Data211-FinalExam

Peyton Hall

04/18/2024

```
library(readxl)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
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(tm)
```

```
## Loading required package: NLP

##
## Attaching package: 'NLP'
```

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

```
library(wordcloud)
```

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

```
library(tidyr)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.5
## v lubridate 1.9.3     v stringr   1.5.1
## v purrr     1.0.2     v tibble    3.2.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x NLP::annotate()  masks ggplot2::annotate()
## x plotly::filter() masks dplyr::filter(), stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
ClassSurvey <- read_excel("~/Desktop/Data211Final/ClassSurvey.xlsx")
COVIDVac <- read_excel("~/Desktop/Data211Final/COVIDVac.xlsx")
MLB_teams <- read_excel("~/Desktop/Data211Final/MLB_teams.xlsx")
ReadingWritingScores <- read_excel("~/Desktop/Data211Final/ReadingWritingScores.XLSX")
TireData <- read_excel("~/Desktop/Data211Final/TireData.XLSX")
```

Question 01

```
# 1. (11 pts) Use the COVIDVac data on D2L, where the first column showed the
#    date that the vaccine was administered, the second column showed the state
#    where the vaccine was administered, and the last column showed the number
#    of vaccines completed. Perform the following:

# a. (4 pts) Use appropriate functions in tidyverse and pipeline to only keep
#    the rows where the Series_Complete is not 0 (Hint: logical operator for not
#    equal to is "!=").
# Filter rows where Series_Complete is not equal to 0
COVIDVac_filtered <- COVIDVac %>%
  filter(Series_Complete != 0)
COVIDVac_filtered # View the filtered data
```

```
## # A tibble: 8,132 x 3
##    Date                State Series_Complete
##    <dttm>              <chr>           <dbl>
##  1 2021-06-07 00:00:00 AK               1476
##  2 2021-06-07 00:00:00 AK               3682
##  3 2021-06-07 00:00:00 AK               3932
```

```
##  4 2021-06-07 00:00:00 AK              8164
##  5 2021-06-07 00:00:00 AK             20156
##  6 2021-06-07 00:00:00 AK              2110
##  7 2021-06-07 00:00:00 AK               615
##  8 2021-06-07 00:00:00 AK              6091
##  9 2021-06-07 00:00:00 AK              8650
## 10 2021-06-07 00:00:00 AK              1743
## # i 8,122 more rows
```

```
# b. (4 pts) Continue the pipeline and use appropriate functions in tidyverse to
#    find the average of Series_Complete by State. Who are the top four
#    completed vaccination states? (Use an appropriate function in tidyverse to
#    show this)
# Continue the pipeline to calculate the average Series_Complete by State
COVIDVac_summary <- COVIDVac_filtered %>%
  group_by(State) %>%
  summarise(Avg_Series_Complete = mean(Series_Complete))
# Identify the top four completed vaccination states
top_four_states <- COVIDVac_summary %>%
  top_n(4, Avg_Series_Complete) %>%
  arrange(desc(Avg_Series_Complete))
# View the summary and top four states
COVIDVac_summary
```

