Final Exam

Josh Jarvey - DS 705

Unlike the Homework RMD files, this one doesn’t contain all of the questions. Rather we want you to read the questions on D2L and use this RMD file to record your R code that you used to answer each question. This file must knit correctly. After you submit the D2L quiz, then upload this RMD file and the knitted version of this file to the Final Exam dropbox on D2L.

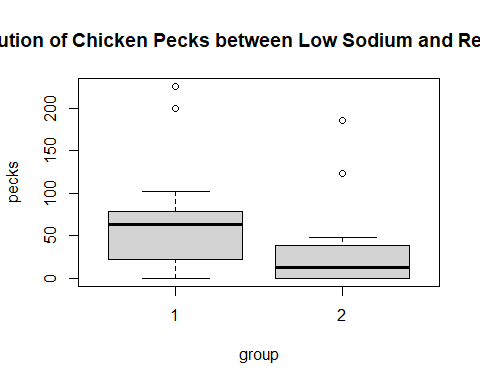
## Questions 1-16

You do not need to submit any work with these questions. Just answer them in D2L.

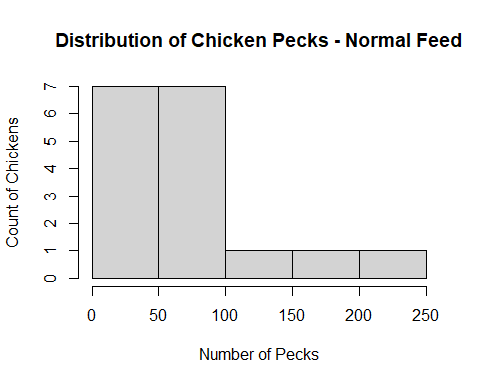
## Problem 1 - Questions 17-21

You’re going to analyze the data in BirdPecks.rda and answer the questions in D2l. Put all of your R in the chunk below:

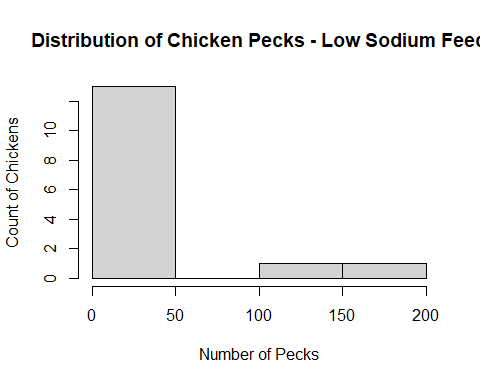
#load in the dataset  
library(DS705data)  
data("BirdPecks")  
  
 #create a boxplot of the number of pecks by feed group.  
boxplot(pecks~group,data = BirdPecks, main = "Distribution of Chicken Pecks between Low Sodium and Regular Feed")



#pull out the vector of pecks for low sodium feed  
group1 = BirdPecks$pecks[BirdPecks$group==1]  
 #pull out the vector of pecks for normal feed  
group2 = BirdPecks$pecks[BirdPecks$group==2]  
 #create a histogram of the chickens with low sodium feed  
hist(group1, main = "Distribution of Chicken Pecks - Normal Feed", xlab = "Number of Pecks", ylab = "Count of Chickens")



#create a histogram of the chickens with normal feed  
hist(group2, main = "Distribution of Chicken Pecks - Low Sodium Feed", xlab = "Number of Pecks", ylab = "Count of Chickens")



#test for normality in the low sodium feed  
shapiro.test(group1)

##   
## Shapiro-Wilk normality test  
##   
## data: group1  
## W = 0.82998, p-value = 0.00539

#test for normality in the normal feed  
shapiro.test(group2)

##   
## Shapiro-Wilk normality test  
##   
## data: group2  
## W = 0.69085, p-value = 0.0001987

#complete a test with intervals for shift in median pecks between low sodium feed and normal feed - 90% confidence level   
wilcox.test(group1,group2, data = BirdPecks,conf.level = 0.90, conf.int = TRUE)

## Warning in wilcox.test.default(group1, group2, data = BirdPecks, conf.level =  
## 0.9, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(group1, group2, data = BirdPecks, conf.level =  
## 0.9, : cannot compute exact confidence intervals with ties

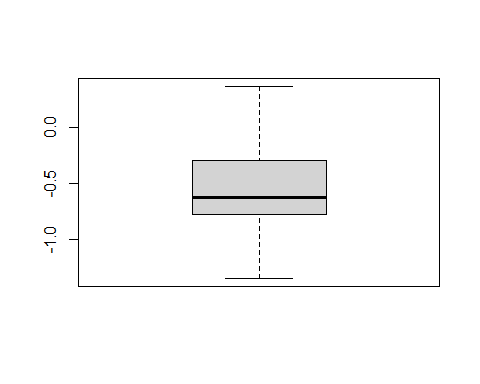
##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: group1 and group2  
## W = 179.5, p-value = 0.04996  
## alternative hypothesis: true location shift is not equal to 0  
## 90 percent confidence interval:  
## 1.999952 59.999911  
## sample estimates:  
## difference in location   
## 34.99995

