```
################################################################################

# Brian Weinstein - bmw2148
# STAT W4201 001
# Homework 8
# 2016-04-06

# set working directory
setwd("~/Documents/advanced-data-analysis/homework_08")

# prevent R from printing large numbers in scientific notation
options(scipen=5)

# load packages
library(Sleuth3) # Data sets from Ramsey and Schafer's "Statistical Sleuth
(3rd ed)"
library(MASS)
library(dplyr)
library(ggplot2); theme_set(theme_bw())
library(GGally)
library(leaps)
library(tidyr)
# library(ggrepel)
# library(gridExtra)



# Problem 1: Ramsey 12.17
#########################################################################

# load data
pollutionData <- Sleuth3::ex1217

# Part a ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ###

# plot a matrix of pairwise scatterplots
plot.pairs <- ggpairs(data=select(pollutionData, Precip:Poor, Mortality),
                      lower=list(continuous=wrap("points", size=0.5))) +
theme_bw(base_size = 0.1)
plot.pairs
png(filename="writeup/1a_pairs.png", width=11, height=11, units="in", res=300)
print(plot.pairs)
dev.off()

# create a dataset with the relevant and transformed explanatory vars
regDataA <- pollutionData %>%
  select(Mortality:Poor) %>%
  mutate(JanTemp2 = JanTemp^2,
         JulyTemp2 = JulyTemp^2,
         Over652 = Over65^2,
         House2 = House^2)

# perform least squares regression on all subsets of variables
```

```
allModels <- regsubsets(x=Mortality ~ ., data=regDataA,
                        nbest=40, nvmax=16,
                        intercept=TRUE, method="exhaustive")
names(summary(allModels))

# create a dataframe summarizing each model
allModelsOutput <- as.data.frame(summary(allModels)$which, row.names = FALSE)
%>%
  mutate(model_id = row_number(),
         cp = summary(allModels)$cp,
         bic = summary(allModels)$bic)
allModelsOutput$num_params <- rowSums(select(allModelsOutput, Precip:House2))

# filter out the models that include a squared term, but don't include
# the associated linear term
allModelsOutput <- allModelsOutput %>%
  filter(JanTemp2==FALSE | JanTemp2==JanTemp) %>%
  filter(JulyTemp2==FALSE | JulyTemp2==JulyTemp) %>%
  filter(Over652==FALSE | Over652==Over65) %>%
  filter(House2==FALSE | House2==House)

# create a cp plot
ggplot(allModelsOutput, aes(x=num_params, y=cp)) +
  geom_point() +
  geom_abline(intercept=0, slope=17/16, linetype="dashed") +
  ylim(0, 20) +
  labs(x="Number of Model Parameters", y="Cp Statistic")
ggsave(filename="writeup/1a_cp.png", width=6.125, height=3.5, units="in")

# create a linear model with the best subset
lmBestSubset <- lm(formula = Mortality ~ Precip + JanTemp +
                     JulyTemp + Educ + Density + NonWhite,
                   data = pollutionData)
summary(lmBestSubset)

# create a linear model with the best subset, plus the 3 pollution vars
lmBestSubsetPoll <- lm(formula = Mortality ~ Precip + JanTemp +
                         JulyTemp + Educ + Density + NonWhite +
                         log(HC) + log(NOX) + log(SO2),
                       data = pollutionData)
summary(lmBestSubsetPoll)

# perform extra sum of squares F test
anova(lmBestSubsetPoll, lmBestSubset)

# Part b ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ###

# create a dataset with the relevant explanatory vars
regDataB <- pollutionData %>%
  select(Mortality:Poor)

# perform forward-selection least squares regression
allModels_forward <- stepAIC(object = lm(formula = Mortality ~ 1,
data=regDataB),
                             scope = list(upper = lm(formula = Mortality ~ .,
```

```
data=regDataB),
                                          lower = lm(formula = Mortality ~ 1,
data=regDataB)),
                            direction = "forward",
                            trace = FALSE)
allModels_forward$anova
allModels_forward$anova

# create a linear model with the forward selection subset
lmForwardSubset <- lm(formula = Mortality ~ NonWhite + Educ + JanTemp +
                        House + JulyTemp + Precip + Density,
                      data = pollutionData)
summary(lmForwardSubset)

# create a linear model with the forward selection subset, plus the 3
pollution vars
lmForwardSubsetPoll <- lm(formula = Mortality ~ NonWhite + Educ + JanTemp +
                            House + JulyTemp + Precip + Density +
                            log(HC) + log(NOX) + log(SO2),
                          data = pollutionData)
summary(lmForwardSubsetPoll)

# perform extra sum of squares F test
anova(lmForwardSubsetPoll, lmForwardSubset)

rm(list = ls()) # clear working environment



