Coding assignement 5 - wrangling text data

Noah Foilb

9/29/2020

Step 0: Load the dplyr library.

# Load the dplyr library  
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

Step 1: Download the txt document “Mandela\_Nobel\_Peace\_Prize\_Address.txt” from Blackboard to your desktop, then read it into R using readLines(). Store it in a vector called txt\_doc.

# Read in the txt doc  
txt\_doc <- readLines("Mandela\_Nobel\_Peace\_Prize\_Address.txt")

Step 2: Display the first 5 “lines” (= paragraphs) of this document.

# Using head to display the first five paragraphs  
head(txt\_doc,5)

## [1] "Nobel Peace Prize Address" "by Nelson Mandela"   
## [3] "December 10, 1993" "Oslo, Norway"   
## [5] "Your Majesty the King,"

Step 3: Find the total number of lines and the total number of characters. (Recall that a “line” is not a sentence, but is actually a “paragraph”; that is, a sequence of characters ending with he new line symbol “” that is stroed when you type the Enter key.)

# Using Length function to find the amount of lines  
length(txt\_doc)

## [1] 57

# Find the amount of characters in each line using nchar then summing them together using sum  
sum(nchar(txt\_doc))

## [1] 9422

Step 4: Use the “seven step process” that we copied here for you directly from the class notes to transform your character vector of lines (txt\_doc) into a data frame of individual words called df1. Just rn this code block as it is. You do not have to create any new code for this step.

t2 <- unlist(strsplit(x = txt\_doc, split = " ")) # 4 - separate into words at each blank space  
t3 <- unlist(strsplit(x = t2, split = "\t")) # 5 - separate at each tab = backslash t  
t4 <-stringr::str\_remove\_all(t3, "[-.,:/()]") # 6 - remove specific special character  
t5 <- tolower(t4) # 7 - convert all uppercase letters to lowercase  
df1 <- as.data.frame(t5) # 8 - convert character vector to a data frame  
df1 <- df1 %>% rename(word = t5) # 9 - change the name of the one variable to word  
df1 <- df1 %>% filter(word != "") # 10 - filter out blank words

Step 5: Display the number of words in this data frame and the mean word length (round this to 2 decimal places). Then display the first 20 words in the data frame. (These should be the first 20 words from the original .txt document since the data frame has not been sorted in any way yet.)

# Call df1 and use pipe   
df1 %>%  
   
 # Call Summarize   
 summarize(  
   
 # Set Number of words   
 Number\_of\_Words = n(),  
   
 # Set Mean world length and round the mean amount of characters  
 Mean\_Word\_Length = round(mean(nchar(word)),2))

## Number\_of\_Words Mean\_Word\_Length  
## 1 1643 4.66

# Display the first 20 values  
head(df1,20)

## word  
## 1 nobel  
## 2 peace  
## 3 prize  
## 4 address  
## 5 by  
## 6 nelson  
## 7 mandela  
## 8 december  
## 9 10  
## 10 1993  
## 11 oslo  
## 12 norway  
## 13 your  
## 14 majesty  
## 15 the  
## 16 king  
## 17 your  
## 18 royal  
## 19 highness  
## 20 honourable

Step 6: Create a new data frame df2 that starts with df1 and includes a new column indicating the length of each word. Then create a second block of code that displays a table listing the number of words of each length. This table should be ordered from longest to shortest words. (Although these directions utilize the word “table”, do not use the table() function from R in your code.)

# Set df1 to df2  
df2 <- df1 %>%  
   
 # Create length of word  
 mutate(Length\_of\_word = nchar(word))  
  
# Display df2  
head(df2)

## word Length\_of\_word  
## 1 nobel 5  
## 2 peace 5  
## 3 prize 5  
## 4 address 7  
## 5 by 2  
## 6 nelson 6

# Call df2  
df2 %>%  
   
 # Group by gth of word   
 group\_by(Length\_of\_word) %>%  
   
 # Summarize group cpunt  
 summarize(group\_count = n(),.groups = 'drop') %>%  
   
 # Reorder table  
 arrange(desc(group\_count))

## # A tibble: 15 x 2  
## Length\_of\_word group\_count  
## <int> <int>  
## 1 2 329  
## 2 3 319  
## 3 4 249  
## 4 5 191  
## 5 6 144  
## 6 7 139  
## 7 8 99  
## 8 9 51  
## 9 10 40  
## 10 1 34  
## 11 11 23  
## 12 12 14  
## 13 13 7  
## 14 14 3  
## 15 16 1

Step 7: Create a new data frame named df3 that starts with df1 but has only one row for each distinct word along with columns indicating the number of times the word occurs and the number of characters in the word. The rows in this data frame should be ordered by frequency from highest to lowest. Then display a chart that lists the 20 most frequently occurring words with their frequencies and lengths.

# Call df3  
df3 <- df1 %>%  
   
 # Create length of word  
 mutate(length\_of\_word = nchar(word)) %>%   
   
 # Group by word  
 group\_by(word) %>%  
   
 # Create frequancy and call summarize  
 summarize(frequency\_of\_occurrence = n(),.groups = 'drop') %>%   
   
 # Create length   
 mutate(length = nchar(word)) %>%   
   
 # Re arrange  
 arrange(desc(frequency\_of\_occurrence))