#keeping things tidy  
rm(group1, group2)

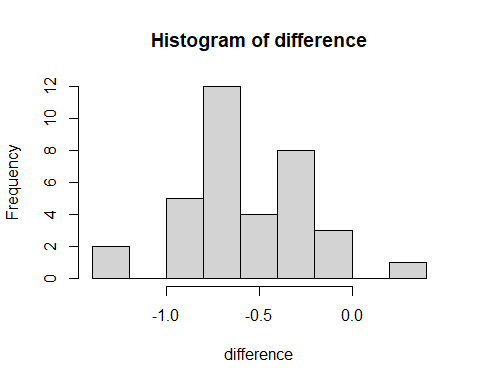
## Problem 2 - Questions 22-26

Analyze StrengthSpeed.rda and put your R below:

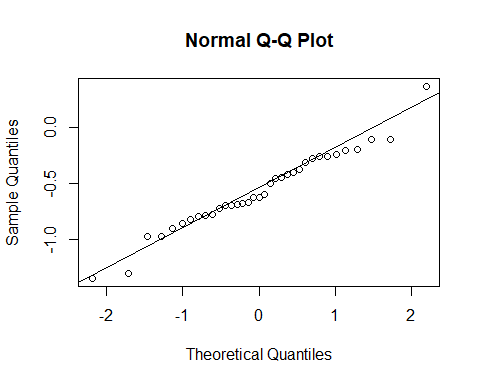
#load in the dataset  
library(DS705data)  
data("StrengthSpeed")  
 #calculating the difference between sprint times after the regiment, minus sprint times before regiment  
 #note: if the difference in speeds is NEGATIVE, then this means they performed the sprint faster.  
difference = StrengthSpeed$after - StrengthSpeed$before  
  
 #visually inspect the boxplot and histogram of the difference in sprint times - these normal normally distributed  
boxplot(difference)



hist(difference)



#generating a qqplot  
qqnorm(difference)  
qqline(difference)



#test for normality, cannot reject the null  
shapiro.test(difference)

##   
## Shapiro-Wilk normality test  
##   
## data: difference  
## W = 0.97539, p-value = 0.6065

#after reviewing the conditions - we will use the paired t-test  
 #we test that the after speed is less than the before speed (i.e. they've improved their sprint times)  
t.test(StrengthSpeed$after,StrengthSpeed$before, alternative = "less", paired = TRUE)

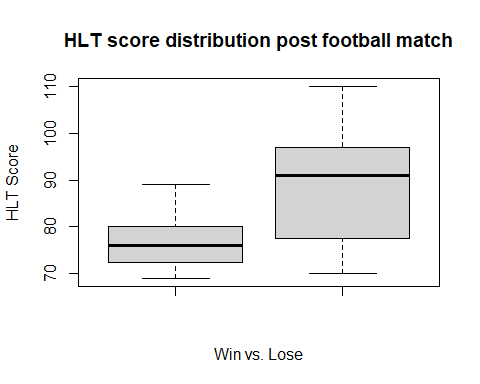
##   
## Paired t-test  
##   
## data: StrengthSpeed$after and StrengthSpeed$before  
## t = -9.6083, df = 34, p-value = 1.606e-11  
## alternative hypothesis: true difference in means is less than 0  
## 95 percent confidence interval:  
## -Inf -0.465004  
## sample estimates:  
## mean of the differences   
## -0.5643154

#keeping things tidy  
rm(difference)

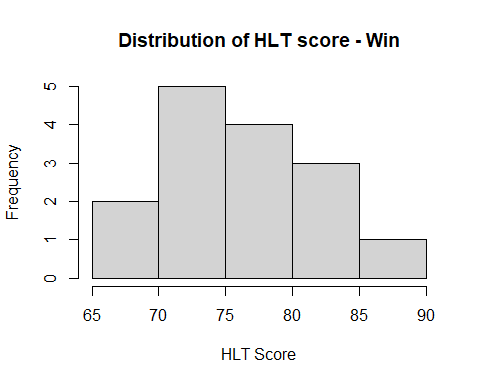
## Problem 3 - Questions 27-36

Analyze GroupHLT scores and put your R here:

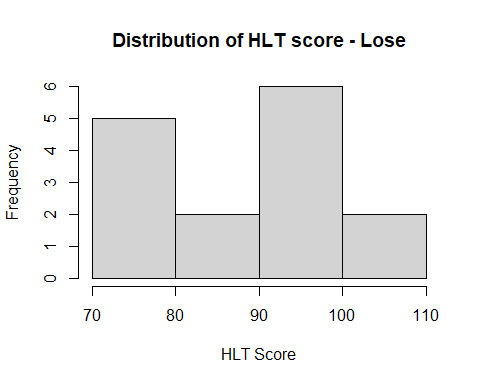
#storing the HLT score data into vectors  
win = c(79, 76, 74, 70, 81, 85, 73, 78, 69, 72, 83, 89, 72, 79, 75)  
lose = c(78, 96, 85, 91, 77,103, 72, 93, 98, 86, 70, 110, 70, 91, 99)  
  
