R Notebook: SOC 5800 Team Project

Malcolm S. Townes, Jacob Eikenberry, Daniel Ferris, Nicholas Sokolis

(November 23, 2019)

## Introduction

This is an R Notebook for a study that examines how formerly incarcerated individuals who successfully reintegrated cope with barriers to re-entry. This study was conducted as a team project for the SOC 5800 Survey Design and Sampling course during the Fall 2019 semester at Saint Louis University.

## Project Set Up

The following code chunk enables the R Notebook to integrate seamlessly with the project organization format. This is normally included in the R Notebook to simplify file calls and enable file portability but it has been causing an error. To work around this problem, I’ve embedded the here() function where I enter a file path when necessary.

knitr::opts\_knit$set(root.dir = here::here())

## Load Dependencies

The following code chunk loads package dependencies required to perform the necessary tasks. Basic tasks include importing, reading, wrangling, and cleaning data; selecting a subset of the data; checking for unique observations; analyzing missing data; and performing various types of regression analyses.

library(tidyverse) # loads the basic R packages  
library(here) # enables file portability  
library(readr) # functions for reading data  
library(dplyr) # functions for data wrangling  
library(janitor) # functions for data cleaning  
library(naniar) # functions for analyzing missing data  
library(expss) # functions for calculating on values  
library(ggplot2) # functions for data visualizations  
library(boot) # functions for regression analysis  
library(generalhoslem) # Hosmer-Lemeshow test for binary and multinomial logistic models  
library(ordinal) # functions for regression models for ordinal data  
library(MASS) # functions for ordered logistic or probit regression  
library(broom) # functions for tidying ordinal logistic regression models  
library(gvlma) # functions for global validation of linear model assumptions  
library(lmtest) # functions for testing linear regression models  
library(ltm) # functions for latent trait models under Item Response Theory  
library(leaps) # functions for regression subset selection  
library(car) # companion to applied regression  
library(aod) # functions to analyze overdispersed data counts and proportions  
library(pscl) # contains function for pseudo R2 measures for logistic regression  
library(ResourceSelection) # contains function for Hosmer-Lemeshow goodness of fit test  
library(psy) # functions for various procedures used in psychometry  
library(psych) #Procedures for Psychological, Psychometric, and Personality Research

## Load Raw Data

The following code chunk imports the raw data from the csv file.

dataRaw <- read.csv(here("Data","DataRaw","SOC5800\_Data\_NumericValues\_Raw\_CSV.csv"),   
 sep = ",", header = TRUE, fill = TRUE, dec = ".")

## Rename Variables

The following code chunk performs several actions to clean the raw data. It first renames the variables with descriptive camel case names. It then removes unused variables.

