**##load dataset**

load("~/DaSH457/Rdataset/aconf2.RData")

**##create library path for packages**

.libPaths("H:/DaSH457/Rpackages")

**##install packages**

install.packages("lattice")

install.packages("sandwich")

install.packages("Rtools")

install.packages("MKmisc")

install.packages("forcats")

install.packages("car")

install.packages("compute.es")

install.packages("effectsize")

install.packages("lsr")

install.packages("ggplot2")

install.packages("ggpubr")

install.packages("AER")

install.packages("dplyr")

install.packages("naniar")

install.packages("mice")

install.packages("MASS")

install.packages("mice")

install.packages("tidyr")

install.packages("dplyr")

install.packages("tidyverse")

install.packages("broom")

install.packages("broom.mixed")

install.packages("miceadds")

install.packages("epitools")

install.packages("effectsize")

install.packages("DescTools")

install.packages("misty")

install.packages("VIM")

install.packages("finalfit")

install.packages("lme4")

install.packages("psych")

install.packages("lava")

install.packages("miceafter")

install.packages("sjstats")

install.packages("moments")

**##load packages**

library("sjmisc")

library("MKmisc")

library("forcats")

library("car")

library("compute.es")

library("effectsize")

library("lsr")

library("ggplot2")

library("ggpubr")

library("AER")

library("dplyr")

library("naniar")

library("mice")

library("MASS")

library("tidyr")

library("dplyr")

library("tidyverse")

library("broom")

library("broom.mixed")

library("miceadds")

library("epitools")

library("effectsize")

library("DescTools")

library("misty")

library("VIM")

library("finalfit")

library("lme4")

library("lava")

library("miclust")

library("sjstats")

library("lattice")

**##save dataset to knew name to manipulate and remove irrelevant variables**

binlogtest <- aconf

haven::as\_factor(binlogtest$fs004)

binlogtest$EdAtt <- binlogtest$q1098\_total

binlogtest2 <- binlogtest

binlogtest2 %>% remove\_var("q1098a", "q1098b", "q1098c", "q1098d", "q1098e", "q1098f", "q1098g", "q1098h", "q1098i", "q1098\_total")

binlogtest2 <- subset(binlogtest2, select = -c(q1098a, q1098b, q1098c, q1098d, q1098e, q1098f, q1098g, q1098h, q1098i, q1098\_total))

view(binlogtest2)

**##recode variables to factors or numeric and define levels**

is.numeric(binlogtest2$fs004)

binlogtest2$fs004 <- as.factor(binlogtest2$fs004)

is.factor(binlogtest2$fs004)

levels(binlogtest2$fs004)

binlogtest2$fs004 <- factor(binlogtest2$fs004, c(1,2), labels = c('Male', 'Female'))

levels(binlogtest2$fs004)

is.numeric(binlogtest2$fs030)

binlogtest2$fs030 <- as.factor(binlogtest2$fs030)

is.factor(binlogtest2$fs030)

levels(binlogtest2$fs030)

binlogtest2$fs030 <- factor(binlogtest2$fs030, c(1,2), labels = c('CWoe', 'CWE'))

is.numeric(binlogtest2$fs071)

binlogtest2$fs071 <- as.factor(binlogtest2$fs071)

as.factor(binlogtest2$fs071)

is.factor(binlogtest2$fs071)

binlogtest2$fs071 <- fct\_collapse(binlogtest2$fs071, No = c("1", "2"), Yes = c("3", "4", "5"))

levels(binlogtest2$fs071)

as.factor(binlogtest2$fs071)

is.numeric(binlogtest2$peerstat)

binlogtest2$peerstat <- as.ordered(binlogtest2$peerstat)

as.factor(binlogtest2$peerstat)

binlogtest2$peerstat <- fct\_collapse(binlogtest2$peerstat, Low = c("2", "3"), High = c("4", "5", "6"))

levels(binlogtest2$peerstat)

as.factor(binlogtest2$peerstat)

is.factor(binlogtest2$peerstat)

is.numeric(binlogtest2$q1073)

as.factor(binlogtest2$q1073)

binlogtest2$q1073 <- as.factor(binlogtest2$q1073)

is.factor(binlogtest2$q1073)

binlogtest2$q1073 <- fct\_collapse(binlogtest2$q1073, Daily = c("1", "2", "7"), WeeklyNever = c("3", "4", "5", "6"))

as.factor(binlogtest2$q1073)

levels(binlogtest2$q1073)

is.numeric(binlogtest2$EdAtt)

binlogtest2$EdAtt <- as.numeric(binlogtest2$EdAtt)

is.numeric(binlogtest2$EdAtt)

binlogtest2$EdAtt[binlogtest2$EdAtt==0]<-NA

is.na(binlogtest2$EdAtt)

binlogtest2$EdAtt <- as.factor(binlogtest2$EdAtt)

levels(binlogtest2$EdAtt)

binlogtest2$EdAtt <- fct\_collapse(binlogtest2$EdAtt, None = c("1", "2", "3"), Moderate = c("4", "5", "6"), Higher = c("7", "8", "9"))

levels(binlogtest2$EdAtt)

as.factor(binlogtest2$EdAtt)

is.factor(binlogtest2$EdAtt)

levels(binlogtest2$EdAtt)

is.numeric(binlogtest2$f\_rgsc91)

binlogtest2$f\_rgsc91 <- as.factor(binlogtest2$f\_rgsc91)

levels(binlogtest2$f\_rgsc91)

binlogtest2$f\_rgsc91 <- fct\_collapse(binlogtest2$f\_rgsc91, ProfManSki = c("1", "2", "3"), PartUn = c("4", "5"))

levels(binlogtest2$f\_rgsc91)

as.factor(binlogtest2$f\_rgsc91)

is.factor(binlogtest2$f\_rgsc91)

is.numeric(binlogtest2$q1rg91)

binlogtest2$q1rg91 <- as.factor(binlogtest2$q1rg91)

binlogtest2$q1rg91 <- fct\_collapse(binlogtest2$q1rg91, ProfManTech = c("1", "2", "3"), PartUn = c("4", "5"))

levels(binlogtest2$q1rg91)

as.factor(binlogtest2$q1rg91)

levels(binlogtest2$q1rg91)

is.factor(binlogtest2$q1rg91)

**##create tables for factors and histograms for continuous to assess distribution and frequencies**

table(binlogtest2$fs030)

table(binlogtest2$fs004)

table(binlogtest2$fs071)

table(binlogtest2$peerstat)

table(binlogtest2$q1073)

table(binlogtest2$EdAtt)

table(binlogtest$f\_rgsc91)

table(binlogtest$q1rg91)

threewaytab1 <- xtabs(~binlogtest2$fs004 + binlogtest2$fs030 + binlogtest2$q1rg91)

threewayftab1 <- ftable(threewaytab1)

threewayftab1

threewaytab2 <- xtabs(~binlogtest2$fs071 + binlogtest2$fs030 + binlogtest2$q1rg91)

threewayftab2 <- ftable(threewaytab2)

threewayftab2

threewaytab3 <- xtabs(~binlogtest2$peerstat + binlogtest2$fs030 + binlogtest2$q1rg91)

threewayftab3 <- ftable(threewaytab3)

threewayftab3

threewaytab4 <- xtabs(~binlogtest2$q1073 + binlogtest2$fs030 + binlogtest2$q1rg91)

threewayftab4 <- ftable(threewaytab4)

threewayftab4

threewaytab5 <- xtabs(~binlogtest2$EdAtt + binlogtest2$fs030 + binlogtest2$q1rg91)

threewayftab5 <- ftable(threewaytab5)

threewayftab5

threewaytab6 <- xtabs(~binlogtest2$f\_rgsc91 + binlogtest2$fs030 + binlogtest2$q1rg91)

threewayftab6 <- ftable(threewaytab6)

threewayftab6

hist(binlogtest2$rs175)

hist(binlogtest2$rs179)

hist(binlogtest2$rs180)

hist(binlogtest2$rs181)

hist(binlogtest2$TotalRutterA)

skewness(binlogtest2$rs175)

skewness(binlogtest2$rs179)

skewness(binlogtest2$rs180)

skewness(binlogtest2$rs181)

skewness(binlogtest2$TotalRutterA)

ggdensity(binlogtest2, x = "rs175", fill = "lightgrey", add = "mean", rug = TRUE, color = "fs030", palette = c("#00AFBB", "#E7B800"), linetype = "dashed")

ggqqplot(binlogtest2, x = "rs175")

ggqqplot(binlogtest2, x = "rs179")

ggqqplot(binlogtest2, x = "rs180")

ggqqplot(binlogtest2, x = "rs181")

ggqqplot(binlogtest2, x = "TotalRutterA")