 #creating a boxplot of HLT scores for win vs. lose  
boxplot(win, lose, main = "HLT score distribution post football match", xlab="Win vs. Lose", ylab="HLT Score")



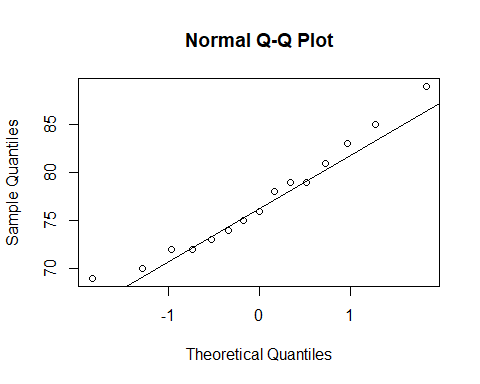
#creating histograms of both win and lose  
hist(win, main = "Distribution of HLT score - Win", xlab = "HLT Score")



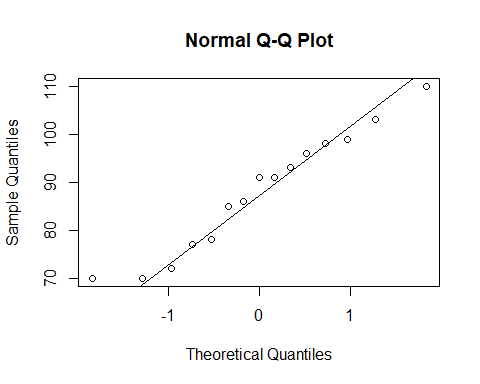
hist(lose, main = "Distribution of HLT score - Lose", xlab = "HLT Score")



#generating a qqplot  
qqnorm(win)  
qqline(win)



qqnorm(lose)  
qqline(lose)



#testing HLT score normality for win and lose.   
shapiro.test(win)

##   
## Shapiro-Wilk normality test  
##   
## data: win  
## W = 0.96154, p-value = 0.7191

shapiro.test(lose)

##   
## Shapiro-Wilk normality test  
##   
## data: lose  
## W = 0.95388, p-value = 0.5876

#complete the t.test to test if there is a difference in HLT scores between the win vs. lost population  
t.test(win,lose,alternative = "two.sided")

##   
## Welch Two Sample t-test  
##   
## data: win and lose  
## t = -3.0854, df = 19.691, p-value = 0.00591  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -18.332516 -3.534151  
## sample estimates:  
## mean of x mean of y   
## 77.00000 87.93333

#generate a confidence interval for the lose population.   
t.test(lose, conf.level = 0.95)$conf.int

## [1] 81.03098 94.83569  
## attr(,"conf.level")  
## [1] 0.95

#load the bootstrap package  
library(boot)  
 #create the auxiliary function to calculate the mean of the resamples.  
bootMean <- function(x,i){  
 return(mean(x[i]))  
}  
 #set seed for reproducability  
set.seed(123)  
 #complete 10000 bootstraps to create my sampling distribution for my parameter of interest (population mean for the "lose" group).  
boot.object = boot(lose, bootMean, R=10000)  
 #calculate the bca confidence interval  
boot.ci(boot.object, conf = 0.95, type = "bca")

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS  
## Based on 10000 bootstrap replicates  
##   
## CALL :   
## boot.ci(boot.out = boot.object, conf = 0.95, type = "bca")  
##   
## Intervals :   
## Level BCa   
## 95% (81.8, 93.8 )   
## Calculations and Intervals on Original Scale

#keeping things tidy  
rm(lose, win, bootMean)

## Problem 4 - Questions 37-41

Analyze the data in treadware.rda and put your R here:

library(DS705data)  
library(car)

## Loading required package: carData

##   
## Attaching package: 'car'

## The following object is masked from 'package:boot':  
##   
## logit

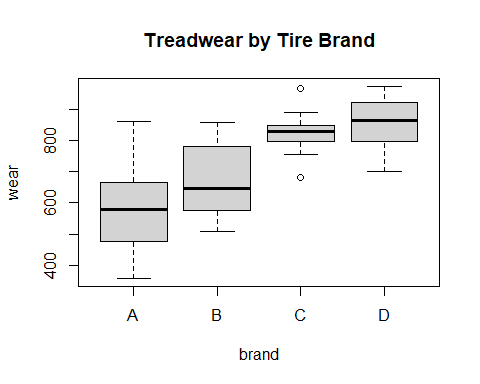
data("treadwear")  
 #apply the mean() function to each "brand's" tirewear  
tapply(treadwear$wear,treadwear$brand,mean)

## A B C D   
## 576.3017 671.1044 825.9870 853.2877

#apply the sd() function to each "brand's" tirewear  
tapply(treadwear$wear,treadwear$brand,sd)

## A B C D   
## 148.10367 111.29454 57.85366 81.23036

#create a boxplot to visualize distributions and variances of tirewear by brand.  
boxplot(wear~brand, data = treadwear, main = "Treadwear by Tire Brand")



