Homework 11

Peyton Hall

04/04/2024

```
rm(list=ls())
```

```
library(readxl)
library(ggplot2)
library(plotly)
```

```
##
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':
##
##     last_plot

## The following object is masked from 'package:stats':
##
##     filter

## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(DT)
library(tm)
```

```
## Loading required package: NLP

##
## Attaching package: 'NLP'

## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```r
library(rvest)
```

```r
marketing_data <- read_excel("~/Desktop/Data211/Week 12/marketing_data.xlsx")
marketing_data
```

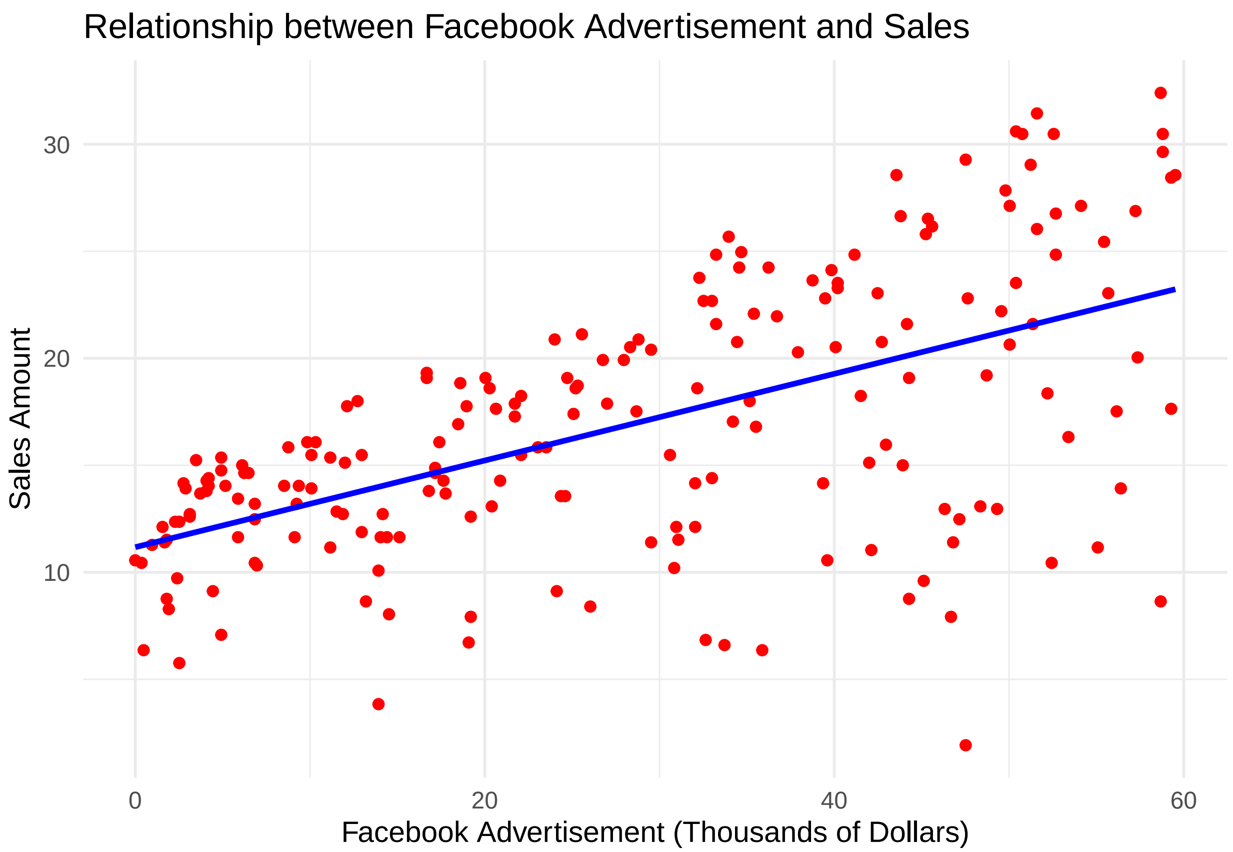
```
## # A tibble: 200 x 2
##    facebook sales
##       <dbl> <dbl>
##  1    45.4  26.5
##  2    47.2  12.5
##  3    55.1  11.2
##  4    49.6  22.2
##  5    13.0  15.5
##  6    58.7   8.64
##  7    39.4  14.2
##  8    23.5  15.8
##  9     2.52  5.76
## 10     3.12 12.7
## # i 190 more rows
```

```r
# 1) The marketing_data on d2l recorded the impact of the advertising media,
#    Facebook, on sales. Data are the advertising budget in thousands of dollars
#    along with the sales. Use the data to answer the following:
# a) Generate an appropriate graph using ggplot to show the relationship between
#    the Facebook advertisement amount (x) and sales amount (y); Color the
#    points red; Fit a linear model to the points without showing the confidence
#    interval band; Color the line blue.
plot_a <- ggplot(marketing_data, aes(x = facebook, y = sales)) +
  geom_point(color = "red") +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(title = "Relationship between Facebook Advertisement and Sales",
       x = "Facebook Advertisement (Thousands of Dollars)",
       y = "Sales Amount") +
  theme_minimal()
# b) Based on a), generate an interactive plot.
plot_b <- ggplotly(plot_a)
```