**##assess mean and range of continuous variables**

tapply(binlogtest2$rs175, binlogtest2$fs030, mean, na.rm=T)

tapply(binlogtest2$rs175, binlogtest2$fs030, median, na.rm=T)

tapply(binlogtest2$rs175, binlogtest2$fs030, range, na.rm=T)

tapply(binlogtest2$rs175, binlogtest2$fs030, IQR, na.rm=T)

tapply(binlogtest2$TotalRutterA, binlogtest2$fs030, mean, na.rm=T)

tapply(binlogtest2$TotalRutterA, binlogtest2$fs030, median, na.rm=T)

tapply(binlogtest2$TotalRutterA, binlogtest2$fs030, range, na.rm=T)

tapply(binlogtest2$TotalRutterA, binlogtest2$fs030, IQR, na.rm=T)

tapply(binlogtest2$rs180, binlogtest2$fs030, mean, na.rm=T)

tapply(binlogtest2$rs180, binlogtest2$fs030, median, na.rm=T)

tapply(binlogtest2$rs180, binlogtest2$fs030, range, na.rm=T)

tapply(binlogtest2$rs180, binlogtest2$fs030, IQR, na.rm=T)

tapply(binlogtest2$rs181, binlogtest2$fs030, mean, na.rm=T)

tapply(binlogtest2$rs181, binlogtest2$fs030, median, na.rm=T)

tapply(binlogtest2$rs181, binlogtest2$fs030, range, na.rm=T)

tapply(binlogtest2$rs181, binlogtest2$fs030, IQR, na.rm=T)

tapply(binlogtest2$rs179, binlogtest2$fs030, mean, na.rm=T)

tapply(binlogtest2$rs179, binlogtest2$fs030, median, na.rm=T)

tapply(binlogtest2$rs179, binlogtest2$fs030, range, na.rm=T)

tapply(binlogtest2$rs179, binlogtest2$fs030, IQR, na.rm=T)

boxplot(binlogtest2$rs175 ~ binlogtest2$fs030)

boxplot(binlogtest2$rs179 ~ binlogtest2$fs030)

boxplot(binlogtest2$rs180 ~ binlogtest2$fs030)

boxplot(binlogtest2$rs181 ~ binlogtest2$fs030)

boxplot(binlogtest2$TotalRutterA ~ binlogtest2$fs030)

tapply(binlogtest2$rs175, binlogtest2$q1rg91, mean, na.rm=T)

tapply(binlogtest2$rs175, binlogtest2$q1rg91, median, na.rm=T)

tapply(binlogtest2$rs175, binlogtest2$q1rg91, range, na.rm=T)

tapply(binlogtest2$rs175, binlogtest2$q1rg91, IQR, na.rm=T)

tapply(binlogtest2$TotalRutterA, binlogtest2$q1rg91, mean, na.rm=T)

tapply(binlogtest2$TotalRutterA, binlogtest2$q1rg91, median, na.rm=T)

tapply(binlogtest2$TotalRutterA, binlogtest2$q1rg91, range, na.rm=T)

tapply(binlogtest2$TotalRutterA, binlogtest2$q1rg91, IQR, na.rm=T)

tapply(binlogtest2$rs180, binlogtest2$q1rg91, mean, na.rm=T)

tapply(binlogtest2$rs180, binlogtest2$q1rg91, median, na.rm=T)

tapply(binlogtest2$rs180, binlogtest2$q1rg91, range, na.rm=T)

tapply(binlogtest2$rs180, binlogtest2$q1rg91, IQR, na.rm=T)

tapply(binlogtest2$rs181, binlogtest2$q1rg91, mean, na.rm=T)

tapply(binlogtest2$rs181, binlogtest2$q1rg91, median, na.rm=T)

tapply(binlogtest2$rs181, binlogtest2$q1rg91, range, na.rm=T)

tapply(binlogtest2$rs181, binlogtest2$q1rg91, IQR, na.rm=T)

tapply(binlogtest2$rs179, binlogtest2$q1rg91, mean, na.rm=T)

tapply(binlogtest2$rs179, binlogtest2$q1rg91, median, na.rm=T)

tapply(binlogtest2$rs179, binlogtest2$q1rg91, range, na.rm=T)

tapply(binlogtest2$rs179, binlogtest2$q1rg91, IQR, na.rm=T)

**#Ho median scores for cotninuous variables are equal between CWoE and CWoE**

**#conduct a two-sided wilcoxon test to test the assumption**

wilcox.test(binlogtest2$rs175 ~ binlogtest2$fs030, mu=0, alt="two.sided", conf.int=TRUE, conf.level=0.95, paired=FALSE, exact=FALSE, correct=TRUE)

wilcox.test(binlogtest2$rs179 ~ binlogtest2$fs030, mu=0, alt="two.sided", conf.int=TRUE, conf.level=0.95, paired=FALSE, exact=FALSE, correct=TRUE)

wilcox.test(binlogtest2$rs180 ~ binlogtest2$fs030, mu=0, alt="two.sided", conf.int=TRUE, conf.level=0.95, paired=FALSE, exact=FALSE, correct=TRUE)

wilcox.test(binlogtest2$rs181 ~ binlogtest2$fs030, mu=0, alt="two.sided", conf.int=TRUE, conf.level=0.95, paired=FALSE, exact=FALSE, correct=TRUE)

wilcox.test(binlogtest2$TotalRutterA ~ binlogtest2$fs030, mu=0, alt="two.sided", conf.int=TRUE, conf.level=0.95, paired=FALSE, exact=FALSE, correct=TRUE)

**##conduct a complete-case analysis and compare findings to findings from pooled imputed data under MAR assumption**

**##chisquare**

sextab <- table(binlogtest2$fs030, binlogtest2$fs004)

chisq.test(binlogtest2$fs030, binlogtest2$fs004)

oddsratio(sextab,method = "wald")

sextab2 <- table(binlogtest2$q1rg91, binlogtest2$fs004)

chisq.test(binlogtest2$q1rg91, binlogtest2$fs004)

oddsratio(sextab2,method = "wald")

fs071tab <- table(binlogtest2$fs030, binlogtest2$fs071)

chisq.test(binlogtest2$fs030, binlogtest2$fs071)

oddsratio(fs071tab,method = "wald")

fs071tab2 <- table(binlogtest2$q1rg91, binlogtest2$fs071)

chisq.test(binlogtest2$q1rg91, binlogtest2$fs071)

oddsratio(fs071tab2,method = "wald")

peertab <- table(binlogtest2$fs030, binlogtest2$peerstat)

chisq.test(binlogtest2$fs030, binlogtest2$peerstat)

oddsratio(peertab,method = "wald")

peertab2 <- table(binlogtest2$q1rg91, binlogtest2$peerstat)

chisq.test(binlogtest2$q1rg91, binlogtest2$peerstat)

oddsratio(peertab2,method = "wald")

q1073tab <- table(binlogtest2$fs030, binlogtest2$q1073)

chisq.test(binlogtest2$fs030, binlogtest2$q1073)

oddsratio(q1073tab, method = "wald")

q1073tab2 <- table(binlogtest2$q1rg91, binlogtest2$q1073)

chisq.test(binlogtest2$q1rg91, binlogtest2$q1073)

oddsratio(q1073tab2, method = "wald")

edtab <- table(binlogtest2$fs030, binlogtest2$EdAtt)

chisq.test(binlogtest2$fs030, binlogtest2$EdAtt)

oddsratio(edtab,method = "wald")

edtab2 <- table(binlogtest2$q1rg91, binlogtest2$EdAtt)

chisq.test(binlogtest2$q1rg91, binlogtest2$EdAtt)

oddsratio(edtab2,method = "wald")

fsestab <- table(binlogtest2$fs030, binlogtest2$f\_rgsc91)

chisq.test(binlogtest2$fs030, binlogtest2$f\_rgsc91)

oddsratio(fsestab,method = "wald")

fsestab2 <- table(binlogtest2$q1rg91, binlogtest2$f\_rgsc91)

chisq.test(binlogtest2$q1rg91, binlogtest2$f\_rgsc91)

oddsratio(fsestab2,method = "wald")

asestab <- table(binlogtest2$fs030, binlogtest2$q1rg91)

chisq.test(binlogtest2$fs030, binlogtest2$q1rg91)

oddsratio(asestab,method = "wald")

asestab2 <- table(binlogtest2$fs030, binlogtest2$q1rg91)

chisq.test(binlogtest2$fs030, binlogtest2$q1rg91)

oddsratio(asestab2,method = "wald")