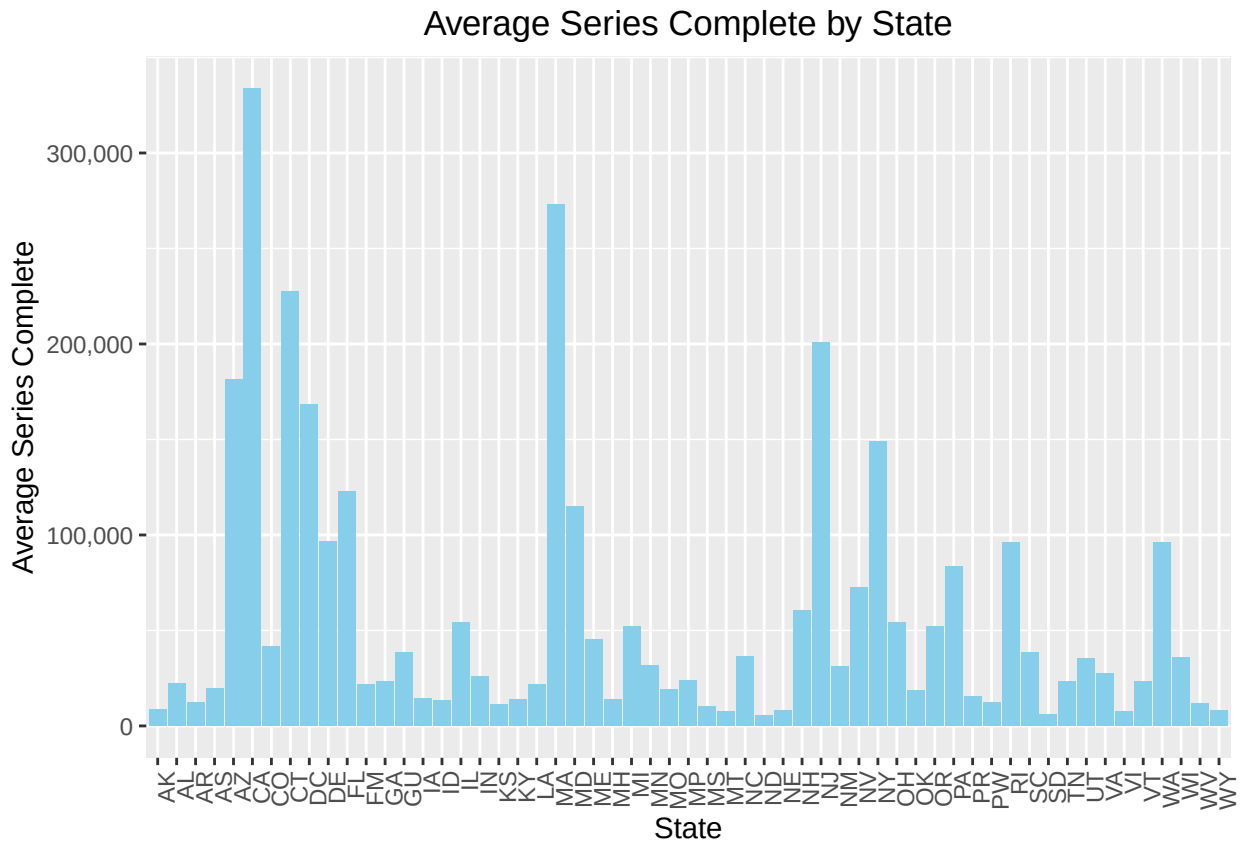
```
## # A tibble: 57 x 2
##    State Avg_Series_Complete
##    <chr>               <dbl>
##  1 AK                  8721.
##  2 AL                 22119.
##  3 AR                 12550.
##  4 AS                 19590
##  5 AZ                181275.
##  6 CA                333688.
##  7 CO                 41706.
##  8 CT                227291.
##  9 DC                168089.
## 10 DE                 96645.
## # i 47 more rows
```

```
top_four_states
```

```
## # A tibble: 4 x 2
##   State Avg_Series_Complete
##   <chr>               <dbl>
## 1 CA                333688.
## 2 MA                272900.
## 3 CT                227291.
## 4 NJ                200579.
```

```
# c. (3 pts) Continue the pipeline and use the ggplot() to generate an
# appropriate graph to show the average Series_Complete by state.
ggplot(data = COVIDVac_summary, aes(x = State, y = Avg_Series_Complete)) +
```

