\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Part1) Strings**

**Code Section 1.a:**

file <- "https://people.bu.edu/kalathur/datasets/mlk.txt"

words <- scan(file,what = character())

-------------------------------------------------------------------------------

#a)

#using str\_subset method with special expression

str\_subset(words,"[:punct:]")

#Alternatively

#1

#using str\_detect functions from stringr

# words[str\_detect(words,"[:punct:]")==TRUE]

#2

#using general expression method

# grep("[[:punct:]]",words,value = TRUE)

**Console section 1.a:**

> file <- "https://people.bu.edu/kalathur/datasets/mlk.txt"

> words <- scan(file,what = character())

Read 288 items

#a)

> #using str\_subset method with special expression

> str\_subset(words,"[:punct:]")

[1] "today," "friends," "moment," "dream." "dream." "creed:" "self-evident:" "equal." "slave-owners"

[10] "brotherhood." "Mississippi," "state," "oppression," "justice." "character." "today." "Alabama," "governor’s"

[19] "nullification," "brothers." "today." "exalted," "low," "plain," "straight," "revealed," "together."

> #Alternatively

> #1

> #using str\_detect functions from stringr

> # words[str\_detect(words,"[:punct:]")==TRUE]

> #2

> #using general expression method

> # grep("[[:punct:]]",words,value = TRUE)

-----------------------------------------------------------------------------------------------

**Code Section 1.b:**

#b)

#using str\_replace method

replace\_punct <- str\_replace\_all(words,"[:punct:]","")

replace\_punct

#Alternatively, we can use gsub method

# gsub("[[:punct:][:blank:]]+", " ",words)

#now converting all words to lower case

new\_words <- str\_to\_lower(replace\_punct)

#alternatively we can use tolower method as well

# tolower(replace\_punct)

**Console section 1.b:**

#b)

> #using str\_replace method

> replace\_punct <- str\_replace\_all(words,"[:punct:]","")

> replace\_punct

[1] "I" "say" "to" "you" "today" "my" "friends" "that" "in" "spite"

[11] "of" "the" "difficulties" "and" "frustrations" "of" "the" "moment" "I" "still"

[21] "have" "a" "dream" "It" "is" "a" "dream" "deeply" "rooted" "in"

[31] "the" "American" "dream" "I" "have" "a" "dream" "that" "one" "day"

[41] "this" "nation" "will" "rise" "up" "and" "live" "out" "the" "true"

[51] "meaning" "of" "its" "creed" "We" "hold" "these" "truths" "to" "be"

[61] "selfevident" "that" "all" "men" "are" "created" "equal" "I" "have" "a"

[71] "dream" "that" "one" "day" "on" "the" "red" "hills" "of" "Georgia"

[81] "the" "sons" "of" "former" "slaves" "and" "the" "sons" "of" "former"

[91] "slaveowners" "will" "be" "able" "to" "sit" "down" "together" "at" "a"

[101] "table" "of" "brotherhood" "I" "have" "a" "dream" "that" "one" "day"

[111] "even" "the" "state" "of" "Mississippi" "a" "desert" "state" "sweltering" "with"

[121] "the" "heat" "of" "injustice" "and" "oppression" "will" "be" "transformed" "into"

[131] "an" "oasis" "of" "freedom" "and" "justice" "I" "have" "a" "dream"

[141] "that" "my" "four" "children" "will" "one" "day" "live" "in" "a"

[151] "nation" "where" "they" "will" "not" "be" "judged" "by" "the" "color"

[161] "of" "their" "skin" "but" "by" "the" "content" "of" "their" "character"

[171] "I" "have" "a" "dream" "today" "I" "have" "a" "dream" "that"

[181] "one" "day" "the" "state" "of" "Alabama" "whose" "governors" "lips" "are"

[191] "presently" "dripping" "with" "the" "words" "of" "interposition" "and" "nullification" "will"

[201] "be" "transformed" "into" "a" "situation" "where" "little" "black" "boys" "and"

[211] "black" "girls" "will" "be" "able" "to" "join" "hands" "with" "little"

[221] "white" "boys" "and" "white" "girls" "and" "walk" "together" "as" "sisters"

[231] "and" "brothers" "I" "have" "a" "dream" "today" "I" "have" "a"

[241] "dream" "that" "one" "day" "every" "valley" "shall" "be" "exalted" "every"

[251] "hill" "and" "mountain" "shall" "be" "made" "low" "the" "rough" "places"

[261] "will" "be" "made" "plain" "and" "the" "crooked" "places" "will" "be"

[271] "made" "straight" "and" "the" "glory" "of" "the" "Lord" "shall" "be"

[281] "revealed" "and" "all" "flesh" "shall" "see" "it" "together"

