**#SCRIPT STARTS HERE**

**#Load necessary libraries - if you have not installed these, start with install.packages(“car”) etc**

library(car)

library(DAAG)

library(lmtest)

**#Load dataset - if it is not in SPSS format, use RCmdr to load the dataset and exclude these commands**

library(foreign, pos=4)

Dataset <- read.spss("C:/dataset.sav", use.value.labels=FALSE,max.value.labels=Inf,to.data.frame=TRUE)

colnames(Dataset) <- tolower(colnames(Dataset))

**#Uncomment and run this command if you need to exclude missing data**

**#Dataset <- na.omit(Dataset)**

n <- nrow(Dataset)

**#Define your dependent variable, factors, etc. Add/delete predictors as necessary keeping the same format (pred4, pred5, etc)**

dep <- Dataset$timedrs

pred1 <- Dataset$phyheal

pred2 <- Dataset$menheal

pred3 <- Dataset$stress

**#Set your number of factors here**

numpred <- 3

df = numpred - 1

violations <- NULL

**#Add/remove predictors as necessary here**

RegModel <- lm(dep~pred1+pred2+pred3, data=Dataset)

summary(RegModel)

**#### Homoscedasticity ####**

heteroscedasticity<-bptest(dep ~ pred1 + pred2 + pred3, studentize=TRUE, data=Dataset)

if (heteroscedasticity$p.value < .01) {

          print("Homoscedasticity assumption violated")

          violations <- violations+1

} else {

          print("No problems with homoscedasticity")

}

**#### Multicollinearity ####**

vif(RegModel)

VIFvals<-vif(RegModel)

VIFvals<-as.data.frame(VIFvals)

mcVi <- 0

for (i in 1:numpred) {

if (VIFvals[i,] > 4){

          mcVi <- mcVi+1

}

}

if (mcVi > 0) {

          violations <- violations+1

          print("Multicollinearity assumption violated")

} else {

          print("No problems with Multicollinearity")

}

**#### NORMALITY ####**

**#Shapiro-Wilk**

normality <-shapiro.test(dep)

summary(normality)

qqPlot(dep, main="QQ Plot", id.method="y", id.n=3)

if (normality$p.value<0.01){

          print("Failed Shapiro-Wilks test of normality")

} else {

          print("Passed Shapiro-Wilks test of normality")

}

**##### LINEARITY #######**

resettest(RegModel, power=2:3, type="regressor",data=Dataset)

**#### MULTIVARIATE OUTLIERS ####**

print(“Multivariate outliers”)

outlierTest(RegModel)