Using ABACAB and KU-UNC as example for analyses discussed in 525

Eric Youngstrom, PhD

Spring Semester 2023

# Overview

This notebook has a bunch of worked examples for different visualizations (especially in Exploratory Data Analysis mode) and statistical analyses.

You will definitely not need all of these for your project, and you may need or want some other things that are not covered in these examples.

A typical student project would use a subset of these chunks, with this sequence as a common work flow:

1. library to load your packages (haven, psych, ggplot, tidyverse, etc.)
2. load your data (usually haven for us)
3. select your variables (look at the modelvars example; remember to include age and sex, maybe race as ways of describing participants)
4. psych::describe to get the descriptives (N, min, max, M, SD, skew, kurtosis)
5. pairs.panels – are the relationships linear? Do the histograms look okay? etc…
6. na.omit – get rid of missing data
7. run analysis (often setCor or lm)
8. make table or figure – could be ggplot, or could be the tricked out version of pairs.panels with the asterisks, etc. (code is in the class notes for today). That covers the bases for most people’s posters. Everything you need (unless you are running logistic regression or something else “off menu”) should be in the two notebooks, and usually all is in the second one.
9. Write out “sufficient data” for someone else to be able to replicate your results.

One way of organizing your final project would be to copy and paste the relevant chunks into a new notebook that you make for it. That will let you streamline things and customize it as you wish.

library(tidyverse) #wrangling

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.4 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.4   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(haven) #importing SPSS and other data formats  
library(ggplot2) #deluxe graphics  
library(ggExtra) #extra fancy graphics  
library(psych) #one package that covers ~80% of what we use in psychology

##   
## Attaching package: 'psych'  
##   
## The following objects are masked from 'package:ggplot2':  
##   
## %+%, alpha

library(rstatix) #a package that offers useful options for correlation tables, ANOVA

##   
## Attaching package: 'rstatix'  
##   
## The following object is masked from 'package:stats':  
##   
## filter

library(vcd) #visual displays of categorical data

## Loading required package: grid

library(mosaic) #more tools for categorical data

## Registered S3 method overwritten by 'mosaic':  
## method from   
## fortify.SpatialPolygonsDataFrame ggplot2  
##   
## The 'mosaic' package masks several functions from core packages in order to add   
## additional features. The original behavior of these functions should not be affected by this.  
##   
## Attaching package: 'mosaic'  
##   
## The following object is masked from 'package:Matrix':  
##   
## mean  
##   
## The following object is masked from 'package:vcd':  
##   
## mplot  
##   
## The following objects are masked from 'package:rstatix':  
##   
## cor\_test, prop\_test, t\_test  
##   
## The following objects are masked from 'package:psych':  
##   
## logit, rescale  
##   
## The following objects are masked from 'package:dplyr':  
##   
## count, do, tally  
##   
## The following object is masked from 'package:purrr':  
##   
## cross  
##   
## The following object is masked from 'package:ggplot2':  
##   
## stat  
##   
## The following objects are masked from 'package:stats':  
##   
## binom.test, cor, cor.test, cov, fivenum, IQR, median, prop.test,  
## quantile, sd, t.test, var  
##   
## The following objects are masked from 'package:base':  
##   
## max, mean, min, prod, range, sample, sum

library(rockchalk) #useful for intermediate/advanced regression analyses

##   
## Attaching package: 'rockchalk'  
##   
## The following object is masked from 'package:dplyr':  
##   
## summarize

options(digits = 3, scipen = NULL)  
knitr::opts\_chunk$set(echo = TRUE)

Note: v3.2 of the data had an error in item 10 (it was accidentally over-written by the 10 item scale score)

# [] come back to this and load the datasets for class  
abacab <- read\_sav("Ultimate Enchilada v3.3 chopped for 525 v4.sav")   
ku2017<-read\_spss("Korea\_Spring\_Fall\_2017 v5 (safe)+2018+2019+Spr2020 April 2023 fix.sav")

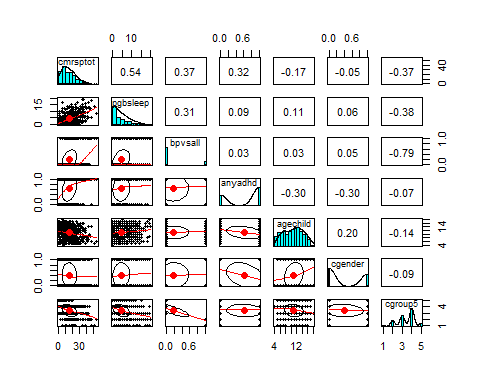
modelvars<-abacab %>%   
 select(cmrsptot, pgbsleep, bpvsall, anyadhd, agechild, cgender, cgroup5)   
#make data subset

# Checking assumptions

describe(modelvars) #Get N, min & max (compare to source docs), skew & kurtosis

## vars n mean sd median trimmed mad min max range skew kurtosis  
## cmrsptot 1 542 15.52 10.60 14 14.53 10.38 0 55 55 0.82 0.33  
## pgbsleep 2 817 4.50 4.39 3 3.89 4.45 0 20 20 1.04 0.50  
## bpvsall 3 825 0.18 0.39 0 0.11 0.00 0 1 1 1.63 0.64  
## anyadhd 4 825 0.63 0.48 1 0.66 0.00 0 1 1 -0.54 -1.71  
## agechild 5 825 10.89 3.43 11 10.89 4.45 4 18 14 -0.04 -1.05  
## cgender 6 825 0.40 0.49 0 0.38 0.00 0 1 1 0.40 -1.84  
## cgroup5 7 825 3.40 0.97 4 3.44 1.48 1 5 4 -0.54 -0.20  
## se  
## cmrsptot 0.46  
## pgbsleep 0.15  
## bpvsall 0.01  
## anyadhd 0.02  
## agechild 0.12  
## cgender 0.02  
## cgroup5 0.03

pairs.panels(modelvars) #Look at histograms, scatterplots, LOESS smoother

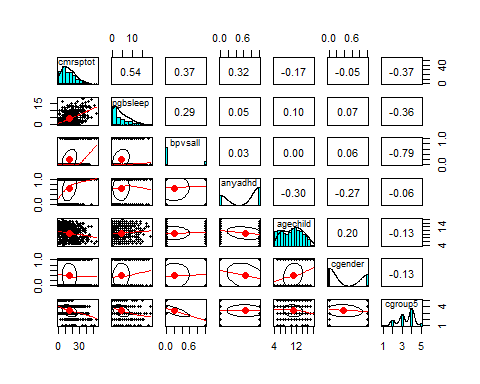


## Basic missing data handling – listwise deletion Contrast this with “pairwise,” earlier.

modelvars<-na.omit(modelvars)  
describe(modelvars) #Get N, min & max (compare to source docs), skew & kurtosis

## vars n mean sd median trimmed mad min max range skew kurtosis  
## cmrsptot 1 542 15.52 10.60 14 14.53 10.38 0 55 55 0.82 0.33  
## pgbsleep 2 542 4.20 4.10 3 3.63 4.45 0 19 19 1.05 0.57  
## bpvsall 3 542 0.19 0.40 0 0.12 0.00 0 1 1 1.55 0.39  
## anyadhd 4 542 0.64 0.48 1 0.68 0.00 0 1 1 -0.59 -1.65  
## agechild 5 542 10.63 3.52 11 10.59 4.45 4 18 14 0.01 -1.09  
## cgender 6 542 0.39 0.49 0 0.36 0.00 0 1 1 0.46 -1.79  
## cgroup5 7 542 3.37 0.95 4 3.40 1.48 1 5 4 -0.45 -0.34  
## se  
## cmrsptot 0.46  
## pgbsleep 0.18  
## bpvsall 0.02  
## anyadhd 0.02  
## agechild 0.15  
## cgender 0.02  
## cgroup5 0.04

pairs.panels(modelvars) #Look at histograms, scatterplots, LOESS smoother



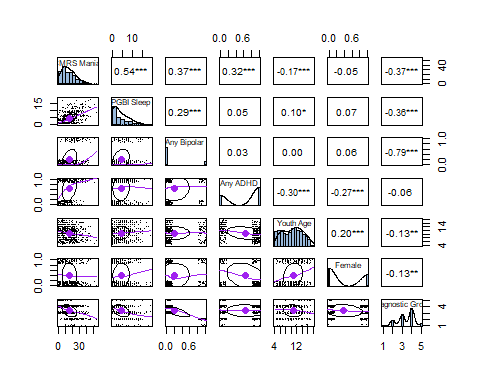
## Tricked out pairs.panels

This example shows how to change a lot of the settings in pair.panels in order to make the figure more “presentation worthy.” Customizing the names on the diagonal is the most complicated piece of the code. You *must* type in your construct names, or it will use the labels provided in the example instead, and you need to edit the length of the list so that it matches the number and order of the variables listed in your object (e.g., example has 7 variables; you many only have 5).

###Pro Tips if using as your main figure Focus: If this is going to be your main figure, you should limit it to only include the variables in your model (e.g., don’t need to include demographics, covariates, or extra variables from sensitivity analysis). Here, less is more: having fewer variables will focus attention, and it will let each of the remaining plots take up more room and show more detail.

Order: Think about what order you want to put the variables. The dependent variable probably should come last, because then it would be positioned on the y-axis on the scatterplots along the bottom row. There may be an order to use for the predictors that could help guide attention, too. The most important could be first (or second to last) for instance.

pairs.panels(modelvars,   
 stars=TRUE, #drops the asterisks next to the correlations  
 jiggle=TRUE, #jiggles/jitters the dots,  
 factor = .8, #0 to 5, how much jiggle to add  
 labels = c("CMRS Mania", "PGBI Sleep", "Any Bipolar", "Any ADHD", "Youth Age", "Female", "Diagnostic Group"), #labels the diagonal (no more variable names!)  
 hist.col="#99BADD", #changes color of histogram  
 col="purple",  
 cex.cor=0.9, #changes the size of the font for the correlations   
 pch=".") #makes the dot in the scatterplots smaller



# Statistical analyses

The next sections include code for running most of the analyses you see commonly in Psychology statistic courses or in student projects. The choice of the analysis depends on a combination of your research question and the level of measurement of your variables (especially nominal/categorical vs dimensional-ish).

You may need to check some additional assumptions depending on your choice of analysis.

These examples would be places to look for “Chunk 7 – run analyses,” and also maybe more checking assumptions (augmenting what we get from psych::describe and psych::pairs.panels). There are some examples of prototypes for presentation quality figures with ggplot2, as well.

# Correlations

Remember to check assumptions – are the variables continuous? If yes, are they normally distributed?

If one or both of the variables are *not* normally distributed, then switch to Spearman’s rho instead of the usual Pearson’s r.

pairs.panels and describe are a good combination to check these. boxplots are a quick visual check for univariate outliers.

psych::corr.test(modelvars)

## Call:psych::corr.test(x = modelvars)  
## Correlation matrix   
## cmrsptot pgbsleep bpvsall anyadhd agechild cgender cgroup5  
## cmrsptot 1.00 0.54 0.37 0.32 -0.17 -0.05 -0.37  
## pgbsleep 0.54 1.00 0.29 0.05 0.10 0.07 -0.36  
## bpvsall 0.37 0.29 1.00 0.03 0.00 0.06 -0.79  
## anyadhd 0.32 0.05 0.03 1.00 -0.30 -0.27 -0.06  
## agechild -0.17 0.10 0.00 -0.30 1.00 0.20 -0.13  
## cgender -0.05 0.07 0.06 -0.27 0.20 1.00 -0.13  
## cgroup5 -0.37 -0.36 -0.79 -0.06 -0.13 -0.13 1.00  
## Sample Size   
## [1] 542  
## Probability values (Entries above the diagonal are adjusted for multiple tests.)   
## cmrsptot pgbsleep bpvsall anyadhd agechild cgender cgroup5  
## cmrsptot 0.00 0.00 0.00 0.00 0.00 1.00 0.00  
## pgbsleep 0.00 0.00 0.00 1.00 0.16 0.79 0.00  
## bpvsall 0.00 0.00 0.00 1.00 1.00 0.96 0.00  
## anyadhd 0.00 0.27 0.42 0.00 0.00 0.00 0.96  
## agechild 0.00 0.02 0.92 0.00 0.00 0.00 0.03  
## cgender 0.28 0.11 0.16 0.00 0.00 0.00 0.03  
## cgroup5 0.00 0.00 0.00 0.17 0.00 0.00 0.00  
##   
## To see confidence intervals of the correlations, print with the short=FALSE option

psych::corr.test(modelvars, method="spearman") %>% print(digits=3)#Default is Pearson; also does Kendall Tau

