Sisler Main Analysis

# LOAD PACKAGES

library(assertive)  
library(broom)  
library(bmem)

## Loading required package: Amelia

## Loading required package: Rcpp

## ##   
## ## Amelia II: Multiple Imputation  
## ## (Version 1.8.0, built: 2021-05-26)  
## ## Copyright (C) 2005-2021 James Honaker, Gary King and Matthew Blackwell  
## ## Refer to http://gking.harvard.edu/amelia/ for more information  
## ##

## Loading required package: MASS

## Loading required package: snowfall

## Loading required package: snow

library(car)

## Loading required package: carData

## Registered S3 methods overwritten by 'car':  
## method from  
## influence.merMod lme4  
## cooks.distance.influence.merMod lme4  
## dfbeta.influence.merMod lme4  
## dfbetas.influence.merMod lme4

library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

library(caTools)  
library(clipr)

## Welcome to clipr. See ?write\_clip for advisories on writing to the clipboard in R.

library(corrplot)

## corrplot 0.89 loaded

library(dgof)

##   
## Attaching package: 'dgof'

## The following object is masked from 'package:stats':  
##   
## ks.test

library(effectsize)  
library(emmeans)  
library(emulator)

## Loading required package: mvtnorm

library(expss)

## Registered S3 methods overwritten by 'expss':  
## method from   
## [.labelled Hmisc  
## as.data.frame.labelled base   
## print.labelled Hmisc

##   
## Use 'expss\_output\_viewer()' to display tables in the RStudio Viewer.  
## To return to the console output, use 'expss\_output\_default()'.

##   
## Attaching package: 'expss'

## The following object is masked from 'package:ggplot2':  
##   
## vars

## The following object is masked from 'package:car':  
##   
## recode

## The following object is masked from 'package:assertive':  
##   
## is\_na

library(FactoMineR)  
library(factoextra)

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

library(fastDummies)  
library(graphics)  
library(gvlma)  
library(haven)

##   
## Attaching package: 'haven'

## The following objects are masked from 'package:expss':  
##   
## is.labelled, read\_spss

library(Hmisc)

## Loading required package: survival

##   
## Attaching package: 'survival'

## The following object is masked from 'package:caret':  
##   
## cluster

## Loading required package: Formula

##   
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':  
##   
## format.pval, units

library(knitr)  
library(janitor)

##   
## Attaching package: 'janitor'

## The following objects are masked from 'package:stats':  
##   
## chisq.test, fisher.test

library(lavaan)

## This is lavaan 0.6-8  
## lavaan is FREE software! Please report any bugs.

library(lattice)  
library(leaps)  
library(magrittr)

##   
## Attaching package: 'magrittr'

## The following objects are masked from 'package:expss':  
##   
## and, equals, not, or

## The following objects are masked from 'package:assertive':  
##   
## is\_greater\_than, is\_less\_than

library(MBESS)

##   
## Attaching package: 'MBESS'

## The following object is masked from 'package:lavaan':  
##   
## cor2cov

library(mediation)

## Loading required package: Matrix

## Loading required package: sandwich

## mediation: Causal Mediation Analysis  
## Version: 4.5.0

library(medmod)

##   
## Attaching package: 'medmod'

## The following object is masked from 'package:magrittr':  
##   
## mod

library(missMDA)  
library(multilevel)

## Loading required package: nlme

library(MVN)

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

## sROC 0.1-2 loaded

library(mvord)

## Loading required package: minqa

## Loading required package: BB

## Loading required package: ucminf

## Loading required package: dfoptim

library(plspm)

##   
## Attaching package: 'plspm'

## The following object is masked from 'package:effectsize':  
##   
## normalize

## The following object is masked from 'package:ggplot2':  
##   
## alpha

library(pscl)

## Classes and Methods for R developed in the  
## Political Science Computational Laboratory  
## Department of Political Science  
## Stanford University  
## Simon Jackman  
## hurdle and zeroinfl functions by Achim Zeileis

library(psych)

##   
## Attaching package: 'psych'

## The following objects are masked from 'package:plspm':  
##   
## alpha, rescale, unidim

## The following object is masked from 'package:mediation':  
##   
## mediate

## The following object is masked from 'package:MBESS':  
##   
## cor2cov

## The following object is masked from 'package:lavaan':  
##   
## cor2cov

## The following object is masked from 'package:Hmisc':  
##   
## describe

## The following object is masked from 'package:emulator':  
##   
## tr

## The following object is masked from 'package:effectsize':  
##   
## phi

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

## The following object is masked from 'package:car':  
##   
## logit

library(rcompanion)

##   
## Attaching package: 'rcompanion'

## The following object is masked from 'package:psych':  
##   
## phi

## The following object is masked from 'package:effectsize':  
##   
## phi

library(readxl)  
library(reshape2)  
library(rockchalk)

##   
## Attaching package: 'rockchalk'

## The following object is masked from 'package:Hmisc':  
##   
## summarize

## The following object is masked from 'package:effectsize':  
##   
## standardize

## The following object is masked from 'package:MASS':  
##   
## mvrnorm

library(rosetta)

##   
## Attaching package: 'rosetta'

## The following object is masked from 'package:psych':  
##   
## reliability

## The following object is masked from 'package:nlme':  
##   
## getData

## The following object is masked from 'package:expss':  
##   
## recode

## The following objects are masked from 'package:lattice':  
##   
## histogram, oneway

library(rstatix)

##   
## Attaching package: 'rstatix'

## The following object is masked from 'package:janitor':  
##   
## make\_clean\_names

## The following objects are masked from 'package:effectsize':  
##   
## cohens\_d, eta\_squared

## The following object is masked from 'package:MASS':  
##   
## select

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

library(semTools)

##

## ###############################################################################

## This is semTools 0.5-4

## All users of R (or SEM) are invited to submit functions or ideas for functions.

## ###############################################################################

##   
## Attaching package: 'semTools'

## The following object is masked from 'package:rosetta':  
##   
## reliability

## The following object is masked from 'package:rockchalk':  
##   
## kurtosis

## The following objects are masked from 'package:psych':  
##   
## reliability, skew

## The following object is masked from 'package:expss':  
##   
## net

library(stargazer)

##   
## Please cite as:

## Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.

## R package version 5.2.2. https://CRAN.R-project.org/package=stargazer

library(stats)  
library(stringr)

##   
## Attaching package: 'stringr'

## The following objects are masked from 'package:expss':  
##   
## fixed, regex

library(sure)  
library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

## ✓ tibble 3.1.2 ✓ purrr 0.3.4  
## ✓ tidyr 1.1.3 ✓ dplyr 1.0.7  
## ✓ readr 1.4.0 ✓ forcats 0.5.1

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## x psych::%+%() masks ggplot2::%+%()  
## x psych::alpha() masks plspm::alpha(), ggplot2::alpha()  
## x magrittr::and() masks expss::and()  
## x dplyr::between() masks expss::between()  
## x readr::clipboard() masks semTools::clipboard()  
## x dplyr::collapse() masks nlme::collapse()  
## x dplyr::compute() masks expss::compute()  
## x dplyr::contains() masks tidyr::contains(), expss::contains()  
## x magrittr::equals() masks expss::equals()  
## x tidyr::expand() masks Matrix::expand()  
## x tidyr::extract() masks magrittr::extract()  
## x dplyr::filter() masks rstatix::filter(), stats::filter()  
## x dplyr::first() masks expss::first()  
## x stringr::fixed() masks expss::fixed()  
## x tibble::has\_rownames() masks assertive::has\_rownames()  
## x purrr::is\_atomic() masks assertive::is\_atomic()  
## x purrr::is\_character() masks assertive::is\_character()  
## x purrr::is\_double() masks assertive::is\_double()  
## x purrr::is\_empty() masks assertive::is\_empty()  
## x purrr::is\_formula() masks assertive::is\_formula()  
## x purrr::is\_function() masks assertive::is\_function()  
## x magrittr::is\_greater\_than() masks assertive::is\_greater\_than()  
## x purrr::is\_integer() masks assertive::is\_integer()  
## x magrittr::is\_less\_than() masks assertive::is\_less\_than()  
## x purrr::is\_list() masks assertive::is\_list()  
## x purrr::is\_logical() masks assertive::is\_logical()  
## x purrr::is\_null() masks assertive::is\_null()  
## x purrr::is\_numeric() masks assertive::is\_numeric()  
## x purrr::is\_vector() masks assertive::is\_vector()  
## x haven::is.labelled() masks expss::is.labelled()  
## x purrr::keep() masks expss::keep()  
## x dplyr::lag() masks stats::lag()  
## x dplyr::last() masks expss::last()  
## x purrr::lift() masks caret::lift()  
## x medmod::mod() masks magrittr::mod()  
## x purrr::modify() masks expss::modify()  
## x purrr::modify\_if() masks expss::modify\_if()  
## x dplyr::na\_if() masks expss::na\_if()  
## x tidyr::nest() masks expss::nest()  
## x magrittr::not() masks expss::not()  
## x magrittr::or() masks expss::or()  
## x tidyr::pack() masks Matrix::pack()  
## x haven::read\_spss() masks expss::read\_spss()  
## x dplyr::recode() masks rosetta::recode(), expss::recode(), car::recode()  
## x stringr::regex() masks expss::regex()  
## x dplyr::select() masks rstatix::select(), MASS::select()  
## x purrr::set\_names() masks magrittr::set\_names()  
## x purrr::some() masks car::some()  
## x dplyr::src() masks Hmisc::src()  
## x dplyr::summarize() masks rockchalk::summarize(), Hmisc::summarize()  
## x purrr::transpose() masks expss::transpose()  
## x tidyr::unpack() masks Matrix::unpack()  
## x dplyr::vars() masks expss::vars(), ggplot2::vars()  
## x purrr::when() masks expss::when()

library(tseries)

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

##   
## Attaching package: 'tseries'

## The following object is masked from 'package:sure':  
##   
## surrogate

library(validate)

##   
## Attaching package: 'validate'

## The following object is masked from 'package:dplyr':  
##   
## expr

## The following objects are masked from 'package:Hmisc':  
##   
## label, label<-

## The following object is masked from 'package:ggplot2':  
##   
## expr

library(writexl)

# PREPARING FILES

#IMPORT file  
 # data.raw <- read\_excel("~/Desktop/Desktop\_Home/UTAH/Stats\_I/R\_Studio/Dissertation\_files/Sisler\_dissertation\_HMS1819\_audit.xlsx",  
 # range=cell\_cols(1:1820), col\_types = "text")  
 #   
 # save("data.raw", file="~/Desktop/Desktop\_Home/UTAH/Stats\_I/R\_Studio/Dissertation\_files/Sisler\_dissertation\_HMS1819\_audit.RData")  
 load("~/Desktop/Desktop\_Home/UTAH/Stats\_I/R\_Studio/Dissertation\_files/Sisler\_dissertation\_HMS1819\_audit.RData")  
 # source("dissertation\_fx.R")  
   
data.v1 <- data.raw %>%  
 mutate(age = as.numeric(age)) %>%  
 filter(age < 26 & age >= 18) %>% # exclusion criteria. Filter out under 18s, and over 26s  
 select(-c( # narrowing columns numbers to reduce output matrices  
  
 # demographic info  
 children\_dep, educ\_par1\_rel, educ\_par2\_rel, hours\_work\_paid, timeclass, timestud, educ\_par1, educ\_par2, age\_US, langpast,   
 langcur, age18:age41pl, educ\_par, eductemp, educ\_1:educ\_4, age18\_22, age31pl,  
 # relationship info  
 relship, relship\_text, rel\_sing:rel\_divw,  
 # HMS-based  
 SweepsIncentiveContact, endmessageSuicide, UniversityofSouthernCalifornia1a, consentFollowup1, excl\_pers\_oth\_TEXT,   
 exp\_rude,exp\_accuse, exp\_othersafraid, exp\_ideaignore, exp\_offendjoke, exp\_stupid, exp\_notserious, exp\_superficial,  
 exp\_re\_prof, exp\_re\_fear, exp\_re\_speakall,  
 # religious affiliations  
 relig\_aff\_ag:relig\_aff\_oth, relig\_aff\_other\_text, relig\_1:relig\_5,  
 # residence  
 res\_dorm:res\_oth, residenc\_text, deport\_worry,  
 # degree categories  
 degree\_ass:degree\_other\_text, enroll\_text, deg\_ass:deg\_nd,  
 # major/field choices  
 field\_hum:field\_other, field\_other\_text,  
 # unused scales  
 phq2\_1, phq2\_2,  
 # sex abuse/assault:  
 ## type  
 assault\_sex\_stranger:assault\_sex\_dontknow, abuse\_recent\_perpet, sa\_exp\_touch:sa\_exp\_pen,   
 sa\_suc\_phyforc\_1, sa\_att\_phyforc, sa\_drunk\_1, sa\_suc\_phyforc\_2\_1:sa\_suc\_phyforc\_2\_6,   
 sa\_suc\_phyforc\_2\_6\_text, aca\_phys\_assault\_1:aca\_phys\_assault\_7, aca\_sex\_assault\_1:aca\_sex\_assault\_7,  
 ## reporting  
 sa\_tell\_1:sa\_tell\_12, sa\_tell\_12\_text, sa\_nodiscl\_1:sa\_nodiscl\_24, sa\_nodiscl\_24\_text,  
 ## person  
 sa\_who\_1:sa\_who\_12, sa\_who\_12\_text, sa\_who\_stud, sa\_who\_employ, sa\_who\_gender, sa\_who\_gender\_text,  
 ## location/context  
 sa\_loc\_on, sa\_op\_substance\_1:sa\_op\_substance\_5, sa\_frighten, sa\_loc\_off, sa\_perc\_1:sa\_perc\_12,   
 sa\_train\_pol, sa\_train\_prev,  
 # sleep variables  
 sleep\_wk1, sleep\_wd1, sleep\_wk2, sleep\_wd2, sleep\_np1, sleep\_np2,  
 # stress variables  
 stress1:compet\_field,  
 # D/O SUBTYPES:  
 ## mood  
 dx\_dk, dx\_dep\_1:dx\_bip\_4, dep\_maj, dep\_oth, dx\_bip\_bi1:dx\_bip\_oth, dep\_imp1:dep\_imp4,  
 ## anxiety  
 dx\_ax\_1:dx\_ax\_7\_new, dx\_ax\_6\_text\_new, aca\_anx\_1:aca\_add\_7, dx\_ax\_ga:dx\_ax\_so,  
 ## OCD  
 dx\_ocd\_1:dx\_ocd\_6, dx\_ocd\_6\_text, dx\_ocd\_o:dx\_ocd\_oth,  
 ## trauma  
 dx\_trauma\_1:dx\_trauma\_4, dx\_trauma\_4\_text, dx\_trauma\_ptsd:dx\_trauma\_oth,  
 ## eating disorder  
 dx\_ea\_1:dx\_ea\_4\_text, ed\_thin, weight\_ideal, diet, ea\_fast, purge\_1:purge\_4, ed\_peer,   
 body\_con\_change, fresh15, weigh\_freq, weight\_ideal\_text, dx\_ea\_an:dx\_ea\_av,  
 ## psychosis  
 dx\_psy\_1:dx\_psy\_8, dx\_psy\_7\_text, dx\_psy\_sa:dx\_psy\_sp,  
 ## substance-related  
 dx\_sa\_1, dx\_sa\_2, dx\_sa\_3, dx\_sa\_2\_text, aca\_substance\_1:aca\_substance\_7, sa\_drunk\_2\_1:sa\_drunk\_2\_7,   
 sa\_drunk\_2\_6\_text, sa\_drunk\_3:sa\_drunk\_6, alc\_prob, sub\_cig1:sub\_cig5, sa\_train\_use:sa\_sch\_help,  
 ## personality d/o  
 dx\_perso\_1:dx\_perso\_12, dx\_perso\_11\_text,  
 ## chronic conditions  
 aca\_phys\_health\_1:aca\_phys\_health\_7, dx\_chronic\_11\_text,  
 ## SIBs  
 sib\_fre1:sub\_cig,  
 # provider-related questions:   
 ther\_helped\_me, prov\_1:prov\_3, prov\_7:prov\_10, prov\_9\_text, anyprovi,  
 # insurance specifics  
 ins\_2:ins\_10, ins\_pare:ins\_unce, ins\_yesb, ins\_mh1:ins\_mh5, ins\_ina\_1:ins\_ina\_8\_text,  
 # diet-related  
 eatprac\_1:eatprac\_3, eatprac\_frveg, eat\_change, eat\_change\_how\_1:eat\_change\_how\_15, eat\_change\_how\_15\_text,  
 # SES  
 fin\_comp2, pay\_worry, pay\_fam, pay\_self, pay\_other, pay\_grant, pay\_loan, fincur\_1:finpas\_5,  
 # CCAPS  
 CCAPS\_1:CCAPS\_34,  
 # CCMH:   
 sib\_ccmh\_1:suic\_ccmh\_6, cons\_viol\_ccmh\_number, caus\_viol\_ccmh\_number, exp\_har\_ccmh\_number,   
 exp\_trauma\_ccmh\_number, cons\_viol\_ccmh\_recent, caus\_viol\_ccmh\_recent, exp\_har\_ccmh\_recent,   
 exp\_trauma\_ccmh\_recent, trauma\_ccmh\_text, traumevent\_ccmh\_1, traumevent\_ccmh\_2, traumevent\_ccmh\_3,   
 traumevent\_ccmh\_4, traumevent\_ccmh\_5, traumevent\_ccmh\_6, traumevent\_ccmh\_7, traumevent\_ccmh\_8,   
 traumevent\_ccmh\_9, traumevent\_ccmh\_10, traumevent\_ccmh\_11, traumevent\_ccmh\_12, traumevent\_ccmh\_13, traumevent\_ccmh\_14,   
 traumevent\_ccmh\_15, traumevent\_ccmh\_16, traumevent\_ccmh\_15\_text, binge\_fr\_ccmh,  
 # HMS-CCMH  
 dep\_CCMH\_score\_temp, dep\_CCMH\_score, dep\_CCMH\_elevated, dep\_CCMH\_mild, dep\_CCMH\_none, genanx\_CCMH\_score\_temp,  
 genanx\_CCMH\_score, genanx\_CCMH\_elevated, genanx\_CCMH\_mild, genanx\_CCMH\_none, socanx\_CCMH\_score\_temp,   
 socanx\_CCMH\_score, socanx\_CCMH\_elevated, socanx\_CCMH\_mild, socanx\_CCMH\_none, eatconcerns\_CCMH\_score\_temp,  
 eatconcerns\_CCMH\_score, eatconcerns\_CCMH\_elevated, eatconcerns\_CCMH\_mild, eatconcerns\_CCMH\_none, dep\_or\_anx\_CCMH,  
 sib\_ccmh\_11:sib\_ccmh\_15, sib\_ccmh\_lifetime\_any, sib\_ccmh\_pastyear,  
 # Qnum:   
 Q2\_11, Q2\_12, Q3\_8, Q3\_9, Q3\_10\_2, Q3\_10\_1, Q3\_11\_1, Q3\_12\_1:Q3\_12\_5, Q2\_10\_152:Q2\_10\_155, Q4\_21\_11,   
 Q9\_19, Q9\_20,  
 # health knowledge:   
 know\_sp, dep\_tx\_know\_1:dep\_tx\_know\_4, dep\_sx\_know\_1:dep\_sx\_know\_4, anx\_help\_know\_1:anx\_help\_know\_4,  
 ea\_sx\_know\_1:ea\_sx\_know\_6,  
 # therapy-related  
 why\_tx\_1:why\_tx\_9\_text, ther\_lifetime, ther\_lifetime2, ther\_any, talk1\_any, talk1\_8\_text, ther\_any2,   
 knowwher1:knowwher6, bar\_any:bar\_plan, ther\_help\_me1:ther\_help\_me4, meds\_help\_me1:meds\_help\_me4,   
 talksup1:talksup4, talk2pro:talk2noo,  
 # school climate  
 cli\_disab:cli\_sh\_oth\_TEXT, excl\_env:exp\_community, val\_faculty:excl\_pers\_oth, group\_TEXT:group\_active,  
 friend\_re:friend\_religid, social\_re:social\_religid, re\_find:time\_activ,  
 # SAT-related  
 sat\_hours\_1, sat\_loc\_1, sat\_qual\_1, sat\_priv\_1, sat\_sched\_1, sat\_hours\_2, sat\_loc\_2, sat\_qual\_2, sat\_priv\_2, sat\_sched\_2,   
 sat\_hours\_3, sat\_loc\_3, sat\_qual\_3, sat\_priv\_3, sat\_sched\_3, sat\_hours\_4, sat\_loc\_4, sat\_qual\_4, sat\_priv\_4, sat\_sched\_4,   
 sat\_hours\_5, sat\_loc\_5, sat\_qual\_5, sat\_priv\_5, sat\_sched\_5, sat\_hours\_6, sat\_loc\_6, sat\_qual\_6, sat\_priv\_6, sat\_sched\_6,   
 sat\_hours\_7, sat\_loc\_7, sat\_qual\_7, sat\_priv\_7, sat\_sched\_7, sat\_hours\_8, sat\_loc\_8, sat\_qual\_8, sat\_priv\_8, sat\_sched\_8,   
 sat\_hours\_9, sat\_loc\_9, sat\_qual\_9, sat\_priv\_9, sat\_sched\_9, sat\_hours\_1\_1:sat\_hours\_1\_6, sat\_loc\_1\_1:sat\_loc\_1\_6,   
 sat\_qual\_1\_1:sat\_qual\_1\_6, sat\_priv\_1\_1:sat\_priv\_1\_6, sat\_sched\_1\_1:sat\_sched\_1\_6, sat\_hours\_2\_1:sat\_hours\_2\_6,   
 sat\_loc\_2\_1:sat\_loc\_2\_6, sat\_qual\_2\_1:sat\_qual\_2\_6, sat\_priv\_2\_1:sat\_priv\_2\_6, sat\_sched\_2\_1:sat\_sched\_2\_6,   
 sat\_hours\_3\_1:sat\_hours\_3\_6, sat\_loc\_3\_1:sat\_loc\_3\_6, sat\_qual\_3\_1:sat\_qual\_3\_6, sat\_priv\_3\_1:sat\_priv\_3\_6,   
 sat\_sched\_3\_1:sat\_sched\_3\_6, sat\_hours\_7\_1:sat\_hours\_7\_6, sat\_loc\_7\_1:sat\_loc\_7\_6, sat\_qual\_7\_1:sat\_qual\_7\_6,   
 sat\_priv\_7\_1:sat\_priv\_7\_6, sat\_sched\_7\_1:sat\_sched\_7\_6, sat\_hours\_8\_1:sat\_hours\_8\_6, sat\_loc\_8\_1:sat\_loc\_8\_6,   
 sat\_qual\_8\_1:sat\_qual\_8\_6, sat\_priv\_8\_1:sat\_priv\_8\_6, sat\_sched\_8\_1:sat\_sched\_8\_6,  
 # peer-related  
 peer\_alc\_1, peer\_alc\_3, peer\_alc\_4, peer\_mar\_est, peer\_alc\_est, peer\_cig\_est, inf\_7\_text, risk\_cig:risk\_presc,  
 # pregnancy-related  
 sex\_partner, sex\_partner\_f, sex\_partner\_m, sex\_partner\_t, sex\_30, sex\_30\_oral, sex\_30\_vag, sex\_30\_anal,   
 birthcontrol\_1:birthcontrol\_12, birthcontrol\_10\_text, birthcontrol\_always, preg\_no, preg\_yes\_u,preg\_yes\_in,   
 preg\_dk, preg\_cur,  
 # medication details  
 meds\_reason\_1:meds\_reason\_5, meds\_reason\_5\_text, meds\_dis, meds\_w\_1:meds\_w\_5, meds\_w\_3\_text, meds\_cur\_1:meds\_cur\_9,   
 meds\_cur\_7\_text, meds\_time\_1:meds\_time\_7, med\_help, meds\_helped\_me, meds\_w\_g:meds\_di5,  
 # campus or community  
 heard, camp\_supp, outreach\_aware, gk\_1, gk\_2, gkt\_any, txfrf, cli\_look, cli\_resp\_fr, timestud\_1:timestud\_7,  
 cli\_resp\_cl, cli\_free, st\_promote, admin\_listen, env\_body, safe\_on\_day, safe\_on\_night, safe\_off\_day,   
 safe\_off\_night, choose\_sch, persist\_challenge\_1:persist\_challenge\_11, persist\_challenge\_11\_text, highestdeg,  
 highestdeg\_text, doubt\_school\_1, doubt\_school\_2, prof\_support\_aca, different\_maj, late\_assign, sleep\_class,  
 aca\_support\_serv, office\_hrs, faculty\_out, advisor\_interact, gradstud\_interact, time\_manage, grade\_curv,  
 # up/bystanding:  
 inter\_hy\_dr, inter\_hy\_sa, inter\_hy\_hl, inter\_y\_1:inter\_y\_7, inter\_y\_6\_text, inter\_help, inter\_n\_1:inter\_n\_7,   
 inter\_n\_6\_text, inter\_n\_why\_1:inter\_n\_why\_9, inter\_n\_why\_9\_text, witness\_1:witness\_7, witness\_6\_text,   
 inf\_any:inf\_oth,  
 # hypothetical:  
 ther\_help, talk2\_1:talk2\_7, talk2\_7\_text, talksup, percneed1:percneed\_cur6, stig\_t\_I:stig\_t\_I6,  
 # misc:   
 bar\_hs\_1:bar\_ns\_8\_text, smok\_freq, sib\_other\_text, activ\_other\_text, knowwher\_temp, exerc\_changed,   
 exerc\_changed\_how, exerc\_changed\_how\_text, height\_ft, height\_in, wgt\_lbs, prov\_A, prov\_B, prov\_C,   
 prov\_loc, prov\_olo, prov\_oth, prov\_dk, adjust\_aca\_1, adjust\_aca\_2, aca\_imp1:aca\_imp4, s\_c\_1\_1:s\_c\_5\_6,   
 s\_n\_1\_1:s\_n\_5\_6))

# Call functions

# delogodd -- de-logs odds ratio when needed  
delogodd <- function(x) {  
 oddsratio <- exp(x)  
 p <- oddsratio / (1 + oddsratio)  
 return(p)  
}  
  
# NA\_preproc -- helps format dataframes with NAs  
NA\_preproc <- function (dat) {  
 for (j in 1:ncol(dat)) {  
 x <- dat[[j]]  
 if (is.factor(x) && anyNA(x)) dat[[j]] <- base::addNA(x)  
 if (is.character(x)) dat[[j]] <- factor(x, exclude = NULL)  
 }  
 dat  
}  
  
# SEM visualization  
plot\_matrix <- function(matrix\_toplot){  
corrplot::corrplot(matrix\_toplot, is.corr = FALSE,  
 type = 'lower',  
 order = "original",  
 tl.col='black', tl.cex=.75)  
 }

# OUTCOME VARIABLE

SUICIDAL IDEATION (si\_type) Creates indicator variable ‘si\_type’ using the subtypes of suicidal ideation using sui\_idea, sui\_plan, sui\_attempt, where: 0/0/0 = no SI 1/0/0 = passive SI 1/1/0 = active SI 1/1/1 = volitional +/- ‘phq9\_9’ and ‘sui\_idea’ into one variable – suic\_idea. Will run the end model twice.

data.v2 <- data.v1 %>%  
  
# create si\_type var  
 mutate(si\_type = case\_when(sui\_idea == 1 ~ 1, # SI present  
 sui\_idea == 0 ~ 0)) %>% # no SI reported  
 mutate(si\_activ = case\_when((sui\_idea == 1 & sui\_plan == 1) ~ 1,  
 (sui\_idea == 1 & sui\_plan == 0) ~ 0,  
 sui\_idea == 0 ~ 0)) %>%  
 filter(!is.na(si\_type)) %>% # filter for SI variable -- then rowwise deletion  
 filter(!is.na(si\_activ)) # filter for SI variable -- then rowwise deletion  
   
data.v2$si\_type <- as.logical(data.v2$si\_type)   
data.v2$si\_activ <- as.logical(data.v2$si\_activ)   
 # tabulate and check output  
  
table(data.v2$si\_type, useNA = "always")

##   
## FALSE TRUE <NA>   
## 44525 6485 0

table(data.v2$si\_activ, useNA = "always")

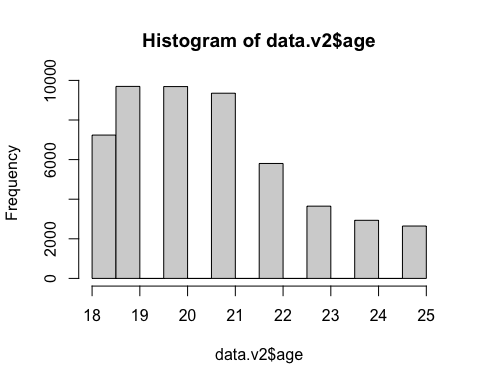
##   
## FALSE TRUE <NA>   
## 48166 2844 0

# PREMOTIVATION VARIABLES

data.v2 <- data.v2

## Age

data.v2$age <- as.numeric(data.v2$age)  
hist(data.v2$age)



table(data.v2$age, useNA = "always")

##   
## 18 19 20 21 22 23 24 25 <NA>   
## 7239 9697 9688 9353 5805 3649 2934 2645 0

jarque.bera.test(data.v2$age)

##   
## Jarque Bera Test  
##   
## data: data.v2$age  
## X-squared = 3174.5, df = 2, p-value < 2.2e-16

data.v2 <- data.v2 %>%  
 mutate(undercls = case\_when(age == 18 ~ 1,   
 age == 19 ~ 1,  
 age == 20 ~ 0,   
 age == 21 ~ 0,   
 age == 22 ~ 0,   
 age == 23 ~ 0,   
 age == 24 ~ 0,   
 age == 25 ~ 0))  
data.v2$undercls <- as.logical(data.v2$undercls)  
table(data.v2$undercls, useNA = "always")

##   
## FALSE TRUE <NA>   
## 34074 16936 0

## Gender identity

data.v3 <- data.v2 %>%   
 # Create 'gender' subtypes into one category  
 mutate(gender = case\_when((gender == 3 | gender == 4) ~ 3, # transgender  
 (gender == 5 | gender == 6) ~ 2, # gender diverse  
 (gender == 1 | gender == 2) ~ 1, # cis-gender  
 FALSE ~ 4)) # sets up non-answers  
table(data.v3$gender)

##   
## 1 2 3   
## 49460 1226 277

# transform into cisgender and non-cisgender  
data.v3 <- data.v3 %>%   
 mutate(gender\_tgd = case\_when(gender == 2 ~ 1,  
 gender == 3 ~ 1,  
 gender == 1 ~ 0)) %>%  
 filter(!is.na(gender))  
data.v3$gender\_tgd <- as.logical(data.v3$gender\_tgd)  
  
table(data.v3$gender\_tgd, useNA = "always")

##   
## FALSE TRUE <NA>   
## 49460 1503 0

jarque.bera.test(data.v3$gender\_tgd)

##   
## Jarque Bera Test  
##   
## data: data.v3$gender\_tgd  
## X-squared = 2040971, df = 2, p-value < 2.2e-16

### Reconcile ‘gender\_text’

# gender-diverse  
data.v3$gender[str\_detect(data.v3$gender\_text, "[D|d]emi")] <- 1  
data.v3$gender[str\_detect(data.v3$gender\_text, "[B|b]inary")] <- 1  
data.v3$gender[str\_detect(data.v3$gender\_text, "[N|n]eutral")] <- 1  
data.v3$gender[str\_detect(data.v3$gender\_text, "[Q|q]ueer")] <- 1  
data.v3$gender[str\_detect(data.v3$gender\_text, "[F|f]luid")] <- 1  
data.v3$gender[str\_detect(data.v3$gender\_text, "[A|a]gender")] <- 1  
data.v3$gender[str\_detect(data.v3$gender\_text, "[A|a]ndrogynous")] <- 1  
data.v3$gender[str\_detect(data.v3$gender\_text, "[P|p]angender")] <- 1  
data.v3$gender[str\_detect(data.v3$gender\_text, "[B|b]igender")] <- 1  
data.v3$gender[data.v3$gender\_text %in% c("she/they", "Closeted genderfluid (female presenting)", "Pan-gender",  
 "female / infinite star being / gender non conforming","He or they",  
 "Not sure, nonbinary I guess", "Non-binary trans maculine", "No preference",   
 "Somewhere between GNC and transfemme", "She/they",  
 "Non-binary leaning towards trans man")] <- 1  
  
 #transgender  
data.v3$gender[data.v3$gender\_text %in% c("Transmasculine", "Trans-masculine person", "Trans Masculine",   
 "gender non-conforming trans man")] <- 1  
  
 # male  
data.v3$gender[str\_detect(data.v3$gender\_text, "[D|d]ude")] <- 0  
data.v3$gender[str\_detect(data.v3$gender\_text, "[M|m]ale")] <- 0  
data.v3$gender[str\_detect(data.v3$gender\_text, "[B|b]oy")] <- 0  
data.v3$gender[data.v3$gender\_text %in% c("Male: there are only 2 genders", "if you have a penis you're a male",   
 "Cisgender man", "Male motherfucker. stop making up gender shit.", "He-male",  
 "I have male parts so therefore I am a MALE and there are only TWO genders!!!!!!!!!!!!!!!!!",  
 "SATSYG (Savage about to steal yo girl", "My sex is male",   
 "I say male, but I think gender is fluid", "Theres only two genders, I'm male",  
 "Probably heterosexual, but so commitment-phobic that I can't actually tell if I'm asexual.",  
 "I'm a male. Born a male, always will be a male. There should only be two choices.")] <- 0  
  
 # female  
data.v3$gender[str\_detect(data.v3$gender\_text, "[F|f]emale")] <- 0  
data.v3$gender[str\_detect(data.v3$gender\_text, "[W|w]oman")] <- 0  
data.v3$gender[str\_detect(data.v3$gender\_text, "[G|g]irl")] <- 0  
data.v3$gender[data.v3$gender\_text %in% c("Male and Female are not genders, they are sexes. I identify as a woman.",   
 "Female, but currently going through a gender identity crisis!")] <- 0  
  
 # questioning/other  
data.v3$gender[str\_detect(data.v3$gender\_text, "[Q|q]uestioning")] <- 1  
data.v3$gender[str\_detect(data.v3$gender\_text, "[U|u]nsure")] <- 1  
data.v3$gender[str\_detect(data.v3$gender\_text, "[I|i]dk")] <- 1  
data.v3$gender[str\_detect(data.v3$gender\_text, "Hermaphrodite")] <- 1  
data.v3$gender[str\_detect(data.v3$gender\_text, "Questioning")] <- 1  
data.v3$gender[str\_detect(data.v3$gender\_text, "Undecided")] <- 1  
data.v3$gender[data.v3$gender\_text %in% c("Literally no clue as of late", "Kynigender", "Adrogynous", "Hermaphrodite",   
 "idk yet", "Questioning", "Questioning; female", "Still questioning", "Not sure",  
 "not female/questioning", "Questioning (unsure)", "Girl ? Idk", "not sure",   
 "Not Sure???", "I don't know, still trying to purse it out", "Undecided")] <- 1  
  
 # non-answers jokesters  
data.v3$gender[str\_detect(data.v3$gender\_text, "[H|h]elicopter")] <- 4  
data.v3$gender[str\_detect(data.v3$gender\_text, "[C|c]hinook")] <- 4  
data.v3$gender[str\_detect(data.v3$gender\_text, "[A|a]pache")] <- 4  
data.v3$gender[str\_detect(data.v3$gender\_text, "[U|u]nicorn")] <- 4  
data.v3$gender[data.v3$gender\_text %in% c("Albino stingray", "Karate Chop Action", "Flying Spaghetti Monster", "Pio Nono",  
 "God", "None", "Baseball bat", "Mountain lion", "Oil Platform", "Chair", "Pizza",   
 "M1A1 Main Battle Tank", "Deppressed", "Fire hydrant/hockey puck", "Microwave",  
 "This doesn't matter and you're an idiot for thinking it does.","REZZBIAN",  
 "Baguette", "fairy", "a 5 year old girl", "tree", "unicorn", "His Majesty",   
 "Call of Duty: Modern Warfare 2", "Attack hellcopter", "Meme God", "AH-64D",   
 "Stainless Steel Toaster", "God himself", "baseball bat", "Mountain","Trans Former",   
 "Potato", "I will literally give you $100 to h\*ck off", "Indian tape worm",   
 "Metallic Hyper-Entity", "2X4 treated lumber", "pizza", "sPecIaL SNowFlAkE",   
 "eh", "George Michael", "Metaknight Main", "Sea turtle", "I identify as a fruit cup",  
 "The Senate", "1941 Nihon Giagun Yamato-Class Battleship", "Attack Hellicopter",   
 "penguin wearing a sombrero and sunglasses")] <- 4  
  
 # non-answers, decline to answer  
data.v3$gender[str\_detect(data.v3$gender\_text, "[D|d]ecline")] <- "non-answers"  
data.v3$gender[data.v3$gender\_text %in% c("I am me. A human.", "I'm not yet sure where I identify", "Human-being", "genders",   
 "okay", "guess", "none", "Just my name.", "the above options are sex, not genders",   
 "the above options are sex, not genders", "its difficult", "any", "?",   
 "Biological male who identifies as a lesbian female.", "Prefer not to answer",  
 "The universe conscious of itself", "no", "I only try to use gender when it's useful",   
 "fuck gender", "Gender and sex are the same", "gender is a lie", "#love", "No",  
 "it's weird to separate out trans male and trans female if you're also asking for birth assignment.",   
 "2 genders", "I don't understand the question.", "Prefer not to specifiy",  
 "'Male' and 'female' are not gender identities.",  
 "I don't have one because it's not a legitimate concept.",   
 "Gender is a construct which serves no purpose.")] <- 4  
  
data.v3 <- data.v3 %>%  
 filter(!is.na(gender\_tgd))  
  
table(data.v3$gender\_tgd, useNA = "always") # check output

##   
## FALSE TRUE <NA>   
## 49460 1503 0

jarque.bera.test(data.v3$gender\_tgd)

##   
## Jarque Bera Test  
##   
## data: data.v3$gender\_tgd  
## X-squared = 2040971, df = 2, p-value < 2.2e-16

## Sexual Identity

data.v4 <- data.v3 %>%  
 # aggregates all 'sexual' answers into one var  
 mutate(  
 sexual = case\_when(sexual\_queer == 1 ~ 1, # queer  
 sexual\_bi == 1 ~ 1, # bisexual  
 sexual\_l == 1 ~ 1, # lesbian   
 sexual\_g == 1 ~ 1, # gay  
 sexual\_quest == 1 ~ 1, # questioning  
 sexual\_h == 1 ~ 0, # heterosexual  
 sexual\_other == 1 ~ 1) # other  
)   
data.v4$sexual <- as.logical(data.v4$sexual)  
table(data.v4$sexual, useNA = "always") # tabulate and check outputs

##   
## FALSE TRUE <NA>   
## 39905 10843 215

### Reconcile sexual\_text

# identifying non-answers  
 ## jokesters  
data.v4$sexual[str\_detect(data.v4$sexual\_text, "[H|h]elicopter")] <- NA  
data.v4$sexual[data.v4$sexual\_text %in% c("Yes", "i identify as a lamp", "I only like baseballs", "helisexual", "Autosexual",  
 "now", "Miguel Garcia", "toasteroven", "must have gills", "Potsexual", "Bobcats",  
 "Tailpipes turn me on", "Tyrannosaurus rex", "Traps are not gay", "Toaster sexual",  
 "Crooked", "Pantsexual", "FREE", "Frozen Foods", "Giraffe",   
 "I'm sexually attracted to fire", "Attracted to Vehicles", "sPecIaL SNowFlAkE",  
 "Attracted to other self-identifying flying spaghetti monsters",   
 "Demiqueer quasiflux asexual foxkin", "I sleep around. Also into bestiality",   
 "Sexually attracted to kitchen appliances", "Elitistual. I'm into elitists",  
 "sexually attracted to fictional characters and men twice my age")] <- NA  
 ## declined to answer  
 data.v4$sexual[str\_detect(data.v4$sexual\_text, "identi[t|f]y")] <- NA  
 data.v4$sexual[str\_detect(data.v4$sexual\_text, "label")] <- NA  
 data.v4$sexual[str\_detect(data.v4$sexual\_text, "matter")] <- NA  
 data.v4$sexual[str\_detect(data.v4$sexual\_text, "[W|w]hatever")] <- NA  
 data.v4$sexual[str\_detect(data.v4$sexual\_text, "box myself")] <- NA  
 data.v4$sexual[str\_detect(data.v4$sexual\_text, "[D|d]ecline")] <- NA  
 data.v4$sexual[str\_detect(data.v4$sexual\_text, "[N|n]ormal")] <- NA  
 data.v4$sexual[str\_detect(data.v4$sexual\_text, "[D|d]isclose")] <- NA  
 data.v4$sexual[str\_detect(data.v4$sexual\_text, "[N|n]one")] <- NA  
 data.v4$sexual[str\_detect(data.v4$sexual\_text, "[U|u]nknown")] <- NA  
 data.v4$sexual[str\_detect(data.v4$sexual\_text, "[N|n]/[A|a]")] <- NA  
 data.v4$sexual[str\_detect(data.v4$sexual\_text, "[I|i] like")] <- NA  
 data.v4$sexual[data.v4$sexual\_text %in% c("dating a trans male", "Attracted to personalities, not genders", "non",  
 "irrelevant", "i like people", "Fuck sexuality", "personality based",   
 "Don't think about it too much", "Anyone that I like", "leave me alone",  
 "define", "This is more bullshit", "No", "Me", "I am who I am", "Just there",  
 "Don't care", "no label", "No Label", "No labels", "No", "Me",  
 "Questions like this are the reason we are even having this debate.",   
 "i like who i like", "prefer not", "I do not understand why this is necessary",  
 "Just there", "non", "Normal", "Human Being", "human being", "Closeted at home",   
 "It doesn't matter my question is why is this an actual question",   
 "I hate this PC bs", "FUCK YOU BIG TIME", "Polyamorous", "gay leaning bi",  
 "Sexuality shouldn't matter", "No preference", "i like boys.",   
 "other", "Prefer not to specify", "i fall in love with a human being",  
 "everyone's a little gay", "I prefer men, but have had women before",   
 "Traps are not gay", "Unknown", "#love", "Whatever I feel like at the time",  
 "Situational", "No one right now", "sometimes i dabble", "People", "Prefer not",  
 "I am", "Whatever Happens", "Crooked", "love is love", "Open", "Autistic",  
 "Human", "No one right now", "Situational", "I don't", "?", "Undefined",   
 "The universe conscious of itself", "Aromantic", "Repressed", "Never had sex",  
 "Not dealing with men or women at the moment", "not sure", "Not sure",  
 "None?", "neutral", "no sexual orientation", "No idea, but I'm not straight",  
 "Unlabeled", "Curious", "Anyone that I like", "Disinterested", "Do not define",  
 "Fake Gay", "irrelevant", "People", "no particular sexual identity",   
 "No preference", "love is love", "there is no such thing",   
 "Sometimes gender doesn't matter and I don't have time to learn about these things life gets too busy",  
 "Prefer not to answer", "just sexual (need no prefix!)", "It's complicated",   
 "Sexuality shouldn't matter",   
 "I'm just me and I love who I love. I don't like these labels.",  
 "I'm not a fucking part of lgbtqrstuxyz bullshit association")] <- NA  
   