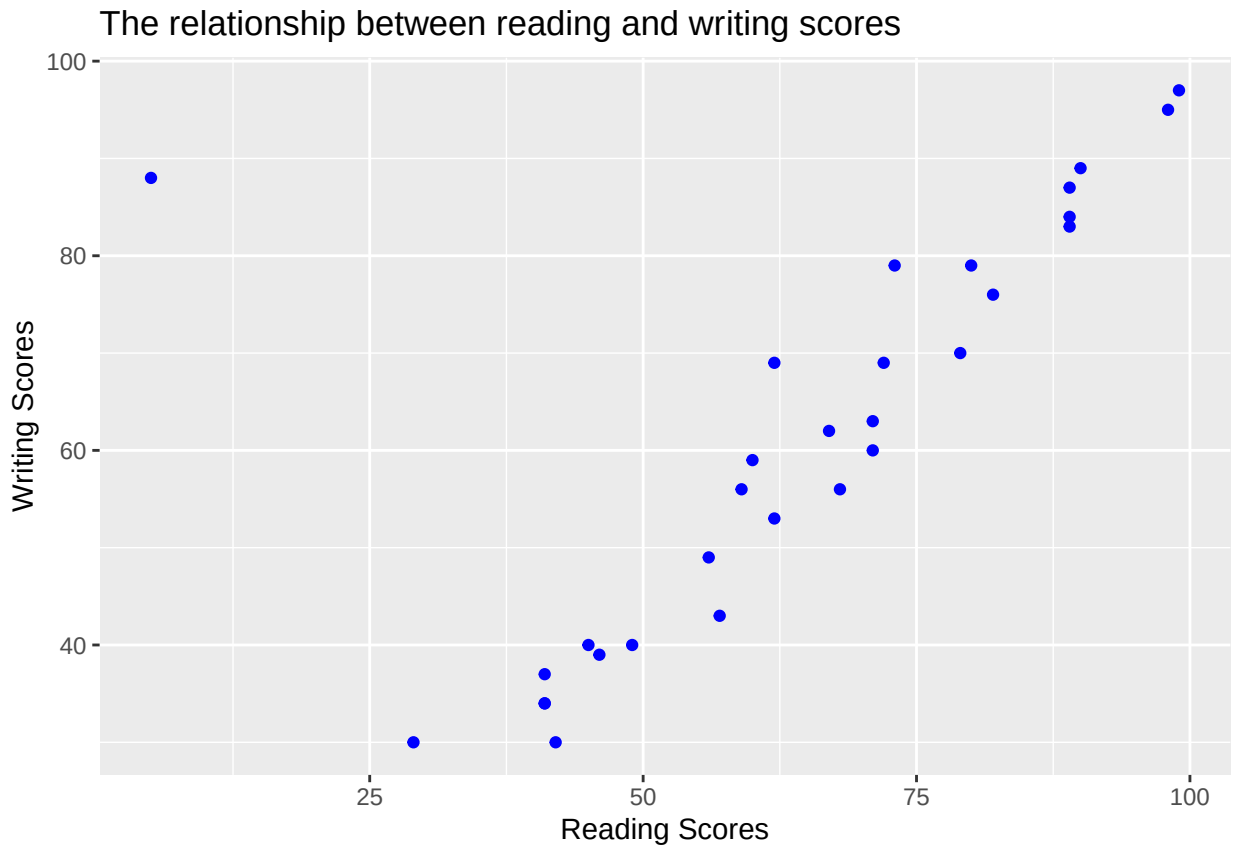
```r
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Average Series Complete by State",
       x = "State",
       y = "Average Series Complete") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
        plot.title = element_text(hjust = 0.5)) +
  scale_y_continuous(labels = scales::comma) # remove y axis scientific notation
```



Question 02

```r
# 2. (11 pts) The data ReadingWritingScores in D2L, recorded the reading scores
#    and writing scores of 30 individuals in a test. Read the
#    ReadingWritingScore data into R and complete the following:

# a) (5pts) Generate an appropriate plot to show the relationship between the
#    reading scores (x) and writing scores (y). Color the points blue. Label the
#    x-axis "Reading Scores" and the y-axis "Writing scores" and title "The
#    relationship between reading and writing scores."
ggplot(ReadingWritingScores, aes(x = ReadingScore, y = WritingScore)) +
  geom_point(color = "blue") +
  labs(x = "Reading Scores", y = "Writing Scores",
       title = "The relationship between reading and writing scores")
```




```
# b) (2pts) Based on the graph in a), generate an interactive graph.
p <- ggplot(ReadingWritingScores, aes(x = ReadingScore, y = WritingScore)) +
  geom_point(color = "blue") +
  labs(x = "Reading Scores", y = "Writing Scores",
       title = "The relationship between reading and writing scores")
# ggplotly(p) # Convert ggplot object to plotly

# c) (4pts) Based on the graph in b), is there any outlier? If so, what are the
#    reading and writing scores of the outlier? Use the RMarkdown report to
#    write answers.
```

