**Appendix**

# Read in libraries

library(tidyr)

library(dplyr)

library(ggplot2)

library(caret)

library(grpreg)

library(glmnet)

library(MASS)

library(mgcv)

library(tidyverse)

library(tidymv)

library(jtools)

library(GGally)

library(boot)

library(ISLR)

library(leaps)

**Question 1 Analysis**

data <- read.csv(unz("csv\_pmd.zip", "psam\_p24.csv"),

stringsAsFactors = FALSE)

newdata <- data[,c("AGEP", "WAGP", "COW", "SEX", "SCHL")]

## omit na values #

P\_MD <- na.omit(newdata)

P\_MD <- as.tibble(P\_MD)

# Convert to factor type variables #

cols <- c("COW", "SEX", "SCHL")

P\_MD[cols] <- lapply(P\_MD[cols], factor)

# Summary data

summary(P\_MD)

# Data Structure

str(P\_MD)

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

# Some Exploratory Analysis #

ggpairs(P\_MD, columns = 1:3)

str(P\_MD)

plot(P\_MD$SCHL, P\_MD$WAGP, data=P\_MD)

plot(P\_MD$AGEP, P\_MD$WAGP)

hist(P\_MD$WAGP)

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

# Initial exploratory models, Model Checking and Refinement #

par(mfrow=c(1,3))

linmod.1 <- lm(WAGP~AGEP, data=P\_MD)

summary(linmod.1)

resid.1 <- linmod.1$residuals

fitted.1 <- linmod.1$fitted.values

plot(fitted.1,resid.1)

## We will report results for only incomes above $100 #

P\_MD$WAGP[P\_MD$WAGP <100] <- NA

P\_MD <- na.omit(P\_MD)

str(P\_MD)

min(P\_MD$WAGP)

# Log transform #

linmod.2 <- lm(log(WAGP)~AGEP, data=P\_MD)

summary(linmod.2)

resid.2 <- linmod.2$residuals

fitted.2 <- linmod.2$fitted.values

plot(fitted.2, resid.2)

str(P\_MD$SCHL)

length(P\_MD$SCHL)

key <- c(0,2,3,4,5,6,7,8,9,10,11,12,13,14,14.5,15,14.75,15.5,16.5,19,21,23,23,25)

length(key)

key[1]

P\_MD$SCHL <- as.numeric(P\_MD$SCHL)

for(i in 1:24){

veci <- P\_MD$SCHL == i

P\_MD$SCHL[veci] <- key[i]

}

str(P\_MD$SCHL)

# Remove zero school observations #

P\_MD$SCHL[P\_MD$SCHL <1] <- NA

P\_MD <- na.omit(P\_MD)

str(P\_MD)

min(P\_MD$SCHL)

P\_MD$NGRAD <- NA

P\_MD$NGRAD <- as.numeric(P\_MD$SCHL)<16

P\_MD$NGRAD <- as.numeric(P\_MD$NGRAD)

str(P\_MD$NGRAD)

str(P\_MD)

## Try adding SCHL, SEX, and NGRAD to the equation ##

linmod.3 <- lm(log(WAGP) ~ AGEP + SEX + SCHL + NGRAD, data=P\_MD)

resid.3 <- linmod.3$residuals

fitted.3 <- linmod.3$fitted.values

plot(fitted.3, resid.3)

# Try adding polynomials of school and age:

linmod.p <- lm(log(WAGP) ~ P\_MD$NGRAD +

poly(P\_MD$AGEP,3, raw=TRUE) + P\_MD$SEX +

poly(P\_MD$SCHL,3,raw=TRUE), data =P\_MD)

summary(linmod.p)

resid.p <- linmod.p$residuals

fitted.p <- linmod.p$fitted.values

ggplot(P\_MD,aes(fitted.p, resid.p)) +

geom\_point(alpha = 0.01, color = "dark blue") +

ggtitle("Fit and Resid for model with age, school polynomials") +

ylab("Resid.p") +

xlab("Fitted.p")

### Let's try a Box Cox transformation for the WAGP variable.

library(MASS)

linmodbc <- lm(WAGP ~ P\_MD$NGRAD + poly(P\_MD$AGEP,3,raw=TRUE) +poly(P\_MD$SCHL,4,raw=TRUE), data=P\_MD)

boxcox(linmodbc, plotit = TRUE)

boxcox(linmodbc, plotit = TRUE, lambda = seq(-.3, .3, by = 0.05))

lambdab <- .24

linmod.bc <- lm((((WAGP ^ lambdab) - 1) / lambdab) ~ P\_MD$NGRAD +

poly(P\_MD$AGEP,3, raw=TRUE) + P\_MD$SEX +

poly(P\_MD$SCHL,4,raw=TRUE), data =P\_MD)

summary(linmod.bc)

resid.bc <- linmod.bc$residuals

fitted.bc <- linmod.bc$fitted.values

par(mfrow=c(1,2))

plot(fitted.bc,resid.bc)