dataRaw %>%  
 dplyr::rename(startDate = StartDate,  
 endDate = EndDate,  
 status = Status,  
 IPaddress = IPAddress,  
 progress = Progress,  
 sessionDuration = Duration..in.seconds.,  
 surveyCompleted = Finished,  
 surveyDate = RecordedDate,  
 responseID = ResponseId,  
 locationLat = LocationLatitude,  
 locationLong = LocationLongitude,  
 consent = Q43,  
 browserName = Q10\_Browser,  
 browserVersion = Q10\_Version,  
 opSyst = Q10\_Operating.System,  
 screenRes = Q10\_Resolution,  
 currentlyIncarcerated = Q1,  
 currentlyDetained = Q2,  
 residentialTreatment = Q3,  
 involCommitment = Q4,  
 gender = Q5,  
 genderSelfDescribe = Q5\_3\_TEXT,  
 transgender = Q7,  
 ethnicityRace = Q6,  
 ethnicityRaceOther = Q6\_10\_TEXT,  
 typeHometown = Q8,  
 religiousAffiliation = Q9,  
 ReligiousOther = Q9\_12\_TEXT,  
 ageNow = Q11,  
 ageRelease = Q13,  
 incarcerationYears = Q14.1\_1,  
 incarcerationMonths = Q14.2\_1,  
 educationLevels = Q15,  
 whenHighestEd = Q16,  
 relationshipStatus = Q17,  
 householdSize = Q18,  
 financialSupportInitial = Q19,  
 financialSupportGovt = Q19\_4\_TEXT,  
 financialSupportNonGovt = Q19\_5\_TEXT,  
 finanicalSupportOther = Q19\_6\_TEXT,  
 selfEmployment = Q20,  
 incomeInitial = Q21,  
 incomeLastYr = Q22,  
 financialStatus = Q23,  
 savings = Q24,  
 pssQ1 = Q41\_1,  
 pssQ2 = Q41\_2,  
 pssQ3 = Q41\_3,  
 pssQ4 = Q41\_4,  
 pssQ5 = Q41\_5,  
 pssQ6 = Q41\_6,  
 pssQ7 = Q41\_7,  
 pssQ8 = Q41\_8,  
 pssQ9 = Q41\_9,  
 pssQ10 = Q41\_10,  
 pssQ11 = Q41\_11,  
 pssQ12 = Q41\_12,  
 supportResidence = Q30\_1,  
 supportJob = Q30\_2,  
 supportAddiction = Q30\_3,  
 supportTransportation = Q30\_4,  
 supportFinancial = Q30\_5,  
 friendResidence = Q31\_1,  
 friendJob = Q31\_2,  
 friendAddiction = Q31\_3,  
 friendTransportation = Q31\_4,  
 friendFinancial = Q31\_5,  
 programsUsed = Q32,  
 importancePublicTrans = Q33\_1,  
 importanceHousing = Q33\_2,  
 importanceSNAP = Q33\_3,  
 importanceWIC = Q33\_4,  
 importanceTANF = Q33\_5,  
 importanceEITC = Q33\_6,  
 importanceJobTraining = Q33\_7,  
 importanceMedicare = Q33\_8,  
 importanceEducAid = Q33\_9,  
 aceQ1 = Q34\_1,  
 aceQ2 = Q34\_2,  
 aceQ3 = Q34\_3,  
 aceQ4 = Q34\_4,  
 aceQ5 = Q34\_5,  
 aceQ6 = Q34\_6,  
 aceQ7 = Q34\_7,  
 aceQ8 = Q34\_8,  
 aceQ9 = Q34\_9,  
 aceQ10 = Q34\_10,  
 helpDuring = Q35,  
 helpDuringType = Q33,  
 helpDuringOther = Q33\_8\_TEXT,  
 helpAfter = Q36,  
 helpAfterType = Q44,  
 helpAfterOther = Q44\_8\_TEXT,  
 gritQ1 = Q35\_1,  
 gritQ2 = Q35\_2,  
 gritQ3 = Q35\_3,  
 gritQ4 = Q35\_4,  
 gritQ5 = Q35\_5,  
 gritQ6 = Q35\_6,  
 gritQ7 = Q35\_7,  
 gritQ8 = Q35\_8,  
 stayInitial = Q36.1,  
 stayInitialOther = Q36\_6\_TEXT,  
 foodSecurityQ1 = Q37\_1,  
 foodSecurityQ2 = Q37\_2,  
 mostHelpful = Q38,  
 greatestObstacle = Q39,  
 helpKind = Q40,  
 justiceInteraction = Q41  
 ) -> dataRenamed  
  
dataRenamed %>%  
 dplyr::select (-c(status, progress, RecipientLastName, RecipientFirstName, RecipientEmail, ExternalReference)) -> dataAugmented

## Change Data Type

The following code chunk changes the data type for certain variables.

dataAugmented$pssQ1 <- as.numeric(dataAugmented$pssQ1)  
dataAugmented$pssQ2 <- as.numeric(dataAugmented$pssQ2)  
dataAugmented$pssQ3 <- as.numeric(dataAugmented$pssQ3)  
dataAugmented$pssQ4 <- as.numeric(dataAugmented$pssQ4)  
dataAugmented$pssQ5 <- as.numeric(dataAugmented$pssQ5)  
dataAugmented$pssQ6 <- as.numeric(dataAugmented$pssQ6)  
dataAugmented$pssQ7 <- as.numeric(dataAugmented$pssQ7)  
dataAugmented$pssQ8 <- as.numeric(dataAugmented$pssQ8)  
dataAugmented$pssQ9 <- as.numeric(dataAugmented$pssQ9)  
dataAugmented$pssQ10 <- as.numeric(dataAugmented$pssQ10)  
dataAugmented$pssQ11 <- as.numeric(dataAugmented$pssQ11)  
dataAugmented$pssQ12 <- as.numeric(dataAugmented$pssQ12)  
dataAugmented$aceQ1 <- as.numeric(dataAugmented$aceQ1)  
dataAugmented$aceQ2 <- as.numeric(dataAugmented$aceQ2)  
dataAugmented$aceQ3 <- as.numeric(dataAugmented$aceQ3)  
dataAugmented$aceQ4 <- as.numeric(dataAugmented$aceQ4)  
dataAugmented$aceQ5 <- as.numeric(dataAugmented$aceQ5)  
dataAugmented$aceQ6 <- as.numeric(dataAugmented$aceQ6)  
dataAugmented$aceQ7 <- as.numeric(dataAugmented$aceQ7)  
dataAugmented$aceQ8 <- as.numeric(dataAugmented$aceQ8)  
dataAugmented$aceQ9 <- as.numeric(dataAugmented$aceQ9)  
dataAugmented$aceQ10 <- as.numeric(dataAugmented$aceQ10)  
dataAugmented$gritQ1 <- as.numeric(dataAugmented$gritQ1)  
dataAugmented$gritQ2 <- as.numeric(dataAugmented$gritQ2)  
dataAugmented$gritQ3 <- as.numeric(dataAugmented$gritQ3)  
dataAugmented$gritQ4 <- as.numeric(dataAugmented$gritQ4)  
dataAugmented$gritQ5 <- as.numeric(dataAugmented$gritQ5)  
dataAugmented$gritQ6 <- as.numeric(dataAugmented$gritQ6)  
dataAugmented$gritQ7 <- as.numeric(dataAugmented$gritQ7)  
dataAugmented$gritQ8 <- as.numeric(dataAugmented$gritQ8)

