```
# UCLA Extension - Introduction to Data Science
#
# Homework #2 Solutions
#
# (c) Copyright 2016-2019 - AMULET Analytics
# ---------------------------------------------------------------------

# ---------------------------------------------------------------------
# Question 1
# ---------------------------------------------------------------------

data(CO2)
head(CO2)

#install.packages("sqldf")
library(sqldf)

sqldf("select Type, avg(uptake) from CO2 group by Type")
#         Type avg(uptake)
#1 Mississippi    20.88333
#2      Quebec    33.54286

# ---------------------------------------------------------------------
# Question 2
# ---------------------------------------------------------------------

Died.At <- c(22,40,72,41)
Writer.At <- c(16, 18, 36, 36)
First.Name <- c("John", "Edgar", "Walt", "Jane")
Second.Name <- c("Doe", "Poe", "Whitman", "Austen")
Sex <- c("MALE", "MALE", "MALE", "FEMALE")
Date.Of.Death <- c("2015-05-10", "1849-10-07", "1892-03-26","1817-07-18")

df <- data.frame(Died.At, Writer.At, First.Name, Second.Name, Sex,
                 Date.Of.Death, stringsAsFactors = FALSE)
str(df)
#'data.frame':  4 obs. of  6 variables:
#  $ Died.At      : num  22 40 72 41
#$ Writer.At    : num  16 18 36 36
#$ First.Name   : chr  "John" "Edgar" "Walt" "Jane"
#$ Second.Name  : chr  "Doe" "Poe" "Whitman" "Austen"
#$ Sex          : chr  "MALE" "MALE" "MALE" "FEMALE"
#$ Date.Of.Death: chr  "2015-05-10" "1849-10-07" "1892-03-26" "1817-07-18"

df$Sex <- as.factor(df$Sex)           # Coerce from character to factor
str(df)
#'data.frame':  4 obs. of  6 variables:
#  $ Died.At      : num  22 40 72 41
#$ Writer.At    : num  16 18 36 36
#$ First.Name   : chr  "John" "Edgar" "Walt" "Jane"
#$ Second.Name  : chr  "Doe" "Poe" "Whitman" "Austen"
#$ Sex          : Factor w/ 2 levels "FEMALE","MALE": 2 2 2 1
#$ Date.Of.Death: chr  "2015-05-10" "1849-10-07" "1892-03-26" "1817-07-18"

names(df) <- c("age_at_death", "age_as_writer", "first_name", "surname", "gender", "date_died")
names(df)
#[1] "age_at_death"  "age_as_writer" "first_name"    "surname"       "gender"
#[6] "date_died"

# John Doe died on his birthdate, so his birthdate is:
d <- as.POSIXlt(df$date_died[1])  # convert into POSIXlt
d$year <- d$year - df$age_at_death[1]
as.Date(d)
#[1] "1993-05-10"

# ---------------------------------------------------------------------
# Question 3
# ---------------------------------------------------------------------

# "Long" format for recording observations when there is one observation
# row per variable. A lot of statistical tests favor this format. Here is an
# example of a long format:

#  Product | Attribute | Value
#        A | Height    | 10
#        A | Width     | 5
#        A | Weight    | 2
#        B | Height    | 20
#        B | Width     | 10

# "Wide" format for recording observations when When you have multiple values,
# spread out over multiple columns, for the same observation. Since different
# functions may require you to input your data either in "long" or "wide"
# format, you might need to reshape your data set. To go from a "wide" to
# a "long" data format, you use the melt() function. Here is an example of a
```

```
# wide format

#  Product | Height | Width | Weight
#        A |     10 |     5 |      2
#        B |     20 |    10 |     NA

# Here is code to produce lONG format:
product <- c("A", "A", "A", "B", "B")
attribute <- c("Height", "width", "Weight", "Height", "Width")
value <- c(10,5,2,20,10)
observations_long <- data.frame(product, attribute, value)
observations_long

# Here is code to produce wIDE format:
product <- c("A", "B")
height <- c(10,20)
width <- c(5,10)
weight <- c(2,NA)
observations_wide <- data.frame(product, height, width, weight)
observations_wide

# Here is code to go from wide to long format:
library(reshape2)
long_reshaped2 <- melt(observations_wide,
                       id.vars=c("product"), na.rm=TRUE)
long_reshaped2[order(long_reshaped2$product),]


# ------------------------------------------------------------------
# Question 4
# ------------------------------------------------------------------

library(datasets)
data(mtcars)
? mtcars     # View a description of the data set

sapply(split(mtcars$mpg, mtcars$cyl), mean)   # Answer is C
#       4        6        8
#26.66364 19.74286 15.10000

# ------------------------------------------------------------------
# Question 5
# ------------------------------------------------------------------

data("mtcars")
hp <- sapply(split(mtcars$hp, mtcars$cyl), mean)
hp           # Numeric vector, length=3
#       4         6         8
#82.63636 122.28571 209.21429

abs(hp[1]-hp[3])
#       4
#126.5779

# ------------------------------------------------------------------
# Question 6
# ------------------------------------------------------------------

mean(airquality$Ozone, na.rm=TRUE)  # Answer is A
#[1] 42.12931

# ------------------------------------------------------------------
# Question 7
# ------------------------------------------------------------------

b <- airquality[airquality$Month == 6,]
a <- c(b[,4])
mean(a)          # Answer is D
#[1] 79.1

# ------------------------------------------------------------------
# Question 8
# ------------------------------------------------------------------

data(mtcars)
boxplot(mpg~cyl,data=mtcars, main="Car Milage Data",
        xlab="Number of Cylinders", ylab="Miles Per Gallon")

# ------------------------------------------------------------------
# Question 9
# ------------------------------------------------------------------

install.packages("scatterplot3d")
library(scatterplot3d)
```

```
attach(mtcars)
scatterplot3d(wt, disp, mpg)

# ----------------------------------------------------------------
# Question 10
# ----------------------------------------------------------------

par(mfrow = c(1,1))

# PART 1: Scatterplot of all observations
data(airquality)
with(airquality,plot(Temp,Ozone))

# PART 2 (option a): Scatterplot of filtered observations
with(airquality,plot(Temp,Ozone,xlim=c(min(Temp,na.rm=TRUE),80),
                     ylim=c(min(Ozone,na.rm=TRUE),100)))

# PART 2 (option b): Different axes
with(airquality,plot(Ozone, Temp,xlim=c(min(Ozone,na.rm=TRUE),100),
                     ylim=c(min(Temp,na.rm=TRUE),80)))

# PART 2 (option c): Subsetting method
aq <- airquality[complete.cases(airquality),]
aq <- aq[aq$Ozone<100 & aq$Temp<80,]
plot(aq$Temp, aq$Ozone)
```