#apply the shapiro.test() function to each "brand's" tirewear to check normality  
tapply(treadwear$wear,treadwear$brand,shapiro.test)

## $A  
##   
## Shapiro-Wilk normality test  
##   
## data: X[[i]]  
## W = 0.94655, p-value = 0.3177  
##   
##   
## $B  
##   
## Shapiro-Wilk normality test  
##   
## data: X[[i]]  
## W = 0.92027, p-value = 0.1003  
##   
##   
## $C  
##   
## Shapiro-Wilk normality test  
##   
## data: X[[i]]  
## W = 0.95389, p-value = 0.4301  
##   
##   
## $D  
##   
## Shapiro-Wilk normality test  
##   
## data: X[[i]]  
## W = 0.95129, p-value = 0.3871

#apply the levene test to check if variances are homogenous between sample groups. does not appear to be equal variance.  
leveneTest(wear~brand, data = treadwear)

## Levene's Test for Homogeneity of Variance (center = median)  
## Df F value Pr(>F)   
## group 3 3.996 0.01068 \*  
## 76   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#completing a "welch corrected" ANOVA. We select ANOVA because we have multiple populations.   
 #We select welch corrected (var.equal=FALSE) because the variances do not appear to be equal after looking at the samples.   
oneway.test(wear~brand, data = treadwear, var.equal = FALSE)

##   
## One-way analysis of means (not assuming equal variances)  
##   
## data: wear and brand  
## F = 27.201, num df = 3.000, denom df = 40.197, p-value = 8.988e-10

#use the custom function from the DS705Data package, complete a Games-Howell test.  
 #var.equals = FALSE means this is games-howell vs. the Tukey test.  
 #adjust = one.step makes this a games-howell or tukey test.  
onewayComp(wear~brand, data = treadwear, var.equal = FALSE, adjust = "one.step")$comp[,c(2,3,6,7)]

## lwr upr p adj rej H\_0  
## B-A -16.87538 206.48079 1.097707e-01 0  
## C-A 151.79715 347.57345 4.666289e-09 1  
## D-A 174.17864 379.79344 1.139764e-09 1  
## C-B 78.39766 231.36753 2.652978e-06 1  
## D-B 99.06516 265.30152 5.337504e-07 1  
## D-C -32.89664 87.49813 6.134985e-01 0

## Problem 5 - Questions 42-59

Analyze the data in diamond.rda and include your R here:

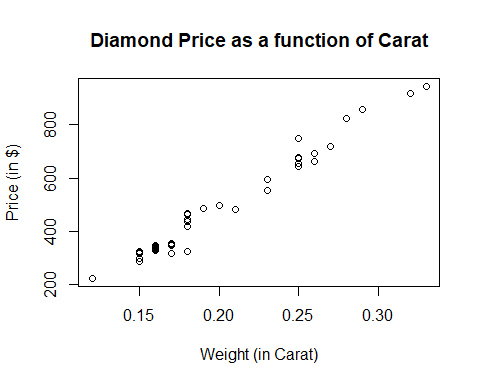
#load the diamond dataset and necessary libraries  
library(DS705data)  
library(lmtest)

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

data("diamond")  
  
 #creating a scatterplot to visualize the relationship between price and carat.   
plot(price~carat, data = diamond, main = "Diamond Price as a function of Carat", xlab = "Weight (in Carat)", ylab = "Price (in $)")



#create a linear regression model with price as a function of carat weight  
model = lm(price~carat, data = diamond)  
summary(model)

##   
## Call:  
## lm(formula = price ~ carat, data = diamond)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -85.283 -20.639 -0.855 15.359 82.719   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -250.57 18.12 -13.83 <2e-16 \*\*\*  
## carat 3671.40 87.17 42.12 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 31.41 on 45 degrees of freedom  
## Multiple R-squared: 0.9753, Adjusted R-squared: 0.9747   
## F-statistic: 1774 on 1 and 45 DF, p-value: < 2.2e-16

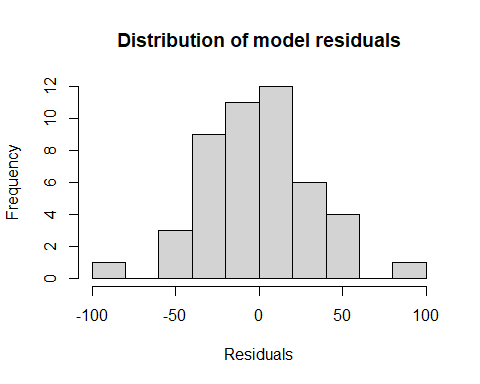
#test the correlation coefficient between price and carat  
with(diamond, cor.test(price, carat, conf.level = 0.99))

##   
## Pearson's product-moment correlation  
##   
## data: price and carat  
## t = 42.116, df = 45, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 99 percent confidence interval:  
## 0.9731306 0.9942549  
## sample estimates:  
## cor   
## 0.9875512