 # identifying queer individuals:  
data.v4$sexual[str\_detect(data.v4$sexual\_text, "[Q|q]ueer")] <- 1  
data.v4$sexual[str\_detect(data.v4$sexual\_text, "[F|f]luid")] <- 1  
data.v4$sexual[str\_detect(data.v4$sexual\_text, "[O|o]pen")] <- 1  
data.v4$sexual[data.v4$sexual\_text %in% c("potentially interested", "somewhere between", "homoflexible", "heteroflexible",  
 "fluid", "Only other trans people", "non-binary", "so I guess I could check queer",  
 "Androsexual", "Not straight", "Polysexual", "Joto", "Non-binary", "nonbinary",  
 "Attracted to masculinity and androgony", "Androsexual")] <- 1  
  
 # identifying questioning  
data.v4$sexual[str\_detect(data.v4$sexual\_text, "[Q|q]uestion")] <- 1  
data.v4$sexual[str\_detect(data.v4$sexual\_text, "[P|p]robably")] <- 1  
data.v4$sexual[data.v4$sexual\_text %in% c("possibly", "curious", "Unknown", "potentially interested", "don't know", "Unknown",   
 "don't know", "not sure", "still figuring it out", "idk", "Idk", "don't know",   
 "I have a girlfriend but i am not attracted to any other females. I am attracted to males",   
 "Sometimes gender doesn't matter and I don't have time to learn about these things life gets too busy",  
 "Not sure. Working on it.")] <- 1  
  
 # identifying pansexual write-ins  
data.v4$sexual[str\_detect(data.v4$sexual\_text, "[P|p]an")] <- 1  
data.v4$sexual[str\_detect(data.v4$sexual\_text, "[S|s]apiosexual")] <- 1  
data.v4$sexual[str\_detect(data.v4$sexual\_text, "[S|s]apiosexual")] <- 1  
data.v4$sexual[data.v4$sexual\_text %in% c("all", "Polyamorous Pansexual", "Pan Sexual", "Pan-sexual",   
 "any but ive only had relationships with men")] <- 1  
  
 # identifying asexual / demisexual write-ins:   
data.v4$sexual[str\_detect(data.v4$sexual\_text, "[G|g]r[a|e]y")] <- 1  
data.v4$sexual[str\_detect(data.v4$sexual\_text, "[N|n]onsexual")] <-1  
data.v4$sexual[str\_detect(data.v4$sexual\_text, "[C|c]elibate")] <- 1  
data.v4$sexual[str\_detect(data.v4$sexual\_text, "[D|d]emi")] <- 1  
data.v4$sexual[str\_detect(data.v4$sexual\_text, "[A|a]sex")] <- 1  
data.v4$sexual[str\_detect(data.v4$sexual\_text, "[C|c]elibate")] <- 1  
data.v4$sexual[str\_detect(data.v4$sexual\_text, "[A|a]ce")] <- 1  
data.v4$sexual[data.v4$sexual\_text %in% c("Mostly asexual", "Most likely asexual", "Masexual", "Tridium Vykon Jace 8000",   
 "Demi-romantic", "asexual lesbian", "Ase", "Bi-demisexual", "A sexual",  
 "Hetroromantic Asexual", "Demiqueer quasiflux asexual foxkin", "DemiAsexual",  
 "a-sexual", "no sexual attraction", "a sexual", "Homoromantic Demisexual",  
 "Gay-ish, I am Demi-sexual which means I don't pay attention to sex/gender",   
 "experience no sexual attraction but do experience a small amount of romantic attraction",   
 "lithromantic", "Pan-Quoiromantic Asexual", "demiromantic asexual")] <- 1  
  
 # identifying heterosexual individuals  
data.v4$sexual[str\_detect(data.v4$sexual\_text, "[H|h]etero")] <- 0  
data.v4$sexual[str\_detect(data.v4$sexual\_text, "[S|s]traight")] <- 0  
data.v4$sexual[str\_detect(data.v4$sexual\_text, "[H|h]etrosex")] <- 0  
data.v4$sexual[data.v4$sexual\_text %in% c("Kinsey scale 1", "I would put myself as a 1-1.5 on the kinsey scale", "Staight",   
 "Stragiht", "strait", "Str8!!!!", "Changing from gay to heterosexual",  
 "Gynophilic", "Into females", "I fuck bitches.", "1 on the Kinsey Scale",  
 "Gynophilic", "gynosexual", "Changing from gay to heterosexual", "100% Hetro")] <- 0  
  
 # identified as gay or lesbian  
data.v4$sexual[str\_detect(data.v4$sexual\_text, "[G|g]ay")] <- 1  
data.v4$sexual[str\_detect(data.v4$sexual\_text, "[H|h]omoflexible")] <- 1  
data.v4$sexual[data.v4$sexual\_text %in% c("Mostly gay", "soooooo gay", "Romantically straight and sexually gay")] <- 1  
data.v4$sexual[str\_detect(data.v4$sexual\_text, "[L|l]esbian")] <- 1  
data.v4$sexual[str\_detect(data.v4$sexual\_text, "[D|d]yke")] <- 1  
  
 # identifying bisexual individuals  
data.v4$sexual[str\_detect(data.v4$sexual\_text, "[B|b]isex")] <- 1  
data.v4$sexual[str\_detect(data.v4$sexual\_text, "[B|b]icurious")] <- 1  
data.v4$sexual[data.v4$sexual\_text %in% c("I'm straight but I like girls too sexually, not emotionally.", "Half bisexual",   
 "straight but DTF sexy females", "bicurious but i like girls more",   
 "hetero with occasional bi tendencies", "gay leaning bi", "Bi-curious",   
 "Herero-romantic bisexual", "Bi with a stronger attraction to same sex",  
 "I'm like halfway to bisexual. Never had a crush on another girl but I'm still interested. Idk.",  
 "No boundaries in place - though probably bisexual")] <- 1  
  
table(data.v4$sexual) # tabulate and check outputs

##   
## 0 1   
## 39941 10675

### ‘Sexual’ transformed

sexual identity: (1) heterosexual; (2) lesbian/gay/bisexual; (3) queer/pan/asex/other, and (4) non-answers

data.v5 <- data.v4 %>%  
 # aggregates all 'sexual' answers into one var  
 mutate(sexual = ifelse(sexual == 1, 1, 0)) %>%  
 filter(!is.na(sexual))  
  
data.v5$sexual <- as.logical(data.v5$sexual)  
   
table(data.v5$sexual, useNA = "always") # tabulate and check outputs

##   
## FALSE TRUE <NA>   
## 39941 10675 0

jarque.bera.test(data.v5$sexual)

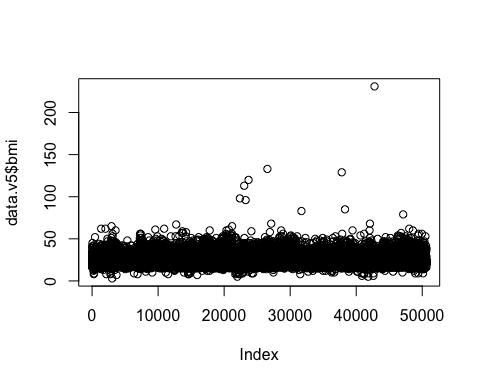
##   
## Jarque Bera Test  
##   
## data: data.v5$sexual  
## X-squared = 16947, df = 2, p-value < 2.2e-16

## BMI

data.v5$bmi <- as.integer(data.v5$bmi)  
table(data.v5$bmi)

##   
## 3 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19   
## 1 2 2 2 12 17 10 9 14 11 42 139 436 1213 2417 3966   
## 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35   
## 4794 6108 5353 5030 4151 3825 2128 1811 1499 1314 983 842 652 545 453 365   
## 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51   
## 354 279 227 216 144 113 104 92 71 58 52 43 37 13 18 26   
## 52 53 54 55 56 57 58 59 60 61 62 65 67 68 79 83   
## 13 16 9 1 10 3 2 2 7 2 4 2 1 2 1 1   
## 85 96 98 113 120 129 133 231   
## 1 1 1 1 1 1 1 1

scatter.smooth(data.v5$bmi)



## Chronic conditions

table(data.v5$dx\_chronic\_1)

##   
## 1   
## 4

table(data.v5$dx\_chronic\_2)

##   
## 1   
## 10

table(data.v5$dx\_chronic\_3)

##   
## 1   
## 96

table(data.v5$dx\_chronic\_4)

##   
## 1   
## 10

table(data.v5$dx\_chronic\_5)

##   
## 1   
## 12

table(data.v5$dx\_chronic\_6)

##   
## 1   
## 5

table(data.v5$dx\_chronic\_7)

##   
## 1   
## 1

table(data.v5$dx\_chronic\_8)

##   
## 1   
## 2

table(data.v5$dx\_chronic\_9)

## < table of extent 0 >

table(data.v5$dx\_chronic\_11)

##   
## 1   
## 25

table(data.v5$dx\_chronic\_12)

##   
## 1   
## 471

table(data.v5$dx\_chronic\_13) # n overall for all chronic = 622, pos \*\* too small to include

##   
## 1   
## 25

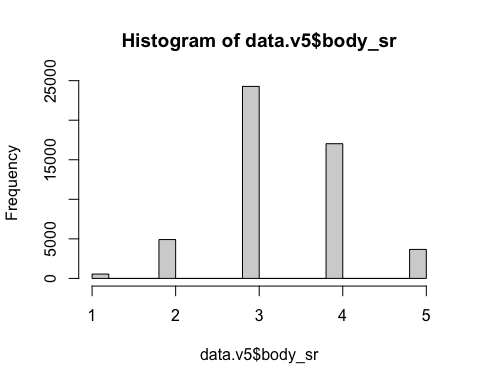
# data.v5 = subset(data.v5, select= -c(dx\_chronic\_1:dx\_chronic\_13)) # drop chronic as possibly flawed variable and low sample

## Subjective weight and body image

data.v5$body\_sr <- as.numeric(data.v5$body\_sr)  
table(data.v5$body\_sr)

##   
## 1 2 3 4 5   
## 548 4910 24274 17028 3664

hist(data.v5$body\_sr)



## Race

table(data.v5$race)

## < table of extent 0 >

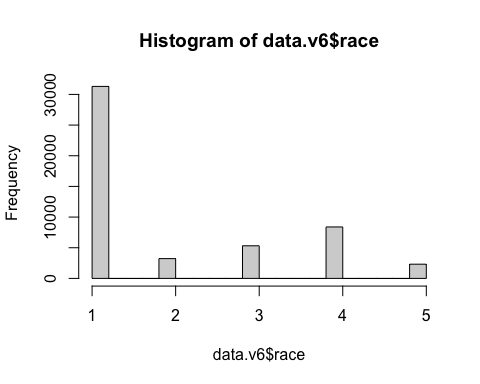
data.v6 <- data.v5 %>%   
# 'race' subtypes combined into one variable  
 mutate(race = case\_when(race\_bla == 1 ~ 2,  
 race\_ame == 1 ~ 5,  
 race\_asi == 1 ~ 4,  
 race\_his == 1 ~ 3,  
 race\_pac == 1 ~ 5,  
 race\_ara == 1 ~ 5,  
 race\_whi == 1 ~ 1,  
 race\_oth == 1 ~ 5,  
 FALSE ~ 5))  
table(data.v6$race) # check output

##   
## 1 2 3 4 5   
## 31307 3226 5318 8389 2318

data.v6 <- data.v6 %>% # filter out NAs for complete data  
 filter(!is.na(race))  
  
table(data.v6$race, useNA = "always") # double check NA output

##   
## 1 2 3 4 5 <NA>   
## 31307 3226 5318 8389 2318 0

hist(data.v6$race) # visualization distribution and skew



jarque.bera.test(data.v6$race)

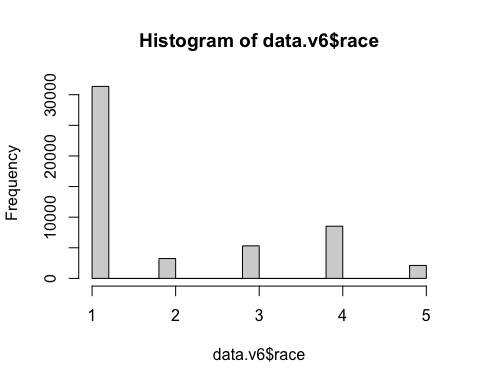
##   
## Jarque Bera Test  
##   
## data: data.v6$race  
## X-squared = 8604.6, df = 2, p-value < 2.2e-16

### Race write-in reconcilation

#decline to answer  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[A|a]merica")] <- 5  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[H|h]uman")] <- 5  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[N|n]one")] <- 5  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[P|p]erson")] <- 5  
data.v6$race[data.v6$race\_other\_text %in% c("prefer not to say", "Prefer not to answer.", "Prefer not to specify",   
 "Prefer not to answer :)", "FUCK YOU", "Caramel American", "Why", "irrelevant",  
 "Prefer not to state", "Prefer no to say", "Brown", "No thx", "Texan",   
 "Race is a problematic social construct", "Race is a socual construct",   
 "Transcendent", "Race/ethnicity should not matter", "Prefer not to answer",   
 "why is this a thing", "White passing, ethnically other", "unfortunately",   
 "While I understand the socio-economic conditions that surround race, race also is a construct which serves no purpose.",  
 "Wakanda Forever", "whatever I can use to be a victim in the given situation",   
 "We need to get rid of these color things", "Me", "I do Not identify racially",   
 "Why is there always a race question? Can't we all just be one race? The Human Race.",   
 "Will not disclose", "Race/ethnicity should not matter", "N/a", "N/A",   
 "There is only one human race: Homo sapien", "Citizen of the World",  
 "don't know due to uncertainity of lineage", "I choose not to answer",  
 "This doesn't matter and you're an idiot for thinking it does.",   
 "POC because now my opinion will matter in the US",   
 "Color doesn't mean anything right? Why do so many surveys ask it",  
 "My skin is lighter than others, and darker than some",   
 "I am not a racial identifying person. I am American. My DNA says something about Korean heritage.",  
 "I do not identify as a race in order to promote social equality for all races",  
 "I don't think race is all that important.",   
 "I do not think that this information should or does make a difference ,so, I chose to not state",   
 "A White, Cherokee, European, Irish man who would be preferred to be called else but white",  
 "I won't allow my race to be used in this study for potential insidious usage")] <- 5  
  
 # Jokesters  
data.v6$race[data.v6$race\_other\_text %in% c("Volvo", "Vulcan", "Snowman", "tree", "General Electric", "Cat", "Earthling",   
 "I have rainbow skin.", "jedi", "Light skinned potato", "I'm stainless steel.",   
 "7-foot tall Chinese woman", "A little tan.", "Apache helicopter", "Aptenodytes",  
 "Wood", "moon man", "Black hawk", "Aviation Heritage", "Ewok", "Aptenodytes",  
 "Desert Camoflauge", "Attack helicopter", "Ethnic Kekistani")] <- 5  
  
 # white  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[J|j]ew")] <- 1 # if Israeli, could be White, Asian, or ME (e.g. Palestine)  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[A|a]shkenazi")] <- 1  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[C|c]aucasian")] <- 1  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[E|e]uropean")] <- 1  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[G|g]reek")] <- 1  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[I|i]rish")] <- 1  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[I|i]talian")] <- 1  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[A|a]lbania")] <- 1  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[S|s]em[e|i]tic")] <- 1  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[R|r]ussian")] <- 1 # could be Asian depending on region  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[R|r]oma")] <- 1  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[T|t]urkish")] <- 1  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[S|s]lavic")] <- 1  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[M|m]etis")] <- 1  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[B|b]asque")] <- 1  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[S|s]icilian")] <- 1  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[P|p]olish")] <- 1  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[S|s]candinavian")] <- 1  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[F|f]innish")] <- 1  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[C|c]eltic")] <- 1  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[S|s]cottish")] <- 1  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[C|c]anad")] <- 1  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[M|m]editerranean")] <- 1  
data.v6$race[data.v6$race\_other\_text %in% c("please refer to me as Caucasian; it's extremely offensive to refer to every race-- excluding Caucasians-- in their politically correct form",  
 "Not sure if I'm Hispanic, as I am Portuguese and Spanish (from my mother's side) . I identify as white as I am of European descent.",  
 "Portugese", "Cape Verdean", "Capeverdean", "Bulgarian-American", "Ginger",   
 "Anglo Saxon", "anglo/white", "Balkan for ethnicity", "Ulster-scots",   
 "I'm white, but I do have some Native American in me but it's not the dominant visible ethnicity.",   
 "White (Jewish/Jamaican-Chinese/Irish)", "ARMENIAN")] <- 1  
 # Identifies as American Indian or other indigenous descent  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[N|n]ative [A|a]merican")] <- 5  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[I|i]ndigenous")] <- 5  
data.v6$race[data.v6$race\_other\_text %in% c("I look white but im also status native", "Indigenous/Mestizo",   
 "american indian that isn'y legally claimed")] <- 5  
 # Identifies as Asian  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[A|a]sian")] <- 4  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[C|c]hinese")] <- 4  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[J|j]apanese")] <- 4  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[I|i]ndian")] <- 4  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[B|b]anglades")] <- 4  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[F|f]ilipino")] <- 4  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[G|g]uyanese")] <- 4  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[M|m]ongolia")] <- 4  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[C|c]hina")] <- 4  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[S|s]outheast")] <- 4  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[S|s]outh [A|a]sian")] <- 4  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[P|p]unjabi")] <- 4  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[S|s]iberia")] <- 4  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[I|i]ndonesia")] <- 4  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[W|w]est [I|i]ndian")] <- 4  
data.v6$race[data.v6$race\_other\_text %in% c("Sri Lankan", "Sri-Lankan", "Malays", "Chinese Ethinicy-Nuosu", "Austronesian",  
 "third culture kid, born and raised in asia until 18yo", "Central Asian",   
 "Indian, UAE native", "Indian/Subcontinental", "Indians arent on the list.....",   
 "American Texan of Indian (South asia) Descent", "Okinawan", "Asian/Chinese",  
 "Japanese American and Native Hawaiian Culture", "Asian. International.",   
 "Australian/Indian","Indo-Caribbean", "Atlantic Islander",   
 "Telugu-speaking Tamil-ethnicity Indian")] <- 4  
 # Identifies as Black  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[A|a]frican")] <- 2  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[A|a]fro")] <- 2  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[K|k]enya")] <- 2  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[B|b]lack")] <- 2  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[C|c]arribean")] <- 2  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[B|b]ahamian")] <- 2  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[H|h]aitian")] <- 2  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[H|h]aitian")] <- 2  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[J|j]amaica")] <- 2  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[Z|z]imbabwea")] <- 2  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[G|g]hana")] <- 2  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[E|e]thiopia")] <- 2  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[T|t]rinidad")] <- 2  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[E|e]ritrean")] <- 2  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[W|w]akanda")] <- 2  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[N|n]igerian")] <- 2  
data.v6$race[data.v6$race\_other\_text %in% c("Cameroonian American", "Creole", "Jamacian American", "North Africa")] <- 2  
  
 # Identifies as Hispanic  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[M|m]exican")] <- 3  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[X|x]ican[o|a]")] <- 3  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[L|l]atin[o|a]")] <- 3  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[P|p]uerto [R|r]ican")] <- 3  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[S|s]panish")] <- 3  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[C|c]hican[o|a]")] <- 3  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[I|i]berian")] <- 3  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[M|m]estiz[o|a]")] <- 3  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[P|p]ortuguese")] <- 3  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[B|b]razil")] <- 3  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[C|c]uba")] <- 3  
data.v6$race[data.v6$race\_other\_text %in% c("Costa Rican", "Central American Native", "Andean")] <- 3  
  
 # Identifies as Middle Eastern or of Arab descent  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[A|a]rmenia")] <- 5  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[P|p]akistani")] <- 5  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[A|a]rab")] <- 5  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[P|p]ersian")] <- 5  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[P|p]alestin")] <- 5  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[I|i]srael")] <- 5  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[A|a]fghan")] <- 5  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[N|n]orth [A|a]frica")] <- 5  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[H|h]azara")] <- 5  
data.v6$race[data.v6$race\_other\_text %in% c("Northern African; Moroccan; Israeli", "Arab Mix", "Berber/North African",   
 "Coptic Egyptian (non-Arab)", "Kekistani")] <- 5  
  
 # Identifies all other data or mixed ethicity  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[B|b]iracial")] <- 5  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[B|b]i-racial")] <- 5  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[M|m]ix")] <- 5  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[H|h]alf")] <- 5  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[I|i]ndo-[C|c]arribean")] <- 5  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[M|m]ulti")] <- 5  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[M|m]ulatto")] <- 5  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[P|p]anda")] <- 5  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[O|o]ther")] <- 5  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[U|u]nkown")] <- 5  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[A|a]mbiguous")] <- 5  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[A|a]mbiguous")] <- 5  
data.v6$race[data.v6$race\_other\_text %in% c("Sub Asian/East African", "two or more races", "Moroccan Mexican",  
 "Transracial", "Adopted", "Part Thetan", "Taiwanese-Greek-Cypriot American",  
 "East Indian, Caribbean", "50/50", "Korean Ecuadorian", "Brazilian, Luso,  
 Native American", "Central American/ Caribbean", "Asian/Indian",  
 "Mut (more than 3)", "mexican,filipino, italian", "Black Hispanic",  
 "Middle Eastern, Filipino and Spanish", "East Indian, Caribbean",  
 "Indo-Portugesse (Fijian/Portugesse) Canadian Born", "Multiple",  
 "White and American Indian (Choctaw)", "White, Black, and Hispanic",  
 "White, black, Native American", "French Lebanese", "Mixed Race",   
 "Lebanese, German, French", "Black/Asian", "Mixed", "Portuguese/Thai",   
 "An American Mutt- a bit of everything somewhere in there",   
 "Black and native", "Black/cuban", "Filipino and Portuguese", "mixed race",  
 "Eastern Indian/white", "white middle eastern", "Transracial",  
 "German, Chinese, Arabic, Melanesian", "Hispanic and Middle Eastern",  
 "mulatto", "mixed & south american", "international", "Biracial",  
 "Multiracial", "Korean Ecuadorian", "Central American/ Caribbean",  
 "An American Mutt- a bit of everything somewhere in there")] <- 5  
 # Identifies as black and white as 'Other/Mixed'  
data.v6$race[data.v6$race\_other\_text %in% c("Black/White","White/African American", "White/African American")] <- 5  
 # Identifies as Asian and White as 'Other/Mixed'  
data.v6$race[str\_detect(data.v6$race\_other\_text, "[E|e]urasian")] <- 5  
data.v6$race[data.v6$race\_other\_text %in% c("Japanese, Chinese, French, British, Greek", "Asian/White",  
 "Asian/European", "Asian and White", "asian and white",   
 "White (Caucasian) and Indian (S. Asia)", "Chinese/White",  
 "American/Asian African/American White", "3/4 White, 1/4 Japanese",   
 "White & Asian", "Chinese and White")] <- 5  
 # Identifies as Hispanic and White as 'Other/Mixed'  
data.v6$race[data.v6$race\_other\_text %in% c("I am 1/4th Hispanic 3/4 German", "Honduran, Irish", "White hispanic",  
 "White, and Hispanic", "Hispanic/Jewish", "White and Hispanic", "White Hispanic",  
 "French and spanish", "LATINA/WHITE", "white/hispanic",  
 "If I identify as puerto rican does that mean i get more scholarship money?")] <- 5  
 # Identifies as MR and White as 'Other/Mixed'  
data.v6$race[data.v6$race\_other\_text %in% c("Half Middle Eastern half white",   
 "Half my family is white and the other middle eastern")] <- 5  
  
# tabulate and check outputs  
table(data.v6$race, useNA = "always")

##   
## 1 2 3 4 5 <NA>   
## 31357 3244 5314 8524 2119 0

hist(data.v6$race) # visualizae distribution and skew



jarque.bera.test(data.v6$race)

##   
## Jarque Bera Test  
##   
## data: data.v6$race  
## X-squared = 8608.3, df = 2, p-value < 2.2e-16

## SIBs and frequency

data.v7 <- data.v6 %>%  
 mutate(sib\_freq = case\_when(sib\_none == 1 ~ 0, # none  
 sib\_freq == 1 ~ 0, # 1-2x/yr  
 sib\_freq == 2 ~ 1, # <1x/mo  
 sib\_freq == 3 ~ 1, # 2-3x/mo  
 sib\_freq == 4 ~ 1, # 1-2x/wk  
 sib\_freq == 5 ~ 1, # 3-5x/wk  
 sib\_freq == 6 ~ 1)) %>% # almost everyday  
 filter(!is.na(sib\_freq)) # filter for NAs  
data.v7$sib\_freq <- as.logical(data.v7$sib\_freq) # move frequency to numeric class  
  
table(data.v7$sib\_freq, useNA = "always") # use for NA changes

##   
## FALSE TRUE <NA>   
## 33937 4529 0

data.v7 = subset(data.v7, select= -c(sib\_none:sib\_oth)) # clean up vars/columns

## SCOFF scale

data.v7$ed\_scoff <- as.numeric(data.v7$ed\_scoff)  
table(data.v7$ed\_scoff)

##   
## 0 1 2 3 4 5   
## 18763 9355 5698 2941 1033 227

hist(data.v7$ed\_scoff)



data.v7 = subset(data.v7, select= -c(scoff\_1:scoff\_5)) # clean up vars/columns

## Past finances

data.v7 <- data.v7 %>%  
 mutate(finpast = case\_when(finpast == 5 ~ 1,  
 finpast == 4 ~ 2,  
 finpast == 3 ~ 3,  
 finpast == 2 ~ 4,  
 finpast == 1 ~ 5)) %>%  
 filter(!is.na(finpast))  
  
table(data.v7$finpast, useNA = "always") # check output

##   
## 1 2 3 4 5 <NA>   
## 7786 12364 9965 5506 2807 0

## Co-occuring MH diagnoses

data.v8 <- data.v7  
  
# To ensure those who skipped the question are not accidentally added to No MH dx category  
data.v8$dx\_dep[is.na(data.v8$dx\_dep)] <- 0  
data.v8$dx\_anx[is.na(data.v8$dx\_anx)] <- 0  
data.v8$dx\_bi[is.na(data.v8$dx\_bi)] <- 0  
data.v8$dx\_psy[is.na(data.v8$dx\_psy)] <- 0  
data.v8$dx\_ea[is.na(data.v8$dx\_ea)] <- 0  
data.v8$dx\_tr[is.na(data.v8$dx\_tr)] <- 0  
data.v8$dx\_sa[is.na(data.v8$dx\_sa)] <- 0  
data.v8$dx\_neurodev\_1[is.na(data.v8$dx\_neurodev\_1)] <- 0  
data.v8$dx\_pers[is.na(data.v8$dx\_pers)] <- 0  
  
data.v8$dx\_adhd <- data.v8$dx\_neurodev\_1 # rename ADHD to a single category  
data.v8$dx\_adhd[is.na(data.v8$dx\_adhd)] <- 0  
  
data.v9 <- data.v8 %>% # filters out any respondents who skipped the MH section   
 mutate(dx\_mh = case\_when(dx\_dep == 1 ~ 1,  
 dx\_anx == 1 ~ 1,  
 dx\_bi == 1 ~ 1,  
 dx\_psy == 1 ~ 1,  
 dx\_ea == 1 ~ 1,  
 dx\_tr == 1 ~ 1,  
 dx\_sa == 1 ~ 1,  
 dx\_adhd == 1 ~ 1,  
 dx\_pers == 1 ~ 1,  
 dx\_any == 1 ~ 1,  
 dx\_none == 0 ~ 0)) %>%  
 mutate(dx\_mh = case\_when(dx\_mh == 1 ~ 1, # creates dichotomous variable  
 dx\_mh == 0 ~ 0,  
 is.na(dx\_mh) ~ 0))  
data.v9$dx\_mh <- as.logical(data.v9$dx\_mh) # turns dx\_mh into boolean  
  
  
table(data.v9$dx\_mh, useNA = "always") # Check output, skew

##   
## FALSE TRUE <NA>   
## 26197 12231 0

jarque.bera.test(data.v9$dx\_mh)

##   
## Jarque Bera Test  
##   
## data: data.v9$dx\_mh  
## X-squared = 6998, df = 2, p-value < 2.2e-16

# DATA DIVISION for final sample

## Data div and stats

data.v10 <- as.data.frame(data.v9)  
  
# Split data.v10 into two halves for training and testing datasets  
set.seed(500)   
split = sample.split(data.v10, SplitRatio = 0.50)  
   
# Create training and testing sets  
traindf.v10 <- subset(data.v10, split == TRUE) # train  
testdf.v10 <- subset(data.v10, split == FALSE) # test  
   
#Check output  
dim(data.v10)

## [1] 38428 449

dim(traindf.v10)

## [1] 19172 449

dim(testdf.v10)

## [1] 19256 449

## AUDIT

\*\* too small for inclusion

traindf.v10 <- traindf.v10 %>%  
 # Combine audit\_3\_f, audit\_3\_m, audit\_3\_o to create 'audit\_3'  
 mutate(audit\_3 = case\_when(audit\_3\_f == 1 ~ 1,  
 audit\_3\_m == 1 ~ 1,  
 audit\_3\_o == 1 ~ 1,  
 audit\_3\_f == 0 ~ 0,  
 audit\_3\_m == 0 ~ 0,  
 audit\_3\_o == 0 ~ 0)) %>%  
 # All outputs needs to be numeric for future addition of AUDIT total  
 mutate(audit\_1 = as.numeric(audit\_1),  
 audit\_2 = as.numeric(audit\_2),  
 audit\_3 = as.numeric(audit\_3),  
 audit\_4 = as.numeric(audit\_4),  
 audit\_5 = as.numeric(audit\_5),  
 audit\_6 = as.numeric(audit\_5),  
 audit\_7 = as.numeric(audit\_7),  
 audit\_8 = as.numeric(audit\_8),  
 audit\_9 = as.numeric(audit\_9),  
 audit\_10 = as.numeric(audit\_10)) %>%  
 # Find audit\_tot -- total AUDIT scores for all participants   
 mutate(audit\_tot = as.numeric(audit\_1) + as.numeric(audit\_2) + as.numeric(audit\_3) +   
 as.numeric(audit\_4) + as.numeric(audit\_5) + as.numeric(audit\_6) + as.numeric(audit\_7) +   
 as.numeric(audit\_8) + as.numeric(audit\_9) + as.numeric(audit\_10))  
table(traindf.v10$audit\_tot) # check output (too small)

##   
## 2 3 4 5 6 7 8 11 12 13 14   
## 173 87 34 8 14 11 5 1 1 1 1

traindf.v10 = subset(traindf.v10, select= -c(audit\_1:audit\_10, audit\_3\_m,   
 audit\_3\_f, audit\_3\_o)) # clean up vars/columns

## ETOH treatment

table(traindf.v10$alc\_tx) # check output (too small)

##   
## 0 1   
## 1940 51

traindf.v10 = subset(traindf.v10, select= -c(alc\_tx)) # clean up vars/columns

## Eating disorder, any type

traindf.v10 <- traindf.v10 %>%  
 filter(!is.na(ed\_any)) # filter out all NAs for complete data  
  
table(traindf.v10$ed\_any, useNA = "always") # check output, confirms no NAs

##   
## 0 1 <NA>   
## 16882 2058 0

## WCS

Combine wcs\_1\_f, wcs\_1\_m, wcs\_1\_o to create ‘wcs\_1’ \*\* too small

# check output (too small)  
table(traindf.v10$wcs\_1\_f)

##   
## 1 2 3 4 5   
## 133 236 428 312 169

table(traindf.v10$wcs\_1\_m)

##   
## 1 2 3 4 5   
## 83 80 187 151 73

table(traindf.v10$wcs\_1\_o)

##   
## 1 2 3 4 5   
## 2 5 8 12 13

table(traindf.v10$wcs\_2)

##   
## 1 2 3 4 5   
## 615 497 433 237 109

table(traindf.v10$wcs\_3)

##   
## 1 2 3 4 5 6 7   
## 1043 201 84 68 61 97 338

table(traindf.v10$wcs\_4)

##   
## 1 2 3 4   
## 648 890 333 19

table(traindf.v10$wcs\_5)

##   
## 1 2 3 4 5   
## 264 373 606 410 237

traindf.v10 = subset(traindf.v10, select= -c(wcs\_1\_f, wcs\_1\_m, wcs\_1\_o,   
 wcs\_2, wcs\_3, wcs\_4, wcs\_5))

## Medication use

# Create stimulant  
traindf.v11 <- traindf.v10 %>%  
 mutate(meds\_sti = case\_when(meds\_1 == 1 ~ 1, # combine stimulant variables  
 meds\_8 == 1 ~ 0,  
 meds\_1 == 0 ~ 0))  
 # Reconcile stimulant write-ins from meds\_7\_text  
 traindf.v11$meds\_sti[str\_detect(traindf.v11$meds\_7\_text, "[V|v]yvan[s|c]e")] <- 1  
 traindf.v11$meds\_sti[str\_detect(traindf.v11$meds\_7\_text, "[T|t]enex")] <- 1  
 traindf.v11$meds\_sti[str\_detect(traindf.v11$meds\_7\_text, "[C|c]oncerta")] <- 1  
 traindf.v11$meds\_sti[str\_detect(traindf.v11$meds\_7\_text, "[A|a]dderall")] <- 1  
 traindf.v11$meds\_sti[str\_detect(traindf.v11$meds\_7\_text, "[R|r]italin")] <- 1  
 traindf.v11$meds\_sti[str\_detect(traindf.v11$meds\_7\_text, "[P|p]rovigil")] <- 1  
 traindf.v11$meds\_sti[str\_detect(traindf.v11$meds\_7\_text, "[A|a]rmodafinil")] <- 1  
 traindf.v11$meds\_sti[str\_detect(traindf.v11$meds\_7\_text, "[F|f]ocalin")] <- 1  
 traindf.v11$meds\_sti[str\_detect(traindf.v11$meds\_7\_text, "[M|m]etadat")] <- 1  
 traindf.v11$meds\_sti[str\_detect(traindf.v11$meds\_7\_text, "[F|f]ocalin")] <- 1  
 traindf.v11$meds\_sti[str\_detect(traindf.v11$meds\_7\_text, "[N|n]uvigil")] <- 1  
 traindf.v11$meds\_sti[str\_detect(traindf.v11$meds\_7\_text, "ADHD")] <- 1  
 traindf.v11$meds\_sti[str\_detect(traindf.v11$meds\_7\_text, "[M|m]odafinil")] <- 1  
 traindf.v11$meds\_sti[traindf.v11$meds\_7\_text %in% c("Vyance", "Vivance adhd", "Stratera", "Aderol", "Aderal", "Adderal",  
 "Adderrall (for ADHD)", "Adderal, vyvance", "Celltech")] <- 1  
   
# Create antidepressant  
traindf.v11 <- traindf.v11 %>%  
 mutate(meds\_dep = case\_when(meds\_2 == 1 ~ 1, # combine depression var types  
 meds\_8 == 1 ~ 0,  
 meds\_2 == 0 ~ 0))  
 # reconcile write-ins for depression from med\_7\_text  
 traindf.v11$meds\_dep[str\_detect(traindf.v11$meds\_7\_text, "[C|c]elexa")] <- 1  
 traindf.v11$meds\_dep[str\_detect(traindf.v11$meds\_7\_text, "[C|c]italopram")] <- 1  
 traindf.v11$meds\_dep[str\_detect(traindf.v11$meds\_7\_text, "[C|c]ymbalta")] <- 1  
 traindf.v11$meds\_dep[str\_detect(traindf.v11$meds\_7\_text, "[E|e]scitalopram")] <- 1  
 traindf.v11$meds\_dep[str\_detect(traindf.v11$meds\_7\_text, "[L|l]exapro")] <- 1  
 traindf.v11$meds\_dep[str\_detect(traindf.v11$meds\_7\_text, "[S|s]ertraline")] <- 1  
 traindf.v11$meds\_dep[str\_detect(traindf.v11$meds\_7\_text, "[Z|z]oloft")] <- 1  
 traindf.v11$meds\_dep[str\_detect(traindf.v11$meds\_7\_text, "SSRI")] <- 1  
 traindf.v11$meds\_dep[str\_detect(traindf.v11$meds\_7\_text, "[E|e]lavil")] <- 1  
 traindf.v11$meds\_dep[str\_detect(traindf.v11$meds\_7\_text, "[P|p]aroxetine")] <- 1  
 traindf.v11$meds\_dep[str\_detect(traindf.v11$meds\_7\_text, "[P|p]rozac")] <- 1  
 traindf.v11$meds\_dep[str\_detect(traindf.v11$meds\_7\_text, "[A|a]ntidepressant")] <- 1  
 traindf.v11$meds\_dep[str\_detect(traindf.v11$meds\_7\_text, "[A|a]mitriptyline")] <- 1  
 traindf.v11$meds\_dep[str\_detect(traindf.v11$meds\_7\_text, "[D|d]uloxetine")] <- 1  
 traindf.v11$meds\_dep[str\_detect(traindf.v11$meds\_7\_text, "[C|c][e|i]pralex")] <- 1  
 traindf.v11$meds\_dep[str\_detect(traindf.v11$meds\_7\_text, "[W|w]ellbutrin")] <- 1  
 traindf.v11$meds\_dep[str\_detect(traindf.v11$meds\_7\_text, "[S|s]trattera")] <- 1   
 traindf.v11$meds\_dep[str\_detect(traindf.v11$meds\_7\_text, "[A|a]tomoxetine")] <- 1  
 traindf.v11$meds\_dep[str\_detect(traindf.v11$meds\_7\_text, "[F|f]luvoxamine")] <- 1  
 traindf.v11$meds\_dep[str\_detect(traindf.v11$meds\_7\_text, "[V|v]enlafaxine")] <- 1  
 traindf.v11$meds\_dep[str\_detect(traindf.v11$meds\_7\_text, "[P|p]ristiq")] <- 1  
 traindf.v11$meds\_dep[str\_detect(traindf.v11$meds\_7\_text, "[D|d]esipramine")] <- 1  
 traindf.v11$meds\_dep[str\_detect(traindf.v11$meds\_7\_text, "[V|v]iibryd")] <- 1  
 traindf.v11$meds\_dep[str\_detect(traindf.v11$meds\_7\_text, "[V|v]enlafaxin")] <- 1  
 traindf.v11$meds\_dep[str\_detect(traindf.v11$meds\_7\_text, "[C|c]lomipramine")] <- 1  
 traindf.v11$meds\_dep[str\_detect(traindf.v11$meds\_7\_text, "[D|d]esvenflaxine")] <- 1  
 traindf.v11$meds\_dep[str\_detect(traindf.v11$meds\_7\_text, "[M|m]irtazapine")] <- 1  
 traindf.v11$meds\_dep[str\_detect(traindf.v11$meds\_7\_text, "[R|r][e|i]meron")] <- 1  
 traindf.v11$meds\_dep[str\_detect(traindf.v11$meds\_7\_text, "[T|t]raz[a|o|i]done")] <- 1  
 traindf.v11$meds\_dep[str\_detect(traindf.v11$meds\_7\_text, "[N|n]ortriptyline")] <- 1  
 traindf.v11$meds\_dep[traindf.v11$meds\_7\_text %in% c("CYMBALTA", "Cymbolta", "Lexipro", "Lexxapro", "Sertaline",  
 "remiron", "Mirtazepene", "bupropin", "Lexapro, Wellbutrin",  
 "Lexapro and Paxil", "Fuoxetine", "Duluxotine",   
 "Forgot the name, anti depressant for anxiety, daily med",   
 "Atomoxitine (nonstimulant ADHD)", "Fluvoximine (luvox)",  
 "Anafranil", "amatryptaline for migraines", "amnitryptaline",   
 "Buproprion (may or may not count as a psychostim?)",   
 "amatryptaline for migraines", "Minipress")] <- 1  
   