Answer to Question 02 Part c): There is undoubtedly an outlier by reason of one dot which is uniquely plotted at the far top right corner of the scatter plot (specified by geom_point). According to the interactive graph, the scores seem unusually unrelated: "ReadingScore: 5. WritingScore: 88".

Question 03

```
# 3. (11pts) The following data frame provides the population size of the top
#    five popularized Minnesota counties in 2022:

# county      population2022
# Hennepin,   1270283
# Ramsey,     549377
# Dakota,     435863
```

```r
# Anoka,      360773
# Washington, 264818

# a) (3pts) Create the data frame, and name it mycounty. Print the mycounty
#    dataset.

# Create the mycounty data frame
mycounty <- data.frame(
  county = c("Hennepin", "Ramsey", "Dakota", "Anoka", "Washington"),
  population2022 = c(1270283, 549377, 435863, 360773, 264818))
mycounty
```

```
##       county population2022
## 1   Hennepin        1270283
## 2     Ramsey         549377
## 3     Dakota         435863
## 4      Anoka         360773
## 5 Washington         264818
```

```r
# b) (8pts) Suppose we know that the population size of the previous year (2021)
#    was 5% less for the Hennepin county, and 10% less for other counties. Use a
#    for-loop to create a new column named population2021, to show the
#    population size in 2021, with the following requirements:
# 1) (2pts) the loop index i should go from 1 to 5
# 2) (2pts) the loop index i must connect with the variable(s) in the data frame
# 3) (2pts) the final data frame should have three columns: county,
#    population2022, and population2021.
# 4) (2pts) Print the final data frame with all three columns.
population2021 <- numeric() # empty vector stores 2021 population size
for (i in 1:5) # Loop through each row in the data frame
{
  # Calculate the population size for 2021 based on the condition
  if (mycounty$county[i] == "Hennepin")
  {
    population2021[i] <- mycounty$population2022[i] * 0.95  # Hennepin = 5% less
  } # end if
  else
  {
    population2021[i] <- mycounty$population2022[i] * 0.90  # others = 10% less
  } # end else
} # end for

mycounty$population2021 <- population2021 # include population2021 in data frame
mycounty # Print the final data frame with all three columns
```

```
##       county population2022 population2021
## 1   Hennepin        1270283      1206768.8
## 2     Ramsey         549377       494439.3
## 3     Dakota         435863       392276.7
## 4      Anoka         360773       324695.7
## 5 Washington         264818       238336.2
```

Question 04

```
# 4. (11pts) Perform the following:
# a) (3pts) Create a text vector named mytext containing three elements:
#    "I didn't do it", "I haven't done it", and "I wouldn't do it".
mytext <- c("I didn't do it", "I haven't done it", "I wouldn't do it")

# b) (6pts) Create a function in R, named textreplacing, to do text editing as
#    follows:
#    - Replace (update) any phrase of "n't" by " not" in a text vector, using
#    gsub(). Note that the body of the function should be very simple.
#    - The function should have one parameter (input), which is the text vector.
#    - The function should have one return value, which is the updated text
#    vector.
textreplacing <- function(x) {
  updated_text <- gsub("n't", " not", x)
  return(updated_text)
}

# c) (2pts) Run the function in 4(b) for the vector mytext.
textreplacing(mytext)
```

```
## [1] "I did not do it"    "I have not done it" "I would not do it"
```