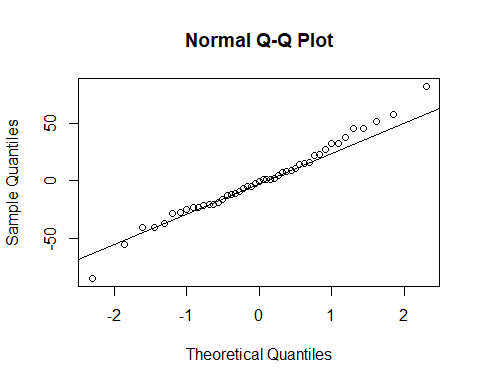
#calculating the 95% CI for the regression model.   
confint(model)

## 2.5 % 97.5 %  
## (Intercept) -287.0568 -214.0793  
## carat 3495.8184 3846.9750

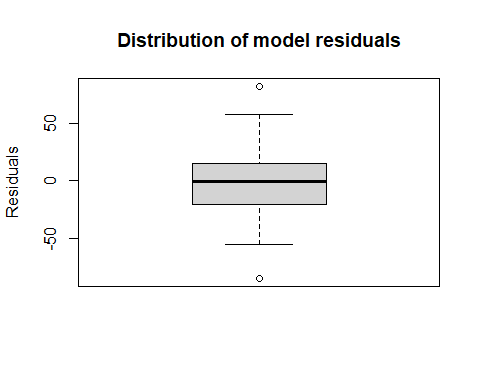
#create a histogram of the residuals  
hist(model$residuals, main = "Distribution of model residuals", xlab = "Residuals")



#create a QQ plot of the residuals  
qqnorm(model$residuals)  
qqline(model$residuals)



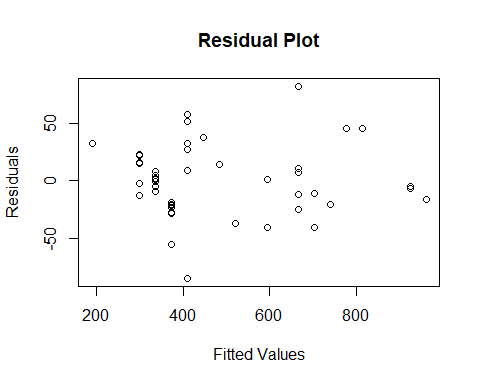
#create a boxplot of the residuals  
boxplot(model$residuals, main = "Distribution of model residuals", ylab = "Residuals")



#test residuals for normality.  
shapiro.test(model$residuals)

##   
## Shapiro-Wilk normality test  
##   
## data: model$residuals  
## W = 0.98604, p-value = 0.8406

#plot the residuals vs. fitted values to visualize variance.   
plot(model$fitted.values, model$residuals, main = "Residual Plot", ylab = "Residuals", xlab = "Fitted Values")



#perform the BP test to check for equal variance.   
bptest(model)

##   
## studentized Breusch-Pagan test  
##   
## data: model  
## BP = 0.18208, df = 1, p-value = 0.6696

#calculate the R^2 (coefficient of determination). This is also calculated directly from the summary() command as well.  
(cor.test(diamond$price,diamond$carat, method = "pearson")$estimate)^2

## cor   
## 0.9752573

#keeping things tidy  
rm(model)

## Problem 6 - Questions 60- 66

Analyze the data in Shells.rda and include your R here:

#loading necessary libraries and dataset.  
library(DS705data)  
library(HH)

## Loading required package: lattice

##   
## Attaching package: 'lattice'

## The following object is masked from 'package:boot':  
##   
## melanoma

## Loading required package: grid

## Loading required package: latticeExtra

## Loading required package: multcomp

## Loading required package: mvtnorm

## Loading required package: survival

##   
## Attaching package: 'survival'

## The following object is masked from 'package:boot':  
##   
## aml

## Loading required package: TH.data

## Loading required package: MASS

##   
## Attaching package: 'TH.data'

## The following object is masked from 'package:MASS':  
##   
## geyser

## Loading required package: gridExtra

## Warning: package 'gridExtra' was built under R version 4.0.2

##   
## Attaching package: 'HH'

## The following objects are masked from 'package:car':  
##   
## logit, vif

## The following object is masked from 'package:boot':  
##   
## logit

data("shells")  
  
 #starting with a full model, do stepwise comparison using the "both" direction.   
 #AIC is -121.99  
model = step(lm(Y~., data=shells), direction = "both", trace = FALSE)  
 #checking the results of the model  
summary(model)

##   
## Call:  
## lm(formula = Y ~ X1 + X2 + X4 + X6 + X7, data = shells)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.200151 -0.080277 -0.007467 0.081058 0.173162   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.42718 0.73427 1.944 0.06375 .   
## X1 1.15023 0.20467 5.620 8.73e-06 \*\*\*  
## X2 0.65317 0.19465 3.356 0.00263 \*\*   
## X4 0.60923 0.21010 2.900 0.00787 \*\*   
## X6 -0.06487 0.01735 -3.738 0.00102 \*\*   
## X7 0.02636 0.01227 2.148 0.04205 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1198 on 24 degrees of freedom  
## Multiple R-squared: 0.9514, Adjusted R-squared: 0.9413   
## F-statistic: 94.05 on 5 and 24 DF, p-value: 5.821e-15

