R Notebook: SOC 5800 Team Project

Malcolm S. Townes

(November 21, 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(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 %>%  
 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,  
 primaryEthnicityRace = 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$ageNow <- as.numeric(dataAugmented$ageNow)  
#dataAugmented$ageRelease <- as.numeric(dataAugmented$ageRelease)  
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)+17)  
dataAugmented$ageRelease <- (as.numeric(dataAugmented$ageRelease)+17)  
dataAugmented[dataAugmented==""] <- NA

## Create Variables

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

dataAugmented %>%  
 mutate (mspssScore = (pssQ1+pssQ2+pssQ2+pssQ2+pssQ2+pssQ2+pssQ2+pssQ2+pssQ2+pssQ2+pssQ2+pssQ2)/12) %>%  
 mutate (aceScore = 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 = (gritQ1R+gritQ2+gritQ3R+gritQ4+gritQ5R+gritQ6R+gritQ7+gritQ8)/8) %>%  
 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)))  
 }  
 }  
 }  
 }  
 }  
 ) -> dataAugmented

## Warning in if (as.numeric(incomeLastYr) <= 6) {: the condition has length >  
## 1 and only the first element will be used

## Warning in if (as.numeric(incomeLastYr) == 7) {: the condition has length >  
## 1 and only the first element will be used

## Warning in if (as.numeric(incomeLastYr) == 8) {: the condition has length >  
## 1 and only the first element will be used

## Warning in if (as.numeric(incomeLastYr) == 9) {: the condition has length >  
## 1 and only the first element will be used

## Warning in if (as.numeric(incomeLastYr) == 10) {: the condition has length  
## > 1 and only the first element will be used

## Select Cases

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

dataAugmented <- dataAugmented[-c(1,2),]  
dataAugmented %>%  
 subset(currentlyIncarcerated=2) %>%  
 subset(currentlyDetained=2) %>%  
 subset(residentialTreatment=2) %>%  
 subset(involCommitment=2) -> dataClean

## Missing Data

The following code chunk evaluates missing data.

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

## # A tibble: 117 x 3  
## variable n\_miss pct\_miss  
## <chr> <int> <dbl>  
## 1 genderSelfDescribe 59 100   
## 2 primaryEthnicityRace 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 107 more rows

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

## # A tibble: 59 x 3  
## case n\_miss pct\_miss  
## <int> <int> <dbl>  
## 1 49 60 51.3  
## 2 50 60 51.3  
## 3 53 47 40.2  
## 4 56 45 38.5  
## 5 51 43 36.8  
## 6 58 22 18.8  
## 7 16 21 17.9  
## 8 57 19 16.2  
## 9 52 18 15.4  
## 10 55 17 14.5  
## # ... with 49 more rows

## 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] "gender"

table(dataClean$gender)

##   
##   
## 0   
## {"ImportId":"QID5"}   
## 0   
## 1   
## 31   
## 2   
## 26   
## What is your gender? (select only one) - Selected Choice   
## 0

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 59 33.19 7.61 34 33.24 8.9 18 47 29 -0.01 -0.87 0.99

print("ageRelease")

## [1] "ageRelease"

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

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 59 32.41 9.5 32 32.22 10.38 18 49 31 0.17 -1.14 1.24

print("selfEmployement")

## [1] "selfEmployement"

table(dataClean$selfEmployment)

##   
##   
## 0   
## {"ImportId":"QID20"}   
## 0   
## 1   
## 1   
## 3   
## 1   
## 4   
## 1   
## 6   
## 2   
## About what percent\nof your total gross household income last year was attributable to\nself-employment or businesses that you owned? (select only one)   
## 0

## 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 MPSS Scale")

## [1] "Reliability for MPSS Scale"

reliabilityMSPSS

## $sample.size  
## [1] 59  
##   
## $number.of.items  
## [1] 12  
##   
## $alpha  
## [1] 0.9587711

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] 59  
##   
## $number.of.items  
## [1] 10  
##   
## $alpha  
## [1] 0.9634455

reliabilityGrit <- cronbach (subset(dataClean, select=c(gritQ1, gritQ2, gritQ3, gritQ4, gritQ5, gritQ6, gritQ7, gritQ8)))  
print("Reliability for Short Grit Scale")

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

reliabilityGrit

## $sample.size  
## [1] 59  
##   
## $number.of.items  
## [1] 8  
##   
## $alpha  
## [1] 0.906496

## Comparison of Means

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

# Comparison of grit for cases grouped by financialStatus  
groups <- group\_by(dataClean, financialStatus)

## Warning: Factor `financialStatus` contains implicit NA, consider using  
## `forcats::fct\_explicit\_na`

aovGroup <- aov(as.numeric(financialStatus) ~ gritScore, data = groups)  
summary(aovGroup)

## Df Sum Sq Mean Sq F value Pr(>F)   
## gritScore 1 2.60 2.5979 2.822 0.0991 .  
## Residuals 51 46.95 0.9206   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
## 6 observations deleted due to missingness

## Correlational Analysis

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

dataClean %>%  
 dplyr::select(povertyRatio, financialStatus, 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

## povertyRatio financialStatus ageNow ageRelease  
## povertyRatio 1.00000000 -0.12702067 0.15261350 0.18136159  
## financialStatus -0.12702067 1.00000000 -0.27436238 -0.09029268  
## ageNow 0.15261350 -0.27436238 1.00000000 0.35681341  
## ageRelease 0.18136159 -0.09029268 0.35681341 1.00000000  
## mspssScore 0.13622780 0.02088214 0.22150099 0.04763569  
## aceScore 0.09382109 0.18032901 0.34336352 0.32248113  
## gritScore 0.06701438 0.22898025 0.08405773 0.08637535  
## mspssScore aceScore gritScore  
## povertyRatio 0.13622780 0.09382109 0.06701438  
## financialStatus 0.02088214 0.18032901 0.22898025  
## ageNow 0.22150099 0.34336352 0.08405773  
## ageRelease 0.04763569 0.32248113 0.08637535  
## mspssScore 1.00000000 0.63305300 0.18757349  
## aceScore 0.63305300 1.00000000 0.07753448  
## gritScore 0.18757349 0.07753448 1.00000000

## Regression Analysis

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

regressionPoverty <- lm(as.numeric(povertyRatio) ~ 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(ageNow) +   
## as.numeric(ageRelease) + mspssScore + aceScore + gritScore,   
## data = dataClean, na.action = na.omit)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.0251 -0.5506 -0.1037 0.6112 2.3136   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.6359965 0.9822861 3.702 0.000582 \*\*\*  
## as.numeric(ageNow) 0.0097614 0.0191025 0.511 0.611851   
## as.numeric(ageRelease) 0.0174512 0.0158113 1.104 0.275582   
## mspssScore 0.0596084 0.0671626 0.888 0.379516   
## aceScore -0.0054573 0.0224945 -0.243 0.809413   
## gritScore -0.0008674 0.2022027 -0.004 0.996596   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.9016 on 45 degrees of freedom  
## (8 observations deleted due to missingness)  
## Multiple R-squared: 0.06231, Adjusted R-squared: -0.04187   
## F-statistic: 0.5981 on 5 and 45 DF, p-value: 0.7015

## 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)

## Warning in write.csv(dataClean, here("Data", "DataClean",  
## "SOC5800\_Data\_NumericValues\_Clean\_CSV.csv"), : attempt to set 'append'  
## ignored