# Create antipsychotic  
traindf.v11 <- traindf.v11 %>%  
 mutate(meds\_psy = case\_when(meds\_3 == 1 ~ 1, # combine anti-psychotic med vars  
 meds\_8 == 1 ~ 0,  
 meds\_3 == 0 ~ 0))  
 # Reconcile anti-psychotic entries from med\_7\_text  
 traindf.v11$meds\_psy[str\_detect(traindf.v11$meds\_7\_text, "[A|a]bilify")] <- 1  
 traindf.v11$meds\_psy[str\_detect(traindf.v11$meds\_7\_text, "[A|a]bility")] <- 1  
 traindf.v11$meds\_psy[str\_detect(traindf.v11$meds\_7\_text, "[V|v]raylar")] <- 1  
 traindf.v11$meds\_psy[str\_detect(traindf.v11$meds\_7\_text, "[O|o]lanzapine")] <- 1  
 traindf.v11$meds\_psy[str\_detect(traindf.v11$meds\_7\_text, "[A|a]ripiprazole")] <- 1  
 traindf.v11$meds\_psy[str\_detect(traindf.v11$meds\_7\_text, "[L|l]atuda")] <- 1  
 traindf.v11$meds\_psy[str\_detect(traindf.v11$meds\_7\_text, "[S|s]eroqu[e|o]l")] <- 1  
 traindf.v11$meds\_psy[str\_detect(traindf.v11$meds\_7\_text, "[G|g]eodon")] <- 1  
 traindf.v11$meds\_psy[str\_detect(traindf.v11$meds\_7\_text, "[Q|q]uetiapine")] <- 1  
 traindf.v11$meds\_psy[traindf.v11$meds\_7\_text %in% c("Ziprazidone", "Quitiapin", "Latuda, Giadon")] <- 1   
   
# Create anti-anxiety   
traindf.v11 <- traindf.v11 %>%   
 mutate(meds\_anx = case\_when(meds\_4 == 1 ~ 1, # combine anti-anxiety var types  
 meds\_8 == 1 ~ 0,  
 meds\_4 == 0 ~ 0))  
 # Reconcile all anti-anxiety drugs from the write-ins from meds\_7\_text  
 traindf.v11$meds\_anx[str\_detect(traindf.v11$meds\_7\_text, "[X|x]anax")] <- 1  
 traindf.v11$meds\_anx[str\_detect(traindf.v11$meds\_7\_text, "[V|v]alium")] <- 1  
 traindf.v11$meds\_anx[str\_detect(traindf.v11$meds\_7\_text, "[P|p]ropranolol")] <- 1  
 traindf.v11$meds\_anx[str\_detect(traindf.v11$meds\_7\_text, "[P|p]ropanol")] <- 1  
 traindf.v11$meds\_anx[str\_detect(traindf.v11$meds\_7\_text, "[L|l]orazepam")] <- 1  
 traindf.v11$meds\_anx[str\_detect(traindf.v11$meds\_7\_text, "[N|n]eurontin")] <- 1  
 traindf.v11$meds\_anx[str\_detect(traindf.v11$meds\_7\_text, "[A|a]tarax")] <- 1  
 traindf.v11$meds\_anx[str\_detect(traindf.v11$meds\_7\_text, "[L|l]yrica")] <- 1  
 traindf.v11$meds\_anx[str\_detect(traindf.v11$meds\_7\_text, "[A|a]tivan")] <- 1  
 traindf.v11$meds\_anx[str\_detect(traindf.v11$meds\_7\_text, "[C|c]lonidine")] <- 1  
 traindf.v11$meds\_anx[str\_detect(traindf.v11$meds\_7\_text, "[P|p]razosin")] <- 1  
 traindf.v11$meds\_anx[str\_detect(traindf.v11$meds\_7\_text, "[B|b]uspar")] <- 1  
 traindf.v11$meds\_anx[str\_detect(traindf.v11$meds\_7\_text, "[V|v]istaril")] <- 1 # antihistamine  
 traindf.v11$meds\_anx[str\_detect(traindf.v11$meds\_7\_text, "[I|i]nderal")] <- 1  
 traindf.v11$meds\_anx[str\_detect(traindf.v11$meds\_7\_text, "[H|h]ydroxyzine")] <- 1 # antihistamine  
 traindf.v11$meds\_anx[str\_detect(traindf.v11$meds\_7\_text, "[K|k]lonopin")] <- 1  
 traindf.v11$meds\_anx[str\_detect(traindf.v11$meds\_7\_text, "[D|d]iazepam")] <- 1  
 traindf.v11$meds\_anx[str\_detect(traindf.v11$meds\_7\_text, "[B|b]uspirone")] <- 1  
 traindf.v11$meds\_anx[str\_detect(traindf.v11$meds\_7\_text, "[G|g]abapentin")] <- 1  
 traindf.v11$meds\_anx[traindf.v11$meds\_7\_text %in% c("Valuim", "you didnt list vallium under anti-anxiety", "Alphrazolam",   
 "Hydroxizine--prescribed as needed for anxiety, only took one because of adverse reaction",   
 "Hydrodoxine, a blood pressure medicine for anxiety that I don't remember the name of")] <- 1  
  
# Create mood stabilizer  
traindf.v11 <- traindf.v11 %>%  
 mutate(meds\_mood = case\_when(meds\_5 == 1 ~ 1, # combine mood stabilizer vars  
 meds\_8 == 1 ~ 0,  
 meds\_5 == 0 ~ 0))  
 # Reconcile mood stabilizers from write-in in med\_7\_text  
 traindf.v11$meds\_mood[str\_detect(traindf.v11$meds\_7\_text, "[L|l]ithium")] <- 1  
 traindf.v11$meds\_mood[str\_detect(traindf.v11$meds\_7\_text, "[L|l]amictal")] <- 1  
 traindf.v11$meds\_mood[str\_detect(traindf.v11$meds\_7\_text, "[L|l]amotrigine")] <- 1  
 traindf.v11$meds\_mood[str\_detect(traindf.v11$meds\_7\_text, "[D|d]epakote")] <- 1  
 traindf.v11$meds\_mood[str\_detect(traindf.v11$meds\_7\_text, "[T|t]rileptal")] <- 1  
 traindf.v11$meds\_mood[str\_detect(traindf.v11$meds\_7\_text, "[O|o]xcarbarzepine")] <- 1  
 traindf.v11$meds\_mood[traindf.v11$meds\_7\_text %in% c("Mood Stabilizer/Birth Control")] <- 1  
   
# Create sleep med  
traindf.v11 <- traindf.v11 %>%  
 mutate(meds\_sle = case\_when(meds\_6 == 1 ~ 1, # combine sleep meds vars  
 meds\_8 == 1 ~ 0,  
 meds\_6 == 0 ~ 0))  
 # Reconcile sleep medications from write-ins in meds\_7\_text  
 traindf.v11$meds\_sle[str\_detect(traindf.v11$meds\_7\_text, "[X|x]yrem")] <- 1  
 traindf.v11$meds\_sle[str\_detect(traindf.v11$meds\_7\_text, "[S|s]leep")] <- 1  
 traindf.v11$meds\_sle[str\_detect(traindf.v11$meds\_7\_text, "[M|m]elatonin")] <- 1  
 traindf.v11$meds\_sle[str\_detect(traindf.v11$meds\_7\_text, "[Z|z]quil")] <- 1  
 traindf.v11$meds\_sle[str\_detect(traindf.v11$meds\_7\_text, "[N|n]yquil")] <- 1  
 traindf.v11$meds\_sle[str\_detect(traindf.v11$meds\_7\_text, "[D|d]iphenhydramine")] <- 1  
 traindf.v11$meds\_sle[traindf.v11$meds\_7\_text %in% c("gabapentin, trazodone (both for sleep on separate occasions)",   
 "generic sleep medication: melatonin supplements",   
 "cold medicine to induce sleep")] <- 1  
   
# Create "Othermeds  
traindf.v11 <- traindf.v11 %>%  
 mutate(meds\_oth = case\_when(meds\_7 == 1 ~ 1, # combine other meds vars  
 meds\_8 == 1 ~ 0,  
 meds\_7 == 0 ~ 0))   
 # Reconcile "other" entries from med\_7\_text  
 traindf.v11$meds\_oth[str\_detect(traindf.v11$meds\_7\_text, "[C|c]annabis")] <- 1  
 traindf.v11$meds\_oth[str\_detect(traindf.v11$meds\_7\_text, "[M|m]arijuana")] <- 1  
 traindf.v11$meds\_oth[str\_detect(traindf.v11$meds\_7\_text, "[W|w]eed")] <- 1  
 traindf.v11$meds\_oth[str\_detect(traindf.v11$meds\_7\_text, "[N|n]altrexone")] <- 1  
 traindf.v11$meds\_oth[str\_detect(traindf.v11$meds\_7\_text, "[S|s]ynthroid")] <- 1  
 traindf.v11$meds\_oth[str\_detect(traindf.v11$meds\_7\_text, "[T|t]opomax")] <- 1  
 traindf.v11$meds\_oth[str\_detect(traindf.v11$meds\_7\_text, "[T|t]estosterone")] <- 1  
 traindf.v11$meds\_oth[str\_detect(traindf.v11$meds\_7\_text, "[T|t]hyroid")] <- 1  
 traindf.v11$meds\_oth[str\_detect(traindf.v11$meds\_7\_text, "[H|h]ydrodocone")] <- 1   
 traindf.v11$meds\_oth[str\_detect(traindf.v11$meds\_7\_text, "[C|c]lonidine")] <- 1  
 traindf.v11$meds\_oth[str\_detect(traindf.v11$meds\_7\_text, "[G|g]uanfacine")] <- 1  
 traindf.v11$meds\_oth[str\_detect(traindf.v11$meds\_7\_text, "[K|k]etamine")] <- 1  
 traindf.v11$meds\_oth[str\_detect(traindf.v11$meds\_7\_text, "[F|f]lexeril")] <- 1  
 traindf.v11$meds\_oth[str\_detect(traindf.v11$meds\_7\_text, "[P|p]ant[o|a]loc")] <- 1  
 traindf.v11$meds\_oth[str\_detect(traindf.v11$meds\_7\_text, "[S|s]pironolactone")] <- 1  
 traindf.v11$meds\_oth[str\_detect(traindf.v11$meds\_7\_text, "[H|h]ormone")] <- 1  
 traindf.v11$meds\_oth[str\_detect(traindf.v11$meds\_7\_text, "[O|o]xycodone")] <- 1  
 traindf.v11$meds\_oth[str\_detect(traindf.v11$meds\_7\_text, "[C|c]annabic")] <- 1  
 traindf.v11$meds\_oth[str\_detect(traindf.v11$meds\_7\_text, "[B|b]eta")] <- 1  
 traindf.v11$meds\_oth[traindf.v11$meds\_7\_text %in% c("Canabis", "Vistiril", "Vistral", "Thyroxine", "The Reefer",  
 "Topimitrate", "topiramate", "Estradiol, Spironactone",   
 "muscle relaxants to help me sleep", "Ondansetron" ,  
 "Sumatriptan for migraines", "Psilocybin Mushrooms", "clonodine",  
 "Muscle Relaxers", "anti-siezure medication", "Codeine",   
 "Androgen Blockers", "Anti-seizure to control racing thoughts",  
 "Anti seizure", "Buprenorphine (Suboxone) for opiate dependence",  
 "Hydroxine", "Hydroxcyzine", "Alyacen", "Mirapex",   
 "hypothyroid medication", "Epilepsy drugs", "Birth control",  
 "Methadone", "Vimpat", "Contrave", "Benztropine", "Intuniv",  
 "Medicinal marijuana", "Pantaloc, for psychogenic vomiting",  
 "medicinal Cannabis", "Testosterone Enanthate", "testosterone",  
 "Cannabis", "Medical marijuana")] <- 1  
   
# Create a group for students who do not take any psychotropic meds  
traindf.v11 <- traindf.v11 %>%  
 mutate(meds\_none = case\_when(meds\_none == 1 ~ 1, # filter for all meds w/ none  
 meds\_none == 0 ~ 0,  
 meds\_sti == 1 ~ 0,  
 meds\_dep == 1 ~ 0,  
 meds\_psy == 1 ~ 0,  
 meds\_anx == 1 ~ 0,  
 meds\_mood == 1 ~ 0,  
 meds\_sle == 1 ~ 0,  
 meds\_oth == 1 ~ 0)) %>%  
 filter(!is.na(meds\_any))  
   
# tabulate and check outputs  
table(traindf.v11$meds\_sti)

##   
## 0 1   
## 14262 1159

table(traindf.v11$meds\_dep)

##   
## 0 1   
## 14262 2663

table(traindf.v11$meds\_psy)

##   
## 0 1   
## 14262 161

table(traindf.v11$meds\_anx)

##   
## 0 1   
## 14262 1426

table(traindf.v11$meds\_mood)

##   
## 0 1   
## 14262 343

table(traindf.v11$meds\_sle)

##   
## 0 1   
## 14262 656

table(traindf.v11$meds\_oth)

##   
## 0 1   
## 14262 302

table(traindf.v11$meds\_none)

##   
## 0 1   
## 4019 14262

table(traindf.v11$meds\_any, useNA = "always")

##   
## 0 1 <NA>   
## 14262 4019 0

## Psychopharm medication class count

# convert all med vars to numeric for future medcount var totaling  
traindf.v11$meds\_1 <- as.numeric(traindf.v11$meds\_1)  
traindf.v11$meds\_2 <- as.numeric(traindf.v11$meds\_2)  
traindf.v11$meds\_3 <- as.numeric(traindf.v11$meds\_3)  
traindf.v11$meds\_4 <- as.numeric(traindf.v11$meds\_4)  
traindf.v11$meds\_5 <- as.numeric(traindf.v11$meds\_5)  
traindf.v11$meds\_6 <- as.numeric(traindf.v11$meds\_6)  
traindf.v11$meds\_7 <- as.numeric(traindf.v11$meds\_7)  
traindf.v11$meds\_8 <- as.numeric(traindf.v11$meds\_8)  
  
# set any NAs equal to zero  
traindf.v11$meds\_1[is.na(traindf.v11$meds\_1)] <- 0  
traindf.v11$meds\_2[is.na(traindf.v11$meds\_2)] <- 0  
traindf.v11$meds\_3[is.na(traindf.v11$meds\_3)] <- 0  
traindf.v11$meds\_4[is.na(traindf.v11$meds\_4)] <- 0  
traindf.v11$meds\_5[is.na(traindf.v11$meds\_5)] <- 0  
traindf.v11$meds\_6[is.na(traindf.v11$meds\_6)] <- 0  
traindf.v11$meds\_7[is.na(traindf.v11$meds\_7)] <- 0  
  
# create meds\_count var  
traindf.v11 <- traindf.v11 %>%  
 mutate(meds\_count = meds\_1 + meds\_2 + meds\_3 + meds\_4 + meds\_5 + meds\_6 + meds\_7) %>%  
 mutate(meds\_count = case\_when(meds\_count == 0 ~ 0,  
 meds\_count == 1 ~ 1,  
 meds\_count == 2 ~ 2,   
 meds\_count == 3 ~ 3,  
 meds\_count == 4 ~ 4,  
 meds\_count == 5 ~ 5,  
 meds\_count == 6 ~ 6))  
traindf.v11$meds\_count <- as.numeric(traindf.v11$meds\_count) # return var to numeric cass  
  
table(traindf.v11$meds\_count, useNA = "always") # check output

##   
## 0 1 2 3 4 5 6 <NA>   
## 14262 2339 1063 428 136 46 7 0

traindf.v11 = subset(traindf.v11, select= -c(meds\_1:meds\_7)) # clean up vars/columns

traindf.v13 <- traindf.v11

## US resident status

# aggregate all residency variables into 1 variable called 'res\_status'  
traindf.v13 <- traindf.v13 %>%   
 mutate(res\_status = case\_when(citizen == 1 ~ 1,  
 permanentres == 1 ~ 2,   
 visa == 1 ~ 3,  
 undoc == 1 ~ 4,  
 otherdoc == 1 ~ 5))  
table(traindf.v13$res\_status) # check output

##   
## 1 2 3 4 5   
## 353 8 49 1 1

# data.v13 = subset(traindf.v12, select= -c(citizen, permanentres, visa, otherdoc, undoc)) # clean up var

## Discrimination

traindf.v13$discrim<- as.numeric(traindf.v13$discrim)  
table(traindf.v13$discrim, useNA = "always") # check output (lots of NAs)

##   
## 1 2 3 4 5 6 <NA>   
## 6118 3253 1432 349 86 53 6990

## Religion

# Label levels of religious importance  
traindf.v13 <- traindf.v13 %>%  
 # flip coding  
 mutate(religios = case\_when(religios == 1 ~ 5, # fhighly religious  
 religios == 2 ~ 4,  
 religios == 3 ~ 3,  
 religios == 4 ~ 2,  
 religios == 5 ~ 1)) %>% # not at all religious  
 filter(!is.na(religios))  
traindf.v13$religios <- as.numeric(traindf.v13$religios)  
  
table(traindf.v13$religios, useNA = "always") # Check output

##   
## 1 2 3 4 5 <NA>   
## 3704 3537 4640 3676 2714 0

## Residence type

# re-organize living situation into on-campus, off-campus, and at home  
traindf.v13 <- traindf.v13 %>%  
 mutate(residenc = case\_when(residenc == 1 ~ 1, # on-campus  
 residenc == 2 ~ 1,  
 residenc == 3 ~ 2, # coop  
 residenc == 4 ~ 2,   
 residenc == 5 ~ 3, # off-campus  
 residenc == 6 ~ 4, # with family  
 residenc == 7 ~ 3)) %>%  
 filter(!is.na(residenc))  
  
table(traindf.v13$residenc, useNA = "always") # check output

##   
## 1 2 3 4 <NA>   
## 7840 498 7620 2309 0

## MH campus environment

traindf.v13$env\_mh <- as.factor(traindf.v13$env\_mh)  
table(traindf.v13$env\_mh, useNA = "always") # check output (excessive NAs)

##   
## 1 2 3 4 5 6 <NA>   
## 1144 1555 2569 2620 2743 634 7002

## Insurance coverage

# create one insurance var from related questions in 'ins\_' module  
traindf.v13 <- traindf.v13 %>%   
 mutate(ins\_no = case\_when((ins\_1 == 1 | ins\_mh == 4 |   
 ins\_mh == 5 | ins\_ade == 3) ~ 1, # inadequate coverage  
 (ins\_1 == 0 | ins\_mh == 1 |   
 ins\_mh == 2 | ins\_ade == 2) ~ 0)) %>% # adequate MH coverage  
 filter(!is.na(ins\_no))   
traindf.v13$ins\_no <- as.logical(traindf.v13$ins\_no) # switch class to boolean  
table(traindf.v13$ins\_no, useNA = "always") # check output, skew

##   
## FALSE TRUE <NA>   
## 10240 3025 0

jarque.bera.test(traindf.v13$ins\_no)

##   
## Jarque Bera Test  
##   
## data: traindf.v13$ins\_no  
## X-squared = 3771.8, df = 2, p-value < 2.2e-16

traindf.v13 = subset(traindf.v13, select= -c(ins\_1, ins\_mh, ins\_ade, ins\_inade)) # clean up var

## Military status

traindf.v14 <- traindf.v13  
table(traindf.v14$military, useNA = "always")

##   
## 0 1 <NA>   
## 13129 128 8

## International student

traindf.v14 <- traindf.v14 %>%  
 mutate(international = case\_when(international == 1 ~ 1,  
 international == 0 ~ 0)) %>%  
 filter(!is.na(international))  
  
traindf.v14$international <- as.logical(traindf.v14$international)  
  
table(traindf.v14$international, useNA = "always")

##   
## FALSE TRUE <NA>   
## 12229 1030 0

## Family support

traindf.v14$fam\_support\_aca <- as.numeric(traindf.v14$fam\_support\_aca)  
table(traindf.v14$fam\_support\_aca, useNA = "always") # check output (excessive NAs)

##   
## 1 2 3 4 5 6 <NA>   
## 977 612 190 56 24 17 11383

## Activity Involvement

# set all NAs in activities equal to 0  
traindf.v14$activ\_ac[is.na(traindf.v14$activ\_ac)] <- 1  
traindf.v14$activ\_athc[is.na(traindf.v14$activ\_athc)] <- 1  
traindf.v14$activ\_athv[is.na(traindf.v14$activ\_athv)] <- 1  
traindf.v14$activ\_athi[is.na(traindf.v14$activ\_athi)] <- 1  
traindf.v14$activ\_cs[is.na(traindf.v14$activ\_cs)] <- 1  
traindf.v14$activ\_cu[is.na(traindf.v14$activ\_cu)] <- 1  
traindf.v14$activ\_da[is.na(traindf.v14$activ\_da)] <- 1  
traindf.v14$activ\_fs[is.na(traindf.v14$activ\_fs)] <- 1  
traindf.v14$activ\_gs[is.na(traindf.v14$activ\_gs)] <- 1  
traindf.v14$activ\_gov[is.na(traindf.v14$activ\_gov)] <- 1  
traindf.v14$activ\_mp[is.na(traindf.v14$activ\_md)] <- 1  
traindf.v14$activ\_md[is.na(traindf.v14$activ\_none)] <- 1  
traindf.v14$activ\_soc[is.na(traindf.v14$activ\_soc)] <- 1  
traindf.v14$activ\_rel[is.na(traindf.v14$activ\_rel)] <- 1  
traindf.v14$activ\_art[is.na(traindf.v14$activ\_art)] <- 1  
traindf.v14$activ\_other[is.na(traindf.v14$activ\_other)] <- 1  
  
# Create one activity variable (dichotomous) based on involvement in campus activities  
traindf.v14 <- traindf.v14 %>%   
 mutate(activ\_no = case\_when(activ\_none == 1 ~ 1,  
 activ\_ac == 1 ~ 0,  
 activ\_athc == 1 ~ 0,  
 activ\_athv == 1 ~ 0,  
 activ\_athi == 1 ~ 0,  
 activ\_cs == 1 ~ 0,  
 activ\_cu == 1 ~ 0,  
 activ\_da == 1 ~ 0,  
 activ\_fs == 1 ~ 0,  
 activ\_gs == 1 ~ 0,  
 activ\_gov == 1 ~ 0,  
 activ\_mp == 1 ~ 0,  
 activ\_md == 1 ~ 0,  
 activ\_rel == 1 ~ 0,  
 activ\_soc == 1 ~ 0,  
 activ\_art == 1 ~ 0,  
 activ\_other == 1 ~ 0))  
traindf.v14$activ\_no <- as.logical(traindf.v14$activ\_no)  
  
table(traindf.v14$activ\_no, useNA = "always") # tabulate and check output

##   
## FALSE TRUE <NA>   
## 9205 4054 0

jarque.bera.test(traindf.v14$activ\_no)

##   
## Jarque Bera Test  
##   
## data: traindf.v14$activ\_no  
## X-squared = 2489.1, df = 2, p-value < 2.2e-16

traindf.v14 = subset(traindf.v14, select= -c(activ\_none, activ\_ac:activ\_other)) # clean up vars

## School variable

# insert each school as a dichotomous var  
traindf.v14 <- traindf.v14 %>%  
 mutate(var\_AlbertaCAD = ifelse((school2 == "AlbertaCAD"), 1, 0)) %>%  
 mutate(var\_Boston = ifelse((school2 == "Boston"), 1, 0)) %>%  
 mutate(var\_CIA = ifelse((school2 == "CIA"), 1, 0)) %>%  
 mutate(var\_CalPoly = ifelse((school2 == "CalPoly"), 1, 0)) %>%  
 mutate(var\_ColoradoCollege = ifelse((school2 == "ColoradoCollege"), 1, 0)) %>%  
 mutate(var\_ColoradoMountain = ifelse((school2 == "ColoradoMountain"), 1, 0)) %>%  
 mutate(var\_ColumbusCAD = ifelse((school2 == "ColumbusCAD"), 1, 0)) %>%  
 mutate(var\_Delaware = ifelse((school2 == "Delaware"), 1, 0)) %>%  
 mutate(var\_DetroitMercy = ifelse((school2 == "DetroitMercy"), 1, 0)) %>%  
 mutate(var\_Emerson = ifelse((school2 == "Emerson"), 1, 0)) %>%  
 mutate(var\_GeorgeMason = ifelse((school2 == "GeorgeMason"), 1, 0)) %>%  
 mutate(var\_GeorgiaTech = ifelse((school2 == "GeorgiaTech"), 1, 0)) %>%  
 mutate(var\_Kalamazoo = ifelse((school2 == "Kalamazoo"), 1, 0)) %>%  
 mutate(var\_Kansas = ifelse((school2 == "Kansas"), 1, 0)) %>%  
 mutate(var\_MDInstituteCollegeArt = ifelse((school2 == "MDInstituteCollegeArt"), 1, 0)) %>%  
 mutate(var\_MassArt = ifelse((school2 == "MassArt"), 1, 0)) %>%  
 mutate(var\_Merrimack = ifelse((school2 == "Merrimack"), 1, 0)) %>%  
 mutate(var\_MichiganState = ifelse((school2 == "MichiganState"), 1, 0)) %>%  
 mutate(var\_MinnesotaCAD = ifelse((school2 == "MinnesotaCAD"), 1, 0)) %>%  
 mutate(var\_MontclairState = ifelse((school2 == "MontclairState"), 1, 0)) %>%  
 mutate(var\_NHInstituteArt = ifelse((school2 == "NHInstituteArt"), 1, 0)) %>%  
 mutate(var\_NevadaReno = ifelse((school2 == "NevadaReno"), 1, 0)) %>%  
 mutate(var\_OaklandCC = ifelse((school2 == "OaklandCC"), 1, 0)) %>%  
 mutate(var\_OklahomaCityCC = ifelse((school2 == "OklahomaCityCC"), 1, 0)) %>%  
 mutate(var\_PaloAltoCollege = ifelse((school2 == "PaloAltoCollege"), 1, 0)) %>%  
 mutate(var\_PennCollegeTech = ifelse((school2 == "PennCollegeTech"), 1, 0)) %>%  
 mutate(var\_Pratt = ifelse((school2 == "RedlandsCC"), 1, 0)) %>%  
 mutate(var\_RhodeIsland = ifelse((school2 == "RhodeIsland"), 1, 0)) %>%  
 mutate(var\_RedlandsCC = ifelse((school2 == "RedlandsCC"), 1, 0)) %>%  
 mutate(var\_Ringling = ifelse((school2 == "Ringling"), 1, 0)) %>%  
 mutate(var\_Rollins = ifelse((school2 == "Rollins"), 1, 0)) %>%  
 mutate(var\_SAIC = ifelse((school2 == "SAIC"), 1, 0)) %>%  
 mutate(var\_Sewanee = ifelse((school2 == "Sewanee"), 1, 0)) %>%  
 mutate(var\_Smith = ifelse((school2 == "Smith"), 1, 0)) %>%  
 mutate(var\_SouthernNazarene = ifelse((school2 == "SouthernNazarene"), 1, 0)) %>%  
 mutate(var\_SouthwesternOklahomaState = ifelse((school2 == "SouthwesternOklahomaState"), 1, 0)) %>%  
 mutate(var\_StJohns = ifelse((school2 == "StJohns"), 1, 0)) %>%  
 mutate(var\_StMarys = ifelse((school2 == "StMarys"), 1, 0)) %>%  
 mutate(var\_StRose = ifelse((school2 == "StRose"), 1, 0)) %>%  
 mutate(var\_TrumanState = ifelse((school2 == "TrumanState"), 1, 0)) %>%  
 mutate(var\_TuftsHealthSciences = ifelse((school2 == "TuftsHealthSciences"), 1, 0)) %>%  
 mutate(var\_TuftsMainCampus = ifelse((school2 == "TuftsMainCampus"), 1, 0)) %>%  
 mutate(var\_UChicago = ifelse((school2 == "UChicago"), 1, 0)) %>%  
 mutate(var\_UFlorida = ifelse((school2 == "UFlorida"), 1, 0)) %>%  
 mutate(var\_UMDearborn = ifelse((school2 == "UMDearborn"), 1, 0)) %>%  
 mutate(var\_UMich = ifelse((school2 == "UMich"), 1, 0)) %>%  
 mutate(var\_UNCSchoolArt = ifelse((school2 == "UNCSchoolArt"), 1, 0)) %>%  
 mutate(var\_UTKnoxville = ifelse((school2 == "UTKnoxville"), 1, 0)) %>%  
 mutate(var\_UniversityofSouthernCalifornia = ifelse((school2 == "UniversityofSouthernCalifornia"), 1, 0)) %>%   
 mutate(var\_Utah = ifelse((school2 == "Utah"), 1, 0)) %>%  
 mutate(var\_VATech = ifelse((school2 == "VATech"), 1, 0)) %>%  
 mutate(var\_VirginiaCommonwealth = ifelse((school2 == "VirginiaCommonwealth"), 1, 0)) %>%  
 mutate(var\_WakeForest = ifelse((school2 == "WakeForest"), 1, 0)) %>%  
 mutate(var\_Watkins = ifelse((school2 == "Watkins"), 1, 0)) %>%  
 mutate(var\_WestVirginia = ifelse((school2 == "WestVirginia"), 1, 0)) %>%  
 mutate(var\_WesternCarolina = ifelse((school2 == "WesternCarolina"), 1, 0)) %>%  
 mutate(var\_WesternMichigan = ifelse((school2 == "WesternMichigan"), 1, 0)) %>%  
 mutate(var\_WesternWA = ifelse((school2 == "WesternWA"), 1, 0)) %>%  
 mutate(var\_Xavier = ifelse((school2 == "Xavier"), 1, 0))

### School type levels

1. 4-year: var\_Boston, var\_CalPoly, var\_ColoradoCollege, var\_ColoradoMountain, var\_Delaware, var\_DetroitMercy, var\_Emerson, var\_GeorgeMason, var\_GeorgiaTech, var\_Kalamazoo, var\_Kansas, var\_Merrimack, var\_MichiganState, var\_MontclairState, var\_NevadaReno, var\_PennCollegeTech, var\_RhodeIsland, var\_Rollins, var\_Sewanee, var\_Smith, var\_SouthernNazarene, var\_SouthwesternOklahomaState, var\_StJohns, var\_StMarys, var\_StRose, var\_TrumanState, var\_TuftsHealthSciences, var\_TuftsMainCampus, var\_UChicago, var\_UFlorida, var\_UMDearborn, var\_UMich, var\_UTKnoxville, var\_UniversityofSouthernCalifornia, var\_Utah, var\_VATech, var\_VirginiaCommonwealth, var\_WakeForest,var\_WestVirginia, var\_WesternCarolina, var\_WesternMichigan, var\_WesternWA, var\_Xavier
2. Community College: var\_OaklandCC, var\_OklahomaCityCC, var\_PaloAltoCollege, var\_RedlandsCC
3. Art & Design, Culinary: var\_AlbertaCAD, var\_CIA, var\_ColumbusCAD, var\_MDInstituteCollegeArt, var\_MassArt, var\_MinnesotaCAD, var\_NHInstituteArt, var\_Pratt, , var\_Ringling, var\_SAIC, var\_UNCSchoolArt, var\_Watkins,

traindf.v14 <- traindf.v14 %>%  
 # 4-year college  
 mutate(school2\_4yr = ifelse(var\_Boston + var\_CalPoly + var\_ColoradoCollege + var\_ColoradoMountain + var\_Delaware +  
 var\_DetroitMercy + var\_Emerson + var\_GeorgeMason + var\_GeorgiaTech + var\_Kalamazoo +   
 var\_Kansas + var\_Merrimack + var\_MichiganState + var\_MontclairState + var\_NevadaReno +  
 var\_PennCollegeTech + var\_RhodeIsland + var\_Rollins + var\_Sewanee + var\_Smith +  
 var\_SouthernNazarene + var\_SouthwesternOklahomaState + var\_StJohns + var\_StMarys + var\_StRose +  
 var\_TrumanState + var\_TuftsHealthSciences + var\_TuftsMainCampus + var\_UChicago + var\_UFlorida +  
 var\_UMDearborn + var\_UMich + var\_UTKnoxville + var\_UniversityofSouthernCalifornia + var\_Utah +  
 var\_VATech + var\_VirginiaCommonwealth + var\_WakeForest + var\_WestVirginia +   
 var\_WesternCarolina + var\_WesternMichigan + var\_WesternWA + var\_Xavier == 1, TRUE,  
 ifelse(  
 var\_Boston + var\_CalPoly + var\_ColoradoCollege + var\_ColoradoMountain + var\_Delaware +  
 var\_DetroitMercy + var\_Emerson + var\_GeorgeMason + var\_GeorgiaTech + var\_Kalamazoo +  
 var\_Kansas + var\_Merrimack + var\_MichiganState + var\_MontclairState + var\_NevadaReno +  
 var\_PennCollegeTech + var\_RhodeIsland + var\_Rollins + var\_Sewanee + var\_Smith +  
 var\_SouthernNazarene + var\_SouthwesternOklahomaState + var\_StJohns + var\_StMarys +   
 var\_StRose + var\_TrumanState + var\_TuftsHealthSciences + var\_TuftsMainCampus + var\_UChicago +  
 var\_UFlorida + var\_UMDearborn + var\_UMich + var\_UTKnoxville +  
 var\_UniversityofSouthernCalifornia + var\_Utah + var\_VATech + var\_VirginiaCommonwealth +  
 var\_WakeForest + var\_WestVirginia + var\_WesternCarolina + var\_WesternMichigan +  
 var\_WesternWA + var\_Xavier == 0, FALSE, NA))) %>%  
   
 # Community colleges  
 mutate(school2\_cc = ifelse(var\_OaklandCC + var\_OklahomaCityCC + var\_PaloAltoCollege + var\_RedlandsCC == 1, TRUE,   
 ifelse(var\_OaklandCC + var\_OklahomaCityCC + var\_PaloAltoCollege + var\_RedlandsCC == 0, FALSE, NA))) %>%  
   
 # Art, design, culinary  
 mutate(school2\_ad = ifelse(var\_AlbertaCAD + var\_CIA + var\_ColumbusCAD + var\_MDInstituteCollegeArt +   
 var\_MassArt + var\_MinnesotaCAD + var\_NHInstituteArt + var\_Pratt + var\_Ringling +  
 var\_SAIC + var\_UNCSchoolArt + var\_Watkins == 1, TRUE,   
 ifelse(  
 var\_AlbertaCAD + var\_CIA + var\_ColumbusCAD + var\_MDInstituteCollegeArt +  
 var\_MassArt + var\_MinnesotaCAD + var\_NHInstituteArt + var\_Pratt +   
 var\_Ringling + var\_SAIC + var\_UNCSchoolArt + var\_Watkins == 0, FALSE, NA)))  
  
# Binary variable for school type  
table(traindf.v14$school2\_4yr)

##   
## FALSE TRUE   
## 2212 11047

table(traindf.v14$school2\_cc)

##   
## FALSE TRUE   
## 12951 308

table(traindf.v14$school2\_ad)

##   
## FALSE TRUE   
## 11735 1524

# create one school type variables  
traindf.v14 <- traindf.v14 %>%  
 mutate(school2\_type = case\_when(school2\_4yr == TRUE ~ 1,  
 school2\_cc == TRUE ~ 2,  
 school2\_ad == TRUE ~ 3)) %>%  
 filter(!is.na(school2\_type))  
traindf.v14$school2\_type <- as.factor(traindf.v14$school2\_type)  
  
table(traindf.v14$school2\_type, useNA = "always") # check output

##   
## 1 2 3 <NA>   
## 11047 308 1499 0

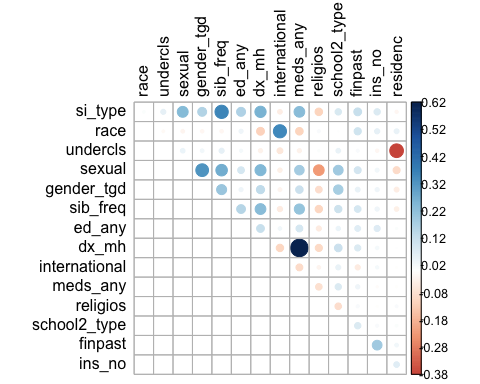
## PREMOTIVATION ANALYSES

traindf.v15 <- traindf.v14 # new dataframe for analysis

Premotivation correlation

# Define diathesis vars for correlation matrix  
traindf\_prem <- traindf.v15 %>% select(si\_type, race, undercls, sexual, gender\_tgd, sib\_freq, ed\_any,   
 dx\_mh, international, meds\_any, religios, school2\_type, finpast,  
 ins\_no, residenc)  
  
# Corr\_simple function  
corr\_simple <- function(traindf=traindf\_prem, sig=0.6){  
 # convert traindf to numeric in order to run correlations   
 traindf\_prem <- traindf\_prem %>% mutate\_if(is.character, as.factor)  
 traindf\_prem <- traindf\_prem %>% mutate\_if(is.factor, as.numeric)  
 # run a correlation and drop the insignificant ones  
 corr <- cor(traindf\_prem, use = "pairwise.complete.obs")   
 corr[lower.tri(corr,diag=TRUE)] <- NA # prepare to drop duplicates and correlations of 1   
 corr[corr %in% c(-1, 1)] <- NA # drop perfect correlations  
 corr <- as.data.frame(as.table(corr)) # turn into a 3-column table  
 corr <- na.omit(corr) # remove the NA values from above   
 corr <- corr[order(-abs(corr$Freq)),] # sort by highest correlation  
 print(corr)  
   
 #turn corr back into matrix in order to plot with corrplot  
 mtx\_corr <- acast(corr, Var1~Var2, value.var="Freq")   
 corrplot(mtx\_corr, is.corr=FALSE, tl.col="black", na.label=" ")  
}  
corr\_simple()

## Var1 Var2 Freq  
## 143 dx\_mh meds\_any 0.6189305058  
## 213 undercls residenc -0.3781576467  
## 76 si\_type sib\_freq 0.3551123237  
## 122 race international 0.3474962707  
## 64 sexual gender\_tgd 0.3264915626  
## 79 sexual sib\_freq 0.2673228293  
## 106 si\_type dx\_mh 0.2538046594  
## 109 sexual dx\_mh 0.2438397610  
## 46 si\_type sexual 0.2373972743  
## 111 sib\_freq dx\_mh 0.2327190537  
## 154 sexual religios -0.2296021106  
## 136 si\_type meds\_any 0.2295440185  
## 141 sib\_freq meds\_any 0.2078425618  
## 80 gender\_tgd sib\_freq 0.2015738967  
## 169 sexual school2\_type 0.1903692807  
## 208 finpast ins\_no 0.1900443785  
## 139 sexual meds\_any 0.1829634514  
## 170 gender\_tgd school2\_type 0.1775903463  
## 91 si\_type ed\_any 0.1609680626  
## 61 si\_type gender\_tgd 0.1589562596  
## 96 sib\_freq ed\_any 0.1545923209  
## 110 gender\_tgd dx\_mh 0.1330741558  
## 107 race dx\_mh -0.1231812198  
## 181 si\_type finpast 0.1214865510  
## 112 ed\_any dx\_mh 0.1210466228  
## 173 dx\_mh school2\_type 0.1157533800  
## 137 race meds\_any -0.1141861671  
## 151 si\_type religios -0.1130466111  
## 140 gender\_tgd meds\_any 0.1122314952  
## 128 dx\_mh international -0.1095249525  
## 156 sib\_freq religios -0.1087331827  
## 171 sib\_freq school2\_type 0.1055579224  
## 182 race finpast 0.1033872081  
## 158 dx\_mh religios -0.1020669707  
## 214 sexual residenc -0.0979670803  
## 184 sexual finpast 0.0951329227  
## 144 international meds\_any -0.0942171333  
## 142 ed\_any meds\_any 0.0923693129  
## 94 sexual ed\_any 0.0914285403  
## 155 gender\_tgd religios -0.0888308404  
## 186 sib\_freq finpast 0.0880701739  
## 176 religios school2\_type -0.0868917562  
## 160 meds\_any religios -0.0857879324  
## 166 si\_type school2\_type 0.0812666337  
## 192 school2\_type finpast 0.0802220310  
## 196 si\_type ins\_no 0.0800422434  
## 175 meds\_any school2\_type 0.0790316740  
## 188 dx\_mh finpast 0.0749999143  
## 202 ed\_any ins\_no 0.0728121115  
## 187 ed\_any finpast 0.0679112569  
## 123 undercls international -0.0674724946  
## 224 ins\_no residenc 0.0637543899  
## 126 sib\_freq international -0.0544589475  
## 197 race ins\_no 0.0529532945  
## 189 international finpast -0.0521256358  
## 157 ed\_any religios -0.0514063283  
## 78 undercls sib\_freq 0.0506262594  
## 31 si\_type undercls 0.0502774020  
## 174 international school2\_type 0.0494487773  
## 212 race residenc 0.0467570001  
## 215 gender\_tgd residenc -0.0464533529  
## 185 gender\_tgd finpast 0.0457403226  
## 121 si\_type international -0.0440133529  
## 168 undercls school2\_type 0.0435029804  
## 48 undercls sexual 0.0417955384  
## 216 sib\_freq residenc -0.0404756883  
## 124 sexual international -0.0397004734  
## 172 ed\_any school2\_type 0.0378156268  
## 201 sib\_freq ins\_no 0.0363262966  
## 190 meds\_any finpast 0.0357573471  
## 138 undercls meds\_any -0.0342805378  
## 108 undercls dx\_mh -0.0342655673  
## 200 gender\_tgd ins\_no 0.0338649164  
## 92 race ed\_any 0.0336766152  
## 95 gender\_tgd ed\_any 0.0332690519  
## 199 sexual ins\_no 0.0305256563  
## 125 gender\_tgd international -0.0299817070  
## 223 finpast residenc 0.0270190706  
## 63 undercls gender\_tgd 0.0263874086  
## 159 international religios -0.0257818865  
## 47 race sexual -0.0257195551  
## 127 ed\_any international 0.0249762136  
## 77 race sib\_freq -0.0241481361  
## 62 race gender\_tgd -0.0227760777  
## 203 dx\_mh ins\_no 0.0227252395  
## 198 undercls ins\_no -0.0221172846  
## 93 undercls ed\_any 0.0194278062  
## 204 international ins\_no 0.0191124756  
## 211 si\_type residenc -0.0189453333  
## 191 religios finpast 0.0179978900  
## 207 school2\_type ins\_no 0.0172153094  
## 152 race religios 0.0168093047  
## 183 undercls finpast 0.0158240222  
## 221 religios residenc 0.0139779594  
## 32 race undercls -0.0133017062  
## 222 school2\_type residenc 0.0116714358  
## 16 si\_type race 0.0111644567  
## 153 undercls religios 0.0076934945  
## 205 meds\_any ins\_no -0.0069937162  
## 217 ed\_any residenc 0.0065992090  
## 220 meds\_any residenc -0.0051922217  
## 206 religios ins\_no -0.0045839575  
## 219 international residenc -0.0021752121  
## 218 dx\_mh residenc 0.0012965719  
## 167 race school2\_type 0.0007879776

