Knowledge Quiz 2

Gracia Larsen-Schmidt

Please answer the following questions, render a pdf, and submit both the qmd and pdf on moodle by 11 PM on Thurs Nov 14. 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: Gracia Larsen-Schmidt
- OR
- place an X here if you intentionally are not signing the pledge:

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

# park_data <- read_csv("~/Sds 264 F24/Class/Data/park_data_KQ2.csv")

park_data <- read_csv("~/Desktop/gitSDS264_F24/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, "North | South | East | West")
```

```
[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?

```
park_data |>
  filter(str_detect(address, "North | South | East | West")) |>
  mutate(direction = str_extract(address, "North | South | East | West")) |>
  count(direction) |>
  arrange(direction)
```

```
# A tibble: 3 x 2
  direction     n
  <chr>     <int>
1 " East "      6
2 " South "     1
3 " West"       2
```

- "East" is the most common

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

```
park_data |>
  filter(str_detect(address, "AK, | HI,"))
```

```
# 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 <- park_data |>
  mutate(street_address = str_extract(address, ".+  "),
         city = str_extract(address, "  .+,"),
         state = str_extract(address, "[A-Z][A-Z]"),
         zip_code = str_extract(address, ", .+"),) |>
mutate(street_address = str_replace(street_address, "  ", ""),
       city = str_replace(city, " [A-Z][A-Z],", ""),
       city = str_replace(city, "^ +", ""),
       zip_code = str_replace(zip_code, ", ", ""))


park_data
```

```
# A tibble: 54 x 7
   park_code address              activities street_address city  state zip_code
   <chr>     <chr>                <chr>      <chr>          <chr> <chr> <chr>
 1 ACAD      25 Visitor Center R~ Arts and ~ "25 Visitor C~ Bar ~ ME    04609
 2 BADL      25216 Ben Reifel Ro~ Auto and ~ "25216 Ben Re~ Inte~ SD    57750
 3 BIBE      1 Panther Junction ~ Auto and ~ "1 Panther Ju~ Big ~ TX    79834
 4 BISC      9700 SW 328th Stree~ Boating, ~ "9700 SW 328t~ Home~ SW    33033
 5 BLCA      9800 Highway 347  M~ Astronomy~ "9800 Highway~ Mont~ CO    81401
 6 BRCA      Highway 63 Bryce Ca~ Astronomy~ "Highway 63 B~ Bryce UT    84764
 7 CARE      52 West Headquarter~ Arts and ~ "52 West Head~ Torr~ UT    84775
 8 CAVE      727 Carlsbad Cavern~ Astronomy~ "727 Carlsbad~ Carl~ NM    88220
 9 CHIS      1901 Spinnaker Driv~ Astronomy~ "1901 Spinnak~ Vent~ CA    93001
10 CONG      100 National Park R~ Camping, ~ "100 National~ Hopk~ SC    29061
# i 44 more rows
```

Use your new tibble from (4) to answer Questions (5) and (6).

5. Print the subset of `street_address` entries where the numerical part is 1000 or greater.

```
str_subset(park_data$street_address, "\\d\\d\\d\\d")
```

```
 [1] "25216 Ben Reifel Road "     "9700 SW 328th Street"
 [3] "9800 Highway 347"           "1901 Spinnaker Drive "
 [5] "6947 Riverview Road"        "40001 SR-9336 "
 [7] "40001 State Road 9336 "     "11999 State Highway 150"
 [9] "74485 National Park Drive " "1000 Silver Street"
[11] "38050 Highway 36 East"      "34840 Hwy 160 "
[13] "55210 238th Avenue East "   "3002 Mount Angeles Road"
[15] "5000 East Entrance Road "   "1111 Second Street "
[17] "1000 US Hwy 36 "            "3693 S Old Spanish Trail "
[19] "47050 Generals Highway "    "3655 U.S. Highway 211"
[21] "26611 US Highway 385 "      "9039 Village Drive "
```

