```r
################################################################################

# Brian Weinstein - bmw2148
# STAT S4201 001
# Homework 1
# 2016-02-03

# set working directory
setwd("~/Documents/advanced-data-analysis/homework_01")

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



# Problem 1: Ramsey 1.17
#######################################################################

# score data (a1, a2, a3, a4, b1, b2, b3)
scores <- c(68, 77, 82, 85, 53, 64, 71)

# create all combinations of 4-3 group assignments
setA <- combn(scores, 4, simplify=FALSE)
setB <- sapply(1:length(setA), function(i){list(setdiff(scores, setA[[i]]))})

# combine group assignments into dataframe
sets <- lapply(1:length(setA), function(i){c(setA[[i]], setB[[i]])}) %>%
  do.call(rbind, .) %>%
  as.data.frame()
colnames(sets) <- c("a1", "a2", "a3", "a4", "b1", "b2", "b3")

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

# calculate difference between sample averages
sets <- sets %>%
  mutate(avg_a=rowMeans(sets[, 1:4]),
         avg_b=rowMeans(sets[, 5:7]),
         avg_diff=avg_a-avg_b)

# define the observed difference in sample averages
observed_diff <- sets$avg_diff[1] # row 1 has the observed data

# caluclate two-sided p-value of observed_diff
pvalue <- sum(abs(sets$avg_diff) >= abs(observed_diff)) / nrow(sets)

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



# Problem 3: Ramsey 1.25 (b)
```

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

# load data
ratData <- Sleuth3::ex0125

# create boxplots
ggplot(ratData, aes(x=Group, y=Zinc)) +
  geom_boxplot() +
  ylab("Zinc concentration (mg/ml)")
ggsave(filename="writeup/3.png", width=5, height=3, units="in")



# Problem 4
########################################################################

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

# calculate the observed difference in group averages
observed_avg <- ratData %>%
  group_by(Group) %>%
  summarize(groupAvg=mean(Zinc)) %>%
  arrange(Group)
observed_diff <- observed_avg$groupAvg[1] - observed_avg$groupAvg[2]

# Null hypothesis: observed_diff = 0
# Alternative hypothesis: observed_diff != 0

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

# created an index
rat.ndx <- 1:nrow(ratData)

# initialize an empty list to store the difference in group averages
avg_diff <- list()

set.seed(1)

# for 1000 group divisions
for(i in 1:1000){

  # select a sample of rats for group A
  ratGroupA <- sample(rat.ndx, size=20, replace=FALSE)
  ratGroupA.zinc <- ratData$Zinc[ratGroupA]

  # place the remaing ratis in group B
  ratGroupB <- setdiff(rat.ndx, ratGroupA)
  ratGroupB.zinc <- ratData$Zinc[ratGroupB]

  # calculate the differenc in group averages
  avg_diff[[i]] <- mean(ratGroupA.zinc) - mean(ratGroupB.zinc)

}
```

```
# vector of differences in group averages
avg_diff <- unlist(avg_diff)

# caluclate two-sided p-value of observed_diff
pvalue <- sum(abs(avg_diff) >= abs(observed_diff)) / length(avg_diff)

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

ggplot(as.data.frame(avg_diff), aes(x=avg_diff)) +
  geom_histogram(bins=30) +
  geom_vline(xintercept=c(observed_diff, -observed_diff), linetype="dotted") +
  xlab("Test statistic t (difference between sample averages)")
ggsave(filename="writeup/4c.png", width=5, height=3, units="in")



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



# Problem 5: Ramsey 2.12
########################################################################

# dt density function; pt cumulative distribution function; qt quantile
function

# Define a fn to calcuate a conf interval for a given mean, std error, df, and
conf level
MakeConfidenceInterval <- function(mean, se, df, confLevel){

  sigLevel <- 1 - confLevel

  error <- qt(p=1-(sigLevel/2), df=df) * se

  lowerBound <- mean - error
  upperBound <- mean + error

  confidenceInterval <- c(lowerBound, upperBound)

  return(confidenceInterval)

}

# define variables for this problem
mean_value <- 280
std_err <- 46.66
df_value <- 1095

# 95% and 90% confidence intervals for (mu_2 - mu_1)
MakeConfidenceInterval(mean=mean_value, se=std_err, df=df_value,
confLevel=0.95)
MakeConfidenceInterval(mean=mean_value, se=std_err, df=df_value,
confLevel=0.90)
```

```
# t-statistic: (mean - hypothesized_mean) / se, for hypothesized_mean=0
t_stat <- (mean_value - 0) / std_err

# two-sided p-value for this t statistic
2 * pt(q=(-1 * abs(t_stat)), df=df_value)

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



# Problem 6: Ramsey 2.14
#######################################################################

# load data
fishOilData <- Sleuth3::ex0112

# compute the 95% confidence interval (in the same way as in Section 2.3.3,
which uses a 2-sided CI)
t.test(formula=BP~Diet, data=fishOilData,
       var.equal=TRUE, conf.level=0.95)

# given that we're using a 1-sided p-value (which we get here), we could
# also use a 1-sided condificne "interval" with an infinite upper bound
t.test(formula=BP~Diet, data=fishOilData,
       var.equal=TRUE, conf.level=0.95,
       alternative="greater")

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



# Problem 7: Ramsey 2.16
#######################################################################

# load data
creativityData <- Sleuth3::case0101

# reorder Treatment factor levels to be consistent with book's analysis
creativityData$Treatment <- relevel(creativityData$Treatment, "Intrinsic")

# compute t-test
t.test(formula=Score~Treatment, data=creativityData,
       var.equal=TRUE, conf.level=0.95)

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



# Problem 8: Ramsey 2.23
#######################################################################

# load data
highwayData <- Sleuth3::ex0223
```

```
# compute t-test
t.test(formula=PctChange~SpeedLimit, data=highwayData,
       var.equal=TRUE, conf.level=0.95)

ggplot(highwayData, aes(x=SpeedLimit, y=PctChange)) +
  geom_boxplot() +
  ylab("% change in traffic fatalities")
ggsave(filename="writeup/8.png", width=5, height=3, units="in")

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