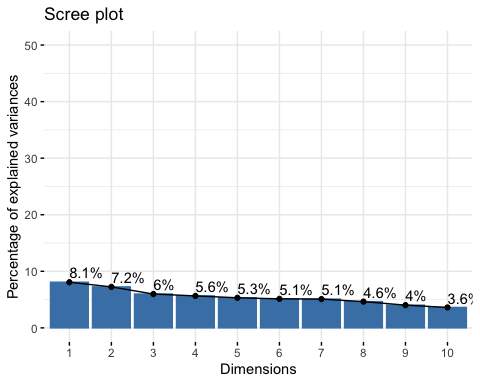


## PCA PREMOTIVATION

# Define diathesis vars for correlation matrix  
traindf\_prem <- traindf.v15 %>% select(sexual, age, gender\_tgd, sib\_freq, race, ed\_any, dx\_mh,   
 international, finpast, meds\_any, religios, school2\_type,   
 residenc, undercls, ins\_no)  
traindf\_prem <- na.omit(traindf\_prem)  
  
# create corr matrix, with dummy vars  
traindf\_prem <- traindf\_prem %>%   
 mutate\_if(is.numeric, as.factor) %>%   
 dummy\_cols(remove\_first\_dummy = TRUE) %>%   
 data.matrix()  
# traindf\_prem # check output  
  
results\_prem <- dummy\_cols(traindf\_prem, select\_columns = c("sexual", "age", "gender\_tgd", "sib\_freq", "finpast",  
 "race", "ed\_any", "dx\_mh", "meds\_any",  
 "religios", "international", "ins\_no",  
 "school2\_type", "residenc", "undercls"),  
 remove\_first\_dummy = TRUE, remove\_selected\_columns = TRUE)  
  
prem\_pca <- PCA(results\_prem, graph=FALSE) # apply PCA  
prem\_pca$eig # output matrix with eigenvalues

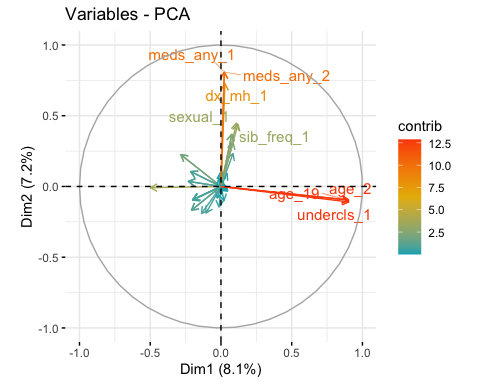
## eigenvalue percentage of variance cumulative percentage of variance  
## comp 1 3.390285e+00 8.072108e+00 8.072108  
## comp 2 3.043509e+00 7.246449e+00 15.318557  
## comp 3 2.513403e+00 5.984293e+00 21.302850  
## comp 4 2.372687e+00 5.649254e+00 26.952103  
## comp 5 2.233960e+00 5.318953e+00 32.271056  
## comp 6 2.155771e+00 5.132789e+00 37.403845  
## comp 7 2.136995e+00 5.088084e+00 42.491929  
## comp 8 1.948784e+00 4.639962e+00 47.131891  
## comp 9 1.688688e+00 4.020686e+00 51.152577  
## comp 10 1.516934e+00 3.611747e+00 54.764324  
## comp 11 1.354127e+00 3.224113e+00 57.988437  
## comp 12 1.319909e+00 3.142641e+00 61.131078  
## comp 13 1.281784e+00 3.051867e+00 64.182944  
## comp 14 1.220569e+00 2.906118e+00 67.089062  
## comp 15 1.193834e+00 2.842463e+00 69.931524  
## comp 16 1.163570e+00 2.770405e+00 72.701929  
## comp 17 1.124648e+00 2.677733e+00 75.379662  
## comp 18 1.061562e+00 2.527528e+00 77.907190  
## comp 19 1.033497e+00 2.460707e+00 80.367898  
## comp 20 1.007688e+00 2.399256e+00 82.767154  
## comp 21 9.051025e-01 2.155006e+00 84.922160  
## comp 22 8.867369e-01 2.111278e+00 87.033438  
## comp 23 8.325044e-01 1.982153e+00 89.015591  
## comp 24 7.838828e-01 1.866388e+00 90.881979  
## comp 25 7.465202e-01 1.777429e+00 92.659408  
## comp 26 6.521796e-01 1.552809e+00 94.212217  
## comp 27 5.915114e-01 1.408360e+00 95.620577  
## comp 28 5.016081e-01 1.194305e+00 96.814882  
## comp 29 4.656694e-01 1.108737e+00 97.923619  
## comp 30 3.566335e-01 8.491274e-01 98.772746  
## comp 31 2.708794e-01 6.449510e-01 99.417697  
## comp 32 2.445672e-01 5.823029e-01 100.000000  
## comp 33 1.167030e-26 2.778643e-26 100.000000  
## comp 34 1.097081e-26 2.612098e-26 100.000000  
## comp 35 6.511427e-27 1.550340e-26 100.000000  
## comp 36 1.931852e-27 4.599648e-27 100.000000  
## comp 37 6.765276e-28 1.610780e-27 100.000000  
## comp 38 6.425453e-28 1.529870e-27 100.000000  
## comp 39 4.675505e-28 1.113215e-27 100.000000  
## comp 40 9.340555e-29 2.223942e-28 100.000000  
## comp 41 8.165019e-29 1.944052e-28 100.000000  
## comp 42 6.419848e-29 1.528535e-28 100.000000

eigval\_prem <- get\_eigenvalue(prem\_pca)  
# eigval\_prem  
  
# visualization for PCA, FAMD  
fviz\_screeplot(prem\_pca, addlabels = TRUE, ylim = c(0, 50))



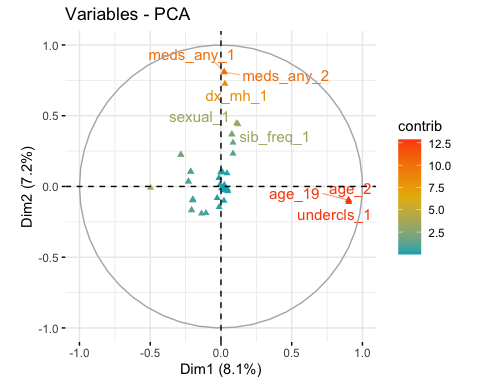
fviz\_pca\_var(prem\_pca,  
 col.var = "contrib", # Color by contributions to the PC  
 gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),  
 repel = TRUE # Avoid text overlapping  
 )

## Warning: ggrepel: 34 unlabeled data points (too many overlaps). Consider  
## increasing max.overlaps

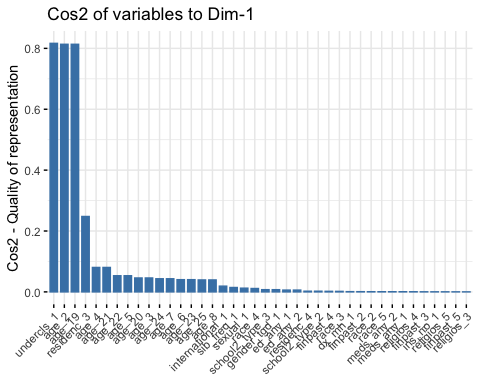


fviz\_famd\_var(prem\_pca,  
 col.var = "contrib", # Color by contributions to the PC  
 gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),  
 repel = TRUE # Avoid text overlapping  
 )

## Warning: ggrepel: 34 unlabeled data points (too many overlaps). Consider  
## increasing max.overlaps



fviz\_cos2(prem\_pca, choice = "var", axes = 1,  
 fill = "steelblue", color = "steelblue",  
 sort.val = c("desc", "asc", "none"),  
 top = 40,  
 xtickslab.rt = 45,  
 ggtheme = theme\_minimal())



## GLM Premotivation regression

glm\_prem <- traindf.v15 %>% select(si\_type, sexual, gender\_tgd, sib\_freq, race, ed\_any, finpast,  
 dx\_mh, international, religios, school2\_type, residenc, ins\_no,  
 meds\_any, undercls)  
glm\_prem <- na.omit(glm\_prem)  
  
# run regression  
glm\_prem <- glm(si\_type ~ sexual + gender\_tgd + sib\_freq + race + ed\_any + dx\_mh + undercls + finpast +  
 religios + school2\_type + international + residenc + meds\_any + ins\_no,  
 data = glm\_prem)  
summary(glm\_prem)

##   
## Call:  
## glm(formula = si\_type ~ sexual + gender\_tgd + sib\_freq + race +   
## ed\_any + dx\_mh + undercls + finpast + religios + school2\_type +   
## international + residenc + meds\_any + ins\_no, data = glm\_prem)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.76873 -0.14770 -0.05720 0.00049 1.02456   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.022524 0.012248 -1.839 0.066 .   
## sexualTRUE 0.073827 0.007267 10.159 < 2e-16 \*\*\*  
## gender\_tgdTRUE 0.079748 0.015159 5.261 1.46e-07 \*\*\*  
## sib\_freqTRUE 0.267287 0.008899 30.037 < 2e-16 \*\*\*  
## race 0.010049 0.002268 4.430 9.50e-06 \*\*\*  
## ed\_any1 0.090179 0.009048 9.966 < 2e-16 \*\*\*  
## dx\_mhTRUE 0.070596 0.007454 9.471 < 2e-16 \*\*\*  
## underclsTRUE 0.030236 0.006527 4.632 3.65e-06 \*\*\*  
## finpast 0.017167 0.002439 7.040 2.03e-12 \*\*\*  
## religios -0.011001 0.002137 -5.148 2.67e-07 \*\*\*  
## school2\_type2 0.025927 0.018654 1.390 0.165   
## school2\_type3 -0.004990 0.008930 -0.559 0.576   
## internationalTRUE -0.022603 0.011274 -2.005 0.045 \*   
## residenc 0.002862 0.002738 1.045 0.296   
## meds\_any1 0.067467 0.007990 8.444 < 2e-16 \*\*\*  
## ins\_noTRUE 0.038245 0.006766 5.653 1.61e-08 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.09784016)  
##   
## Null deviance: 1561.6 on 12853 degrees of freedom  
## Residual deviance: 1256.1 on 12838 degrees of freedom  
## AIC: 6618  
##   
## Number of Fisher Scoring iterations: 2

coef(glm\_prem) # retrieve coefficients

## (Intercept) sexualTRUE gender\_tgdTRUE sib\_freqTRUE   
## -0.022523517 0.073826810 0.079747929 0.267287178   
## race ed\_any1 dx\_mhTRUE underclsTRUE   
## 0.010048529 0.090178687 0.070595764 0.030236113   
## finpast religios school2\_type2 school2\_type3   
## 0.017167384 -0.011001338 0.025926790 -0.004990267   
## internationalTRUE residenc meds\_any1 ins\_noTRUE   
## -0.022603241 0.002862157 0.067467022 0.038245141

ci\_prem <- confint(glm\_prem) # retrieve confidence intervals

## Waiting for profiling to be done...

ci\_prem

## 2.5 % 97.5 %  
## (Intercept) -0.046530089 0.0014830552  
## sexualTRUE 0.059583190 0.0880704297  
## gender\_tgdTRUE 0.050036471 0.1094593876  
## sib\_freqTRUE 0.249846392 0.2847279640  
## race 0.005602710 0.0144943480  
## ed\_any1 0.072443954 0.1079134195  
## dx\_mhTRUE 0.055986894 0.0852046346  
## underclsTRUE 0.017443050 0.0430291756  
## finpast 0.012387682 0.0219470859  
## religios -0.015189790 -0.0068128862  
## school2\_type2 -0.010634616 0.0624881956  
## school2\_type3 -0.022492808 0.0125122742  
## internationalTRUE -0.044700585 -0.0005058984  
## residenc -0.002504664 0.0082289787  
## meds\_any1 0.051807533 0.0831265117  
## ins\_noTRUE 0.024984042 0.0515062396

model.prem <- tidy(glm\_prem) # Convert model to dataframe  
model.prem

## # A tibble: 16 x 5  
## term estimate std.error statistic p.value  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) -0.0225 0.0122 -1.84 6.60e- 2  
## 2 sexualTRUE 0.0738 0.00727 10.2 3.74e- 24  
## 3 gender\_tgdTRUE 0.0797 0.0152 5.26 1.46e- 7  
## 4 sib\_freqTRUE 0.267 0.00890 30.0 1.25e-191  
## 5 race 0.0100 0.00227 4.43 9.50e- 6  
## 6 ed\_any1 0.0902 0.00905 9.97 2.61e- 23  
## 7 dx\_mhTRUE 0.0706 0.00745 9.47 3.24e- 21  
## 8 underclsTRUE 0.0302 0.00653 4.63 3.65e- 6  
## 9 finpast 0.0172 0.00244 7.04 2.03e- 12  
## 10 religios -0.0110 0.00214 -5.15 2.67e- 7  
## 11 school2\_type2 0.0259 0.0187 1.39 1.65e- 1  
## 12 school2\_type3 -0.00499 0.00893 -0.559 5.76e- 1  
## 13 internationalTRUE -0.0226 0.0113 -2.00 4.50e- 2  
## 14 residenc 0.00286 0.00274 1.05 2.96e- 1  
## 15 meds\_any1 0.0675 0.00799 8.44 3.39e- 17  
## 16 ins\_noTRUE 0.0382 0.00677 5.65 1.61e- 8

model.prem %>%  
 mutate(or = exp(estimate), # Odds ratio/gradient  
 var.diag = diag(vcov(glm\_prem)), # Variance of each coefficient  
 or.se = sqrt(or^2 \* var.diag)) # Odds-ratio adjusted

## # A tibble: 16 x 8  
## term estimate std.error statistic p.value or var.diag or.se  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) -0.0225 0.0122 -1.84 6.60e- 2 0.978 1.50e-4 0.0120   
## 2 sexualTRUE 0.0738 0.00727 10.2 3.74e- 24 1.08 5.28e-5 0.00782  
## 3 gender\_tgdTRUE 0.0797 0.0152 5.26 1.46e- 7 1.08 2.30e-4 0.0164   
## 4 sib\_freqTRUE 0.267 0.00890 30.0 1.25e-191 1.31 7.92e-5 0.0116   
## 5 race 0.0100 0.00227 4.43 9.50e- 6 1.01 5.15e-6 0.00229  
## 6 ed\_any1 0.0902 0.00905 9.97 2.61e- 23 1.09 8.19e-5 0.00990  
## 7 dx\_mhTRUE 0.0706 0.00745 9.47 3.24e- 21 1.07 5.56e-5 0.00800  
## 8 underclsTRUE 0.0302 0.00653 4.63 3.65e- 6 1.03 4.26e-5 0.00673  
## 9 finpast 0.0172 0.00244 7.04 2.03e- 12 1.02 5.95e-6 0.00248  
## 10 religios -0.0110 0.00214 -5.15 2.67e- 7 0.989 4.57e-6 0.00211  
## 11 school2\_type2 0.0259 0.0187 1.39 1.65e- 1 1.03 3.48e-4 0.0191   
## 12 school2\_type3 -0.00499 0.00893 -0.559 5.76e- 1 0.995 7.97e-5 0.00889  
## 13 international… -0.0226 0.0113 -2.00 4.50e- 2 0.978 1.27e-4 0.0110   
## 14 residenc 0.00286 0.00274 1.05 2.96e- 1 1.00 7.50e-6 0.00275  
## 15 meds\_any1 0.0675 0.00799 8.44 3.39e- 17 1.07 6.38e-5 0.00855  
## 16 ins\_noTRUE 0.0382 0.00677 5.65 1.61e- 8 1.04 4.58e-5 0.00703

# LIFE EVENTS

traindf.v16 <- traindf.v14

## Illicit drug use

Creates illicit drug use var (limitation – 30 days reflection)

## adjusted for nested question status  
traindf.v16$drug\_coc[is.na(traindf.v16$drug\_coc)] <- 0  
traindf.v16$drug\_her[is.na(traindf.v16$drug\_her)] <- 0  
traindf.v16$drug\_met[is.na(traindf.v16$drug\_met)] <- 0  
traindf.v16$drug\_stim[is.na(traindf.v16$drug\_stim)] <- 0  
traindf.v16$drug\_ect[is.na(traindf.v16$drug\_ect)] <- 0  
traindf.v16$drug\_opi[is.na(traindf.v16$drug\_opi)] <- 0  
traindf.v16$drug\_other[is.na(traindf.v16$drug\_other)] <- 0  
  
# visualize current output before transformation  
table(traindf.v16$drug\_coc)

##   
## 0 1   
## 12613 241

table(traindf.v16$drug\_her) # too small

##   
## 0 1   
## 12852 2

table(traindf.v16$drug\_met) # too small

##   
## 0 1   
## 12848 6

table(traindf.v16$drug\_stim)

##   
## 0 1   
## 12560 294

table(traindf.v16$drug\_ect) # too small

##   
## 0 1   
## 12793 61

table(traindf.v16$drug\_opi) # too small

##   
## 0 1   
## 12786 68

table(traindf.v16$drug\_other) # too vague, small

##   
## 0 1   
## 12715 139

# Sample is too small to analyze individual drugs, except for marijuana.   
# Transform to illicit drug y/n  
traindf.v16 <- traindf.v16 %>%  
 mutate(drugs\_yn = case\_when(drug\_coc == 1 ~ 1,  
 drug\_her == 1 ~ 1,  
 drug\_met == 1 ~ 1,  
 drug\_stim == 1 ~ 1,  
 drug\_ect == 1 ~ 1,  
 drug\_opi == 1 ~ 1,  
 drug\_other == 1 ~ 1))  
traindf.v16$drugs\_yn[is.na(traindf.v16$drugs\_yn)] <- 0  
traindf.v16$drugs\_yn <- as.logical(traindf.v16$drugs\_yn)  
  
table(traindf.v16$drugs\_yn, useNA = "always")

##   
## FALSE TRUE <NA>   
## 12265 589 0

jarque.bera.test(traindf.v16$drugs\_yn)

##   
## Jarque Bera Test  
##   
## data: traindf.v16$drugs\_yn  
## X-squared = 192881, df = 2, p-value < 2.2e-16

## Marijuana use

traindf.v16$drug\_mar[is.na(traindf.v16$drug\_mar)] <- 0  
table(traindf.v16$drug\_mar)

##   
## 0 1   
## 10300 2554

traindf.v16 <- traindf.v16 %>%  
 mutate(drug\_mar = case\_when(drug\_mar == 1 ~ 1,   
 drug\_mar == 0 ~ 0)) %>%  
 filter(!is.na(drug\_mar)) # filters out NAs  
traindf.v16$drug\_mar <- as.logical(traindf.v16$drug\_mar)  
  
table(traindf.v16$drug\_mar, useNA = "always")

##   
## FALSE TRUE <NA>   
## 10300 2554 0

jarque.bera.test(traindf.v16$drug\_mar)

##   
## Jarque Bera Test  
##   
## data: traindf.v16$drug\_mar  
## X-squared = 4928.6, df = 2, p-value < 2.2e-16

## Current financial stressors

traindf.v16 <- traindf.v16 %>%  
 # make dichotomous  
 mutate(fincur = case\_when(fincur == 3 ~ 0, # sometimes  
 fincur == 4 ~ 0, # rarely or never  
 fincur == 5 ~ 0,  
 fincur == 1 ~ 1, # always or often stressful  
 fincur == 2 ~ 1)) %>%  
 filter(!is.na(fincur)) # filters for NAs  
traindf.v16$fincur <- as.logical(traindf.v16$fincur)  
  
table(traindf.v16$fincur, useNA = "always") # check output, skew

##   
## FALSE TRUE <NA>   
## 8266 4587 0

jarque.bera.test(traindf.v16$fincur)

##   
## Jarque Bera Test  
##   
## data: traindf.v16$fincur  
## X-squared = 2210.4, df = 2, p-value < 2.2e-16

## ISI scale

table(traindf.v16$isi\_1)

##   
## 1 2 3 4 5   
## 121 58 48 24 14

table(traindf.v16$isi\_2)

##   
## 1 2 3 4 5   
## 143 49 42 22 8

traindf.v16 = subset(traindf.v16, select= -c(isi\_1, isi\_2, isi\_3, isi\_4,   
 isi\_5, isi\_6, isi\_7)) # clean up var

## Abuse or assault, in the last year only

table(traindf.v16$abuse\_life)

##   
## 1 2 3 4 5   
## 7228 1224 2172 598 1619

table(traindf.v16$abuse\_recent)

##   
## 1 2 3 4 5   
## 95 112 642 1977 2782

# create dichotomous variable called 'abuse\_year' to reflect abuse that occurred in last 12 months  
traindf.v17 <- traindf.v16 %>%  
 mutate(abuse\_year = case\_when(abuse\_recent == 1 ~ 1,  
 abuse\_recent == 2 ~ 1,  
 abuse\_recent == 3 ~ 1,  
 abuse\_recent == 4 ~ 0,  
 abuse\_recent == 5 ~ 0,  
 abuse\_life == 0 ~ 0)) %>%  
 filter(!is.na(abuse\_life))  
table(traindf.v17$abuse\_year) # check output

##   
## 0 1   
## 4759 849

# restructure sexual assault var  
traindf.v17 <- traindf.v17 %>%  
 mutate(assault = case\_when(  
 assault\_sex == 1 ~ 1,  
 assault\_emo == 1 ~ 1,  
 assault\_phys == 1 ~ 1,   
 abuse\_year == 1 ~ 1,  
 abuse\_year == 0 ~ 0,  
 assault\_sex == 2 ~ 0,  
 assault\_emo == 0 ~ 0,  
 assault\_phys == 1 ~ 0)) %>%  
 mutate(assault = case\_when(assault == 1 ~ 1,  
 assault == 0 ~ 0)) %>%  
 filter(!is.na(assault))  
traindf.v17$assault <- as.logical(traindf.v17$assault)  
  
# tabulate and check output  
table(traindf.v17$assault, useNA = "always")

##   
## FALSE TRUE <NA>   
## 7927 4914 0

jarque.bera.test(traindf.v17$assault)

##   
## Jarque Bera Test  
##   
## data: traindf.v17$assault  
## X-squared = 2169.2, df = 2, p-value < 2.2e-16

traindf.v17 = subset(traindf.v17, select= -c(abuse\_life, abuse\_recent)) # clean up var

## Binging frequency

# set all binge\_fr values equal to 0  
traindf.v17$binge\_fr\_f[is.na(traindf.v17$binge\_fr\_f)] <- 0  
traindf.v17$binge\_fr\_m[is.na(traindf.v17$binge\_fr\_m)] <- 0  
traindf.v17$binge\_fr\_o[is.na(traindf.v17$binge\_fr\_o)] <- 0  
  
# create one binge var  
traindf.v18 <- traindf.v17 %>%   
 mutate(binge\_fr = as.numeric(binge\_fr\_f) + as.numeric(binge\_fr\_m) + as.numeric(binge\_fr\_o))  
table(traindf.v18$binge\_fr)

##   
## 0 1 2 3 4 5 6 7   
## 4706 2795 1988 1515 1416 285 88 48

traindf.v18 <- traindf.v18 %>%  
 mutate(binge = case\_when(binge\_fr >= 4 ~ 1,  
 binge\_fr < 4 ~ 0)) %>%  
 filter(!is.na(binge))  
traindf.v18$binge <-as.logical(traindf.v18$binge)  
  
table(traindf.v18$binge, useNA = "always") # check output

##   
## FALSE TRUE <NA>   
## 11004 1837 0

jarque.bera.test(traindf.v18$binge)

##   
## Jarque Bera Test  
##   
## data: traindf.v18$binge  
## X-squared = 11387, df = 2, p-value < 2.2e-16

traindf.v18 = subset(traindf.v18, select= -c(binge\_fr\_f, binge\_fr\_m, binge\_fr\_o)) # clean up var

## Recent Psychiatric Hospitalization, last 12 months

# Create variable for any psychiatric hospitalization in the last 12 months  
traindf.v18 <- traindf.v18 %>%   
 mutate(psyhx = case\_when((prov\_pes == 1 | prov\_inp == 1 | prov\_par == 1) ~ 1, # inpatient, IOPs, etc.  
 (prov\_pes == 0 & prov\_inp == 0 & prov\_par == 0) ~ 0, # no crises intervened on  
 ther\_ever == 0 ~ 0)) %>%  
 filter(!is.na(psyhx)) # filter out NAs  
traindf.v18$psyhx <- as.logical(traindf.v18$psyhx) # change class to boolean  
  
table(traindf.v18$psyhx, useNA = "always") # check output, skew

##   
## FALSE TRUE <NA>   
## 12636 191 0

jarque.bera.test(traindf.v18$psyhx)

##   
## Jarque Bera Test  
##   
## data: traindf.v18$psyhx  
## X-squared = 2203074, df = 2, p-value < 2.2e-16

traindf.v18 = subset(traindf.v18, select= -c(prov\_pes, prov\_inp, prov\_par)) # clean up var

## Current GPA

traindf.v18 <- traindf.v18 %>%  
# bin grades to As, Bs, Cs, and D+ and below  
 mutate(gpa\_sr = case\_when(gpa\_sr == 0 ~ 1, # A+  
 gpa\_sr == 1 ~ 1, # A  
 gpa\_sr == 2 ~ 1, # A-  
 gpa\_sr == 3 ~ 2, # B+  
 gpa\_sr == 4 ~ 2, # B  
 gpa\_sr == 5 ~ 2, # B-  
 gpa\_sr == 6 ~ 3, # C+  
 gpa\_sr == 7 ~ 3, # C  
 gpa\_sr == 8 ~ 3, # C-  
 gpa\_sr == 9 ~ 4, # D+ and below  
 gpa\_sr == 10 ~ 5)) %>% # Don't know  
 filter(!is.na(gpa\_sr)) # filter out NAs  
traindf.v18$gpa\_sr <- as.numeric(traindf.v18$gpa\_sr) # change to numeric scale  
  
table(traindf.v18$gpa\_sr, useNA = "always") # check output, skew

##   
## 1 2 3 4 5 <NA>   
## 6741 4133 671 44 1145 0

jarque.bera.test(traindf.v18$gpa\_sr)

##   
## Jarque Bera Test  
##   
## data: traindf.v18$gpa\_sr  
## X-squared = 9321.5, df = 2, p-value < 2.2e-16

## Life events analyses

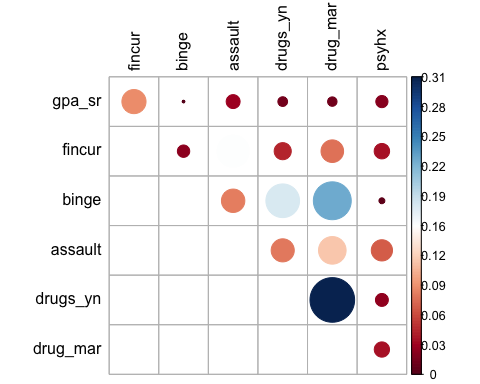
traindf.v19 <- traindf.v18

### LE correlation

# Define life event var candidates for correlation matrix  
traindf\_le <- traindf.v19 %>% select(gpa\_sr, fincur, binge, assault, drugs\_yn, drug\_mar, psyhx)  
  
# convert traindf to numeric in order to run correlations  
traindf\_le <- traindf\_le %>% mutate\_if(is.character, as.factor)  
traindf\_le <- traindf\_le %>% mutate\_if(is.factor, as.numeric)  
  
# run a correlation and drop the insignificant ones  
corr <- cor(traindf\_le, use = "pairwise.complete.obs")   
corr[lower.tri(corr,diag=TRUE)] <- NA # prepare to drop dups and correlations of 1   
corr[corr %in% c(-1, 1)] <- NA # drop perfect correlations  
corr <- as.data.frame(as.table(corr)) #turn into a 3-column table  
corr <- na.omit(corr) #remove the NA values from above   
corr <- corr[order(-abs(corr$Freq)),] #sort by highest correlation  
print(corr)

## Var1 Var2 Freq  
## 40 drugs\_yn drug\_mar 0.3107783860  
## 38 binge drug\_mar 0.2240331337  
## 31 binge drugs\_yn 0.1768956585  
## 23 fincur assault 0.1563402247  
## 39 assault drug\_mar 0.1188106270  
## 8 gpa\_sr fincur 0.0909423749  
## 24 binge assault 0.0855083838  
## 32 assault drugs\_yn 0.0832408745  
## 37 fincur drug\_mar 0.0796093374  
## 46 assault psyhx 0.0703364785  
## 30 fincur drugs\_yn 0.0458673371  
## 44 fincur psyhx 0.0373132226  
## 48 drug\_mar psyhx 0.0365390171  
## 22 gpa\_sr assault 0.0304309780  
## 47 drugs\_yn psyhx 0.0262054428  
## 16 fincur binge 0.0240081203  
## 43 gpa\_sr psyhx 0.0232356031  
## 29 gpa\_sr drugs\_yn 0.0147791004  
## 36 gpa\_sr drug\_mar 0.0139879328  
## 45 binge psyhx 0.0055430220  
## 15 gpa\_sr binge 0.0001975266

#turn corr back into matrix in order to plot with corrplot  
mtx\_corr <- acast(corr, Var1~Var2, value.var="Freq")   
corrplot(mtx\_corr, is.corr=FALSE, tl.col="black", na.label=" ")



### PCA LE

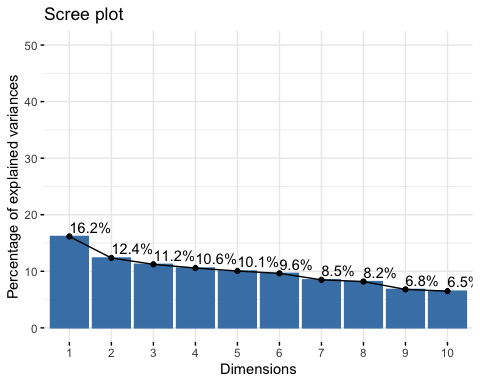
traindf\_le <- traindf.v19 %>% select(gpa\_sr, fincur, assault, binge, drugs\_yn, drug\_mar, psyhx)  
traindf\_le <- na.omit(traindf\_le)  
  
# create corr matrix, with dummy vars  
traindf\_le <- traindf\_le %>%   
 mutate\_if(is.numeric, as.factor) %>%   
 dummy\_cols(remove\_first\_dummy = TRUE) %>%   
 data.matrix()  
# traindf\_le # check output  
  
results\_le <- dummy\_cols(traindf\_le, select\_columns = c("gpa\_sr", "fincur", "binge", "drugs\_yn", "drug\_mar",   
 "assault", "psyhx"),  
 remove\_first\_dummy = TRUE, remove\_selected\_columns = TRUE)  
  
le\_pca <- PCA(results\_le, graph=FALSE) # apply PCA  
le\_pca$eig # output matrix with eigenvalues

## eigenvalue percentage of variance cumulative percentage of variance  
## comp 1 1.6159498 16.159498 16.15950  
## comp 2 1.2354791 12.354791 28.51429  
## comp 3 1.1235979 11.235979 39.75027  
## comp 4 1.0579792 10.579792 50.33006  
## comp 5 1.0053031 10.053031 60.38309  
## comp 6 0.9637989 9.637989 70.02108  
## comp 7 0.8493968 8.493968 78.51505  
## comp 8 0.8185410 8.185410 86.70046  
## comp 9 0.6803631 6.803631 93.50409  
## comp 10 0.6495911 6.495911 100.00000

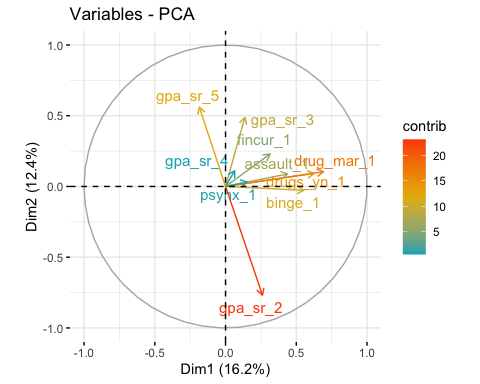
eigval\_le <- get\_eigenvalue(le\_pca)  
eigval\_le

## eigenvalue variance.percent cumulative.variance.percent  
## Dim.1 1.6159498 16.159498 16.15950  
## Dim.2 1.2354791 12.354791 28.51429  
## Dim.3 1.1235979 11.235979 39.75027  
## Dim.4 1.0579792 10.579792 50.33006  
## Dim.5 1.0053031 10.053031 60.38309  
## Dim.6 0.9637989 9.637989 70.02108  
## Dim.7 0.8493968 8.493968 78.51505  
## Dim.8 0.8185410 8.185410 86.70046  
## Dim.9 0.6803631 6.803631 93.50409  
## Dim.10 0.6495911 6.495911 100.00000

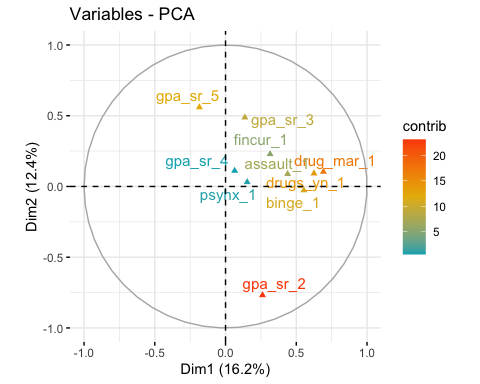
# visualization for PCA, FAMD  
fviz\_screeplot(le\_pca, addlabels = TRUE, ylim = c(0, 50))



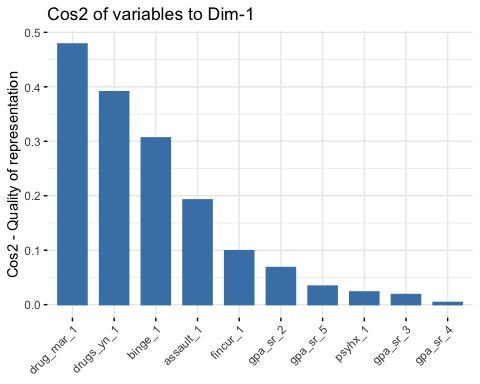
fviz\_pca\_var(le\_pca,  
 col.var = "contrib", # Color by contributions to the PC  
 gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),  
 repel = TRUE # Avoid text overlapping  
 )



fviz\_famd\_var(le\_pca,  
 col.var = "contrib", # Color by contributions to the PC  
 gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),  
 repel = TRUE # Avoid text overlapping  
 )



fviz\_cos2(le\_pca, choice = "var", axes = 1,  
 fill = "steelblue", color = "steelblue",  
 sort.val = c("desc", "asc", "none"),  
 top = 40,  
 xtickslab.rt = 45,  
 ggtheme = theme\_minimal())



### GLM Life event regression

# prepare dataframe for regression  
glm\_le <- traindf.v19 %>% select(si\_type, gpa\_sr, assault, binge, drugs\_yn,   
 drug\_mar, psyhx, fincur)  
glm\_le <- na.omit(glm\_le)  
  
# run regression  
glm\_le <- glm(si\_type ~ gpa\_sr + assault + binge + drugs\_yn + drug\_mar +   
 psyhx + fincur,  
 data = glm\_le)  
summary(glm\_le)

##   
## Call:  
## glm(formula = si\_type ~ gpa\_sr + assault + binge + drugs\_yn +   
## drug\_mar + psyhx + fincur, data = glm\_le)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.8382 -0.1636 -0.1051 -0.0495 0.9763   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.035073 0.006192 5.664 1.51e-08 \*\*\*  
## gpa\_sr 0.014431 0.002545 5.670 1.46e-08 \*\*\*  
## assaultTRUE 0.114074 0.006231 18.308 < 2e-16 \*\*\*  
## bingeTRUE -0.025844 0.008751 -2.953 0.00315 \*\*   
## drugs\_ynTRUE 0.084018 0.015068 5.576 2.51e-08 \*\*\*  
## drug\_marTRUE 0.050073 0.007980 6.275 3.61e-10 \*\*\*  
## psyhxTRUE 0.484997 0.024565 19.743 < 2e-16 \*\*\*  
## fincurTRUE 0.055555 0.006294 8.827 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.1115799)  
##   
## Null deviance: 1549.9 on 12733 degrees of freedom  
## Residual deviance: 1420.0 on 12726 degrees of freedom  
## AIC: 8221.7  
##   
## Number of Fisher Scoring iterations: 2

coef(glm\_le)

## (Intercept) gpa\_sr assaultTRUE bingeTRUE drugs\_ynTRUE drug\_marTRUE   
## 0.03507269 0.01443055 0.11407414 -0.02584399 0.08401773 0.05007289   
## psyhxTRUE fincurTRUE   
## 0.48499726 0.05555460

ci\_le <- confint(glm\_le)

## Waiting for profiling to be done...

ci\_le

## 2.5 % 97.5 %  
## (Intercept) 0.022937216 0.047208172  
## gpa\_sr 0.009442431 0.019418666  
## assaultTRUE 0.101861958 0.126286321  
## bingeTRUE -0.042995688 -0.008692294  
## drugs\_ynTRUE 0.054485918 0.113549538  
## drug\_marTRUE 0.034432412 0.065713358  
## psyhxTRUE 0.436850497 0.533144025  
## fincurTRUE 0.043219478 0.067889718

model.le <- tidy(glm\_le) # Convert model to dataframe for easy manipulation  
model.le

## # A tibble: 8 x 5  
## term estimate std.error statistic p.value  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) 0.0351 0.00619 5.66 1.51e- 8  
## 2 gpa\_sr 0.0144 0.00255 5.67 1.46e- 8  
## 3 assaultTRUE 0.114 0.00623 18.3 6.33e-74  
## 4 bingeTRUE -0.0258 0.00875 -2.95 3.15e- 3  
## 5 drugs\_ynTRUE 0.0840 0.0151 5.58 2.51e- 8  
## 6 drug\_marTRUE 0.0501 0.00798 6.27 3.61e-10  
## 7 psyhxTRUE 0.485 0.0246 19.7 1.73e-85  
## 8 fincurTRUE 0.0556 0.00629 8.83 1.21e-18

model.le %>%  
 mutate(or = exp(estimate), # Odds ratio/gradient  
 var.diag = diag(vcov(glm\_le)), # Variance of each coefficient  
 or.se = sqrt(or^2 \* var.diag)) # Odds-ratio adjusted

## # A tibble: 8 x 8  
## term estimate std.error statistic p.value or var.diag or.se  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) 0.0351 0.00619 5.66 1.51e- 8 1.04 0.0000383 0.00641  
## 2 gpa\_sr 0.0144 0.00255 5.67 1.46e- 8 1.01 0.00000648 0.00258  
## 3 assaultTRUE 0.114 0.00623 18.3 6.33e-74 1.12 0.0000388 0.00698  
## 4 bingeTRUE -0.0258 0.00875 -2.95 3.15e- 3 0.974 0.0000766 0.00853  
## 5 drugs\_ynTRUE 0.0840 0.0151 5.58 2.51e- 8 1.09 0.000227 0.0164   
## 6 drug\_marTRUE 0.0501 0.00798 6.27 3.61e-10 1.05 0.0000637 0.00839  
## 7 psyhxTRUE 0.485 0.0246 19.7 1.73e-85 1.62 0.000603 0.0399   
## 8 fincurTRUE 0.0556 0.00629 8.83 1.21e-18 1.06 0.0000396 0.00665

# DEFEAT-ENTRAPMENT

traindf.v20 <- traindf.v18

## PHQ risk/severity

traindf.v20$deprawsc <- as.numeric(traindf.v20$deprawsc)  
  
traindf.v20 <- traindf.v20 %>%  
 filter(!is.na(deprawsc)) # filters out NAs  
  
table(traindf.v20$deprawsc, useNA ="always")

##   
## 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15   
## 577 626 901 992 963 869 874 875 710 671 567 506 418 421 351 312   
## 16 17 18 19 20 21 22 23 24 25 26 27 <NA>   
## 301 279 273 181 153 137 127 104 101 64 43 80 0

# Categorize by clinical translation for later --  
traindf.v20 <- traindf.v20 %>%  
 mutate(dep\_none = case\_when(deprawsc < 5 ~ 1, # creates bin for no/minimal depression  
 deprawsc > 4 ~ 0)) %>%  
 mutate(dep\_mild = case\_when((deprawsc > 4 & deprawsc < 10) ~ 1, # bin for mild depression  
 (deprawsc < 5 | deprawsc > 9) ~ 0)) %>%  
 mutate(dep\_mod = case\_when((deprawsc > 9 & deprawsc < 15) ~ 1, # bin for moderate depression  
 (deprawsc > 14 | deprawsc < 10) ~ 0)) %>%  
 mutate(dep\_modsev = case\_when((deprawsc > 14 & deprawsc < 20) ~ 1, # bin for mod-severe depression  
 (deprawsc > 19 | deprawsc < 15) ~ 0)) %>%  
 mutate(dep\_sev = case\_when(deprawsc > 19 ~ 1, # bin for severe depression  
 deprawsc < 20 ~ 0)) %>%  
 mutate(dep\_na = case\_when(is.na(deprawsc) ~ 1, # filter out any NAs  
 !is.na(deprawsc) ~ 0))  
traindf.v20$deprawsc <- as.numeric(traindf.v20$deprawsc)  
  