**##complete-case analysis - t-test**

shapiro.test(binlogtest2$TotalRutterA)

shapiro.test(binlogtest2$rs175)

shapiro.test(binlogtest2$rs179)

shapiro.test(binlogtest2$rs180)

shapiro.test(binlogtest2$rs181)

wilcox.test(binlogtest2$TotalRutterA ~ binlogtest2$fs030)

wilcox.test(binlogtest2$rs175 ~ binlogtest2$fs030)

wilcox.test(binlogtest2$rs179 ~ binlogtest2$fs030)

wilcox.test(binlogtest2$rs180 ~ binlogtest2$fs030)

wilcox.test(binlogtest2$rs181 ~ binlogtest2$fs030)

mwu(binlogtest2, rs175, fs030)

mwu(binlogtest2, TotalRutterA, fs030)

mwu(binlogtest2, rs179, fs030)

mwu(binlogtest2, rs180, fs030)

mwu(binlogtest2, rs181, fs030)

binlogtest2$logrs175 <- log(binlogtest2$rs175)\*binlogtest2$rs175

binlogtest2$logrs179 <- log(binlogtest2$rs179)\*binlogtest2$rs179

binlogtest2$logrs180 <- log(binlogtest2$rs180)\*binlogtest2$rs180

binlogtest2$logrs181 <- log(binlogtest2$rs181)\*binlogtest2$rs181

binlogtest2$logTotalRutterA <- log(binlogtest2$TotalRutterA)\*binlogtest2$TotalRutterA

hist(binlogtest2$logrs175)

hist(binlogtest2$logrs179)

hist(binlogtest2$logrs180)

hist(binlogtest2$logrs181)

hist(binlogtest2$logTotalRutterA)

shapiro.test(binlogtest2$logTotalRutterA)

shapiro.test(binlogtest2$logrs175)

shapiro.test(binlogtest2$logrs179)

shapiro.test(binlogtest2$logrs180)

shapiro.test(binlogtest2$logrs181)

wilcox.test(binlogtest2$logTotalRutterA ~ binlogtest2$fs030)

wilcox.test(binlogtest2$logrs175 ~ binlogtest2$fs030)

wilcox.test(binlogtest2$logrs179 ~ binlogtest2$fs030)

wilcox.test(binlogtest2$logrs180 ~ binlogtest2$fs030)

wilcox.test(binlogtest2$logrs181 ~ binlogtest2$fs030)

mwu(binlogtest2, logrs175, fs030)

mwu(binlogtest2, logTotalRutterA, fs030)

mwu(binlogtest2, logrs179, fs030)

mwu(binlogtest2, logrs180, fs030)

mwu(binlogtest2, logrs181, fs030)

TRA <- t.test(binlogtest2$TotalRutterA ~ binlogtest2$fs030, var.equal = T)

cohens\_d(TRA)

fit1 <- lm(binlogtest2$logTotalRutterA ~ binlogtest2$fs030)

fit1

lattice::densityplot(~residuals(fit1), group = fs030, data = binlogtest2, auto.key = T)

fit2 <- lm(binlogtest2$logrs175b ~ binlogtest2$fs030)

fit2

lattice::densityplot(~residuals(fit2), group = fs030, data = binlogtest2, auto.key = T)

fit3 <- lm(binlogtest2$logrs179 ~ binlogtest2$fs030)

fit3

lattice::densityplot(~residuals(fit3), group = fs030, data = binlogtest2, auto.key = T)

fit4 <- lm(binlogtest2$logrs180 ~ binlogtest2$fs030)

fit4

lattice::densityplot(~residuals(fit4), group = fs030, data = binlogtest2, auto.key = T)

fit5 <- lm(binlogtest2$logrs181 ~ binlogtest2$fs030)

fit5

lattice::densityplot(~residuals(fit5), group = fs030, data = binlogtest2, auto.key = T)

t.test(binlogtest2$TotalRutterA ~ binlogtest2$fs030)

cohens\_d(binlogtest2$TotalRutterA ~ binlogtest2$fs030)

MeanDiffCI(binlogtest2$TotalRutterA ~ binlogtest2$fs030)

t.test(binlogtest2$rs175 ~ binlogtest2$fs030)

cohens\_d(binlogtest2$rs175 ~ binlogtest2$fs030)

MeanDiffCI(binlogtest2$rs175 ~ binlogtest2$fs030)

t.test(binlogtest2$rs179 ~ binlogtest2$fs030)

cohens\_d(binlogtest2$rs179 ~ binlogtest2$fs030)

MeanDiffCI(binlogtest2$rs179 ~ binlogtest2$fs030)

t.test(binlogtest2$rs180 ~ binlogtest2$fs030)

cohens\_d(binlogtest2$rs180 ~ binlogtest2$fs030)

MeanDiffCI(binlogtest2$rs180 ~ binlogtest2$fs030)

t.test(binlogtest2$rs181 ~ binlogtest2$fs030)

cohens\_d(binlogtest2$rs181 ~ binlogtest2$fs030)

MeanDiffCI(binlogtest2$rs181 ~ binlogtest2$fs030)

t.test(binlogtest2$TotalRutterA ~ binlogtest2$q1rg91)

cohens\_d(binlogtest2$TotalRutterA ~ binlogtest2$q1rg91)

MeanDiffCI(binlogtest2$TotalRutterA ~ binlogtest2$q1rg91)

t.test(binlogtest2$rs175 ~ binlogtest2$q1rg91)

cohens\_d(binlogtest2$rs175 ~ binlogtest2$q1rg91)

MeanDiffCI(binlogtest2$rs175 ~ binlogtest2$q1rg91)

t.test(binlogtest2$rs179 ~ binlogtest2$q1rg91)

cohens\_d(binlogtest2$rs179 ~ binlogtest2$q1rg91)

MeanDiffCI(binlogtest2$rs179 ~ binlogtest2$q1rg91)

t.test(binlogtest2$rs180 ~ binlogtest2$q1rg91)

cohens\_d(binlogtest2$rs180 ~ binlogtest2$q1rg91)

MeanDiffCI(binlogtest2$rs180 ~ binlogtest2$q1rg91)

t.test(binlogtest2$rs181 ~ binlogtest2$q1rg91)

cohens\_d(binlogtest2$rs181 ~ binlogtest2$q1rg91)

MeanDiffCI(binlogtest2$rs181 ~ binlogtest2$q1rg91)

**##conduct unadjusted binary logistic regression on complete-case data**

binlogtest3 <- na.omit(binlogtest2)

CCunfit <- glm(q1rg91 ~ fs030, family = binomial(link="logit"), data = binlogtest3)

summary(CCunfit)

AIC(logLik(CCunfit))

BIC(logLik(CCunfit))

PseudoR2(CCunfit)

ctable <- coef(summary(CCunfit))

print(ctable)

parameters::model\_parameters(CCunfit, exp = TRUE)

**##conduct adjusted binary logistic regression on complete-case data for IVs associated with outcome only**

binlogtest3 <- na.omit(binlogtest2)

CCadfit2 <- glm(q1rg91 ~ fs030 + EdAtt + f\_rgsc91 + rs175 + rs179 + rs180 + rs181 + fs004, family = binomial(link="logit"), data = binlogtest3)

summary(CCadfit2)

AIC(logLik(CCadfit2))

BIC(logLik(CCadfit2))

PseudoR2(CCadfit2)

check.collin(CCadfit2)

ctable <- coef(summary(CCadfit2))

print(ctable2)

parameters::model\_parameters(CCadfit2, exp = TRUE)

**##conduct adjusted binary logistic regression on complete-case data for all IVs with outcome**

binlogtest3 <- na.omit(binlogtest2)

CCadfit <- glm(q1rg91 ~ fs030 + fs071 + peerstat + q1073 + EdAtt + f\_rgsc91 + rs175 + rs179 + rs180 + rs181 + TotalRutterA + fs004, family = binomial(link="logit"), data = binlogtest3)

summary(CCadfit)

AIC(logLik(CCadfit))

BIC(logLik(CCadfit))

PseudoR2(CCadfit)

check.collin(CCadfit)

ctable <- coef(summary(CCadfit))

print(ctable)

parameters::model\_parameters(CCadfit, exp = TRUE)

**##check residuals of complete case data**

binlogtest3$residuals <- resid(CCadfit)

binlogtest3$standardized.residuals <- rstandard(CCadfit)

binlogtest3$studentized.residuals <- rstudent(CCadfit)

binlogtest3$cooks.distance <- cooks.distance(CCadfit)

binlogtest3$dfbeta <- dfbeta(CCadfit)

binlogtest3$dffit <- dffits(CCadfit)

binlogtest3$leverage <- hatvalues(CCadfit)

binlogtest3$covariance.ratios <- covratio(CCadfit)

binlogtest3

binlogtest3$standardized.residuals > 2 | binlogtest3$standardized.residuals < -2

round(binlogtest3, digits = 3)

binlogtest3$large.residual <- binlogtest3$standardized.residuals > 2 | binlogtest3$standardized.residuals < -2

sum(binlogtest3$large.residual)

binlogtest3[binlogtest3$large.residual, c("fs030", "fs071", "peerstat", "q1073", "EdAtt", "f\_rgsc91", "rs175", "rs179", "rs180", "rs181", "TotalRutterA", "fs004", "standardized.residuals")]

cooks <- binlogtest3[binlogtest3$large.residual, c("cooks.distance", "leverage", "covariance.ratios", "standardized.residuals", "studentized.residuals", "dfbeta")]

print(cooks, n=23)