## Call:psych::corr.test(x = modelvars, method = "spearman")  
## Correlation matrix   
## cmrsptot pgbsleep bpvsall anyadhd agechild cgender cgroup5  
## cmrsptot 1.000 0.527 0.360 0.338 -0.161 -0.059 -0.352  
## pgbsleep 0.527 1.000 0.308 0.056 0.095 0.063 -0.399  
## bpvsall 0.360 0.308 1.000 0.035 0.006 0.061 -0.727  
## anyadhd 0.338 0.056 0.035 1.000 -0.300 -0.267 -0.031  
## agechild -0.161 0.095 0.006 -0.300 1.000 0.204 -0.149  
## cgender -0.059 0.063 0.061 -0.267 0.204 1.000 -0.132  
## cgroup5 -0.352 -0.399 -0.727 -0.031 -0.149 -0.132 1.000  
## Sample Size   
## [1] 542  
## Probability values (Entries above the diagonal are adjusted for multiple tests.)   
## cmrsptot pgbsleep bpvsall anyadhd agechild cgender cgroup5  
## cmrsptot 0.000 0.000 0.000 0.000 0.002 0.988 0.000  
## pgbsleep 0.000 0.000 0.000 0.988 0.215 0.988 0.000  
## bpvsall 0.000 0.000 0.000 1.000 1.000 0.988 0.000  
## anyadhd 0.000 0.191 0.418 0.000 0.000 0.000 1.000  
## agechild 0.000 0.027 0.898 0.000 0.000 0.000 0.005  
## cgender 0.173 0.141 0.159 0.000 0.000 0.000 0.019  
## cgroup5 0.000 0.000 0.000 0.466 0.000 0.002 0.000  
##   
## To see confidence intervals of the correlations, print with the short=FALSE option

cor.mat <- modelvars %>% cor\_mat() #table of coefficients (Pearson as default)  
cor.mat

## # A tibble: 7 × 8  
## rowname cmrsptot pgbsleep bpvsall anyadhd agechild cgender cgroup5  
## \* <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 cmrsptot 1 0.54 0.37 0.32 -0.17 -0.046 -0.37   
## 2 pgbsleep 0.54 1 0.29 0.047 0.1 0.068 -0.36   
## 3 bpvsall 0.37 0.29 1 0.035 0.0042 0.061 -0.79   
## 4 anyadhd 0.32 0.047 0.035 1 -0.3 -0.27 -0.059  
## 5 agechild -0.17 0.1 0.0042 -0.3 1 0.2 -0.13   
## 6 cgender -0.046 0.068 0.061 -0.27 0.2 1 -0.13   
## 7 cgroup5 -0.37 -0.36 -0.79 -0.059 -0.13 -0.13 1

cor.mat %>% cor\_get\_pval() #table of p values

## # A tibble: 7 × 8  
## rowname cmrsptot pgbsleep bpvsall anyadhd agechild cgender cgroup5  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 cmrsptot 0 1 e-42 7.54e- 19 1.75e-14 6.33e- 5 2.8 e- 1 2.04e- 19  
## 2 pgbsleep 1 e-42 0 1.12e- 11 2.75e- 1 1.99e- 2 1.14e- 1 1.84e- 18  
## 3 bpvsall 7.54e-19 1.12e-11 0 4.18e- 1 9.23e- 1 1.59e- 1 2.49e-116  
## 4 anyadhd 1.75e-14 2.75e- 1 4.18e- 1 0 9.31e-13 2.55e-10 1.72e- 1  
## 5 agechild 6.33e- 5 1.99e- 2 9.23e- 1 9.31e-13 0 1.49e- 6 3.51e- 3  
## 6 cgender 2.8 e- 1 1.14e- 1 1.59e- 1 2.55e-10 1.49e- 6 0 3.25e- 3  
## 7 cgroup5 2.04e-19 1.84e-18 2.49e-116 1.72e- 1 3.51e- 3 3.25e- 3 0

cor.mat %>%  
 cor\_as\_symbols() %>%  
 pull\_lower\_triangle() #triangle of asterisks

## rowname cmrsptot pgbsleep bpvsall anyadhd agechild cgender cgroup5  
## 1 cmrsptot   
## 2 pgbsleep +   
## 3 bpvsall . .   
## 4 anyadhd .   
## 5 agechild .   
## 6 cgender .   
## 7 cgroup5 . . \*

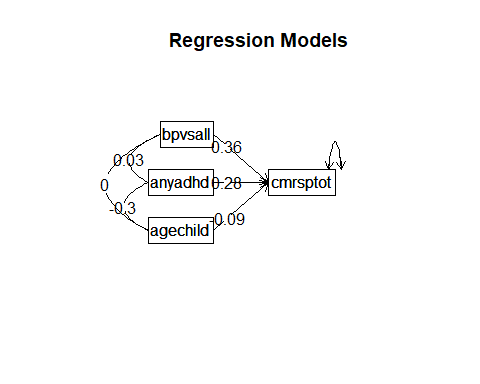
# Regression analyses using psych package

## Checklist for regression analysis ###Dependent variable \* Is your dependent variable continuous?

(If it’s a categorical variable with two options, consider logistic regression instead) \* Is your DV normally distributed (more or less) - Skew and Kurtosis < |3| - no extreme outliers in boxplot - If not, would a transformation fix?

* Predictors If dimensional, also check for outliers and skew, kurtosis as above (though normality is less crucial for predictor – but look out for outliers! They can be “high leverage” like in that Anscomb Quartet scatterplot example) If they are categorical, are they already dummy coded? If not, use as.factor() and R will dummy code them for us! (also possible to pick which group is the comparison group)

basic<-setCor(cmrsptot~ bpvsall+anyadhd+agechild, data=modelvars, std=TRUE)



plot(basic)  
summary(basic)

##   
## Multiple Regression from raw data   
## setCor(y = cmrsptot ~ bpvsall + anyadhd + agechild, data = modelvars,   
## std = TRUE)  
##   
## Multiple Regression from matrix input   
##   
## Beta weights   
## cmrsptot  
## (Intercept) 0.000  
## bpvsall 0.359  
## anyadhd 0.283  
## agechild -0.088  
##   
## Multiple R   
## cmrsptot   
## 0.49   
##   
## Multiple R2   
## cmrsptot   
## 0.24   
##   
## Cohen's set correlation R2   
## [1] 0.24  
##   
## Squared Canonical Correlations  
## NULL

basic

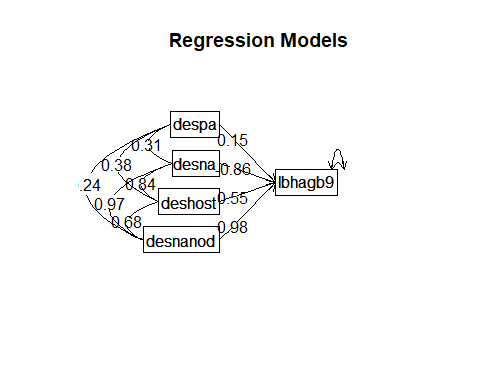
## Call: setCor(y = cmrsptot ~ bpvsall + anyadhd + agechild, data = modelvars,   
## std = TRUE)  
##   
## Multiple Regression from raw data   
##   
## DV = cmrsptot   
## slope se t p lower.ci upper.ci VIF Vy.x  
## (Intercept) 0.00 0.04 0.00 1.0e+00 -0.07 0.07 1.0 0.00  
## bpvsall 0.36 0.04 9.52 5.6e-20 0.28 0.43 1.0 0.13  
## anyadhd 0.28 0.04 7.15 2.8e-12 0.20 0.36 1.1 0.09  
## agechild -0.09 0.04 -2.22 2.7e-02 -0.17 -0.01 1.1 0.01  
##   
## Residual Standard Error = 0.88 with 538 degrees of freedom  
##   
## Multiple Regression  
## R R2 Ruw R2uw Shrunken R2 SE of R2 overall F df1 df2 p  
## cmrsptot 0.49 0.24 0.33 0.11 0.23 0.03 56 3 538 1.68e-31

## Collinearity in regression

book<-setCor(lbhagb9~despa+ desna+ deshost+desnanod, data=abacab, std=TRUE)

## In smc, smcs > 1 were set to 1.0

## Warning in sqrt(MSE[i] \* diag(x.inv)): NaNs produced



plot(book)  
summary(book)

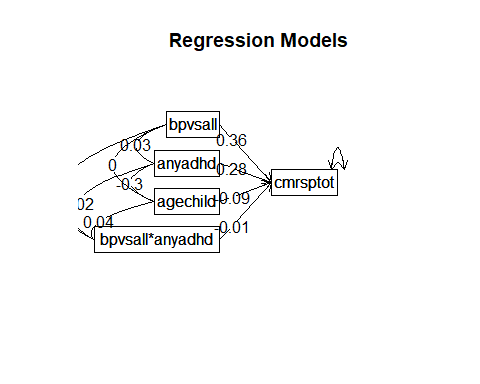
##   
## Multiple Regression from raw data   
## setCor(y = lbhagb9 ~ despa + desna + deshost + desnanod, data = abacab,   
## std = TRUE)  
##   
## Multiple Regression from matrix input   
##   
## Beta weights   
## lbhagb9  
## (Intercept) 0.00  
## despa 0.15  
## desna -0.86  
## deshost 0.55  
## desnanod 0.98  
##   
## Multiple R   
## lbhagb9   
## 0.62   
##   
## Multiple R2   
## lbhagb9   
## 0.38   
##   
## Cohen's set correlation R2   
## [1] 0.38  
##   
## Squared Canonical Correlations  
## NULL

book

## Call: setCor(y = lbhagb9 ~ despa + desna + deshost + desnanod, data = abacab,   
## std = TRUE)  
##   
## Multiple Regression from raw data   
##   
## DV = lbhagb9   
## slope se t p lower.ci upper.ci VIF Vy.x  
## (Intercept) 0.00 0.03 0.00 1.0e+00 -0.05 0.05 1.00 0.00  
## despa 0.15 0.03 5.08 4.6e-07 0.09 0.21 1.17 0.05  
## desna -0.86 NaN NaN NaN NaN NaN Inf -0.51  
## deshost 0.55 NaN NaN NaN NaN NaN Inf 0.30  
## desnanod 0.98 NaN NaN NaN NaN NaN Inf 0.55  
##   
## Residual Standard Error = 0.79 with 823 degrees of freedom  
##   
## Multiple Regression  
## R R2 Ruw R2uw Shrunken R2 SE of R2 overall F df1 df2 p  
## lbhagb9 0.62 0.38 0.62 0.38 0.38 0.03 128 4 823 3.7e-85

The colon or \*sets up the interaction effect for you (no manual construction of the interaction term) ##Interaction (aka moderation) in regression

eayInt<-setCor(cmrsptot~bpvsall\*anyadhd+agechild, data=modelvars, std=TRUE)



plot(eayInt)  
summary(eayInt)

##   
## Multiple Regression from raw data   
## setCor(y = cmrsptot ~ bpvsall \* anyadhd + agechild, data = modelvars,   
## std = TRUE)  
##   
## Multiple Regression from matrix input   
##   
## Beta weights   
## cmrsptot  
## (Intercept) 0.0000  
## bpvsall 0.3591  
## anyadhd 0.2824  
## agechild -0.0874  
## bpvsall\*anyadhd -0.0058  
##   
## Multiple R   
## cmrsptot   
## 0.49   
##   
## Multiple R2   
## cmrsptot   
## 0.24   
##   
## Cohen's set correlation R2   
## [1] 0.24  
##   
## Squared Canonical Correlations  
## NULL

eayInt

## Call: setCor(y = cmrsptot ~ bpvsall \* anyadhd + agechild, data = modelvars,   
## std = TRUE)  
##   
## Multiple Regression from raw data   
##   
## DV = cmrsptot   
## slope se t p lower.ci upper.ci VIF Vy.x  
## (Intercept) 0.00 0.04 0.00 1.0e+00 -0.07 0.07 1.00 0.00  
## bpvsall 0.36 0.04 9.51 6.3e-20 0.28 0.43 1.00 0.13  
## anyadhd 0.28 0.04 7.15 2.9e-12 0.20 0.36 1.10 0.09  
## agechild -0.09 0.04 -2.21 2.8e-02 -0.17 -0.01 1.10 0.01  
## bpvsall\*anyadhd -0.01 0.04 -0.15 8.8e-01 -0.08 0.07 1.01 0.00  
##   
## Residual Standard Error = 0.88 with 537 degrees of freedom  
##   
## Multiple Regression  
## R R2 Ruw R2uw Shrunken R2 SE of R2 overall F df1 df2 p  
## cmrsptot 0.49 0.24 0.27 0.08 0.23 0.03 41.9 4 537 1.36e-30

Note: In psychology we would not include a 3-way interaction without also including all of the 2-way interactions underneath it.