# Check outputs  
table(traindf.v20$dep\_none)

##   
## 0 1   
## 8417 4059

table(traindf.v20$dep\_mild)

##   
## 0 1   
## 8477 3999

table(traindf.v20$dep\_mod)

##   
## 0 1   
## 10213 2263

table(traindf.v20$dep\_modsev)

##   
## 0 1   
## 11130 1346

table(traindf.v20$dep\_sev)

##   
## 0 1   
## 11667 809

table(traindf.v20$dep\_na)

##   
## 0   
## 12476

traindf.v20 <- traindf.v20 %>%  
 mutate(dep\_type = case\_when(dep\_none == 1 ~ 1,  
 dep\_mild == 1 ~ 2,  
 dep\_mod == 1 ~ 3,  
 dep\_modsev == 1 ~ 4,  
 dep\_sev == 1 ~ 5))  
  
table(traindf.v20$dep\_type, useNA = "always")

##   
## 1 2 3 4 5 <NA>   
## 4059 3999 2263 1346 809 0

## GAD7 total

traindf.v20$anx\_score <- as.numeric(traindf.v20$anx\_score)  
  
# Convert to severity  
traindf.v20 <- traindf.v20 %>%  
 mutate(anx\_type = case\_when(anx\_score > 14 ~ 4,  
 anx\_score < 5 ~ 1, # abing for no/minimal anxiety  
 anx\_score < 10 ~ 2, # bin for mild anxiety  
 anx\_score < 15 ~ 3)) %>% # bin for severe anxiety  
 filter(!is.na(anx\_type)) # filters out all NAs  
  
traindf.v20$anx\_type <- as.numeric(traindf.v20$anx\_type) # check output, skew  
jarque.bera.test(traindf.v20$anx\_type)

##   
## Jarque Bera Test  
##   
## data: traindf.v20$anx\_type  
## X-squared = 1130.3, df = 2, p-value < 2.2e-16

table(traindf.v20$anx\_type, useNA = "always") # check output

##   
## 1 2 3 4 <NA>   
## 4853 3587 2197 1630 0

table(traindf.v20$anx\_score, useNA = "always")

##   
## 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15   
## 1054 837 1004 1017 941 897 806 868 524 492 467 465 415 409 441 314   
## 16 17 18 19 20 21 <NA>   
## 289 245 204 192 118 268 0

## Academic impact

traindf.v20 <- traindf.v20 %>%  
 filter(!is.na(aca\_impa)) # filter for NAs  
traindf.v20$aca\_impa <- as.numeric(traindf.v20$aca\_impa)  
  
table(traindf.v20$aca\_impa, useNA = "always") # check output, skew

##   
## 1 2 3 4 <NA>   
## 2754 4133 2975 2378 0

jarque.bera.test(traindf.v20$aca\_impa)

##   
## Jarque Bera Test  
##   
## data: traindf.v20$aca\_impa  
## X-squared = 716.35, df = 2, p-value < 2.2e-16

## Academic failure

table(traindf.v20$failed)

##   
## 0 1   
## 1347 245

# sample too small in the elective, remove  
traindf.v20 = subset(traindf.v20, select= -c(failed)) # clean up var

## Depression impact

traindf.v20 <- traindf.v20 %>%  
 filter(!is.na(dep\_impa)) # filters out NAs  
traindf.v20$dep\_impa <- as.numeric(traindf.v20$dep\_impa)  
  
table(traindf.v20$dep\_impa, useNA = "always") # check output, NAs, skew

##   
## 1 2 3 4 <NA>   
## 3325 6232 1476 607 0

jarque.bera.test(traindf.v20$dep\_impa)

##   
## Jarque Bera Test  
##   
## data: traindf.v20$dep\_impa  
## X-squared = 1156.8, df = 2, p-value < 2.2e-16

## Anxiety impact

traindf.v20$gad7\_impa <- as.numeric(traindf.v20$gad7\_impa)  
table(traindf.v20$gad7\_impa, useNA = "always") # check output, skew, NAs

##   
## 1 2 3 4 <NA>   
## 2603 6029 1653 625 730

# Perceived MH needs

traindf.v20 <- traindf.v20 %>%  
 # reverse coding  
 mutate(percneed = case\_when(percneed == 1 ~ 6, # strongly agrees that needs help  
 percneed == 2 ~ 5,  
 percneed == 3 ~ 4,  
 percneed == 4 ~ 3,  
 percneed == 5 ~ 2,  
 percneed == 6 ~ 1)) %>% # strongly disagree that needs help  
 filter(!is.na(percneed)) # filters for NAs  
  
table(traindf.v20$percneed, useNA = "always") # check output, NAs

##   
## 1 2 3 4 5 6 <NA>   
## 1800 1826 777 1960 2112 3160 0

## Defeat-entrapment analysis

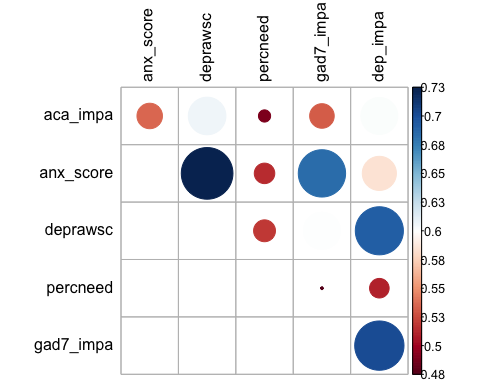
traindf.v21 <- traindf.v20

### Defeat-entrapment correlation

# Define defeat variable candidates for correlation matrix  
traindf\_def <- traindf.v21 %>% select(aca\_impa, anx\_score, deprawsc, percneed, gad7\_impa, dep\_impa)  
  
 # convert traindf to numeric in order to run correlations  
traindf\_def <- traindf\_def %>% mutate\_if(is.character, as.factor)  
traindf\_def <- traindf\_def %>% mutate\_if(is.factor, as.numeric)  
  
# run a correlation and drop the insignificant ones  
corr <- cor(traindf\_def, use = "pairwise.complete.obs")   
corr[lower.tri(corr, diag=TRUE)] <- NA # prepare to drop duplicates and correlations of 1   
corr[corr %in% c(-1, 1)] <- NA # drop perfect correlations  
corr <- as.data.frame(as.table(corr)) # turn into a 3-column table  
corr <- na.omit(corr) # remove the NA values from above   
corr <- corr[order(-abs(corr$Freq)),] # sort by highest correlation  
print(corr)

## Var1 Var2 Freq  
## 14 anx\_score deprawsc 0.7279235  
## 35 gad7\_impa dep\_impa 0.7052641  
## 33 deprawsc dep\_impa 0.6943472  
## 26 anx\_score gad7\_impa 0.6874530  
## 13 aca\_impa deprawsc 0.6093345  
## 31 aca\_impa dep\_impa 0.6037926  
## 27 deprawsc gad7\_impa 0.6029671  
## 32 anx\_score dep\_impa 0.5860469  
## 7 aca\_impa anx\_score 0.5377206  
## 25 aca\_impa gad7\_impa 0.5332678  
## 21 deprawsc percneed 0.5215798  
## 20 anx\_score percneed 0.5161293  
## 34 percneed dep\_impa 0.5125966  
## 19 aca\_impa percneed 0.4906232  
## 28 percneed gad7\_impa 0.4766213

#turn corr back into matrix in order to plot with corrplot  
mtx\_corr <- acast(corr, Var1~Var2, value.var="Freq")   
corrplot(mtx\_corr, is.corr=FALSE, tl.col="black", na.label=" ")

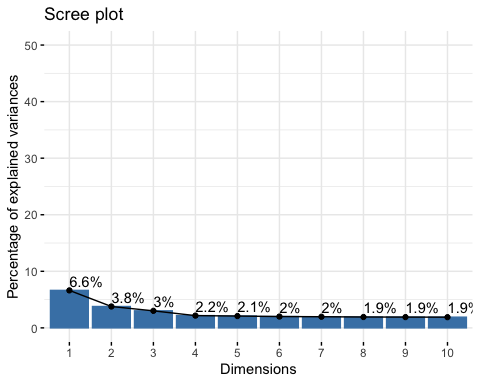


### PCA defeat-entrapment

traindf\_def <- traindf.v21 %>% select(aca\_impa, deprawsc, anx\_score, percneed, gad7\_impa, dep\_impa)  
traindf\_def <- na.omit(traindf\_def)  
  
# create corr matrix, with dummy vars   
traindf\_def <- traindf\_def %>%   
 mutate\_if(is.numeric, as.factor) %>%   
 dummy\_cols(remove\_first\_dummy = TRUE) %>%   
 data.matrix()  
# traindf\_def # check output  
  
results\_def <- dummy\_cols(traindf\_def, select\_columns = c("aca\_impa", "deprawsc", "anx\_score",  
 "percneed", "gad7\_impa", "dep\_impa"),  
 remove\_first\_dummy = TRUE, remove\_selected\_columns = TRUE)  
  
def\_pca <- PCA(results\_def, graph=FALSE) # apply PCA  
def\_pca$eig # output matrix with eigenvalues

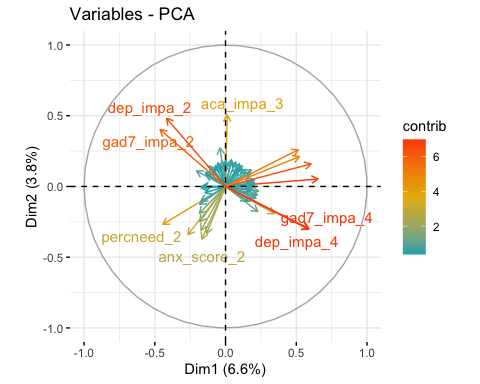
## eigenvalue percentage of variance cumulative percentage of variance  
## comp 1 3.97559768 6.62599613 6.625996  
## comp 2 2.26724683 3.77874472 10.404741  
## comp 3 1.80727252 3.01212086 13.416862  
## comp 4 1.30046863 2.16744772 15.584309  
## comp 5 1.25415743 2.09026238 17.674572  
## comp 6 1.20213206 2.00355343 19.678125  
## comp 7 1.18143971 1.96906618 21.647191  
## comp 8 1.15232176 1.92053627 23.567728  
## comp 9 1.14256915 1.90428192 25.472010  
## comp 10 1.13498454 1.89164089 27.363651  
## comp 11 1.12218020 1.87030033 29.233951  
## comp 12 1.11426595 1.85710992 31.091061  
## comp 13 1.11039328 1.85065547 32.941716  
## comp 14 1.10449070 1.84081783 34.782534  
## comp 15 1.09823961 1.83039934 36.612933  
## comp 16 1.09398272 1.82330453 38.436238  
## comp 17 1.09133628 1.81889380 40.255132  
## comp 18 1.08362639 1.80604399 42.061176  
## comp 19 1.08049389 1.80082315 43.861999  
## comp 20 1.07924716 1.79874526 45.660744  
## comp 21 1.06924845 1.78208075 47.442825  
## comp 22 1.06740090 1.77900150 49.221826  
## comp 23 1.06162277 1.76937129 50.991198  
## comp 24 1.05743740 1.76239567 52.753593  
## comp 25 1.05484817 1.75808029 54.511674  
## comp 26 1.05007223 1.75012038 56.261794  
## comp 27 1.03843503 1.73072505 57.992519  
## comp 28 1.03547190 1.72578650 59.718306  
## comp 29 1.03253251 1.72088751 61.439193  
## comp 30 1.02925292 1.71542153 63.154615  
## comp 31 1.02654239 1.71090398 64.865519  
## comp 32 1.02276828 1.70461380 66.570132  
## comp 33 1.01867442 1.69779070 68.267923  
## comp 34 1.00831004 1.68051673 69.948440  
## comp 35 1.00765797 1.67942994 71.627870  
## comp 36 1.00460633 1.67434388 73.302214  
## comp 37 1.00231809 1.67053015 74.972744  
## comp 38 0.99841385 1.66402308 76.636767  
## comp 39 0.99097243 1.65162071 78.288388  
## comp 40 0.98506299 1.64177164 79.930159  
## comp 41 0.98136327 1.63560545 81.565765  
## comp 42 0.97905992 1.63176653 83.197531  
## comp 43 0.96930361 1.61550602 84.813037  
## comp 44 0.95167561 1.58612602 86.399163  
## comp 45 0.93835908 1.56393180 87.963095  
## comp 46 0.91132829 1.51888049 89.481976  
## comp 47 0.89396124 1.48993541 90.971911  
## comp 48 0.84027178 1.40045296 92.372364  
## comp 49 0.82910447 1.38184079 93.754205  
## comp 50 0.75326081 1.25543468 95.009639  
## comp 51 0.73044167 1.21740278 96.227042  
## comp 52 0.63995287 1.06658812 97.293630  
## comp 53 0.45524165 0.75873609 98.052366  
## comp 54 0.35543129 0.59238549 98.644752  
## comp 55 0.31045278 0.51742130 99.162173  
## comp 56 0.15988726 0.26647877 99.428652  
## comp 57 0.13564196 0.22606994 99.654722  
## comp 58 0.12479239 0.20798732 99.862709  
## comp 59 0.05360320 0.08933866 99.952048  
## comp 60 0.02877131 0.04795219 100.000000

eigval\_def <- get\_eigenvalue(def\_pca)  
# eigval\_def  
  
# visualization for PCA, FAMD  
fviz\_screeplot(def\_pca, addlabels = TRUE, ylim = c(0, 50))



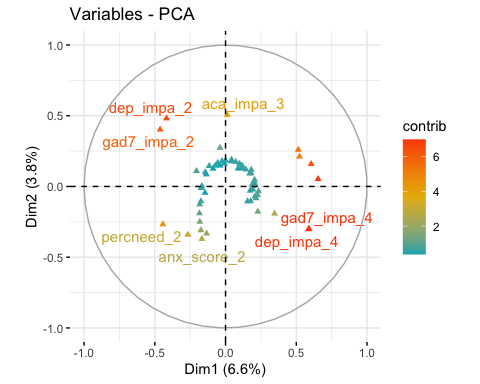
fviz\_pca\_var(def\_pca,  
 col.var = "contrib", # Color by contributions to the PC  
 gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),  
 repel = TRUE # Avoid text overlapping  
 )

## Warning: ggrepel: 53 unlabeled data points (too many overlaps). Consider  
## increasing max.overlaps

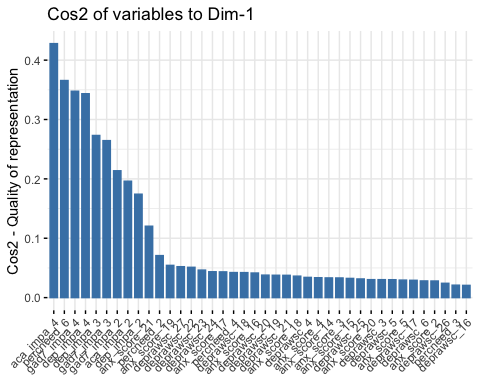


fviz\_famd\_var(def\_pca,  
 col.var = "contrib", # Color by contributions to the PC  
 gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),  
 repel = TRUE # Avoid text overlapping  
 )

## Warning: ggrepel: 53 unlabeled data points (too many overlaps). Consider  
## increasing max.overlaps



fviz\_cos2(def\_pca, choice = "var", axes = 1,   
 fill = "steelblue", color = "steelblue",  
 sort.val = c("desc", "asc", "none"),  
 top = 40,  
 xtickslab.rt = 45,  
 ggtheme = theme\_minimal())



### GLM defeat-entrapment

# prepare regression analysis  
results2\_def <- traindf.v21 %>% select(si\_type, aca\_impa, deprawsc, anx\_score, percneed, gad7\_impa, dep\_impa)  
glm\_def <- na.omit(results2\_def)  
  
# run defeat regression  
glm\_def <- glm(si\_type ~ aca\_impa + deprawsc + anx\_score + percneed + gad7\_impa + dep\_impa,   
 data = glm\_def)  
  
summary(glm\_def)

##   
## Call:  
## glm(formula = si\_type ~ aca\_impa + deprawsc + anx\_score + percneed +   
## gad7\_impa + dep\_impa, data = glm\_def)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.60054 -0.18701 -0.07270 0.04445 1.09542   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.1656009 0.0105606 -15.681 < 2e-16 \*\*\*  
## aca\_impa -0.0043909 0.0040923 -1.073 0.28331   
## deprawsc 0.0221211 0.0008737 25.320 < 2e-16 \*\*\*  
## anx\_score -0.0028582 0.0009353 -3.056 0.00225 \*\*   
## percneed 0.0289049 0.0021348 13.540 < 2e-16 \*\*\*  
## gad7\_impa 0.0049139 0.0064670 0.760 0.44736   
## dep\_impa 0.0133429 0.0065035 2.052 0.04023 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.1048845)  
##   
## Null deviance: 1424.6 on 10904 degrees of freedom  
## Residual deviance: 1143.0 on 10898 degrees of freedom  
## AIC: 6366.4  
##   
## Number of Fisher Scoring iterations: 2

coef(glm\_def) # retrieve coefficients

## (Intercept) aca\_impa deprawsc anx\_score percneed gad7\_impa   
## -0.165600851 -0.004390942 0.022121077 -0.002858204 0.028904946 0.004913929   
## dep\_impa   
## 0.013342853

ci\_def <- confint(glm\_def) # retrieve confidence intervals

## Waiting for profiling to be done...

ci\_def

## 2.5 % 97.5 %  
## (Intercept) -0.1862992212 -0.144902480  
## aca\_impa -0.0124117616 0.003629877  
## deprawsc 0.0204087015 0.023833453  
## anx\_score -0.0046913151 -0.001025092  
## percneed 0.0247207441 0.033089147  
## gad7\_impa -0.0077611524 0.017589010  
## dep\_impa 0.0005961427 0.026089564

model.def <- tidy(glm\_def) # Convert model to dataframe for easy manipulation  
model.def

## # A tibble: 7 x 5  
## term estimate std.error statistic p.value  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) -0.166 0.0106 -15.7 8.09e- 55  
## 2 aca\_impa -0.00439 0.00409 -1.07 2.83e- 1  
## 3 deprawsc 0.0221 0.000874 25.3 1.75e-137  
## 4 anx\_score -0.00286 0.000935 -3.06 2.25e- 3  
## 5 percneed 0.0289 0.00213 13.5 1.97e- 41  
## 6 gad7\_impa 0.00491 0.00647 0.760 4.47e- 1  
## 7 dep\_impa 0.0133 0.00650 2.05 4.02e- 2

model.def %>%  
 mutate(or = exp(estimate), # Odds ratio/gradient  
 var.diag = diag(vcov(glm\_def)), # Variance of each coefficient  
 or.se = sqrt(or^2 \* var.diag)) # Odds-ratio adjusted

## # A tibble: 7 x 8  
## term estimate std.error statistic p.value or var.diag or.se  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) -0.166 0.0106 -15.7 8.09e- 55 0.847 0.000112 0.00895   
## 2 aca\_impa -0.00439 0.00409 -1.07 2.83e- 1 0.996 0.0000167 0.00407   
## 3 deprawsc 0.0221 0.000874 25.3 1.75e-137 1.02 0.000000763 0.000893  
## 4 anx\_score -0.00286 0.000935 -3.06 2.25e- 3 0.997 0.000000875 0.000933  
## 5 percneed 0.0289 0.00213 13.5 1.97e- 41 1.03 0.00000456 0.00220   
## 6 gad7\_impa 0.00491 0.00647 0.760 4.47e- 1 1.00 0.0000418 0.00650   
## 7 dep\_impa 0.0133 0.00650 2.05 4.02e- 2 1.01 0.0000423 0.00659

# THREATS

traindf.v22 <- traindf.v20

## Depression secret

## Secretive MH  
traindf.v22$dep\_secret <- as.numeric(traindf.v22$dep\_secret)  
table(traindf.v22$dep\_secret, useNA = "always") # check output, too many NAs

##   
## 1 2 3 4 5 6 <NA>   
## 1272 1792 1946 1039 804 391 4391

## AAQ total

# create AAQ\_tot variable to sum up imputed AAQ questions  
traindf.v22 <- traindf.v22 %>%  
 mutate(aaq\_tot = as.numeric(AAQ\_1) + as.numeric(AAQ\_2) + as.numeric(AAQ\_3) + as.numeric(AAQ\_4) +   
 as.numeric(AAQ\_5) + as.numeric(AAQ\_6) + as.numeric(AAQ\_7))  
  
table(traindf.v22$aaq\_tot) # check output

##   
## 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26   
## 287 135 179 197 175 172 216 231 204 231 233 213 230 248 280 227 243 245 232 205   
## 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46   
## 214 245 188 165 163 161 128 118 113 105 88 88 59 52 53 41 33 33 25 22   
## 47 48 49   
## 17 15 34

# for loop for predictive cutoff point needed  
 rs\_df <- data.frame("cutoff" = c(), "rs" = c())  
 for (i in 8:48) {  
 aaq\_bin <- ifelse(traindf.v22$aaq\_tot >= i, TRUE, FALSE)  
 model = lm(traindf.v22$sui\_idea ~ aaq\_bin)  
 rs\_df <- rbind(rs\_df, c(i, summary(model)$r.squared))  
  
 }  
 names(rs\_df) <- c("cutoff", "rs")  
 rs\_df\_arranged <- rs\_df %>%  
 arrange(desc(rs))  
 rs\_df\_arranged[1,1]

## [1] 31

# deem 'rs\_df\_arranged[1,1]' as SI cutoff  
 traindf.v22 <- traindf.v22 %>%  
 mutate(aaq\_dum = case\_when(aaq\_tot > 40 ~ 4,  
 aaq\_tot > 30 ~ 3,  
 aaq\_tot > 20 ~ 2,  
 aaq\_tot < 21 ~ 1)) %>%  
 filter(!is.na(aaq\_dum)) # filter for NAs  
traindf.v22$aaq\_dum <- as.numeric(traindf.v22$aaq\_dum)  
  
table(traindf.v22$aaq\_dum, useNA = "always") # check output, NAs

##   
## 1 2 3 4 <NA>   
## 2951 2244 1075 273 0

## Overall school satisfaction

traindf.v22 <- traindf.v22 %>%  
 mutate(satisfied\_overall = case\_when(satisfied\_overall == 1 ~ 6, # reverse coding  
 satisfied\_overall == 2 ~ 5,  
 satisfied\_overall == 3 ~ 4,  
 satisfied\_overall == 4 ~ 3,  
 satisfied\_overall == 5 ~ 2,  
 satisfied\_overall == 6 ~ 1)) %>%   
 filter(!is.na(satisfied\_overall)) # filters out NAs  
traindf.v22$satisfied\_overall <- as.numeric(traindf.v22$satisfied\_overall)  
  
table(traindf.v22$satisfied\_overall, useNA = "always")

##   
## 1 2 3 4 5 6 <NA>   
## 868 3142 1585 572 248 122 0

jarque.bera.test(traindf.v22$satisfied\_overall)

##   
## Jarque Bera Test  
##   
## data: traindf.v22$satisfied\_overall  
## X-squared = 1689.6, df = 2, p-value < 2.2e-16

## Persist

traindf.v22 <- traindf.v22 %>%  
# review levels  
 mutate(persist = case\_when(persist == 1 ~ 1, # confident  
 persist == 2 ~ 2,   
 persist == 3 ~ 3,   
 persist == 4 ~ 4,   
 persist == 5 ~ 5,   
 persist == 6 ~ 6)) %>% # not confident  
 filter(!is.na(persist))   
traindf.v22$persist <- as.numeric(traindf.v22$persist)   
  
table(traindf.v22$persist, useNA = "always") # check output, skew

##   
## 1 2 3 4 5 6 <NA>   
## 3395 2007 882 158 68 27 0

jarque.bera.test(traindf.v22$persist)

##   
## Jarque Bera Test  
##   
## data: traindf.v22$persist  
## X-squared = 4036.8, df = 2, p-value < 2.2e-16

## Belonging

traindf.v22 <- traindf.v22 %>%  
 filter(!is.na(belong1)) # filters out NAs  
table(traindf.v22$belong1, useNA = "always")

##   
## 1 2 3 4 5 6 <NA>   
## 666 1972 2075 778 722 318 0

traindf.v22$belong1 <- as.numeric(traindf.v22$belong1)  
summary(traindf.v22$belong1)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.00 2.00 3.00 2.98 4.00 6.00

jarque.bera.test(traindf.v22$belong1)

##   
## Jarque Bera Test  
##   
## data: traindf.v22$belong1  
## X-squared = 394.82, df = 2, p-value < 2.2e-16

table(traindf.v22$belong2) # elective question, but common

##   
## 1 2 3 4 5 6   
## 592 2074 1485 565 299 164

table(traindf.v22$belong8) # nested x 2

##   
## 1 2 3 4 5 6   
## 349 733 1321 993 1377 401

table(traindf.v22$belong9) # not as good of a proxy

##   
## 1 2 3 4 5 6   
## 804 1490 1369 711 612 184

## Flourishing Scale

traindf.v22$flourish <- as.numeric(traindf.v22$flourish)  
  
# for loop for predictive cutoff point needed  
rs\_df <- data.frame("cutoff" = c(), "rs" = c())  
for (i in 9:55) {  
 flourish\_bin <- ifelse(traindf.v22$flourish >= i, TRUE, FALSE)  
 model = lm(traindf.v22$sui\_idea ~ flourish\_bin)  
 rs\_df <- rbind(rs\_df, c(i, summary(model)$r.squared))  
  
}  
names(rs\_df) <- c("cutoff", "rs")  
rs\_df\_arranged <- rs\_df %>%   
 arrange(desc(rs))   
rs\_df\_arranged[1,1] # chi-squared high

## [1] 38

# reverse levels for scoring  
traindf.v23 <- traindf.v22 %>%  
 mutate(flourish = case\_when(flourish == 56 ~ 8,  
 flourish == 55 ~ 9,  
 flourish == 54 ~ 10,  
 flourish == 53 ~ 11,  
 flourish == 52 ~ 12,  
 flourish == 51 ~ 13,  
 flourish == 50 ~ 14,  
 flourish == 49 ~ 15,  
 flourish == 48 ~ 16,  
 flourish == 47 ~ 17,  
 flourish == 46 ~ 18,  
 flourish == 45 ~ 19,  
 flourish == 44 ~ 20,  
 flourish == 43 ~ 21,  
 flourish == 42 ~ 22,  
 flourish == 41 ~ 23,  
 flourish == 40 ~ 24,  
 flourish == 39 ~ 25,  
 flourish == 38 ~ 26,  
 flourish == 37 ~ 27,  
 flourish == 36 ~ 28,  
 flourish == 35 ~ 29,  
 flourish == 34 ~ 30,  
 flourish == 33 ~ 31,  
 flourish == 32 ~ 32,  
 flourish == 31 ~ 33,  
 flourish == 30 ~ 34,  
 flourish == 29 ~ 35,  
 flourish == 28 ~ 36,  
 flourish == 27 ~ 37,  
 flourish == 26 ~ 38,  
 flourish == 25 ~ 39,  
 flourish == 24 ~ 40,  
 flourish == 23 ~ 41,  
 flourish == 22 ~ 42,  
 flourish == 21 ~ 43,  
 flourish == 20 ~ 44,  
 flourish == 19 ~ 45,  
 flourish == 18 ~ 46,  
 flourish == 17 ~ 47,  
 flourish == 16 ~ 48,  
 flourish == 15 ~ 49,  
 flourish == 14 ~ 50,  
 flourish == 13 ~ 51,  
 flourish == 12 ~ 52,  
 flourish == 11 ~ 53,  
 flourish == 10 ~ 54,  
 flourish == 9 ~ 55,  
 flourish == 8 ~ 56)) %>%  
 filter(!is.na(flourish))  
  
traindf.v23$flourish <- as.numeric(traindf.v23$flourish)  
table(traindf.v23$flourish, useNA = "always")

##   
## 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23   
## 231 145 168 195 257 297 295 354 623 416 410 327 286 288 221 214   
## 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39   
## 203 188 154 146 152 106 107 101 81 99 61 59 33 37 38 25   
## 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55   
## 26 25 23 12 16 10 14 5 12 3 2 6 1 8 5 2   
## 56 <NA>   
## 18 0

## Informal Mental Health Support

traindf.v23$inf\_1[is.na(traindf.v23$inf\_1)] <- 0  
traindf.v23$inf\_2[is.na(traindf.v23$inf\_2)] <- 0  
traindf.v23$inf\_3[is.na(traindf.v23$inf\_3)] <- 0  
traindf.v23$inf\_4[is.na(traindf.v23$inf\_4)] <- 0  
traindf.v23$inf\_5[is.na(traindf.v23$inf\_5)] <- 0  
traindf.v23$inf\_6[is.na(traindf.v23$inf\_6)] <- 0  
traindf.v23$inf\_7[is.na(traindf.v23$inf\_7)] <- 0  
  
# change inf\_ var to numeric  
traindf.v24 <- traindf.v23 %>%   
 # create dichotomous variable for support  
 mutate(inf = case\_when(inf\_8 == 1 ~ 0, # no informal support  
 inf\_1 == 1 ~ 1, # support  
 inf\_2 == 1 ~ 1,  
 inf\_3 == 1 ~ 1,  
 inf\_4 == 1 ~ 1,  
 inf\_5 == 1 ~ 1,  
 inf\_6 == 1 ~ 1,  
 inf\_7 == 1 ~ 1)) %>%  
 filter(!is.na(inf)) # filtters out all NAs  
traindf.v24$inf <- as.logical(traindf.v24$inf)  
  
table(traindf.v24$inf, useNA ="always") # check output, skew

##   
## FALSE TRUE <NA>   
## 1742 4754 0

jarque.bera.test(traindf.v24$inf)

##   
## Jarque Bera Test  
##   
## data: traindf.v24$inf  
## X-squared = 1407.5, df = 2, p-value < 2.2e-16

traindf.v24 = subset(traindf.v24, select= -c(inf\_1:inf\_help)) # clean up vars

## BRS

traindf.v24 <- traindf.v24 %>%  
 mutate(brs\_tot = as.integer(BRS\_1) + as.integer(BRS\_2) + as.integer(BRS\_3) + as.integer(BRS\_4) + as.integer(BRS\_5))  
  
table(traindf.v24$brs\_tot)

##   
## 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20   
## 26 2 1 3 11 39 87 188 468 1134 1465 1589 869 347 134 62   
## 21 22 23 24 25   
## 24 8 5 1 3

traindf.v24$brs\_tot <- as.numeric(traindf.v24$brs\_tot)  
  
# for loop for predictive cutoff point needed  
rs\_df <- data.frame("cutoff" = c(), "rs" = c())  
for (i in 6:24) {  
 brs\_bin <- ifelse(traindf.v24$brs\_tot >= i, TRUE, FALSE)  
 model = lm(traindf.v24$sui\_idea ~ brs\_bin)  
 rs\_df <- rbind(rs\_df, c(i, summary(model)$r.squared))  
}  
  
names(rs\_df) <- c("cutoff", "rs")  
rs\_df\_arranged <- rs\_df %>%   
 arrange(desc(rs))   
  
# deem 'rs\_df\_arranged[1,1]' as SI cutoff  
traindf.v24 <- traindf.v24 %>%  
 mutate(brs\_dum = ifelse(brs\_tot > rs\_df\_arranged[1,1], 1, 0)) %>%  
 mutate(brs\_dum = case\_when(brs\_dum == 1 ~ 1,  
 brs\_dum == 0 ~ 0))  
traindf.v24$brs\_dum <- as.logical(traindf.v24$brs\_dum)  
  
# output  
rs\_df\_arranged[1,1]

## [1] 15

table(traindf.v24$brs\_tot, useNA = "always")

##   
## 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20   
## 26 2 1 3 11 39 87 188 468 1134 1465 1589 869 347 134 62   
## 21 22 23 24 25 <NA>   
## 24 8 5 1 3 30

table(traindf.v24$brs\_dum, useNA = "always")

##   
## FALSE TRUE <NA>   
## 3424 3042 30

## Distress & social support

traindf.v24$talk1\_1[is.na(traindf.v24$talk1\_1)] <- 1  
traindf.v24$talk1\_2[is.na(traindf.v24$talk1\_2)] <- 1  
traindf.v24$talk1\_3[is.na(traindf.v24$talk1\_3)] <- 1  
traindf.v24$talk1\_4[is.na(traindf.v24$talk1\_4)] <- 1  
traindf.v24$talk1\_5[is.na(traindf.v24$talk1\_5)] <- 1  
traindf.v24$talk1\_6[is.na(traindf.v24$talk1\_6)] <- 1  
traindf.v24$talk1\_7[is.na(traindf.v24$talk1\_7)] <- 1  
traindf.v24$talk1\_8[is.na(traindf.v24$talk1\_8)] <- 1  
  
traindf.v25 <- traindf.v24 %>%   
 # create dictomous variable for no talking or talking to support systems  
 mutate(talk = case\_when((talk1\_9 == 1 | talk1no == 1) ~ 1, # no support  
 (talk1\_2 == 1 | talk1\_3 == 1 | talk1\_4 == 1 | talk1\_5 == 1) ~ 0, # informal support  
 (talk1\_1 == 1 | talk1\_6 == 1 | talk1\_7 == 1 | talk1\_8 == 1) ~ 0)) # formal and informal  
traindf.v25$talk <- as.logical(traindf.v25$talk)  
  
table(traindf.v25$talk, useNA = "always") # check output

##   
## FALSE TRUE <NA>   
## 6100 396 0

traindf.v25 = subset(traindf.v25, select= -c(talk1\_1:talk1\_9, talk1sig, talk1fr,   
talk1rel, talk1no, talk1sup, talk1rm, talk1pro, talk1fam)) # clean up

## Perceived MH as failure, stigma

# original likert  
table(traindf.v25$stig\_pcv\_2)

##   
## 1 2 3 4 5 6   
## 876 2220 1628 676 219 64

table(traindf.v25$stig\_pcv\_3)

##   
## 1 2 3 4 5 6   
## 270 855 1564 1198 1346 449

table(traindf.v25$stig\_per\_2)

##   
## 1 2 3 4 5 6   
## 3506 1689 369 75 31 14

table(traindf.v25$stig\_per\_3)

##   
## 1 2 3 4 5 6   
## 76 169 360 486 1509 3081

# transform into y/n   
traindf.v26 <- traindf.v25 %>%  
 mutate(stig\_pcv2 = case\_when((stig\_pcv\_2 == 1 | stig\_pcv\_2 == 2 | stig\_pcv\_2 == 3) ~ 1,  
 (stig\_pcv\_2 == 4 | stig\_pcv\_2 == 5 | stig\_pcv\_2 == 6) ~ 0)) %>%  
 mutate(stig\_pcv3 = case\_when((stig\_pcv\_3 == 1 | stig\_pcv\_3 == 2 | stig\_pcv\_3 == 3) ~ 1,  
 (stig\_pcv\_3 == 4 | stig\_pcv\_3 == 5 | stig\_pcv\_3 == 6) ~ 0)) %>%   
 mutate(stig\_per2 = case\_when((stig\_per\_2 == 1 | stig\_per\_2 == 2 | stig\_per\_2 == 3) ~ 1,  
 (stig\_per\_2 == 4 | stig\_per\_2 == 5 | stig\_per\_2 == 6) ~ 0)) %>%  
 mutate(stig\_per3 = case\_when((stig\_per\_3 == 1 | stig\_per\_3 == 2 | stig\_per\_3 == 3) ~ 1,  
 (stig\_per\_3 == 4 | stig\_per\_3 == 5 | stig\_per\_3 == 6) ~ 0))   
  
traindf.v26$stig\_pcv2 <- as.logical(traindf.v26$stig\_pcv2)  
traindf.v26$stig\_pcv3 <- as.logical(traindf.v26$stig\_pcv3)  
traindf.v26$stig\_per2 <- as.logical(traindf.v26$stig\_per2)  
traindf.v26$stig\_per3 <- as.logical(traindf.v26$stig\_per3)  
  
table(traindf.v26$stig\_pcv2) # Most people feel that MH tx is a sign of personal failure.

##   
## FALSE TRUE   
## 959 4724

table(traindf.v26$stig\_pcv3) # Most people think less of a person who has received MH tx.

##   
## FALSE TRUE   
## 2993 2689

table(traindf.v26$stig\_per2) # I feel that receiving MH tx is a sign of personal failure.

##   
## FALSE TRUE   
## 120 5564

table(traindf.v26$stig\_per3) # I would think less of a person who has received MH tx.

