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

# Brian Weinstein - bmw2148
# STAT W4201 001
# Homework 6
# 2016-03-09

# set working directory
setwd("~/Documents/advanced-data-analysis/homework_06")

# 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(ggplot2); theme_set(theme_bw())
library(GGally)
library(dplyr)
library(tidyr)
library(formula.tools)
library(gridExtra)



# Problem 1: Ramsey 9.14
#########################################################################

# load data
paceData <- Sleuth3::ex0914

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

# plot a matrix of pairwise scatterplots
plot.pairs <- ggpairs(data=paceData,
                      lower=list(continuous=wrap("points", size=0.7)))
plot.pairs
png(filename="writeup/1a.png", width=11, height=9, units="in", res=300)
print(plot.pairs)
dev.off()

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

# least squares fit of lin reg of Heart on Bank, Walk, Talk
lmPace <- lm(formula=Heart ~ Bank + Walk + Talk, data=paceData)
summary(lmPace)

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

# check the residuals of the fitted model
ggplot(lmPace, aes(x=.fitted, y=.resid)) +
  geom_point() +
  geom_hline(yintercept=0, linetype="dashed") +
```

```
  labs(x="Fitted values", y="Residuals")
ggsave(filename="writeup/1c.png", width=6.125, height=3.5, units="in")

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

# get 95% CIs for the coefficients in lmPace
confint(lmPace, level = 0.95)

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



# Problem 2: Ramsey 9.16
######################################################################

# load data
pollenData <- Sleuth3::ex0327

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

# scatterplot of pollen vs duration, by bee type
ggplot(pollenData, aes(x=DurationOfVisit, y=PollenRemoved, color=BeeType,
shape=BeeType)) +
  geom_point(size=2.5)
ggsave(filename="writeup/2a.png", width=6.125, height=3.5, units="in")

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

# create a logit-transformed "proportion of pollen" variable
pollenData <- pollenData %>%
  mutate(LogitPollenRemoved=log(PollenRemoved/(1-PollenRemoved)))

# scatterplot of proportion of pollen pollen vs duration, by bee type
ggplot(pollenData, aes(x=DurationOfVisit, y=LogitPollenRemoved, color=BeeType,
shape=BeeType)) +
  geom_point(size=2.5)
ggsave(filename="writeup/2b.png", width=6.125, height=3.5, units="in")

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

# create a log-transformed duration variable
pollenData <- pollenData %>%
  mutate(LogDurationOfVisit=log(DurationOfVisit))

# scatterplot of proportion of pollen pollen vs duration, by bee type
ggplot(pollenData, aes(x=LogDurationOfVisit, y=LogitPollenRemoved,
color=BeeType, shape=BeeType)) +
  geom_point(size=2.5)
ggsave(filename="writeup/2c.png", width=6.125, height=3.5, units="in")

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

# create a linear regression model of LogitPollenRemoved on LogDurationOfVisit
* BeeType
lmPollen <- lm(formula=LogitPollenRemoved ~ LogDurationOfVisit * BeeType,
```

```
               data=pollenData)
summary(lmPollen)$coefficients

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

# create a linear regression model of LogitPollenRemoved on LogDurationOfVisit
+ BeeType
lmPollenNoInt <- lm(formula=LogitPollenRemoved ~ LogDurationOfVisit + BeeType,
                    data=pollenData)
summary(lmPollenNoInt)$coefficients

# get 95% CIs for the coefficients in lmPollenNoInt
confint(lmPollenNoInt, level = 0.95)

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



# Problem 3: Ramsey 9.18
#########################################################################

# load data
wingData <- Sleuth3::ex0918

# convert data to long format
wingDataLong <- wingData %>%
  gather(data=., key=Sex, value=Avg_WingSize, c(Females, Males)) %>%
  mutate(SE_WingSize=ifelse(Sex=="Females", SE_Females, SE_Males),
         Ratio=ifelse(Sex=="Females", Ratio, NA),
         SE_Ratio=ifelse(Sex=="Females", SE_Ratio, NA)) %>%
  select(Continent, Latitude, Sex, Avg_WingSize, SE_WingSize, Ratio, SE_Ratio)
%>%
  mutate(Sex=as.factor(Sex))

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

# scatterplot of Avg_WingSize vs Latitude, by Continent and Sex
ggplot(wingDataLong, aes(x=Latitude, y=Avg_WingSize,
                         color=interaction(Continent, Sex),
                         shape=interaction(Continent, Sex))) +
  geom_point(size=2.5) +
  scale_shape_manual(values=c(16, 17, 15, 18))
ggsave(filename="writeup/3a.png", width=6.125, height=3.5, units="in")