## The Box Cox transformation helped a bit, let's check whether a log transform is better.

linmod.lw <- lm(log(WAGP) ~ P\_MD$NGRAD +

poly(P\_MD$AGEP,3, raw=TRUE) + P\_MD$SEX +

poly(P\_MD$SCHL,4,raw=TRUE),data =P\_MD)

summary(linmod.lw)

resid.lw <- linmod.lw$residuals

fitted.lw <- linmod.lw$fitted.values

plot(fitted.lw,resid.lw)

# Log transform of WAGP #

P\_MD$WAGP.l <- log(P\_MD$WAGP)

str(P\_MD)

P\_MD <- dplyr::select(P\_MD, -c("WAGP"))

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

## Model Selection Process

## We use best subsets to look for a good model that considers polynomials

# of age and school, as well as interactions of age and school.

regfit.full=regsubsets(WAGP.l ~ NGRAD + poly(AGEP,3, raw=TRUE) +

SEX + poly(SCHL,3, raw=TRUE) + AGEP\*SCHL, data =P\_MD, nvmax=20)

reg.summ <- summary(regfit.full)

# Now we evaluate the results of the best subsets selection method using RSS,

# Adusted R squared, cP and BIC statistics.

par(mfrow=c(2,2))

plot(reg.summ$rss, xlab="Number of Variables", ylab="RSS",

type="l")

which.min(reg.summ$rss)

points(9,reg.summ$rss[9], col ="green",cex =2, pch =20)

plot(reg.summ$adjr2,xlab="Number of Variables",

ylab=" Adjusted RSq",type="l")

which.max(reg.summ$adjr2)

points(9,reg.summ$adjr2[9], col ="green",cex =2, pch =20)

plot(reg.summ$cp,xlab="Number of Variables",ylab="Cp",

type='l')

which.min(reg.summ$cp)

points(9,reg.summ$cp[9],col="green",cex=2,pch=20)

which.min(reg.summ$bic)

plot(reg.summ$bic,xlab="Number of Variables",ylab="BIC",

type='l')

points(9,reg.summ$bic[9],col="green",cex=2,pch=20)

par(mfrow=c(2,2))

plot(regfit.full,scale="r2")

plot(regfit.full,scale="adjr2")

plot(regfit.full,scale="Cp")

plot(regfit.full,scale="bic")

# Most statistics are optimal with a 9 variable model, so we will use the 9 variable model.

coef(regfit.full, 9)

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

P\_MD\_D <- fastDummies::dummy\_cols(P\_MD)

str(P\_MD\_D)

P\_MD\_D$AGEPsq <- P\_MD\_D$AGEP^2

P\_MD\_D$SCHLsq <- P\_MD\_D$SCHL^2

P\_MD\_D$AGEPcu <- P\_MD\_D$AGEP^3

P\_MD\_D$SCHLcu <- P\_MD\_D$SCHL^3

pred\_best <-lm(P\_MD\_D$WAGP.l ~ NGRAD + P\_MD\_D$AGEP + P\_MD\_D$AGEPsq + P\_MD\_D$AGEPcu +

P\_MD\_D$SEX\_2 + P\_MD\_D$SCHL +

P\_MD\_D$SCHLsq + P\_MD\_D$SCHLcu + P\_MD\_D$COW\_2+ P\_MD\_D$COW\_3 +

P\_MD\_D$COW\_4 + P\_MD\_D$COW\_5 + P\_MD\_D$COW\_6 + P\_MD\_D$COW\_7 + P\_MD\_D$COW\_8 +

P\_MD\_D$COW\_9, data = P\_MD\_D)

## Interpretation: look at the R-sq and F-stat of the model #

best\_summ <- summary(pred\_best)

names(best\_summ)

coefs.b <- (((exp(best\_summ$coefficients) -1)\*100))

best\_summ$coefficients

coefs.b

confint(pred\_best, level = 0.95)

**Question 2 Analysis**

# Read in csv file

#data <- read.csv(unz("csv\_pmd.zip", "psam\_p24.csv"),

# nrows = 10000,

# stringsAsFactors = FALSE)

# Get a summary of the data

summary(q2data)

str(q2data)