## Adjust Value Assignments

The following code chunk adjusts the values assigned to fixed pre-determined response choices for select variables and recodes blanks with NA in the dataset.

dataAugmented$ageNow <- (as.numeric(dataAugmented$ageNow)+11)  
dataAugmented$ageRelease <- (as.numeric(dataAugmented$ageRelease)+11)  
dataAugmented[dataAugmented==""] <- NA

## Create Variables

The following code chunk creates additional variables needed for the study.

dataAugmented %>%  
 mutate (mspssScore = case\_when(  
 !is.na(pssQ1)|!is.na(pssQ2)|!is.na(pssQ3)|!is.na(pssQ4)|!is.na(pssQ5)|!is.na(pssQ6)|!is.na(pssQ7)|  
 !is.na(pssQ8)|!is.na(pssQ9)|!is.na(pssQ10)|!is.na(pssQ11)|!is.na(pssQ12) ~  
 (pssQ1+pssQ2+pssQ3+pssQ4+pssQ5+pssQ6+pssQ7+pssQ8+pssQ9+pssQ10+pssQ11+pssQ12)/12  
 ))%>%  
 mutate (aceScore = case\_when (  
 !is.na(aceQ1)|!is.na(aceQ2)|!is.na(aceQ3)|!is.na(aceQ4)|!is.na(aceQ5)|!is.na(aceQ6)|!is.na(aceQ7)|  
 !is.na(aceQ8)|!is.na(aceQ9)|!is.na(aceQ10) ~   
 (aceQ1+aceQ1+aceQ2+aceQ3+aceQ4+aceQ5+aceQ6+aceQ7+aceQ8+aceQ9+aceQ10)  
 )) %>%  
 mutate (gritQ1R = 6-gritQ1, gritQ3R = 6-gritQ3, gritQ5R = 6-gritQ5, gritQ6R = 6-gritQ6) %>%  
 mutate (gritScore = case\_when(  
 !is.na(gritQ1R)|!is.na(gritQ2)|!is.na(gritQ2)|!is.na(gritQ2)|!is.na(gritQ2)|  
 !is.na(gritQ2)|!is.na(gritQ2)|!is.na(gritQ2) ~   
 ((gritQ1R+gritQ2+gritQ3R+gritQ4+gritQ5R+gritQ6R+gritQ7+gritQ8)/8)  
 ))%>%  
 mutate (incarcerationTime = (incarcerationYears\*12)+incarcerationMonths) %>%  
 mutate (povertyRatio = if(as.numeric(incomeLastYr) <= 6) {  
 (8070+(as.numeric(incomeLastYr)\*4420))/(8070+(as.numeric(householdSize)\*4420))  
 } else {  
 if(as.numeric(incomeLastYr) == 7) {  
 (34591/(8070+(as.numeric(householdSize)\*4420)))  
 } else {  
 if(as.numeric(incomeLastYr) == 8) {  
 (50001/(8070+(as.numeric(householdSize)\*4420)))  
 } else {  
 if(as.numeric(incomeLastYr) == 9) {  
 (75001/(8070+(as.numeric(householdSize)\*4420)))  
 } else {  
 if(as.numeric(incomeLastYr) == 10) {  
 (100001/(8070+(as.numeric(householdSize\*4420))))  
 } else {  
 (150000/(8070+(as.numeric(householdSize)\*4420)))  
 }  
 }  
 }  
 }  
 }  
 ) %>%  
 mutate (programsUsedBinary = case\_when(  
 is.na(programsUsed) ~ 0,  
 !is.na(programsUsed) ~ 1)) -> dataAugmented

## Select Cases

The following code chunk removes cases where the respondent was part of a protected population or did not provide informed consent.

dataAugmented %>%  
 subset(currentlyIncarcerated=2) %>%  
 subset(currentlyDetained=2) %>%  
 subset(residentialTreatment=2) %>%  
 subset(involCommitment=2) -> dataClean

## Evaluate Missing Data

The following code chunk evaluates missing data.

miss\_var\_summary(dataClean, order = TRUE)

## # A tibble: 119 x 3  
## variable n\_miss pct\_miss  
## <chr> <int> <dbl>  
## 1 genderSelfDescribe 59 100   
## 2 ethnicityRaceOther 56 94.9  
## 3 financialSupportNonGovt 54 91.5  
## 4 selfEmployment 54 91.5  
## 5 helpAfterOther 53 89.8  
## 6 finanicalSupportOther 51 86.4  
## 7 helpDuringOther 51 86.4  
## 8 ReligiousOther 49 83.1  
## 9 stayInitialOther 47 79.7  
## 10 financialSupportGovt 39 66.1  
## # ... with 109 more rows

miss\_case\_summary(dataClean, order = TRUE)