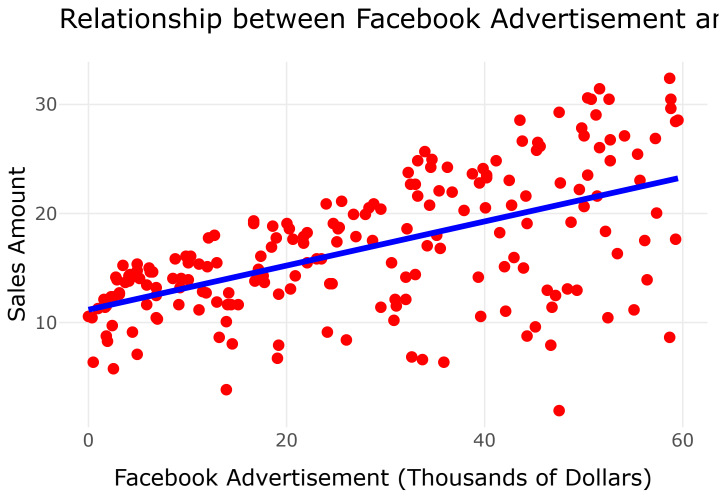
```
## `geom_smooth()` using formula = 'y ~ x'
```

```r
plot_a
```

```
## `geom_smooth()` using formula = 'y ~ x'
```


Relationship between Facebook Advertisement and Sales
Sales Amount
30
20
10
0
20
40
60
Facebook Advertisement (Thousands of Dollars)