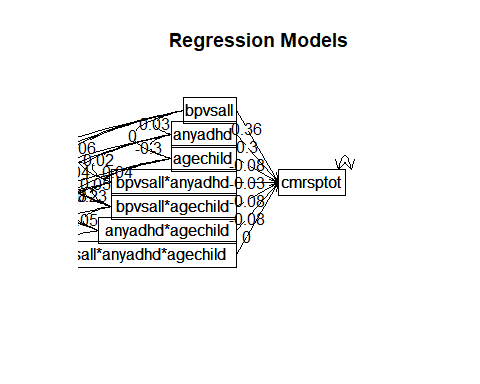
eay3way<-setCor(cmrsptot~(bpvsall:anyadhd:agechild), data=modelvars, std=TRUE)

plot(eay3way)  
eay3way

## Call: setCor(y = cmrsptot ~ (bpvsall:anyadhd:agechild), data = modelvars,   
## std = TRUE)  
##   
## Multiple Regression from raw data   
##   
## DV = cmrsptot   
## slope se t p lower.ci upper.ci VIF Vy.x  
## (Intercept) 0.00 0.04 0.00 1.0e+00 -0.07 0.07 1.00 0.00  
## bpvsall 0.36 0.04 9.24 5.9e-19 0.28 0.43 1.06 0.13  
## anyadhd 0.28 0.04 7.15 2.9e-12 0.20 0.36 1.10 0.09  
## agechild -0.09 0.04 -2.21 2.8e-02 -0.17 -0.01 1.10 0.01  
## bpvsall\*anyadhd\*agechild -0.01 0.04 -0.15 8.8e-01 -0.08 0.07 1.06 0.00  
##   
## Residual Standard Error = 0.88 with 537 degrees of freedom  
##   
## Multiple Regression  
## R R2 Ruw R2uw Shrunken R2 SE of R2 overall F df1 df2 p  
## cmrsptot 0.49 0.24 0.24 0.06 0.23 0.03 41.9 4 537 1.36e-30

This would be the right way to run a model with a 3-way and all the 2-way interactions. Interaction models get complicated to interpret. I wouldn’t recommend pushing it all the way to a three way for class. But I wanted to give you an example so you could see what it would look like.

threeway<-setCor(cmrsptot~(bpvsall\*anyadhd\*agechild), data=modelvars, std=TRUE)



plot(threeway)  
threeway

## Call: setCor(y = cmrsptot ~ (bpvsall \* anyadhd \* agechild), data = modelvars,   
## std = TRUE)  
##   
## Multiple Regression from raw data   
##   
## DV = cmrsptot   
## slope se t p lower.ci upper.ci VIF Vy.x  
## (Intercept) 0.00 0.04 0.00 1.0e+00 -0.07 0.07 1.00 0.00  
## bpvsall 0.36 0.04 9.40 1.6e-19 0.29 0.44 1.07 0.13  
## anyadhd 0.30 0.04 7.53 2.2e-13 0.22 0.38 1.13 0.10  
## agechild -0.08 0.04 -2.13 3.3e-02 -0.16 -0.01 1.11 0.01  
## bpvsall\*anyadhd -0.03 0.04 -0.65 5.2e-01 -0.10 0.05 1.09 0.00  
## bpvsall\*agechild -0.08 0.04 -2.08 3.8e-02 -0.16 0.00 1.07 0.01  
## anyadhd\*agechild -0.08 0.04 -2.03 4.3e-02 -0.15 0.00 1.04 0.00  
## bpvsall\*anyadhd\*agechild 0.00 0.04 0.03 9.7e-01 -0.08 0.08 1.09 0.00  
##   
## Residual Standard Error = 0.87 with 534 degrees of freedom  
##   
## Multiple Regression  
## R R2 Ruw R2uw Shrunken R2 SE of R2 overall F df1 df2 p  
## cmrsptot 0.5 0.25 0.14 0.02 0.24 0.03 25.5 7 534 4.53e-30

summary(threeway)

##   
## Multiple Regression from raw data   
## setCor(y = cmrsptot ~ (bpvsall \* anyadhd \* agechild), data = modelvars,   
## std = TRUE)  
##   
## Multiple Regression from matrix input   
##   
## Beta weights   
## cmrsptot  
## (Intercept) 0.0000  
## bpvsall 0.3637  
## anyadhd 0.3003  
## agechild -0.0841  
## bpvsall\*anyadhd -0.0253  
## bpvsall\*agechild -0.0803  
## anyadhd\*agechild -0.0776  
## bpvsall\*anyadhd\*agechild 0.0013  
##   
## Multiple R   
## cmrsptot   
## 0.5   
##   
## Multiple R2   
## cmrsptot   
## 0.25   
##   
## Cohen's set correlation R2   
## [1] 0.25  
##   
## Squared Canonical Correlations  
## NULL

anova(basic, threeway) #show how to compare simple and augmented models

## Model 1 = setCor(y = cmrsptot ~ bpvsall + anyadhd + agechild, data = modelvars,   
## std = TRUE)  
## Model 2 = setCor(y = cmrsptot ~ (bpvsall \* anyadhd \* agechild), data = modelvars,   
## std = TRUE)

## $cmrsptot  
## Res Df Res SS Diff df Diff SS F Pr(F > )  
## 1 538 412 NA NA NA NA  
## 2 534 406 4 6.73 2.21 0.0663

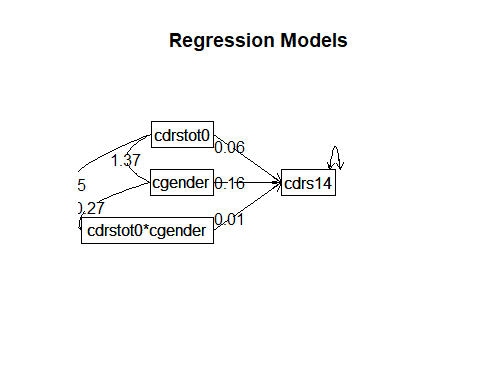
The four interaction terms (including a 3-way interaction!)

Differential item functioning (DIF) is an example of a conceptual question that interaction effects can help test.

abacab$cdrstot0<-abacab$cdrstot-17  
describe(abacab$cdrstot0)

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 773 14.4 13.3 10 12.7 11.9 0 67 67 0.97 0.23 0.48

casiVera<-setCor(cdrs14~cdrstot0:cgender, data=abacab, std=FALSE)



plot(casiVera)  
summary(casiVera)

##   
## Multiple Regression from raw data   
## setCor(y = cdrs14 ~ cdrstot0:cgender, data = abacab, std = FALSE)  
##   
## Multiple Regression from matrix input   
##   
## Beta weights   
## cdrs14  
## (Intercept) 1.9922  
## cdrstot0 0.0625  
## cgender 0.1630  
## cdrstot0\*cgender 0.0057  
##   
## Multiple R   
## cdrs14   
## 0.61   
##   
## Multiple R2   
## cdrs14   
## 0.37   
##   
## Cohen's set correlation R2   
## [1] 0.37  
##   
## Squared Canonical Correlations  
## NULL

casiVera

## Call: setCor(y = cdrs14 ~ cdrstot0:cgender, data = abacab, std = FALSE)  
##   
## Multiple Regression from raw data   
##   
## DV = cdrs14   
## slope se t p lower.ci upper.ci VIF Vy.x  
## (Intercept) 1.99 0.04 50.06 4.1e-252 1.91 2.07 1.00 0.00  
## cdrstot0 0.06 0.00 20.39 3.7e-75 0.06 0.07 1.09 0.36  
## cgender 0.16 0.08 2.01 4.5e-02 0.00 0.32 1.05 0.01  
## cdrstot0\*cgender 0.01 0.01 0.96 3.4e-01 -0.01 0.02 1.05 0.00  
##   
## Residual Standard Error = 1.12 with 824 degrees of freedom  
##   
## Multiple Regression  
## R R2 Ruw R2uw Shrunken R2 SE of R2 overall F df1 df2 p  
## cdrs14 0.61 0.37 0.47 0.22 0.37 0.03 162 3 824 1.25e-82

## Plotting interaction effects

To make a “presentation” version, what we want to do next to show the interaction is borrow Lia Follet’s ggplot2 code from her poster. :-)

#with gender as a moderator ggMarginal codes for the marginal histograms  
DIFGraph <-ggplot(abacab, aes(x= cdrstot, y=cdrs14, colour=as.factor(cgender)))+geom\_point() + scale\_color\_manual(values=c("#99BADD", "purple")) + geom\_smooth(method="lm", se=FALSE) + xlab("CDRS-R Total Score")+ ylab("CDRS Item 14 - Cries a lot") +geom\_jitter()+ scale\_fill\_discrete(name = "Sex", labels = c("Male", "Female", "Missing"))  
   
ggMarginal(DIFGraph, groupColour=TRUE, groupFill = TRUE)

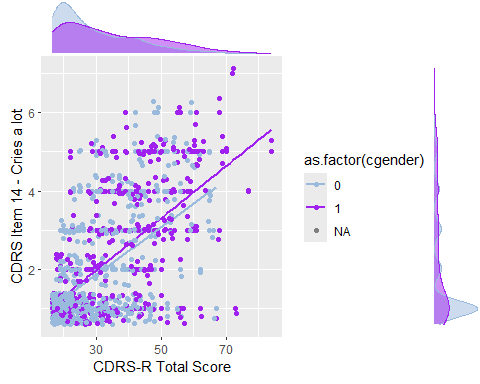
## `geom\_smooth()` using formula = 'y ~ x'

## Warning: Removed 55 rows containing non-finite outside the scale range  
## (`stat\_smooth()`).

## `geom\_smooth()` using formula = 'y ~ x'

## Warning: Removed 55 rows containing non-finite outside the scale range  
## (`stat\_smooth()`).

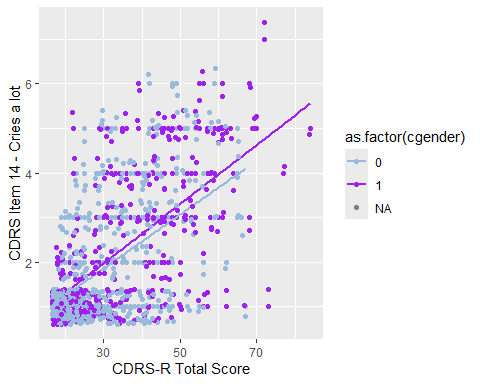
## Warning: Removed 55 rows containing missing values or values outside the scale range  
## (`geom\_point()`).  
## Removed 55 rows containing missing values or values outside the scale range  
## (`geom\_point()`).



DIFGraph

## `geom\_smooth()` using formula = 'y ~ x'

## Warning: Removed 55 rows containing non-finite outside the scale range  
## (`stat\_smooth()`).  
## Removed 55 rows containing missing values or values outside the scale range  
## (`geom\_point()`).  
## Removed 55 rows containing missing values or values outside the scale range  
## (`geom\_point()`).



# Regression analyses using psych package

The command switches to mediate, and the mediator is in (parentheses) Documentation: <https://cran.r-project.org/web/packages/psychTools/vignettes/mediation.pdf> It can handle simple mediation, multiple mediators, serial mediation, and moderated mediation!

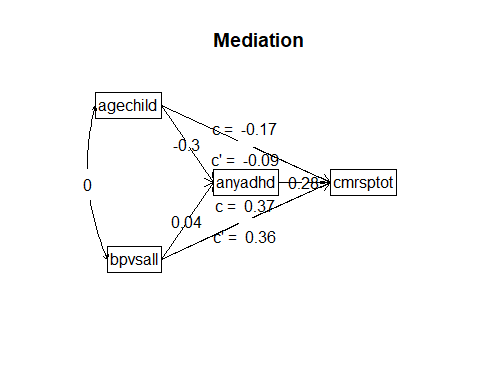
Revelle acknowledges that the Hayes macros have been bundled into the rProcess package, which would be another way of doing these.

## Mediation – silly example This is an example of modeling mediation. It’s a silly example because

the model does not actually make any sense conceptually – it flunks the “sniff test” just using common sense. But it’s showing two things: (a) how to use syntax to rearrange the variables to run a mediational model (b) why it is important to understand the variables and the model, because R will happily do stuff that makes no sense if you ask it!