## # A tibble: 59 x 3  
## case n\_miss pct\_miss  
## <int> <int> <dbl>  
## 1 49 100 84.0  
## 2 50 100 84.0  
## 3 53 85 71.4  
## 4 56 84 70.6  
## 5 51 80 67.2  
## 6 58 35 29.4  
## 7 16 22 18.5  
## 8 57 20 16.8  
## 9 52 19 16.0  
## 10 55 18 15.1  
## # ... with 49 more rows

## Calculate Descriptive Statistics

The following code calculates descriptive statistics for select variables of interest.

print("Number of cases")

## [1] "Number of cases"

nrow(dataClean)

## [1] 59

print("gender; 1 - male, 2 - female, 3 - other, 4 - undisclosed")

## [1] "gender; 1 - male, 2 - female, 3 - other, 4 - undisclosed"

table(as.numeric(dataClean$gender))

##   
## 1 2   
## 31 26

print("ethnicityRace; 1 - East Asian, 2 - Central Asian, 3 - Southern Asian, 4 - Black or African-American,  
 5 - Hispanic or Latino, 6 - Middle Eastern or North African, 7 - Native American or Alaska,  
 8 - Native Hawaiian or Pacific Islander, 9 - White or European, 10 - Other")

## [1] "ethnicityRace; 1 - East Asian, 2 - Central Asian, 3 - Southern Asian, 4 - Black or African-American,\n 5 - Hispanic or Latino, 6 - Middle Eastern or North African, 7 - Native American or Alaska,\n 8 - Native Hawaiian or Pacific Islander, 9 - White or European, 10 - Other"

table(as.numeric(dataClean$ethnicityRace))

##   
## 3 4 5 7 9 10   
## 1 13 4 2 34 3

print("religiousAffiliation; 1 - Atheism, 2 - Agnoticism, 3 - Buddhist, 4 - Eastern Orthodox, 5 - Hindu,  
 6 - Islam, 7 - Jewish, 8 - Latter Day Saints, 9 - Non-denominational, 10 - Protestant,  
 11 - Roman Catholic, 12 - Other, 13 - prefer not to answer")

## [1] "religiousAffiliation; 1 - Atheism, 2 - Agnoticism, 3 - Buddhist, 4 - Eastern Orthodox, 5 - Hindu,\n 6 - Islam, 7 - Jewish, 8 - Latter Day Saints, 9 - Non-denominational, 10 - Protestant,\n 11 - Roman Catholic, 12 - Other, 13 - prefer not to answer"

table(as.numeric(dataClean$religousAffiliation))

## < table of extent 0 >

print("educationLevels; 1 - high schoor or GED, 2 - trade school, 3 - some college, 4 - associate,  
 5 - bachelor, 6 - master, 7 - doctorate")

## [1] "educationLevels; 1 - high schoor or GED, 2 - trade school, 3 - some college, 4 - associate,\n 5 - bachelor, 6 - master, 7 - doctorate"

table(as.numeric(dataClean$educationLevels))

##   
## 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17   
## 1 1 1 2 2 2 1 2 1 1 12 2 1 12 9 7

print("whenHighestEd; 1 - before, 2 - during, 3 - after")

## [1] "whenHighestEd; 1 - before, 2 - during, 3 - after"

table(as.numeric(dataClean$whenHighestEd))

##   
## 1 2 3   
## 10 8 39

print("typeHometown; 1 - rural, 2 - suburban, 3 - urban")

## [1] "typeHometown; 1 - rural, 2 - suburban, 3 - urban"

table(as.numeric(dataClean$typeHometown))

##   
## 1 2 3   
## 1 21 34

print("ageNow")

## [1] "ageNow"

describe(as.numeric(dataClean$ageNow), na.rm = TRUE)

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 57 43.65 9.08 44 43.19 8.9 25 70 45 0.64 0.69 1.2

print("ageRelease")

## [1] "ageRelease"

describe(as.numeric(dataClean$ageRelease), na.rm = TRUE)

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 54 33.54 10.23 33 32.95 9.64 15 63 48 0.44 -0.15  
## se  
## X1 1.39

print("incarcerationTime")

## [1] "incarcerationTime"

summary(as.numeric(dataClean$incarcerationTime))

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 23.0 40.0 72.5 103.8 124.0 380.0 25

print("incomeInitial; ordinal scale")

## [1] "incomeInitial; ordinal scale"

table(as.numeric(dataClean$incomeInitial))

##   
## 1 2 3 4 5 6 7 8 10 11 12   
## 23 8 7 2 3 3 3 2 1 1 2

print("incomeLastYr; ordinal scale")