6. Arrange `city` names from longest to shortest.

```
park_data |>
  select(city) |>
  mutate(city_length = str_count(city, ".")) |>
  arrange(desc(city_length))
```

```
# A tibble: 54 x 2
   city                      city_length
   <chr>                           <int>
 1 Yellowstone National Park          25
 2 Big Bend National Park             22
 3 Hawaii National Park               20
 4 International Falls                19
 5 Twentynine Palms                   16
 6 Petrified Forest                   16
 7 Port Alsworth                      13
 8 Sedro-Woolley                      13
 9 Crescent City                      13
10 Copper Center                      13
# i 44 more rows
```

Section 3

7. 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 <- park_data |>
  mutate(num_activities = 1 + str_count(activities, "\\b[A-Za-z\\s\\-\\(\\)]+,")) |>
  arrange(desc(num_activities))

park_data
```

```
# A tibble: 54 x 8
   park_code address              activities street_address city  state zip_code
   <chr>     <chr>                <chr>      <chr>          <chr> <chr> <chr>
 1 GRSA      11999 State Highway~ Arts and ~ "11999 State ~ Mosca CO    81146
 2 GRTE      103 Headquarters Lo~ Arts and ~ "103 Headquar~ Moose WY    83012
 3 OLYM      3002 Mount Angeles ~ Astronomy~ "3002 Mount A~ Port~ WA    98362
 4 YELL      2 Officers Row  Yel~ Arts and ~ "2 Officers R~ Yell~ WY    82190
 5 VOYA      360 Hwy 11 East  In~ Arts and ~ "360 Hwy 11 E~ Inte~ MN    56649
 6 LAVO      38050 Highway 36 Ea~ Auto and ~ "38050 Highwa~ Mine~ CA    96063
 7 ACAD      25 Visitor Center R~ Arts and ~ "25 Visitor C~ Bar ~ ME    04609
 8 EVER      40001 State Road 93~ Auto and ~ "40001 State ~ Home~ FL    33034
 9 WRST      Mile 106.8 Richards~ Arts and ~ "Mile 106.8 R~ Copp~ AK    99573
10 GLAC      64 Grinnell Drive  ~ Arts and ~ "64 Grinnell ~ West~ MT    59936
# i 44 more rows
# i 1 more variable: num_activities <dbl>
```

8. 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

```r
list_verbs <- str_extract_all(park_data$activities, "\\b[A-Za-z\\s\\-\\(\\)\\/]+ing\\b")

verbs <- as.tibble((unlist(list_verbs))) |>
  mutate(verb_activity = value) |>
  count(verb_activity) |>
  arrange(desc(n))
```

```
Warning: `as.tibble()` was deprecated in tibble 2.0.0.
i Please use `as_tibble()` instead.
i The signature and semantics have changed, see `?as_tibble`.
```

```r
verbs
```

```
# A tibble: 72 x 2
   verb_activity            n
   <chr>                <int>
 1 Camping                 53
 2 Hiking                  52
 3 Shopping                51
 4 Wildlife Watching       48
 5 Backcountry Camping     46
 6 Birdwatching            43
 7 Backcountry Hiking      39
 8 Front-Country Hiking    39
 9 Biking                  38
10 Fishing                 37
# i 62 more rows
```

- My counts aren't the same as your example, but when I check my regular expression using str_view, I don't see any obvious errors in how I am extracting the verbs.