In Revelle’s notation (author of psych package) c = “total effect” <– effect of predictor without including the mediator c’ = “direct effect” <– leftover direct effect that is not mediated.

mediationex<-mediate(cmrsptot~(anyadhd)+agechild+bpvsall, data=modelvars, std=TRUE)



plot(mediationex)  
summary(mediationex)

## Call: mediate(y = cmrsptot ~ (anyadhd) + agechild + bpvsall, data = modelvars,   
## std = TRUE)  
##   
## Direct effect estimates (traditional regression) (c') X + M on Y   
## cmrsptot se t df Prob  
## Intercept 0.00 0.04 0.00 538 1.00e+00  
## agechild -0.09 0.04 -2.22 538 2.68e-02  
## bpvsall 0.36 0.04 9.52 538 5.60e-20  
## anyadhd 0.28 0.04 7.15 538 2.76e-12  
##   
## R = 0.49 R2 = 0.24 F = 56 on 3 and 538 DF p-value: 1.68e-31   
##   
## Total effect estimates (c) (X on Y)   
## cmrsptot se t df Prob  
## Intercept 0.00 0.04 0.00 539 1.00e+00  
## agechild -0.17 0.04 -4.38 539 1.41e-05  
## bpvsall 0.37 0.04 9.38 539 1.88e-19  
##   
## 'a' effect estimates (X on M)   
## anyadhd se t df Prob  
## Intercept 0.00 0.04 0.00 539 1.00e+00  
## agechild -0.30 0.04 -7.32 539 9.20e-13  
## bpvsall 0.04 0.04 0.88 539 3.79e-01  
##   
## 'b' effect estimates (M on Y controlling for X)   
## cmrsptot se t df Prob  
## anyadhd 0.28 0.04 7.15 538 2.76e-12  
##   
## 'ab' effect estimates (through all mediators)  
## cmrsptot boot sd lower upper  
## agechild -0.08 -0.08 0.01 -0.12 -0.06  
## bpvsall 0.01 0.01 0.01 -0.12 -0.06

mediationex

##   
## Mediation/Moderation Analysis   
## Call: mediate(y = cmrsptot ~ (anyadhd) + agechild + bpvsall, data = modelvars,   
## std = TRUE)  
##   
## The DV (Y) was cmrsptot . The IV (X) was agechild bpvsall . The mediating variable(s) = anyadhd .  
##   
## Total effect(c) of agechild on cmrsptot = -0.17 S.E. = 0.04 t = -4.38 df= 539 with p = 1.4e-05  
## Direct effect (c') of agechild on cmrsptot removing anyadhd = -0.09 S.E. = 0.04 t = -2.22 df= 538 with p = 0.027  
## Indirect effect (ab) of agechild on cmrsptot through anyadhd = -0.08   
## Mean bootstrapped indirect effect = -0.08 with standard error = 0.01 Lower CI = -0.12 Upper CI = -0.06  
##   
## Total effect(c) of bpvsall on cmrsptot = 0.37 S.E. = 0.04 t = 9.38 df= 539 with p = 1.9e-19  
## Direct effect (c') of bpvsall on cmrsptot removing anyadhd = 0.36 S.E. = 0.04 t = 9.52 df= 538 with p = 5.6e-20  
## Indirect effect (ab) of bpvsall on cmrsptot through anyadhd = 0.01   
## Mean bootstrapped indirect effect = 0.01 with standard error = 0.01 Lower CI = -0.01 Upper CI = 0.03  
## R = 0.49 R2 = 0.24 F = 56 on 3 and 538 DF p-value: 1.81e-39   
##   
## To see the longer output, specify short = FALSE in the print statement or ask for the summary

#Now showing how to do stuff with t-test, ANOVA using rstatix.

<https://www.rdocumentation.org/packages/rstatix/versions/0.7.0>

## Still To Do: Block entry regression in R, with lm or psych

# Oneway ANOVA using rstatix

<https://rdrr.io/cran/rstatix/f/README.md>

onewayResults<- aov((cmrsptot ~ as.factor(cgroup5)), data = modelvars)  
onewayResults

## Call:  
## aov(formula = (cmrsptot ~ as.factor(cgroup5)), data = modelvars)  
##   
## Terms:  
## as.factor(cgroup5) Residuals  
## Sum of Squares 10768 50007  
## Deg. of Freedom 4 537  
##   
## Residual standard error: 9.65  
## Estimated effects may be unbalanced

anova\_summary(onewayResults, effect.size="pes", detailed = TRUE)

## Effect DFn DFd SSn SSd F p p<.05 pes  
## 1 1 4 537 10768 50007 28.9 8.85e-22 1 0.177

# Mistakes to avoid

doh<- aov((cmrsptot ~ cgroup5), data = modelvars)  
doh

## Call:  
## aov(formula = (cmrsptot ~ cgroup5), data = modelvars)  
##   
## Terms:  
## cgroup5 Residuals  
## Sum of Squares 8492 52283  
## Deg. of Freedom 1 540  
##   
## Residual standard error: 9.84  
## Estimated effects may be unbalanced

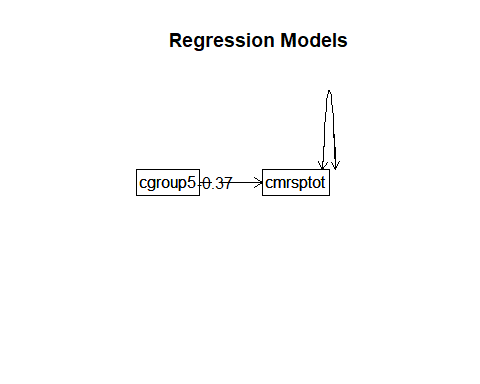
anova\_summary(doh, effect.size="pes", detailed = TRUE)

## Effect DFn DFd SSn SSd F p p<.05 pes  
## 1 1 1 540 8492 52283 87.7 2.04e-19 1 0.14

The “tell” is that there is only 1 df for the variable, when it should be (# of categories)-1

Here’s what it would look like in regression treating cgroup5 as (1) a continuous variable <– WRONG!!! (2) a factor <– correct!!!! (3) picking your comparison group <– fancy!!!!

sadlyWrong2<- setCor(cmrsptot ~ cgroup5, data = modelvars)



sadlyWrong2

## Call: setCor(y = cmrsptot ~ cgroup5, data = modelvars)  
##   
## Multiple Regression from raw data   
##   
## DV = cmrsptot   
## slope se t p lower.ci upper.ci VIF Vy.x  
## (Intercept) 0.00 0.04 0.00 1e+00 -0.08 0.08 1 0.00  
## cgroup5 -0.37 0.04 -9.37 2e-19 -0.45 -0.30 1 0.14  
##   
## Residual Standard Error = 0.93 with 540 degrees of freedom  
##   
## Multiple Regression  
## R R2 Ruw R2uw Shrunken R2 SE of R2 overall F df1 df2 p  
## cmrsptot 0.37 0.14 -0.37 0.14 0.14 0.03 87.7 1 540 2.04e-19

summary(sadlyWrong2, detailed = TRUE)

##   
## Multiple Regression from raw data   
## setCor(y = cmrsptot ~ cgroup5, data = modelvars)  
##   
## Multiple Regression from matrix input   
##   
## Beta weights   
## cmrsptot  
## (Intercept) 0.00  
## cgroup5 -0.37  
##   
## Multiple R   
## cmrsptot   
## 0.37   
##   
## Multiple R2   
## cmrsptot   
## 0.14   
##   
## Cohen's set correlation R2   
## [1] 0.14  
##   
## Squared Canonical Correlations  
## NULL

## a fix: declaring a categorical variable as a factor

modelvars$cgroup5<- as.factor(modelvars$cgroup5)  
rockOn<- lm(cmrsptot ~ cgroup5, data = modelvars)  
rockOn

##   
## Call:  
## lm(formula = cmrsptot ~ cgroup5, data = modelvars)  
##   
## Coefficients:  
## (Intercept) cgroup52 cgroup53 cgroup54 cgroup55   
## 24.44 -1.13 -10.02 -10.06 -18.18

summary(rockOn, detailed = TRUE)

##   
## Call:  
## lm(formula = cmrsptot ~ cgroup5, data = modelvars)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -20.30 -6.42 -2.30 5.68 31.70   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 24.44 2.41 10.13 < 2e-16 \*\*\*  
## cgroup52 -1.13 2.62 -0.43 0.67   
## cgroup53 -10.02 2.53 -3.96 8.6e-05 \*\*\*  
## cgroup54 -10.06 2.49 -4.04 6.2e-05 \*\*\*  
## cgroup55 -18.18 2.84 -6.41 3.2e-10 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 9.65 on 537 degrees of freedom  
## Multiple R-squared: 0.177, Adjusted R-squared: 0.171   
## F-statistic: 28.9 on 4 and 537 DF, p-value: <2e-16

## Changing the reference level for your dummy codes

R will automatically create dummy codes for your factor. It’s possible to pick which group gets assigned “0” for all the comparisons. This makes it the reference group, or the one that all other groups get compared against.

fancy<- lm(cmrsptot ~ relevel(cgroup5, ref="5"), data = modelvars)  
fancy

##   
## Call:  
## lm(formula = cmrsptot ~ relevel(cgroup5, ref = "5"), data = modelvars)  
##   
## Coefficients:  
## (Intercept) relevel(cgroup5, ref = "5")1   
## 6.26 18.18   
## relevel(cgroup5, ref = "5")2 relevel(cgroup5, ref = "5")3   
## 17.04 8.16   
## relevel(cgroup5, ref = "5")4   
## 8.11

summary(fancy, detailed = TRUE)

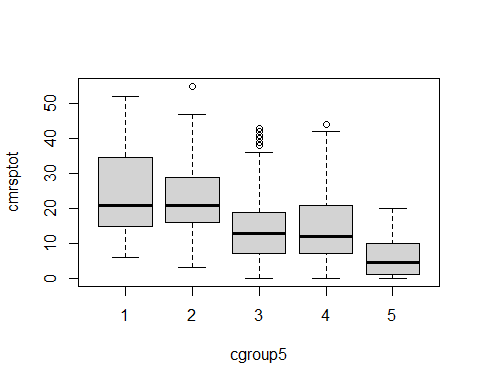
##   
## Call:  
## lm(formula = cmrsptot ~ relevel(cgroup5, ref = "5"), data = modelvars)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -20.30 -6.42 -2.30 5.68 31.70   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 6.26 1.49 4.21 3.1e-05 \*\*\*  
## relevel(cgroup5, ref = "5")1 18.18 2.84 6.41 3.2e-10 \*\*\*  
## relevel(cgroup5, ref = "5")2 17.04 1.81 9.43 < 2e-16 \*\*\*  
## relevel(cgroup5, ref = "5")3 8.16 1.68 4.87 1.5e-06 \*\*\*  
## relevel(cgroup5, ref = "5")4 8.11 1.62 5.02 7.0e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 9.65 on 537 degrees of freedom  
## Multiple R-squared: 0.177, Adjusted R-squared: 0.171   
## F-statistic: 28.9 on 4 and 537 DF, p-value: <2e-16

## Boxplots (instead of bar charts)

…to figure out what the effect is If we decide that this is one of the main figures in the presentation, then we can give it some “TLC” in terms of axis labels, as well as general aesthetics. This is plenty helpful for a quick plot to get oriented, though!

# Group differences

boxplot(cmrsptot~cgroup5, data=modelvars)



Different post hoc tests compare the group means, while also trying to protect against “false positive” results (Type I errors).

Pairwise t-tests with a correction is one common approach.

rstatix can do all of these with a t-test: “holm” - Holm’s (1979) stepdown correction “hochberg” - Hochberg’s (1988) correction ” hommel” -Hommel’s (1988) correction “bonferroni” - Bonferroni correction –common, but it’s the worst for statistical power “BH”-Benjamini-Hochberg (1995) correction, also know as false discovery rate (FDR) “BY”- Benjamini-Yekutieli (2001) “fdr” - alias for Benjamini-Hochberg, above; commonly used in genetics and imaging work “none” - no error protection!

EY recommendations: FDR or Holm if you want to balance errors. Bonferroni if you want to be conservative or make it hard for groups to be significantly different.