## [1] "incomeLastYr; ordinal scale"

table(as.numeric(dataClean$incomeLastYr))

##   
## 1 2 3 4 5 6 7 8 9 10 11   
## 13 1 3 7 3 4 8 7 3 4 2

print("selfEmployment; 1 - 0%, 2 - 1-25%, 3 - 26-50%, 4 - 51-75%, 5 - 75-99%, 6 - 100%")

## [1] "selfEmployment; 1 - 0%, 2 - 1-25%, 3 - 26-50%, 4 - 51-75%, 5 - 75-99%, 6 - 100%"

table(as.numeric(dataClean$selfEmployment))

##   
## 1 3 4 6   
## 1 1 1 2

## Compute Scale Reliabilities

The following code chunk evaluates the reliability for the MSPSS, ACE, and grit scales by calculating the Cronbach’s alpha for each variable.

reliabilityMSPSS <- cronbach (subset(dataClean, select=c(pssQ1, pssQ2, pssQ3, pssQ4, pssQ5, pssQ6, pssQ7, pssQ8, pssQ9, pssQ10, pssQ11, pssQ12)))  
print("Reliability for MSPSS")

## [1] "Reliability for MSPSS"

reliabilityMSPSS

## $sample.size  
## [1] 52  
##   
## $number.of.items  
## [1] 12  
##   
## $alpha  
## [1] 0.9231374

reliabilityACE <- cronbach (subset(dataClean, select=c(aceQ1, aceQ2, aceQ3, aceQ4, aceQ5, aceQ6, aceQ7, aceQ8, aceQ9, aceQ10)))  
print("Reliability for ACE Scale")

## [1] "Reliability for ACE Scale"

reliabilityACE

## $sample.size  
## [1] 51  
##   
## $number.of.items  
## [1] 10  
##   
## $alpha  
## [1] 0.6815232

reliabilityGrit <- cronbach (subset(dataClean, select=c(gritQ1R, gritQ2, gritQ3R, gritQ4, gritQ5R, gritQ6R, gritQ7, gritQ8)))  
print("Reliability for Short Grit Scale")

## [1] "Reliability for Short Grit Scale"

reliabilityGrit

## $sample.size  
## [1] 53  
##   
## $number.of.items  
## [1] 8  
##   
## $alpha  
## [1] 0.7431265

## Perform Comparison of Means of Grit

The following code chunk performs calculations for comparison of means of the primary variables of interest.

# Comparison of means for `gritScore` grouped by various operationalizations of success using ANOVA  
# Coding for `gritscore`: values range from 1 (low grit) to 8 (high grit)  
# Null hypothesis: the means of the different groups are the same  
# Alternative hypothesis: the sample mean of at least one group is not equal to the others  
  
  
# Comparison of `gritScore` for cases grouped by `incomeLastYr` using ANOVA  
# Coding for `incomeLastYr` (Qualitrics Q22): ordinal ranging 1 - 10  
groupIncome <- group\_by(dataClean, incomeLastYr)  
  
groupIncome %>%  
 summarise(  
 count = n(),  
 mean = mean(gritScore, na.rm = TRUE),  
 sd = sd(gritScore, na.rm = TRUE)  
 )

## # A tibble: 12 x 4  
## incomeLastYr count mean sd  
## <int> <int> <dbl> <dbl>  
## 1 1 13 3.79 0.596  
## 2 2 1 3.62 NaN   
## 3 3 3 4 0.760  
## 4 4 7 3.30 0.710  
## 5 5 3 3.38 0.331  
## 6 6 4 4.41 0.413  
## 7 7 8 3.46 0.425  
## 8 8 7 3.90 0.713  
## 9 9 3 4.29 0.191  
## 10 10 4 4 0.941  
## 11 11 2 2.75 0.884  
## 12 NA 4 NaN NaN

aovGroupIncome <- aov(gritScore ~ as.numeric(incomeLastYr), data = groupIncome)  
summary(aovGroupIncome)

## Df Sum Sq Mean Sq F value Pr(>F)  
## as.numeric(incomeLastYr) 1 0.002 0.0018 0.004 0.951  
## Residuals 51 23.886 0.4683   
## 6 observations deleted due to missingness

# Comparison of `gritScore` for cases grouped by `povertyLevel` using ANOVA  
# Coding for `povertyLevel` (created variable): 1 - above, 0 - below  
dataClean %>%  
 mutate(povertyLevel = if (povertyRatio >1) {1} else {0})-> dataClean  
  
dataClean[dataClean==""] <- NA  
  
groupPoverty <- group\_by(dataClean, povertyLevel)  
  
groupPoverty %>%  
 summarise(  
 count = n(),  
 mean = mean(gritScore, na.rm = TRUE),  
 sd = sd(gritScore, na.rm = TRUE)  
 )