# Remove anyone who did not attend school

q2data$SCHL[q2data$SCHL == 1] <- N

# Remove the top earners from the data set. These values are top coded and are

# considered outliers for the purpose of this question.

q2data$WAGP[q2data$WAGP > 400000] <- NA

q2data$WAGP[q2data$WAGP < 10000] <- NA

# Create an indicator variable for high school graduates

q2data$GRAD <- if\_else(q2data$SCHL >15, 1, 0)

# Remove any remaining rows containing NA values.

q2data <- na.omit(q2data)

# Convert COW variable to factor variable

q2data$COW <- as.factor(q2data$COW)

# Plot class of worker against income

ggplot(q2data) +

geom\_boxplot(aes(COW, WAGP))

# Plot class of worker against income

ggplot(q2data) +

geom\_boxplot(aes(COW, log(WAGP)))

# Pull the first three characters from the SOCP variable to decode the industry

# the worker is in

q2data$SOCP.3 <- substr(q2data$SOCP, 1, 3)

head(q2data)

# Decode the industry according to the technical documentation

q2data$SOCP.3 <- recode(q2data$SOCP.3,

'111'= 'MGR',

'112'= 'MGR',

'113'= 'MGR',

'119'= 'MGR',

'131'= 'BUS',

'132' = 'FIN',

'151' = 'CMM',

'152' = 'CMM',

'171' = 'ENG',

'172' = 'ENG',

'173' = 'ENG',

'191' = 'SCI',

'192' = 'SCI',

'193' = 'SCI',

'194' = 'SCI',

'195' = 'SCI',

'211' = 'CMS',

'212' = 'CMS',

'231' = 'LGL',

'232' = 'LGL',

'251' = 'EDU',

'252' = 'EDU',

'253' = 'EDU',

'254' = 'EDU',

'259' = 'EDU',

'271' = 'ENT',

'272' = 'ENT',

'274' = 'ENT',

'291' = 'MED',

'292' = 'MED',

'299' = 'MED',

'311'= 'HLS',

'312'= 'HLS',

'319' = 'HLS',

'331'= 'PRT',

'332'= 'PRT',

'333'= 'PRT',

'339' = 'PRT',

'351'= 'EAT',

'352'= 'EAT',

'353'= 'EAT',

'359' = 'EAT',

'371' = 'CLN',

'372' = 'CMM',

'373' = 'CMM',

'391' = 'PRS',

'392' = 'PRS',

'393' = 'PRS',

'394' = 'PRS',

'395' = 'PRS',

'396' = 'PRS',

'397' = 'PRS',

'399' = 'PRS',

'411' = 'SAL',

'412' = 'SAL',

'413' = 'SAL',

'414' = 'SAL',

'419' = 'SAL',

'431' = 'OFF',

'432' = 'OFF',

'433' = 'OFF',

'434' = 'OFF',

'435' = 'OFF',

'436' = 'OFF',

'439' = 'OFF',

'451' = 'FFF',

'452' = 'FFF',

'453' = 'FFF',

'454' = 'FFF',

'471' = 'CON',

'472' = 'CON',

'473' = 'CON',

'474' = 'CON',

'475' = 'EXT',

'491' = 'RPR',

'492' = 'RPR',

'493' = 'RPR',

'499' = 'RPR',

'511' = 'PRD',

'512' = 'PRD',

'513' = 'PRD',

'514' = 'PRD',

'515' = 'PRD',

'516' = 'PRD',

'517' = 'PRD',

'518' = 'PRD',

'519' = 'PRD',

'531' = 'TRN',

'532' = 'TRN',

'533' = 'TRN',

'534' = 'TRN',

'535' = 'TRN',

'536' = 'TRN',

'537' = 'TRN',

'551' = 'MIL',

'552' = 'MIL',

'553' = 'MIL',

'559' = 'MIL',

'999' = 'UMP',

.default = 'UNK'

)

# Convert columns of interest to factor variables

q2data$SOCP.3 <- as.factor(q2data$SOCP.3)

# Generate the first linear model and provide summary statistics

q2lm <- lm(data = q2data, log(WAGP) ~ COW + SOCP.3 + AGEP + SCHL + GRAD + SCHL\*GRAD)

summary(q2lm)

plot(q2lm$fitted.values, q2lm$residuals)

# Transform the data into one hot encoded variables for COW and SOCP variables

q2data.onehot <- q2data[, c(4,3,8,1,6,7)]

q2data.onehot$WAGP <- log(q2data.onehot$WAGP)

q2dmy <- dummyVars("~.", data = q2data.onehot)

q2trnsf <- data.frame(predict(q2dmy, newdata = q2data.onehot))

head(q2trnsf)

