Homework 7

[Ra-Zakee Muhammad]

Due Tuesday 10/26/2021

Classmates/other resources consulted: [type answer here]

Note this assignment is only worth 50 points, and is half the length of a normal assignment.

Note that to get this assignment to knit correctly, you may have to comment out some of the text in some of the questions, like in a previous assignment

```
library(tidyverse)
library(nycflights13)
```

Question 1 (6 points)

> a. **Explain the difference between str_view and str_view_all, and give an example of inputs on which these two functions will produce different outputs.**

str_view shows the first match; str_view_all shows all the matches.

```
string_01 <- "happy sappy nappy"

str_view(string = string_01, "pp" )
```

```
happy sappy nappy
```

```
str_view_all(string = string_01, "pp" )
```

```
happy sappy nappy
```

> b. **Explain the difference between str_view and str_detect. Give an example of a situation in which str_view is the correct function to use, and give an example of a situation in which str_detect is the correct function to use.**

str_detect out puts a logical value depending on whether or not a string pattern is present in the string argument. str_view shows the first instance of a pattern in the string argument.

use str_detect if you want to filter a character factor but the string values contain more than just the sub string you are looking for. Use str_view if you want to find a specific location of a sub string if it appears at all in a larger text.

Question 2 (16 points)

> **Consider the following list of example words, which are a subset of the entire words list.**

```
words_example <- words[seq(1, length(words), 50)]
words_example
```

```
##  [1] "a"         "arm"       "boat"      "Christmas" "course"    "during"
##  [7] "family"    "game"      "hope"      "know"      "man"       "next"
## [13] "park"      "process"   "resource"  "sheet"     "standard"  "terrible"
## [19] "type"      "whole"
```

> **For each part below, give a regular expression that will find the described patterns, and display these patterns using str_view. You should only display the words in which the pattern was found.**

> **(Be sure the regular expressions you give are general enough to work on the whole words data set, even though I'm not asking you to do that here. For example, If I ask for words containing the letter x, you should not simple match the string "next", instead your command should be general enough to find any word containing the letter "x", even though there are no other such strings in the words_example list.)**

> a. **Words containing the substring "ar"**

```
words_example %>%
  str_view("ar")
```

```
a
arm
boat
Christmas
course
during
family
game
hope
know
man
next
park
process
resource
sheet
standard
terrible
type
whole
```

> b. **Words containing the letter s followed by a vowel.**

```
words_example %>%
  str_view("s(a|e|i|o|u)")
```

```
a
arm
boat
Christmas
course
during
family
game
hope
know
man
next
park
```

```
process

resource

sheet

standard

terrible

type

whole
```

c. **Words containing the letter s followed by a character that is not a vowel**

```
words_example %>%
  str_view("s[^aeiou]")
```

```
a

arm

boat

Christmas

course

during

family

game

hope

know

man

next

park

process

resource

sheet

standard

terrible

type

whole
```

d. **Words that start with the letter s**

```
words_example %>%
  str_view("^s")
```

```
a

arm

boat

Christmas

course

during

family

game

hope

know

man

next
```

```
park

process

resource

sheet

standard

terrible

type

whole
```

e. **Words that end with the letter s**

```
words_example %>%
  str_view("s$")
```

```
a

arm

boat

Christmas

course

during

family

game

hope

know

man

next

park

process

resource

sheet

standard

terrible

type

whole
```

f. **Words that have a repeated letter that appears twice in a row**

```
words_example %>%
  str_view("(.)\\1")
```

```
a

arm

boat

Christmas

course

during

family

game

hope

know

man
```

```
next
park
process
resource
sheet
standard
terrible
type
whole
```

g. **Words that have a repeated letter that appears twice anywhere in the string but not consecutively**

```
words_example %>%
  str_view("(.)(.+)\\1")
```

```
a
arm
boat
Christmas
course
during
family
game
hope
know
man
next
park
process
resource
sheet
standard
terrible
type
whole
```

h. **Words that end with the letter “e” but not with the string “pe”**

```
words_example %>%
  str_view("([^(p)])e$")
```

```
a
arm
boat
Christmas
course
during
family
game
hope
know
```

```
man

next

park

process

resource

sheet

standard

terrible

type

whole
```

Question 3 (3 points)

In a particular data set, the string “`^`” is used to represent a dollar amount that is unknown. Write a regular expression that finds this substring. You can use the following example string to check your answer.

```
string <- "The salad costs $10, the dessert costs $^$, and the drink costs $4."
```

```
string%>%
  str_view("[$]\\^[$]")
```

```
The salad costs $10, the dessert costs $^$, and the drink costs $4.
```

Question 4 (3 points)

Look up, using any resources you’d like, how to create a string for a regular expression that matches a single backslash . Use your regular expression to match the backslash in the following string which, when displayed, says files.

```
str <- "files\\folder"
writeLines(str)
```

```
## files\folder
```

```
str %>%
  str_view('\\\\')
```

```
files\folder
```

Question 5 (12 points)

> **Genomic data can be specified as a string of the characters A, C, G, and T.**