Question 05

```
# 5. (11pts) Use the MLB_teams data on d2l. The data recorded 210 baseball
#    teams' number of wins (W), number of losses (L), payroll amount (payroll),
#    team name (name), team ID (teamID), ID and year. Use one pipeline to do
#    (a, b, and c) the following:

# a. (2pts) Keep the columns ID, W, L, and payroll only
MLB_teams_filtered <- MLB_teams %>%
  select(ID, W, L, payroll)
MLB_teams_filtered
```

```
## # A tibble: 210 x 4
##    ID        W     L   payroll
##    <chr> <dbl> <dbl>     <dbl>
##  1 NL       82    80  66202712
##  2 NL       72    90 102365683
##  3 AL       68    93  67196246
##  4 AL       95    67 133390035
##  5 AL       89    74 121189332
##  6 NL       97    64 118345833
##  7 NL       74    88  74117695
##  8 AL       81    81  78970066
##  9 NL       74    88  68655500
## 10 AL       74    88 137685196
## # i 200 more rows
```

```r
# b. (2pts) Keep only the NL (national leagues) teams of ID variable
MLB_teams_filtered <- MLB_teams %>%
  select(ID, W, L, payroll) %>%
  filter(substr(ID, 1, 2) == "NL")
MLB_teams_filtered
```

```
## # A tibble: 110 x 4
##    ID        W     L   payroll
##    <chr> <dbl> <dbl>     <dbl>
##  1 NL       82    80  66202712
##  2 NL       72    90 102365683
##  3 NL       97    64 118345833
##  4 NL       74    88  74117695
##  5 NL       74    88  68655500
##  6 NL       84    77  21811500
##  7 NL       86    75  88930414
##  8 NL       84    78 118588536
##  9 NL       90    72  80937499
## 10 NL       89    73 137793376
## # i 100 more rows
```

```r
# c. (3pts) Add two new variables: winpercent=W/(W+L) and
#    payrollm=payroll/1000000 to the end of the dataset. The winpercent records
#    the percent of the number of wins, and the payrollm is a conversion from
#    dollar to million dollars.
MLB_teams_modified <- MLB_teams %>%
  select(ID, W, L, payroll) %>%
  filter(substr(ID, 1, 2) == "NL") %>%
  mutate(winpercent = W / (W + L),
         payrollm = payroll / 1000000)
MLB_teams_modified
```

```
## # A tibble: 110 x 6
##    ID        W     L   payroll winpercent payrollm
##    <chr> <dbl> <dbl>     <dbl>      <dbl>    <dbl>
##  1 NL       82    80  66202712      0.506     66.2
##  2 NL       72    90 102365683      0.444    102.
##  3 NL       97    64 118345833      0.602    118.
##  4 NL       74    88  74117695      0.457     74.1
##  5 NL       74    88  68655500      0.457     68.7
##  6 NL       84    77  21811500      0.522     21.8
##  7 NL       86    75  88930414      0.534     88.9
##  8 NL       84    78 118588536      0.519    119.
##  9 NL       90    72  80937499      0.556     80.9
## 10 NL       89    73 137793376      0.549    138.
## # i 100 more rows
```

```r
# d. (2pts) Save the pipe of a), b) and c) to a name, MLBNL.
MLBNL <- function(data) {
  data %>%
    select(ID, W, L, payroll) %>%
    filter(substr(ID, 1, 2) == "NL") %>%
```

```
    mutate(winpercent = W / (W + L),
           payrollm = payroll / 1000000)
}
# Apply the MLBNL function to the data
MLB_teams_modified <- MLBNL(MLB_teams)
MLB_teams_modified
```

```
## # A tibble: 110 x 6
##    ID        W     L   payroll winpercent payrollm
##    <chr> <dbl> <dbl>     <dbl>      <dbl>    <dbl>
##  1 NL       82    80  66202712      0.506     66.2
##  2 NL       72    90 102365683      0.444    102.
##  3 NL       97    64 118345833      0.602    118.
##  4 NL       74    88  74117695      0.457     74.1
##  5 NL       74    88  68655500      0.457     68.7
##  6 NL       84    77  21811500      0.522     21.8
##  7 NL       86    75  88930414      0.534     88.9
##  8 NL       84    78 118588536      0.519    119.
##  9 NL       90    72  80937499      0.556     80.9
## 10 NL       89    73 137793376      0.549    138.
## # i 100 more rows
```

```
# e. (2pts) Print the first 6 lines of MLBNL
head(MLBNL(MLB_teams))
```