# Create X and Y objects for grouped lasso regression model

X <- as.matrix(q2trnsf[, 2:38])

y <- as.numeric(q2trnsf[, 1])

# Plot the values of coefficients for different values of lambda

plot(grplasso)

group <- c(rep(1,8), rep(2, 26), 0,0,0)

grplasso <- grpreg(X, y, group, penalty="grLasso")

# Perform model selection using cross validation

grplasso.cv <- cv.grpreg(X, y, penalty="grLasso")

plot(grplasso.cv)

# Using Elastic net, find the coefficient estimates for the cross validation selected variables

sel.1se <- which.min(grplasso.cv$cve >

(grplasso.cv$cve[grplasso.cv$min] +

grplasso.cv$cvse[grplasso.cv$min]))

sel.lambda <- grplasso.cv$lambda[sel.1se]

grplasso1 <- grpreg(X, y, lambda = sel.lambda, penalty="grLasso")

grplasso1$beta

**Question 3 Analysis**

#people\_MD <- read.csv("psam\_p24.csv")

data <- read.csv("psam\_p24.csv")

newdata <- data[c("AGEP","COW", "MAR", "SCHL", "SEX", "WAGP")]

P\_MD <- na.omit(newdata)

dim(P\_MD)

summary(P\_MD)

#summary(people\_MD)

#P\_MD <- na.omit(people\_MD)

sum(is.na(P\_MD))

dim(P\_MD)

set.seed(1)

train <- sample(c(TRUE,FALSE), nrow(P\_MD), replace=TRUE)

test <- (!train)

regfit.best=regsubsets(WAGP ~ SCHL + AGEP + COW + MAR + SEX, data=P\_MD[train,], nvmax=5)

test.mat <- model.matrix(WAGP ~ SCHL + AGEP + COW + MAR + SEX, data=P\_MD[test,])

val.errors <- rep(NA,5)

for(i in 1:5){

coefi <- coef(regfit.best, id=i)

pred <- test.mat[,names(coefi)] %\*% coefi

val.errors[i] <- mean((P\_MD$WAGP[test]-pred)^2)

}

val.errors

which.min(val.errors)

coef(regfit.best, 5)

k <- 10

set.seed(1)

folds <- sample(1:k, nrow(P\_MD), replace=TRUE)

cv.errors <- matrix(NA, k, 5, dimnames=list(NULL, paste(1:5)))

predict.regsubsets <- function(object, newdata, id, ...){

form <- as.formula(object$call[[2]])

mat <- model.matrix(form, newdata)

coefi <- coef(object,id=id)

xvars <- names(coefi)

mat[,xvars]%\*%coefi

}

for(j in 1:k){

best.fit <- regsubsets(WAGP ~ SCHL + AGEP + COW + MAR + SEX, data=P\_MD[folds!=j,], nvmax=5)

for(i in 1:5){

pred <- predict(best.fit,P\_MD[folds==j,], id=i)

cv.errors[j,i] <- mean((P\_MD$WAGP[folds==j]-pred)^2)

}

}

mean.cv.errors <- apply(cv.errors ,2,mean)

mean.cv.errors

par(mfrow=c(1,1))

plot(mean.cv.errors, type='b')

reg.best <- regsubsets(WAGP ~ SCHL + AGEP + COW + MAR + SEX, data=P\_MD, nvmax=5)

coef(reg.best, 5)

regfit.full <- regsubsets(WAGP ~ SCHL + AGEP + COW + MAR + SEX, data=P\_MD, nvmax = 5)

summary(regfit.full)

plot(regfit.full, scale="r2")

plot(regfit.full, scale="adjr2")

plot(regfit.full, scale="Cp")

plot(regfit.full, scale="bic")

pred\_best <-lm(WAGP ~ SCHL + MAR + SEX, data = P\_MD)

summary(pred\_best)

confint(pred\_best, level = 0.95)

pred\_best\_diag <- augment(pred\_best)

head(pred\_best\_diag)

#qplot(AGEP, .resid, data = pred\_best\_diag) +

# geom\_hline(aes(yintercept=0))

#qplot(COW, .resid, data = pred\_best\_diag) +

# geom\_hline(aes(yintercept=0))

qplot(MAR, .resid, data = pred\_best\_diag) +

geom\_hline(aes(yintercept=0))

qplot(SCHL, .resid, data = pred\_best\_diag) +

geom\_hline(aes(yintercept=0))

qplot(SEX, .resid, data = pred\_best\_diag) +

geom\_hline(aes(yintercept=0))

qplot(.fitted, .resid, data = pred\_best\_diag) +

geom\_hline(aes(yintercept=0))