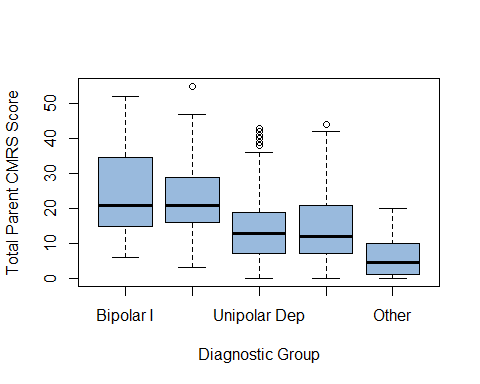
pairwise.t.test(modelvars$cmrsptot, modelvars$cgroup5,  
 p.adjust.method = "BH") #Benjamini-Hochberg False Discovery Rate correction

##   
## Pairwise comparisons using t tests with pooled SD   
##   
## data: modelvars$cmrsptot and modelvars$cgroup5   
##   
## 1 2 3 4   
## 2 0.7 - - -   
## 3 1e-04 4e-11 - -   
## 4 9e-05 2e-12 1.0 -   
## 5 8e-10 <2e-16 2e-06 1e-06  
##   
## P value adjustment method: BH

modelvars %>% games\_howell\_test(cmrsptot ~ cgroup5)

## # A tibble: 10 × 8  
## .y. group1 group2 estimate conf.low conf.high p.adj p.adj.signif  
## \* <chr> <chr> <chr> <dbl> <dbl> <dbl> <dbl> <chr>   
## 1 cmrsptot 1 2 -1.13 -11.7 9.48 9.97e- 1 ns   
## 2 cmrsptot 1 3 -10.0 -20.5 0.440 6.4 e- 2 ns   
## 3 cmrsptot 1 4 -10.1 -20.5 0.344 6.1 e- 2 ns   
## 4 cmrsptot 1 5 -18.2 -28.7 -7.64 5.22e- 4 \*\*\*   
## 5 cmrsptot 2 3 -8.89 -12.6 -5.17 4.84e- 9 \*\*\*\*   
## 6 cmrsptot 2 4 -8.93 -12.4 -5.44 6.13e-10 \*\*\*\*   
## 7 cmrsptot 2 5 -17.0 -21.1 -13.0 3.61e-14 \*\*\*\*   
## 8 cmrsptot 3 4 -0.0443 -2.76 2.67 1 e+ 0 ns   
## 9 cmrsptot 3 5 -8.16 -11.5 -4.78 1.13e- 8 \*\*\*\*   
## 10 cmrsptot 4 5 -8.11 -11.3 -4.97 2.71e- 9 \*\*\*\*

boxplot(cmrsptot ~ cgroup5, data = modelvars,  
 names = c("Bipolar I","Other bipolar","Unipolar Dep","ADHD/DBD","Other"), col="#99BADD", ylab = "Total Parent CMRS Score", xlab = "Diagnostic Group")



# Grouped data  
modelvars %>%  
 group\_by(anyadhd) %>%  
 games\_howell\_test(cmrsptot ~ bpvsall)

## # A tibble: 2 × 9  
## anyadhd .y. group1 group2 estimate conf.low conf.high p.adj p.adj.signif  
## \* <dbl+lb> <chr> <chr> <chr> <dbl> <dbl> <dbl> <dbl> <chr>   
## 1 0 [No c… cmrs… 0 1 9.92 6.24 13.6 2.78e- 6 \*\*\*\*   
## 2 1 [ADHD] cmrs… 0 1 9.41 6.64 12.2 9.23e-10 \*\*\*\*

<https://rpubs.com/pinkrpub/667245> for EDA visualization examples

# Chi Squared!

This page was helpful: <http://www.sthda.com/english/wiki/chi-square-test-of-independence-in-r> also <https://statsandr.com/blog/chi-square-test-of-independence-in-r/#chi-square-test-of-independence-in-r>

and Andy Fields has the helpful tip about requesting output as “SPSS” format. On p. 821 f. he recommends and walks through the gmodels package, which provides helpful output.

<http://www.sthda.com/english/articles/32-r-graphics-essentials/129-visualizing-multivariate-categorical-data/>

library (vcd)  
library (graphics)  
library (gplots)

##   
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':  
##   
## lowess

library (gmodels)

## Registered S3 method overwritten by 'gdata':  
## method from   
## reorder.factor gplots

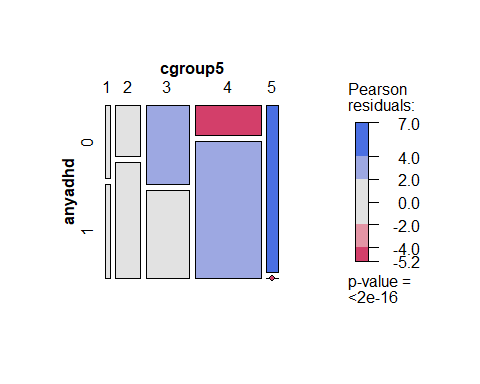
chisqResults<-chisq.test(table(modelvars$cgroup5, modelvars$anyadhd))  
chisqResults #covers the basics, but not detailed

##   
## Pearson's Chi-squared test  
##   
## data: table(modelvars$cgroup5, modelvars$anyadhd)  
## X-squared = 118, df = 4, p-value <2e-16

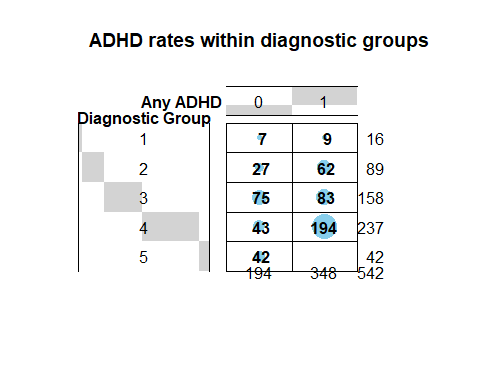
gmodels::CrossTable(modelvars$anyadhd, modelvars$cgroup5, chisq = TRUE, expected = TRUE,   
 sresid=TRUE, format = "SPSS")

##   
## Cell Contents  
## |-------------------------|  
## | Count |  
## | Expected Values |  
## | Chi-square contribution |  
## | Row Percent |  
## | Column Percent |  
## | Total Percent |  
## | Std Residual |  
## |-------------------------|  
##   
## Total Observations in Table: 542   
##   
## | modelvars$cgroup5   
## modelvars$anyadhd | 1 | 2 | 3 | 4 | 5 | Row Total |   
## ------------------|-----------|-----------|-----------|-----------|-----------|-----------|  
## 0 | 7 | 27 | 75 | 43 | 42 | 194 |   
## | 5.727 | 31.856 | 56.554 | 84.830 | 15.033 | |   
## | 0.283 | 0.740 | 6.017 | 20.627 | 48.373 | |   
## | 3.608% | 13.918% | 38.660% | 22.165% | 21.649% | 35.793% |   
## | 43.750% | 30.337% | 47.468% | 18.143% | 100.000% | |   
## | 1.292% | 4.982% | 13.838% | 7.934% | 7.749% | |   
## | 0.532 | -0.860 | 2.453 | -4.542 | 6.955 | |   
## ------------------|-----------|-----------|-----------|-----------|-----------|-----------|  
## 1 | 9 | 62 | 83 | 194 | 0 | 348 |   
## | 10.273 | 57.144 | 101.446 | 152.170 | 26.967 | |   
## | 0.158 | 0.413 | 3.354 | 11.499 | 26.967 | |   
## | 2.586% | 17.816% | 23.851% | 55.747% | 0.000% | 64.207% |   
## | 56.250% | 69.663% | 52.532% | 81.857% | 0.000% | |   
## | 1.661% | 11.439% | 15.314% | 35.793% | 0.000% | |   
## | -0.397 | 0.642 | -1.831 | 3.391 | -5.193 | |   
## ------------------|-----------|-----------|-----------|-----------|-----------|-----------|  
## Column Total | 16 | 89 | 158 | 237 | 42 | 542 |   
## | 2.952% | 16.421% | 29.151% | 43.727% | 7.749% | |   
## ------------------|-----------|-----------|-----------|-----------|-----------|-----------|  
##   
##   
## Statistics for All Table Factors  
##   
##   
## Pearson's Chi-squared test   
## ------------------------------------------------------------  
## Chi^2 = 118 d.f. = 4 p = 1.16e-24   
##   
##   
##   
## Minimum expected frequency: 5.73

vcd::mosaic(~cgroup5 + anyadhd, direction=c("v","h"), data=modelvars, shade=TRUE)



tableInput<-table(modelvars$cgroup5, modelvars$anyadhd )  
gplots::balloonplot(t(tableInput),   
 main="ADHD rates within diagnostic groups", xlab="Any ADHD", ylab="Diagnostic Group")



<https://rdrr.io/cran/rstatix/man/games_howell_test.html>

# Sufficient data for replication

There are two values in tension with research: wanting to have control of the data versus wanting to be transparent and replicable.

The ideas of sharing and reproducibility are core to Open Science. They also are newer ideas, so how to do it is evolving rapidly.

A “sweet spot” that tries to balance the two is to make a shareable version of the data that contains everything one would need to replicate… and not a single row or column more. That checks the boxes for reproducibility and transparency, and it lets you keep control of the rest of the data (and not “overshare”).

The easiest way to do it is to take the object that has the variables that made the “final cut” for this particular project, and then write out a copy of it (not the whole original dataset).

Then share that smaller file, but not the other one.

The simples way to do that is via the “write.csv” command. The default options spit out a comma separated value (= .csv) text file that can be read by anything that can read text, with the variable names (and nothing else) at the top. No missing value codes, no variable or value labels, none of the other goodies come through. But the variable names and the data are enough to check your results.

There are ways of adding back more information (there’s a write.sav command, and other ways of writing to SPSS, SAS, Excel, or other formats). Writing the sufficient data as a .sav would keep a lot more information, but it would also require the next person to use haven (or foreign, or expss… ) to read it (or fork over a lot of money to IBM, Microsoft, or whatever big tech firm is extracting the cash!).

A vanilla .csv is enough to get full credit for the class. A .sav file would be coming “full circle” –> we started with the data in an .sav file (curated by someone else), added value with a new question, research, analyses, and visualization, and the make it reproducible… and pay it forward with another labelled .sav file… completing “the Circle of Science”! (<– cue The Lion King soundtrack!)

write.csv(modelvars, "YourNameSufficientData.csv")

#Advanced stuff

Extra bling: could get Cronbach’s alpha from own data!

Item names are cmrsp01 to cmrsp21.

##need to input items  
cmrsitems<-select(abacab, cmrsp01:cmrsp21) # make a smaller object with the variables we'll use for the examples  
  
describe(cmrsitems) #note the different Ns due to missing data

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## cmrsp01 1 544 0.72 0.83 1.0 0.59 1.48 0 3 3 1.08 0.65 0.04  
## cmrsp02 2 543 1.14 0.99 1.0 1.05 1.48 0 3 3 0.47 -0.85 0.04  
## cmrsp03 3 545 0.98 1.06 1.0 0.86 1.48 0 3 3 0.69 -0.82 0.05  
## cmrsp04 4 544 0.26 0.64 0.0 0.09 0.00 0 3 3 2.80 7.56 0.03  
## cmrsp05 5 544 0.50 0.83 0.0 0.31 0.00 0 3 3 1.69 2.06 0.04  
## cmrsp06 6 542 1.10 1.12 1.0 1.00 1.48 0 3 3 0.57 -1.07 0.05  
## cmrsp07 7 544 1.29 1.10 1.0 1.24 1.48 0 3 3 0.29 -1.24 0.05  
## cmrsp08 8 543 0.76 0.96 0.0 0.60 0.00 0 3 3 1.04 -0.04 0.04  
## cmrsp09 9 544 0.79 0.96 0.5 0.64 0.74 0 3 3 0.98 -0.12 0.04  
## cmrsp10 10 547 0.79 0.98 0.0 0.62 0.00 0 3 3 1.01 -0.14 0.04  
## cmrsp11 11 543 1.49 1.06 1.0 1.49 1.48 0 3 3 0.04 -1.23 0.05  
## cmrsp12 12 544 0.71 0.86 0.0 0.58 0.00 0 3 3 1.06 0.37 0.04  
## cmrsp13 13 543 0.23 0.63 0.0 0.06 0.00 0 3 3 3.12 9.58 0.03  
## cmrsp14 14 544 0.28 0.61 0.0 0.14 0.00 0 3 3 2.31 5.13 0.03  
## cmrsp15 15 543 0.47 0.75 0.0 0.30 0.00 0 3 3 1.62 2.04 0.03  
## cmrsp16 16 544 1.14 1.05 1.0 1.05 1.48 0 3 3 0.50 -0.96 0.05  
## cmrsp17 17 545 0.90 0.94 1.0 0.78 1.48 0 3 3 0.77 -0.37 0.04  
## cmrsp18 18 544 1.15 1.08 1.0 1.07 1.48 0 3 3 0.50 -1.05 0.05  
## cmrsp19 19 541 0.40 0.71 0.0 0.24 0.00 0 3 3 1.88 3.11 0.03  
## cmrsp20 20 541 0.25 0.61 0.0 0.09 0.00 0 3 3 2.85 8.27 0.03  
## cmrsp21 21 543 0.20 0.54 0.0 0.06 0.00 0 3 3 3.23 11.30 0.02

cmrsitemsShrunk<-na.omit(cmrsitems) #listwise delete for complete items  
describe(cmrsitemsShrunk)