**##Assess linearity of the logit for cont.var**

binlogtest3$logrs175 <- log(binlogtest3$rs175)\*binlogtest3$rs175

binlogtest3$logrs179 <- log(binlogtest3$rs179)\*binlogtest3$rs179

binlogtest3$logrs180 <- log(binlogtest3$rs180)\*binlogtest3$rs180

binlogtest3$logrs181 <- log(binlogtest3$rs181)\*binlogtest3$rs181

binlogtest3$logTotalRutterA <- log(binlogtest3$TotalRutterA)\*binlogtest3$TotalRutterA

penaltyTest.1 <- glm(q1rg91 ~ fs030 + fs071 + peerstat + q1073 + EdAtt + f\_rgsc91 + rs175 + rs179 + rs180 + rs181 + TotalRutterA + fs004 + logrs175 + logrs179 + logrs180 + logrs181 + logTotalRutterA, family = binomial(link="logit"), data = binlogtest3)

summary(penaltyTest.1)

CCadfitLogs <- glm(q1rg91 ~ fs030 + fs071 + peerstat + q1073 + EdAtt + f\_rgsc91 + fs004 + logrs175 + logrs179 + logrs180 + logrs181 + logTotalRutterA, family = binomial(link="logit"), data = binlogtest3)

summary(CCadfitLogs)

AIC(logLik(CCadfitLogs))

BIC(logLik(CCadfitLogs))

PseudoR2(CCadfitLogs)

check.collin(CCadfitLogs)

ctable <- coef(summary(CCadfitLogs))

print(ctable)

parameters::model\_parameters(CCadfitLogs, exp = TRUE)

**##conduct MLM binariy logistic regression with 2 levels, fixed slope, random intercept**

MLM2 <- glmer(q1rg91 ~ fs030 + (1 | fs004), data = binlogtest, family = binomial(link="logit"), control = glmerControl(optimizer = "bobyqa"), nAGQ = 0)

summary(MLM2)

AIC(logLik(MLM2))

**##conduct binariy logistic regression with 2 levels, randome slope, random intercept**

ccfit <- glmer(q1rg91 ~ fs030 + fs071 + peerstat + q1073 + q1098\_total + f\_rgsc91 + rs175 + rs179 + rs180 + rs181 + TotalRutterA + (fs030 | fs004), data = binlogtest, family = binomial(link="logit"), control = glmerControl(optimizer = "bobyqa"), nAGQ = 0)

summary(ccfit)

ctable <- coef(summary(ccfit))

print(ctable)

exp(fixef(ccfit))

parameters::model\_parameters(ccfit, exp = TRUE)

**##check correlation of variables (correlation test or vif)**

**##check linearity between continuous predictor variable and the log odds of the outcomes (box-tidwell test or graphs)**

##correlation of MLM

library(car)

install.packages("sjstats")

library(sjstats)

cor.test(ccfit, method = "spearman")

performance::icc(ccfit)

**##boxtidwell**

library(car)

boxTidwell(q1rg91 ~ rs175 + rs179 + rs180 + rs181, ~ fs030 + fs004 + fs071 + peerstat + q1073 + q1098\_total + f\_rgsc91, data=binlogtest)

**##scatterplot**

library(dplyr)

binlogtest2 <- na.omit(binlogtest)

ccfit2 <- glmer(q1rg91 ~ fs030 + fs071 + peerstat + q1073 + q1098\_total + f\_rgsc91 + rs175 + rs179 + rs180 + rs181 + TotalRutterA + (fs030 | fs004), data = binlogtest2, family = binomial(link="logit"), control = glmerControl(optimizer = "bobyqa"), nAGQ = 0)

summary(ccfit2)

probabilities <- predict(ccfit2, type = "response")

predicted.classes <- ifelse(probabilities > 0.5, "pos", "neg")

mydata <- binlogtest2 %>%

dplyr::select\_if(is.numeric)

predictors <- colnames(mydata)

mydata <- mydata %>%

mutate(logit = log(probabilities/(1-probabilities))) %>%

gather(key = "predictors", value = "predictor.value", -logit)

ggplot(mydata, aes(logit, predictor.value)) + geom\_point(size = 0.5, alpha = 0.5) + geom\_smooth(method = "loess") + theme\_bw() + facet\_wrap(~predictors, scales = "free\_y")

**##variance inflation factor**

check.collin(ccfit2)

check.collin(MLM1a)

**##missing data analysis - frequency, pattern, and MCAR test**

install.packages("naniar")

library(naniar)

ggplot2::aes(x = binlogtest2$q1rg91, y = binlogtest2$f\_rgsc91)

miss\_case\_summary(binlogtest2)

miss\_var\_summary(binlogtest2)

mcarbinlogtest <- mcar\_test(data = binlogtest2)

mcarbinlogtest

na.test(binlogtest2, digits = 2, p.digits = 3, as.na = NULL, check = TRUE, output = TRUE)

missing\_pattern(binlogtest2)

**##check if missing data is associated with observed data**

**#recode variables with missing data as 0 or 1 and conduct ANOVA with non-missing variables**

**##conduct regression on observed and missing data to assess if observed variables are associated with missing data points, this was done in SPSS**

**##Multiple imputation of missing data using MCMC method**

binlogtestIMP <- mice(data = binlogtest2, m = 20, method = "rf", maxit = 100, seed = 123456, remove.collinear = FALSE)

**##summarise imputation**

summary(binlogtestIMP)

**##view imputations for each variable**

binlogtestIMP$imp$fs071

binlogtestIMP$imp$peerstat

binlogtestIMP$imp$q1073

binlogtestIMP$imp$EdAtt

binlogtestIMP$imp$EdAtt2

binlogtestIMP$imp$f\_rgsc91

binlogtestIMP$imp$q1rg91

**##extract each imputed dataset**

finalIMP1 <- complete(binlogtestIMP, 1)

finalIMP2 <- complete(binlogtestIMP, 2)

finalIMP3 <- complete(binlogtestIMP, 3)

finalIMP4 <- complete(binlogtestIMP, 4)

finalIMP5 <- complete(binlogtestIMP, 5)

finalIMP6 <- complete(binlogtestIMP, 6)

finalIMP7 <- complete(binlogtestIMP, 7)

finalIMP8 <- complete(binlogtestIMP, 8)

finalIMP9 <- complete(binlogtestIMP, 9)

finalIMP10 <- complete(binlogtestIMP, 10)

finalIMP11 <- complete(binlogtestIMP, 11)

finalIMP12 <- complete(binlogtestIMP, 12)

finalIMP13 <- complete(binlogtestIMP, 13)

finalIMP14 <- complete(binlogtestIMP, 14)

finalIMP15 <- complete(binlogtestIMP, 15)

finalIMP16 <- complete(binlogtestIMP, 16)

finalIMP17 <- complete(binlogtestIMP, 17)

finalIMP18 <- complete(binlogtestIMP, 18)

finalIMP19 <- complete(binlogtestIMP, 19)

finalIMP20 <- complete(binlogtestIMP, 20)

**##check no missing data in each imputed dataset**

sapply(finalIMP1, function(x) sum(is.na(x)))

sapply(finalIMP2, function(x) sum(is.na(x)))

sapply(finalIMP3, function(x) sum(is.na(x)))

sapply(finalIMP4, function(x) sum(is.na(x)))

sapply(finalIMP5, function(x) sum(is.na(x)))

sapply(finalIMP6, function(x) sum(is.na(x)))

sapply(finalIMP7, function(x) sum(is.na(x)))

sapply(finalIMP8, function(x) sum(is.na(x)))

sapply(finalIMP9, function(x) sum(is.na(x)))

sapply(finalIMP10, function(x) sum(is.na(x)))

sapply(finalIMP11, function(x) sum(is.na(x)))

sapply(finalIMP12, function(x) sum(is.na(x)))

sapply(finalIMP13, function(x) sum(is.na(x)))

sapply(finalIMP14, function(x) sum(is.na(x)))

sapply(finalIMP15, function(x) sum(is.na(x)))

sapply(finalIMP16, function(x) sum(is.na(x)))

sapply(finalIMP17, function(x) sum(is.na(x)))

sapply(finalIMP18, function(x) sum(is.na(x)))

sapply(finalIMP19, function(x) sum(is.na(x)))

sapply(finalIMP20, function(x) sum(is.na(x)))

**##list imp data frame**

write.datlist(datlist = binlogtestIMP, name = "implist", type = "Rdata")

**##create a list of imputed datasets for mi.t.test**

dataset1 <- complete(binlogtestIMP, 1)

dataset2 <- complete(binlogtestIMP, 2)

dataset3 <- complete(binlogtestIMP, 3)

dataset4 <- complete(binlogtestIMP, 4)

dataset5 <- complete(binlogtestIMP, 5)

dataset6 <- complete(binlogtestIMP, 6)

dataset7 <- complete(binlogtestIMP, 7)

dataset8 <- complete(binlogtestIMP, 8)

dataset9 <- complete(binlogtestIMP, 9)

dataset10 <- complete(binlogtestIMP, 10)

dataset11 <- complete(binlogtestIMP, 11)

dataset12 <- complete(binlogtestIMP, 12)

dataset13 <- complete(binlogtestIMP, 13)

dataset14 <- complete(binlogtestIMP, 14)

dataset15 <- complete(binlogtestIMP, 15)

dataset16 <- complete(binlogtestIMP, 16)

dataset17 <- complete(binlogtestIMP, 17)

dataset18 <- complete(binlogtestIMP, 18)

dataset19 <- complete(binlogtestIMP, 19)

dataset20 <- complete(binlogtestIMP, 20)

