HW4

Jadon Fowler jaf582 5778191

2022-11-03

Instructions

Download a copy of this markdown. Change the `author:` tag above to have your name and NAU's ID. Fill in the file with your solution to the proposed problems. Knit your final document to PDF and submit it through BBLearn in the assignment `[HW] Homework 4: web scrapping` by the end of the day on **Thursday, November 3 (11:59:00 PM)**.

Note: to knit your homework to PDF, you need to have MikTex installed. You can download and install MikTex from here. If you don't want to install MikTex, you can knit to HTML, open the file in the browser and print the page to PDF. You can also submit the `.Rmd` file.

For all the problems, please import the useful libraries in this chunk of code:

Problem 1: Reshaping data

Consider the following dataframe about the weather:

```
weather <- read.delim(file = "https://ramnathv.github.io/pycon2014-r/explore/data/weather.txt", strings

glimpse(weather)
```

```
## Rows: 22
## Columns: 35
## $ id      <chr> "MX000017004", "MX000017004", "MX000017004", "MX000017004", "M~
## $ year    <int> 2010, 2010, 2010, 2010, 2010, 2010, 2010, 2010, 2010, 2010, 20~
## $ month   <int> 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 10, 10, 11, 11~
## $ element <chr> "TMAX", "TMIN", "TMAX", "TMIN", "TMAX", "TMIN", "TMAX", "TMIN"~
## $ d1      <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ d2      <int> NA, NA, 273, 144, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ d3      <int> NA, NA, 241, 144, NA, NA, NA, NA, NA, NA, NA, NA, 286, 175, NA~
## $ d4      <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ d5      <int> NA, NA, NA, NA, 321, 142, NA, NA, NA, NA, NA, NA, NA, NA, 296,~
## $ d6      <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ d7      <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ d8      <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 290, 1~
## $ d9      <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ d10     <int> NA, NA, NA, NA, 345, 168, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ d11     <int> NA, NA, 297, 134, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ d12     <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ d13     <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 298, 1~
## $ d14     <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 299, 165, NA, ~
## $ d15     <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ d16     <int> NA, NA, NA, NA, 311, 176, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ d17     <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 280, 175, NA, NA, NA, ~
## $ d18     <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ d19     <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ d20     <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ d21     <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ d22     <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ d23     <int> NA, NA, 299, 107, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 264,~
## $ d24     <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ d25     <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 297, 1~
## $ d26     <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ d27     <int> NA, NA, NA, NA, NA, NA, 363, 167, 332, 182, NA, NA, NA, NA, NA~
## $ d28     <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ d29     <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 301, 180, NA, NA, 280,~
## $ d30     <int> 278, 145, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ d31     <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 254, 1~
```

The dataset contains the temperature for As we can see, we have many NAs in this dataset, but this may be due to the organization of variables/observations. Explain why this data is untidy.

> The columns prefixed with `d` should be rows and the element column should be two separate rows: TMIN & TMAX.

Then, write a code that converts the data into a tidy and clean dataframe. Make sure your dataset follows these guidelines:

- the dataframe is tidy;
- NA measurements for temperature are dropped;
- day of the month is a numerical data;
- sorted by year, month, and day, respectively.

```r
# tidy dataframe
w <- as.data.frame(pivot_wider(
  pivot_longer(
    weather,
    cols = starts_with("d"),
    names_to = "day",
    names_transform = list(day = readr::parse_number),
    values_to = "count",
    values_drop_na = TRUE
  ),
  names_from = element,
  values_from = count
))

glimpse(w)
```

```
## Rows: 33
## Columns: 6
## $ id    <chr> "MX000017004", "MX000017004", "MX000017004", "MX000017004", "MX0~
## $ year  <int> 2010, 2010, 2010, 2010, 2010, 2010, 2010, 2010, 2010, 2010, 2010~
## $ month <int> 1, 2, 2, 2, 2, 3, 3, 3, 4, 5, 6, 6, 7, 7, 8, 8, 8, 8, 8, 8, 8, 1~
## $ day   <dbl> 30, 2, 3, 11, 23, 5, 10, 16, 27, 27, 17, 29, 3, 14, 5, 8, 13, 23~
## $ TMAX  <int> 278, 273, 241, 297, 299, 321, 345, 311, 363, 332, 280, 301, 286,~
## $ TMIN  <int> 145, 144, 144, 134, 107, 142, 168, 176, 167, 182, 175, 180, 175,~
```

```r
# drop the null measurements


# convert day into numerical


#arrange by year, month, day
```

Problem 2: web scraping

We will scrape the page https://quotes.toscrape.com/ and create a dataframe to analyse authorship and tags of quotes in that page. Investigate the page a little bit. You'll notice that there is only 10 quotes in the page, but there is a `Next` button at the end, and the quotes continues on. We want to scrape 100 quotes (10 of these pages).