# Problem 2: Ramsey 12.20
#########################################################################

# load data
galapData <- Sleuth3::ex1220

# define Nonnative species, remove Total variable
galapData <- galapData %>%
  mutate(Nonnative=Total-Native) %>%
  select(Island, Native, Nonnative, Area:AreaNear)

# plot a matrix of pairwise scatterplots
plot.pairs <- ggpairs(data=select(galapData, Area:AreaNear, Nonnative,
Native),
                      lower=list(continuous=wrap("points", size=0.5)))
plot.pairs

# log transform the necessary variables
galapDataTransf <- galapData %>%
  mutate(LogArea=log(Area), LogElev=log(Elev), LogDistNear=log(DistNear),
         LogDistSc=log(1 + DistSc), LogAreaNear=log(AreaNear)) %>%
  select(Island, Native, Nonnative, LogArea, LogElev, LogDistNear, LogDistSc,
LogAreaNear)

# plot a matrix of transformed pairwise scatterplots
plot.pairs.transf <- ggpairs(data=select(galapDataTransf, LogArea:LogAreaNear,
```

```
Nonnative, Native),
                                lower=list(continuous=wrap("points", size=0.5)))
plot.pairs.transf
png(filename="writeup/2_pairs_tranf.png", width=11, height=9, units="in",
res=300)
print(plot.pairs.transf)
dev.off()

# Part a ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ###

# create a linear model with response=Native
lmNative <- lm(formula = Native ~ LogArea + LogElev +
                 LogDistNear + LogDistSc + LogAreaNear,
               data = galapDataTransf)
summary(lmNative)$coefficients

# test for any influential observations
galapDataTransf <- galapDataTransf %>%
  mutate(cdLmNative=cooks.distance(lmNative))
ggplot(galapDataTransf, aes(x=Island, y=cdLmNative)) +
  geom_point() +
  geom_hline(yintercept=1, linetype="dotted") +
  labs(x="Island", y="Cook's Distance") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
ggsave(filename="writeup/2a_cd.png", width=6.125, height=3.5, units="in")

# Part b ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ###

# create a linear model with response=Nonnative
lmNonnative <- lm(formula = Nonnative ~ LogArea + LogElev +
                    LogDistNear + LogDistSc + LogAreaNear,
                  data = galapDataTransf)
summary(lmNonnative)$coefficients

# test for any influential observations
galapDataTransf <- galapDataTransf %>%
  mutate(cdLmNonnative=cooks.distance(lmNonnative))
ggplot(galapDataTransf, aes(x=Island, y=cdLmNonnative)) +
  geom_point() +
  geom_hline(yintercept=1, linetype="dotted") +
  labs(x="Island", y="Cook's Distance") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
ggsave(filename="writeup/2b_cd.png", width=6.125, height=3.5, units="in")

rm(list = ls()) # clear working environment



# Problem 3: Ramsey 20.11
#########################################################################

# load data
shuttleData <- Sleuth3::ex2011 %>%
  mutate(Failure=ifelse(Failure=="Yes", 1, 0))
```

```
# Part a ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ###

# fit a logistic regression model
glmShuttle <- glm(formula = Failure ~ Temperature,
                  data = shuttleData, family = binomial)
summary(glmShuttle)$coefficients

# Part b ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ###

# calcualte the Z statistic and associated pvalue
tempEst <- (summary(glmShuttle)$coefficients)["Temperature", "Estimate"]
tempSe <- (summary(glmShuttle)$coefficients)["Temperature", "Std. Error"]
tempZstat <- (tempEst - 0)/tempSe
tempPval <- pnorm(q = -1 * abs(tempZstat), mean = 0, sd = 1) ; tempPval # one-
sided

# Part c ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ###

# fit a reduced model
glmShuttleReduced <- glm(formula = Failure ~ 1,
                         data = shuttleData, family = binomial)

# calculate the likelihood ratio test statistic and associated pvalue
lrtStat <- summary(glmShuttleReduced)$deviance - summary(glmShuttle)$deviance
lrtDf <- summary(glmShuttleReduced)$df[2] - summary(glmShuttle)$df[2]
lrtPval <- pchisq(q = lrtStat, df = lrtDf, lower.tail = FALSE); lrtPval

# verify lrtPval using anova function
anova(glmShuttle, glmShuttleReduced, test="LRT")

# Part d ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ###

# compute a 95% CI for the Temperature estimate
tempEst
tempEst + c(-1, 1) * (qnorm(p = 0.975, mean = 0, sd = 1) * tempSe)

# Part e ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ###

# estimated logit of failure probability at Temperature=31
logitPi <- predict(glmShuttle, data.frame(Temperature=31)) ; logitPi

# estimated probability of failure probability at Temperature=31
exp(logitPi) / (1 + exp(logitPi))

rm(list = ls()) # clear working environment



