# Knowledge Quiz 2

**Gwynnie Hayes**

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: Gwynnie Hayes
- OR
- place an X here if you intentionally are not signing the pledge: _____

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

park_data <- read_csv("~/Desktop/15/SDS264/quizzes/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).

```
park_data |>
  filter(str_detect(address, "North|South|East|West"))
```

```
# A tibble: 9 x 3
  park_code address                                        activities
  <chr>     <chr>                                          <chr>
1 CARE      52 West Headquarters Drive   Torrey UT, 84775  Arts and Culture, Cu~
2 GLAC      64 Grinnell Drive  West Glacier MT, 59936      Arts and Culture, Cu~
3 GRCA      20 South Entrance Road  Grand Canyon AZ, 86023 Arts and Culture, Cu~
4 ISRO      800 East Lakeshore Drive   Houghton MI, 49931  Arts and Culture, As~
5 LAVO      38050 Highway 36 East  Mineral CA, 96063       Auto and ATV, Scenic~
```

```
6 MORA      55210 238th Avenue East   Ashford WA, 98304    Auto and ATV, Scenic~
7 PINN      5000 East Entrance Road   Paicines CA, 95043   Astronomy, Stargazin~
8 SHEN      3655 U.S. Highway 211  East Luray VA, 22835    Auto and ATV, Scenic~
9 VOYA      360 Hwy 11 East  International Falls MN, 56649 Arts and Culture, Cu~
```

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 |>
  mutate(address_direction = str_extract(address, "North|South|East|West")) |>
  count(address_direction)
```

```
# A tibble: 4 x 2
  address_direction     n
  <chr>             <int>
1 East                  6
2 South                 1
3 West                  2
4 <NA>                 45
```

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...

```
new_park <- park_data |>
  mutate(street_address = str_extract(address, "^.+?(?=\\s{2})"),
         city = str_extract(address, "\\s{2}[A-z ]+"),
         city = str_remove(city, "^(\\s+)"),
         city = str_remove(city, "[A-Z]{2}$"),
         state = str_extract(address, "[A-Z]{2}"),
         zip_code = str_extract(address, "\\d{5}")) |>
  select(c("park_code", "activities", "street_address", "city", "state", "zip_code"))

head(new_park)
```

```
# A tibble: 6 x 6
  park_code activities                       street_address city  state zip_code
  <chr>     <chr>                            <chr>          <chr> <chr> <chr>
1 ACAD      Arts and Culture, Cultural Demo~ 25 Visitor Ce~ "Bar~ ME    04609
2 BADL      Auto and ATV, Scenic Driving, A~ 25216 Ben Rei~ "Int~ SD    25216
3 BIBE      Auto and ATV, Scenic Driving, A~ 1 Panther Jun~ "Big~ TX    79834
4 BISC      Boating, Motorized Boating, Sai~ 9700 SW 328th~ "Hom~ SW    33033
5 BLCA      Astronomy, Stargazing, Camping,~ 9800 Highway ~ "Mon~ CO    81401
6 BRCA      Astronomy, Stargazing, Biking, ~ Highway 63 Br~ "Bry~ UT    84764
```

## 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, ",") + 1)
```

```
# A tibble: 54 x 4
   park_code address                                      activities activity_count
   <chr>     <chr>                                        <chr>               <dbl>
 1 ACAD      25 Visitor Center Road   Bar Harbor ME, ~ Arts and ~              46
 2 BADL      25216 Ben Reifel Road   Interior SD, 577~ Auto and ~              21
 3 BIBE      1 Panther Junction   Big Bend National P~ Auto and ~              24
 4 BISC      9700 SW 328th Street  Homestead FL, 33033 Boating, ~              24
 5 BLCA      9800 Highway 347  Montrose CO, 81401      Astronomy~              26
 6 BRCA      Highway 63 Bryce Canyon National Park  B~ Astronomy~              30
 7 CARE      52 West Headquarters Drive   Torrey UT, ~ Arts and ~              42
 8 CAVE      727 Carlsbad Caverns Highway   Carlsbad ~ Astronomy~              17
 9 CHIS      1901 Spinnaker Drive   Ventura CA, 93001  Astronomy~              39
10 CONG      100 National Park Road   Hopkins SC, 290~ Camping, ~              26
# 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:

#| eval: FALSE

## A tibble: 57 × 2

verb_activity n 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

```
park_activities <- park_data |>
  select(activities) |>
  mutate(activities = str_split(activities, ",\\s")) |>
  unlist(recursive = TRUE) |>
  tibble(verb_activity = _) |>
  count(verb_activity) |>
  filter(str_detect(verb_activity, "ing$")) |>
  arrange(desc(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).

```
park_activities |>
  filter(str_detect(verb_activity, "([A-Z][a-z]).*\\1"))
```

```
# A tibble: 2 x 2
  verb_activity                    n
  <chr>                        <int>
1 Car or Front Country Camping    36
2 Canoe or Kayak Camping          31
```

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

```
park_activities |>
  filter(str_detect(verb_activity, "([^aeiou])\\1"))
```

```
# A tibble: 14 x 2
   verb_activity                  n
   <chr>                      <int>
 1 Shopping                      51
 2 Paddling                      28
 3 Horse Trekking                27
 4 Cross-Country Skiing          19
 5 Swimming                      14
 6 Off-Trail Permitted Hiking    13
 7 Stand Up Paddleboarding        9
 8 Freshwater Swimming            5
 9 Saltwater Swimming             5
10 Auto Off-Roading               3
11 Downhill Skiing                3
12 ATV Off-Roading                2
13 Dog Sledding                   2
14 Pool Swimming                  1
```