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## cmrsp01 1 522 0.72 0.83 1 0.59 1.48 0 3 3 1.11 0.72 0.04  
## cmrsp02 2 522 1.15 1.00 1 1.06 1.48 0 3 3 0.46 -0.86 0.04  
## cmrsp03 3 522 0.98 1.06 1 0.84 1.48 0 3 3 0.72 -0.78 0.05  
## cmrsp04 4 522 0.25 0.64 0 0.09 0.00 0 3 3 2.79 7.59 0.03  
## cmrsp05 5 522 0.49 0.83 0 0.30 0.00 0 3 3 1.73 2.18 0.04  
## cmrsp06 6 522 1.08 1.11 1 0.98 1.48 0 3 3 0.60 -1.02 0.05  
## cmrsp07 7 522 1.28 1.10 1 1.22 1.48 0 3 3 0.31 -1.22 0.05  
## cmrsp08 8 522 0.75 0.95 0 0.59 0.00 0 3 3 1.06 0.02 0.04  
## cmrsp09 9 522 0.79 0.95 1 0.64 1.48 0 3 3 0.98 -0.08 0.04  
## cmrsp10 10 522 0.79 0.98 0 0.62 0.00 0 3 3 1.02 -0.10 0.04  
## cmrsp11 11 522 1.49 1.06 1 1.49 1.48 0 3 3 0.04 -1.23 0.05  
## cmrsp12 12 522 0.71 0.86 0 0.58 0.00 0 3 3 1.07 0.37 0.04  
## cmrsp13 13 522 0.23 0.64 0 0.06 0.00 0 3 3 3.11 9.46 0.03  
## cmrsp14 14 522 0.28 0.61 0 0.14 0.00 0 3 3 2.34 5.29 0.03  
## cmrsp15 15 522 0.47 0.75 0 0.30 0.00 0 3 3 1.60 1.99 0.03  
## cmrsp16 16 522 1.13 1.05 1 1.04 1.48 0 3 3 0.51 -0.97 0.05  
## cmrsp17 17 522 0.89 0.93 1 0.77 1.48 0 3 3 0.78 -0.35 0.04  
## cmrsp18 18 522 1.16 1.09 1 1.08 1.48 0 3 3 0.49 -1.08 0.05  
## cmrsp19 19 522 0.40 0.71 0 0.24 0.00 0 3 3 1.87 3.06 0.03  
## cmrsp20 20 522 0.25 0.61 0 0.09 0.00 0 3 3 2.83 8.19 0.03  
## cmrsp21 21 522 0.20 0.54 0 0.06 0.00 0 3 3 3.17 10.85 0.02

headTail(cmrsitemsShrunk)

## cmrsp01 cmrsp02 cmrsp03 cmrsp04 cmrsp05 cmrsp06 cmrsp07 cmrsp08 cmrsp09  
## 1 1 1 1 1 1 1 1 1 1  
## 2 1 0 1 0 0 0 0 0 0  
## 3 1 2 0 0 1 3 3 1 1  
## 4 3 3 3 0 2 3 3 3 3  
## 5 ... ... ... ... ... ... ... ... ...  
## 6 0 0 0 0 1 3 2 1 1  
## 7 1 2 0 0 0 1 0 0 0  
## 8 0 0 0 0 0 0 2 1 1  
## 9 0 3 0 0 0 0 3 0 3  
## cmrsp10 cmrsp11 cmrsp12 cmrsp13 cmrsp14 cmrsp15 cmrsp16 cmrsp17 cmrsp18  
## 1 1 1 1 1 1 1 1 1 1  
## 2 0 1 1 0 0 0 0 0 1  
## 3 2 3 0 0 1 1 3 0 3  
## 4 3 3 3 1 2 0 3 2 3  
## 5 ... ... ... ... ... ... ... ... ...  
## 6 0 2 0 0 0 0 0 0 1  
## 7 1 1 1 1 0 0 1 0 1  
## 8 1 1 0 0 0 0 1 1 1  
## 9 0 3 0 0 0 0 3 0 3  
## cmrsp19 cmrsp20 cmrsp21  
## 1 1 0 1  
## 2 0 0 0  
## 3 0 0 0  
## 4 3 3 3  
## 5 ... ... ...  
## 6 0 1 0  
## 7 1 1 1  
## 8 0 0 0  
## 9 0 0 0

##need to input items  
bdsitems<-select(ku2017, bds01, bds02, bds04, bds05, bds06, bds08, bds10) # make a smaller object with the variables we'll use for the examples  
  
  
describe(bdsitems) #note the different Ns due to missing data

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## bds01 1 689 1.22 0.41 1 1.15 0 1 2 1 1.37 -0.14 0.02  
## bds02 2 539 1.35 0.48 1 1.31 0 1 2 1 0.62 -1.61 0.02  
## bds04 3 350 1.34 0.47 1 1.30 0 1 2 1 0.69 -1.53 0.03  
## bds05 4 350 1.39 0.49 1 1.36 0 1 2 1 0.44 -1.81 0.03  
## bds06 5 350 1.57 0.50 2 1.59 0 1 2 1 -0.30 -1.92 0.03  
## bds08 6 350 1.77 0.42 2 1.83 0 1 2 1 -1.25 -0.44 0.02  
## bds10 7 350 1.34 0.48 1 1.30 0 1 2 1 0.66 -1.57 0.03

bdsitemsShrunk<-na.omit(bdsitems) #listwise delete for complete items  
describe(bdsitemsShrunk)

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## bds01 1 350 1.00 0.00 1 1.00 0 1 1 0 NaN NaN 0.00  
## bds02 2 350 1.00 0.00 1 1.00 0 1 1 0 NaN NaN 0.00  
## bds04 3 350 1.34 0.47 1 1.30 0 1 2 1 0.69 -1.53 0.03  
## bds05 4 350 1.39 0.49 1 1.36 0 1 2 1 0.44 -1.81 0.03  
## bds06 5 350 1.57 0.50 2 1.59 0 1 2 1 -0.30 -1.92 0.03  
## bds08 6 350 1.77 0.42 2 1.83 0 1 2 1 -1.25 -0.44 0.02  
## bds10 7 350 1.34 0.48 1 1.30 0 1 2 1 0.66 -1.57 0.03

headTail(bdsitemsShrunk)

## bds01 bds02 bds04 bds05 bds06 bds08 bds10  
## 1 1 1 1 1 1 1 1  
## 2 1 1 1 2 2 2 2  
## 3 1 1 1 2 1 2 2  
## 4 1 1 1 2 2 2 1  
## 5 ... ... ... ... ... ... ...  
## 6 1 1 2 2 2 2 1  
## 7 1 1 2 2 2 2 1  
## 8 1 1 1 2 2 2 2  
## 9 1 1 1 1 1 2 2

# Reliability

## Alpha

Cronbach’s alpha, the most widely reported form of internal consistency reliability, is in the psych package

psych::alpha(bdsitemsShrunk)

## Warning in psych::alpha(bdsitemsShrunk): Item = bds01 had no variance and was  
## deleted but still is counted in the score

## Warning in psych::alpha(bdsitemsShrunk): Item = bds02 had no variance and was  
## deleted but still is counted in the score

##   
## Reliability analysis   
## Call: psych::alpha(x = bdsitemsShrunk)  
##   
## raw\_alpha std.alpha G6(smc) average\_r S/N ase mean sd median\_r  
## 0.66 0.66 0.65 0.28 2 0.028 1.3 0.22 0.27  
##   
## 95% confidence boundaries   
## lower alpha upper  
## Feldt 0.60 0.66 0.72  
## Duhachek 0.61 0.66 0.72  
##   
## Reliability if an item is dropped:  
## raw\_alpha std.alpha G6(smc) average\_r S/N alpha se var.r med.r  
## bds04 0.69 0.69 0.65 0.36 2.2 0.027 0.0158 0.33  
## bds05 0.55 0.56 0.53 0.24 1.3 0.039 0.0268 0.21  
## bds06 0.56 0.56 0.50 0.24 1.3 0.038 0.0077 0.24  
## bds08 0.60 0.60 0.55 0.27 1.5 0.035 0.0137 0.27  
## bds10 0.64 0.64 0.61 0.30 1.8 0.032 0.0328 0.31  
##   
## Item statistics   
## n raw.r std.r r.cor r.drop mean sd  
## bds04 350 0.52 0.52 0.30 0.24 1.3 0.47  
## bds05 350 0.74 0.73 0.64 0.53 1.4 0.49  
## bds06 350 0.73 0.73 0.67 0.51 1.6 0.50  
## bds08 350 0.66 0.67 0.58 0.45 1.8 0.42  
## bds10 350 0.62 0.61 0.44 0.36 1.3 0.48  
##   
## Non missing response frequency for each item  
## 1 2 miss  
## bds04 0.66 0.34 0  
## bds05 0.61 0.39 0  
## bds06 0.43 0.57 0  
## bds08 0.23 0.77 0  
## bds10 0.66 0.34 0

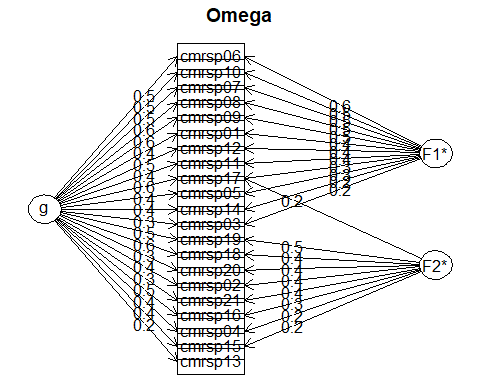
## Omega – an alternative reliability measure

Omega also is in the psych package.

psych::omega(cmrsitemsShrunk, nfactors=2)

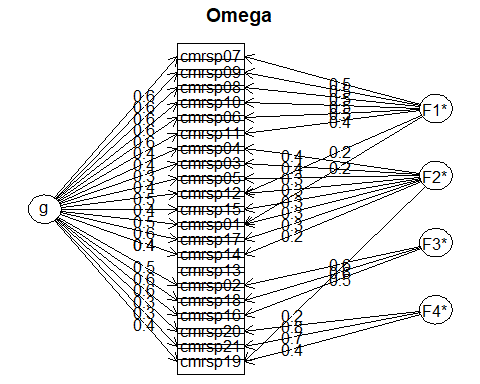
## Loading required namespace: GPArotation

##   
## Three factors are required for identification -- general factor loadings set to be equal.   
## Proceed with caution.   
## Think about redoing the analysis with alternative values of the 'option' setting.



## Omega   
## Call: omegah(m = m, nfactors = nfactors, fm = fm, key = key, flip = flip,   
## digits = digits, title = title, sl = sl, labels = labels,   
## plot = plot, n.obs = n.obs, rotate = rotate, Phi = Phi, option = option,   
## covar = covar)  
## Alpha: 0.89   
## G.6: 0.91   
## Omega Hierarchical: 0.62   
## Omega H asymptotic: 0.68   
## Omega Total 0.9   
##   
## Schmid Leiman Factor loadings greater than 0.2   
## g F1\* F2\* h2 h2 u2 p2 com  
## cmrsp01 0.43 0.38 0.33 0.33 0.67 0.56 1.98  
## cmrsp02 0.44 0.40 0.36 0.36 0.64 0.55 1.98  
## cmrsp03 0.29 0.21 0.14 0.86 0.63 1.95  
## cmrsp04 0.36 0.23 0.20 0.20 0.80 0.67 1.93  
## cmrsp05 0.40 0.25 0.24 0.24 0.76 0.66 1.93  
## cmrsp06 0.54 0.59 0.65 0.65 0.35 0.44 2.02  
## cmrsp07 0.54 0.52 0.56 0.56 0.44 0.52 2.00  
## cmrsp08 0.57 0.49 0.57 0.57 0.43 0.57 1.98  
## cmrsp09 0.57 0.49 0.56 0.56 0.44 0.58 1.98  
## cmrsp10 0.52 0.54 0.56 0.56 0.44 0.48 2.01  
## cmrsp11 0.41 0.35 0.29 0.29 0.71 0.57 1.98  
## cmrsp12 0.51 0.37 0.41 0.41 0.59 0.63 1.95  
## cmrsp13 0.20 0.07 0.93 0.57 1.98  
## cmrsp14 0.36 0.22 0.20 0.20 0.80 0.67 1.93  
## cmrsp15 0.40 0.21 0.23 0.23 0.77 0.68 1.92  
## cmrsp16 0.52 0.31 0.40 0.40 0.60 0.67 1.93  
## cmrsp17 0.55 0.28 0.25 0.45 0.45 0.55 0.69 1.92  
## cmrsp18 0.55 0.43 0.50 0.50 0.50 0.61 1.96  
## cmrsp19 0.46 0.47 0.43 0.43 0.57 0.49 2.01  
## cmrsp20 0.35 0.40 0.29 0.29 0.71 0.42 2.03  
## cmrsp21 0.30 0.38 0.24 0.24 0.76 0.37 2.04  
##   
## With Sums of squares of:  
## g F1\* F2\* h2   
## 4.3 2.1 1.2 3.3   
##   
## general/max 1.3 max/min = 2.74  
## mean percent general = 0.57 with sd = 0.09 and cv of 0.16   
## Explained Common Variance of the general factor = 0.56   
##   
## The degrees of freedom are 169 and the fit is 1.96   
## The number of observations was 522 with Chi Square = 1005 with prob < 6.8e-119  
## The root mean square of the residuals is 0.07   
## The df corrected root mean square of the residuals is 0.07  
## RMSEA index = 0.097 and the 10 % confidence intervals are 0.092 0.103  
## BIC = -52.5  
##   
## Compare this with the adequacy of just a general factor and no group factors  
## The degrees of freedom for just the general factor are 189 and the fit is 3.27   
## The number of observations was 522 with Chi Square = 1676 with prob < 3e-236  
## The root mean square of the residuals is 0.13   
## The df corrected root mean square of the residuals is 0.14   
##   
## RMSEA index = 0.123 and the 10 % confidence intervals are 0.117 0.128  
## BIC = 493   
##   
## Measures of factor score adequacy   
## g F1\* F2\*  
## Correlation of scores with factors 0.79 0.72 0.68  
## Multiple R square of scores with factors 0.63 0.52 0.46  
## Minimum correlation of factor score estimates 0.25 0.05 -0.07  
##   
## Total, General and Subset omega for each subset  
## g F1\* F2\*  
## Omega total for total scores and subscales 0.90 0.88 0.77  
## Omega general for total scores and subscales 0.62 0.52 0.45  
## Omega group for total scores and subscales 0.22 0.35 0.32

psych::omega(cmrsitemsShrunk, nfactors=4)