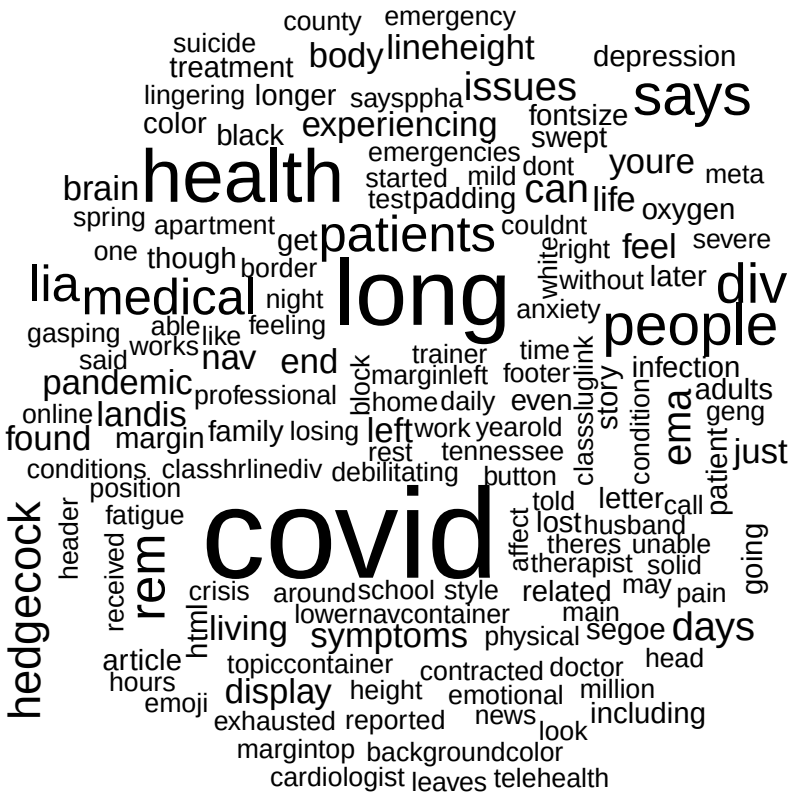
```
## # A tibble: 6 x 6
##   ID        W     L   payroll winpercent payrollm
##   <chr> <dbl> <dbl>     <dbl>      <dbl>    <dbl>
## 1 NL       82    80  66202712      0.506     66.2
## 2 NL       72    90 102365683      0.444    102.
## 3 NL       97    64 118345833      0.602    118.
## 4 NL       74    88  74117695      0.457     74.1
## 5 NL       74    88  68655500      0.457     68.7
## 6 NL       84    77  21811500      0.522     21.8
```

Question 06

```
# 6. (11pts) Use the text website https://text.npr.org/1164284653 . The text
#    site is about long COVID patients. Based on this text, and follow along
#    with what we did in class for text mining (create VectorSource, Corpus,
#    convert all to lower cases, remove all numbers, punctuations, white space,
#    and English stop words, etc.) to generate a word cloud. Use min.freq=2.
url <- "https://text.npr.org/1164284653"
text <- readLines(url) # Read the text from the URL
text <- tolower(text) # Convert the text to lowercase
text <- gsub("\\d+", "", text) # Remove numbers
text <- gsub("[[:punct:]]", "", text) # Remove punctuation
text <- gsub("\\s+", " ", text) # Remove white space
corpus <- Corpus(VectorSource(text)) # Create a corpus from the text
corpus <- tm_map(corpus, removeWords, stopwords("en")) # Omit English stop words
```