```r
str_view(park_data$activities, "\\b([A-Za-z\\s\\-\\(\\)\\/])+ing\\b")
```

```
 [1] | Arts and Culture, Craft Demonstrations, Live Music, Auto and ATV, <Auto Off-Roading>,
 [2] | Arts and Culture, Cultural Demonstrations, Auto and ATV, <Scenic Driving>, <Biking>, <
 [3] | Astronomy, <Stargazing>, <Biking>, <Road Biking>, <Boating>, <Camping>, <Backcountry C
 [4] | Arts and Culture, Auto and ATV, <Scenic Driving>, Astronomy, <Biking>, <Mountain Bikin
 [5] │ Arts and Culture, Cultural Demonstrations, Astronomy, <Stargazing>, <Biking>, <Boating
 [6] │ Auto and ATV, <Scenic Driving>, Astronomy, <Stargazing>, <Biking>, <Road Biking>, <Boa
 [7] │ Arts and Culture, Cultural Demonstrations, Astronomy, <Stargazing>, <Biking>, <Boating
 [8] │ Auto and ATV, <Scenic Driving>, Astronomy, <Stargazing>, <Biking>, <Road Biking>, <Boa
 [9] │ Arts and Culture, Auto and ATV, <ATV Off-Roading>, <Auto Off-Roading>, <Scenic Driving
[10] │ Arts and Culture, Cultural Demonstrations, Live Music, Auto and ATV, <Scenic Driving>
[11] │ Auto and ATV, <Scenic Driving>, Astronomy, <Stargazing>, <Biking>, <Road Biking>, <Boa
[12] │ Arts and Culture, Cultural Demonstrations, Astronomy, <Stargazing>, <Biking>, <Road Bi
[13] │ Auto and ATV, <Scenic Driving>, Astronomy, <Stargazing>, <Biking>, <Road Biking>, <Can
[14] │ Astronomy, <Stargazing>, <Boating>, <Motorized Boating>, <Sailing>, Boat Tour, <Campin
[15] │ Astronomy, <Stargazing>, <Biking>, <Boating>, <Motorized Boating>, <Camping>, <Backcou
[16] │ Arts and Culture, Cultural Demonstrations, Auto and ATV, <Scenic Driving>, <Biking>,
[17] │ Auto and ATV, <Scenic Driving>, Astronomy, <Stargazing>, <Camping>, <Backcountry Campi
[18] │ Astronomy, <Stargazing>, <Boating>, Boat Tour, <Camping>, <Backcountry Camping>, <Cano
[19] │ Arts and Culture, Cultural Demonstrations, Auto and ATV, <Scenic Driving>, Astronomy,
[20] │ Arts and Culture, Live Music, Auto and ATV, <Scenic Driving>, Astronomy, <Stargazing>
... and 34 more
```

- I also checked a few of of the individually, and they seem to be right, but it's possible my regular expression for checking also contains a mistake.

```
park_data |>
  mutate(Camping = str_detect(activities, "\\bCamping\\b")) |>
  count(Camping)
```

```
# A tibble: 2 x 2
  Camping     n
  <lgl>   <int>
1 FALSE       1
2 TRUE       53
```

```
park_data |>
  mutate(Hiking = str_detect(activities, "\\bHiking\\b")) |>
  count(Hiking)
```

```
# A tibble: 2 x 2
  Hiking     n
  <lgl>  <int>
1 FALSE      2
2 TRUE      52
```

```
park_data |>
  mutate(Wildlife_watching = str_detect(activities, "\\bWildlife Watching\\b")) |>
  count(Wildlife_watching)
```

```
# A tibble: 2 x 2
  Wildlife_watching     n
  <lgl>             <int>
1 FALSE                 6
2 TRUE                 48
```

```
park_data |>
   mutate(Biking = str_detect(activities, "\\bBiking\\b")) |>
  count(Biking)
```

```
# A tibble: 2 x 2
  Biking     n
  <lgl>  <int>
1 FALSE     16
2 TRUE      38
```

- I also checked by just looking through the park_data dataset, and these numbers seemed to be correct, unless I am interpreting the question incorrectly?

Use your tibble from (8) to answer Questions (9)-(10).

9. 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).

```
str_subset(verbs$verb_activity, "([A-Z])([a-z]).*\\1\\2")
```

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

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

```
str_subset(verbs$verb_activity, "([^AEIOUaeiou])\\1")
```

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