## # A tibble: 1 x 4  
## povertyLevel count mean sd  
## <dbl> <int> <dbl> <dbl>  
## 1 0 59 3.73 0.678

aovGroupPoverty <- aov(gritScore ~ as.numeric(povertyLevel), data = groupPoverty)  
summary(aovGroupPoverty)

## Df Sum Sq Mean Sq F value Pr(>F)  
## Residuals 52 23.89 0.4594   
## 6 observations deleted due to missingness

# Comparison of `gritScore` for cases grouped by `financialStatus` using ANOVA  
# Coding for `financialStatus` (Qualitrics Q23): 1 - not able, 2 - just able, 3 - comfortable no savings, 4 - comfortable and saving  
groupFinStat <- group\_by(dataClean, financialStatus)  
  
groupFinStat %>%  
 summarise(  
 count = n(),  
 mean = mean(gritScore, na.rm = TRUE),  
 sd = sd(gritScore, na.rm = TRUE)  
 )

## # A tibble: 5 x 4  
## financialStatus count mean sd  
## <int> <int> <dbl> <dbl>  
## 1 1 12 3.70 0.631   
## 2 2 19 3.54 0.723   
## 3 3 15 3.62 0.604   
## 4 4 7 4.34 0.519   
## 5 NA 6 4.31 0.0884

aovGroupFinStat <- aov(gritScore ~ as.numeric(financialStatus), data = groupFinStat)  
summary(aovGroupFinStat)

## Df Sum Sq Mean Sq F value Pr(>F)  
## as.numeric(financialStatus) 1 1.255 1.2553 2.806 0.1  
## Residuals 49 21.922 0.4474   
## 8 observations deleted due to missingness

# Comparison of `gritScore` for cases grouped by `savings` using ANOVA  
# Coding for `savings` (Qualitrics Q24): 1 - yes, 2 - no  
groupSavings <- group\_by(dataClean, savings)  
  
groupSavings %>%  
 summarise(  
 count = n(),  
 mean = mean(gritScore, na.rm = TRUE),  
 sd = sd(gritScore, na.rm = TRUE)  
 )

## # A tibble: 3 x 4  
## savings count mean sd  
## <int> <int> <dbl> <dbl>  
## 1 1 15 4.18 0.551  
## 2 2 40 3.56 0.648  
## 3 NA 4 NaN NaN

aovGroupSavings <- aov(gritScore ~ as.numeric(savings), data = groupSavings)  
summary(aovGroupSavings)

## Df Sum Sq Mean Sq F value Pr(>F)   
## as.numeric(savings) 1 4.122 4.122 10.63 0.00198 \*\*  
## Residuals 51 19.766 0.388   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
## 6 observations deleted due to missingness

## Perform Correlational Analysis

The following code chunk performs correlational analysis for the primary variables of interest.

dataClean %>%  
 dplyr::select(incomeLastYr, povertyRatio, financialStatus, savings, programsUsedBinary, ageNow, ageRelease, mspssScore, aceScore, gritScore) -> correlationDataset   
col\_names <- names(correlationDataset)  
correlationDataset[,col\_names] <- lapply(correlationDataset[,col\_names], as.numeric)  
correlation <- cor(correlationDataset, use="pairwise.complete.obs")  
correlation

## incomeLastYr povertyRatio financialStatus savings  
## incomeLastYr 1.00000000 0.90578130 0.40586231 -0.17256374  
## povertyRatio 0.90578130 1.00000000 0.24798815 -0.10726547  
## financialStatus 0.40586231 0.24798815 1.00000000 -0.46516980  
## savings -0.17256374 -0.10726547 -0.46516980 1.00000000  
## programsUsedBinary -0.22678323 -0.12165382 0.16810158 -0.11895774  
## ageNow -0.11081109 0.03311116 -0.24607676 0.07952369  
## ageRelease -0.24684348 -0.08304453 -0.28127030 0.05394767  
## mspssScore 0.08568878 0.06090829 -0.05715269 -0.07887455  
## aceScore 0.12372821 0.18268279 0.05294731 -0.11583557  
## gritScore 0.00860284 0.01599544 0.23272742 -0.41539499  
## programsUsedBinary ageNow ageRelease mspssScore  
## incomeLastYr -0.226783228 -0.11081109 -0.246843481 0.08568878  
## povertyRatio -0.121653817 0.03311116 -0.083044529 0.06090829  
## financialStatus 0.168101584 -0.24607676 -0.281270301 -0.05715269  
## savings -0.118957738 0.07952369 0.053947668 -0.07887455  
## programsUsedBinary 1.000000000 0.06561539 -0.005941853 0.09952873  
## ageNow 0.065615387 1.00000000 0.820731148 0.11755203  
## ageRelease -0.005941853 0.82073115 1.000000000 0.25976062  
## mspssScore 0.099528730 0.11755203 0.259760622 1.00000000  
## aceScore 0.048407327 0.21280707 0.276536370 0.14027692  
## gritScore 0.331767042 0.09270469 -0.020837932 0.07119170  
## aceScore gritScore  
## incomeLastYr 0.12372821 0.00860284  
## povertyRatio 0.18268279 0.01599544  
## financialStatus 0.05294731 0.23272742  
## savings -0.11583557 -0.41539499  
## programsUsedBinary 0.04840733 0.33176704  
## ageNow 0.21280707 0.09270469  
## ageRelease 0.27653637 -0.02083793  
## mspssScore 0.14027692 0.07119170  
## aceScore 1.00000000 0.11068394  
## gritScore 0.11068394 1.00000000

## Perform Regression Analysis

The following code chunk performs a multiple regression using select variables of interest.

