Tutorial Exercises Week 4

Read in the dataset: tutorial-data-cleaning.csv. After reading in the "raw" data, the first six rows of the data should look like this:

```
  Sales_Data     Date     Sales Promotion.Sales
1         NA 03.16.18      9657              NA
2         NA 02.08.18      8886              NA
3         NA 04.13.18 Promotion           42312
4         NA 04.14.18 Promotion           35969
5         NA 02.04.18      6500              NA
6         NA 03.24.18      4854              NA
```

The goal of this exercise is to clean this dataset and provide some summary statistics about the cleaned data. When the data is cleaned, the first six rows should look like this:

```
        date sales promotion
1 2018-02-01 22455      TRUE
2 2018-02-02 43011      TRUE
3 2018-02-03  6471     FALSE
4 2018-02-04  6500     FALSE
5 2018-02-05 26509      TRUE
6 2018-02-06  2247     FALSE
```

Complete the following steps to clean the data to get it to look like the 2nd data extract:

- Drop the variable `Sales_Data`.
- Correctly format the "Date" variable as a date.
- Sort the dataset by date. Create a logical variable called `promotion` which is `TRUE` whenever there was a promotion (indicated by a non-NA value in the `Promotion.Sales` variable or the word `Promotion` in the `Sales` variable) and `FALSE` otherwise.
- Whenever the word `"Promotion"` appears in the `Sales` variable, replace it with the corresponding value in `Promotion Sales`.
- Drop the `Promotion.Sales` variable.
- Convert the `Sales` variable from character to numeric.
- Dropping any remaining rows with missing values.
- Convert all variable names to lower case.

Use the techniques discussed in Chapter 13 of the online book to create these data, and use the resulting data to answer the following questions.

* Data Cleaning Steps

We first read in the data and take a peak at it with the `head()` function:

```
df <- read.csv("tutorial-data-cleaning.csv")
head(df)
```

```
  Sales_Data     Date     Sales Promotion.Sales
1         NA 03.16.18      9657              NA
2         NA 02.08.18      8886              NA
3         NA 04.13.18 Promotion           42312
4         NA 04.14.18 Promotion           35969
5         NA 02.04.18      6500              NA
6         NA 03.24.18      4854              NA
```

To drop a variable we assign `NULL` to it:

```
df$Sales_Data <- NULL
```

Let's take a look at the format of the data variable:

```
head(df$Date)
```

```
[1] "03.16.18" "02.08.18" "04.13.18" "04.14.18" "02.04.18" "03.24.18"
```

We see that it's in month-day-year format separated with dots. We know it's month-day-year because we see values larger than 12 in the middle part of the date, indicating that these must be the days and not the months. We can format this with:

```
df$Date <- as.Date(df$Date, format = "%m.%d.%y")
```

We can check that this worked as expected:

```
summary(df$Date)
```

```
        Min.      1st Qu.       Median         Mean      3rd Qu.         Max. 
"2018-02-01" "2018-03-10" "2018-04-16" "2018-04-16" "2018-05-23" "2018-06-30" 
```

There is a promotion whenever the `Sales` variable equals `"Promotion"`. We can create a logical variable out of this with:

```
df$promotion <- df$Sales == "Promotion"
```

Whenever there is a promotion, the value of sales is in the Promotion.Sales variable. What we want to do is put those values into the Sales variable. We can do this by assigning to Sales the values from Promotion.Sales whenever promotion is TRUE:

```
df$Sales[df$promotion] <- df$Promotion.Sales[df$promotion]
head(df$Sales)
```

```
[1] "9657"  "8886"  "42312" "35969" "6500"  "4854"
```

Now that all the character values in Sales are gone, we can convert it to numeric with the as.numeric() function:

```
df$Sales <- as.numeric(df$Sales)
head(df$Sales)
```

```
[1]  9657  8886 42312 35969  6500  4854
```

We now no longer need the Promotion.Sales variable so we can drop it:

```
df$Promotion.Sales <- NULL
```

We can convert all variable names to lower case using the tolower() function. We assign to names(df) the new names, which are the lower case versions of names(df):

```
names(df) <- tolower(names(df))
```

We then drop any rows containing missing values:

```
df <- na.omit(df)
```

Finally, we sort the data by date:

```
df <- df[order(df$date), ]
```

We can check that our data now matches the clean extract:

```
head(df)
```

```
        date sales promotion
1 2018-02-01 22455      TRUE
```

```
2 2018-02-02 43011      TRUE
3 2018-02-03  6471     FALSE
4 2018-02-04  6500     FALSE
5 2018-02-05 26509      TRUE
6 2018-02-06  2247     FALSE
```

Question 1

How many rows are in the final cleaned dataset?

- Solution

```
nrow(df)
```

```
[1] 146
```

Question 2

On how many days were there promotions?

- Solution

We sum the number of times the promotion variable is `TRUE`:

```
sum(df$promotion)
```

```
[1] 20
```

Question 3

What is the average of the cleaned `sales` variable?

- Solution

```
mean(df$sales)
```

```
[1] 9131.774
```

Question 4

What is the average daily sales on days where there were promotions?

* Solution

To obtain the values of sales when there was a promotion, we subset based on what the promotion variable is `TRUE`:

```
df$sales[df$promotion]
```

```
 [1] 22455 43011 26509 36031 44255 31464 30284 28072 35098 42312 35969 46184
[13] 31043 35871 38554 35136 31209 38818 34780 35655
```

We get the average of these values with:

```
mean(df$sales[df$promotion])
```

```
[1] 35135.5
```

Question 5

On which date in April is the median date of the cleaned dataset?

* Solution

```
median(df$date)
```

```
[1] "2018-04-17"
```

We can see that it's in April 17.