## Omega   
## Call: omegah(m = m, nfactors = nfactors, fm = fm, key = key, flip = flip,   
## digits = digits, title = title, sl = sl, labels = labels,   
## plot = plot, n.obs = n.obs, rotate = rotate, Phi = Phi, option = option,   
## covar = covar)  
## Alpha: 0.89   
## G.6: 0.91   
## Omega Hierarchical: 0.68   
## Omega H asymptotic: 0.73   
## Omega Total 0.92   
##   
## Schmid Leiman Factor loadings greater than 0.2   
## g F1\* F2\* F3\* F4\* h2 h2 u2 p2 com  
## cmrsp01 0.47 0.24 0.27 0.37 0.37 0.63 0.61 2.20  
## cmrsp02 0.46 0.60 0.59 0.59 0.41 0.37 1.90  
## cmrsp03 0.32 0.37 0.25 0.25 0.75 0.42 2.04  
## cmrsp04 0.38 0.45 0.35 0.35 0.65 0.42 2.12  
## cmrsp05 0.43 0.32 0.31 0.31 0.69 0.60 2.07  
## cmrsp06 0.59 0.49 0.62 0.62 0.38 0.56 2.11  
## cmrsp07 0.58 0.52 0.62 0.62 0.38 0.55 2.01  
## cmrsp08 0.61 0.51 0.64 0.64 0.36 0.58 2.01  
## cmrsp09 0.61 0.51 0.63 0.63 0.37 0.58 2.01  
## cmrsp10 0.57 0.50 0.58 0.58 0.42 0.55 2.00  
## cmrsp11 0.44 0.36 0.33 0.33 0.67 0.58 2.10  
## cmrsp12 0.55 0.23 0.30 0.45 0.45 0.55 0.66 1.97  
## cmrsp13 0.20 0.07 0.93 0.55 2.67  
## cmrsp14 0.38 0.20 0.22 0.22 0.78 0.66 2.06  
## cmrsp15 0.41 0.29 0.27 0.27 0.73 0.62 2.06  
## cmrsp16 0.55 0.55 0.61 0.61 0.39 0.50 2.09  
## cmrsp17 0.58 0.26 0.46 0.46 0.54 0.72 1.83  
## cmrsp18 0.57 0.59 0.68 0.68 0.32 0.48 2.03  
## cmrsp19 0.44 0.23 0.36 0.41 0.41 0.59 0.47 2.95  
## cmrsp20 0.32 0.76 0.67 0.67 0.33 0.15 1.37  
## cmrsp21 0.27 0.73 0.60 0.60 0.40 0.12 1.29  
##   
## With Sums of squares of:  
## g F1\* F2\* F3\* F4\* h2   
## 4.8 1.6 0.9 1.1 1.3 5.2   
##   
## general/max 0.93 max/min = 5.71  
## mean percent general = 0.51 with sd = 0.15 and cv of 0.3   
## Explained Common Variance of the general factor = 0.49   
##   
## The degrees of freedom are 132 and the fit is 0.78   
## The number of observations was 522 with Chi Square = 397 with prob < 2.8e-28  
## The root mean square of the residuals is 0.03   
## The df corrected root mean square of the residuals is 0.04  
## RMSEA index = 0.062 and the 10 % confidence intervals are 0.055 0.069  
## BIC = -429  
##   
## Compare this with the adequacy of just a general factor and no group factors  
## The degrees of freedom for just the general factor are 189 and the fit is 3.1   
## The number of observations was 522 with Chi Square = 1590 with prob < 9.3e-220  
## The root mean square of the residuals is 0.12   
## The df corrected root mean square of the residuals is 0.12   
##   
## RMSEA index = 0.119 and the 10 % confidence intervals are 0.114 0.125  
## BIC = 407   
##   
## Measures of factor score adequacy   
## g F1\* F2\* F3\* F4\*  
## Correlation of scores with factors 0.83 0.72 0.70 0.77 0.86  
## Multiple R square of scores with factors 0.69 0.52 0.49 0.59 0.75  
## Minimum correlation of factor score estimates 0.38 0.05 -0.03 0.19 0.49  
##   
## Total, General and Subset omega for each subset  
## g F1\* F2\* F3\* F4\*  
## Omega total for total scores and subscales 0.92 0.89 0.73 0.83 0.76  
## Omega general for total scores and subscales 0.68 0.51 0.49 0.38 0.18  
## Omega group for total scores and subscales 0.15 0.37 0.24 0.45 0.58

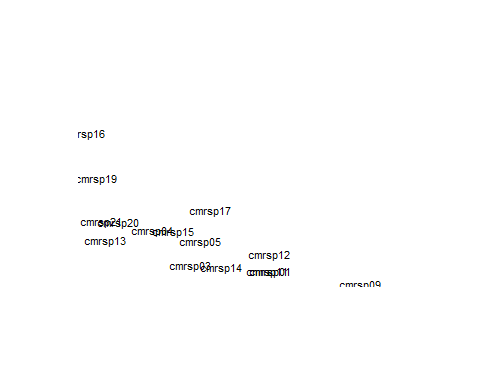
## Exploratory factor analysis

Factor analysis is an “easter egg” – not something required for a poster in 525.

# Maximum Likelihood Factor Analysis  
# entering raw data and extracting 2 factors,  
# with varimax rotation  
  
cmrsmlfit <- factanal(cmrsitemsShrunk, 2, rotation="promax")  
print(cmrsmlfit, digits=2, cutoff=.3, sort=TRUE)

##   
## Call:  
## factanal(x = cmrsitemsShrunk, factors = 2, rotation = "promax")  
##   
## Uniquenesses:  
## cmrsp01 cmrsp02 cmrsp03 cmrsp04 cmrsp05 cmrsp06 cmrsp07 cmrsp08 cmrsp09 cmrsp10   
## 0.69 0.43 0.88 0.84 0.77 0.40 0.42 0.38 0.40 0.42   
## cmrsp11 cmrsp12 cmrsp13 cmrsp14 cmrsp15 cmrsp16 cmrsp17 cmrsp18 cmrsp19 cmrsp20   
## 0.70 0.61 0.95 0.81 0.80 0.46 0.58 0.33 0.72 0.88   
## cmrsp21   
## 0.90   
##   
## Loadings:  
## Factor1 Factor2  
## cmrsp01 0.54   
## cmrsp06 0.87   
## cmrsp07 0.81   
## cmrsp08 0.84   
## cmrsp09 0.80   
## cmrsp10 0.88   
## cmrsp11 0.52   
## cmrsp12 0.53   
## cmrsp02 -0.30 0.93   
## cmrsp16 0.75   
## cmrsp18 0.91   
## cmrsp19 0.52   
## cmrsp03   
## cmrsp04   
## cmrsp05 0.32   
## cmrsp13   
## cmrsp14 0.39   
## cmrsp15   
## cmrsp17 0.35 0.35   
## cmrsp20   
## cmrsp21   
##   
## Factor1 Factor2  
## SS loadings 5.07 3.12  
## Proportion Var 0.24 0.15  
## Cumulative Var 0.24 0.39  
##   
## Factor Correlations:  
## Factor1 Factor2  
## Factor1 1.00 -0.71  
## Factor2 -0.71 1.00  
##   
## Test of the hypothesis that 2 factors are sufficient.  
## The chi square statistic is 950 on 169 degrees of freedom.  
## The p-value is 4.96e-109

plot.new()  
load <- cmrsmlfit$loadings[,1:2]  
  
text(load,labels=names(cmrsitems),cex=.7) # add variable names



# Principal Axis Factor Analysis  
  
  
  
#library(psych)  
cmrsefa <- factor.pa(cmrsitemsShrunk, nfactors=5, rotate="promax")

## Warning in factor.pa(cmrsitemsShrunk, nfactors = 5, rotate = "promax"):  
## factor.pa is deprecated. Please use the fa function with fm=pa