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

# relevel the sex variable
wingDataLong$Sex <- relevel(wingDataLong$Sex, "Males")

# create a linear regression model of Avg_WingSize
# on Latitude + Sex * Continent + Latitude * Sex * Continent
lmWing <- lm(formula=Avg_WingSize ~ Latitude + Sex * Continent + Latitude *
Sex * Continent,
             data=wingDataLong)
summary(lmWing)$coefficients
```

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



# Problem 4: Ramsey 9.20
######################################################################

# load data
derbyData <- Sleuth3::ex0920

# create a Year^2 variable
derbyData <- derbyData %>%
  mutate(Year2=Year^2)

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

# plot (Time vs Year) and (Speed vs Year)
ggplot(derbyData, aes(x=Year, y=Time)) + geom_point()
ggplot(derbyData, aes(x=Year, y=Speed)) + geom_point()
ggsave(filename="writeup/4a.png", width=6.125, height=3.5, units="in")

# compare the linear models with Time as the response vs with Speed as the
response
lmDerbyTime <- lm(formula=Time ~ Year + Year2, data=derbyData)
summary(lmDerbyTime)
lmDerbySpeed <- lm(formula=Speed ~ Year + Year2, data=derbyData)
summary(lmDerbySpeed) # higher R-squared

rm(lmDerbyTime, lmDerbySpeed)

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

# create a linear regression model of Speed on Year + Year2 + Conditions
lmDerbyB <- lm(formula=Speed ~ Year + Year2 + Conditions, data=derbyData)
summary(lmDerbyB)$coefficients

# get 95% CIs for the coefficients in lmDerbyB
confint(lmDerbyB, level = 0.95)

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

# create a linear regression model of Speed on Year + Year2 + Conditions *
Starters
lmDerbyC <- lm(formula=Speed ~ Year + Year2 + Conditions * Starters,
data=derbyData)
summary(lmDerbyC)$coefficients

# get 95% CIs for the coefficients in lmDerbyC
confint(lmDerbyC, level = 0.95)

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

```
# Problem 5: Ramsey 10.19
#########################################################################

# load data
meadowData <- Sleuth3::case0901
# Time==2: 24 days before; Time==1: 0 days before

# transform data
meadowData <- meadowData %>%
  mutate(Time24=factor(ifelse(Time==2, 1, 0), levels=c(1, 0)))

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

# create a linear regression model and anova table
# for Flowers on Intensity + Time24
lmMeadowA <- lm(formula=Flowers ~ Intensity + Time24, data=meadowData)
anovaA <- anova(lmMeadowA); anovaA

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

# create a linear regression model and anova table
# for Flowers on factor(Intensity) * Time24
lmMeadowB <- lm(formula=Flowers ~ factor(Intensity) * Time24, data=meadowData)
anovaB <- anova(lmMeadowB); anovaB

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

# lmMeadowB (full) and lmMeadowA (reduced)

# compute the extra sum of squares
esos <- anovaA$`Sum Sq`[3] - anovaB$`Sum Sq`[4]

# define the numerator and denominator degrees of freedom
dfn <- anovaA$Df[3] - anovaB$Df[4]
dfd <- summary(lmMeadowB)$df[2]

# compute the f stat
fstat <- (esos / (length(lmMeadowB$coefficients) -
length(lmMeadowA$coefficients)) ) /
  (summary(lmMeadowB)$sigma)^2
fstat

# compute the pvalue
pval <- 1 - pf(q=fstat, df1=dfn, df2=dfd); pval

# verify the result by using anova to perform extra sum of
# squares F test comparing lmMeadowB (full) and lmMeadowA (reduced)
anova(lmMeadowB, lmMeadowA)

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



# Problem 6: Ramsey 10.28
#########################################################################
```

```
# load data
ninoData <- Sleuth3::ex1028

# plot a matrix of pairwise scatterplots
plot.pairs <- ggpairs(data=select(ninoData,
                                  Year, Temperature, WestAfrica,
                                  Storms, Hurricanes, StormIndex),
                      lower=list(continuous=wrap("points", size=0.7)))
plot.pairs
png(filename="writeup/6_pairs.png", width=11, height=9, units="in", res=300)
print(plot.pairs)
dev.off()

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

# create a linear regression model of Storms on Temperature * WestAfrica
lmNinoA1 <- lm(formula=Storms ~ Temperature * WestAfrica, data=ninoData)
summary(lmNinoA1)$coefficients

# check the residuals of the fitted model
plot.A1.resid <- ggplot(lmNinoA1, aes(x=.fitted, y=.resid)) +
  geom_point() +
  geom_hline(yintercept=0, linetype="dashed") +
  labs(x="Fitted values", y="Residuals",
title=as.character(formula(lmNinoA1)))

# check for serial correlation
plot.A1.serial <- ggplot(ninoData, aes(x=ninoData$Year, y=lmNinoA1$residuals))
+
  geom_point() +
  geom_hline(yintercept=0, linetype="dashed") +
  labs(x="Year", y="Residuals", title=as.character(formula(lmNinoA1)))

# combine both plots
plot.A1.grid <- grid.arrange(plot.A1.resid, plot.A1.serial, nrow=1, ncol=2)
ggsave(filename="writeup/6a1_resid_serial.png", plot=plot.A1.grid, width=11,
height=4.5, units="in")