>

> #Alternatively, we can use gsub method

> # gsub("[[:punct:][:blank:]]+", " ",words)

>

>

> #now converting all words to lower case

> new\_words <- str\_to\_lower(replace\_punct)

>

> #alternatively we can use tolower method as well

> # tolower(replace\_punct)

-------------------------------------------------------------------------------------------------------------

**Code section 1.c:**

#c)

# getting top 5 numbers of words involves 3 steps

#1. table functions count the number of each words present

#2. sort function sort he words based on their frequency in increasing order

#3 decreasing() method arrage the words based decreasing order of their

#frequencies and finally seling top 5 words.

top5.words <- sort(table(new\_words),decreasing = TRUE)[1:5]

top5.words

**Console section 1.c:**

#c)

> # getting top 5 numbers of words involves 3 steps

> #1. table functions count the number of each words present

> #2. sort function sort he words based on their frequency in increasing order

> #3 decreasing() method arrage the words based decreasing order of their

> #frequencies and finally seling top 5 words.

> top5.words <- sort(table(new\_words),decreasing = TRUE)[1:5]

> top5.words

new\_words

the of a and be

17 15 14 14 11

--------------------------------------------------------------------------------------------------

**Code setion 1.d:**

#d)

# in this solution, I am going to find the lengths of each word and then find

# the frequency of each length types.

length.of.words <- str\_length(new\_words)

frequency.of.word.length <- table(length.of.words)

#alternatively we can also use nchar() method to check the frequencies.

#table(nchar(new\_words))

#now showing frequencies using bar plot.

par(mar=c(5,5,2,2))

barplot(frequency.of.word.length,xlab = "Length of words",ylab = "Frequency",

main = "Frequency of length of words",col = rainbow(14),ylim = c(0,60))

**Console section1.d:**

#d)

> # in this solution, I am going to find the lengths of each word and then find

> # the frequency of each length types.

> length.of.words <- str\_length(new\_words)

> frequency.of.word.length <- table(length.of.words)

>

> #alternatively we can also use nchar() method to check the frequencies.

> #table(nchar(new\_words))

>

> #now showing frequencies using bar plot.

> par(mar=c(5,5,2,2))

> barplot(frequency.of.word.length,xlab = "Length of words",ylab = "Frequency",

+ main = "Frequency of length of words",col = rainbow(14),ylim = c(0,60))

**Plot section 1.d:**

A picture containing diagram, screenshot, text, colorfulness

Description automatically generated

------------------------------------------------------------------------------------------------------------------

**Code section 1.e:**

#e)

#for this,I will calculate the length of each words and then find the maximum value

# i.e and find the word/s which have that maximum length.

longest.words <- new\_words[str\_length(new\_words)==max(str\_length(new\_words))]

longest.words

**Console section 1.e:**

#e)

> #for this,I will calculate the length of each words and then find the maximum value

> # i.e and find the word/s which have that maximum length.

> longest.words <- new\_words[str\_length(new\_words)==max(str\_length(new\_words))]

> longest.words

[1] "interposition" "nullification"

------------------------------------------------------------------------------------------------------------------

**Code section 1.f:**

#f)

#for this I will be suing str\_detect method as follows

str\_subset(new\_words,"^c")

#alternatively

#new\_words[str\_detect(new\_words,"^c")]

**Console section 1.f:**

> #f)

> #for this I will be suing str\_detect method as follows

> str\_subset(new\_words,"^c")

[1] "creed" "created" "children" "color" "content" "character" "crooked"

>

> #alternatively

> #new\_words[str\_detect(new\_words,"^c")]

----------------------------------------------------------------------------------------------------------------------

**Code section 1.g:**

#g)

# again for this, i will be using str\_detect method as follows

str\_subset(new\_words,"r$")

# there are words that ends with r and occurs more than one time.We can take one

# words using unique method

#Alternatively

#new\_words[str\_detect(new\_words,"r$")]

**Console section 1.g:**

#g)

> # again for this, i will be using str\_detect method as follows

> str\_subset(new\_words,"r$")

[1] "former" "former" "together" "four" "color" "their" "their" "character" "together" "together"

>

> # there are words that ends with r and occurs more than one time.We can take one

> # words using unique method

>

> #Alternatively

> #new\_words[str\_detect(new\_words,"r$")]

----------------------------------------------------------------------------------------------------------

**Code section 1.h:**

#h)

#for this we can combine solution in f and g.

# using str\_subset method.

str\_subset(new\_words,"^c(.)\*r$")

#alternatively

# \\b word boundry \\w\* represent any words with zero or more characters.

# matches.br <- str\_detect(new\_words, "\\bc\\w\*r\\b")

# new\_words[matches.br]

**Console section 1.h**

> #h)

> #for this we can combine solution in f and g.