#checking the AIC of the model  
extractAIC(model)

## [1] 6.0000 -121.9933

#checking for collinearity between predictor variables.   
vif(model)

## X1 X2 X4 X6 X7   
## 4.286121 3.804041 2.697711 4.428297 3.172575

#creating a 2nd order term using X2.  
X2\_sqrd = shells$X2^2  
 #fit the second model that uses a second order term  
modelB = lm(Y~X1 + X2 + X2\_sqrd + X4 + X6, data = shells)  
 #check results of the modelB  
summary(modelB)

##   
## Call:  
## lm(formula = Y ~ X1 + X2 + X2\_sqrd + X4 + X6, data = shells)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.17500 -0.05247 -0.00598 0.05188 0.21059   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 9.16461 2.31597 3.957 0.000587 \*\*\*  
## X1 1.11286 0.17872 6.227 1.96e-06 \*\*\*  
## X2 -5.45450 1.70283 -3.203 0.003811 \*\*   
## X2\_sqrd 1.14091 0.31981 3.567 0.001559 \*\*   
## X4 0.79526 0.16439 4.838 6.28e-05 \*\*\*  
## X6 -0.03961 0.01106 -3.581 0.001507 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1058 on 24 degrees of freedom  
## Multiple R-squared: 0.9622, Adjusted R-squared: 0.9543   
## F-statistic: 122.1 on 5 and 24 DF, p-value: 2.955e-16

#checking the AIC of the modelB  
extractAIC(modelB)

## [1] 6.0000 -129.4831

### adjusted R^2 is within the summary statement....  
  
 #keeping things tidy  
rm(model,modelB, X2\_sqrd)

## Problem 7 - Questions 67-70

Analyze the “Primary News Source for Americans” data described in the problem statement. Put your R below:

#loading the data into R  
observed = c(38,20,15,42)  
proportion = c(0.45,0.18,0.16,0.21)  
 #completing the chi-sq goodness of fit test to test the population proportions against stated proportions  
test = chisq.test(x=observed,p=proportion); test

##   
## Chi-squared test for given probabilities  
##   
## data: observed  
## X-squared = 17.499, df = 3, p-value = 0.000558

#double checking expected values to ensure there is no category below 5. There are none.  
test$expected

## [1] 51.75 20.70 18.40 24.15

#keeping things tidy  
rm(test, observed, proportion)

## Problem 8 - Questions 71-75

Analyze the data in cheese.rda and put your R below:

#load the required library and dataset.  
library(DS705data)  
data("cheese")  
  
 #create a new factor variable that puts not acceptable as the 0, and acceptable as 1 (since alphabetically, it was mismatched)  
cheese$taste2 = ifelse(cheese$taste=="Not Acceptable",0,1)  
  
 #create a logistic regression model with taste2 as a function of acetic and person  
model = glm(taste2~acetic + person, data = cheese, family = "binomial")  
summary(model)

##   
## Call:  
## glm(formula = taste2 ~ acetic + person, family = "binomial",   
## data = cheese)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.2245 -0.4998 -0.2002 0.3040 1.6066   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -18.709 8.517 -2.197 0.0280 \*  
## acetic 2.787 1.412 1.975 0.0483 \*  
## personAdult 3.096 1.371 2.258 0.0239 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 34.795 on 29 degrees of freedom  
## Residual deviance: 20.550 on 27 degrees of freedom  
## AIC: 26.55  
##   
## Number of Fisher Scoring iterations: 6

#convert the logit number to an odds ratio number.   
exp(model$coefficients[2])

## acetic   
## 16.23497

#using the model, predict the probability an "acceptable" response.  
newdata = data.frame(acetic=6,person="Child")  
predict(model,newdata,type = "response")

## 1   
## 0.1206732

#using the standard error, we can create a confidence interval for this prediction.  
out = predict(model,newdata,se.fit = TRUE)  
 #CI = 95%  
C = 0.95  
crit = qnorm(1-(1-C)/2)  
  
 #create the CI bounds and display.  
lower = exp(out$fit-crit\*out$se.fit)/(1+exp(out$fit-crit\*out$se.fit))  
upper = exp(out$fit+crit\*out$se.fit)/(1+exp(out$fit+crit\*out$se.fit))  
c(lower,upper)

## 1 1   
## 0.01591904 0.53793967

#keeping things tidy  
rm(model,out,C,crit,lower,upper,newdata)

## Problem 9 - Questions 76-90

Analyze the data in careerbarrier.rda and put your R below:

#loading the required libraries and the dataset  
library(DS705data)  
library(psych)