##   
## FALSE TRUE   
## 5076 605

# combine sstigma questions into one binary variable  
traindf.v26 <- traindf.v26 %>%  
 mutate(mh\_stigma = case\_when(stig\_pcv2 == 1 ~ 1,  
 stig\_pcv3 == 1 ~ 1,  
 stig\_per2 == 1 ~ 1,  
 stig\_per3 == 1 ~ 1,  
 stig\_pcv2 == 0 ~ 0,  
 stig\_pcv3 == 0 ~ 0,  
 stig\_per2 == 0 ~ 0,  
 stig\_per3 == 0 ~ 0)) %>%  
 filter(!is.na(mh\_stigma)) # filters out NAs  
traindf.v26$mh\_stigma <- as.logical(traindf.v26$mh\_stigma)  
  
table(traindf.v26$mh\_stigma, useNA = "always") # check out, NAs

##   
## FALSE TRUE <NA>   
## 34 5657 0

## Therapy

# create new var for medication and/or therapy utilization  
traindf.v26 <- traindf.v26 %>%  
 mutate(ther = case\_when(ther\_cur == 1 ~ 1,  
 ther\_cur1 == 1 ~ 1,  
 ther\_ever == 3 ~ 1,  
 ther\_ever == 4 ~ 1,  
 ther\_cur == 0 ~ 0,  
 ther\_ever == 0 ~ 0)) %>%  
 filter(!is.na(ther))  
traindf.v26$ther <- as.logical(traindf.v26$ther)  
  
table(traindf.v26$ther, useNA = "always")

##   
## FALSE TRUE <NA>   
## 3767 1924 0

## Threats analyses

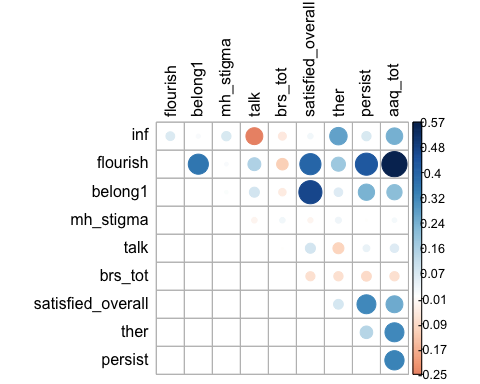
traindf.v27 <- traindf.v26

### Threats correlation

# Define MM variable candidates for correlation matrix  
traindf\_mm <- traindf.v27 %>% select(inf, flourish, belong1, mh\_stigma, talk, brs\_tot,   
 satisfied\_overall, ther, persist, aaq\_tot)  
  
 # convert traindf to numeric in order to run correlations  
traindf\_mm <- traindf\_mm %>% mutate\_if(is.character, as.factor)  
traindf\_mm <- traindf\_mm %>% mutate\_if(is.factor, as.numeric)  
  
# run a correlation and drop the insignificant ones  
corr <- cor(traindf\_mm, use = "pairwise.complete.obs")   
corr[lower.tri(corr,diag=TRUE)] <- NA # prepare to drop duplicates and correlations of 1   
corr[corr %in% c(-1, 1)] <- NA # drop perfect correlations  
corr <- as.data.frame(as.table(corr)) # turn into a 3-column table  
corr <- na.omit(corr) # remove the NA values from above   
corr <- corr[order(-abs(corr$Freq)),] # sort by highest correlation  
print(corr)

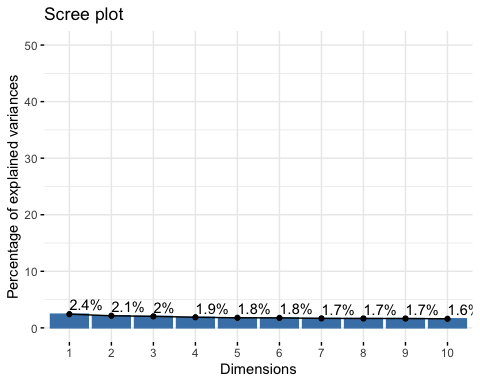
## Var1 Var2 Freq  
## 92 flourish aaq\_tot 0.566170513  
## 63 belong1 satisfied\_overall 0.475121390  
## 82 flourish persist 0.434417997  
## 62 flourish satisfied\_overall 0.396847642  
## 22 flourish belong1 0.359701784  
## 99 persist aaq\_tot 0.329657960  
## 98 ther aaq\_tot 0.321997473  
## 87 satisfied\_overall persist 0.319794196  
## 71 inf ther 0.271230553  
## 41 inf talk -0.254636776  
## 97 satisfied\_overall aaq\_tot 0.254583715  
## 91 inf aaq\_tot 0.243078160  
## 83 belong1 persist 0.236190948  
## 93 belong1 aaq\_tot 0.207582119  
## 72 flourish ther 0.176444096  
## 42 flourish talk 0.148248104  
## 88 ther persist 0.140311485  
## 52 flourish brs\_tot -0.118150524  
## 75 talk ther -0.107141138  
## 65 talk satisfied\_overall 0.090057873  
## 43 belong1 talk 0.089052310  
## 86 brs\_tot persist -0.085252530  
## 77 satisfied\_overall ther 0.084374721  
## 31 inf mh\_stigma 0.077027893  
## 76 brs\_tot ther -0.076343461  
## 96 brs\_tot aaq\_tot -0.075959247  
## 81 inf persist 0.075704965  
## 66 brs\_tot satisfied\_overall -0.073858255  
## 11 inf flourish 0.068047937  
## 73 belong1 ther 0.064803326  
## 95 talk aaq\_tot 0.063430085  
## 51 inf brs\_tot -0.056185422  
## 53 belong1 brs\_tot -0.049047941  
## 85 talk persist 0.040291334  
## 74 mh\_stigma ther 0.031304792  
## 44 mh\_stigma talk -0.026870427  
## 54 mh\_stigma brs\_tot 0.025955225  
## 61 inf satisfied\_overall 0.025452202  
## 64 mh\_stigma satisfied\_overall -0.023472747  
## 94 mh\_stigma aaq\_tot 0.018252771  
## 21 inf belong1 0.014401531  
## 32 flourish mh\_stigma 0.011382749  
## 33 belong1 mh\_stigma 0.008977791  
## 84 mh\_stigma persist -0.003051071  
## 55 talk brs\_tot -0.001475572

mtx\_corr <- acast(corr, Var1~Var2, value.var="Freq")   
corrplot(mtx\_corr, is.corr=FALSE, tl.col="black", na.label=" ")



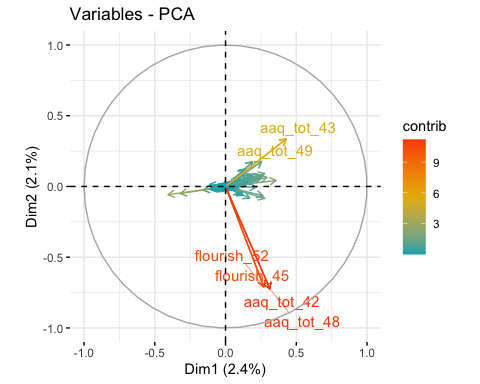
### PCA MMTSM

traindf\_mmtsm <- traindf.v27 %>% select(inf, flourish, belong1, talk, mh\_stigma, persist,   
 satisfied\_overall, aaq\_tot, ther)  
traindf\_mmtsm <- na.omit(traindf\_mmtsm)  
  
# create corr matrix, with dummy vars   
traindf\_mmtsm <- traindf\_mmtsm %>%   
 mutate\_if(is.numeric, as.factor) %>%   
 dummy\_cols(remove\_first\_dummy = TRUE) %>%   
 data.matrix()  
# traindf\_mmtsm # check output  
results\_mmtsm <- dummy\_cols(traindf\_mmtsm, select\_columns = c("flourish", "belong1", "inf", "ther",  
 "mh\_stigma", "talk", "persist", "aaq\_tot",  
 "satisfied\_overall"),  
 remove\_first\_dummy = TRUE, remove\_selected\_columns = TRUE)  
mmtsm\_pca <- PCA(results\_mmtsm, graph=FALSE) # apply PCA  
eigval\_mmtsm <- get\_eigenvalue(mmtsm\_pca)  
# eigval\_mmtsm  
  
# Visualization of PCA, FAMD  
fviz\_screeplot(mmtsm\_pca, addlabels = TRUE, ylim = c(0, 50))



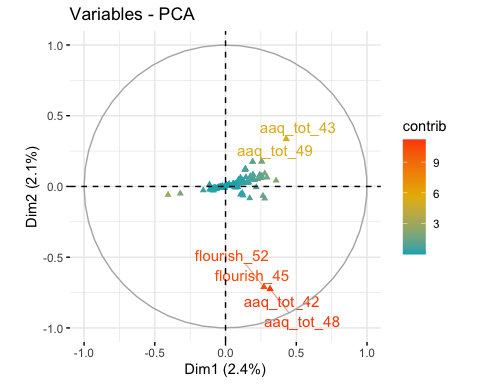
fviz\_pca\_var(mmtsm\_pca,  
 col.var = "contrib", # Color by contributions to the PC  
 gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),  
 repel = TRUE # Avoid text overlapping  
 )

## Warning: ggrepel: 116 unlabeled data points (too many overlaps). Consider  
## increasing max.overlaps

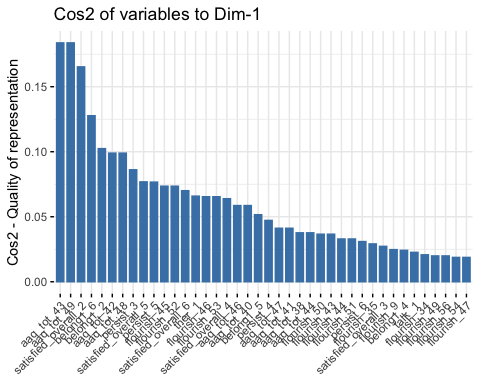


fviz\_famd\_var(mmtsm\_pca,  
 col.var = "contrib", # Color by contributions to the PC  
 gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),  
 repel = TRUE # Avoid text overlapping  
 )

## Warning: ggrepel: 116 unlabeled data points (too many overlaps). Consider  
## increasing max.overlaps



fviz\_cos2(mmtsm\_pca, choice = "var", axes = 1,   
 fill = "steelblue", color = "steelblue",  
 sort.val = c("desc", "asc", "none"),  
 top = 40,  
 xtickslab.rt = 45,  
 ggtheme = theme\_minimal())



### GLM Threats regression

# prepare data for regression  
glm\_mmtsm <- traindf.v27 %>% select(si\_type, inf, flourish, belong1, talk, mh\_stigma, persist,   
 satisfied\_overall, ther, aaq\_tot)  
glm\_mmtsm <- na.omit(glm\_mmtsm)  
  
# run MM regression  
glm\_mmtsm <- glm(si\_type ~ inf + flourish + belong1 + mh\_stigma + talk + persist +   
 satisfied\_overall + aaq\_tot + ther,   
 data = glm\_mmtsm)  
  
summary(glm\_mmtsm)

##   
## Call:  
## glm(formula = si\_type ~ inf + flourish + belong1 + mh\_stigma +   
## talk + persist + satisfied\_overall + aaq\_tot + ther, data = glm\_mmtsm)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.66198 -0.18637 -0.07388 0.04125 1.11404   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.3664985 0.0560945 -6.534 6.98e-11 \*\*\*  
## infTRUE 0.0138599 0.0104068 1.332 0.1830   
## flourish 0.0073454 0.0006918 10.618 < 2e-16 \*\*\*  
## belong1 -0.0012826 0.0037641 -0.341 0.7333   
## mh\_stigmaTRUE 0.1012082 0.0546969 1.850 0.0643 .   
## talkTRUE 0.0446428 0.0182574 2.445 0.0145 \*   
## persist 0.0324527 0.0051923 6.250 4.40e-10 \*\*\*  
## satisfied\_overall -0.0083774 0.0046984 -1.783 0.0746 .   
## aaq\_tot 0.0086745 0.0005648 15.358 < 2e-16 \*\*\*  
## therTRUE 0.0618767 0.0096316 6.424 1.43e-10 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.1003503)  
##   
## Null deviance: 703.98 on 5690 degrees of freedom  
## Residual deviance: 570.09 on 5681 degrees of freedom  
## AIC: 3078.2  
##   
## Number of Fisher Scoring iterations: 2

coef\_mmtsm <- coef(summary(glm\_mmtsm)) # retrieve coefficients  
  
ci\_mmtsm <- confint(glm\_mmtsm) # retrieve confidence intervals

## Waiting for profiling to be done...

ci\_mmtsm

## 2.5 % 97.5 %  
## (Intercept) -0.476441703 -0.2565552984  
## infTRUE -0.006536930 0.0342567887  
## flourish 0.005989525 0.0087012765  
## belong1 -0.008660141 0.0060948863  
## mh\_stigmaTRUE -0.005995864 0.2084121819  
## talkTRUE 0.008858848 0.0804267086  
## persist 0.022275996 0.0426293904  
## satisfied\_overall -0.017586172 0.0008313188  
## aaq\_tot 0.007567449 0.0097815242  
## therTRUE 0.042999142 0.0807541614

model.mmtsm <- tidy(glm\_mmtsm) # Convert model to dataframe for easy manipulation  
model.mmtsm

## # A tibble: 10 x 5  
## term estimate std.error statistic p.value  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) -0.366 0.0561 -6.53 6.98e-11  
## 2 infTRUE 0.0139 0.0104 1.33 1.83e- 1  
## 3 flourish 0.00735 0.000692 10.6 4.31e-26  
## 4 belong1 -0.00128 0.00376 -0.341 7.33e- 1  
## 5 mh\_stigmaTRUE 0.101 0.0547 1.85 6.43e- 2  
## 6 talkTRUE 0.0446 0.0183 2.45 1.45e- 2  
## 7 persist 0.0325 0.00519 6.25 4.40e-10  
## 8 satisfied\_overall -0.00838 0.00470 -1.78 7.46e- 2  
## 9 aaq\_tot 0.00867 0.000565 15.4 3.47e-52  
## 10 therTRUE 0.0619 0.00963 6.42 1.43e-10

model.mmtsm %>%  
 mutate(or = exp(estimate), # Odds ratio/gradient  
 var.diag = diag(vcov(glm\_mmtsm)), # Variance of each coefficient  
 or.se = sqrt(or^2 \* var.diag)) # Odds-ratio adjusted

## # A tibble: 10 x 8  
## term estimate std.error statistic p.value or var.diag or.se  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) -0.366 0.0561 -6.53 6.98e-11 0.693 3.15e-3 3.89e-2  
## 2 infTRUE 0.0139 0.0104 1.33 1.83e- 1 1.01 1.08e-4 1.06e-2  
## 3 flourish 0.00735 0.000692 10.6 4.31e-26 1.01 4.79e-7 6.97e-4  
## 4 belong1 -0.00128 0.00376 -0.341 7.33e- 1 0.999 1.42e-5 3.76e-3  
## 5 mh\_stigmaTRUE 0.101 0.0547 1.85 6.43e- 2 1.11 2.99e-3 6.05e-2  
## 6 talkTRUE 0.0446 0.0183 2.45 1.45e- 2 1.05 3.33e-4 1.91e-2  
## 7 persist 0.0325 0.00519 6.25 4.40e-10 1.03 2.70e-5 5.36e-3  
## 8 satisfied\_ove… -0.00838 0.00470 -1.78 7.46e- 2 0.992 2.21e-5 4.66e-3  
## 9 aaq\_tot 0.00867 0.000565 15.4 3.47e-52 1.01 3.19e-7 5.70e-4  
## 10 therTRUE 0.0619 0.00963 6.42 1.43e-10 1.06 9.28e-5 1.02e-2

### ALL regression

# prepare data for regression  
glm\_all <- traindf.v27 %>% select(  
 # outcome  
 si\_type,  
 # premotivation  
 age, sexual, race, gender\_tgd, sib\_freq, ed\_any, dx\_mh, meds\_any, finpast,  
 religios, residenc, school2\_type, international, undercls,  
 # life events  
 ins\_no, gpa\_sr, psyhx, fincur, drugs\_yn, assault, binge, drug\_mar,  
 # defeat-entrapment  
 deprawsc, anx\_score, dep\_impa, percneed,  
 # threats to internal/external moderation  
 inf, flourish, belong1, talk, mh\_stigma, persist, satisfied\_overall, aaq\_tot, ther)  
  
glm\_all <- na.omit(glm\_all)  
  
# run MM regression  
glm\_all <- glm(si\_type ~ .,   
 data = glm\_all)  
  
summary(glm\_all)

##   
## Call:  
## glm(formula = si\_type ~ ., data = glm\_all)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.80403 -0.15645 -0.04700 0.04648 1.10314   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.1189776 0.0866654 -1.373 0.169857   
## age -0.0080263 0.0031140 -2.577 0.009978 \*\*   
## sexualTRUE 0.0418623 0.0104646 4.000 6.40e-05 \*\*\*  
## race 0.0051708 0.0032908 1.571 0.116171   
## gender\_tgdTRUE 0.0643994 0.0230496 2.794 0.005224 \*\*   
## sib\_freqTRUE 0.1646736 0.0130396 12.629 < 2e-16 \*\*\*  
## ed\_any1 0.0171302 0.0132326 1.295 0.195530   
## dx\_mhTRUE 0.0165254 0.0114988 1.437 0.150736   
## meds\_any1 0.0275240 0.0117916 2.334 0.019620 \*   
## finpast 0.0051224 0.0038656 1.325 0.185181   
## religios -0.0036219 0.0031268 -1.158 0.246766   
## residenc 0.0097150 0.0041360 2.349 0.018863 \*   
## school2\_type2 0.0235701 0.0210470 1.120 0.262813   
## school2\_type3 -0.0702418 0.0245014 -2.867 0.004161 \*\*   
## internationalTRUE -0.0201307 0.0177417 -1.135 0.256568   
## underclsTRUE -0.0113076 0.0128205 -0.882 0.377816   
## ins\_noTRUE -0.0012334 0.0098714 -0.125 0.900569   
## gpa\_sr 0.0053129 0.0035546 1.495 0.135057   
## psyhxTRUE 0.3583672 0.0341763 10.486 < 2e-16 \*\*\*  
## fincurTRUE 0.0024483 0.0094177 0.260 0.794902   
## drugs\_ynTRUE 0.0429197 0.0215385 1.993 0.046343 \*   
## assaultTRUE 0.0261277 0.0086779 3.011 0.002617 \*\*   
## bingeTRUE -0.0015210 0.0123341 -0.123 0.901864   
## drug\_marTRUE 0.0124247 0.0114113 1.089 0.276285   
## deprawsc 0.0114018 0.0012439 9.166 < 2e-16 \*\*\*  
## anx\_score -0.0042889 0.0011229 -3.819 0.000135 \*\*\*  
## dep\_impa -0.0040770 0.0074398 -0.548 0.583714   
## percneed 0.0147115 0.0032359 4.546 5.57e-06 \*\*\*  
## infTRUE -0.0094842 0.0102620 -0.924 0.355420   
## flourish 0.0031219 0.0006976 4.475 7.79e-06 \*\*\*  
## belong1 -0.0017444 0.0037220 -0.469 0.639329   
## talkTRUE 0.0319051 0.0173644 1.837 0.066205 .   
## mh\_stigmaTRUE 0.0748973 0.0518198 1.445 0.148418   
## persist 0.0184442 0.0050336 3.664 0.000250 \*\*\*  
## satisfied\_overall -0.0059239 0.0045107 -1.313 0.189139   
## aaq\_tot 0.0026144 0.0006709 3.897 9.87e-05 \*\*\*  
## therTRUE 0.0062160 0.0104902 0.593 0.553506   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.08949745)  
##   
## Null deviance: 703.98 on 5690 degrees of freedom  
## Residual deviance: 506.02 on 5654 degrees of freedom  
## AIC: 2453.8  
##   
## Number of Fisher Scoring iterations: 2

coef\_mm <- coef(summary(glm\_all)) # retrieve coefficients  
  
ci\_all <- confint(glm\_all) # retrieve confidence intervals

## Waiting for profiling to be done...

ci\_all

## 2.5 % 97.5 %  
## (Intercept) -0.2888386417 0.050883531  
## age -0.0141297105 -0.001922866  
## sexualTRUE 0.0213521482 0.062372548  
## race -0.0012790422 0.011620700  
## gender\_tgdTRUE 0.0192230428 0.109575814  
## sib\_freqTRUE 0.1391163571 0.190230765  
## ed\_any1 -0.0088052366 0.043065714  
## dx\_mhTRUE -0.0060119689 0.039062676  
## meds\_any1 0.0044129147 0.050635026  
## finpast -0.0024540127 0.012698884  
## religios -0.0097502383 0.002506422  
## residenc 0.0016086198 0.017821424  
## school2\_type2 -0.0176812853 0.064821516  
## school2\_type3 -0.1182637364 -0.022219879  
## internationalTRUE -0.0549038753 0.014642388  
## underclsTRUE -0.0364353014 0.013820057  
## ins\_noTRUE -0.0205809009 0.018114085  
## gpa\_sr -0.0016539274 0.012279671  
## psyhxTRUE 0.2913828037 0.425351552  
## fincurTRUE -0.0160101618 0.020906685  
## drugs\_ynTRUE 0.0007049535 0.085134414  
## assaultTRUE 0.0091192547 0.043136170  
## bingeTRUE -0.0256953182 0.022653410  
## drug\_marTRUE -0.0099409757 0.034790340  
## deprawsc 0.0089638275 0.013839759  
## anx\_score -0.0064897604 -0.002087960  
## dep\_impa -0.0186587160 0.010504724  
## percneed 0.0083693362 0.021053727  
## infTRUE -0.0295972797 0.010628970  
## flourish 0.0017545731 0.004489155  
## belong1 -0.0090394072 0.005550660  
## talkTRUE -0.0021284958 0.065938630  
## mh\_stigmaTRUE -0.0266677774 0.176462293  
## persist 0.0085785531 0.028309802  
## satisfied\_overall -0.0147646831 0.002916971  
## aaq\_tot 0.0012993622 0.003929417  
## therTRUE -0.0143444394 0.026776380

model.all <- tidy(glm\_all) # Convert model to dataframe for easy manipulation  
model.all

## # A tibble: 37 x 5  
## term estimate std.error statistic p.value  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) -0.119 0.0867 -1.37 1.70e- 1  
## 2 age -0.00803 0.00311 -2.58 9.98e- 3  
## 3 sexualTRUE 0.0419 0.0105 4.00 6.40e- 5  
## 4 race 0.00517 0.00329 1.57 1.16e- 1  
## 5 gender\_tgdTRUE 0.0644 0.0230 2.79 5.22e- 3  
## 6 sib\_freqTRUE 0.165 0.0130 12.6 4.49e-36  
## 7 ed\_any1 0.0171 0.0132 1.29 1.96e- 1  
## 8 dx\_mhTRUE 0.0165 0.0115 1.44 1.51e- 1  
## 9 meds\_any1 0.0275 0.0118 2.33 1.96e- 2  
## 10 finpast 0.00512 0.00387 1.33 1.85e- 1  
## # … with 27 more rows

model.all %>%  
 mutate(or = exp(estimate), # Odds ratio/gradient  
 var.diag = diag(vcov(glm\_all)), # Variance of each coefficient  
 or.se = sqrt(or^2 \* var.diag)) # Odds-ratio adjusted

## # A tibble: 37 x 8  
## term estimate std.error statistic p.value or var.diag or.se  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) -0.119 0.0867 -1.37 1.70e- 1 0.888 0.00751 0.0769   
## 2 age -0.00803 0.00311 -2.58 9.98e- 3 0.992 0.00000970 0.00309  
## 3 sexualTRUE 0.0419 0.0105 4.00 6.40e- 5 1.04 0.000110 0.0109   
## 4 race 0.00517 0.00329 1.57 1.16e- 1 1.01 0.0000108 0.00331  
## 5 gender\_tgdTRUE 0.0644 0.0230 2.79 5.22e- 3 1.07 0.000531 0.0246   
## 6 sib\_freqTRUE 0.165 0.0130 12.6 4.49e-36 1.18 0.000170 0.0154   
## 7 ed\_any1 0.0171 0.0132 1.29 1.96e- 1 1.02 0.000175 0.0135   
## 8 dx\_mhTRUE 0.0165 0.0115 1.44 1.51e- 1 1.02 0.000132 0.0117   
## 9 meds\_any1 0.0275 0.0118 2.33 1.96e- 2 1.03 0.000139 0.0121   
## 10 finpast 0.00512 0.00387 1.33 1.85e- 1 1.01 0.0000149 0.00389  
## # … with 27 more rows

# MISSINGNESS

## Absolute missingness, by school

traindf.v28 <- traindf.v26  
# school structural analysis  
var\_names <- c(  
 # premotivation  
 "age", "sexual", "race", "gender\_tgd", "sib\_freq", "ed\_any", "dx\_mh", "meds\_any", "finpast",  
 "religios", "residenc", "school2\_type", "international", "undercls",  
 # life events  
 "ins\_no", "gpa\_sr", "psyhx", "fincur", "drugs\_yn", "assault", "binge", "drug\_mar",  
 # defeat-entrapment  
 "deprawsc", "anx\_score", "dep\_impa", "percneed",  
 # threats to internal/external moderation  
 "persist", "aaq\_tot", "flourish", "satisfied\_overall", "talk", "ther")   
  
var\_names <- sort(var\_names)  
  
# build table for absolute missingness based on school2  
school2 <- count(traindf.v28, school2)  
school2\_ncol <- ncol(school2)  
  
for (j in 1:length(var\_names)) {  
 school2[, school2\_ncol + j] <- 999  
}  
  
# create column headers and vectors for school, total students, then each variable count  
names <- c("school2\_values", "total", var\_names)  
names(school2) <- names  
  
# input values for each school missingness across all variables  
for (j in 1:(length(var\_names))) {  
for (i in 1:nrow(school2)) { # i corresponds to rows, j to columns  
 school2[i, 2 + j] <- sum(!is.na(traindf.v28[traindf.v28$school2 == school2[[i,1]], var\_names[j]]))  
}  
}  
# add percentage in each column  
school2\_m <- as.matrix(school2[-1])   
props <- school2\_m[,2:ncol(school2\_m)] / school2\_m[,1]  
school2 <- data.frame(school2[, 1:2], props)  
  
# view missingness matrix  
# view(school2)  
  
# save matrix  
write\_xlsx(school2, path = "sch2\_missing.xlsx", col\_names = TRUE, format\_headers = TRUE)

## Missingness of each var w/in each var

missing\_dfs <- list()  
for (k in 1:length(var\_names)) {  
  
#create skeleton of missingness matrix  
var <- count(traindf.v28, traindf.v28[var\_names[k]])  
var\_ncol <- ncol(var)  
  
for (j in 1:length(var\_names)) {  
 var[, var\_ncol + j] <- 999  
}  
# create column headers and vectors for school, total students, then each variable count  
names <- c(paste0(var\_names[k], "\_values"), "total", var\_names)  
names(var) <- names  
  
# input values for each school missingness across all variables  
for (j in 1:(length(var\_names))) {  
for (i in 1:nrow(var)) { # i corresponds to rows, j to columns  
 var[i, 2 + j] <- sum(!is.na(traindf.v28[traindf.v28[[var\_names[k]]] == var[[i,1]], var\_names[j]]))  
}  
}  
# add percentage in each column  
var\_m <- as.matrix(var[-1])   
props <- var\_m[,2:ncol(var\_m)] / var\_m[,1]  
var <- data.frame(var[, 1:2], props)  
replace\_na("NA")  
missing\_dfs[[k]] <- var  
}  
  
names(missing\_dfs) <- var\_names  
save(missing\_dfs, file = "missing\_dfs.RData")  
load("missing\_dfs.RData")  
write\_xlsx(missing\_dfs, path = "missing.xlsx", col\_names = TRUE, format\_headers = TRUE)

# DATA PREP

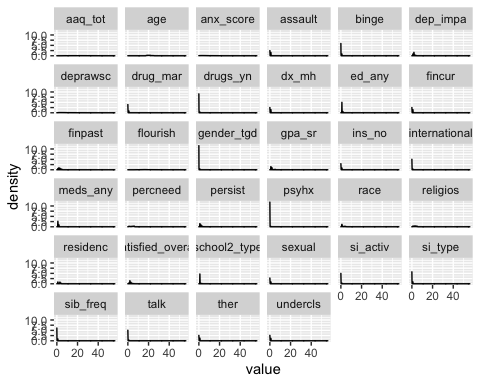
## Select all vars for SEM

# Prep SEM data vars  
traindf\_sem <- traindf.v28 %>% select(  
 # outcome var  
 si\_type, si\_activ,  
 # premotivation  
 age, sexual, race, gender\_tgd, sib\_freq, ed\_any, dx\_mh, meds\_any, finpast,  
 religios, residenc, school2\_type, international, undercls, ins\_no,   
 # life events  
 gpa\_sr, psyhx, fincur, drugs\_yn, drug\_mar, assault, binge,  
 # defeat-entrapment  
 deprawsc, anx\_score, dep\_impa, percneed,  
 # threats to internal/external moderation  
 persist, aaq\_tot, flourish, talk, ther, satisfied\_overall)   
  
# create sem matrix   
traindf\_sem <- traindf\_sem %>%   
 # mutate\_if(is.numeric, as.factor) %>%   
 data.matrix()  
# traindf\_sem <- na.omit(traindf\_sem)  
# view(traindf\_sem) # check output  
  
# # xlsx output  
traindf\_sem\_out <- as.data.frame(traindf\_sem)  
save(traindf\_sem\_out , file = "traindf\_nondummy.RData")  
load("traindf\_nondummy.RData")  
write\_xlsx(traindf\_sem\_out, path = "traindf\_nondummy.xls", col\_names = TRUE, format\_headers = TRUE)

# SUMMARY STATISTICS

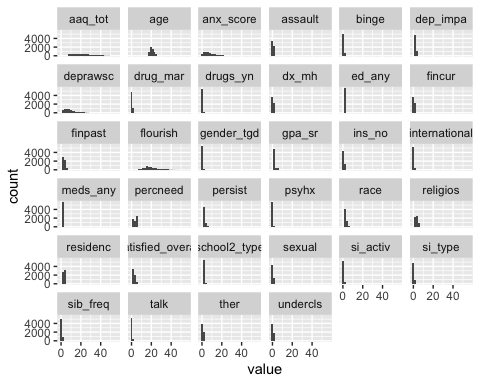
## Descriptive stats

data <- traindf\_sem\_out %>%   
 pivot\_longer(cols = everything())  
  
ggplot(data, aes(x = value)) +   
 geom\_density() +  
 facet\_wrap(~name)

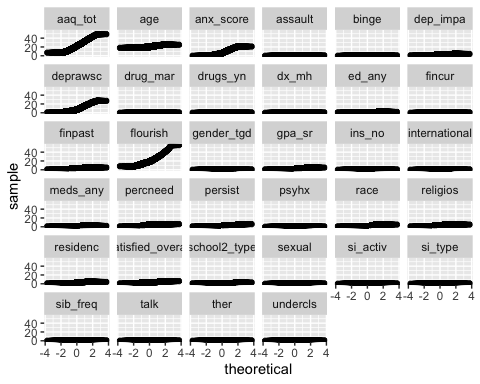


ggplot(data, aes(x = value)) +   
 geom\_histogram() +  
 facet\_wrap(~name)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



ggplot(data, aes(sample = value)) +   
 geom\_qq() +  
 facet\_wrap(~name)



# demographics  
table(traindf.v28$age)

##   
## 18 19 20 21 22 23 24 25   
## 767 1023 1104 1013 612 419 426 327

table(traindf.v28$si\_type)

##   
## FALSE TRUE   
## 4868 823

table(traindf.v28$gender\_tgd)

##   
## FALSE TRUE   
## 5493 198

table(traindf.v28$sexual)

##   
## FALSE TRUE   
## 4261 1430

table(traindf.v28$race)

##   
## 1 2 3 4 5   
## 3665 364 569 836 257

table(traindf.v28$dx\_mh)

##   
## FALSE TRUE   
## 3523 2168

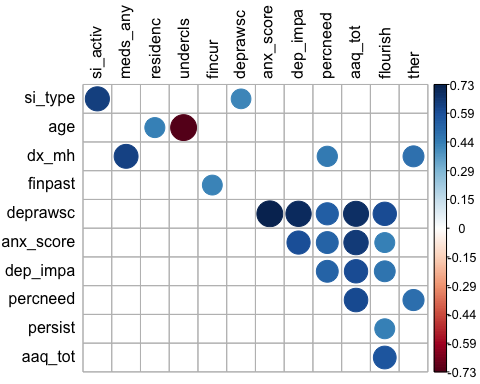
# training data  
var\_desc <- describe(traindf\_sem)  
view(var\_desc)  
  
# xlsx output  
save(var\_desc, file = "var\_desc.RData")  
load("var\_desc.RData")  
write\_xlsx(var\_desc, path = "var\_desc.xls", col\_names = TRUE, format\_headers = TRUE)

## Model w/ levels correlations

results\_corr <- NA\_preproc(traindf\_sem)  
sig <- 0.4  
  
# run a correlation and drop the insignificant ones   
corr <- cor(results\_corr, use = "pairwise.complete.obs")   
corr[lower.tri(corr,diag=TRUE)] <- NA # prepare to drop duplicates and correlations of 1   
corr[corr %in% c(-1, 1)] <- NA # drop perfect correlations  
corr <- as.data.frame(as.table(corr)) # turn into a 3-column table  
corr <- na.omit(corr) # remove the NA values from above   
corr <- subset(corr, abs(Freq) > sig) # select significant values   
corr <- corr[order(-abs(corr$Freq)),] # sort by highest correlation  
print(corr)

## Var1 Var2 Freq  
## 513 age undercls -0.7328972  
## 875 deprawsc anx\_score 0.7317606  
## 909 deprawsc dep\_impa 0.7000313  
## 1011 deprawsc aaq\_tot 0.6845522  
## 1012 anx\_score aaq\_tot 0.6505136  
## 35 si\_type si\_activ 0.6357476  
## 315 dx\_mh meds\_any 0.6246419  
## 1014 percneed aaq\_tot 0.6060967  
## 1045 deprawsc flourish 0.5964479  
## 1013 dep\_impa aaq\_tot 0.5961905  
## 910 anx\_score dep\_impa 0.5869218  
## 1050 aaq\_tot flourish 0.5661705  
## 943 deprawsc percneed 0.5359562  
## 944 anx\_score percneed 0.5224797  
## 945 dep\_impa percneed 0.5212923  
## 1116 percneed ther 0.4868092  
## 1097 dx\_mh ther 0.4796261  
## 1047 dep\_impa flourish 0.4738001  
## 927 dx\_mh percneed 0.4478193  
## 411 age residenc 0.4378124  
## 1049 persist flourish 0.4344180  
## 1046 anx\_score flourish 0.4334504  
## 657 finpast fincur 0.4312425  
## 817 si\_type deprawsc 0.4239213

#turn corr back into matrix in order to plot with corrplot  
mtx\_corr <- acast(corr, Var1~Var2, value.var="Freq")   
corrplot(mtx\_corr, is.corr=FALSE, tl.col="black", na.label=" ")



# SEM ANALYSIS

## PLS output model

# Prep SEM data vars  
traindf\_sem1 <- traindf.v28 %>% select(  
 # outcome var  
 si\_type,  
 # premotivation  
 age, sexual, race, gender\_tgd, sib\_freq, ed\_any, dx\_mh, meds\_any, finpast,  
 religios, international, undercls, ins\_no,  
 # life events  
 gpa\_sr, psyhx, fincur, drugs\_yn, drug\_mar, assault, binge,  
 # defeat-entrapment  
 deprawsc, anx\_score, dep\_impa, percneed,  
 # threats to internal/external moderation  
 persist, aaq\_tot, flourish, talk, ther, satisfied\_overall)   
  
# create sem matrix   
traindf\_sem1 <- data.matrix(traindf\_sem1)  
   
## Dummy col SEM w/ continuous  
results\_tr\_com1 <- dummy\_cols(traindf\_sem1, select\_columns = c("si\_type", "race"),  
 remove\_first\_dummy = TRUE, remove\_selected\_columns = TRUE)  
  
# xlsx output  
save(results\_tr\_com1, file = "traindf\_pls.RData")  
load("traindf\_pls.RData")  
write\_xlsx(results\_tr\_com1, path = "traindf\_pls.xlsx", col\_names = TRUE, format\_headers = TRUE)

TESTING DATA

## AUDIT

\*\* too small for inclusion

testdf.v10 <- testdf.v10 %>%  
 # Combine audit\_3\_f, audit\_3\_m, audit\_3\_o to create 'audit\_3'  
 mutate(audit\_3 = case\_when(audit\_3\_f == 1 ~ 1,  
 audit\_3\_m == 1 ~ 1,  
 audit\_3\_o == 1 ~ 1,  
 audit\_3\_f == 0 ~ 0,  
 audit\_3\_m == 0 ~ 0,  
 audit\_3\_o == 0 ~ 0)) %>%  
 # All outputs needs to be numeric for future addition of AUDIT total  
 mutate(audit\_1 = as.numeric(audit\_1),  
 audit\_2 = as.numeric(audit\_2),  
 audit\_3 = as.numeric(audit\_3),  
 audit\_4 = as.numeric(audit\_4),  
 audit\_5 = as.numeric(audit\_5),  
 audit\_6 = as.numeric(audit\_5),  
 audit\_7 = as.numeric(audit\_7),  
 audit\_8 = as.numeric(audit\_8),  
 audit\_9 = as.numeric(audit\_9),  
 audit\_10 = as.numeric(audit\_10)) %>%  
 # Find audit\_tot -- total AUDIT scores for all participants   
 mutate(audit\_tot = as.numeric(audit\_1) + as.numeric(audit\_2) + as.numeric(audit\_3) +   
 as.numeric(audit\_4) + as.numeric(audit\_5) + as.numeric(audit\_6) + as.numeric(audit\_7) +   
 as.numeric(audit\_8) + as.numeric(audit\_9) + as.numeric(audit\_10))  
table(testdf.v10$audit\_tot) # check output (too small)

##   
## 2 3 4 5 6 7 8 10 22   
## 201 96 39 11 9 7 2 1 1

testdf.v10 = subset(testdf.v10, select= -c(audit\_1:audit\_10, audit\_3\_m,   
 audit\_3\_f, audit\_3\_o)) # clean up vars/columns

## Eating disorder, any type

testdf.v10 <- testdf.v10 %>%  
 filter(!is.na(ed\_any)) # filter out all NAs for complete data  
  
table(testdf.v10$ed\_any, useNA = "always") # check output, confirms no NAs

##   
## 0 1 <NA>   
## 16903 2141 0

## WCS

Combine wcs\_1\_f, wcs\_1\_m, wcs\_1\_o to create ‘wcs\_1’ \*\* too small

# check output (too small)  
table(testdf.v10$wcs\_1\_f)

##   
## 1 2 3 4 5   
## 116 207 434 316 182

table(testdf.v10$wcs\_1\_m)

##   
## 1 2 3 4 5   
## 83 105 220 161 53

table(testdf.v10$wcs\_1\_o)

##   
## 1 2 3 4 5   
## 4 7 7 9 11

table(testdf.v10$wcs\_2)

##   
## 1 2 3 4 5   
## 640 480 432 256 106

table(testdf.v10$wcs\_3)

##   
## 1 2 3 4 5 6 7   
## 1046 193 104 62 60 82 369

table(testdf.v10$wcs\_4)

##   
## 1 2 3 4   
## 644 932 323 17

table(testdf.v10$wcs\_5)

##   
## 1 2 3 4 5   
## 296 374 605 387 255

testdf.v10 = subset(testdf.v10, select= -c(wcs\_1\_f, wcs\_1\_m, wcs\_1\_o,   
 wcs\_2, wcs\_3, wcs\_4, wcs\_5))

## Medication use

# Create stimulant  
testdf.v11 <- testdf.v10 %>%  
 mutate(meds\_sti = case\_when(meds\_1 == 1 ~ 1, # combine stimulant variables  
 meds\_8 == 1 ~ 0,  
 meds\_1 == 0 ~ 0))  
 # Reconcile stimulant write-ins from meds\_7\_text  
 testdf.v11$meds\_sti[str\_detect(testdf.v11$meds\_7\_text, "[V|v]yvan[s|c]e")] <- 1  
 testdf.v11$meds\_sti[str\_detect(testdf.v11$meds\_7\_text, "[T|t]enex")] <- 1  
 testdf.v11$meds\_sti[str\_detect(testdf.v11$meds\_7\_text, "[C|c]oncerta")] <- 1  
 testdf.v11$meds\_sti[str\_detect(testdf.v11$meds\_7\_text, "[A|a]dderall")] <- 1  
 testdf.v11$meds\_sti[str\_detect(testdf.v11$meds\_7\_text, "[R|r]italin")] <- 1  
 testdf.v11$meds\_sti[str\_detect(testdf.v11$meds\_7\_text, "[P|p]rovigil")] <- 1  
 testdf.v11$meds\_sti[str\_detect(testdf.v11$meds\_7\_text, "[A|a]rmodafinil")] <- 1  
 testdf.v11$meds\_sti[str\_detect(testdf.v11$meds\_7\_text, "[F|f]ocalin")] <- 1  
 testdf.v11$meds\_sti[str\_detect(testdf.v11$meds\_7\_text, "[M|m]etadat")] <- 1  
 testdf.v11$meds\_sti[str\_detect(testdf.v11$meds\_7\_text, "[F|f]ocalin")] <- 1  
 testdf.v11$meds\_sti[str\_detect(testdf.v11$meds\_7\_text, "[N|n]uvigil")] <- 1  
 testdf.v11$meds\_sti[str\_detect(testdf.v11$meds\_7\_text, "ADHD")] <- 1  
 testdf.v11$meds\_sti[str\_detect(testdf.v11$meds\_7\_text, "[M|m]odafinil")] <- 1  
 testdf.v11$meds\_sti[testdf.v11$meds\_7\_text %in% c("Vyance", "Vivance adhd", "Stratera", "Aderol", "Aderal", "Adderal",  
 "Adderrall (for ADHD)", "Adderal, vyvance", "Celltech")] <- 1  
   
# Create antidepressant  
testdf.v11 <- testdf.v11 %>%  
 mutate(meds\_dep = case\_when(meds\_2 == 1 ~ 1, # combine depression var types  
 meds\_8 == 1 ~ 0,  
 meds\_2 == 0 ~ 0))  
 # reconcile write-ins for depression from med\_7\_text  
 testdf.v11$meds\_dep[str\_detect(testdf.v11$meds\_7\_text, "[C|c]elexa")] <- 1  
 testdf.v11$meds\_dep[str\_detect(testdf.v11$meds\_7\_text, "[C|c]italopram")] <- 1  
 testdf.v11$meds\_dep[str\_detect(testdf.v11$meds\_7\_text, "[C|c]ymbalta")] <- 1  
 testdf.v11$meds\_dep[str\_detect(testdf.v11$meds\_7\_text, "[E|e]scitalopram")] <- 1  
 testdf.v11$meds\_dep[str\_detect(testdf.v11$meds\_7\_text, "[L|l]exapro")] <- 1  
 testdf.v11$meds\_dep[str\_detect(testdf.v11$meds\_7\_text, "[S|s]ertraline")] <- 1  
 testdf.v11$meds\_dep[str\_detect(testdf.v11$meds\_7\_text, "[Z|z]oloft")] <- 1  
 testdf.v11$meds\_dep[str\_detect(testdf.v11$meds\_7\_text, "SSRI")] <- 1  
 testdf.v11$meds\_dep[str\_detect(testdf.v11$meds\_7\_text, "[E|e]lavil")] <- 1  
 testdf.v11$meds\_dep[str\_detect(testdf.v11$meds\_7\_text, "[P|p]aroxetine")] <- 1  
 testdf.v11$meds\_dep[str\_detect(testdf.v11$meds\_7\_text, "[P|p]rozac")] <- 1  
 testdf.v11$meds\_dep[str\_detect(testdf.v11$meds\_7\_text, "[A|a]ntidepressant")] <- 1  
 testdf.v11$meds\_dep[str\_detect(testdf.v11$meds\_7\_text, "[A|a]mitriptyline")] <- 1  
 testdf.v11$meds\_dep[str\_detect(testdf.v11$meds\_7\_text, "[D|d]uloxetine")] <- 1  
 testdf.v11$meds\_dep[str\_detect(testdf.v11$meds\_7\_text, "[C|c][e|i]pralex")] <- 1  
 testdf.v11$meds\_dep[str\_detect(testdf.v11$meds\_7\_text, "[W|w]ellbutrin")] <- 1  
 testdf.v11$meds\_dep[str\_detect(testdf.v11$meds\_7\_text, "[S|s]trattera")] <- 1   
 testdf.v11$meds\_dep[str\_detect(testdf.v11$meds\_7\_text, "[A|a]tomoxetine")] <- 1  
 testdf.v11$meds\_dep[str\_detect(testdf.v11$meds\_7\_text, "[F|f]luvoxamine")] <- 1  
 testdf.v11$meds\_dep[str\_detect(testdf.v11$meds\_7\_text, "[V|v]enlafaxine")] <- 1  
 testdf.v11$meds\_dep[str\_detect(testdf.v11$meds\_7\_text, "[P|p]ristiq")] <- 1  
 testdf.v11$meds\_dep[str\_detect(testdf.v11$meds\_7\_text, "[D|d]esipramine")] <- 1  
 testdf.v11$meds\_dep[str\_detect(testdf.v11$meds\_7\_text, "[V|v]iibryd")] <- 1  
 testdf.v11$meds\_dep[str\_detect(testdf.v11$meds\_7\_text, "[V|v]enlafaxin")] <- 1  
 testdf.v11$meds\_dep[str\_detect(testdf.v11$meds\_7\_text, "[C|c]lomipramine")] <- 1  
 testdf.v11$meds\_dep[str\_detect(testdf.v11$meds\_7\_text, "[D|d]esvenflaxine")] <- 1  
 testdf.v11$meds\_dep[str\_detect(testdf.v11$meds\_7\_text, "[M|m]irtazapine")] <- 1  
 testdf.v11$meds\_dep[str\_detect(testdf.v11$meds\_7\_text, "[R|r][e|i]meron")] <- 1  
 testdf.v11$meds\_dep[str\_detect(testdf.v11$meds\_7\_text, "[T|t]raz[a|o|i]done")] <- 1  
 testdf.v11$meds\_dep[str\_detect(testdf.v11$meds\_7\_text, "[N|n]ortriptyline")] <- 1  
 testdf.v11$meds\_dep[testdf.v11$meds\_7\_text %in% c("CYMBALTA", "Cymbolta", "Lexipro", "Lexxapro", "Sertaline",  
 "remiron", "Mirtazepene", "bupropin", "Lexapro, Wellbutrin",  
 "Lexapro and Paxil", "Fuoxetine", "Duluxotine",   
 "Forgot the name, anti depressant for anxiety, daily med",   
 "Atomoxitine (nonstimulant ADHD)", "Fluvoximine (luvox)",  
 "Anafranil", "amatryptaline for migraines", "amnitryptaline",   
 "Buproprion (may or may not count as a psychostim?)",   
 "amatryptaline for migraines", "Minipress")] <- 1  
   