**##define epilepsy as factor variable in each imp dataset**

dataset1$fs030 <- factor(dataset1$fs030)

dataset2$fs030 <- factor(dataset2$fs030)

dataset3$fs030 <- factor(dataset3$fs030)

dataset4$fs030 <- factor(dataset4$fs030)

dataset5$fs030 <- factor(dataset5$fs030)

dataset6$fs030 <- factor(dataset6$fs030)

dataset7$fs030 <- factor(dataset7$fs030)

dataset8$fs030 <- factor(dataset8$fs030)

dataset9$fs030 <- factor(dataset9$fs030)

dataset10$fs030 <- factor(dataset10$fs030)

dataset11$fs030 <- factor(dataset11$fs030)

dataset12$fs030 <- factor(dataset12$fs030)

dataset13$fs030 <- factor(dataset13$fs030)

dataset14$fs030 <- factor(dataset14$fs030)

dataset15$fs030 <- factor(dataset15$fs030)

dataset16$fs030 <- factor(dataset16$fs030)

dataset17$fs030 <- factor(dataset17$fs030)

dataset18$fs030 <- factor(dataset18$fs030)

dataset19$fs030 <- factor(dataset19$fs030)

dataset20$fs030 <- factor(dataset20$fs030)

**##assign imp datasets to list object dataset.imp**

dataset.imp <-list(dataset1, dataset2, dataset3, dataset4, dataset5, dataset6, dataset7, dataset8, dataset9, dataset10, dataset11, dataset12, dataset13, dataset14, dataset15, dataset16, dataset17, dataset18, dataset19, dataset20)

**##check convergence of imputed data using graphs**

densityplot(binlogtestIMP, ~ q1073)

densityplot(binlogtestIMP, ~ fs071)

densityplot(binlogtestIMP, ~ peerstat)

densityplot(binlogtestIMP, ~ EdAtt)

densityplot(binlogtestIMP, ~ f\_rgsc91)

densityplot(binlogtestIMP, ~ q1rg91)

bwplot(binlogtestIMP, q1073 ~ .imp)

bwplot(binlogtestIMP, fs071 ~ .imp)

bwplot(binlogtestIMP, peerstat ~ .imp)

bwplot(binlogtestIMP, EdAtt ~ .imp)

bwplot(binlogtestIMP, f\_rgsc91 ~ .imp)

bwplot(binlogtestIMP, q1rg91 ~ .imp)

xyplot(binlogtestIMP, q1073 ~ .imp)

xyplot(binlogtestIMP, fs071 ~ .imp)

xyplot(binlogtestIMP, peerstat ~ .imp)

xyplot(binlogtestIMP, EdAtt ~ .imp)

xyplot(binlogtestIMP, f\_rgsc91 ~ .imp)

xyplot(binlogtestIMP, q1rg91 ~ .imp)

**##convert imputed dataset to long format and label foreign**

complete(binlogtestIMP)

binlogtestIMP\_long <- complete(binlogtestIMP, action = "long", include = TRUE)

labelled::foreign\_to\_labelled(binlogtestIMP\_long)

**##save imputed data set**

data1 <- complete(binlogtestIMP)

save(binlogtestIMP, file = "data.Rdata")

class(data1)

getdata(data1)

summary(data1)

length(data1) -1

summary(data1[[1]])

data2 <- getdata(binlogtestIMP[[1]])

class(data2)

names(data2)

data2$impdata

**##get pooled means and standard deviations of cont vars**

pool\_mean1 <- with(binlogtestIMP\_long, by(binlogtestIMP\_long, .imp, function(x) c(mean(x$rs175), sd(x$rs175))))

pool\_mean1

Reduce("+",pool\_mean1)/length(pool\_mean1)

pool\_mean2 <- with(binlogtestIMP\_long, by(binlogtestIMP\_long, .imp, function(x) c(mean(x$rs179), sd(x$rs179))))

pool\_mean2

Reduce("+",pool\_mean2)/length(pool\_mean2)

pool\_mean3 <- with(binlogtestIMP\_long, by(binlogtestIMP\_long, .imp, function(x) c(mean(x$rs180), sd(x$rs180))))

pool\_mean3

Reduce("+",pool\_mean3)/length(pool\_mean3)

pool\_mean4 <- with(binlogtestIMP\_long, by(binlogtestIMP\_long, .imp, function(x) c(mean(x$rs181), sd(x$rs181))))

pool\_mean4

Reduce("+",pool\_mean4)/length(pool\_mean4)

pool\_mean5 <- with(binlogtestIMP\_long, by(binlogtestIMP\_long, .imp, function(x) c(mean(x$TotalRutterA), sd(x$TotalRutterA))))

pool\_mean5

Reduce("+",pool\_mean5)/length(pool\_mean5)

**##conduct a chi-square for imputed datasets 1, 5, 10, 20**

**##chisquare**

sextab1 <- table(dataset1$fs030, dataset1$fs004)

chisq.test(dataset1$fs030, dataset1$fs004)

oddsratio(sextab1,method = "wald")

sextab5 <- table(dataset5$fs030, dataset5$fs004)

chisq.test(dataset5$fs030, dataset1$fs004)

oddsratio(sextab5,method = "wald")

sextab10 <- table(dataset10$fs030, dataset10$fs004)

chisq.test(dataset10$fs030, dataset10$fs004)

oddsratio(sextab10,method = "wald")

sextab20 <- table(dataset20$fs030, dataset20$fs004)

chisq.test(dataset20$fs030, dataset20$fs004)

odds\_ratio(sextab20,method = "wald")

fs071tab1 <- table(dataset1$fs030, dataset1$fs071)

chisq.test(dataset1$fs030, dataset1$fs071)

oddsratio(fs071tab1,method = "wald")

fs071tab5 <- table(dataset5$fs030, dataset5$fs071)

chisq.test(dataset5$fs030, dataset5$fs071)

oddsratio(fs071tab5,method = "wald")

fs071tab10 <- table(dataset10$fs030, dataset10$fs071)

chisq.test(dataset10$fs030, dataset10$fs071)

oddsratio(fs071tab10,method = "wald")

fs071tab20 <- table(dataset20$fs030, dataset20$fs071)

chisq.test(dataset20$fs030, dataset20$fs071)

oddsratio(fs071tab20,method = "wald")

peertab1 <- table(dataset1$fs030, dataset1$peerstat)

chisq.test(dataset1$fs030, dataset1$peerstat)

oddsratio(peertab1,method = "wald")

peertab5 <- table(dataset5$fs030, dataset5$peerstat)

chisq.test(dataset5$fs030, dataset5$peerstat)

oddsratio(peertab5,method = "wald")

peertab10 <- table(dataset10$fs030, dataset10$peerstat)

chisq.test(dataset10$fs030, dataset10$peerstat)

oddsratio(peertab10,method = "wald")

peertab20 <- table(dataset20$fs030, dataset20$peerstat)

chisq.test(dataset20$fs030, dataset20$peerstat)

oddsratio(peertab20,method = "wald")

q1073tab1 <- table(dataset1$fs030, dataset1$q1073)

chisq.test(dataset1$fs030, dataset1$q1073)

oddsratio(q1073tab1, method = "wald")

q1073tab5 <- table(dataset5$fs030, dataset5$q1073)

chisq.test(dataset5$fs030, dataset5$q1073)

oddsratio(q1073tab5, method = "wald")

q1073tab10 <- table(dataset10$fs030, dataset10$q1073)

chisq.test(dataset10$fs030, dataset10$q1073)

oddsratio(q1073tab10, method = "wald")

q1073tab20 <- table(dataset20$fs030, dataset20$q1073)

chisq.test(dataset20$fs030, dataset20$q1073)

oddsratio(q1073tab20, method = "wald")

edtab1 <- table(dataset1$fs030, dataset1$EdAtt)

chisq.test(dataset1$fs030, dataset1$EdAtt)

oddsratio(edtab1,method = "wald")

edtab5 <- table(dataset5$fs030, dataset5$EdAtt)

chisq.test(dataset5$fs030, dataset5$EdAtt)

oddsratio(edtab5,method = "wald")

edtab10 <- table(dataset10$fs030, dataset10$EdAtt)

chisq.test(dataset10$fs030, dataset10$EdAtt)

oddsratio(edtab10,method = "wald")

edtab20 <- table(dataset20$fs030, dataset20$EdAtt)

chisq.test(dataset20$fs030, dataset20$EdAtt)

oddsratio(edtab20,method = "wald")

fsestab1 <- table(dataset1$fs030, dataset1$f\_rgsc91)

chisq.test(dataset1$fs030, dataset1$f\_rgsc91)

oddsratio(fsestab1,method = "wald")

fsestab5 <- table(dataset5$fs030, dataset5$f\_rgsc91)

chisq.test(dataset5$fs030, dataset5$f\_rgsc91)

oddsratio(fsestab5,method = "wald")

fsestab10 <- table(dataset10$fs030, dataset10$f\_rgsc91)

chisq.test(dataset10$fs030, dataset10$f\_rgsc91)

oddsratio(fsestab10,method = "wald")

fsestab20 <- table(dataset20$fs030, dataset20$f\_rgsc91)

chisq.test(dataset20$fs030, dataset20$f\_rgsc91)

oddsratio(fsestab20,method = "wald")

asestab1 <- table(dataset1$fs030, dataset1$q1rg91)

chisq.test(dataset1$fs030, dataset1$q1rg91)

oddsratio(asestab1,method = "wald")

asestab5 <- table(dataset5$fs030, dataset5$q1rg91)

chisq.test(dataset5$fs030, dataset5$q1rg91)

oddsratio(asestab5,method = "wald")

asestab10 <- table(dataset10$fs030, dataset10$q1rg91)

chisq.test(dataset110$fs030, dataset10$q1rg91)

oddsratio(asestab10,method = "wald")

asestab20 <- table(dataset20$fs030, dataset20$q1rg91)

chisq.test(dataset20$fs030, dataset20$q1rg91)

oddsratio(asestab20,method = "wald")