> a. **Restriction enzymes are used to cut DNA strands at locations when specific sequences are identified. The BsaWI enzyme recognizes sequences that look like WCCGGW, where W could stand for A or T (a *weak* enzyme). Would the BsaWI enzyme find such a sequence in the following DNA example? Use regular expressions to find out. Your regular expression should be clear and simple, using as few characters as possible.**

```
DNAex1 <- "ACCGCTAGCTCGCTAGATCGATCGGCGGGCTCTAGATCGATCGGCTAGATAGCTTCCGGAATCGTCGTCTA"
```

```
DNAex1 %>%
  str_view("(.)(.)\\2(.)\\3\\1")
```

```
ACCGCTAGCTCGCTAGATCGATCGGCGGGCTCTAGATCGATCGGCTAGATAGCTTCCGGAATCGTCGTCTA
```

it wouldnt find such a sequence in thi example

> b. **In fact, this enzyme can recognize this sequence when it appears either forwards or backwards, that is, it also recognizes the sequence WGGCCW. How many times would the BsaWI enzyme cut the following DNA sequence?**

```
DNAex2 <- "ACCGCTAGCTTCGTAGATCGCTCGCAGGCCATGGGCTCTAGATCGATCGGCTAGATAGCTTCCGGAATCGTCGTCTA"
```

```
DNAex2 %>%
  str_view_all("(.)(.)\\2(.)\\3\\1")
```

```
ACCGCTAGCTTCGTAGATCGCTCGCAGGCCATGGGCTCTAGATCGATCGGCTAGATAGCTTCCGGAATCGTCGTCTA
```

once

> c. **Huntington’s disease can be characterized by having the three nucleotide sequence CAG repeating 36 or more times in a row in a particular gene. Write a command to check whether the following DNA sequence meets this criteria:**

```
DNAex3 <- "ACGTCGCTAGCTAGCTCGCTAGATACGCTCCCCCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAG
CAGCAGCAGCAGCGCGGAT"
```

```
DNAex3 %>%
  str_view("(CAG){36}")
```

```
ACGTCGCTAGCTAGCTCGCTAGATACGCTCCCCCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGC
```

it doesnt match this criteria

> d. **Consider the (artificially generated) genomic patient data in the attached patient_genetic_data.csv file. Import this data, and determine which of the patients have the genetic marker for Huntington’s Disease.**

```
gene_data <- read_csv("patient_genetic_data.csv")
```

```
## Rows: 20 Columns: 2
```

```
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): PatientDNA
## dbl (1): PatientID
```

```
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
gene_data%>%
  filter(str_detect(PatientDNA,"(CAG){36}")) %>%
  select(PatientID)
```

```
## # A tibble: 3 × 1
##   PatientID
##       <dbl>
## 1         7
## 2        11
## 3        18
```

Question 6 (10 points)

> **For this question, use the starwars tibble.**

> a. (3 points) **Give a tibble containing all characters whose skin color is mottled, that is, all characters whose skin_color description contains “mottle”.**

```
starwars %>% filter(str_detect(skin_color,"mottle"))
```

```
## # A tibble: 2 × 14
##   name    height  mass hair_color skin_color  eye_color birth_year sex   gender
##   <chr>    <int> <dbl> <chr>      <chr>       <chr>          <dbl> <chr> <chr>
## 1 Ackbar     180    83 none       brown mott… orange            41 male  mascul…
## 2 Nute G…    191    90 none       mottled gr… red               NA male  mascul…
## # … with 5 more variables: homeworld <chr>, species <chr>, films <list>,
## #   vehicles <list>, starships <list>
```

> b. (3 points) **Give a tibble containing all characters with multiple hair colors.**

```
starwars %>% filter(str_detect(hair_color,","))
```

```
## # A tibble: 3 × 14
##   name     height  mass hair_color  skin_color eye_color birth_year sex   gender
##   <chr>     <int> <dbl> <chr>       <chr>      <chr>          <dbl> <chr> <chr>
## 1 Owen La…    178   120 brown, grey light      blue              52 male  mascu…
## 2 Obi-Wan…    182    77 auburn, wh… fair       blue-gray         57 male  mascu…
## 3 Wilhuff…    180    NA auburn, gr… fair       blue              64 male  mascu…
## # … with 5 more variables: homeworld <chr>, species <chr>, films <list>,
## #   vehicles <list>, starships <list>
```

> c. (4 points) **Are there any characters with digits appearing in their name that are not Droids? Are there any Droids that do not have digits in their names? Give commands that would answer this question even for a data set with several thousand characters. That is, your answer should not rely on looking through all entries in some tibbles, because if there were more characters that tibble might be too big to look through in its entirety.**

```
starwars %>%
  filter(( str_detect(name,"\\d") & species != "Droid") |
         ( !str_detect(name,"\\d") & species == "Droid"))
```

```
## # A tibble: 0 × 14
## # … with 14 variables: name <chr>, height <int>, mass <dbl>, hair_color <chr>,
## #   skin_color <chr>, eye_color <chr>, birth_year <dbl>, sex <chr>,
## #   gender <chr>, homeworld <chr>, species <chr>, films <list>,
## #   vehicles <list>, starships <list>
```

REMINDER: FINAL PROJECT TOPIC PROPOSAL ALSO DUE 10/26

The final project topic proposal is due at the same time as this homework. The details can be found on Sakai, and you can submit your answers directly within Gradescope.