# Multiple regression using `incomeLastYr` as dependent variable  
regressionIncome <- lm(as.numeric(incomeLastYr) ~ as.numeric(programsUsedBinary) + as.numeric(ageNow) + as.numeric(ageRelease) + mspssScore + aceScore + gritScore,   
 data = dataClean, na.action = na.omit)  
summary(regressionIncome)

##   
## Call:  
## lm(formula = as.numeric(incomeLastYr) ~ as.numeric(programsUsedBinary) +   
## as.numeric(ageNow) + as.numeric(ageRelease) + mspssScore +   
## aceScore + gritScore, data = dataClean, na.action = na.omit)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -5.7961 -2.4642 0.0358 2.2996 4.1183   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 9.78494 4.34513 2.252 0.03003 \*   
## as.numeric(programsUsedBinary) -7.75324 3.16187 -2.452 0.01878 \*   
## as.numeric(ageNow) 0.15004 0.08932 1.680 0.10101   
## as.numeric(ageRelease) -0.21806 0.07784 -2.801 0.00788 \*\*  
## mspssScore 0.37103 0.28880 1.285 0.20647   
## aceScore 0.10716 0.16659 0.643 0.52384   
## gritScore 0.06379 0.66060 0.097 0.92356   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.902 on 39 degrees of freedom  
## (13 observations deleted due to missingness)  
## Multiple R-squared: 0.2549, Adjusted R-squared: 0.1403   
## F-statistic: 2.224 on 6 and 39 DF, p-value: 0.06103

# Multiple regressiong using `povertyRatio` as dependent variable  
regressionPoverty <- lm(as.numeric(povertyRatio) ~ as.numeric(programsUsedBinary) +   
 as.numeric(ageNow) + as.numeric(ageRelease) + mspssScore + aceScore + gritScore,   
 data = dataClean, na.action = na.omit)  
summary(regressionPoverty)

##   
## Call:  
## lm(formula = as.numeric(povertyRatio) ~ as.numeric(programsUsedBinary) +   
## as.numeric(ageNow) + as.numeric(ageRelease) + mspssScore +   
## aceScore + gritScore, data = dataClean, na.action = na.omit)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.72049 -0.34659 -0.06313 0.33049 0.77768   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 0.836482 0.738327 1.133 0.265  
## as.numeric(programsUsedBinary) -0.537065 0.524116 -1.025 0.313  
## as.numeric(ageNow) 0.022002 0.015135 1.454 0.155  
## as.numeric(ageRelease) -0.020430 0.012919 -1.581 0.123  
## mspssScore 0.021861 0.051382 0.425 0.673  
## aceScore 0.014792 0.030142 0.491 0.627  
## gritScore -0.006524 0.116590 -0.056 0.956  
##   
## Residual standard error: 0.4754 on 33 degrees of freedom  
## (19 observations deleted due to missingness)  
## Multiple R-squared: 0.1002, Adjusted R-squared: -0.06344   
## F-statistic: 0.6122 on 6 and 33 DF, p-value: 0.7188

# Multiple regression using `financialStatus` as dependent variable  
regressionFinStat <- lm(as.numeric(financialStatus) ~ as.numeric(programsUsedBinary) + as.numeric(ageNow) +   
 as.numeric(ageRelease) + mspssScore + aceScore + gritScore,   
 data = dataClean, na.action = na.omit)  
summary(regressionFinStat)

##   
## Call:  
## lm(formula = as.numeric(financialStatus) ~ as.numeric(programsUsedBinary) +   
## as.numeric(ageNow) + as.numeric(ageRelease) + mspssScore +   
## aceScore + gritScore, data = dataClean, na.action = na.omit)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.8122 -0.9299 0.0470 0.8560 1.4546   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 1.984179 1.509946 1.314 0.197  
## as.numeric(programsUsedBinary) -0.459070 1.097905 -0.418 0.678  
## as.numeric(ageNow) -0.004842 0.031262 -0.155 0.878  
## as.numeric(ageRelease) -0.025577 0.027548 -0.928 0.359  
## mspssScore -0.015915 0.101484 -0.157 0.876  
## aceScore 0.031918 0.058200 0.548 0.587  
## gritScore 0.362383 0.232527 1.558 0.127  
##   
## Residual standard error: 1.008 on 38 degrees of freedom  
## (14 observations deleted due to missingness)  
## Multiple R-squared: 0.1474, Adjusted R-squared: 0.01279   
## F-statistic: 1.095 on 6 and 38 DF, p-value: 0.3831