# Create antipsychotic  
testdf.v11 <- testdf.v11 %>%  
 mutate(meds\_psy = case\_when(meds\_3 == 1 ~ 1, # combine anti-psychotic med vars  
 meds\_8 == 1 ~ 0,  
 meds\_3 == 0 ~ 0))  
 # Reconcile anti-psychotic entries from med\_7\_text  
 testdf.v11$meds\_psy[str\_detect(testdf.v11$meds\_7\_text, "[A|a]bilify")] <- 1  
 testdf.v11$meds\_psy[str\_detect(testdf.v11$meds\_7\_text, "[A|a]bility")] <- 1  
 testdf.v11$meds\_psy[str\_detect(testdf.v11$meds\_7\_text, "[V|v]raylar")] <- 1  
 testdf.v11$meds\_psy[str\_detect(testdf.v11$meds\_7\_text, "[O|o]lanzapine")] <- 1  
 testdf.v11$meds\_psy[str\_detect(testdf.v11$meds\_7\_text, "[A|a]ripiprazole")] <- 1  
 testdf.v11$meds\_psy[str\_detect(testdf.v11$meds\_7\_text, "[L|l]atuda")] <- 1  
 testdf.v11$meds\_psy[str\_detect(testdf.v11$meds\_7\_text, "[S|s]eroqu[e|o]l")] <- 1  
 testdf.v11$meds\_psy[str\_detect(testdf.v11$meds\_7\_text, "[G|g]eodon")] <- 1  
 testdf.v11$meds\_psy[str\_detect(testdf.v11$meds\_7\_text, "[Q|q]uetiapine")] <- 1  
 testdf.v11$meds\_psy[testdf.v11$meds\_7\_text %in% c("Ziprazidone", "Quitiapin", "Latuda, Giadon")] <- 1   
   
# Create anti-anxiety   
testdf.v11 <- testdf.v11 %>%   
 mutate(meds\_anx = case\_when(meds\_4 == 1 ~ 1, # combine anti-anxiety var types  
 meds\_8 == 1 ~ 0,  
 meds\_4 == 0 ~ 0))  
 # Reconcile all anti-anxiety drugs from the write-ins from meds\_7\_text  
 testdf.v11$meds\_anx[str\_detect(testdf.v11$meds\_7\_text, "[X|x]anax")] <- 1  
 testdf.v11$meds\_anx[str\_detect(testdf.v11$meds\_7\_text, "[V|v]alium")] <- 1  
 testdf.v11$meds\_anx[str\_detect(testdf.v11$meds\_7\_text, "[P|p]ropranolol")] <- 1  
 testdf.v11$meds\_anx[str\_detect(testdf.v11$meds\_7\_text, "[P|p]ropanol")] <- 1  
 testdf.v11$meds\_anx[str\_detect(testdf.v11$meds\_7\_text, "[L|l]orazepam")] <- 1  
 testdf.v11$meds\_anx[str\_detect(testdf.v11$meds\_7\_text, "[N|n]eurontin")] <- 1  
 testdf.v11$meds\_anx[str\_detect(testdf.v11$meds\_7\_text, "[A|a]tarax")] <- 1  
 testdf.v11$meds\_anx[str\_detect(testdf.v11$meds\_7\_text, "[L|l]yrica")] <- 1  
 testdf.v11$meds\_anx[str\_detect(testdf.v11$meds\_7\_text, "[A|a]tivan")] <- 1  
 testdf.v11$meds\_anx[str\_detect(testdf.v11$meds\_7\_text, "[C|c]lonidine")] <- 1  
 testdf.v11$meds\_anx[str\_detect(testdf.v11$meds\_7\_text, "[P|p]razosin")] <- 1  
 testdf.v11$meds\_anx[str\_detect(testdf.v11$meds\_7\_text, "[B|b]uspar")] <- 1  
 testdf.v11$meds\_anx[str\_detect(testdf.v11$meds\_7\_text, "[V|v]istaril")] <- 1 # antihistamine  
 testdf.v11$meds\_anx[str\_detect(testdf.v11$meds\_7\_text, "[I|i]nderal")] <- 1  
 testdf.v11$meds\_anx[str\_detect(testdf.v11$meds\_7\_text, "[H|h]ydroxyzine")] <- 1 # antihistamine  
 testdf.v11$meds\_anx[str\_detect(testdf.v11$meds\_7\_text, "[K|k]lonopin")] <- 1  
 testdf.v11$meds\_anx[str\_detect(testdf.v11$meds\_7\_text, "[D|d]iazepam")] <- 1  
 testdf.v11$meds\_anx[str\_detect(testdf.v11$meds\_7\_text, "[B|b]uspirone")] <- 1  
 testdf.v11$meds\_anx[str\_detect(testdf.v11$meds\_7\_text, "[G|g]abapentin")] <- 1  
 testdf.v11$meds\_anx[testdf.v11$meds\_7\_text %in% c("Valuim", "you didnt list vallium under anti-anxiety", "Alphrazolam",   
 "Hydroxizine--prescribed as needed for anxiety, only took one because of adverse reaction",   
 "Hydrodoxine, a blood pressure medicine for anxiety that I don't remember the name of")] <- 1  
  
# Create mood stabilizer  
testdf.v11 <- testdf.v11 %>%  
 mutate(meds\_mood = case\_when(meds\_5 == 1 ~ 1, # combine mood stabilizer vars  
 meds\_8 == 1 ~ 0,  
 meds\_5 == 0 ~ 0))  
 # Reconcile mood stabilizers from write-in in med\_7\_text  
 testdf.v11$meds\_mood[str\_detect(testdf.v11$meds\_7\_text, "[L|l]ithium")] <- 1  
 testdf.v11$meds\_mood[str\_detect(testdf.v11$meds\_7\_text, "[L|l]amictal")] <- 1  
 testdf.v11$meds\_mood[str\_detect(testdf.v11$meds\_7\_text, "[L|l]amotrigine")] <- 1  
 testdf.v11$meds\_mood[str\_detect(testdf.v11$meds\_7\_text, "[D|d]epakote")] <- 1  
 testdf.v11$meds\_mood[str\_detect(testdf.v11$meds\_7\_text, "[T|t]rileptal")] <- 1  
 testdf.v11$meds\_mood[str\_detect(testdf.v11$meds\_7\_text, "[O|o]xcarbarzepine")] <- 1  
 testdf.v11$meds\_mood[testdf.v11$meds\_7\_text %in% c("Mood Stabilizer/Birth Control")] <- 1  
   
# Create sleep med  
testdf.v11 <- testdf.v11 %>%  
 mutate(meds\_sle = case\_when(meds\_6 == 1 ~ 1, # combine sleep meds vars  
 meds\_8 == 1 ~ 0,  
 meds\_6 == 0 ~ 0))  
 # Reconcile sleep medications from write-ins in meds\_7\_text  
 testdf.v11$meds\_sle[str\_detect(testdf.v11$meds\_7\_text, "[X|x]yrem")] <- 1  
 testdf.v11$meds\_sle[str\_detect(testdf.v11$meds\_7\_text, "[S|s]leep")] <- 1  
 testdf.v11$meds\_sle[str\_detect(testdf.v11$meds\_7\_text, "[M|m]elatonin")] <- 1  
 testdf.v11$meds\_sle[str\_detect(testdf.v11$meds\_7\_text, "[Z|z]quil")] <- 1  
 testdf.v11$meds\_sle[str\_detect(testdf.v11$meds\_7\_text, "[N|n]yquil")] <- 1  
 testdf.v11$meds\_sle[str\_detect(testdf.v11$meds\_7\_text, "[D|d]iphenhydramine")] <- 1  
 testdf.v11$meds\_sle[testdf.v11$meds\_7\_text %in% c("gabapentin, trazodone (both for sleep on separate occasions)",   
 "generic sleep medication: melatonin supplements",   
 "cold medicine to induce sleep")] <- 1  
   
# Create "Othermeds  
testdf.v11 <- testdf.v11 %>%  
 mutate(meds\_oth = case\_when(meds\_7 == 1 ~ 1, # combine other meds vars  
 meds\_8 == 1 ~ 0,  
 meds\_7 == 0 ~ 0))   
 # Reconcile "other" entries from med\_7\_text  
 testdf.v11$meds\_oth[str\_detect(testdf.v11$meds\_7\_text, "[C|c]annabis")] <- 1  
 testdf.v11$meds\_oth[str\_detect(testdf.v11$meds\_7\_text, "[M|m]arijuana")] <- 1  
 testdf.v11$meds\_oth[str\_detect(testdf.v11$meds\_7\_text, "[W|w]eed")] <- 1  
 testdf.v11$meds\_oth[str\_detect(testdf.v11$meds\_7\_text, "[N|n]altrexone")] <- 1  
 testdf.v11$meds\_oth[str\_detect(testdf.v11$meds\_7\_text, "[S|s]ynthroid")] <- 1  
 testdf.v11$meds\_oth[str\_detect(testdf.v11$meds\_7\_text, "[T|t]opomax")] <- 1  
 testdf.v11$meds\_oth[str\_detect(testdf.v11$meds\_7\_text, "[T|t]estosterone")] <- 1  
 testdf.v11$meds\_oth[str\_detect(testdf.v11$meds\_7\_text, "[T|t]hyroid")] <- 1  
 testdf.v11$meds\_oth[str\_detect(testdf.v11$meds\_7\_text, "[H|h]ydrodocone")] <- 1   
 testdf.v11$meds\_oth[str\_detect(testdf.v11$meds\_7\_text, "[C|c]lonidine")] <- 1  
 testdf.v11$meds\_oth[str\_detect(testdf.v11$meds\_7\_text, "[G|g]uanfacine")] <- 1  
 testdf.v11$meds\_oth[str\_detect(testdf.v11$meds\_7\_text, "[K|k]etamine")] <- 1  
 testdf.v11$meds\_oth[str\_detect(testdf.v11$meds\_7\_text, "[F|f]lexeril")] <- 1  
 testdf.v11$meds\_oth[str\_detect(testdf.v11$meds\_7\_text, "[P|p]ant[o|a]loc")] <- 1  
 testdf.v11$meds\_oth[str\_detect(testdf.v11$meds\_7\_text, "[S|s]pironolactone")] <- 1  
 testdf.v11$meds\_oth[str\_detect(testdf.v11$meds\_7\_text, "[H|h]ormone")] <- 1  
 testdf.v11$meds\_oth[str\_detect(testdf.v11$meds\_7\_text, "[O|o]xycodone")] <- 1  
 testdf.v11$meds\_oth[str\_detect(testdf.v11$meds\_7\_text, "[C|c]annabic")] <- 1  
 testdf.v11$meds\_oth[str\_detect(testdf.v11$meds\_7\_text, "[B|b]eta")] <- 1  
 testdf.v11$meds\_oth[testdf.v11$meds\_7\_text %in% c("Canabis", "Vistiril", "Vistral", "Thyroxine", "The Reefer",  
 "Topimitrate", "topiramate", "Estradiol, Spironactone",   
 "muscle relaxants to help me sleep", "Ondansetron" ,  
 "Sumatriptan for migraines", "Psilocybin Mushrooms", "clonodine",  
 "Muscle Relaxers", "anti-siezure medication", "Codeine",   
 "Androgen Blockers", "Anti-seizure to control racing thoughts",  
 "Anti seizure", "Buprenorphine (Suboxone) for opiate dependence",  
 "Hydroxine", "Hydroxcyzine", "Alyacen", "Mirapex",   
 "hypothyroid medication", "Epilepsy drugs", "Birth control",  
 "Methadone", "Vimpat", "Contrave", "Benztropine", "Intuniv",  
 "Medicinal marijuana", "Pantaloc, for psychogenic vomiting",  
 "medicinal Cannabis", "Testosterone Enanthate", "testosterone",  
 "Cannabis", "Medical marijuana")] <- 1  
   
# Create a group for students who do not take any psychotropic meds  
testdf.v11 <- testdf.v11 %>%  
 mutate(meds\_none = case\_when(meds\_none == 1 ~ 1, # filter for all meds w/ none  
 meds\_none == 0 ~ 0,  
 meds\_sti == 1 ~ 0,  
 meds\_dep == 1 ~ 0,  
 meds\_psy == 1 ~ 0,  
 meds\_anx == 1 ~ 0,  
 meds\_mood == 1 ~ 0,  
 meds\_sle == 1 ~ 0,  
 meds\_oth == 1 ~ 0)) %>%  
 filter(!is.na(meds\_any))  
   
# tabulate and check outputs  
table(testdf.v11$meds\_sti)

##   
## 0 1   
## 14254 1189

table(testdf.v11$meds\_dep)

##   
## 0 1   
## 14254 2716

table(testdf.v11$meds\_psy)

##   
## 0 1   
## 14254 190

table(testdf.v11$meds\_anx)

##   
## 0 1   
## 14254 1388

table(testdf.v11$meds\_mood)

##   
## 0 1   
## 14254 339

table(testdf.v11$meds\_sle)

##   
## 0 1   
## 14254 642

table(testdf.v11$meds\_oth)

##   
## 0 1   
## 14254 316

table(testdf.v11$meds\_none)

##   
## 0 1   
## 4107 14254

table(testdf.v11$meds\_any, useNA = "always")

##   
## 0 1 <NA>   
## 14254 4107 0

## Psychopharm medication class count

# convert all med vars to numeric for future medcount var totaling  
testdf.v11$meds\_1 <- as.numeric(testdf.v11$meds\_1)  
testdf.v11$meds\_2 <- as.numeric(testdf.v11$meds\_2)  
testdf.v11$meds\_3 <- as.numeric(testdf.v11$meds\_3)  
testdf.v11$meds\_4 <- as.numeric(testdf.v11$meds\_4)  
testdf.v11$meds\_5 <- as.numeric(testdf.v11$meds\_5)  
testdf.v11$meds\_6 <- as.numeric(testdf.v11$meds\_6)  
testdf.v11$meds\_7 <- as.numeric(testdf.v11$meds\_7)  
testdf.v11$meds\_8 <- as.numeric(testdf.v11$meds\_8)  
  
# set any NAs equal to zero  
testdf.v11$meds\_1[is.na(testdf.v11$meds\_1)] <- 0  
testdf.v11$meds\_2[is.na(testdf.v11$meds\_2)] <- 0  
testdf.v11$meds\_3[is.na(testdf.v11$meds\_3)] <- 0  
testdf.v11$meds\_4[is.na(testdf.v11$meds\_4)] <- 0  
testdf.v11$meds\_5[is.na(testdf.v11$meds\_5)] <- 0  
testdf.v11$meds\_6[is.na(testdf.v11$meds\_6)] <- 0  
testdf.v11$meds\_7[is.na(testdf.v11$meds\_7)] <- 0  
  
# create meds\_count var  
testdf.v11 <- testdf.v11 %>%  
 mutate(meds\_count = meds\_1 + meds\_2 + meds\_3 + meds\_4 + meds\_5 + meds\_6 + meds\_7) %>%  
 mutate(meds\_count = case\_when(meds\_count == 0 ~ 0,  
 meds\_count == 1 ~ 1,  
 meds\_count == 2 ~ 2,   
 meds\_count == 3 ~ 3,  
 meds\_count == 4 ~ 4,  
 meds\_count == 5 ~ 5,  
 meds\_count == 6 ~ 6))  
testdf.v11$meds\_count <- as.numeric(testdf.v11$meds\_count) # return var to numeric cass  
  
table(testdf.v11$meds\_count, useNA = "always") # check output

##   
## 0 1 2 3 4 5 6 <NA>   
## 14254 2422 1090 415 129 45 6 0

testdf.v11 = subset(testdf.v11, select= -c(meds\_1:meds\_7)) # clean up vars/columns

testdf.v13 <- testdf.v11

## Discrimination

testdf.v13$discrim<- as.numeric(testdf.v13$discrim)  
table(testdf.v13$discrim, useNA = "always") # check output (lots of NAs)

##   
## 1 2 3 4 5 6 <NA>   
## 6157 3149 1509 347 87 47 7065

## Religion

# Label levels of religious importance  
testdf.v13 <- testdf.v13 %>%  
 # flip coding  
 mutate(religios = case\_when(religios == 1 ~ 5, # fhighly religious  
 religios == 2 ~ 4,  
 religios == 3 ~ 3,  
 religios == 4 ~ 2,  
 religios == 5 ~ 1)) %>% # not at all religious  
 filter(!is.na(religios))  
testdf.v13$religios <- as.numeric(testdf.v13$religios)  
  
table(testdf.v13$religios, useNA = "always") # Check output

##   
## 1 2 3 4 5 <NA>   
## 3815 3452 4624 3721 2739 0

## Residence type

# re-organize living situation into on-campus, off-campus, and at home  
testdf.v13 <- testdf.v13 %>%  
 mutate(residenc = case\_when(residenc == 1 ~ 1, # on-campus  
 residenc == 2 ~ 1,  
 residenc == 3 ~ 2, # coop  
 residenc == 4 ~ 2,   
 residenc == 5 ~ 3, # off-campus  
 residenc == 6 ~ 4, # with family  
 residenc == 7 ~ 3)) %>%  
 filter(!is.na(residenc))  
  
table(testdf.v13$residenc, useNA = "always") # check output

##   
## 1 2 3 4 <NA>   
## 7849 444 7692 2362 0

## MH campus environment

testdf.v13$env\_mh <- as.factor(testdf.v13$env\_mh)  
table(testdf.v13$env\_mh, useNA = "always") # check output (excessive NAs)

##   
## 1 2 3 4 5 6 <NA>   
## 1179 1627 2563 2543 2709 664 7062

## Military status

testdf.v14 <- testdf.v13  
table(testdf.v14$military, useNA = "always")

##   
## 0 1 <NA>   
## 18180 154 13

## International student

testdf.v14 <- testdf.v14 %>%  
 mutate(international = case\_when(international == 1 ~ 1,  
 international == 0 ~ 0)) %>%  
 filter(!is.na(international))  
  
testdf.v14$international <- as.logical(testdf.v14$international)  
  
table(testdf.v14$international, useNA = "always")

##   
## FALSE TRUE <NA>   
## 16731 1604 0

## Insurance coverage

# create one insurance var from related questions in 'ins\_' module  
testdf.v14 <- testdf.v14 %>%   
 mutate(ins\_no = case\_when((ins\_1 == 1 | ins\_mh == 4 |   
 ins\_mh == 5 | ins\_ade == 3) ~ 1, # inadequate coverage  
 (ins\_1 == 0 | ins\_mh == 1 |   
 ins\_mh == 2 | ins\_ade == 2) ~ 0)) %>% # adequate MH coverage  
 filter(!is.na(ins\_no))   
testdf.v14$ins\_no <- as.logical(testdf.v14$ins\_no) # switch class to boolean  
table(testdf.v14$ins\_no, useNA = "always") # check output, skew

##   
## FALSE TRUE <NA>   
## 10078 3084 0

jarque.bera.test(testdf.v14$ins\_no)

##   
## Jarque Bera Test  
##   
## data: testdf.v14$ins\_no  
## X-squared = 3552.1, df = 2, p-value < 2.2e-16

testdf.v14 = subset(testdf.v14, select= -c(ins\_1, ins\_mh, ins\_ade, ins\_inade)) # clean up var

## Family support

testdf.v14$fam\_support\_aca <- as.numeric(testdf.v14$fam\_support\_aca)  
table(testdf.v14$fam\_support\_aca, useNA = "always") # check output (excessive NAs)

##   
## 1 2 3 4 5 6 <NA>   
## 940 596 224 50 23 19 11310

## Activity Involvement

# set all NAs in activities equal to 0  
testdf.v14$activ\_ac[is.na(testdf.v14$activ\_ac)] <- 1  
testdf.v14$activ\_athc[is.na(testdf.v14$activ\_athc)] <- 1  
testdf.v14$activ\_athv[is.na(testdf.v14$activ\_athv)] <- 1  
testdf.v14$activ\_athi[is.na(testdf.v14$activ\_athi)] <- 1  
testdf.v14$activ\_cs[is.na(testdf.v14$activ\_cs)] <- 1  
testdf.v14$activ\_cu[is.na(testdf.v14$activ\_cu)] <- 1  
testdf.v14$activ\_da[is.na(testdf.v14$activ\_da)] <- 1  
testdf.v14$activ\_fs[is.na(testdf.v14$activ\_fs)] <- 1  
testdf.v14$activ\_gs[is.na(testdf.v14$activ\_gs)] <- 1  
testdf.v14$activ\_gov[is.na(testdf.v14$activ\_gov)] <- 1  
testdf.v14$activ\_mp[is.na(testdf.v14$activ\_md)] <- 1  
testdf.v14$activ\_md[is.na(testdf.v14$activ\_none)] <- 1  
testdf.v14$activ\_soc[is.na(testdf.v14$activ\_soc)] <- 1  
testdf.v14$activ\_rel[is.na(testdf.v14$activ\_rel)] <- 1  
testdf.v14$activ\_art[is.na(testdf.v14$activ\_art)] <- 1  
testdf.v14$activ\_other[is.na(testdf.v14$activ\_other)] <- 1  
  
# Create one activity variable (dichotomous) based on involvement in campus activities  
testdf.v14 <- testdf.v14 %>%   
 mutate(activ\_no = case\_when(activ\_none == 1 ~ 1,  
 activ\_ac == 1 ~ 0,  
 activ\_athc == 1 ~ 0,  
 activ\_athv == 1 ~ 0,  
 activ\_athi == 1 ~ 0,  
 activ\_cs == 1 ~ 0,  
 activ\_cu == 1 ~ 0,  
 activ\_da == 1 ~ 0,  
 activ\_fs == 1 ~ 0,  
 activ\_gs == 1 ~ 0,  
 activ\_gov == 1 ~ 0,  
 activ\_mp == 1 ~ 0,  
 activ\_md == 1 ~ 0,  
 activ\_rel == 1 ~ 0,  
 activ\_soc == 1 ~ 0,  
 activ\_art == 1 ~ 0,  
 activ\_other == 1 ~ 0))  
testdf.v14$activ\_no <- as.logical(testdf.v14$activ\_no)  
  
table(testdf.v14$activ\_no, useNA = "always") # tabulate and check output

##   
## FALSE TRUE <NA>   
## 9259 3903 0

jarque.bera.test(testdf.v14$activ\_no)

##   
## Jarque Bera Test  
##   
## data: testdf.v14$activ\_no  
## X-squared = 2539.2, df = 2, p-value < 2.2e-16

testdf.v14 = subset(testdf.v14, select= -c(activ\_none, activ\_ac:activ\_other)) # clean up vars

## School variable

# insert each school as a dichotomous var  
testdf.v14 <- testdf.v14 %>%  
 mutate(var\_AlbertaCAD = ifelse((school2 == "AlbertaCAD"), 1, 0)) %>%  
 mutate(var\_Boston = ifelse((school2 == "Boston"), 1, 0)) %>%  
 mutate(var\_CIA = ifelse((school2 == "CIA"), 1, 0)) %>%  
 mutate(var\_CalPoly = ifelse((school2 == "CalPoly"), 1, 0)) %>%  
 mutate(var\_ColoradoCollege = ifelse((school2 == "ColoradoCollege"), 1, 0)) %>%  
 mutate(var\_ColoradoMountain = ifelse((school2 == "ColoradoMountain"), 1, 0)) %>%  
 mutate(var\_ColumbusCAD = ifelse((school2 == "ColumbusCAD"), 1, 0)) %>%  
 mutate(var\_Delaware = ifelse((school2 == "Delaware"), 1, 0)) %>%  
 mutate(var\_DetroitMercy = ifelse((school2 == "DetroitMercy"), 1, 0)) %>%  
 mutate(var\_Emerson = ifelse((school2 == "Emerson"), 1, 0)) %>%  
 mutate(var\_GeorgeMason = ifelse((school2 == "GeorgeMason"), 1, 0)) %>%  
 mutate(var\_GeorgiaTech = ifelse((school2 == "GeorgiaTech"), 1, 0)) %>%  
 mutate(var\_Kalamazoo = ifelse((school2 == "Kalamazoo"), 1, 0)) %>%  
 mutate(var\_Kansas = ifelse((school2 == "Kansas"), 1, 0)) %>%  
 mutate(var\_MDInstituteCollegeArt = ifelse((school2 == "MDInstituteCollegeArt"), 1, 0)) %>%  
 mutate(var\_MassArt = ifelse((school2 == "MassArt"), 1, 0)) %>%  
 mutate(var\_Merrimack = ifelse((school2 == "Merrimack"), 1, 0)) %>%  
 mutate(var\_MichiganState = ifelse((school2 == "MichiganState"), 1, 0)) %>%  
 mutate(var\_MinnesotaCAD = ifelse((school2 == "MinnesotaCAD"), 1, 0)) %>%  
 mutate(var\_MontclairState = ifelse((school2 == "MontclairState"), 1, 0)) %>%  
 mutate(var\_NHInstituteArt = ifelse((school2 == "NHInstituteArt"), 1, 0)) %>%  
 mutate(var\_NevadaReno = ifelse((school2 == "NevadaReno"), 1, 0)) %>%  
 mutate(var\_OaklandCC = ifelse((school2 == "OaklandCC"), 1, 0)) %>%  
 mutate(var\_OklahomaCityCC = ifelse((school2 == "OklahomaCityCC"), 1, 0)) %>%  
 mutate(var\_PaloAltoCollege = ifelse((school2 == "PaloAltoCollege"), 1, 0)) %>%  
 mutate(var\_PennCollegeTech = ifelse((school2 == "PennCollegeTech"), 1, 0)) %>%  
 mutate(var\_Pratt = ifelse((school2 == "RedlandsCC"), 1, 0)) %>%  
 mutate(var\_RhodeIsland = ifelse((school2 == "RhodeIsland"), 1, 0)) %>%  
 mutate(var\_RedlandsCC = ifelse((school2 == "RedlandsCC"), 1, 0)) %>%  
 mutate(var\_Ringling = ifelse((school2 == "Ringling"), 1, 0)) %>%  
 mutate(var\_Rollins = ifelse((school2 == "Rollins"), 1, 0)) %>%  
 mutate(var\_SAIC = ifelse((school2 == "SAIC"), 1, 0)) %>%  
 mutate(var\_Sewanee = ifelse((school2 == "Sewanee"), 1, 0)) %>%  
 mutate(var\_Smith = ifelse((school2 == "Smith"), 1, 0)) %>%  
 mutate(var\_SouthernNazarene = ifelse((school2 == "SouthernNazarene"), 1, 0)) %>%  
 mutate(var\_SouthwesternOklahomaState = ifelse((school2 == "SouthwesternOklahomaState"), 1, 0)) %>%  
 mutate(var\_StJohns = ifelse((school2 == "StJohns"), 1, 0)) %>%  
 mutate(var\_StMarys = ifelse((school2 == "StMarys"), 1, 0)) %>%  
 mutate(var\_StRose = ifelse((school2 == "StRose"), 1, 0)) %>%  
 mutate(var\_TrumanState = ifelse((school2 == "TrumanState"), 1, 0)) %>%  
 mutate(var\_TuftsHealthSciences = ifelse((school2 == "TuftsHealthSciences"), 1, 0)) %>%  
 mutate(var\_TuftsMainCampus = ifelse((school2 == "TuftsMainCampus"), 1, 0)) %>%  
 mutate(var\_UChicago = ifelse((school2 == "UChicago"), 1, 0)) %>%  
 mutate(var\_UFlorida = ifelse((school2 == "UFlorida"), 1, 0)) %>%  
 mutate(var\_UMDearborn = ifelse((school2 == "UMDearborn"), 1, 0)) %>%  
 mutate(var\_UMich = ifelse((school2 == "UMich"), 1, 0)) %>%  
 mutate(var\_UNCSchoolArt = ifelse((school2 == "UNCSchoolArt"), 1, 0)) %>%  
 mutate(var\_UTKnoxville = ifelse((school2 == "UTKnoxville"), 1, 0)) %>%  
 mutate(var\_UniversityofSouthernCalifornia = ifelse((school2 == "UniversityofSouthernCalifornia"), 1, 0)) %>%   
 mutate(var\_Utah = ifelse((school2 == "Utah"), 1, 0)) %>%  
 mutate(var\_VATech = ifelse((school2 == "VATech"), 1, 0)) %>%  
 mutate(var\_VirginiaCommonwealth = ifelse((school2 == "VirginiaCommonwealth"), 1, 0)) %>%  
 mutate(var\_WakeForest = ifelse((school2 == "WakeForest"), 1, 0)) %>%  
 mutate(var\_Watkins = ifelse((school2 == "Watkins"), 1, 0)) %>%  
 mutate(var\_WestVirginia = ifelse((school2 == "WestVirginia"), 1, 0)) %>%  
 mutate(var\_WesternCarolina = ifelse((school2 == "WesternCarolina"), 1, 0)) %>%  
 mutate(var\_WesternMichigan = ifelse((school2 == "WesternMichigan"), 1, 0)) %>%  
 mutate(var\_WesternWA = ifelse((school2 == "WesternWA"), 1, 0)) %>%  
 mutate(var\_Xavier = ifelse((school2 == "Xavier"), 1, 0))

### School type levels

1. 4-year: var\_Boston, var\_CalPoly, var\_ColoradoCollege, var\_ColoradoMountain, var\_Delaware, var\_DetroitMercy, var\_Emerson, var\_GeorgeMason, var\_GeorgiaTech, var\_Kalamazoo, var\_Kansas, var\_Merrimack, var\_MichiganState, var\_MontclairState, var\_NevadaReno, var\_PennCollegeTech, var\_RhodeIsland, var\_Rollins, var\_Sewanee, var\_Smith, var\_SouthernNazarene, var\_SouthwesternOklahomaState, var\_StJohns, var\_StMarys, var\_StRose, var\_TrumanState, var\_TuftsHealthSciences, var\_TuftsMainCampus, var\_UChicago, var\_UFlorida, var\_UMDearborn, var\_UMich, var\_UTKnoxville, var\_UniversityofSouthernCalifornia, var\_Utah, var\_VATech, var\_VirginiaCommonwealth, var\_WakeForest,var\_WestVirginia, var\_WesternCarolina, var\_WesternMichigan, var\_WesternWA, var\_Xavier
2. Community College: var\_OaklandCC, var\_OklahomaCityCC, var\_PaloAltoCollege, var\_RedlandsCC
3. Art & Design, Culinary: var\_AlbertaCAD, var\_CIA, var\_ColumbusCAD, var\_MDInstituteCollegeArt, var\_MassArt, var\_MinnesotaCAD, var\_NHInstituteArt, var\_Pratt, , var\_Ringling, var\_SAIC, var\_UNCSchoolArt, var\_Watkins,

testdf.v14 <- testdf.v14 %>%  
 # 4-year college  
 mutate(school2\_4yr = ifelse(var\_Boston + var\_CalPoly + var\_ColoradoCollege + var\_ColoradoMountain + var\_Delaware +  
 var\_DetroitMercy + var\_Emerson + var\_GeorgeMason + var\_GeorgiaTech + var\_Kalamazoo +   
 var\_Kansas + var\_Merrimack + var\_MichiganState + var\_MontclairState + var\_NevadaReno +  
 var\_PennCollegeTech + var\_RhodeIsland + var\_Rollins + var\_Sewanee + var\_Smith +  
 var\_SouthernNazarene + var\_SouthwesternOklahomaState + var\_StJohns + var\_StMarys + var\_StRose +  
 var\_TrumanState + var\_TuftsHealthSciences + var\_TuftsMainCampus + var\_UChicago + var\_UFlorida +  
 var\_UMDearborn + var\_UMich + var\_UTKnoxville + var\_UniversityofSouthernCalifornia + var\_Utah +  
 var\_VATech + var\_VirginiaCommonwealth + var\_WakeForest + var\_WestVirginia +   
 var\_WesternCarolina + var\_WesternMichigan + var\_WesternWA + var\_Xavier == 1, TRUE,  
 ifelse(  
 var\_Boston + var\_CalPoly + var\_ColoradoCollege + var\_ColoradoMountain + var\_Delaware +  
 var\_DetroitMercy + var\_Emerson + var\_GeorgeMason + var\_GeorgiaTech + var\_Kalamazoo +  
 var\_Kansas + var\_Merrimack + var\_MichiganState + var\_MontclairState + var\_NevadaReno +  
 var\_PennCollegeTech + var\_RhodeIsland + var\_Rollins + var\_Sewanee + var\_Smith +  
 var\_SouthernNazarene + var\_SouthwesternOklahomaState + var\_StJohns + var\_StMarys +   
 var\_StRose + var\_TrumanState + var\_TuftsHealthSciences + var\_TuftsMainCampus + var\_UChicago +  
 var\_UFlorida + var\_UMDearborn + var\_UMich + var\_UTKnoxville +  
 var\_UniversityofSouthernCalifornia + var\_Utah + var\_VATech + var\_VirginiaCommonwealth +  
 var\_WakeForest + var\_WestVirginia + var\_WesternCarolina + var\_WesternMichigan +  
 var\_WesternWA + var\_Xavier == 0, FALSE, NA))) %>%  
   
 # Community colleges  
 mutate(school2\_cc = ifelse(var\_OaklandCC + var\_OklahomaCityCC + var\_PaloAltoCollege + var\_RedlandsCC == 1, TRUE,   
 ifelse(var\_OaklandCC + var\_OklahomaCityCC + var\_PaloAltoCollege + var\_RedlandsCC == 0, FALSE, NA))) %>%  
   
 # Art, design, culinary  
 mutate(school2\_ad = ifelse(var\_AlbertaCAD + var\_CIA + var\_ColumbusCAD + var\_MDInstituteCollegeArt +   
 var\_MassArt + var\_MinnesotaCAD + var\_NHInstituteArt + var\_Pratt + var\_Ringling +  
 var\_SAIC + var\_UNCSchoolArt + var\_Watkins == 1, TRUE,   
 ifelse(  
 var\_AlbertaCAD + var\_CIA + var\_ColumbusCAD + var\_MDInstituteCollegeArt +  
 var\_MassArt + var\_MinnesotaCAD + var\_NHInstituteArt + var\_Pratt +   
 var\_Ringling + var\_SAIC + var\_UNCSchoolArt + var\_Watkins == 0, FALSE, NA)))  
  
# Binary variable for school type  
table(testdf.v14$school2\_4yr)

##   
## FALSE TRUE   
## 2139 11023

table(testdf.v14$school2\_cc)

##   
## FALSE TRUE   
## 12853 309

table(testdf.v14$school2\_ad)

##   
## FALSE TRUE   
## 11680 1482

# create one school type variables  
testdf.v14 <- testdf.v14 %>%  
 mutate(school2\_type = case\_when(school2\_4yr == TRUE ~ 1,  
 school2\_cc == TRUE ~ 2,  
 school2\_ad == TRUE ~ 3)) %>%  
 filter(!is.na(school2\_type))  
testdf.v14$school2\_type <- as.factor(testdf.v14$school2\_type)  
  
table(testdf.v14$school2\_type, useNA = "always") # check output

##   
## 1 2 3 <NA>   
## 11023 309 1443 0

# LIFE EVENTS

testdf.v16 <- testdf.v14

## Illicit drug use

Creates illicit drug use var (limitation – 30 days reflection)

## adjusted for nested question status  
testdf.v16$drug\_coc[is.na(testdf.v16$drug\_coc)] <- 0  
testdf.v16$drug\_her[is.na(testdf.v16$drug\_her)] <- 0  
testdf.v16$drug\_met[is.na(testdf.v16$drug\_met)] <- 0  
testdf.v16$drug\_stim[is.na(testdf.v16$drug\_stim)] <- 0  
testdf.v16$drug\_ect[is.na(testdf.v16$drug\_ect)] <- 0  
testdf.v16$drug\_opi[is.na(testdf.v16$drug\_opi)] <- 0  
testdf.v16$drug\_other[is.na(testdf.v16$drug\_other)] <- 0  
  
# visualize current output before transformation  
table(testdf.v16$drug\_coc)

##   
## 0 1   
## 12535 240

table(testdf.v16$drug\_her) # too small

##   
## 0 1   
## 12772 3

table(testdf.v16$drug\_met) # too small

##   
## 0 1   
## 12768 7

table(testdf.v16$drug\_stim)

##   
## 0 1   
## 12472 303

table(testdf.v16$drug\_ect) # too small

##   
## 0 1   
## 12710 65

table(testdf.v16$drug\_opi) # too small

##   
## 0 1   
## 12698 77

table(testdf.v16$drug\_other) # too vague, small

##   
## 0 1   
## 12641 134

# Sample is too small to analyze individual drugs, except for marijuana.   
# Transform to illicit drug y/n  
testdf.v16 <- testdf.v16 %>%  
 mutate(drugs\_yn = case\_when(drug\_coc == 1 ~ 1,  
 drug\_her == 1 ~ 1,  
 drug\_met == 1 ~ 1,  
 drug\_stim == 1 ~ 1,  
 drug\_ect == 1 ~ 1,  
 drug\_opi == 1 ~ 1,  
 drug\_other == 1 ~ 1))  
testdf.v16$drugs\_yn[is.na(testdf.v16$drugs\_yn)] <- 0  
testdf.v16$drugs\_yn <- as.logical(testdf.v16$drugs\_yn)  
  
table(testdf.v16$drugs\_yn, useNA = "always")

##   
## FALSE TRUE <NA>   
## 12150 625 0

jarque.bera.test(testdf.v16$drugs\_yn)

##   
## Jarque Bera Test  
##   
## data: testdf.v16$drugs\_yn  
## X-squared = 164984, df = 2, p-value < 2.2e-16

## Marijuana use

testdf.v16$drug\_mar[is.na(testdf.v16$drug\_mar)] <- 0  
table(testdf.v16$drug\_mar)

##   
## 0 1   
## 10248 2527

testdf.v16 <- testdf.v16 %>%  
 mutate(drug\_mar = case\_when(drug\_mar == 1 ~ 1,   
 drug\_mar == 0 ~ 0)) %>%  
 filter(!is.na(drug\_mar)) # filters out NAs  
testdf.v16$drug\_mar <- as.logical(testdf.v16$drug\_mar)  
  
table(testdf.v16$drug\_mar, useNA = "always")

##   
## FALSE TRUE <NA>   
## 10248 2527 0

jarque.bera.test(testdf.v16$drug\_mar)

##   
## Jarque Bera Test  
##   
## data: testdf.v16$drug\_mar  
## X-squared = 4949.9, df = 2, p-value < 2.2e-16

## Current financial stressors

testdf.v16 <- testdf.v16 %>%  
 # make dichotomous  
 mutate(fincur = case\_when(fincur == 3 ~ 0, # sometimes  
 fincur == 4 ~ 0, # rarely or never  
 fincur == 5 ~ 0,  
 fincur == 1 ~ 1, # always or often stressful  
 fincur == 2 ~ 1)) %>%  
 filter(!is.na(fincur)) # filters for NAs  
testdf.v16$fincur <- as.logical(testdf.v16$fincur)  
  
table(testdf.v16$fincur, useNA = "always") # check output, skew

##   
## FALSE TRUE <NA>   
## 8313 4459 0

jarque.bera.test(testdf.v16$fincur)

##   
## Jarque Bera Test  
##   
## data: testdf.v16$fincur  
## X-squared = 2214.1, df = 2, p-value < 2.2e-16

## Abuse or assault, in the last year only

table(testdf.v16$abuse\_life)

##   
## 1 2 3 4 5   
## 7091 1177 2170 685 1631

table(testdf.v16$abuse\_recent)

##   
## 1 2 3 4 5   
## 82 119 685 2035 2733

# create dichotomous variable called 'abuse\_year' to reflect abuse that occurred in last 12 months  
testdf.v17 <- testdf.v16 %>%  
 mutate(abuse\_year = case\_when(abuse\_recent == 1 ~ 1,  
 abuse\_recent == 2 ~ 1,  
 abuse\_recent == 3 ~ 1,  
 abuse\_recent == 4 ~ 0,  
 abuse\_recent == 5 ~ 0,  
 abuse\_life == 0 ~ 0)) %>%  
 filter(!is.na(abuse\_life))  
table(testdf.v17$abuse\_year) # check output

##   
## 0 1   
## 4768 886

# restructure sexual assault var  
testdf.v17 <- testdf.v17 %>%  
 mutate(assault = case\_when(  
 assault\_sex == 1 ~ 1,  
 assault\_emo == 1 ~ 1,  
 assault\_phys == 1 ~ 1,   
 abuse\_year == 1 ~ 1,  
 abuse\_year == 0 ~ 0,  
 assault\_sex == 2 ~ 0,  
 assault\_emo == 0 ~ 0,  
 assault\_phys == 1 ~ 0)) %>%  
 mutate(assault = case\_when(assault == 1 ~ 1,  
 assault == 0 ~ 0)) %>%  
 filter(!is.na(assault))  
testdf.v17$assault <- as.logical(testdf.v17$assault)  
  
# tabulate and check output  
table(testdf.v17$assault, useNA = "always")

##   
## FALSE TRUE <NA>   
## 7817 4937 0

jarque.bera.test(testdf.v17$assault)

##   
## Jarque Bera Test  
##   
## data: testdf.v17$assault  
## X-squared = 2150.2, df = 2, p-value < 2.2e-16

testdf.v17 = subset(testdf.v17, select= -c(abuse\_life, abuse\_recent)) # clean up var