# create a linear regression model of Storms on Temperature * WestAfrica +
Year + I(Year^2)
lmNinoA2 <- lm(formula=Storms ~ Temperature * WestAfrica + Year + I(Year^2),
data=ninoData)
summary(lmNinoA2)$coefficients

# check the residuals of the fitted model
plot.A2.resid <- ggplot(lmNinoA2, aes(x=.fitted, y=.resid)) +
  geom_point() +
  geom_hline(yintercept=0, linetype="dashed") +
  labs(x="Fitted values", y="Residuals",
title=as.character(formula(lmNinoA2)))

# check for serial correlation
plot.A2.serial <- ggplot(ninoData, aes(x=ninoData$Year, y=lmNinoA2$residuals))
+
```

```
  geom_point() +
  geom_hline(yintercept=0, linetype="dashed") +
  labs(x="Year", y="Residuals", title=as.character(formula(lmNinoA2)))

# combine both plots
plot.A2.grid <- grid.arrange(plot.A2.resid, plot.A2.serial, nrow=1, ncol=2)
ggsave(filename="writeup/6a2_resid_serial.png", plot=plot.A2.grid, width=11,
height=4.5, units="in")

# get 95% CIs for the coefficients in lmNinoA2
confint(lmNinoA2, level = 0.95)

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

# create a linear regression model of Hurricanes on Temperature * WestAfrica
lmNinoB1 <- lm(formula=Hurricanes ~ Temperature * WestAfrica, data=ninoData)
summary(lmNinoB1)$coefficients

# check the residuals of the fitted model
plot.B1.resid <- ggplot(lmNinoB1, aes(x=.fitted, y=.resid)) +
  geom_point() +
  geom_hline(yintercept=0, linetype="dashed") +
  labs(x="Fitted values", y="Residuals",
title=as.character(formula(lmNinoB1)))

# check for serial correlation
plot.B1.serial <- ggplot(ninoData, aes(x=ninoData$Year, y=lmNinoB1$residuals))
+
  geom_point() +
  geom_hline(yintercept=0, linetype="dashed") +
  labs(x="Year", y="Residuals", title=as.character(formula(lmNinoB1)))

# combine both plots
plot.B1.grid <- grid.arrange(plot.B1.resid, plot.B1.serial, nrow=1, ncol=2)
ggsave(filename="writeup/6b1_resid_serial.png", plot=plot.B1.grid, width=11,
height=4.5, units="in")

# get 95% CIs for the coefficients in lmNinoB1
confint(lmNinoB1, level = 0.95)

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

# create a linear regression model of StormIndex on Temperature * WestAfrica
lmNinoC1 <- lm(formula=StormIndex ~ Temperature * WestAfrica, data=ninoData)
summary(lmNinoC1)$coefficients

# check the residuals of the fitted model
plot.C1.resid <- ggplot(lmNinoC1, aes(x=.fitted, y=.resid)) +
  geom_point() +
  geom_hline(yintercept=0, linetype="dashed") +
  labs(x="Fitted values", y="Residuals",
title=as.character(formula(lmNinoC1)))

# check for serial correlation
plot.C1.serial <- ggplot(ninoData, aes(x=ninoData$Year, y=lmNinoC1$residuals))
```

```
+
  geom_point() +
  geom_hline(yintercept=0, linetype="dashed") +
  labs(x="Year", y="Residuals", title=as.character(formula(lmNinoC1)))

# combine both plots
plot.C1.grid <- grid.arrange(plot.C1.resid, plot.C1.serial, nrow=1, ncol=2)
ggsave(filename="writeup/6c1_resid_serial.png", plot=plot.C1.grid, width=11,
height=4.5, units="in")

# create a linear regression model of StormIndex on Temperature * WestAfrica +
Year + I(Year^2)
lmNinoC2 <- lm(formula=StormIndex ~ Temperature * WestAfrica + Year +
I(Year^2), data=ninoData)
summary(lmNinoC2)$coefficients

# check the residuals of the fitted model
plot.C2.resid <- ggplot(lmNinoC2, aes(x=.fitted, y=.resid)) +
  geom_point() +
  geom_hline(yintercept=0, linetype="dashed") +
  labs(x="Fitted values", y="Residuals",
title=as.character(formula(lmNinoC2)))

# check for serial correlation
plot.C2.serial <- ggplot(ninoData, aes(x=ninoData$Year, y=lmNinoC2$residuals))
+
  geom_point() +
  geom_hline(yintercept=0, linetype="dashed") +
  labs(x="Year", y="Residuals", title=as.character(formula(lmNinoC2)))

# combine both plots
plot.C2.grid <- grid.arrange(plot.C2.resid, plot.C2.serial, nrow=1, ncol=2)
ggsave(filename="writeup/6c2_resid_serial.png", plot=plot.A2.grid, width=11,
height=4.5, units="in")

# get 95% CIs for the coefficients in lmNinoA2
confint(lmNinoC2, level = 0.95)

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