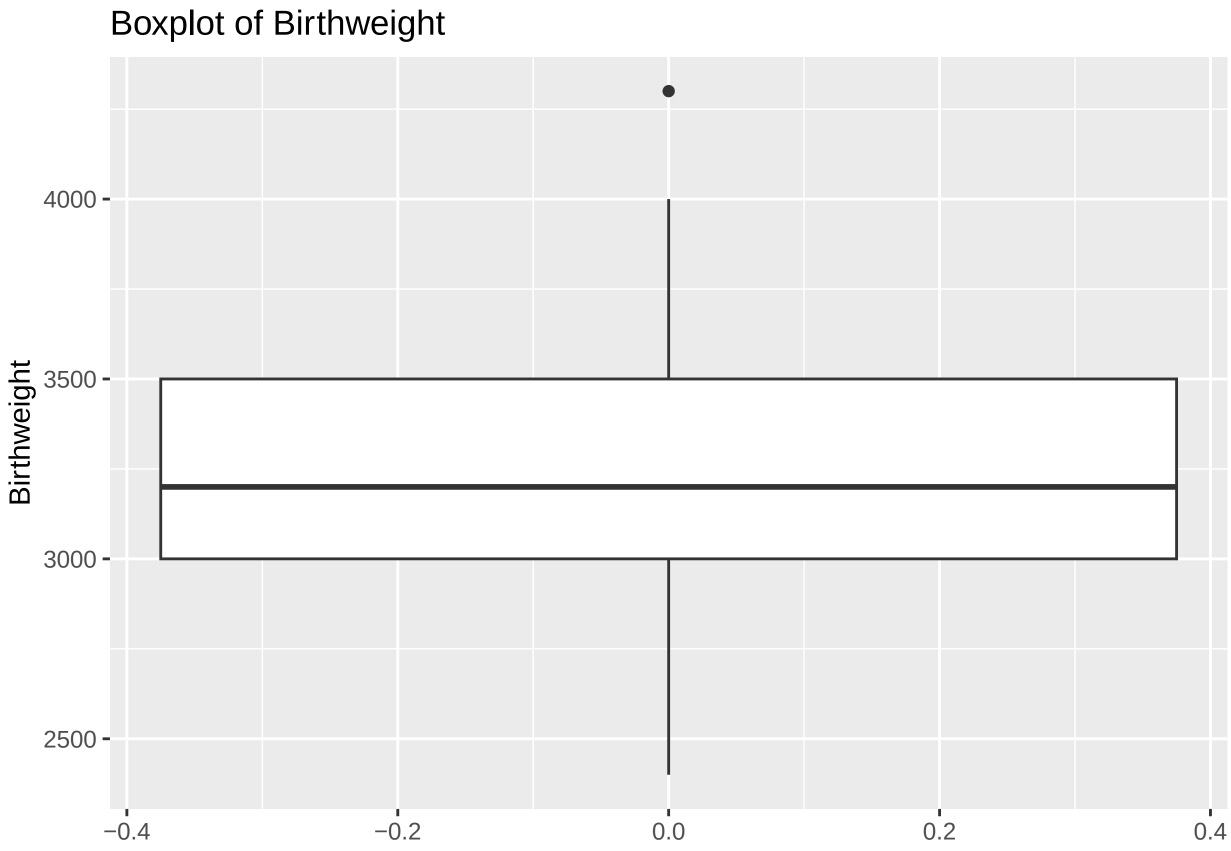
```
plot_b
```



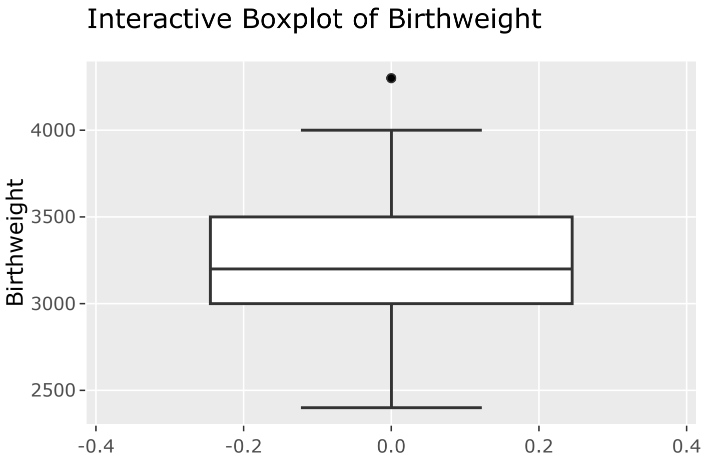
```
Estriol_and_birthweight <- read_excel("~/Desktop/Data211/Week 12/Estriol and birthweight.xlsx")
Estriol_and_birthweight
```

```
## # A tibble: 27 x 2
##    Estriol Birthweight
##      <dbl>       <dbl>
##  1      14        2700
##  2      16        2700
##  3      16        2400
##  4      14        3000
##  5      16        3000
##  6      16        3100
##  7      17        3000
##  8      19        3100
##  9      21        3000
## 10      24        2800
## # i 17 more rows
```

```
# 2) The Estriol and Birthweight data on d2l recorded the mothers' estriol
#    levels at pregnancy and the newborns' birthweight.
# a) Use the data to generate an appropriate graph to show the five-number
#    summary for birth weight.
boxplot <- ggplot(Estriol_and_birthweight, aes(y = Birthweight)) +
  geom_boxplot() +
  ylab("Birthweight") +
  ggtitle("Boxplot of Birthweight")
boxplot
```



```r
# b) Based on a), create an interactive plot.
p <- ggplot(Estriol_and_birthweight, aes(y = Birthweight)) +
  geom_boxplot() +
  ylab("Birthweight") +
  ggtitle("Interactive Boxplot of Birthweight")
interactive_p <- ggplotly(p)
interactive_p
```