> # using str\_subset method.

>

> str\_subset(new\_words,"^c(.)\*r$")

[1] "color" "character"

> #alternatively

> # \\b word boundry \\w\* represent any words with zero or more characters.

> # matches.br <- str\_detect(new\_words, "\\bc\\w\*r\\b")

> # new\_words[matches.br]

>

-----------------------------------------------------------------------------------------------------------------------------

**Last part of Part one:**

**Code Part:**

#last part of Part 1

stopfile <- "https://people.bu.edu/kalathur/datasets/stopwords.txt"

stopwords <- scan (stopfile, what=character())

# removing the stop words

new.words <- subset(new\_words,!new\_words %in% stopwords)

length(new.words)

#now finding the top 5 frequent words

# using same steps as answer in c.

new.top5.words <- sort(table(new.words),decreasing = TRUE)[1:5]

new.top5.words

# finding frequency of word lengths

frequency.word.length.2 <- table(nchar(new.words))

par(mar=c(5,5,5,2))

barplot(frequency.word.length.2,xlab = "Length of word",ylab = "Frequency",

main = " Frequency of different word lengths", col = rainbow(12),

ylim = c(0,50))

**Console part:**

> #last part of Part 1

> stopfile <- "https://people.bu.edu/kalathur/datasets/stopwords.txt"

> stopwords <- scan(stopfile, what=character())

Read 176 items

> # removing the stop words

> new.words <- subset(new\_words,!new\_words %in% stopwords)

> length(new.words)

[1] 139

>

> #now finding the top 5 frequent words

> # using same steps as answer in c.

> new.top5.words <- sort(table(new.words),decreasing = TRUE)[1:5]

> new.top5.words

new.words

dream day one shall made

11 6 6 4 3

>

> # finding frequency of word lengths

> frequency.word.length.2 <- table(nchar(new.words))

> par(mar=c(5,5,5,2))

> barplot(frequency.word.length.2,xlab = "Length of word",ylab = "Frequency",

+ main = " Frequency of different word lengths", col = rainbow(12),

+ ylim = c(0,50))

>

**Plot part:**

A picture containing diagram, plot, line, screenshot

Description automatically generated

-------------------------------------------------------------------------------------------------------------------------

**Part2) Data Wrangling**

**Code section 2.a:**

temp.data <- read.csv("/Users/kokildhakal/Desktop/STUDY/BU/7.CS544/Module6/usa\_daily\_avg\_temps.csv”)

#a)

usaDailyTemps <- as\_tibble(temp.data)

head(usaDailyTemps)

**Console section 2.a:**

> temp.data <- read.csv("/Users/kokildhakal/Desktop/STUDY/BU/7.CS544/Module6/usa\_daily\_avg\_temps.csv")

>

>

> #a)

> usaDailyTemps <- as\_tibble(temp.data)

> head(usaDailyTemps)

# A tibble: 6 × 6

state city month day year avgtemp

<chr> <chr> <int> <int> <int> <dbl>

1 Alabama Birmingham 1 1 1995 50.7

2 Alabama Birmingham 1 1 1996 56.8

3 Alabama Birmingham 1 1 1997 60.9

4 Alabama Birmingham 1 1 1998 35.6

5 Alabama Birmingham 1 1 1999 41

6 Alabama Birmingham 1 1 2000 59

>

------------------------------------------------------------------------------------------------------------------------

Cose section2.b:

max.temp <- usaDailyTemps|>

group\_by(year) |>

summarise(maximum\_Temp=max(avgtemp))

#maximu temperature by Year

max.temp

#plotting

plot\_ly(y=max.temp$maximum\_Temp,x=max.temp$year,type = "scatter",

color = max.temp$maximum\_Temp,colors = "Paired",mode ="markers") |>

layout(title = "Maximum Temperature in each from 1995 to 2015",

xaxis=list(title="Year"),

yaxis=list(title="Temperature"))

**Console section 2.b:**

> max.temp <- usaDailyTemps|>

+ group\_by(year) |>

+ summarise(maximum\_Temp=max(avgtemp))

> #maximu temperature by Year

> max.temp

# A tibble: 21 × 2

year maximum\_Temp

<int> <dbl>

1 1995 104.

2 1996 104.

3 1997 101.

4 1998 103

5 1999 100.

6 2000 102.

7 2001 104.

8 2002 103.

9 2003 106.