## Warning: package 'psych' was built under R version 4.0.2

##   
## Attaching package: 'psych'

## The following object is masked from 'package:HH':  
##   
## logit

## The following object is masked from 'package:car':  
##   
## logit

## The following object is masked from 'package:boot':  
##   
## logit

data("careerbarrier")  
 #creating a correlation matrix using the variables from the dataset  
matrx = cor(careerbarrier)  
 #testing if the correlation matrix is the identity matrix, which would indicate factor analysis is not useful here.  
cortest.bartlett(matrx,n=76)

## $chisq  
## [1] 287.4985  
##   
## $p.value  
## [1] 6.756016e-19  
##   
## $df  
## [1] 105

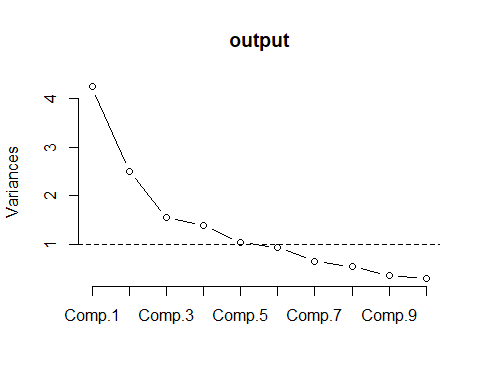
#now that factor analysis shows promise, conducting KMO() analysis to calculate the MSA score to see overall score, but also if there are any variables that stand on their own and dropped from the factor analysis.  
KMO(matrx)

## Kaiser-Meyer-Olkin factor adequacy  
## Call: KMO(r = matrx)  
## Overall MSA = 0.68  
## MSA for each item =   
## money lazy law noexp math support health reading   
## 0.36 0.67 0.67 0.61 0.67 0.66 0.67 0.73   
## english aoda grades disc social relatshp looks   
## 0.69 0.82 0.67 0.66 0.75 0.72 0.58

#"money" has a MSA score of <0.50, so we drop it and perform the KMO() analysis again.  
matrx2 = cor(careerbarrier[,-1])  
KMO(matrx2)

## Kaiser-Meyer-Olkin factor adequacy  
## Call: KMO(r = matrx2)  
## Overall MSA = 0.7  
## MSA for each item =   
## lazy law noexp math support health reading english   
## 0.68 0.67 0.61 0.68 0.66 0.68 0.74 0.70   
## aoda grades disc social relatshp looks   
## 0.82 0.67 0.70 0.77 0.71 0.62

#create a scree plot to check "knee" (where the line starts to even out - 3), and number of eigenvalues greater than 1 (the dotted line) - 5.  
output = princomp(matrx2, cor = TRUE)  
plot(output, type = "lines")  
abline(h=1, lty=2)



#perform the PCA using the varimax rotation. Suppress factor loadings under 0.50, and sort them by the most variance first.   
fa.out = principal(matrx2,nfactors = 5, rotate = "varimax")  
print.psych(fa.out,cut = 0.5, sort = TRUE)

## Principal Components Analysis  
## Call: principal(r = matrx2, nfactors = 5, rotate = "varimax")  
## Standardized loadings (pattern matrix) based upon correlation matrix  
## item RC2 RC1 RC3 RC4 RC5 h2 u2 com  
## looks 14 0.79 0.67 0.33 1.2  
## relatshp 13 0.76 0.65 0.35 1.2  
## disc 11 0.71 0.53 0.47 1.1  
## social 12 0.59 0.50 0.50 1.9  
## reading 7 0.82 0.78 0.22 1.3  
## english 8 0.74 0.67 0.33 1.4  
## health 6 0.63 0.59 0.41 2.0  
## aoda 9 0.58 0.59 0.41 2.2  
## lazy 1 0.81 0.77 0.23 1.4  
## law 2 0.76 0.65 0.35 1.3  
## grades 10 0.65 0.53 0.47 1.5  
## support 5 0.82 0.71 0.29 1.1  
## math 4 0.64 0.60 0.40 2.0  
## noexp 3 0.89 0.85 0.15 1.1  
##   
## RC2 RC1 RC3 RC4 RC5  
## SS loadings 2.28 2.28 2.11 1.27 1.15  
## Proportion Var 0.16 0.16 0.15 0.09 0.08  
## Cumulative Var 0.16 0.33 0.48 0.57 0.65  
## Proportion Explained 0.25 0.25 0.23 0.14 0.13  
## Cumulative Proportion 0.25 0.50 0.73 0.87 1.00  
##   
## Mean item complexity = 1.5  
## Test of the hypothesis that 5 components are sufficient.  
##   
## The root mean square of the residuals (RMSR) is 0.08   
##   
## Fit based upon off diagonal values = 0.89

#keeping things tidy  
rm(matrx,matrx2,output,fa.out)

## Problem 10 - Questions 91-98

Analyze the data on seat postion vs. nausea described in the problem. Put your R below:

#loading necessary packages  
library(mosaic)

## Loading required package: dplyr

##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:gridExtra':  
##   
## combine