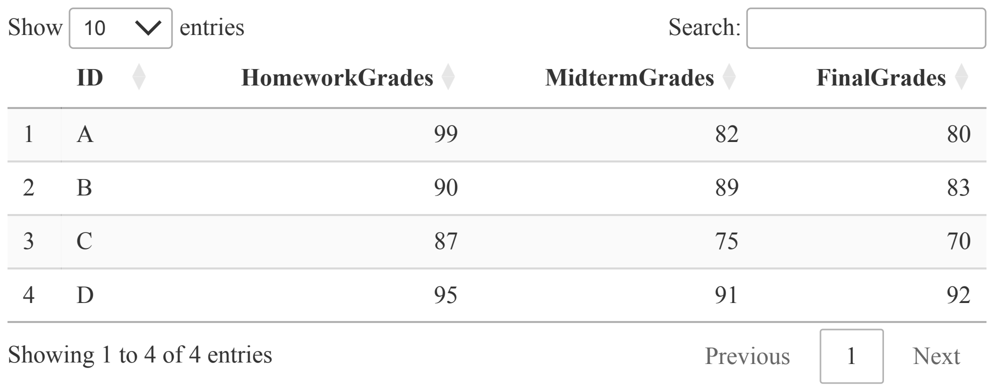


```
# c) Based on the plot, is there any outlier? If so, what is the birth weight
#    of that outlier? Use a comment to write it in R Markdown.
# Based on the plot, an outlier can be visualiszd at greater than 4,000 for
# birthweight. Therefore, there is one clear outlier.
```

```
# 3)
# a) Create the following table as a data frame named as StudentGrades
# ID, HomeworkGrades, MidtermGrades, FinalGrades
# A,  99,             82,            80
# B,  90,             89,            83
# C,  87,             75,            70
# D,  95,             91,            92
# Create the StudentGrades data frame
StudentGrades <- data.frame(
  ID = c("A","B","C","D"),
  HomeworkGrades = c(99, 90, 87, 95),
  MidtermGrades = c(82, 89, 75, 91),
  FinalGrades = c(80, 83, 70, 92)
)
StudentGrades
```

```
##   ID HomeworkGrades MidtermGrades FinalGrades
## 1  A             99            82          80
## 2  B             90            89          83
## 3  C             87            75          70
## 4  D             95            91          92
```

```
# b) Create an interactive table of a)
datatable(StudentGrades)
```



Show 10 entries Search:

	ID	HomeworkGrades	MidtermGrades	FinalGrades
1	A	99	82	80
2	B	90	89	83
3	C	87	75	70
4	D	95	91	92

Showing 1 to 4 of 4 entries

```
# 4)
# a) Create a character vector, a, with these store names: Walmart,
```

```r
#    Walmart-marketplace, Walmart-online, Target, Target-marketplace, Amazon,
#    Amazon.com, AmazonFresh;
a <- c("Walmart", "Walmart-marketplace", "Walmart-online",
       "Target", "Target-marketplace",
       "Amazon", "Amazon.com", "AmazonFresh")
# b) Create a for-loop with the index goes from 1 to the end of the vector a;
#    if the first five letters of the store names are the same (think about using
#    substr()), then those are the same store. We want to count the number of
#    unique stores in the vector, a. You may want to create two for-loops, with one
#    to scrape the first five letters of each element of a and the other loop to
#    compare whether the scraped phrases are the same. At the end, you should get 3
#    stores.
# Initialize an empty vector to store the first five letters of each store name
first_five_letters <- character(length(a))

# Loop through each store name and extract the first five letters
for (i in 1:length(a)) {
  first_five_letters[i] <- substr(a[i], 1, 5)
} # end for

# Initialize a counter for unique stores
unique_stores <- 0

# Loop through the first five letters and compare them
for (i in 1:length(first_five_letters)) {
  # Initialize boolean to check if the current store is unique
  is_unique <- TRUE

  # Compare the current store's first five letters with the previous ones
  for (j in 1:(i - 1)) {
    if (first_five_letters[i] == first_five_letters[j]) {
      # If the first five letters match, set boolean to False and break the loop
      is_unique <- FALSE
      break
    } # end if
  } # end for

  # If the current store is unique, increment the counter
  if (is_unique) {
    unique_stores <- unique_stores + 1
  } # end if
} # end for

# Display the number of unique stores
unique_stores
```

```
## [1] 2
```

```r
# 5) Use the text on
#    https://www.nytimes.com/2023/03/16/business/media/tiktok-buyer-biden.html
#    to generate a word cloud, following along with what we did in class for
#    text mining (create VectorSource, Corpus, remove all numbers, punctuations,
#    white space, English common words, etc.), generate a word cloud.
```

```r
url <- "https://www.nytimes.com/2023/03/16/business/media/tiktok-buyer-biden.html"
text <- read_html(url) %>%
  html_text()

# Create a Corpus object
corpus <- Corpus(VectorSource(text))

# Perform text preprocessing
corpus <- tm_map(corpus, content_transformer(tolower))
```

```
## Warning in tm_map.SimpleCorpus(corpus, content_transformer(tolower)):
## transformation drops documents
```

```r
corpus <- tm_map(corpus, removeNumbers)
```

```
## Warning in tm_map.SimpleCorpus(corpus, removeNumbers): transformation drops
## documents
```

```r
corpus <- tm_map(corpus, removePunctuation)
```

```
## Warning in tm_map.SimpleCorpus(corpus, removePunctuation): transformation drops
## documents
```

```r
corpus <- tm_map(corpus, removeWords, stopwords("en"))
```

```
## Warning in tm_map.SimpleCorpus(corpus, removeWords, stopwords("en")):
## transformation drops documents
```

```r
corpus <- tm_map(corpus, stripWhitespace)
```