10 2004 101

# ℹ 11 more rows

# ℹ Use `print(n = ...)` to see more rows

>

>

> #plotting

> plot\_ly(y=max.temp$maximum\_Temp,x=max.temp$year,type = "scatter",

+ color = max.temp$maximum\_Temp,colors = "Paired",mode ="markers") |>

+ layout(title = "Maximum Temperature in each from 1995 to 2015",

+ xaxis=list(title="Year"),

+ yaxis=list(title="Temperature"))

>

Plot section 2.b:

A picture containing text, screenshot, line, font

Description automatically generated

--------------------------------------------------------------------------------------------------------------------

**Code section 2.c:**

#c)

#finding maximum temperature by states

max.temp.by.states <- usaDailyTemps |>

group\_by(state)|>

summarise(Maximum\_Temp=max(avgtemp))

#maximum temperature by states

max.temp.by.states

plot\_ly(y=max.temp.by.states$Maximum\_Temp,x=max.temp.by.states$state,type = "scatter",

color =max.temp.by.states$state,mode ="markers",colors = "Paired") |>

layout(title = "Maximum Temperature in each State",yaxis=list(title="Temperature"))

**Console section 2.c:**

> #c)

> #finding maximum temperature by states

> max.temp.by.states <- usaDailyTemps |>

+ group\_by(state)|>

+ summarise(Maximum\_Temp=max(avgtemp))

> #maximum temperature by states

> max.temp.by.states

# A tibble: 50 × 2

state Maximum\_Temp

<chr> <dbl>

1 Alabama 91.5

2 Alaska 79.5

3 Arizona 108.

4 Arkansas 101.

5 California 103.

6 Colorado 94.7

7 Connecticut 89.8

8 Delaware 89.7

9 Florida 92.8

10 Georgia 97.7

# ℹ 40 more rows

# ℹ Use `print(n = ...)` to see more rows

>

> plot\_ly(y=max.temp.by.states$Maximum\_Temp,x=max.temp.by.states$state,type = "scatter",

+ color =max.temp.by.states$state,mode ="markers",colors = "Paired") |>

+ layout(title = "Maximum Temperature in each State",yaxis=list(title="Temperature"))

**Plot section 2.c:**

A picture containing text, screenshot, line, plot

Description automatically generated

----------------------------------------------------------------------------------------------------

**Code section 2.d:**

#d)

#filtering data for Boston Only

bostonDailyTemps <- usaDailyTemps |>

filter(city=="Boston")

head(bostonDailyTemps)

**console section 2.d:**

**> #d)**

> #filtering data for Boston Only

> bostonDailyTemps <- usaDailyTemps |>

+ filter(city=="Boston")

> head(bostonDailyTemps)

# A tibble: 6 × 6

state city month day year avgtemp

<chr> <chr> <int> <int> <int> <dbl>

1 Massachusetts Boston 1 1 1995 38.5

2 Massachusetts Boston 1 1 1996 34.1

3 Massachusetts Boston 1 1 1997 10

4 Massachusetts Boston 1 1 1998 14.2

5 Massachusetts Boston 1 1 1999 21.7

6 Massachusetts Boston 1 1 2000 34.8

>

---------------------------------------------------------------------------------------

**Code section 2.e:**

#e)

#finding average monthly temperatures of Boston

montly.avg.temp.boston <- bostonDailyTemps|>

group\_by(month)|>

summarise(avg\_temp=mean(avgtemp))

#average montly temperature in Boston

montly.avg.temp.boston

plot\_ly(y=montly.avg.temp.boston$avg\_temp,x=montly.avg.temp.boston$month,

type = "scatter",color = montly.avg.temp.boston$month,colors=rainbow(12))|>

layout (title="Average Montly Temperatures in Boston",

yaxis=list(title="Temperature"),

xaxis=list(title=” Month”))

**Console section 2.e:**

#e)

> #finding average monthly temperatures of Boston

> montly.avg.temp.boston <- bostonDailyTemps|>

+ group\_by(month)|>

+ summarise(avg\_temp=mean(avgtemp))

>

> #average montly temperature in Boston

> montly.avg.temp.boston

# A tibble: 12 × 2

month avg\_temp

<int> <dbl>

1 1 29.8

2 2 31.5

3 3 37.6

4 4 47.1

5 5 57.6

6 6 66.1

7 7 73.6

8 8 71.7

9 9 65.1

10 10 54.7

11 11 44.9

12 12 35.0

>

> plot\_ly(y=montly.avg.temp.boston$avg\_temp,x=montly.avg.temp.boston$month,

+ type = "scatter",color = montly.avg.temp.boston$month,colors=rainbow(12))|>

+ layout(title="Average Montly Temperatures in Boston",

+ yaxis=list(title="Temperature"),

xaxis=list(title=”Month”))

**Plot section 2.e:**

A graph with green and blue dots

Description automatically generated with low confidence

The End:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*8