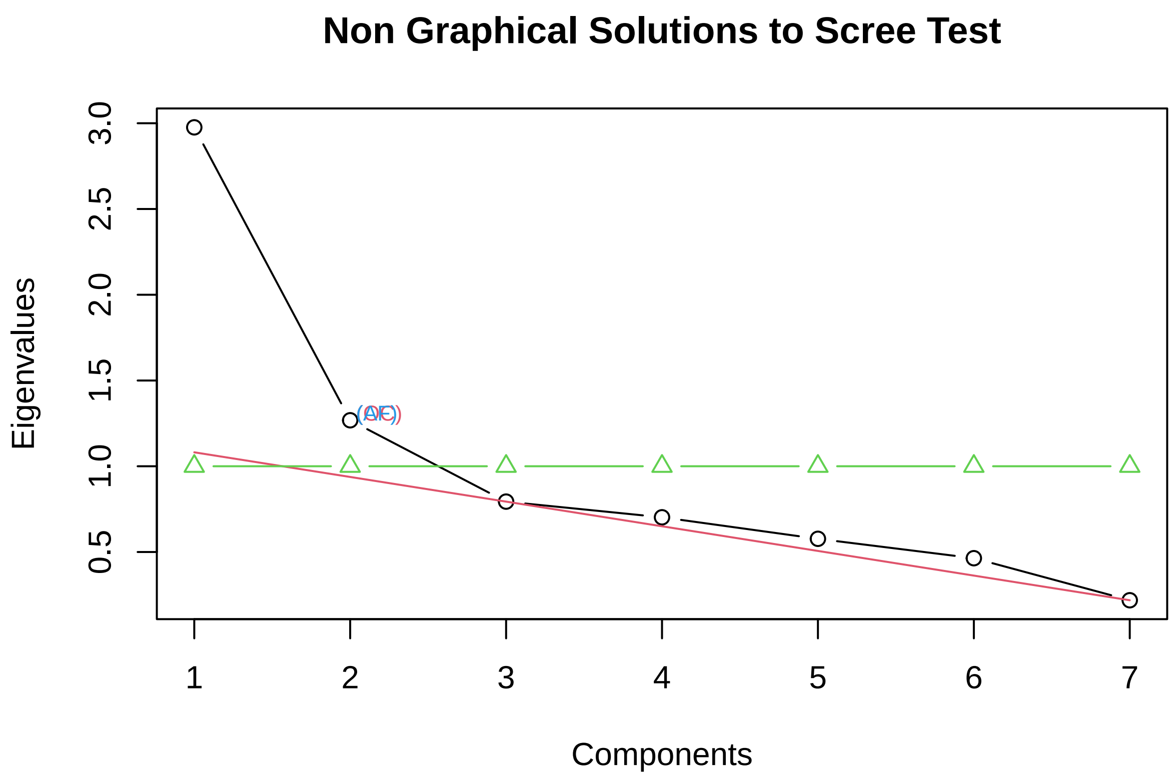
```
m2p3_selected_noid = m2p3 %>%
  select(B3TWLITU, B3TDBS, B3TCTFLU, B3TNSTOT, B3TBKTOT, B3TWLDTU, B3TSMXBB) %>%
  mutate(B3TSMXBB = -1 * B3TSMXBB) %>% # higher scores indicated faster response times
  filter(B3TWLITU != 98,  # filter out the invalid numbers
         B3TDBS != 98,
         B3TCTFLU != 98,
         B3TBKTOT != 998,
         B3TWLDTU != 98,
         B3TSMXBB != -98,
         B3TNSTOT != 8)
```

To start, we conducted the exploratory factor analysis. Let's plot Scree plot to figure out how many factors (or groupings) we should keep. One cut-off is having eigenvalues greater than 1.

```
summary(m2p3_selected_noid)
```

```
##     B3TWLITU          B3TDBS         B3TCTFLU        B3TNSTOT
##  Min.   : 0.000   Min.   :0.000   Min.   : 0.00   Min.   :0.000
##  1st Qu.: 5.000   1st Qu.:4.000   1st Qu.:15.00   1st Qu.:1.000
##  Median : 7.000   Median :5.000   Median :18.00   Median :2.000
##  Mean   : 6.771   Mean   :4.993   Mean   :18.81   Mean   :2.224
##  3rd Qu.: 8.000   3rd Qu.:6.000   3rd Qu.:23.00   3rd Qu.:3.000
##  Max.   :15.000   Max.   :8.000   Max.   :42.00   Max.   :5.000
##     B3TBKTOT         B3TWLDTU         B3TSMXBB
##  Min.   :-13.00   Min.   : 0.000   Min.   :-7.361
##  1st Qu.: 29.00   1st Qu.: 3.000   1st Qu.:-1.197
##  Median : 36.00   Median : 4.000   Median :-1.041
##  Mean   : 37.06   Mean   : 4.393   Mean   :-1.096
##  3rd Qu.: 44.00   3rd Qu.: 6.000   3rd Qu.:-0.930
##  Max.   : 90.00   Max.   :14.000   Max.   :-0.221
```

```
ev = eigen(cor(m2p3_selected_noid))
nS = nScree(x = ev$values)
plotnScree(nS, legend = F)
```



