PS: 8 NAME:

Strings I

The following questions concern the R object called `fruit` that is stored in the `stringr` package.

```
library(stringr)
head(fruit)
```

```
[1] "apple"       "apricot"     "avocado"     "banana"      "bell pepper"
[6] "bilberry"
```

1. What data structure is `fruit` and what is its length?

2. Which fruits have the longest and shortest names?

3. Replace the entry `"currant"` with `"red currant"`.

4. Form a single string that starts with `"My favorite fruits are"` and then lists all of the fruits, separated by a comma and a space. The end of the list should have `"and"` before the last fruit and end with a period.

5. Form a string like the following but with another fruit and emoji[1] of your choice. Note the fruit name should come directly from `fruit`.

```
[1] "🍋 lemon 🍋"
```

[1] Each emoji is defined by an 8 digit code called Unicode. See https://r4ds.hadley.nz/strings.html#other-special-characters.

6. Print all of the fruit names containing berry.

7. How many fruit names contain the letter `"o"`?

8. Which fruits names contain the most `"o"`s?

9. How many fruits feature two `"a"`s separated by one other character?

Strings II

These questions deal with the `babynames` data set inside the package of the same name. Read about the data set with `?babynames`.

```r
# install.packages("babynames")
library(babynames)
babynames
```

```
# A tibble: 1,924,665 × 5
    year sex   name          n   prop
   <dbl> <chr> <chr>     <int>  <dbl>
 1  1880 F     Mary       7065 0.0724
 2  1880 F     Anna       2604 0.0267
 3  1880 F     Emma       2003 0.0205
 4  1880 F     Elizabeth  1939 0.0199
 5  1880 F     Minnie     1746 0.0179
 6  1880 F     Margaret   1578 0.0162
 7  1880 F     Ida        1472 0.0151
 8  1880 F     Alice      1414 0.0145
 9  1880 F     Bertha     1320 0.0135
10  1880 F     Sarah      1288 0.0132
# i 1,924,655 more rows
```

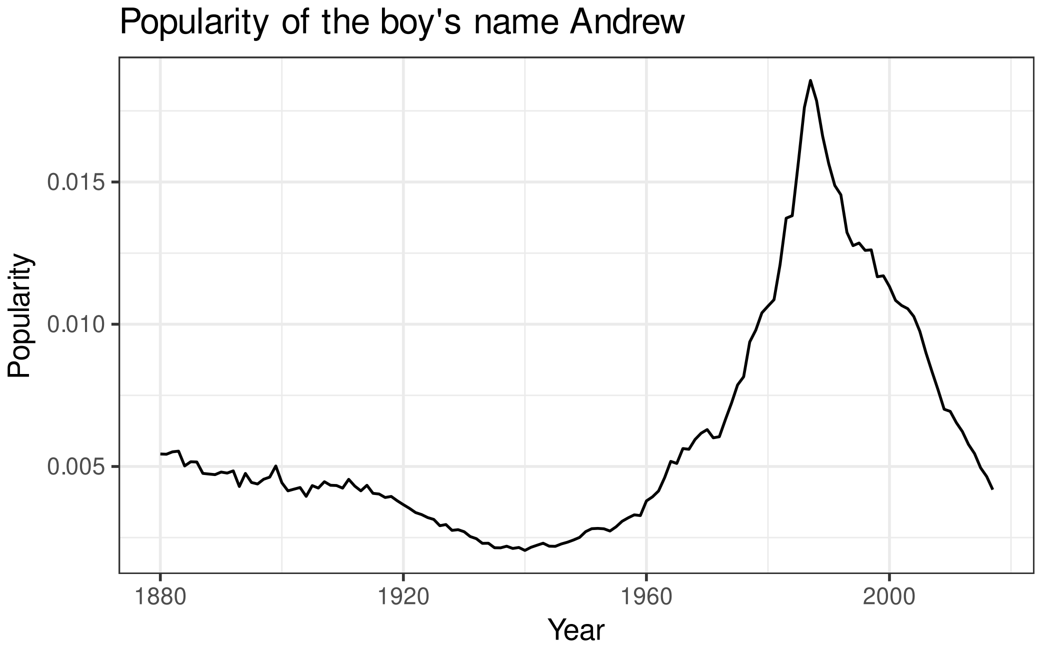
10. What does every row refer to, i.e. what is the unit of observation? What time range is covered by this data? How is `prop` calculated (what is the numerator and demoninator)?

11. In 2010, what proportion of boys names ended with a vowel? How about girls names? How does those proportions differ when calculated in 1910?

12. Narrow down the data frame to the rows with names that end in a diminuitive (e.g. “Joey” and “Juanita”). Do some research to gather a set of diminuitives used in languages found in American baby names.

13. How many distinct names rhyme with “Aiden”?

14. Create a plot like the following but for your name.



15. Create a plot that shows the relatively frequency of a name shared by a historic figure over time (e.g. “Barack”).

16. *Optional Challenge*: For each year since 2000, which names have been the most gender neutral?

17. *Optional Challenge*: Plot the proportion of diminuitive names over time.

---

18. R has a built in palette of colors that you can access with `colors()`. Note that many of the colors in that palette have numbered variants that are different darknesses. How many colors are in this palette after removing the numbered variants of a color?

The following questions deal with the chorus of the song, “Golden”, but Huntrix. Be sure they copy into R as a string that captures the new line characters. Provide the code to programmatically answer each question. There are many ways to answer these questions but `str_split()` might be helpful.

```r
x <- "We're goin' up, up, up, it's our moment
You know together we're glowing
Gonna be, gonna be golden
Oh, up, up, up, with our voices
\uc601\uc6d0\ud788 \uae68\uc9c8 \uc218 \uc5c6\ub294
Gonna be, gonna be golden"
```

19. How many lines are in the chorus?

20. How many times is the word `"up"`” used in the chorus?

21. How many distinct four letter words are used?

22. Replace each of the contractions with the expanded english equivalent.

23. What is the first word of each line?

24. How many words are in each line?

25. Provide a new string that writes the Korean word for “Golden” using the appropriate Unicode characters.

Text Analysis

Following a similar structure to the lecture, you will perform an elemenary exploratory analysis of a work of your choosing cataloged by Project Gutenberg (https://www.gutenberg.org/). You are welcome to search the website and read in the text file directly, or you can use the `gutenbergr` package in R and search the metadata file using `View(gutenberg_metadata)` in Positron.

Once you have selected a work to study, record the code you us to perform each of the following.

26. Read the data into R.

27. What is unit of observation in your text file and how many units are there (i.e. how many lines are there)?

28. Browse the text and note any unexpected syntax or structure. Clean the data to remove any rows that are not relevant to your analysis.

29. How long is the longest unit (likely a line) in terms of characters? How about in terms of words?

30. Create a data frame where each row is a single token. Then create a bar chart of the top 15 tokens, ordered in terms of their frequency.

31. Remove stop words then remake the previous plot.