**##conduct t-test estimate for imputed data and pool**

mi.t.test(dataset.imp, x ="TotalRutterA", y = "fs030")

fit.t.test1 <- with(data=binlogtestIMP, exp=lm(TotalRutterA ~ fs030))

t.test.estimates1 <- pool(fit.t.test1)

summary(t.test.estimates1)

mi.t.test(dataset.imp, x ="rs175", y = "fs030")

fit.t.test2 <- with(data=binlogtestIMP, exp=lm(rs175 ~ fs030))

t.test.estimates2 <- pool(fit.t.test2)

summary(t.test.estimates2)

cohens\_d(t.test.estimates2)

mi.t.test(dataset.imp, x ="rs179", y = "fs030")

fit.t.test3 <- with(data=binlogtestIMP, exp=lm(rs179 ~ fs030))

t.test.estimates3 <- pool(fit.t.test3)

summary(t.test.estimates3)

mi.t.test(dataset.imp, x ="rs180", y = "fs030")

fit.t.test4 <- with(data=binlogtestIMP, exp=lm(rs180 ~ fs030))

t.test.estimates4 <- pool(fit.t.test4)

summary(t.test.estimates4)

mi.t.test(dataset.imp, x ="rs181", y = "fs030")

fit.t.test5 <- with(data=binlogtestIMP, exp=lm(rs181 ~ fs030))

t.test.estimates5 <- pool(fit.t.test5)

summary(t.test.estimates5)

**##conduct chisquare test on categorical imputed data and pool**

library(miceadds)

chi1 <- with(binlogtestIMP, chisq.test(fs030, fs004, correct=FALSE))

chi1

lapply(chi1$analyses, '[', 'statistic')

dk1 <- c(0.3204, 0.3204, 0.3204, 0.3204, 0.3204, 0.3204, 0.3204, 0.3204, 0.3204, 0.3204, 0.3204, 0.3204, 0.3204, 0.3204, 0.3204, 0.3204, 0.3204, 0.3204, 0.3204, 0.3204)

dk1.combine <- micombine.chisquare(dk=dk1, df=1, display=TRUE, version=1)

ep\_sex <- with( data = binlogtestIMP, glm(fs004 ~ fs030, family = binomial(link="logit")))

summary(ep\_sex)

poolepsex <- summary(pool(ep\_sex), conf.int = TRUE, exponentiate = TRUE)

poolepsex

chi2 <- with(binlogtestIMP, chisq.test(fs030, fs071, correct=FALSE))

chi2

lapply(chi2$analyses, '[', 'statistic')

dk2 <- c(0.002456567, 0.05595806, 0.009620191, 0.04696141, 0.09379976, 0.02479965, 0.002456567, 0.009620191, 0.1321195, 0.004762252, 0.1252993, 0.03877383, 0.4732543, 0.03498137, 0.04696141, 0.03138873, 0.0427669, 0.2332652, 0.02799509, 0.007815251)

dk2.comb <- micombine.chisquare(dk=dk2, df=1, display=TRUE, version=1)

ep\_ph <- with( data = binlogtestIMP, glm(fs071 ~ fs030, family = binomial(link="logit")))

summary(ep\_ph)

poolepph <- summary(pool(ep\_ph), conf.int = TRUE, exponentiate = TRUE)

poolepph

chi3 <- with(binlogtestIMP, chisq.test(fs030, peerstat, correct = FALSE))

chi3

lapply(chi3$analyses, '[', 'statistic')

dk3 <- c(1.107727, 0.6884642, 1.574886, 0.2810818, 3.052043, 2.389598, 0.07257418, 0.1320353, 0.2711239, 0.2423856, 1.087317, 9.113567e-05, 0.5812656, 0.6108876, 0.4280756, 0.93161, 0.01443672, 2.483233, 0.4577508, 1.726669)

dk3.comb <- micombine.chisquare(dk=dk3, df=1, display=TRUE, version=1)

ep\_peer <- with( data = binlogtestIMP, glm(peerstat ~ fs030, family = binomial(link="logit")))

summary(ep\_peer)

pooleppeer <- summary(pool(ep\_peer), conf.int = TRUE, exponentiate = TRUE)

pooleppeer

chi4 <- with(binlogtestIMP, chisq.test(fs030, q1073, correct = FALSE))

lapply(chi4$analyses, '[', 'statistic')

dk4 <- c(0.2966763 , 2.814145, 2.298081, 0.8424846, 0.5044397, 1.197839, 1.624591, 2.019278, 2.107242, 1.264266, 2.545135, 2.848924, 1.574886, 0.3750988, 4.706342, 1.525538, 3.580836, 5.533642, 3.052043, 0.823231)

dk4.comb <- micombine.chisquare(dk=dk4, df=1, display=TRUE, version=1)

ep\_alc <- with( data = binlogtestIMP, glm(q1073 ~ fs030, family = binomial(link="logit")))

summary(ep\_alc)

poolepalc <- summary(pool(ep\_alc), conf.int = TRUE, exponentiate = TRUE)

poolepalc

chi5 <- with(binlogtestIMP, chisq.test(fs030, EdAtt, correct = FALSE))

lapply(chi5$analyses, '[', 'statistic')

dk5 <- c(4.941577, 8.175107, 5.311482, 8.298241, 6.763507, 8.392935, 4.438608, 7.152103, 6.732094, 11.16612, 4.567845, 8.257224, 4.95824, 8.635976, 6.885923, 2.409907, 3.589219, 5.196242, 6.514291, 5.282639)

dk5.comb <- micombine.chisquare(dk=dk5, df=2, display=TRUE, version=1)

ep\_edatt <- with( data = binlogtestIMP, glm(EdAtt ~ fs030, family = binomial(link="logit")))

summary(ep\_edatt)

poolepedatt <- summary(pool(ep\_edatt), conf.int = TRUE, exponentiate = TRUE)

poolepedatt

chi6 <- with(binlogtestIMP, chisq.test(fs030, f\_rgsc91, correct = FALSE))

lapply(chi6$analyses, '[', 'statistic')

dk6 <- c(0.6469898, 3.431766, 6.941262, 3.082302, 4.851368, 0.8779701, 1.556446, 4.266635, 9.09248, 15.04168, 1.347971, 0.6928585, 5.649209, 1.271605, 2.721433, 0.07798749, 5.550187, 4.432933, 0.9359989, 2.084497)

dk6.comb <- micombine.chisquare(dk=dk6, df=4, display=TRUE, version=1)

ep\_SESch <- with( data = binlogtestIMP, glm(f\_rgsc91 ~ fs030, family = binomial(link="logit")))

summary(ep\_SESch)

poolepSESch <- summary(pool(ep\_SESch), conf.int = TRUE, exponentiate = TRUE)

poolepSESch

chi7 <- with(binlogtestIMP, chisq.test(fs030, q1rg91, correct = FALSE))

lapply(chi7$analyses, '[', 'statistic')

dk7 <- c(4.537438, 2.310056, 0.187187, 2.109506, 3.228915, 0.001204907, 1.46251, 8.963126, 1.303466, 3.863768, 3.938924, 2.868496, 1.608764, 2.016245, 0.689147, 0.1060457, 1.840575, 1.238061, 0.6057707, 8.357385)

dk7.comb <- micombine.chisquare(dk=dk7, df=4, display=TRUE, version=1)

ep\_SESad <- with( data = binlogtestIMP, glm(q1rg91 ~ fs030, family = binomial(link="logit")))

summary(ep\_SESad)

poolepSESad <- summary(pool(ep\_SESad), conf.int = TRUE, exponentiate = TRUE)

poolepSESad

**##install packages and conduct binariy logistic regression**

**##assumption 2, independence of observations - MLM required for clusted data**

install.packages("lme4")

library(lme4)

install.packages("car")

install.packages("ggplot2")

install.packages("nlme")

install.packages("reshape")

library(car)

library(ggplot2)

library(nlme)

library(reshape)

install.packages("MASS")

library(MASS)

install.packages("stats")

library(stats)