The green-triangle line is the cutoff. We have confirmed that there are 2 eigenvalues that are greater than 1. Next, I did the oblique rotation. This method would reorient the factors to better represent the manifest variables. This methodology is according to Grace's email.

```
# oblique rotation, with 2 factors
fit = factanal(m2p3_selected_noid, 2, rotation = "oblimin")
print(fit, digits = 2, cutoff = 0.3, sort = T)
```

```
##
## Call:
## factanal(x = m2p3_selected_noid, factors = 2, rotation = "oblimin")
##
## Uniquenesses:
## B3TWLITU   B3TDBS B3TCTFLU B3TNSTOT B3TBKTOT B3TWLDTU B3TSMXBB
##     0.20     0.77     0.69     0.61     0.38     0.24     0.69
##
## Loadings:
##          Factor1 Factor2
## B3TCTFLU  0.51
## B3TNSTOT  0.60
## B3TBKTOT  0.81
## B3TSMXBB  0.56
## B3TWLITU          0.89
## B3TWLDTU          0.88
## B3TDBS    0.31
##
##                Factor1 Factor2
## SS loadings       1.69    1.64
## Proportion Var    0.24    0.23
## Cumulative Var    0.24    0.48
```

```
##
## Factor Correlations:
##         Factor1 Factor2
## Factor1    1.00   -0.42
## Factor2   -0.42    1.00
##
## Test of the hypothesis that 2 factors are sufficient.
## The chi square statistic is 112.68 on 8 degrees of freedom.
## The p-value is 1.07e-20
```

We can see there are two different loadings, representing the correlation between the item and the factor. The factor2 have loadings with Word List Immediate (B3TWLITU) and Word List Delayed (B3TWLDTU) and factor1 have loadings with the rest. The result confirms the following quote:

"We also found a good fit for the two factor model using confirmatory factor analysis. The first factor represented Episodic Memory (B3TEM) and was comprised of Word List Immediate and Word List Delayed. The second factor, Executive Functioning (B3TEF), was made up of the remaining variables."

Confirmatory factor analysis

```
# define two factors
model <- '
EM = ~B3TCTFLU + B3TNSTOT + B3TBKTOT + B3TSMXBB + B3TDBS
EF = ~B3TWLITU + B3TWLDTU'

# fit the model
fit2 = cfa(model, data = m2p3_selected_noid)

# display summary output
summary(fit2,standardized = TRUE)
```

```
## lavaan 0.6-12 ended normally after 63 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        15
##
##   Number of observations                          4122
##
## Model Test User Model:
##
##   Test statistic                               359.964
##   Degrees of freedom                                13
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   EM =~
##     B3TCTFLU          1.000                               3.459    0.567
##     B3TNSTOT          0.285    0.010   28.428    0.000    0.987    0.647
```

```
##     B3TBKTOT          2.452    0.082   30.033    0.000    8.480    0.740
##     B3TSMXBB          0.044    0.002   25.453    0.000    0.153    0.541
##     B3TDBS            0.202    0.009   22.817    0.000    0.700    0.465
##   EF =~
##     B3TWLITU          1.000                               2.072    0.922
##     B3TWLDTU          1.068    0.031   34.916    0.000    2.213    0.846
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   EM ~~
##     EF                3.392    0.171   19.782    0.000    0.473    0.473
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .B3TCTFLU         25.198    0.650   38.754    0.000   25.198    0.678
##    .B3TNSTOT          1.354    0.039   35.151    0.000    1.354    0.581
##    .B3TBKTOT         59.407    2.094   28.370    0.000   59.407    0.452
##    .B3TSMXBB          0.056    0.001   39.631    0.000    0.056    0.707
##    .B3TDBS            1.781    0.043   41.597    0.000    1.781    0.784
##    .B3TWLITU          0.752    0.115    6.560    0.000    0.752    0.149
##    .B3TWLDTU          1.946    0.136   14.274    0.000    1.946    0.284
##     EM               11.964    0.690   17.329    0.000    1.000    1.000
##     EF                4.293    0.158   27.182    0.000    1.000    1.000
```

I am still little bit unsure of how to map the above results back to each participants as unstandardized values. But let's did the factor analysis on M3.

