SEM Code

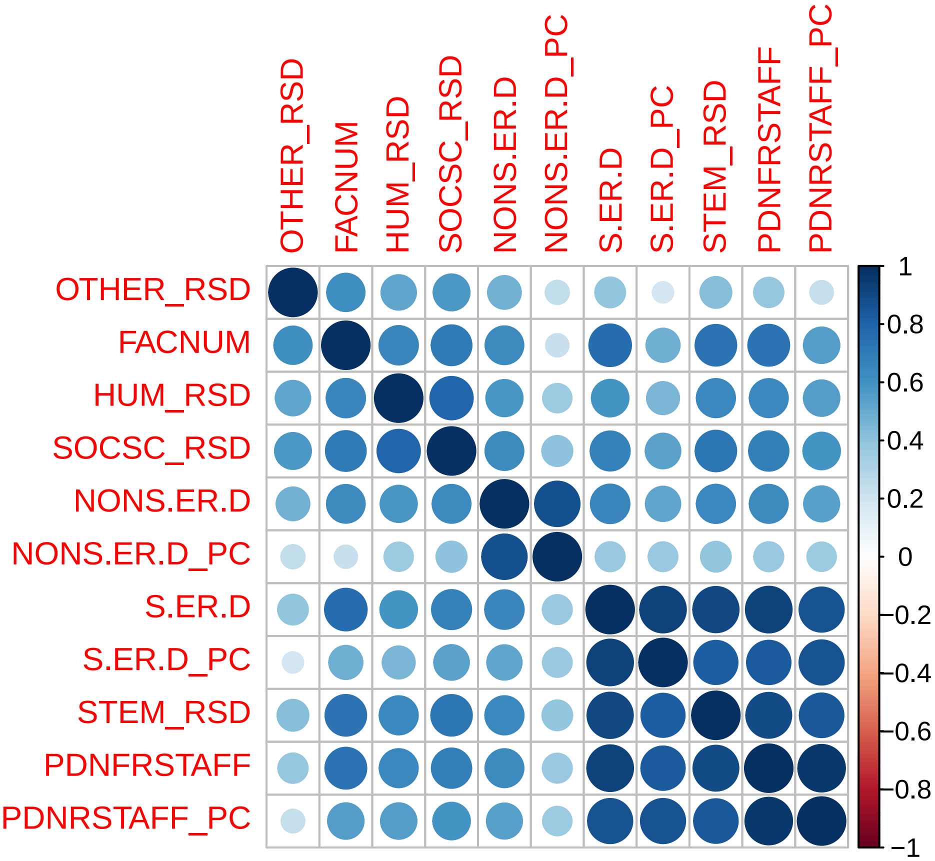
Paul

March 6, 2018

Correlation Plot

We used the correlation plot below to inform the two latent factors that we are using.

```r
setwd("~/Carnegie-SEM/data")
cc2015 <- read.csv("CC2015data.csv",header = TRUE)

#######2015################
cc2015.full <- read.csv("CC2015data.csv", header = TRUE, as.is = TRUE)
#updated file
#cc2015.full <- read.csv("Updated2015.csv", header = TRUE)

cc2015 <- cc2015.full[(cc2015.full$BASIC2015>14&cc2015.full$BASIC2015<18),]
cc2015$BASIC2015 <- factor(cc2015$BASIC2015)


#function for ranking the data
minrank <- function(x){rank(x, ties.method = "min")}

#dataset that we want to use
cc2015Ps<-
  na.omit(cc2015[,c("NAME","BASIC2010","BASIC2015","FACNUM","HUM_RSD",
                    "OTHER_RSD","SOCSC_RSD","STEM_RSD","PDNFRSTAFF","S.ER.D","NONS.ER.D")])

#calculate the ranked data
cc2015.r <- data.frame(cc2015Ps[,1:3],sapply(cc2015Ps[,-c(1:3)],minrank))

cc2015percap <- cc2015Ps[,c("PDNFRSTAFF","S.ER.D","NONS.ER.D")]/cc2015Ps$FACNUM
colnames(cc2015percap) <- c("PDNRSTAFF_PC", "S.ER.D_PC", "NONS.ER.D_PC")
cc2015percap.r<-data.frame(sapply(cc2015percap,minrank))
cc2015_r <- cbind(cc2015.r, cc2015percap.r)

cc2015_matrix2 <- as.matrix(cc2015_r[-c(1:3)])
corrmatrix <- Hmisc::rcorr(cc2015_matrix2)
corrplot::corrplot(corrmatrix$r, order="hclust")
```



SEM Code:

The SEM fit below on the STEM and Non-STEM latent factors is given with the following code.

```
model4 <- '
HUMANITIES=~HUM_RSD+OTHER_RSD+SOCSC_RSD+NONS.ER.D+FACNUM
STEM=~STEM_RSD+PDNFRSTAFF+S.ER.D+FACNUM
Aggregate=~HUMANITIES+STEM'

lavaan_sem_new <- lavaan::sem(model4, data=cc2015_r, std.lv=TRUE,
                              orthogonal=FALSE, se="robust.huber.white")
lavaan::summary(lavaan_sem_new, standardized=TRUE, fit.measures=TRUE)
```

```
## lavaan (0.5-23.1097) converged normally after 128 iterations
##
##   Number of observations                           276
##
##   Estimator                                         ML
##   Minimum Function Test Statistic              110.024
##   Degrees of freedom                                17
##   P-value (Chi-square)                           0.000
##
## Model test baseline model:
##
##   Minimum Function Test Statistic             2223.162
##   Degrees of freedom                                28
##   P-value                                       0.000
##
## User model versus baseline model:
##
##   Comparative Fit Index (CFI)                   0.958
##   Tucker-Lewis Index (TLI)                      0.930
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)            -11847.548
##   Loglikelihood unrestricted model (H1)    -11792.536
##
##   Number of free parameters                        27
##   Akaike (AIC)                              23749.096
##   Bayesian (BIC)                            23846.847
##   Sample-size adjusted Bayesian (BIC)       23761.234
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                         0.141
##   90 Percent Confidence Interval          0.116  0.166
##   P-value RMSEA <= 0.05                         0.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                          0.041
##
## Parameter Estimates:
##
##   Information                                Observed
##   Standard Errors                  Robust.huber.white
##
## Latent Variables:
##                     Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   HUMANITIES =~
##     HUM_RSD           38.108    9.787    3.894    0.000   81.310    0.847
##     OTHER_RSD         24.120    6.468    3.729    0.000   51.463    0.639
##     SOCSC_RSD         37.677    9.672    3.895    0.000   80.390    0.906
##     NONS.ER.D         27.306    6.978    3.913    0.000   58.262    0.729
##     FACNUM            18.010    6.675    2.698    0.007   38.427    0.482
##   STEM =~
##     STEM_RSD          33.562    8.626    3.891    0.000   77.096    0.939
##     PDNFRSTAFF        34.448    8.124    4.240    0.000   79.131    0.953
##     S.ER.D            33.529    8.408    3.988    0.000   77.021    0.967
##     FACNUM            13.886    5.838    2.379    0.017   31.897    0.400
##   Aggregate =~
##     HUMANITIES         1.885    0.615    3.065    0.002    0.883    0.883
##     STEM               2.068    0.634    3.260    0.001    0.900    0.900
##
## Intercepts:
##                     Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .HUM_RSD          123.025    5.781   21.281    0.000  123.025    1.281
##    .OTHER_RSD        137.188    4.847   28.305    0.000  137.188    1.704
##    .SOCSC_RSD        130.529    5.338   24.451    0.000  130.529    1.472
##    .NONS.ER.D       138.344    4.811   28.759    0.000  138.344    1.731
##    .FACNUM          138.446    4.797   28.859    0.000  138.446    1.737
##    .STEM_RSD        136.554    4.942   27.632    0.000  136.554    1.663
##    .PDNFRSTAFF      136.101    4.997   27.238    0.000  136.101    1.640
##    .S.ER.D          138.500    4.796   28.879    0.000  138.500    1.738
##     HUMANITIES        0.000                               0.000    0.000
##     STEM              0.000                               0.000    0.000
##     Aggregate         0.000                               0.000    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .HUM_RSD        2612.532  366.089    7.136    0.000 2612.532    0.283
##    .OTHER_RSD      3835.029  334.063   11.480    0.000 3835.029    0.592
##    .SOCSC_RSD      1403.253  236.802    5.926    0.000 1403.253    0.178
##    .NONS.ER.D      2992.509  305.049    9.810    0.000 2992.509    0.469
##    .FACNUM         1908.187  223.671    8.531    0.000 1908.187    0.300
##    .STEM_RSD        796.821  117.031    6.809    0.000  796.821    0.118
##    .PDNFRSTAFF      629.148  162.047    3.883    0.000  629.148    0.091
##    .S.ER.D          415.725   88.612    4.692    0.000  415.725    0.065
##     HUMANITIES        1.000                               0.220    0.220
##     STEM              1.000                               0.190    0.190
##     Aggregate         1.000                               1.000    1.000
```

```
CCScores <- as.data.frame(lavaan::predict(lavaan_sem_new))
rownames(CCScores) <- cc2015Ps$NAME
```