Knowledge Quiz 2

Solveig Senf

Please answer the following questions, render a pdf, and submit both the qmd and pdf on Moodle by **11 PM on Sun May 4**. Please also leave a copy of your qmd in your Submit folder on the St. Olaf RStudio server.

Guidelines:

- No consulting with anyone else
- You may use only materials from this class (our class webpage, links on Moodle, our 3 online textbooks, files posted to the RStudio server, your personal notes from class)
- No online searches or use of large language models like ChatGPT

Pledge:

I pledge my honor that on this quiz I have neither given nor received assistance not explicitly approved by the professor and that I an aware of no dishonest work.

- type your name here to acknowledge the pledge: Solveig Senf
- OR
- place an X here if you intentionally are not signing the pledge: _____

```
library(tidyverse)
library(rvest)
library(tidytext)

park_data <- read_csv("~/SDS264/Data/park_data_KQ2.csv")
```

National Park Data

`park_data` is a 54x3 tibble containing information scraped from national park webpages for a past SDS264 final project. A few notes about the 3 columns:

- `park_code` is a 4-letter code used as a key when merging files
- `address` is comprised of 4 pieces (described from *right* to *left*):
  - the final piece (following a comma and space) is a zip code (usually 5 digits but sometimes 5 digits then a dash then 4 more digits)
  - the 2nd to last piece is the state (an abbreviation with 2 capital letters)
  - the 3rd to last piece is the city (usually one or two words long, occasionally 3; always follows two or more spaces)
  - the first piece is the street address (often a number and a street, but will always be followed by at least two spaces)
- `activities` is a string of activities offered at each park, where activities are separated by commas

Quiz Questions

Please answer the following questions using your knowledge of strings, regular expressions, and text analysis. Please use `stringr` functions as much as possible, aim for efficient code, and use good style to make your code as readable as possible!

Section 1

1. Find the subset of all `address` entries that contain a direction (north, south, east, or west).

```
str_subset(park_data$address, "East|North|South|West") #| is "or"
```

```
[1] "52 West Headquarters Drive   Torrey UT, 84775"
[2] "64 Grinnell Drive  West Glacier MT, 59936"
[3] "20 South Entrance Road  Grand Canyon AZ, 86023"
[4] "800 East Lakeshore Drive   Houghton MI, 49931"
[5] "38050 Highway 36 East  Mineral CA, 96063"
[6] "55210 238th Avenue East   Ashford WA, 98304"
[7] "5000 East Entrance Road   Paicines CA, 95043"
[8] "3655 U.S. Highway 211  East Luray VA, 22835"
[9] "360 Hwy 11 East  International Falls MN, 56649"
```

2. Produce a tibble showing how often each of the 4 directions from (1) occurs among the 54 `address` entries. Which direction is most common?

```
directions_tibble <- park_data |>
  mutate(has_direction = str_detect(park_data$address, "East|North|South|West")) |>
  filter(has_direction == "TRUE") |> #filters addresses with directions
  summarize(east = sum(str_count(address, "East")), #summarizes how many times each directio
            west = sum(str_count(address, "West")),
            north = sum(str_count(address, "North")),
            south = sum(str_count(address, "South")))

directions_tibble
```

```
# A tibble: 1 x 4
   east  west north south
  <int> <int> <int> <int>
1     6     2     0     1
```

East is the most common direction.

3. Create a new tibble containing only national parks in Alaska (AK) and Hawaii (HI).

```
ak_hi_parks <- park_data |>
  mutate(hi_ak = str_detect(park_data$address, "AK|HI")) |> #state is Hawaii or Alaska
  filter(hi_ak == "TRUE") |>
  select(-hi_ak)

ak_hi_parks
```

```
# A tibble: 10 x 3
   park_code address                                                  activities
   <chr>     <chr>                                                    <chr>
 1 DENA      Mile 237 Highway 3   Denali Park AK, 99755               Arts and Cu~
 2 GAAR      101 Dunkel St   Fairbanks AK, 99701                      Camping, Ba~
 3 GLBA      1 Park Road   Gustavus AK, 99826                         Arts and Cu~
 4 HALE      Haleakala National Park Route 378  Kula HI, 96790       Camping, Ba~
 5 HAVO      1 Crater Rim Drive   Hawaii National Park HI, 96718      Arts and Cu~
 6 KATM      1000 Silver Street  King Salmon AK, 99613                Boating, Ca~
 7 KEFJ      411 Washington Street   Seward AK, 99664                 Astronomy, ~
 8 KOVA      171 3rd Ave   Kotzebue AK, 99752                         Boating, Ca~
 9 LACL      1 Park Place   Port Alsworth AK, 99653                   Astronomy, ~
10 WRST      Mile 106.8 Richardson Highway  Copper Center AK, 99573 Arts and Cu~
```