**##conduct Binary logistic regression on imputed datasets 1, 5, 10, 20 to check if multicollinearity a potential problem**

adfit1 <- glm(q1rg91 ~ fs030 + fs071 + peerstat + q1073 + EdAtt + f\_rgsc91 + rs175 + rs179 + rs180 + rs181 + TotalRutterA + fs004, family = binomial(link="logit"), data = dataset1)

summary(adfit1)

AIC(logLik(adfit1))

BIC(logLik(adfit1))

PseudoR2(adfit1)

ctable <- exp(coef(summary(adfit1)))

print(ctable)

check.collin(adfit1)

parameters::model\_parameters(adfit1, exp = TRUE)

**##check residuals of 1st imputed dataset**

dataset1$residuals <- resid(adfit1)

dataset1$standardized.residuals <- rstandard(adfit1)

dataset1$studentized.residuals <- rstudent(adfit1)

dataset1$cooks.distance <- cooks.distance(adfit1)

dataset1$dfbeta <- dfbeta(adfit1)

dataset1$dffit <- dffits(adfit1)

dataset1$leverage <- hatvalues(adfit1)

dataset1$covariance.ratios <- covratio(adfit1)

dataset1

dataset1$standardized.residuals > 2 | dataset1$standardized.residuals < -2

round(dataset1, digits = 3)

dataset1$large.residual <- dataset1$standardized.residuals > 2 | dataset1$standardized.residuals < -2

sum(dataset1$large.residual)

dataset1[dataset1$large.residual, c("fs030", "fs071", "peerstat", "q1073", "EdAtt", "f\_rgsc91", "rs175", "rs179", "rs180", "rs181", "TotalRutterA", "fs004", "standardized.residuals")]

dataset1[dataset1$large.residual, c("cooks.distance", "leverage", "covariance.ratios", "standardized.residuals", "studentized.residuals", "dfbeta")]

**##Assess linearity of the logit for cont.var**

dataset1$logrs175 <- log(dataset1$rs175)\*dataset1$rs175

dataset1$logrs179 <- log(dataset1$rs179)\*dataset1$rs179

dataset1$logrs180 <- log(dataset1$rs180)\*dataset1$rs180

dataset1$logrs181 <- log(dataset1$rs181)\*dataset1$rs181

dataset1$logTotalRutterA <- log(dataset1$TotalRutterA)\*dataset1$TotalRutterA

penaltyTest.1 <- glm(q1rg91 ~ fs030 + fs071 + peerstat + q1073 + EdAtt + f\_rgsc91 + rs175 + rs179 + rs180 + rs181 + TotalRutterA + fs004 + logrs175 + logrs179 + logrs180 + logrs181 + logTotalRutterA, family = binomial(link="logit"), data = dataset1)

summary(penaltyTest.1)

**##Conduct an adjusted binary logistic regression on imputed dataset 5**

adfit5 <- glm(q1rg91 ~ fs030 + fs071 + peerstat + q1073 + EdAtt + f\_rgsc91 + rs175 + rs179 + rs180 + rs181 + TotalRutterA + fs004, family = binomial(link="logit"), data = dataset5)

summary(adfit5)

AIC(logLik(adfit5))

BIC(logLik(adfit5))

PseudoR2(adfit5)

ctable <- exp(coef(summary(adfit5)))

print(ctable)

check.collin(adfit5)

parameters::model\_parameters(adfit5, exp = TRUE)

**##check residuals of 5th imputed dataset**

dataset5$residuals <- resid(adfit5)

dataset5$standardized.residuals <- rstandard(adfit5)

dataset5$studentized.residuals <- rstudent(adfit5)

dataset5$cooks.distance <- cooks.distance(adfit5)

dataset5$dfbeta <- dfbeta(adfit5)

dataset5$dffit <- dffits(adfit5)

dataset5$leverage <- hatvalues(adfit5)

dataset5$covariance.ratios <- covratio(adfit5)

dataset5

dataset5$standardized.residuals > 2 | dataset5$standardized.residuals < -2

round(dataset5, digits = 3)

dataset5$large.residual <- dataset5$standardized.residuals > 2 | dataset5$standardized.residuals < -2

sum(dataset5$large.residual)

dataset5[dataset5$large.residual, c("fs030", "fs071", "peerstat", "q1073", "EdAtt", "f\_rgsc91", "rs175", "rs179", "rs180", "rs181", "TotalRutterA", "fs004", "standardized.residuals")]

dataset5[dataset5$large.residual, c("cooks.distance", "leverage", "covariance.ratios", "standardized.residuals", "studentized.residuals", "dfbeta")]

**##Assess linearity of the logit for cont.var**

dataset5$logrs175 <- log(dataset5$rs175)\*dataset5$rs175

dataset5$logrs179 <- log(dataset5$rs179)\*dataset5$rs179

dataset5$logrs180 <- log(dataset5$rs180)\*dataset5$rs180

dataset5$logrs181 <- log(dataset5$rs181)\*dataset5$rs181

dataset5$logTotalRutterA <- log(dataset5$TotalRutterA)\*dataset5$TotalRutterA

penaltyTest.5 <- glm(q1rg91 ~ fs030 + fs071 + peerstat + q1073 + EdAtt + f\_rgsc91 + rs175 + rs179 + rs180 + rs181 + TotalRutterA + fs004 + logrs175 + logrs179 + logrs180 + logrs181 + logTotalRutterA, family = binomial(link="logit"), data = dataset5)

summary(penaltyTest.5)

**##Conduct a binary regression on dataset 10**

adfit10 <- glm(q1rg91 ~ fs030 + fs071 + peerstat + q1073 + EdAtt + f\_rgsc91 + rs175 + rs179 + rs180 + rs181 + TotalRutterA + fs004, family = binomial(link="logit"), data = dataset10)

summary(adfit10)

AIC(logLik(adfit10))

BIC(logLik(adfit10))

PseudoR2(adfit10)

ctable <- exp(coef(summary(adfit10)))

print(ctable)

check.collin(adfit10)

parameters::model\_parameters(adfit10, exp = TRUE)

**##check residuals of 10th imputed dataset**

dataset10$residuals <- resid(adfit10)

dataset10$standardized.residuals <- rstandard(adfit10)

dataset10$studentized.residuals <- rstudent(adfit10)

dataset10$cooks.distance <- cooks.distance(adfit10)

dataset10$dfbeta <- dfbeta(adfit10)

dataset10$dffit <- dffits(adfit10)

dataset10$leverage <- hatvalues(adfit10)

dataset10$covariance.ratios <- covratio(adfit10)

dataset10

dataset10$standardized.residuals > 2 | dataset10$standardized.residuals < -2

round(dataset10, digits = 3)

dataset10$large.residual <- dataset10$standardized.residuals > 2 | dataset10$standardized.residuals < -2

sum(dataset10$large.residual)

dataset10[dataset10$large.residual, c("fs030", "fs071", "peerstat", "q1073", "EdAtt", "f\_rgsc91", "rs175", "rs179", "rs180", "rs181", "TotalRutterA", "fs004", "standardized.residuals")]

dataset10[dataset10$large.residual, c("cooks.distance", "leverage", "covariance.ratios", "standardized.residuals", "studentized.residuals", "dfbeta")]

**##Assess linearity of the logit for cont.var**

dataset10$logrs175 <- log(dataset10$rs175)\*dataset10$rs175

dataset10$logrs179 <- log(dataset10$rs179)\*dataset10$rs179

dataset10$logrs180 <- log(dataset10$rs180)\*dataset10$rs180

dataset10$logrs181 <- log(dataset10$rs181)\*dataset10$rs181

dataset10$logTotalRutterA <- log(dataset10$TotalRutterA)\*dataset10$TotalRutterA

penaltyTest.10 <- glm(q1rg91 ~ fs030 + fs071 + peerstat + q1073 + EdAtt + f\_rgsc91 + rs175 + rs179 + rs180 + rs181 + TotalRutterA + fs004 + logrs175 + logrs179 + logrs180 + logrs181 + logTotalRutterA, family = binomial(link="logit"), data = dataset10)

summary(penaltyTest.10)

**##Conduct binary regression on dataset 20**

adfit20 <- glm(q1rg91 ~ fs030 + fs071 + peerstat + q1073 + EdAtt + f\_rgsc91 + rs175 + rs179 + rs180 + rs181 + TotalRutterA + fs004, family = binomial(link="logit"), data = dataset20)

summary(adfit20)

AIC(logLik(adfit20))

BIC(logLik(adfit20))

PseudoR2(adfit20)

ctable <- exp(coef(summary(adfit20)))

print(ctable)

check.collin(adfit20)

parameters::model\_parameters(adfit20, exp = TRUE)