Repeated the same steps for M3

```
m3p3 = read.table(file = "./data/ICPSR_37095/DS0001/37095-0001-Data.tsv", sep = '\t', header = TRUE)
```

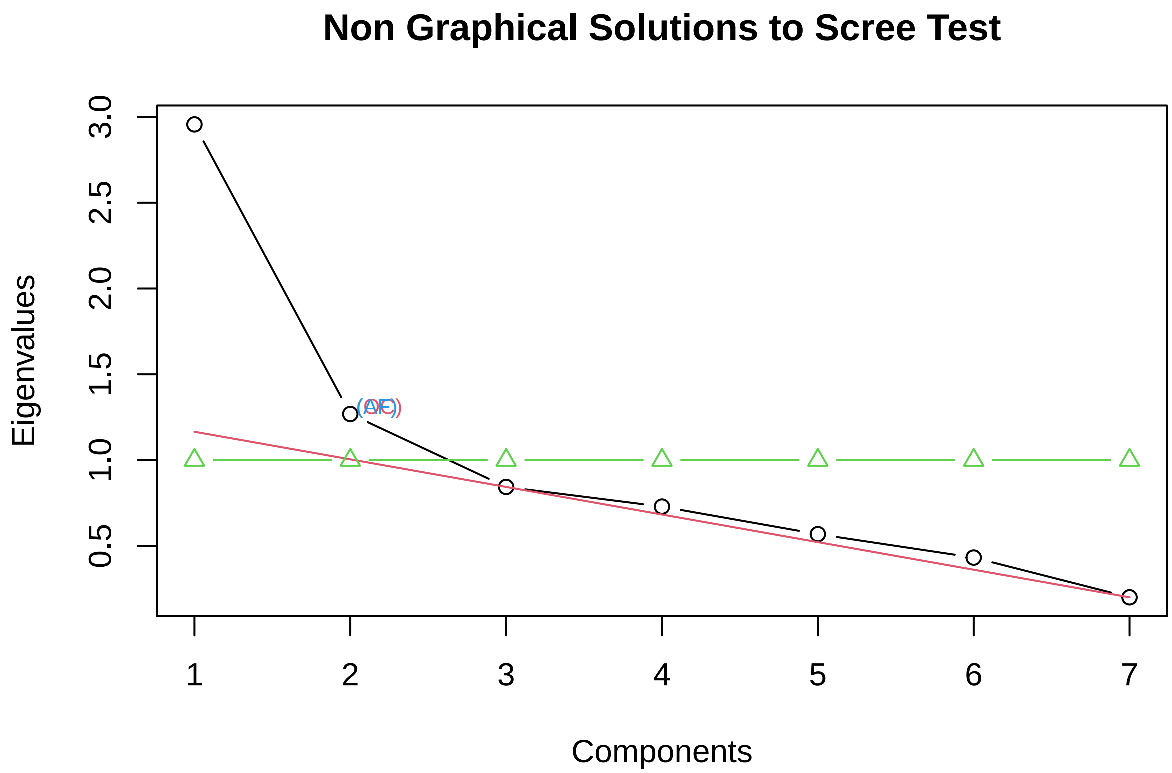
```
#B3TWLITU, B3TDBS, B3TCTFLU, B3TNSTOT, B3TBKTOT, B3TWLDTU, B3TSMXBB
m3p3_selected_noid = m3p3 %>%
  select(C3TWLITU, C3TDBS, C3TCTFLU, C3TNSTOT, C3TBKTOT, C3TWLDTU, C3TSMXBB) %>%
  mutate(C3TSMXBB = C3TSMXBB * -1) %>%
  filter(C3TWLITU != 98,  # filter out the invalid
         C3TDBS != 98,
         C3TCTFLU != 98,
         C3TBKTOT != 998,
         C3TWLDTU != 98,
         C3TSMXBB != -98,
         C3TNSTOT != 8) %>%
  drop_na()  # C3TBKTOT has some NA's, drop before proceeding
```

```
summary(m3p3_selected_noid)
```

```
##     C3TWLITU          C3TDBS         C3TCTFLU        C3TNSTOT
##  Min.   : 0.000   Min.   :0.000   Min.   : 0.00   Min.   :0.000
##  1st Qu.: 5.000   1st Qu.:4.000   1st Qu.:15.00   1st Qu.:1.000
##  Median : 7.000   Median :5.000   Median :18.00   Median :2.000
##  Mean   : 6.806   Mean   :4.999   Mean   :18.77   Mean   :2.192
##  3rd Qu.: 8.000   3rd Qu.:6.000   3rd Qu.:23.00   3rd Qu.:4.000
##  Max.   :15.000   Max.   :8.000   Max.   :42.00   Max.   :5.000
##     C3TBKTOT         C3TWLDTU         C3TSMXBB
##  Min.   :-12.00   Min.   : 0.000   Min.   :-7.6657
##  1st Qu.: 28.00   1st Qu.: 2.000   1st Qu.:-1.5834
```

```
##  Median : 36.00   Median : 4.000   Median :-1.3740
##  Mean   : 35.87   Mean   : 4.384   Mean   :-1.4079
##  3rd Qu.: 43.00   3rd Qu.: 6.000   3rd Qu.:-1.1653
##  Max.   : 90.00   Max.   :15.000   Max.   :-0.7027
```

```
ev2 = eigen(cor(m3p3_selected_noid))
nS2 = nScree(x = ev2$values)
plotnScree(nS2, legend = F)
```