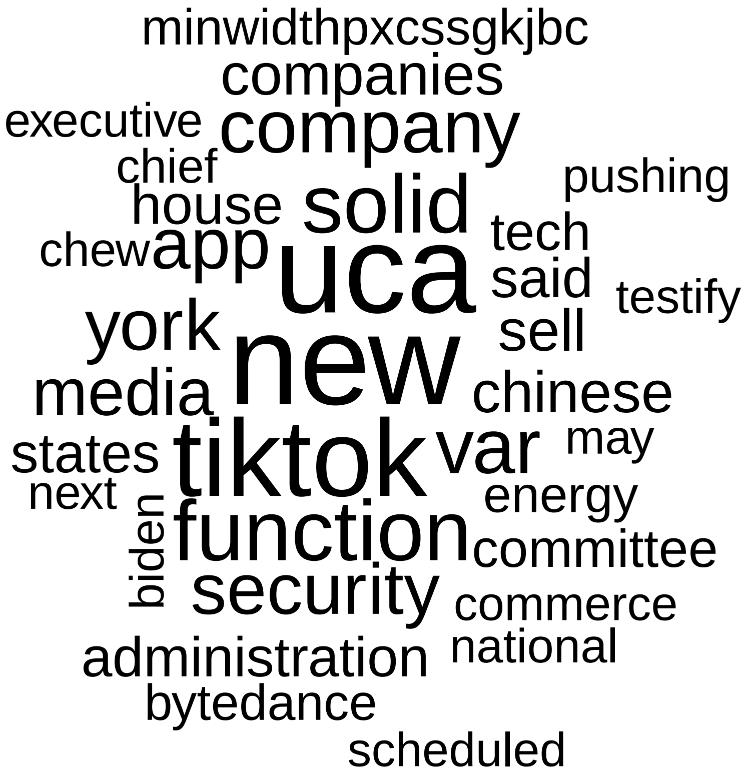
```
## Warning in tm_map.SimpleCorpus(corpus, stripWhitespace): transformation drops
## documents
```

```r
# Create a term-document matrix
tdm <- TermDocumentMatrix(corpus)

# Convert the term-document matrix to a matrix
m <- as.matrix(tdm)

# Calculate word frequencies
word_freq <- sort(rowSums(m), decreasing = TRUE)

# Create a word cloud
wordcloud(names(word_freq), word_freq, min.freq = 10, random.order = FALSE)
```


minwidthpxcssgkjbc
companies
executive
company
chief
pushing
house
solid
tech
chew
app
uca
said
testify
york
sell
new
media
chinese
states
tiktok
var
may
next
energy
biden
function
committee
security
commerce
administration
national
bytedance
scheduled