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

```
dtm <- DocumentTermMatrix(corpus) # Convert the corpus to a Document-Term Matrix
matrix <- as.matrix(dtm) # Convert the Document-Term Matrix to a matrix
word_freq <- colSums(matrix) # Calculate word frequencies
# Create a dataframe of word frequencies
word_freq_df <- data.frame(word = names(word_freq), freq = word_freq)
word_freq_df <- subset(word_freq_df, freq >= 2) # Filter words at frequency >= 2
# Generate the word cloud
wordcloud(words = word_freq_df$word, freq = word_freq_df$freq, min.freq = 2)
```



Question 07

$H_0 : p = 0.5$ vs $H_a : p < 0.5$

```
# 7. (11pts) Use the data ClassSurvey on D2L. The data recorded the survey to a
#    class asking whether each student has ever cheated on any assignment,
#    including exams.
# a) (4pts) Use the data and the function table() to find the number of students
#    who ever cheated.
cheating_count <- table(ClassSurvey$Ever_Cheat)
cheating_count
```

```
##
##  No Yes
## 171  55
```

```
# b) (7pts) Use the data to test the claim that the proportion of students who
#    have ever cheated on any assignment is significantly less than 50% at a
#    0.05 significance level. Use report format in RMarkdown to write down your
#    H0, Ha, and decision.
n <- nrow(ClassSurvey) # Define the sample size
cheated <- sum(ClassSurvey$Ever_Cheat == "Yes") # count the amount of cheaters
# Perform the one-sample proportion test
prop.test(cheated, n, p = 0.5, alternative = "less", conf.level = 0.95) #(x,n,p)
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  cheated out of n, null probability 0.5
## X-squared = 58.518, df = 1, p-value = 1.007e-14
## alternative hypothesis: true p is less than 0.5
## 95 percent confidence interval:
##  0.0000000 0.2954971
## sample estimates:
##         p
## 0.2433628
```

The p-value of 1.007e-14 is significantly less than the 0.05 significance level. This indicates strong evidence against the H_0. Therefore, the H_0 must be rejected and the H_a is true. There is sufficient evidence to support the claim that the proportion of students who have ever cheated on any assignment is significantly less than 50% at a 0.05 significance level.

Question 08

$H_0 : \mu_l = \mu_r$ vs $H_a : \mu_l < \mu_r$

```
# 8. (11pts) The data TireData (data source: www.statcrunch.com ) on D2L
#    recorded the tire pressure (in psi.) of cars. The data is from paired
#    samples. Each row in the first column showed the right tire pressure, and
#    the second column showed the left tire pressure from the same car.
TireData <- read_excel("~/Desktop/Data211Final/TireData.XLSX")
TireData
```

```
## # A tibble: 18 x 2
##    RightTire LeftTire
##        <dbl>    <dbl>
##  1        48       42
##  2        80       75
##  3        34       24
##  4        63       56
##  5        51       52
##  6        45       56
##  7        29       23
##  8        58       55
```

```
##  9        50       46
## 10        50       52
## 11        50       47
## 12        69       62
## 13        55       55
## 14        65       62
## 15        48       42
## 16        73       75
## 17        22       24
## 18        59       56
```

```
# column headers: "RightTire", "LeftTire"

# a) (3pts) The data provided is a wide format. Use the appropriate R function
#    to change it to a long format data and save the converted data as NewTire.
NewTire <- pivot_longer(TireData,
                        cols = everything(),
                        names_to = "TireSide",
                        values_to = "TirePressure")
NewTire
```

```
## # A tibble: 36 x 2
##    TireSide  TirePressure
##    <chr>            <dbl>
##  1 RightTire           48
##  2 LeftTire            42
##  3 RightTire           80
##  4 LeftTire            75
##  5 RightTire           34
##  6 LeftTire            24
##  7 RightTire           63
##  8 LeftTire            56
##  9 RightTire           51
## 10 LeftTire            52
## # i 26 more rows
```

```
# b) (1pts) Print the first 6 lines of the NewTire data.
head(NewTire)
```

```
## # A tibble: 6 x 2
##   TireSide  TirePressure
##   <chr>            <dbl>
## 1 RightTire           48
## 2 LeftTire            42
## 3 RightTire           80
## 4 LeftTire            75
## 5 RightTire           34
## 6 LeftTire            24
```

```
# c) (7pts) Use the data NewTire to test if the average right tire pressure is
#    significantly higher than the average left tire pressure (use 0.05
#    significance level). Use report format in RMarkdown to write the H0 and
#    Ha, the decision to H0, and explain the decision in the context.
```

```r
t_test_result <- t.test(TirePressure ~ TireSide, data = NewTire, paired = TRUE,
                        alternative = "greater")
t_test_result
```

```
##
##  Paired t-test
##
## data:  TirePressure by TireSide
## t = -2.1744, df = 17, p-value = 0.978
## alternative hypothesis: true mean difference is greater than 0
## 95 percent confidence interval:
##  -4.50009      Inf
## sample estimates:
## mean difference
##            -2.5
```