cmrsefa # print results

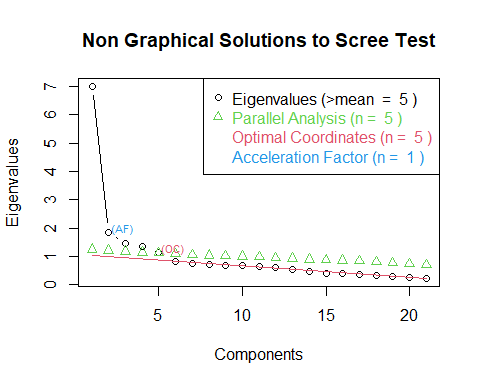
## Factor Analysis using method = pa  
## Call: factor.pa(r = cmrsitemsShrunk, nfactors = 5, rotate = "promax")  
## Unstandardized loadings (pattern matrix) based upon covariance matrix  
## PA1 PA3 PA4 PA2 PA5 h2 u2 H2 U2  
## cmrsp01 0.19 -0.01 0.65 0.04 -0.19 0.48 0.52 0.48 0.52  
## cmrsp02 -0.11 -0.81 -0.05 -0.04 0.06 0.58 0.42 0.58 0.42  
## cmrsp03 -0.11 0.06 0.66 0.02 -0.03 0.32 0.68 0.32 0.68  
## cmrsp04 -0.19 0.04 0.43 0.04 0.36 0.34 0.66 0.34 0.66  
## cmrsp05 0.04 -0.05 0.39 -0.06 0.17 0.31 0.69 0.31 0.69  
## cmrsp06 0.63 0.04 0.37 -0.04 -0.12 0.65 0.35 0.65 0.35  
## cmrsp07 0.74 -0.08 0.08 0.03 -0.12 0.62 0.38 0.62 0.38  
## cmrsp08 0.75 0.05 -0.03 0.06 0.14 0.65 0.35 0.65 0.35  
## cmrsp09 0.79 -0.01 -0.13 0.02 0.17 0.67 0.33 0.67 0.33  
## cmrsp10 0.71 0.09 0.09 -0.01 0.07 0.58 0.42 0.58 0.42  
## cmrsp11 0.53 -0.13 -0.05 -0.02 0.01 0.33 0.67 0.33 0.67  
## cmrsp12 0.24 0.01 0.33 -0.05 0.24 0.45 0.55 0.45 0.55  
## cmrsp13 -0.01 0.01 -0.19 -0.09 0.57 0.21 0.79 0.21 0.79  
## cmrsp14 0.21 0.13 -0.01 -0.06 0.52 0.31 0.69 0.31 0.69  
## cmrsp15 0.02 -0.01 0.17 -0.02 0.42 0.30 0.70 0.30 0.70  
## cmrsp16 0.09 -0.75 0.03 -0.10 -0.02 0.61 0.39 0.61 0.39  
## cmrsp17 0.15 -0.20 0.31 0.03 0.16 0.46 0.54 0.46 0.54  
## cmrsp18 0.02 -0.82 0.01 0.06 -0.04 0.70 0.30 0.70 0.30  
## cmrsp19 -0.10 -0.17 0.05 0.26 0.41 0.43 0.57 0.43 0.57  
## cmrsp20 0.01 0.05 0.08 0.87 -0.01 0.76 0.24 0.76 0.24  
## cmrsp21 0.07 -0.01 -0.09 0.75 0.01 0.58 0.42 0.58 0.42  
##   
## PA1 PA3 PA4 PA2 PA5  
## SS loadings 3.45 2.03 1.92 1.45 1.47  
## Proportion Var 0.16 0.10 0.09 0.07 0.07  
## Cumulative Var 0.16 0.26 0.35 0.42 0.49  
## Proportion Explained 0.33 0.20 0.19 0.14 0.14  
## Cumulative Proportion 0.33 0.53 0.72 0.86 1.00  
##   
## Standardized loadings (pattern matrix)  
## item PA1 PA3 PA4 PA2 PA5 h2 u2  
## cmrsp01 1 0.19 -0.01 0.65 0.04 -0.19 0.48 0.52  
## cmrsp02 2 -0.11 -0.81 -0.05 -0.04 0.06 0.58 0.42  
## cmrsp03 3 -0.11 0.06 0.66 0.02 -0.03 0.32 0.68  
## cmrsp04 4 -0.19 0.04 0.43 0.04 0.36 0.34 0.66  
## cmrsp05 5 0.04 -0.05 0.39 -0.06 0.17 0.31 0.69  
## cmrsp06 6 0.63 0.04 0.37 -0.04 -0.12 0.65 0.35  
## cmrsp07 7 0.74 -0.08 0.08 0.03 -0.12 0.62 0.38  
## cmrsp08 8 0.75 0.05 -0.03 0.06 0.14 0.65 0.35  
## cmrsp09 9 0.79 -0.01 -0.13 0.02 0.17 0.67 0.33  
## cmrsp10 10 0.71 0.09 0.09 -0.01 0.07 0.58 0.42  
## cmrsp11 11 0.53 -0.13 -0.05 -0.02 0.01 0.33 0.67  
## cmrsp12 12 0.24 0.01 0.33 -0.05 0.24 0.45 0.55  
## cmrsp13 13 -0.01 0.01 -0.19 -0.09 0.57 0.21 0.79  
## cmrsp14 14 0.21 0.13 -0.01 -0.06 0.52 0.31 0.69  
## cmrsp15 15 0.02 -0.01 0.17 -0.02 0.42 0.30 0.70  
## cmrsp16 16 0.09 -0.75 0.03 -0.10 -0.02 0.61 0.39  
## cmrsp17 17 0.15 -0.20 0.31 0.03 0.16 0.46 0.54  
## cmrsp18 18 0.02 -0.82 0.01 0.06 -0.04 0.70 0.30  
## cmrsp19 19 -0.10 -0.17 0.05 0.26 0.41 0.43 0.57  
## cmrsp20 20 0.01 0.05 0.08 0.87 -0.01 0.76 0.24  
## cmrsp21 21 0.07 -0.01 -0.09 0.75 0.01 0.58 0.42  
##   
## PA1 PA3 PA4 PA2 PA5  
## SS loadings 3.46 2.04 1.92 1.45 1.47  
## Proportion Var 0.16 0.10 0.09 0.07 0.07  
## Cumulative Var 0.16 0.26 0.35 0.42 0.49  
## Cum. factor Var 0.33 0.53 0.72 0.86 1.00  
##   
## With factor correlations of   
## PA1 PA3 PA4 PA2 PA5  
## PA1 1.00 -0.51 0.60 0.23 0.50  
## PA3 -0.51 1.00 -0.50 -0.32 -0.56  
## PA4 0.60 -0.50 1.00 0.18 0.57  
## PA2 0.23 -0.32 0.18 1.00 0.41  
## PA5 0.50 -0.56 0.57 0.41 1.00  
##   
## Mean item complexity = 1.5  
## Test of the hypothesis that 5 factors are sufficient.  
##   
## df null model = 210 with the objective function = 8.5 with Chi Square = 4361  
## df of the model are 115 and the objective function was 0.56   
##   
## The root mean square of the residuals (RMSR) is 0.02   
## The df corrected root mean square of the residuals is 0.03   
##   
## The harmonic n.obs is 522 with the empirical chi square 134 with prob < 0.1   
## The total n.obs was 522 with Likelihood Chi Square = 288 with prob < 8.2e-17   
##   
## Tucker Lewis Index of factoring reliability = 0.923  
## RMSEA index = 0.054 and the 90 % confidence intervals are 0.046 0.062  
## BIC = -432  
## Fit based upon off diagonal values = 0.99  
## Measures of factor score adequacy   
## PA1 PA3 PA4 PA2 PA5  
## Correlation of (regression) scores with factors 0.95 0.93 0.89 0.91 0.86  
## Multiple R square of scores with factors 0.90 0.86 0.79 0.83 0.74  
## Minimum correlation of possible factor scores 0.80 0.72 0.58 0.66 0.48

library (nFactors)

##   
## Attaching package: 'nFactors'

## The following object is masked from 'package:lattice':  
##   
## parallel

ev <- eigen(cor(cmrsitemsShrunk)) # get eigenvalues  
ap <- nFactors::parallel(subject=nrow(cmrsitems),var=ncol(cmrsitems),  
 rep=100,cent=.05)  
nS <- nScree(x=ev$values, aparallel=ap$eigen$qevpea)  
plotnScree(nS)



## Item Response Theory (IRT)

library(mirt)

## Loading required package: stats4

modelcmrs.grm<-'cmrs = 1-21'  
resultscmrs.grm<-mirt(data=cmrsitemsShrunk,   
 model=modelcmrs.grm, itemtype="graded", SE=TRUE, verbose=FALSE)  
coefcmrs.grm<-coef(resultscmrs.grm, IRTpars=TRUE, simplify=TRUE, na.rm=TRUE)  
itemscmrs.grm<-as.data.frame(coefcmrs.grm$items)  
print(itemscmrs.grm)

## a b1 b2 b3  
## cmrsp01 1.409 -1.17e-01 1.658 2.57  
## cmrsp02 0.992 -9.64e-01 0.842 2.28  
## cmrsp03 0.757 -3.27e-01 1.422 2.75  
## cmrsp04 1.186 1.66e+00 2.800 3.63  
## cmrsp05 1.222 7.45e-01 2.055 2.85  
## cmrsp06 2.342 -2.85e-01 0.610 1.16  
## cmrsp07 2.299 -5.94e-01 0.391 1.09  
## cmrsp08 2.845 9.79e-02 0.941 1.62  
## cmrsp09 2.781 -2.85e-05 0.948 1.65  
## cmrsp10 2.337 7.99e-02 0.995 1.65  
## cmrsp11 1.249 -1.27e+00 0.127 1.31  
## cmrsp12 1.839 2.44e-02 1.313 2.25  
## cmrsp13 0.643 2.91e+00 4.788 5.65  
## cmrsp14 1.298 1.29e+00 2.597 3.92  
## cmrsp15 1.107 7.36e-01 2.324 3.72  
## cmrsp16 1.443 -5.78e-01 0.722 1.59  
## cmrsp17 1.769 -2.73e-01 1.015 2.01  
## cmrsp18 1.539 -5.47e-01 0.635 1.39  
## cmrsp19 1.162 9.48e-01 2.461 3.68  
## cmrsp20 0.838 2.08e+00 3.839 4.95  
## cmrsp21 0.834 2.34e+00 4.278 5.37

##write\_csv(itemscmrs.grm, "itemscmrs.csv")

#Sandbox

This is where stuff that is not finished goes. To get R to do a diagram for us, we can use the rockchalk package. Unfortunately, rockchalk doesn’t read the results from setCorr, so we have to switch to lm.

casiVera2<-lm(cdrs14~cdrstot0\*cgender, data=abacab) summary(casiVera2) sketch<-plotSlopes(casiVera2, plotx=“cdrstot0”, xlab=“Total CDRS-R Score (0=minimum)”, interval=“confidence”, opacity=80, col=“purple”, ylim=c(1,7), legendArgs=list(x=“topright”)) sketch

Code from EGC:

Reg <- lm(hopelec~CPSStot+fadtotalpomp, data=RES2) summary(Reg) RegModel <- lm(hopelec~CPSStot\*fadtotalpomp, data=RES2) summary(RegModel)

m3ps <- plotSlopes(RegModel, modx = “fadtotalpomp”, plotx = “CPSStot”, n=3, modxVals=“std.dev”) m3psts <- testSlopes(m3ps) round(m3psts$hypotests,4)

basicStats(RES) #Get N, min & max (compare to source docs), skew & kurtosis pairs.panels(RES) #Look at histograms, scatterplots, LOESS smoother #ggpairs(RES) pairs.panels(RES) #regression findings plot\_summs(RegModel, robust = TRUE, plot.distributions = TRUE, rescale.distributions = TRUE) export\_summs(RegModel, model.names = c(“Global Family Functioning’s Impact on the Relationship of Child Post-Traumatic Symptoms on Child Hopelessness and Helplessness Outlook”), robust = TRUE, VIF = TRUE, error\_format = “({statistic}, p = {p.value})[CI 95% {conf.low}, {conf.high}]”, model.fit = TRUE, to.file = “xlsx”, file.name = “finalmodels1.xlsx”)

The “codebook” package generates a huge amount of information about a data file. The defaults on this are extensive and take a LONG time to run. Prohibitively large for most uses. We could explore how to turn off a bunch of the features to make it more manageable.

#library(codebook)  
#codebook(bfi)

## Various ways of getting descriptives and checking assumptions

rstatix::get\_summary\_stats(): Compute summary statistics for one or multiple numeric variables. Can handle grouped data.

rstatix::freq\_table(): Compute frequency table of categorical variables.

rstatix::get\_mode(): Compute the mode of a vector, that is the most frequent values.

rstatix::identify\_outliers(): Detect univariate outliers using boxplot methods.

rstatix::mahalanobis\_distance(): Compute Mahalanobis Distance and Flag Multivariate Outliers.

rstatix::shapiro\_test() and mshapiro\_test(): Univariate and multivariate Shapiro-Wilk normality test.

get\_summary\_stats(modelvars) #gives IQR, but no skew or kurtosis

## Warning: attributes are not identical across measure variables; they will be  
## dropped

## # A tibble: 6 × 13  
## variable n min max median q1 q3 iqr mad mean sd se  
## <fct> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 cmrsptot 542 0 55 14 8 22 14 10.4 15.5 10.6 0.455  
## 2 pgbsleep 542 0 19 3 1 7 6 4.45 4.20 4.10 0.176  
## 3 bpvsall 542 0 1 0 0 0 0 0 0.194 0.396 0.017  
## 4 anyadhd 542 0 1 1 0 1 1 0 0.642 0.48 0.021  
## 5 agechild 542 4 18 11 8 13 5 4.45 10.6 3.52 0.151  
## 6 cgender 542 0 1 0 0 1 1 0 0.387 0.488 0.021  
## # ℹ 1 more variable: ci <dbl>

freq\_table(modelvars, cgroup5) # Gives counts and bin percentages, not cumulative

## # A tibble: 5 × 3  
## cgroup5 n prop  
## <fct> <int> <dbl>  
## 1 1 16 3   
## 2 2 89 16.4  
## 3 3 158 29.2  
## 4 4 237 43.7  
## 5 5 42 7.7

mahalanobis\_distance(modelvars) %>% arrange(desc(mahal.dist)) #very helpful for outliers

## # A tibble: 542 × 8  
## cmrsptot pgbsleep bpvsall anyadhd agechild cgender mahal.dist is.outlier  
## <dbl> <dbl> <dbl+lbl> <dbl+l> <dbl> <dbl+l> <dbl> <lgl>   
## 1 6 17 0 [No Bipol… 0 [No … 16 1 [fem… 20.4 FALSE   
## 2 47 18 1 [Bipolar … 0 [No … 14 1 [fem… 18.6 FALSE   
## 3 24 15 1 [Bipolar … 0 [No … 6 0 [mal… 18.1 FALSE   
## 4 34 2 1 [Bipolar … 0 [No … 16 1 [fem… 17.5 FALSE   
## 5 55 8 1 [Bipolar … 1 [ADH… 6 0 [mal… 17.5 FALSE   
## 6 34 19 0 [No Bipol… 1 [ADH… 9 0 [mal… 17.0 FALSE   
## 7 36 19 0 [No Bipol… 1 [ADH… 9 0 [mal… 17.0 FALSE   
## 8 43 17 0 [No Bipol… 1 [ADH… 14 0 [mal… 16.2 FALSE   
## 9 27 17 0 [No Bipol… 1 [ADH… 16 0 [mal… 15.5 FALSE   
## 10 52 14 1 [Bipolar … 1 [ADH… 5 0 [mal… 15.1 FALSE   
## # ℹ 532 more rows