**##check residuals of 20th imputed dataset**

dataset20$residuals <- resid(adfit20)

dataset20$standardized.residuals <- rstandard(adfit20)

dataset20$studentized.residuals <- rstudent(adfit20)

dataset20$cooks.distance <- cooks.distance(adfit20)

dataset20$dfbeta <- dfbeta(adfit20)

dataset20$dffit <- dffits(adfit20)

dataset20$leverage <- hatvalues(adfit20)

dataset20$covariance.ratios <- covratio(adfit20)

dataset20

dataset20$standardized.residuals > 2 | dataset20$standardized.residuals < -2

round(dataset20, digits = 3)

dataset20$large.residual <- dataset20$standardized.residuals > 2 | dataset20$standardized.residuals < -2

sum(dataset20$large.residual)

dataset20[dataset20$large.residual, c("fs030", "fs071", "peerstat", "q1073", "EdAtt", "f\_rgsc91", "rs175", "rs179", "rs180", "rs181", "TotalRutterA", "fs004", "standardized.residuals")]

dataset20[dataset20$large.residual, c("cooks.distance", "leverage", "covariance.ratios", "standardized.residuals", "studentized.residuals", "dfbeta")]

**##Assess linearity of the logit for cont.var**

dataset20$logrs175 <- log(dataset20$rs175)\*dataset20$rs175

dataset20$logrs179 <- log(dataset20$rs179)\*dataset20$rs179

dataset20$logrs180 <- log(dataset20$rs180)\*dataset20$rs180

dataset20$logrs181 <- log(dataset20$rs181)\*dataset20$rs181

dataset20$logTotalRutterA <- log(dataset20$TotalRutterA)\*dataset20$TotalRutterA

penaltyTest.20 <- glm(q1rg91 ~ fs030 + fs071 + peerstat + q1073 + EdAtt + f\_rgsc91 + rs175 + rs179 + rs180 + rs181 + TotalRutterA + fs004 + logrs175 + logrs179 + logrs180 + logrs181 + logTotalRutterA, family = binomial(link="logit"), data = dataset20)

summary(penaltyTest.20)

**##conduct binariy logistic regression on pooled imputed data**

unfit <- with(data = binlogtestIMP, glm(q1rg91 ~ fs030, family = binomial(link="logit")))

summary(unfit)

poolunfit <- summary(pool(unfit), conf.int = TRUE, exponentiate = TRUE)

poolunfit

adfit <- with(data = binlogtestIMP, exp = stats::glm(q1rg91 ~ fs030 + fs071 + peerstat + q1073 + EdAtt + f\_rgsc91 + rs175 + rs179 + rs180 + rs181 + TotalRutterA + fs004, family = binomial(link="logit")))

pooladfit <- summary(pool(adfit), conf.int = TRUE, exponentiate = TRUE)

pooladfit

summary(pooladfit)

**##compare model parameters between CC and imp 1, 5, 10, 20**

parameters::model\_parameters(CCadfit, exp = TRUE)

parameters::model\_parameters(adfit1, exp = TRUE)

parameters::model\_parameters(adfit5, exp = TRUE)

parameters::model\_parameters(adfit10, exp = TRUE)

parameters::model\_parameters(adfit20, exp = TRUE)

**##MEAN AIC, BIC, PseudoR2 for 1, 5, 10, 20th IMPUTED estimates**

AIC1 <- AIC(logLik(adfit1))

AIC5 <- AIC(logLik(adfit5))

AIC10 <- AIC(logLik(adfit10))

AIC20 <- AIC(logLik(adfit20))

meanAIC <- (AIC1 + AIC5 + AIC10 + AIC20)/4

meanAIC

BIC1 <- BIC(logLik(adfit1))

BIC5 <- BIC(logLik(adfit5))

BIC10 <- BIC(logLik(adfit10))

BIC20 <- BIC(logLik(adfit20))

meanBIC <- (BIC1 + BIC5 + BIC10 + BIC20)/4

meanBIC

Pseudo1 <- PseudoR2(adfit1)

Pseudo5 <- PseudoR2(adfit5)

Pseudo10 <- PseudoR2(adfit10)

Pseudo20 <- PseudoR2(adfit20)

meanPseudo <- (Pseudo1 + Pseudo5 + Pseudo10 + Pseudo20)/4

meanPseudo

**##conduct binariy logistic regression with 2 levels, randome slope, random intercept on pooled imputed data**

fit1 <- with(data = binlogtestIMP, exp = lme4::glmer(q1rg91 ~ fs030 + fs071 + peerstat + q1073 + q1098\_total + f\_rgsc91 + rs175 + rs179 + rs180 + rs181 + TotalRutterA + (fs004 | fs004), family = binomial(link="logit"), control = glmerControl(optimizer = "bobyqa"), nAGQ = 0))

summary(fit1)

summary(pool(fit1), conf.int = TRUE, exponentiate = TRUE)

**##scatterplot**

library(dplyr)

binlogtestna <- na.omit(binlogtest)

ccfit2 <- glmer(q1rg91 ~ fs030 + fs071 + peerstat + q1073 + q1098\_total + f\_rgsc91 + rs175 + rs179 + rs180 + rs181 + TotalRutterA + (fs004 | fs004), data = binlogtest2, family = binomial(link="logit"), control = glmerControl(optimizer = "bobyqa"), nAGQ = 0)

summary(ccfit2)

probabilities <- predict(ccfit2, type = "response")

predicted.classes <- ifelse(probabilities > 0.5, "pos", "neg")

mydata <- binlogtest2 %>%

dplyr::select\_if(is.numeric)

predictors <- colnames(mydata)

mydata <- mydata %>%

mutate(logit = log(probabilities/(1-probabilities))) %>%

gather(key = "predictors", value = "predictor.value", -logit)

ggplot(mydata, aes(logit, predictor.value)) + geom\_point(size = 0.5, alpha = 0.5) + geom\_smooth(method = "loess") + theme\_bw() + facet\_wrap(~predictors, scales = "free\_y")

**##check correlation of variables (correlation test or vif)**

**##assumption 3 no collinearity between variables**

**##check linearity between continuous predictor variable and the log odds of the outcomes (box-tidwell test or graphs)**

**##assumption 4 continuous predictor variable linearly related to the log odds of the outcomes**

**##correlation**

library(car)

install.packages("sjstats")

library(sjstats)

cor.test(pooled, method = "spearman")

performance::icc(pooled)

as\_tibble(pooled)

**##ICC**

library(DescTools)

ICC(binlogTEST, type = c("all", conf.level = 0.95, na.rm = TRUE))

print(fit1, digits = 3)

install.packages("irr")

library("irr")

install.packages("psych")

library("psych")

ICC(binlogtest)

**##boxtidwell**

library(car)

boxTidwell(q1rg91 ~ rs175 + rs179 + rs180 + rs181, ~ as.factorfs030 + fs004 + fs071 + peerstat + q1073 + q1098\_total + f\_rgsc91, data=binlogtest)

**##scatterplot**

library(dplyr)

pooled2 <- na.omit(pooled)

fit2 <- with(data = pooled, exp = lme4::glmer(q1rg91 ~ fs030 + fs071 + peerstat + q1073 + q1098\_total + f\_rgsc91 + rs175 + rs179 + rs180 + rs181 + TotalRutterA + (fs030 | fs004), family = binomial(link="logit"), control = glmerControl(optimizer = "bobyqa"), nAGQ = 0))

summary(fit2)

probabilities <- predict(fit2, type = "response")

predicted.classes <- ifelse(probabilities > 0.5, "pos", "neg")

mydata <- binlogtestIMP2 %>%

dplyr::select\_if(is.numeric)

predictors <- colnames(mydata)

mydata <- mydata %>%

mutate(logit = log(probabilities/(1-probabilities))) %>%

gather(key = "predictors", value = "predictor.value", -logit)

ggplot(mydata, aes(logit, predictor.value)) + geom\_point(size = 0.5, alpha = 0.5) + geom\_smooth(method = "loess") + theme\_bw() + facet\_wrap(~predictors, scales = "free\_y")

**##variance inflation factor**

install.packages("car")

library(car)

?vif

vif(binlogtest)

install.packages("misty")

library(misty)

install.packages("usdm")

library(usdm)

install.packages("usdm")

check.collin(fit1)

**##assess model diagnsotics using pseudo rsquared and AIC BIC**

PseudoR2(fit1)

print(poolfit, correlation=TRUE)

**##compare MAR assumption to MNAR assumption if possible**

MNAR <- mice(data = aconfB, method = c("pmm"), ums = "-2+1\*rs179")

MNARrun <- complete(MNAR, action = "long", include = TRUE)

fit2 <- with(data = MNAR, exp = polr(q1rg91 ~ fs030 + fs004 + fs071 + rs175 + rs179 + rs180 + rs181 + peerstat + q1073 + q1098\_total + f\_rgsc91 + TotalRutterA, Hess = TRUE))

fit2

poolfit2 <- summary(pool(fit2), conf.int = TRUE, exponentiate = TRUE)

poolfit2

p <- pnorm(abs(ctable[, "t value"]), lower.tail = FALSE) \* 2

ctable <- cbind(ctable, "p value" = p)

ctable

p

exp(cbind(OR = coef(poolfit2), ci))

miss\_case\_summary(MNARrun)

miss\_var\_summary(MNARrun)

mcar\_test(MNAR)