## The following object is masked from 'package:MASS':  
##   
## select

## The following object is masked from 'package:car':  
##   
## recode

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

## Loading required package: ggformula

## Loading required package: ggplot2

##   
## Attaching package: 'ggplot2'

## The following objects are masked from 'package:psych':  
##   
## %+%, alpha

## The following object is masked from 'package:latticeExtra':  
##   
## layer

## Loading required package: ggstance

##   
## Attaching package: 'ggstance'

## The following objects are masked from 'package:ggplot2':  
##   
## geom\_errorbarh, GeomErrorbarh

##   
## New to ggformula? Try the tutorials:   
## learnr::run\_tutorial("introduction", package = "ggformula")  
## learnr::run\_tutorial("refining", package = "ggformula")

## Loading required package: mosaicData

## Loading required package: Matrix

## Registered S3 method overwritten by 'mosaic':  
## method from   
## fortify.SpatialPolygonsDataFrame ggplot2

##   
## The 'mosaic' package masks several functions from core packages in order to add   
## additional features. The original behavior of these functions should not be affected by this.  
##   
## Note: If you use the Matrix package, be sure to load it BEFORE loading mosaic.  
##   
## Have you tried the ggformula package for your plots?

##   
## Attaching package: 'mosaic'

## The following object is masked from 'package:Matrix':  
##   
## mean

## The following object is masked from 'package:ggplot2':  
##   
## stat

## The following objects are masked from 'package:dplyr':  
##   
## count, do, tally

## The following objects are masked from 'package:psych':  
##   
## logit, rescale

## The following object is masked from 'package:HH':  
##   
## logit

## The following objects are masked from 'package:car':  
##   
## deltaMethod, logit

## The following object is masked from 'package:boot':  
##   
## logit

## The following objects are masked from 'package:stats':  
##   
## binom.test, cor, cor.test, cov, fivenum, IQR, median, prop.test,  
## quantile, sd, t.test, var

## The following objects are masked from 'package:base':  
##   
## max, mean, min, prod, range, sample, sum

#create a matrix to store the data within. first row is nausea, second row is no nausea.   
bus = matrix(c(98,110,161,264,321,280),nrow = 2, byrow=TRUE)  
 #label the columns and rows accordingly.  
colnames(bus) = c("Front", "Middle", "Rear")  
rownames(bus) = c("Nausea", "No Nausea")  
 #add sum margins to the table so we can perform the chisq test for independence.  
addmargins(bus)

## Front Middle Rear Sum  
## Nausea 98 110 161 369  
## No Nausea 264 321 280 865  
## Sum 362 431 441 1234

#perform the chisq test of independence.  
result = chisq.test(bus); result

##   
## Pearson's Chi-squared test  
##   
## data: bus  
## X-squared = 14.509, df = 2, p-value = 0.000707

#double checking there are no expected counts less than 5, otherwise we'd need to use fisher exact test.   
result$expected

## Front Middle Rear  
## Nausea 108.248 128.8809 131.8712  
## No Nausea 253.752 302.1191 309.1288

#setting up the difference in front vs back. for nausea and no nausea   
diff\_nausea = abs(bus[1,1] - bus[1,3])  
diff\_no\_nausea = abs(bus[2,1] - bus[2,3])  
diff = c(diff\_nausea,diff\_no\_nausea)  
 #pulling totals in own vector  
total = c(369,865)  
  
 #performing the prop test in differences between front vs. back for no nausea vs. nausea respectively.   
prop.test(diff,total, correct = FALSE, conf.level = 0.90)

##   
## 2-sample test for equality of proportions without continuity  
## correction  
##   
## data: diff out of total  
## X-squared = 100.04, df = 1, p-value < 2.2e-16  
## alternative hypothesis: two.sided  
## 90 percent confidence interval:  
## 0.1191456 0.1853236  
## sample estimates:  
## prop 1 prop 2   
## 0.17073171 0.01849711

#create a matrix to store the data within. first row is nausea, second row is no nausea.   
bus2 = matrix(c(98,161,264,280),nrow = 2, byrow=TRUE)  
 #label the columns and rows accordingly.  
colnames(bus2) = c("Rear", "Front")  
rownames(bus2) = c("Nausea", "No Nausea")  
 #add sum margins to the table so we can calculate the odds ratio.  
addmargins(bus2)

## Rear Front Sum  
## Nausea 98 161 259  
## No Nausea 264 280 544  
## Sum 362 441 803

#compute the odds ratio using the mosaic package.   
oddsRatio(bus2, verbose = TRUE)

##   
## Odds Ratio  
##   
## Proportions  
## Prop. 1: 0.3784   
## Prop. 2: 0.4853   
## Rel. Risk: 1.283   
##   
## Odds  
## Odds 1: 0.6087   
## Odds 2: 0.9429   
## Odds Ratio: 1.549   
##   
## 95 percent confidence interval:  
## 1.073 < RR < 1.533   
## 1.145 < OR < 2.096   
## NULL

## [1] 1.54898

#keeping things tidy  
rm(diff,diff\_nausea, diff\_no\_nausea,total,bus,bus2,result)

## Question 99

Make sure both this RMD and the resulting knitted Word document are uploaded to the Dropbox “Final Exam R Code.”