The 0.978 p-value is not lower than the 0.05 significance level. Therefore, the rejection of the H_0 is a failure. There is insufficient evidence to support the claim that the average right tire pressure is significantly higher than the average left tire pressure.

Question 09 $H_0 : \mu_s = \mu_v$ vs $H_a : \mu_s > \mu_v$

```r
# 9. (12pts) Use the iris data in R. If you do View(iris) in the R console
#    window, you can see the data. The variables, Sepal.Length, Sepal.Width,
#    Petal.Length, and Petal.Width recorded the sepal length and width and petal
#    length and width of 150 iris flowers, respectively. The variable, Species,
#    recorded the type of species that each flower belongs to. Complete the
#    following:
# View(iris) # built-in dataset
# a) (2pts) There are three species recorded in the data, but we only want to
#    keep the setosa and virginica. Use an R pipeline and appropriate function
#    in tidyverse to do this. Name the new dataset as iris2.
iris2 <- iris %>%
  filter(Species %in% c("setosa", "virginica"))
# b) (3pts) Print rows 46 to 55 of the iris2 dataset.
print(iris2[46:55,])
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 46          4.8         3.0          1.4         0.3    setosa
## 47          5.1         3.8          1.6         0.2    setosa
## 48          4.6         3.2          1.4         0.2    setosa
## 49          5.3         3.7          1.5         0.2    setosa
## 50          5.0         3.3          1.4         0.2    setosa
## 51          6.3         3.3          6.0         2.5 virginica
## 52          5.8         2.7          5.1         1.9 virginica
## 53          7.1         3.0          5.9         2.1 virginica
## 54          6.3         2.9          5.6         1.8 virginica
## 55          6.5         3.0          5.8         2.2 virginica
```

```r
# c) (7pts) Use iris2 data, and at 0.05 significance level, test if the average
#    sepal width of Setosa is significantly higher than the average sepal width
#    of Virginica. Use the RMarkdown report format to write the H0 and Ha,
```

```r
#    and the decision in R Markdown.
# Conducting a two-sample t-test
t_test_result <- t.test(Sepal.Width ~ Species, data = iris2)
t_test_result
```

```
##
##  Welch Two Sample t-test
##
## data:  Sepal.Width by Species
## t = 6.4503, df = 95.547, p-value = 4.571e-09
## alternative hypothesis: true difference in means between group setosa and group virginica is not equa
## 95 percent confidence interval:
##  0.3142808 0.5937192
## sample estimates:
##    mean in group setosa mean in group virginica
##                   3.428                   2.974
```

The 4.571e-09 p-value is significantly lower than the 0.05 significance level. Therefore, the H_0 must be rejected. Based on the results of the t-test, there is sufficient evidence against the claim that the average sepal width of Setosa is significantly higher than the average sepal width of Virginica.