Again, we confirmed from the above graph that we have two eigenvalues greater than 1.

```
# oblique rotation, with 2 factors
fit3 = factanal(m3p3_selected_noid, 2, rotation = "oblimin")
print(fit3, digits = 2, cutoff = 0.3, sort = T)
```

```
##
## Call:
## factanal(x = m3p3_selected_noid, factors = 2, rotation = "oblimin")
##
## Uniquenesses:
## C3TWLITU   C3TDBS C3TCTFLU C3TNSTOT C3TBKTOT C3TWLDTU C3TSMXBB
##     0.21     0.76     0.69     0.52     0.36     0.19     0.82
##
## Loadings:
##          Factor1 Factor2
## C3TCTFLU  0.52
## C3TNSTOT  0.69
## C3TBKTOT  0.82
## C3TWLITU          0.87
## C3TWLDTU          0.91
## C3TDBS    0.35
## C3TSMXBB  0.42
```

```
##
##                  Factor1 Factor2
## SS loadings         1.71    1.66
## Proportion Var      0.24    0.24
## Cumulative Var      0.24    0.48
##
## Factor Correlations:
##         Factor1 Factor2
## Factor1    1.00    0.41
## Factor2    0.41    1.00
##
## Test of the hypothesis that 2 factors are sufficient.
## The chi square statistic is 75.59 on 8 degrees of freedom.
## The p-value is 3.76e-13
```

Confirmatory factor analysis

```
# define two factors
model2 <- '
EM = ~C3TCTFLU + C3TNSTOT + C3TBKTOT + C3TSMXBB + C3TDBS
EF = ~C3TWLITU + C3TWLDTU'

# fit the model
fit4 = cfa(model2, data = m3p3_selected_noid)

# display summary output
summary(fit4,standardized = TRUE)
```

```
## lavaan 0.6-12 ended normally after 65 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        15
##
##   Number of observations                          2987
##
## Model Test User Model:
##
##   Test statistic                               229.399
##   Degrees of freedom                                13
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   EM =~
##     C3TCTFLU          1.000                               3.421    0.570
##     C3TNSTOT          0.320    0.012   25.685    0.000    1.095    0.698
##     C3TBKTOT          2.640    0.100   26.463    0.000    9.030    0.768
##     C3TSMXBB          0.045    0.002   18.366    0.000    0.155    0.424
```

```
##     C3TDBS             0.207    0.010   19.973    0.000    0.708    0.473
##   EF =~
##     C3TWLITU           1.000                               2.243    0.949
##     C3TWLDTU           1.016    0.034   30.126    0.000    2.278    0.840
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   EM ~~
##     EF                 3.496    0.207   16.885    0.000    0.456    0.456
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .C3TCTFLU          24.352    0.731   33.305    0.000   24.352    0.675
##    .C3TNSTOT           1.265    0.046   27.662    0.000    1.265    0.513
##    .C3TBKTOT          56.821    2.530   22.459    0.000   56.821    0.411
##    .C3TSMXBB           0.110    0.003   36.255    0.000    0.110    0.820
##    .C3TDBS             1.743    0.049   35.497    0.000    1.743    0.777
##    .C3TWLITU           0.555    0.155    3.583    0.000    0.555    0.099
##    .C3TWLDTU           2.162    0.169   12.812    0.000    2.162    0.294
##     EM                11.703    0.782   14.973    0.000    1.000    1.000
##     EF                 5.031    0.211   23.849    0.000    1.000    1.000
```