# Binary logistic regression using `savings` as dependent variable  
# Prepare data  
# Recode `savings` with 1 - yes, 0 - no  
dataClean %>%  
 dplyr::select(incomeLastYr, povertyRatio, povertyLevel, financialStatus, savings, programsUsedBinary,   
 ageNow, ageRelease, mspssScore, aceScore, gritScore) -> dataBinary  
dataBinary$savings <- ifelse(dataBinary$savings==2,0,ifelse(dataBinary$savings==1,1,dataBinary$savings))  
  
# Remove cases with missing data in the `savings` variable  
dataBinary %>%  
 drop\_na(savings, programsUsedBinary, ageNow, ageRelease, mspssScore, aceScore, gritScore) -> dataBinary  
  
logitSavings <- glm(savings ~ programsUsedBinary + ageNow + ageRelease + mspssScore + aceScore + gritScore,   
 data = dataBinary, family = binomial, na.action = na.omit)  
summary(logitSavings)

##   
## Call:  
## glm(formula = savings ~ programsUsedBinary + ageNow + ageRelease +   
## mspssScore + aceScore + gritScore, family = binomial, data = dataBinary,   
## na.action = na.omit)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.3849 -0.6959 -0.3094 0.4761 1.9046   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.941e+01 2.400e+03 -0.008 0.99354   
## programsUsedBinary 1.075e+01 2.400e+03 0.004 0.99643   
## ageNow -1.373e-01 8.866e-02 -1.549 0.12148   
## ageRelease -2.546e-04 6.669e-02 -0.004 0.99695   
## mspssScore 4.129e-01 3.263e-01 1.265 0.20579   
## aceScore 1.656e-01 1.640e-01 1.010 0.31252   
## gritScore 2.277e+00 8.663e-01 2.628 0.00859 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 54.777 on 45 degrees of freedom  
## Residual deviance: 38.943 on 39 degrees of freedom  
## AIC: 52.943  
##   
## Number of Fisher Scoring iterations: 15

# Raise e to the coefficients  
print(exp(coef(logitSavings)))

## (Intercept) programsUsedBinary ageNow   
## 3.705257e-09 4.660373e+04 8.717094e-01   
## ageRelease mspssScore aceScore   
## 9.997454e-01 1.511203e+00 1.180139e+00   
## gritScore   
## 9.742986e+00

# Obtain various pseudo R-squared measures  
print(pR2(logitSavings))

## llh llhNull G2 McFadden r2ML r2CU   
## -19.4714248 -27.3884131 15.8339766 0.2890634 0.2912248 0.4184131

# Confidence intervals for the coefficients  
print(exp(confint(logitSavings, level = 0.95)))

## 2.5 % 97.5 %  
## (Intercept) NA 6.526162e+193  
## programsUsedBinary 1.712179e-195 NA  
## ageNow 7.139347e-01 1.021074e+00  
## ageRelease 8.777893e-01 1.149878e+00  
## mspssScore 8.247072e-01 3.071789e+00  
## aceScore 8.654503e-01 1.672234e+00  
## gritScore 2.250240e+00 7.122626e+01

# Hosmer-Lemeshow Goodness of Fit Test  
# Null hypothesis: the model is a good fit for the data  
# Alternative hypothesis: the model is NOT a good fit for the data  
HosLemLogitSavings <- hoslem.test(dataBinary$savings,   
 fitted(logitSavings), g=10)  
print(HosLemLogitSavings)

##   
## Hosmer and Lemeshow goodness of fit (GOF) test  
##   
## data: dataBinary$savings, fitted(logitSavings)  
## X-squared = 3.4505, df = 8, p-value = 0.903

print(cbind(HosLemLogitSavings$expected, HosLemLogitSavings$observed))

## yhat0 yhat1 y0 y1  
## [6.39e-08,0.00789] 4.976349 0.02365138 5 0  
## (0.00789,0.0345] 4.902222 0.09777832 5 0  
## (0.0345,0.0891] 3.689023 0.31097700 4 0  
## (0.0891,0.171] 4.369081 0.63091948 4 1  
## (0.171,0.208] 3.269486 0.73051444 3 1  
## (0.208,0.29] 3.667567 1.33243309 3 2  
## (0.29,0.373] 2.702858 1.29714185 2 2  
## (0.373,0.527] 2.678299 2.32170118 4 1  
## (0.527,0.616] 1.594707 2.40529261 2 2  
## (0.616,0.921] 1.150409 3.84959071 1 4

## Save Data

The following code chunk saves the cleaned data used for the analysis.

write.csv(dataClean, here("Data","DataClean","SOC5800\_Data\_NumericValues\_Clean\_CSV.csv"), append = FALSE)