## Binging frequency

# set all binge\_fr values equal to 0  
testdf.v17$binge\_fr\_f[is.na(testdf.v17$binge\_fr\_f)] <- 0  
testdf.v17$binge\_fr\_m[is.na(testdf.v17$binge\_fr\_m)] <- 0  
testdf.v17$binge\_fr\_o[is.na(testdf.v17$binge\_fr\_o)] <- 0  
  
# create one binge var  
testdf.v18 <- testdf.v17 %>%   
 mutate(binge\_fr = as.numeric(binge\_fr\_f) + as.numeric(binge\_fr\_m) + as.numeric(binge\_fr\_o))  
table(testdf.v18$binge\_fr)

##   
## 0 1 2 3 4 5 6 7   
## 4718 2694 2006 1535 1404 269 83 45

testdf.v18 <- testdf.v18 %>%  
 mutate(binge = case\_when(binge\_fr >= 4 ~ 1,  
 binge\_fr < 4 ~ 0)) %>%  
 filter(!is.na(binge))  
testdf.v18$binge <-as.logical(testdf.v18$binge)  
  
table(testdf.v18$binge, useNA = "always") # check output

##   
## FALSE TRUE <NA>   
## 10953 1801 0

jarque.bera.test(testdf.v18$binge)

##   
## Jarque Bera Test  
##   
## data: testdf.v18$binge  
## X-squared = 11707, df = 2, p-value < 2.2e-16

testdf.v18 = subset(testdf.v18, select= -c(binge\_fr\_f, binge\_fr\_m, binge\_fr\_o)) # clean up var

## Recent Psychiatric Hospitalization, last 12 months

# Create variable for any psychiatric hospitalization in the last 12 months  
testdf.v18 <- testdf.v18 %>%   
 mutate(psyhx = case\_when((prov\_pes == 1 | prov\_inp == 1 | prov\_par == 1) ~ 1, # inpatient, IOPs, etc.  
 (prov\_pes == 0 & prov\_inp == 0 & prov\_par == 0) ~ 0, # no crises intervened on  
 ther\_ever == 0 ~ 0)) %>%  
 filter(!is.na(psyhx)) # filter out NAs  
testdf.v18$psyhx <- as.logical(testdf.v18$psyhx) # change class to boolean  
  
table(testdf.v18$psyhx, useNA = "always") # check output, skew

##   
## FALSE TRUE <NA>   
## 12550 183 0

jarque.bera.test(testdf.v18$psyhx)

##   
## Jarque Bera Test  
##   
## data: testdf.v18$psyhx  
## X-squared = 2354935, df = 2, p-value < 2.2e-16

testdf.v18 = subset(testdf.v18, select= -c(prov\_pes, prov\_inp, prov\_par)) # clean up var

## Current GPA

testdf.v18 <- testdf.v18 %>%  
# bin grades to As, Bs, Cs, and D+ and below  
 mutate(gpa\_sr = case\_when(gpa\_sr == 0 ~ 1, # A+  
 gpa\_sr == 1 ~ 1, # A  
 gpa\_sr == 2 ~ 1, # A-  
 gpa\_sr == 3 ~ 2, # B+  
 gpa\_sr == 4 ~ 2, # B  
 gpa\_sr == 5 ~ 2, # B-  
 gpa\_sr == 6 ~ 3, # C+  
 gpa\_sr == 7 ~ 3, # C  
 gpa\_sr == 8 ~ 3, # C-  
 gpa\_sr == 9 ~ 4, # D+ and below  
 gpa\_sr == 10 ~ 5)) %>% # Don't know  
 filter(!is.na(gpa\_sr)) # filter out NAs  
testdf.v18$gpa\_sr <- as.numeric(testdf.v18$gpa\_sr) # change to numeric scale  
  
table(testdf.v18$gpa\_sr, useNA = "always") # check output, skew

##   
## 1 2 3 4 5 <NA>   
## 6751 4050 662 38 1130 0

jarque.bera.test(testdf.v18$gpa\_sr)

##   
## Jarque Bera Test  
##   
## data: testdf.v18$gpa\_sr  
## X-squared = 9462.2, df = 2, p-value < 2.2e-16

# DEFEAT-ENTRAPMENT

testdf.v20 <- testdf.v18

## PHQ risk/severity

testdf.v20$deprawsc <- as.numeric(testdf.v20$deprawsc)  
  
testdf.v20 <- testdf.v20 %>%  
 filter(!is.na(deprawsc)) # filters out NAs  
  
table(testdf.v20$deprawsc, useNA ="always")

##   
## 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15   
## 562 613 842 937 1013 868 842 821 744 744 566 472 431 413 357 349   
## 16 17 18 19 20 21 22 23 24 25 26 27 <NA>   
## 299 266 237 183 155 132 137 100 104 82 46 81 0

# Categorize by clinical translation for later --  
testdf.v20 <- testdf.v20 %>%  
 mutate(dep\_none = case\_when(deprawsc < 5 ~ 1, # creates bin for no/minimal depression  
 deprawsc > 4 ~ 0)) %>%  
 mutate(dep\_mild = case\_when((deprawsc > 4 & deprawsc < 10) ~ 1, # bin for mild depression  
 (deprawsc < 5 | deprawsc > 9) ~ 0)) %>%  
 mutate(dep\_mod = case\_when((deprawsc > 9 & deprawsc < 15) ~ 1, # bin for moderate depression  
 (deprawsc > 14 | deprawsc < 10) ~ 0)) %>%  
 mutate(dep\_modsev = case\_when((deprawsc > 14 & deprawsc < 20) ~ 1, # bin for mod-severe depression  
 (deprawsc > 19 | deprawsc < 15) ~ 0)) %>%  
 mutate(dep\_sev = case\_when(deprawsc > 19 ~ 1, # bin for severe depression  
 deprawsc < 20 ~ 0)) %>%  
 mutate(dep\_na = case\_when(is.na(deprawsc) ~ 1, # filter out any NAs  
 !is.na(deprawsc) ~ 0))  
testdf.v20$deprawsc <- as.numeric(testdf.v20$deprawsc)  
  
# Check outputs  
table(testdf.v20$dep\_none)

##   
## 0 1   
## 8429 3967

table(testdf.v20$dep\_mild)

##   
## 0 1   
## 8377 4019

table(testdf.v20$dep\_mod)

##   
## 0 1   
## 10157 2239

table(testdf.v20$dep\_modsev)

##   
## 0 1   
## 11062 1334

table(testdf.v20$dep\_sev)

##   
## 0 1   
## 11559 837

table(testdf.v20$dep\_na)

##   
## 0   
## 12396

testdf.v20 <- testdf.v20 %>%  
 mutate(dep\_type = case\_when(dep\_none == 1 ~ 1,  
 dep\_mild == 1 ~ 2,  
 dep\_mod == 1 ~ 3,  
 dep\_modsev == 1 ~ 4,  
 dep\_sev == 1 ~ 5))  
  
table(testdf.v20$dep\_type, useNA = "always")

##   
## 1 2 3 4 5 <NA>   
## 3967 4019 2239 1334 837 0

## GAD7 total

testdf.v20$anx\_score <- as.numeric(testdf.v20$anx\_score)  
  
# Convert to severity  
testdf.v20 <- testdf.v20 %>%  
 mutate(anx\_type = case\_when(anx\_score > 14 ~ 4,  
 anx\_score < 5 ~ 1, # abing for no/minimal anxiety  
 anx\_score < 10 ~ 2, # bin for mild anxiety  
 anx\_score < 15 ~ 3)) %>% # bin for severe anxiety  
 filter(!is.na(anx\_type)) # filters out all NAs  
  
testdf.v20$anx\_type <- as.numeric(testdf.v20$anx\_type) # check output, skew  
jarque.bera.test(testdf.v20$anx\_type)

##   
## Jarque Bera Test  
##   
## data: testdf.v20$anx\_type  
## X-squared = 1092.1, df = 2, p-value < 2.2e-16

table(testdf.v20$anx\_type, useNA = "always") # check output

##   
## 1 2 3 4 <NA>   
## 4726 3669 2206 1594 0

## Academic impact

testdf.v20 <- testdf.v20 %>%  
 filter(!is.na(aca\_impa)) # filter for NAs  
testdf.v20$aca\_impa <- as.numeric(testdf.v20$aca\_impa)  
  
table(testdf.v20$aca\_impa, useNA = "always") # check output, skew

##   
## 1 2 3 4 <NA>   
## 2683 4058 3027 2400 0

jarque.bera.test(testdf.v20$aca\_impa)

##   
## Jarque Bera Test  
##   
## data: testdf.v20$aca\_impa  
## X-squared = 706.04, df = 2, p-value < 2.2e-16

## Depression impact

testdf.v20 <- testdf.v20 %>%  
 filter(!is.na(dep\_impa)) # filters out NAs  
testdf.v20$dep\_impa <- as.numeric(testdf.v20$dep\_impa)  
  
table(testdf.v20$dep\_impa, useNA = "always") # check output, NAs, skew

##   
## 1 2 3 4 <NA>   
## 3283 6147 1603 546 0

jarque.bera.test(testdf.v20$dep\_impa)

##   
## Jarque Bera Test  
##   
## data: testdf.v20$dep\_impa  
## X-squared = 960.63, df = 2, p-value < 2.2e-16

## Anxiety impact

testdf.v20$gad7\_impa <- as.numeric(testdf.v20$gad7\_impa)  
table(testdf.v20$gad7\_impa, useNA = "always") # check output, skew, NAs

##   
## 1 2 3 4 <NA>   
## 2528 6083 1665 604 699

# Perceived MH needs

testdf.v20 <- testdf.v20 %>%  
 # reverse coding  
 mutate(percneed = case\_when(percneed == 1 ~ 6, # strongly agrees that needs help  
 percneed == 2 ~ 5,  
 percneed == 3 ~ 4,  
 percneed == 4 ~ 3,  
 percneed == 5 ~ 2,  
 percneed == 6 ~ 1)) %>% # strongly disagree that needs help  
 filter(!is.na(percneed)) # filters for NAs  
  
table(testdf.v20$percneed, useNA = "always") # check output, NAs

##   
## 1 2 3 4 5 6 <NA>   
## 1688 1762 760 1945 2219 3199 0

# THREATS

testdf.v22 <- testdf.v20

## Depression secret

## Secretive MH  
testdf.v22$dep\_secret <- as.numeric(testdf.v22$dep\_secret)  
table(testdf.v22$dep\_secret, useNA = "always") # check output, too many NAs

##   
## 1 2 3 4 5 6 <NA>   
## 1301 1861 1866 1008 792 373 4372

## Overall school satisfaction

testdf.v22 <- testdf.v22 %>%  
 mutate(satisfied\_overall = case\_when(satisfied\_overall == 1 ~ 6, # reverse coding  
 satisfied\_overall == 2 ~ 5,  
 satisfied\_overall == 3 ~ 4,  
 satisfied\_overall == 4 ~ 3,  
 satisfied\_overall == 5 ~ 2,  
 satisfied\_overall == 6 ~ 1)) %>%   
 filter(!is.na(satisfied\_overall)) # filters out NAs  
testdf.v22$satisfied\_overall <- as.numeric(testdf.v22$satisfied\_overall)  
  
table(testdf.v22$satisfied\_overall, useNA = "always")

##   
## 1 2 3 4 5 6 <NA>   
## 1515 5458 2727 1145 490 225 0

jarque.bera.test(testdf.v22$satisfied\_overall)

##   
## Jarque Bera Test  
##   
## data: testdf.v22$satisfied\_overall  
## X-squared = 2466.8, df = 2, p-value < 2.2e-16

## AAQ total

# create AAQ\_tot variable to sum up imputed AAQ questions  
testdf.v22 <- testdf.v22 %>%  
 mutate(aaq\_tot = as.numeric(AAQ\_1) + as.numeric(AAQ\_2) + as.numeric(AAQ\_3) + as.numeric(AAQ\_4) +   
 as.numeric(AAQ\_5) + as.numeric(AAQ\_6) + as.numeric(AAQ\_7))  
  
table(testdf.v22$aaq\_tot) # check output

##   
## 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26   
## 274 132 162 173 158 157 183 258 211 216 223 218 216 186 252 230 239 244 253 216   
## 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46   
## 202 279 204 180 174 153 163 133 125 88 76 71 74 46 52 51 41 33 21 29   
## 47 48 49   
## 17 19 29

# for loop for predictive cutoff point needed  
 rs\_df <- data.frame("cutoff" = c(), "rs" = c())  
 for (i in 8:48) {  
 aaq\_bin <- ifelse(testdf.v22$aaq\_tot >= i, TRUE, FALSE)  
 model = lm(testdf.v22$sui\_idea ~ aaq\_bin)  
 rs\_df <- rbind(rs\_df, c(i, summary(model)$r.squared))  
  
 }  
 names(rs\_df) <- c("cutoff", "rs")  
 rs\_df\_arranged <- rs\_df %>%  
 arrange(desc(rs))  
 rs\_df\_arranged[1,1]

## [1] 30

# deem 'rs\_df\_arranged[1,1]' as SI cutoff  
 testdf.v22 <- testdf.v22 %>%  
 mutate(aaq\_dum = case\_when(aaq\_tot > 40 ~ 4,  
 aaq\_tot > 30 ~ 3,  
 aaq\_tot > 20 ~ 2,  
 aaq\_tot < 21 ~ 1)) %>%  
 filter(!is.na(aaq\_dum)) # filter for NAs  
testdf.v22$aaq\_dum <- as.numeric(testdf.v22$aaq\_dum)  
  
table(testdf.v22$aaq\_dum, useNA = "always") # check output, NAs

##   
## 1 2 3 4 <NA>   
## 2767 2299 1103 292 0

## Persist

testdf.v22 <- testdf.v22 %>%  
# review levels  
 mutate(persist = case\_when(persist == 1 ~ 1, # confident  
 persist == 2 ~ 2,   
 persist == 3 ~ 3,   
 persist == 4 ~ 4,   
 persist == 5 ~ 5,   
 persist == 6 ~ 6)) %>% # not confident  
 filter(!is.na(persist))   
testdf.v22$persist <- as.numeric(testdf.v22$persist)   
  
table(testdf.v22$persist, useNA = "always") # check output, skew

##   
## 1 2 3 4 5 6 <NA>   
## 3408 1991 801 152 74 35 0

jarque.bera.test(testdf.v22$persist)

##   
## Jarque Bera Test  
##   
## data: testdf.v22$persist  
## X-squared = 5278.8, df = 2, p-value < 2.2e-16

## Belonging

testdf.v22 <- testdf.v22 %>%  
 filter(!is.na(belong1)) # filters out NAs  
table(testdf.v22$belong1, useNA = "always")

##   
## 1 2 3 4 5 6 <NA>   
## 673 1968 1998 838 683 296 0

testdf.v22$belong1 <- as.numeric(testdf.v22$belong1)  
summary(testdf.v22$belong1)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 2.000 3.000 2.966 4.000 6.000

jarque.bera.test(testdf.v22$belong1)

##   
## Jarque Bera Test  
##   
## data: testdf.v22$belong1  
## X-squared = 374.93, df = 2, p-value < 2.2e-16

table(testdf.v22$belong2) # elective question, but common

##   
## 1 2 3 4 5 6   
## 573 2006 1475 549 355 163

table(testdf.v22$belong8) # nested x 2

##   
## 1 2 3 4 5 6   
## 375 729 1288 1038 1306 382

table(testdf.v22$belong9) # not as good of a proxy

##   
## 1 2 3 4 5 6   
## 787 1443 1357 723 654 148

## Flourishing Scale

testdf.v22$flourish <- as.numeric(testdf.v22$flourish)  
  
# for loop for predictive cutoff point needed  
rs\_df <- data.frame("cutoff" = c(), "rs" = c())  
for (i in 9:55) {  
 flourish\_bin <- ifelse(testdf.v22$flourish >= i, TRUE, FALSE)  
 model = lm(testdf.v22$sui\_idea ~ flourish\_bin)  
 rs\_df <- rbind(rs\_df, c(i, summary(model)$r.squared))  
  
}  
names(rs\_df) <- c("cutoff", "rs")  
rs\_df\_arranged <- rs\_df %>%   
 arrange(desc(rs))   
rs\_df\_arranged[1,1] # chi-squared high

## [1] 39

# reverse levels for scoring  
testdf.v23 <- testdf.v22 %>%  
 mutate(flourish = case\_when(flourish == 56 ~ 8,  
 flourish == 55 ~ 9,  
 flourish == 54 ~ 10,  
 flourish == 53 ~ 11,  
 flourish == 52 ~ 12,  
 flourish == 51 ~ 13,  
 flourish == 50 ~ 14,  
 flourish == 49 ~ 15,  
 flourish == 48 ~ 16,  
 flourish == 47 ~ 17,  
 flourish == 46 ~ 18,  
 flourish == 45 ~ 19,  
 flourish == 44 ~ 20,  
 flourish == 43 ~ 21,  
 flourish == 42 ~ 22,  
 flourish == 41 ~ 23,  
 flourish == 40 ~ 24,  
 flourish == 39 ~ 25,  
 flourish == 38 ~ 26,  
 flourish == 37 ~ 27,  
 flourish == 36 ~ 28,  
 flourish == 35 ~ 29,  
 flourish == 34 ~ 30,  
 flourish == 33 ~ 31,  
 flourish == 32 ~ 32,  
 flourish == 31 ~ 33,  
 flourish == 30 ~ 34,  
 flourish == 29 ~ 35,  
 flourish == 28 ~ 36,  
 flourish == 27 ~ 37,  
 flourish == 26 ~ 38,  
 flourish == 25 ~ 39,  
 flourish == 24 ~ 40,  
 flourish == 23 ~ 41,  
 flourish == 22 ~ 42,  
 flourish == 21 ~ 43,  
 flourish == 20 ~ 44,  
 flourish == 19 ~ 45,  
 flourish == 18 ~ 46,  
 flourish == 17 ~ 47,  
 flourish == 16 ~ 48,  
 flourish == 15 ~ 49,  
 flourish == 14 ~ 50,  
 flourish == 13 ~ 51,  
 flourish == 12 ~ 52,  
 flourish == 11 ~ 53,  
 flourish == 10 ~ 54,  
 flourish == 9 ~ 55,  
 flourish == 8 ~ 56)) %>%  
 filter(!is.na(flourish))  
  
testdf.v23$flourish <- as.numeric(testdf.v23$flourish)  
table(testdf.v23$flourish, useNA = "always")

##   
## 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23   
## 235 141 178 205 222 263 312 363 557 441 392 329 317 250 231 227   
## 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39   
## 197 161 160 138 124 98 116 97 96 84 69 63 55 41 44 39   
## 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55   
## 32 25 17 24 10 11 10 8 8 7 2 7 4 1 2 1   
## 56 <NA>   
## 15 0

## Informal Mental Health Support

testdf.v23$inf\_1[is.na(testdf.v23$inf\_1)] <- 0  
testdf.v23$inf\_2[is.na(testdf.v23$inf\_2)] <- 0  
testdf.v23$inf\_3[is.na(testdf.v23$inf\_3)] <- 0  
testdf.v23$inf\_4[is.na(testdf.v23$inf\_4)] <- 0  
testdf.v23$inf\_5[is.na(testdf.v23$inf\_5)] <- 0  
testdf.v23$inf\_6[is.na(testdf.v23$inf\_6)] <- 0  
testdf.v23$inf\_7[is.na(testdf.v23$inf\_7)] <- 0  
  
# change inf\_ var to numeric  
testdf.v24 <- testdf.v23 %>%   
 # create dichotomous variable for support  
 mutate(inf = case\_when(inf\_8 == 1 ~ 0, # no informal support  
 inf\_1 == 1 ~ 1, # support  
 inf\_2 == 1 ~ 1,  
 inf\_3 == 1 ~ 1,  
 inf\_4 == 1 ~ 1,  
 inf\_5 == 1 ~ 1,  
 inf\_6 == 1 ~ 1,  
 inf\_7 == 1 ~ 1)) %>%  
 filter(!is.na(inf)) # filtters out all NAs  
testdf.v24$inf <- as.logical(testdf.v24$inf)  
  
table(testdf.v24$inf, useNA ="always") # check output, skew

##   
## FALSE TRUE <NA>   
## 1680 4744 0

jarque.bera.test(testdf.v24$inf)

##   
## Jarque Bera Test  
##   
## data: testdf.v24$inf  
## X-squared = 1442.1, df = 2, p-value < 2.2e-16

testdf.v24 = subset(testdf.v24, select= -c(inf\_1:inf\_help)) # clean up vars

## BRS

testdf.v24 <- testdf.v24 %>%  
 mutate(brs\_tot = as.integer(BRS\_1) + as.integer(BRS\_2) + as.integer(BRS\_3) + as.integer(BRS\_4) + as.integer(BRS\_5))  
  
table(testdf.v24$brs\_tot)

##   
## 5 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21   
## 13 3 9 11 39 82 192 540 1119 1484 1464 854 341 155 52 14   
## 22 23 24 25   
## 7 2 1 7

testdf.v24$brs\_tot <- as.numeric(testdf.v24$brs\_tot)  
  
# for loop for predictive cutoff point needed  
rs\_df <- data.frame("cutoff" = c(), "rs" = c())  
for (i in 6:24) {  
 brs\_bin <- ifelse(testdf.v24$brs\_tot >= i, TRUE, FALSE)  
 model = lm(testdf.v24$sui\_idea ~ brs\_bin)  
 rs\_df <- rbind(rs\_df, c(i, summary(model)$r.squared))  
}  
  
names(rs\_df) <- c("cutoff", "rs")  
rs\_df\_arranged <- rs\_df %>%   
 arrange(desc(rs))   
  
# deem 'rs\_df\_arranged[1,1]' as SI cutoff  
testdf.v24 <- testdf.v24 %>%  
 mutate(brs\_dum = ifelse(brs\_tot > rs\_df\_arranged[1,1], 1, 0)) %>%  
 mutate(brs\_dum = case\_when(brs\_dum == 1 ~ 1,  
 brs\_dum == 0 ~ 0))  
testdf.v24$brs\_dum <- as.logical(testdf.v24$brs\_dum)  
  
# output  
rs\_df\_arranged[1,1]

## [1] 14

table(testdf.v24$brs\_tot, useNA = "always")

##   
## 5 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21   
## 13 3 9 11 39 82 192 540 1119 1484 1464 854 341 155 52 14   
## 22 23 24 25 <NA>   
## 7 2 1 7 35

table(testdf.v24$brs\_dum, useNA = "always")

##   
## FALSE TRUE <NA>   
## 2008 4381 35

## Distress & social support

testdf.v24$talk1\_1[is.na(testdf.v24$talk1\_1)] <- 1  
testdf.v24$talk1\_2[is.na(testdf.v24$talk1\_2)] <- 1  
testdf.v24$talk1\_3[is.na(testdf.v24$talk1\_3)] <- 1  
testdf.v24$talk1\_4[is.na(testdf.v24$talk1\_4)] <- 1  
testdf.v24$talk1\_5[is.na(testdf.v24$talk1\_5)] <- 1  
testdf.v24$talk1\_6[is.na(testdf.v24$talk1\_6)] <- 1  
testdf.v24$talk1\_7[is.na(testdf.v24$talk1\_7)] <- 1  
testdf.v24$talk1\_8[is.na(testdf.v24$talk1\_8)] <- 1  
  
testdf.v25 <- testdf.v24 %>%   
 # create dictomous variable for no talking or talking to support systems  
 mutate(talk = case\_when((talk1\_9 == 1 | talk1no == 1) ~ 1, # no support  
 (talk1\_2 == 1 | talk1\_3 == 1 | talk1\_4 == 1 | talk1\_5 == 1) ~ 0, # informal support  
 (talk1\_1 == 1 | talk1\_6 == 1 | talk1\_7 == 1 | talk1\_8 == 1) ~ 0)) # formal and informal  
testdf.v25$talk <- as.logical(testdf.v25$talk)  
  
table(testdf.v25$talk, useNA = "always") # check output

##   
## FALSE TRUE <NA>   
## 6027 397 0

testdf.v25 = subset(testdf.v25, select= -c(talk1\_1:talk1\_9, talk1sig, talk1fr,   
talk1rel, talk1no, talk1sup, talk1rm, talk1pro, talk1fam)) # clean up

## Perceived MH as failure, stigma

# original likert  
table(testdf.v25$stig\_pcv\_2)

##   
## 1 2 3 4 5 6   
## 829 2219 1631 641 247 67

table(testdf.v25$stig\_pcv\_3)

##   
## 1 2 3 4 5 6   
## 225 862 1475 1245 1346 478

table(testdf.v25$stig\_per\_2)

##   
## 1 2 3 4 5 6   
## 3422 1732 370 76 24 17

table(testdf.v25$stig\_per\_3)

##   
## 1 2 3 4 5 6   
## 80 155 369 475 1482 3077

# transform into y/n   
testdf.v26 <- testdf.v25 %>%  
 mutate(stig\_pcv2 = case\_when((stig\_pcv\_2 == 1 | stig\_pcv\_2 == 2 | stig\_pcv\_2 == 3) ~ 1,  
 (stig\_pcv\_2 == 4 | stig\_pcv\_2 == 5 | stig\_pcv\_2 == 6) ~ 0)) %>%  
 mutate(stig\_pcv3 = case\_when((stig\_pcv\_3 == 1 | stig\_pcv\_3 == 2 | stig\_pcv\_3 == 3) ~ 1,  
 (stig\_pcv\_3 == 4 | stig\_pcv\_3 == 5 | stig\_pcv\_3 == 6) ~ 0)) %>%   
 mutate(stig\_per2 = case\_when((stig\_per\_2 == 1 | stig\_per\_2 == 2 | stig\_per\_2 == 3) ~ 1,  
 (stig\_per\_2 == 4 | stig\_per\_2 == 5 | stig\_per\_2 == 6) ~ 0)) %>%  
 mutate(stig\_per3 = case\_when((stig\_per\_3 == 1 | stig\_per\_3 == 2 | stig\_per\_3 == 3) ~ 1,  
 (stig\_per\_3 == 4 | stig\_per\_3 == 5 | stig\_per\_3 == 6) ~ 0))   
  
testdf.v26$stig\_pcv2 <- as.logical(testdf.v26$stig\_pcv2)  
testdf.v26$stig\_pcv3 <- as.logical(testdf.v26$stig\_pcv3)  
testdf.v26$stig\_per2 <- as.logical(testdf.v26$stig\_per2)  
testdf.v26$stig\_per3 <- as.logical(testdf.v26$stig\_per3)  
  
table(testdf.v26$stig\_pcv2) # Most people feel that MH tx is a sign of personal failure.

##   
## FALSE TRUE   
## 955 4679

table(testdf.v26$stig\_pcv3) # Most people think less of a person who has received MH tx.

##   
## FALSE TRUE   
## 3069 2562

table(testdf.v26$stig\_per2) # I feel that receiving MH tx is a sign of personal failure.

##   
## FALSE TRUE   
## 117 5524

table(testdf.v26$stig\_per3) # I would think less of a person who has received MH tx.

##   
## FALSE TRUE   
## 5034 604

# combine sstigma questions into one binary variable  
testdf.v26 <- testdf.v26 %>%  
 mutate(mh\_stigma = case\_when(stig\_pcv2 == 1 ~ 1,  
 stig\_pcv3 == 1 ~ 1,  
 stig\_per2 == 1 ~ 1,  
 stig\_per3 == 1 ~ 1,  
 stig\_pcv2 == 0 ~ 0,  
 stig\_pcv3 == 0 ~ 0,  
 stig\_per2 == 0 ~ 0,  
 stig\_per3 == 0 ~ 0)) %>%  
 filter(!is.na(mh\_stigma)) # filters out NAs  
testdf.v26$mh\_stigma <- as.logical(testdf.v26$mh\_stigma)  
  
table(testdf.v26$mh\_stigma, useNA = "always") # check out, NAs

##   
## FALSE TRUE <NA>   
## 43 5603 0

## Therapy

# create new var for medication and/or therapy utilization  
testdf.v26 <- testdf.v26 %>%  
 mutate(ther = case\_when(ther\_cur == 1 ~ 1,  
 ther\_cur1 == 1 ~ 1,  
 ther\_ever == 3 ~ 1,  
 ther\_ever == 4 ~ 1,  
 ther\_cur == 0 ~ 0,  
 ther\_ever == 0 ~ 0)) %>%  
 filter(!is.na(ther))  
testdf.v26$ther <- as.logical(testdf.v26$ther)  
  
table(testdf.v26$ther, useNA = "always")

##   
## FALSE TRUE <NA>   
## 3728 1918 0

# TESTING DATA PREP

## Select all vars for SEM

# Prep SEM data vars  
testdf\_sem <- testdf.v26 %>% select(  
 # outcome var  
 si\_type, si\_activ,  
 # premotivation  
 age, sexual, race, gender\_tgd, sib\_freq, ed\_any, dx\_mh, meds\_any, finpast,  
 religios, ins\_no, international, undercls,  
 # life events  
 gpa\_sr, psyhx, fincur, drugs\_yn, drug\_mar, assault, binge,  
 # defeat-entrapment  
 deprawsc, anx\_score, dep\_impa, percneed,  
 # threats to internal/external moderation  
 persist, aaq\_tot, flourish, talk, ther, satisfied\_overall)   
  
# create sem matrix   
testdf\_sem <- testdf\_sem %>%   
 # mutate\_if(is.numeric, as.factor) %>%   
 data.matrix()  
# testdf\_sem <- na.omit(testdf\_sem)  
# view(testdf\_sem) # check output  
  
# # xlsx output  
testdf\_sem\_out <- as.data.frame(testdf\_sem)  
save(testdf\_sem\_out , file = "testdf\_nondummy.RData")  
load("testdf\_nondummy.RData")  
write\_xlsx(testdf\_sem\_out, path = "testdf\_nondummy.xls", col\_names = TRUE, format\_headers = TRUE)

# SUMMARY STATISTICS

## Descriptive stats

# demographics  
table(testdf.v26$age)

##   
## 18 19 20 21 22 23 24 25   
## 795 1008 1065 1009 629 448 366 326

table(testdf.v26$si\_type)

##   
## FALSE TRUE   
## 4756 890

table(testdf.v26$gender\_tgd)

##   
## FALSE TRUE   
## 5459 187

table(testdf.v26$sexual)

##   
## FALSE TRUE   
## 4221 1425

table(testdf.v26$race)

##   
## 1 2 3 4 5   
## 3657 354 569 799 267

table(testdf.v26$dx\_mh)

##   
## FALSE TRUE   
## 3404 2242

# training data  
var\_desc\_test <- describe(testdf\_sem)  
view(var\_desc\_test)  
  
# xlsx output  
save(var\_desc\_test, file = "var\_desc\_test.RData")  
load("var\_desc\_test.RData")  
write\_xlsx(var\_desc\_test, path = "var\_desc\_test.xls", col\_names = TRUE, format\_headers = TRUE)

# SEM ANALYSIS

## PLS output model

# Prep SEM data vars  
testdf\_sem1 <- testdf.v26 %>% select(  
 # outcome var  
 si\_type, si\_activ,  
 # premotivation  
 age, sexual, race, gender\_tgd, sib\_freq, ed\_any, dx\_mh, meds\_any, finpast,  
 religios, residenc, school2\_type, international, undercls, ins\_no,   
 # life events  
 gpa\_sr, psyhx, fincur, drugs\_yn, drug\_mar, assault, binge,  
 # defeat-entrapment  
 deprawsc, anx\_score, dep\_impa, percneed,  
 # threats to internal/external moderation  
 persist, aaq\_tot, flourish, talk, ther, satisfied\_overall)   
  
# create sem matrix   
testdf\_sem1 <- data.matrix(testdf\_sem1)  
   
## Dummy col SEM w/ continuous  
results\_test <- dummy\_cols(testdf\_sem1, select\_columns = c("si\_type", "race"),  
 remove\_first\_dummy = TRUE, remove\_selected\_columns = TRUE)  
  
# xlsx output  
save(results\_test, file = "testdf\_pls.RData")  
load("testdf\_pls.RData")  
write\_xlsx(results\_test, path = "testdf\_pls.xlsx", col\_names = TRUE, format\_headers = TRUE)

# Running the test dataframe for 18 and 19 year olds

test18.v26 <- testdf.v26 %>%  
 filter(age == 18 | age == 19)  
  
nrow(test18.v26)

## [1] 1803

table(test18.v26$age)

##   
## 18 19   
## 795 1008

table(test18.v26$si\_type)

##   
## FALSE TRUE   
## 1463 340

table(test18.v26$gender\_tgd)

##   
## FALSE TRUE   
## 1729 74

table(test18.v26$sexual)

##   
## FALSE TRUE   
## 1270 533

table(test18.v26$race)

##   
## 1 2 3 4 5   
## 1169 130 190 228 86

table(test18.v26$dx\_mh)

##   
## FALSE TRUE   
## 1108 695

# Prep SEM data vars  
test18\_sem <- test18.v26 %>% select(  
 # outcome var  
 si\_type, si\_activ,  
 # premotivation  
 age, sexual, race, gender\_tgd, sib\_freq, ed\_any, dx\_mh, meds\_any, finpast,  
 religios, residenc, school2\_type, international, undercls, ins\_no,  
 # life events  
 gpa\_sr, psyhx, fincur, drugs\_yn, drug\_mar, assault, binge,  
 # defeat-entrapment  
 deprawsc, anx\_score, dep\_impa, percneed,  
 # threats to internal/external moderation  
 persist, aaq\_tot, flourish, talk, ther, satisfied\_overall)   
   
## Dummy col SEM w/ continuous  
results\_test\_18 <- dummy\_cols(test18\_sem, select\_columns = c("si\_type", "race", "residenc", "school2\_type"),  
 remove\_first\_dummy = TRUE, remove\_selected\_columns = TRUE)  
  
# save output for offline SMART PLS  
results\_test\_18 <- as.data.frame(results\_test\_18)  
save(results\_test\_18, file = "results\_test\_18.RData")  
load("results\_test\_18.RData")  
write\_xlsx(results\_test\_18, path = "results\_test\_18.xls", col\_names = TRUE, format\_headers = TRUE)

# Running test frame for 20-25 year olds

test20.v26 <- testdf.v26 %>%  
 filter(age == 20 | age == 21 | age == 22 | age == 23 | age == 24 | age == 25)  
  
table(test20.v26$age)

##   
## 20 21 22 23 24 25   
## 1065 1009 629 448 366 326

table(test20.v26$si\_type)

##   
## FALSE TRUE   
## 3293 550

table(test20.v26$gender\_tgd)

##   
## FALSE TRUE   
## 3730 113

table(test20.v26$sexual)

##   
## FALSE TRUE   
## 2951 892

table(test20.v26$race)

##   
## 1 2 3 4 5   
## 2488 224 379 571 181

table(test20.v26$dx\_mh)

##   
## FALSE TRUE   
## 2296 1547

# Prep SEM data vars  
test20\_sem <- test20.v26 %>% select(  
 # outcome var  
 si\_type, si\_activ,   
 # premotivation  
 age, sexual, race, gender\_tgd, sib\_freq, ed\_any, dx\_mh, meds\_any, finpast,  
 religios, residenc, school2\_type, international, undercls, ins\_no,  
 # life events  
 gpa\_sr, psyhx, fincur, drugs\_yn, drug\_mar, assault, binge,  
 # defeat-entrapment  
 deprawsc, anx\_score, dep\_impa, percneed,  
 # threats to internal/external moderation  
 persist, aaq\_tot, flourish, talk, ther, satisfied\_overall)   
  
# create SEM matrix   
test20\_sem <- data.matrix(test20\_sem)  
   
## Dummy columns for SEM  
results\_test\_20 <- dummy\_cols(test20\_sem, select\_columns = c("si\_type", "race", "residenc", "school2\_type"),  
 remove\_first\_dummy = TRUE, remove\_selected\_columns = TRUE)  
  
nrow(testdf.v26) # 5700

## [1] 5646

nrow(traindf.v28) # 5638

## [1] 5691

nrow(test20.v26) # 3862

## [1] 3843

nrow(test18.v26) # 1838

## [1] 1803

# Save offline for SMART PLS  
results\_test\_20 <- as.data.frame(results\_test\_20)  
save(results\_test\_20, file = "results\_test\_20.RData")  
load("results\_test\_20.RData")  
write\_xlsx(results\_test\_20, path = "results\_test\_20.xls", col\_names = TRUE, format\_headers = TRUE)

## Wilcox test

testdf.v26B <- testdf.v26  
traindf.v28B <- traindf.v28  
test20.v26B <- test20.v26  
test18.v26B <- test18.v26  
  
testdf.v26B$dx\_mh <- as.numeric(testdf.v26B$dx\_mh)  
traindf.v28B$dx\_mh <- as.numeric(traindf.v28B$dx\_mh)  
test20.v26B$dx\_mh <- as.numeric(test20.v26B$dx\_mh)  
test18.v26B$dx\_mh <- as.numeric(test18.v26B$dx\_mh)  
  
testdf.v26B$si\_type <- as.numeric(testdf.v26B$si\_type)  
traindf.v28B$si\_type <- as.numeric(traindf.v28B$si\_type)  
test20.v26B$si\_type <- as.numeric(test20.v26B$si\_type)  
test18.v26B$si\_type <- as.numeric(test18.v26B$si\_type)  
  
testdf.v26B$gender\_tgd <- as.numeric(testdf.v26B$gender\_tgd)  
traindf.v28B$gender\_tgd <- as.numeric(traindf.v28B$gender\_tgd)  
test20.v26B$gender\_tgd <- as.numeric(test20.v26B$gender\_tgd)  
test18.v26B$gender\_tgd <- as.numeric(test18.v26B$gender\_tgd)  
  
testdf.v26B$sexual<- as.numeric(testdf.v26B$sexual)  
traindf.v28B$sexual <- as.numeric(traindf.v28B$sexual)  
test20.v26B$sexual <- as.numeric(test20.v26B$sexual)  
test18.v26B$sexual <- as.numeric(test18.v26B$sexual)  
  
  
# training v. testing  
wilcox.test(testdf.v26B$age,traindf.v28B$age, alternative = "two.sided") # W = 15974521, p-value = 0.5965

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: testdf.v26B$age and traindf.v28B$age  
## W = 15974521, p-value = 0.5965  
## alternative hypothesis: true location shift is not equal to 0

wilcox.test(testdf.v26B$si\_type,traindf.v28B$si\_type, alternative = "two.sided") # W = 16274859, p-value = 0.05296

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: testdf.v26B$si\_type and traindf.v28B$si\_type  
## W = 16274859, p-value = 0.05296  
## alternative hypothesis: true location shift is not equal to 0

wilcox.test(testdf.v26B$gender\_tgd,traindf.v28B$gender\_tgd, alternative = "two.sided") # W = 16038848, p-value = 0.6233

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: testdf.v26B$gender\_tgd and traindf.v28B$gender\_tgd  
## W = 16038848, p-value = 0.6233  
## alternative hypothesis: true location shift is not equal to 0

wilcox.test(testdf.v26B$sexual, traindf.v28B$sexual, alternative = "two.sided") # W = 16083640, p-value = 0.891

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: testdf.v26B$sexual and traindf.v28B$sexual  
## W = 16083640, p-value = 0.891  
## alternative hypothesis: true location shift is not equal to 0

wilcox.test(testdf.v26B$race, traindf.v28B$race, alternative = "two.sided") # W = 16010644, p-value = 0.7108

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: testdf.v26B$race and traindf.v28B$race  
## W = 16010644, p-value = 0.7108  
## alternative hypothesis: true location shift is not equal to 0

wilcox.test(testdf.v26B$dx\_mh, traindf.v28B$dx\_mh, alternative = "two.sided") # W = 16325040, p-value = 0.07795

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: testdf.v26B$dx\_mh and traindf.v28B$dx\_mh  
## W = 16325040, p-value = 0.07795  
## alternative hypothesis: true location shift is not equal to 0

# 18-19 v. 20-25  
wilcox.test(test20.v26B$age, test18.v26B$age, alternative = "two.sided") # W = 6928929, p-value < 2.2e-16 \*\*\*

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: test20.v26B$age and test18.v26B$age  
## W = 6928929, p-value < 2.2e-16  
## alternative hypothesis: true location shift is not equal to 0

wilcox.test(test20.v26B$si\_type, test18.v26B$si\_type, alternative = "two.sided") # W = 3306980, p-value = 1.244e-05 \*\*\*

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: test20.v26B$si\_type and test18.v26B$si\_type  
## W = 3306980, p-value = 1.244e-05  
## alternative hypothesis: true location shift is not equal to 0

wilcox.test(test20.v26B$gender\_tgd, test18.v26B$gender\_tgd, alternative = "two.sided") # W = 3424143, p-value = 0.02272 \*

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: test20.v26B$gender\_tgd and test18.v26B$gender\_tgd  
## W = 3424143, p-value = 0.02272  
## alternative hypothesis: true location shift is not equal to 0

wilcox.test(test20.v26B$sexual, test18.v26B$sexual, alternative = "two.sided") # W = 3244443, p-value = 3.035e-07 \*\*\*

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: test20.v26B$sexual and test18.v26B$sexual  
## W = 3244443, p-value = 3.035e-07  
## alternative hypothesis: true location shift is not equal to 0

wilcox.test(test20.v26B$race, test18.v26B$race, alternative = "two.sided") # = 3492851, p-value = 0.5591

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: test20.v26B$race and test18.v26B$race  
## W = 3492851, p-value = 0.5591  
## alternative hypothesis: true location shift is not equal to 0

wilcox.test(test20.v26B$dx\_mh, test18.v26B$dx\_mh, alternative = "two.sided") # W = 3523642, p-value = 0.2214

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: test20.v26B$dx\_mh and test18.v26B$dx\_mh  
## W = 3523642, p-value = 0.2214  
## alternative hypothesis: true location shift is not equal to 0