# Problem 4: Ramsey 20.15
######################################################################

# load data
owlData <- Sleuth3::ex2015

# convert to long format
```

```
owlData2 <- owlData %>%
  gather(key="Ring", value="PctMature", PctRing1:PctRing7) %>%
  mutate(Ring=factor(Ring))

# create a column indicating the radius associated with each Ring level
dict <- data.frame(Ring=c("PctRing1", "PctRing2", "PctRing3", "PctRing4",
"PctRing5", "PctRing6", "PctRing7"),
                   Radius=c(0.91, 1.18, 1.40, 1.60, 1.77, 2.41, 3.38))
owlData2 <- left_join(owlData2, dict, by="Ring")
rm(dict)

# Part a ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ###

# create boxplots
ggplot(owlData2, aes(x=Site, y=PctMature)) +
  geom_violin(alpha=0.15) +
  geom_boxplot()
ggsave(filename="writeup/4a.png", width=6.125, height=3.5, units="in")

# compututе the estimated difference, 1-sided p-value, and 95% CI
tt <- t.test(formula=PctMature~Site, data=owlData2,
             var.equal=TRUE, conf.level=0.95)
diff(tt$estimate)[[1]] * -1
tt$p.value / 2
tt$conf.int

# Part b ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ###

ggplot(owlData2, aes(x=Ring, y=PctMature)) +
  geom_jitter(aes(color=Site, shape=Site), width=0.3, size=2)
ggsave(filename="writeup/4b_jitter.png", width=6.125, height=3.5, units="in")

ggplot(owlData2, aes(x=Site, y=PctMature)) +
  geom_violin(alpha=0.15) +
  geom_boxplot() +
  facet_grid(facets=.~Ring) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(x=NULL)
ggsave(filename="writeup/4b_boxplots.png", width=6.125, height=3.5,
units="in")


# create a nest site indicator variable
owlData <- owlData %>%
  mutate(Nest=ifelse(Site=="Nest", 1, 0))

# fit the 7 possible logistic regression models
glm7 <- glm(formula = Nest ~ .,
            data = select(owlData, Nest, PctRing1:PctRing7),
            family = binomial)
glm6 <- glm(formula = Nest ~ .,
            data = select(owlData, Nest, PctRing1:PctRing6),
            family = binomial)
glm5 <- glm(formula = Nest ~ .,
            data = select(owlData, Nest, PctRing1:PctRing5),
```

```
            family = binomial)
glm4 <- glm(formula = Nest ~ .,
            data = select(owlData, Nest, PctRing1:PctRing4),
            family = binomial)
glm3 <- glm(formula = Nest ~ .,
            data = select(owlData, Nest, PctRing1:PctRing3),
            family = binomial)
glm2 <- glm(formula = Nest ~ .,
            data = select(owlData, Nest, PctRing1:PctRing2),
            family = binomial)
glm1 <- glm(formula = Nest ~ .,
            data = select(owlData, Nest, PctRing1:PctRing1),
            family = binomial)

# perform the likelihood ratio test (drop in deviance test), get pvalues
anova(glm7, glm6, test="LRT")$"Pr(>Chi)"[[2]]
anova(glm7, glm5, test="LRT")$"Pr(>Chi)"[[2]]
anova(glm7, glm4, test="LRT")$"Pr(>Chi)"[[2]]
anova(glm7, glm3, test="LRT")$"Pr(>Chi)"[[2]]
anova(glm7, glm2, test="LRT")$"Pr(>Chi)"[[2]]
anova(glm7, glm1, test="LRT")$"Pr(>Chi)"[[2]]

# use the model with PctRing1 through PctRing5
summary(glm5)$coefficients

rm(list = ls()) # clear working environment



# Problem 5
#########################################################################

# no code needed



# Problem 6
#########################################################################

# Part d ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ###

# input data
data6 <- data.frame(Y=c(0, 0, 1, 1), X=c(-2, -1, 1, 2))

# fit a generalized linear model
glm6 <- glm(formula = Y ~ X - 1, data = data6, family = binomial) ; glm6

# plot the logistic fit
ggplot(data6, aes(x = X, y = Y)) +
  geom_point(size = 2) +
  geom_smooth(method = "glm", se=FALSE,
              method.args = list(family = "binomial"))
ggsave(filename="writeup/6d_logistic.png", width=6.125, height=3.5,
units="in")
```

```
# define function to calculate the probability for a given outcome
piI <- function(beta1, X){
  exp(beta1 * X) / (1 + exp(beta1 * X))
}

# define function to calculate the log likelihood of a set of outcomes
logLikelihood <- function(beta1){
  log(
    prod(
      piI(beta1, data6$X)^(data6$Y) * (1 - piI(beta1, data6$X))^(1-(data6$Y))
    )
  )
}

# compute the logLIkelihood for various choices of beta1
llTest <-data.frame(beta1=seq(-15, 50, 1)) %>%
  mutate(logLikelihood=sapply(beta1, function(beta1){logLikelihood(beta1)}))

# plot the logLIkelihood as a function of beta1
ggplot(llTest, aes(x=beta1, y=logLikelihood)) +
  geom_point(size=0.75) +
  geom_line() +
  geom_vline(xintercept = glm6$coefficients[[1]], linetype="dashed")
ggsave(filename="writeup/6d_loglikelihood.png", width=6.125, height=3.5,
units="in")

rm(list = ls()) # clear working environment
```