IE6700 ASSIGNMENT 6

Group 02

Names:

Karan Paresh

Rashmi Nambiar Pappinisseri Puthenveettil

Problem 1

Task 3

a.

Code

#**Task1**

```
#a Establish spark connection

library(sparklyr)
library(dplyr)
library(DBI)
library(tidyr)

system("java -version")
Sys.setenv(JAVA_HOME = "C:/Program Files/Java/jdk1.8.0_231")

#b Load the text file
spark_install("2.1.0")
sc <- spark_connect(master = "local", version = "2.1.0")
myoldman_path <- paste0("", getwd(), "/My_old_man.txt")
Oldman <-  spark_read_text(sc, "oldman", myoldman_path)

#c Remove empty lines
all_lines<- Oldman %>%
  filter(nchar(line) > 0)

head(all_lines,20)

#d Remove punctuations
all_lines<- all_lines %>%
  mutate(line = regexp_replace(line, "[_\"\'():;,.!?\\-]", " "))

head(all_lines,10)

#e Separate each word using Spark API ft_tokenizer
all_words <- all_lines %>%
  ft_tokenizer(input_col= "line", output_col= "word_list")

head(all_words, 4)

#f Remove stop words (e.g., I, me, my, …)
all_words<- all_words %>%
  ft_stop_words_remover(input_col = "word_list", output_col= "wo_stop_words")
head(all_words, 4)

#g Unnesting the tokens into their own row using explode; filtering the result with
ncahr(word) > 1
all_words <- all_words %>%
```

```
  mutate(word = explode(wo_stop_words)) %>%
  select(word) %>%
  filter(nchar(word) > 1)
head(all_words, 4)

#h Cache the result into Spark memory using compute()
all_words <- all_words %>%
  compute("all_words")

#Task2
#a Generate a list of (word, count) in descending order of count

word_count <- all_words %>%
  group_by(word) %>%
  tally() %>%
  arrange(desc(n))

word_count
word_count<- copy_to(sc, word_count)

#b Create a list of the first 20 words with counts
first_20 <- dbGetQuery(sc, "select * from word_count limit 20")
first_20

#c How many distinct words are there in the list
total_count=dbGetQuery(sc, "select distinct(count(*)) from word_count")
total_count
```

b.

Total number of distinct words in list = 933

The list of the first 20 words with counts -

	word	n
1	old	74
2	man	69
3	going	34
4	around	33
5	like	27
6	back	25
7	get	25
8	big	23
9	one	22
10	went	21
11	came	21
12	looking	20
13	kzar	19
14	horse	19
15	got	19
16	way	19
17	said	18
18	go	17
19	george	16
20	say	16