Section 2

4. Build a tibble which adds 4 columns to `park_data`:

- street_address
- city
- state
- zip_code

Hint: sometimes you can extract more than you want, and then remove the extra stuff...

```
park_data_new <- park_data |>
  mutate(state = str_extract(address, "[A-Z][A-Z]"), #two upper-case letters
         zip = str_extract(address, "\\d{5}"), #five numbers in a row
         street_address = str_extract(address, "^.*  "),
         city_start = str_extract(address, "  .*"), #extracts the part of the address after t
         city = str_remove(city_start, "[A-Z][A-Z], \\d{5}.*$")) |> #removes the state and zi
  select(-city_start)

park_data_new
```

```
# A tibble: 54 x 7
   park_code address                activities state zip   street_address city
   <chr>     <chr>                  <chr>      <chr> <chr> <chr>          <chr>
 1 ACAD      25 Visitor Center Road~ Arts and ~ ME    04609 "25 Visitor C~ "   ~
 2 BADL      25216 Ben Reifel Road ~ Auto and ~ SD    25216 "25216 Ben Re~ "   ~
 3 BIBE      1 Panther Junction   B~ Auto and ~ TX    79834 "1 Panther Ju~ "   ~
 4 BISC      9700 SW 328th Street  ~ Boating, ~ SW    33033 "9700 SW 328t~ "  H~
 5 BLCA      9800 Highway 347  Mont~ Astronomy~ CO    81401 "9800 Highway~ "  M~
 6 BRCA      Highway 63 Bryce Canyo~ Astronomy~ UT    84764 "Highway 63 B~ "  B~
 7 CARE      52 West Headquarters D~ Arts and ~ UT    84775 "52 West Head~ "   ~
 8 CAVE      727 Carlsbad Caverns H~ Astronomy~ NM    88220 "727 Carlsbad~ "   ~
 9 CHIS      1901 Spinnaker Drive  ~ Astronomy~ CA    93001 "1901 Spinnak~ "   ~
10 CONG      100 National Park Road~ Camping, ~ SC    29061 "100 National~ "   ~
# i 44 more rows
```

Section 3

5. Create a new column in `park_data` which records the total number of activities in each park, then sort the parks from most activities to least.

```
park_data |>
  mutate(activity_count = str_count(activities, ", ")) |>
  arrange(-activity_count)
```

```
# A tibble: 54 x 4
   park_code address                                    activities activity_count
   <chr>     <chr>                                      <chr>               <int>
 1 GRSA      11999 State Highway 150  Mosca CO, 81146  Arts and ~             55
 2 GRTE      103 Headquarters Loop   Moose WY, 83012   Arts and ~             53
 3 OLYM      3002 Mount Angeles Road  Port Angeles WA~ Astronomy~             53
 4 YELL      2 Officers Row  Yellowstone National Par~ Arts and ~             52
 5 VOYA      360 Hwy 11 East  International Falls MN,~ Arts and ~             47
 6 LAVO      38050 Highway 36 East  Mineral CA, 96063  Auto and ~             46
 7 ACAD      25 Visitor Center Road   Bar Harbor ME, ~ Arts and ~             45
 8 EVER      40001 State Road 9336   Homestead FL, 33~ Auto and ~             45
 9 WRST      Mile 106.8 Richardson Highway  Copper Ce~ Arts and ~             45
10 GLAC      64 Grinnell Drive  West Glacier MT, 59936 Arts and ~             44
# i 44 more rows
```

6. Pick off all of the activities that end in "ing"; we'll refer to these as "verb activities". Produce a count of the number of parks where each "verb activity" appears, and print the "verb activities" and their counts in order from most parks to fewest. (Note that you should consider something like "Group Camping" as different from "RV Camping" or just plain "Camping".) Your answer should look like the tibble below:

```
# A tibble: 57 × 2
   verb_activity              n
   <chr>                  <int>
 1 Hiking                    50
 2 Shopping                  46
 3 Stargazing                34
 4 Wildlife Watching         31
 5 Camping                   30
 6 Scenic Driving            26
 7 Horse Trekking            23
 8 Canoe or Kayak Camping    22
 9 Group Camping             22
10 Paddling                  21
#   47 more rows```
```

Hint: if you produce a list where each element in the list is a vector (with differing numbers of strings), you can use `unlist` to produce a single character vector

```
activities <- park_data |>
  select(activities) |>
  mutate(activities = str_split(activities, ", ")) #splits the list of activites

activities <- as.list(activities) |> #makes a list of activities
  unlist(activities)

verb_activities <- as_tibble(activities) |> #turns the activites into a tibble
  mutate(activity = str_extract_all(value, ".*ing$")) |>  #extracts -ing activities
  filter(activity != "character(0)") |> #removes non-ing activities
  group_by(activity) |>
  count(activity) |>
  arrange(-n)
```

Use your tibble from (6) to answer Questions (7)-(8).

7. Print all the "verb activities" that have a capital letter / lower case letter combination that repeats later in the phrase (e.g. "Gh" appears twice).

```
repeat_upper_lower <- str_subset(verb_activities$activity, "([A-Z][a-z]).*\\1")
print(repeat_upper_lower)
```

```
[1] "Car or Front Country Camping" "Canoe or Kayak Camping"
```

8. Print all the "verb activities" that have the same consonant appear twice in a row.

```
repeat_consonant <- str_subset(verb_activities$activity, "([^aeiou])\\1")
print(repeat_consonant)
```

```
 [1] "Shopping"                  "Paddling"
 [3] "Horse Trekking"            "Cross-Country Skiing"
 [5] "Swimming"                  "Off-Trail Permitted Hiking"
 [7] "Stand Up Paddleboarding"   "Freshwater Swimming"
 [9] "Saltwater Swimming"        "Downhill Skiing"
[11] "Auto Off-Roading"          "Dog Sledding"
[13] "ATV Off-Roading"           "Pool Swimming"
```