Let's create an empty dataframe to begin with. I completed this step for you, by creating `df_quotes` with one column (`quotes`).

```r
# creating an empty dataframe with 0 rows and 3 cols
df_quotes <- data.frame(matrix(nrow = 0, ncol = 1))

# assign column names to the dataframe
colnames(df_quotes) = c("quotes")
glimpse(df_quotes)
```

```
## Rows: 0
## Columns: 1
## $ quotes <lgl>
```

Step 1: use a for loop to scrape the 10 first pages of quotes.toscrape. In each page, there is an html tag `<div class=quote>`, which represents the quotes in the page. This `div` comes with the quote, the authors and the tags (when available).

Use the `rvest` functions to scrape these `div.quote` elements from each page. Once you have the list with the 10 codes from a page, use the `rbind` function to add the quotes into our `df_quotes` dataframe.

```r
for (x in 1:10) {
  url <- paste("https://quotes.toscrape.com/page/",x, sep = '')

  page <- read_html(url)
  quotes <- html_elements(page, ".quote")
  quotes <- quotes %>% html_text2()

  #binding the scraped quotes to our dataframe:
  df_quotes <- rbind(df_quotes, as.data.frame(quotes))
}
```

Step 2: Our `df_quotes` dataframe is one single column that contains quotes, author names and list of tags. We want to break this into three separate columns. Based on what we learned in this course so far, transform the `df_quotes` into a dataframe of dimensions `(100, 3)`, where the rows contain the 100 quotes from the page and the columns contain `quotes`, `authors`, and `tags`. A glimpse of this dataframe would be as following:

Rows: 100 Columns: 3 $ quotes ""The world as we have created it is a process of our thinking. It cannot be changed w… $ authors "Albert Einstein", "J.K. Rowling", "Albert Einstein", "Jane Austen", "Marilyn Monroe",… $ tags "change deep-thoughts thinking world", "abilities choices", "inspirational life live m…

Tips: `stringr` library may have useful functions for this problem!

```r
quotes <- str_split(df_quotes$quotes, '.\\"by ', simplify = TRUE)
author_tags <- quotes[,2]
authors <- str_split(author_tags, ' \\(about\\)\nTags: ', simplify = TRUE)
tags <- authors[,2]
quotes <- quotes[,1]
authors <- authors[,1]
df_quotes <- cbind.data.frame(quotes, authors, tags)
glimpse(df_quotes)
```

```
## Rows: 100
## Columns: 3
## $ quotes  <chr> ""The world as we have created it is a process of our thinking~
## $ authors <chr> "Albert Einstein", "J.K. Rowling", "Albert Einstein", "Jane Au~
## $ tags    <chr> "change deep-thoughts thinking world", "abilities choices", "i~
```

** Step 3:** Notice that the column `tags` has several values in each row. This is because one quote may have zero or more tags. Tags are separated by whitespace.

Reshape this dataframe to: (1) separate the tags into several columns (one column for each tag); (2) reshape these columns in a longer format; (3) remove all the rows that do not have any tags. A glimpse of this dataframe would be as follows:

Rows: 232 Columns: 3 $ quotes ""The world as we have created it is a process of our thinking. It cannot be changed w… $ authors "Albert Einstein", "Albert Einstein", "Albert Einstein", "Albert Einstein", "J.K. Rowl… $ tag "change", "deep-thoughts", "thinking", "world", "abilities", "choices", "inspirational…

Notice that the first quote from Albert Einstein has four tags (change, deep-thoughts, thinking, and world). Thus, "Albert Einstein" and the quote itself is repeat four times, one per tag.

Tips: You may want to review `stringr`, `tidyr`, and `dplyr` libraries to solve this problem.

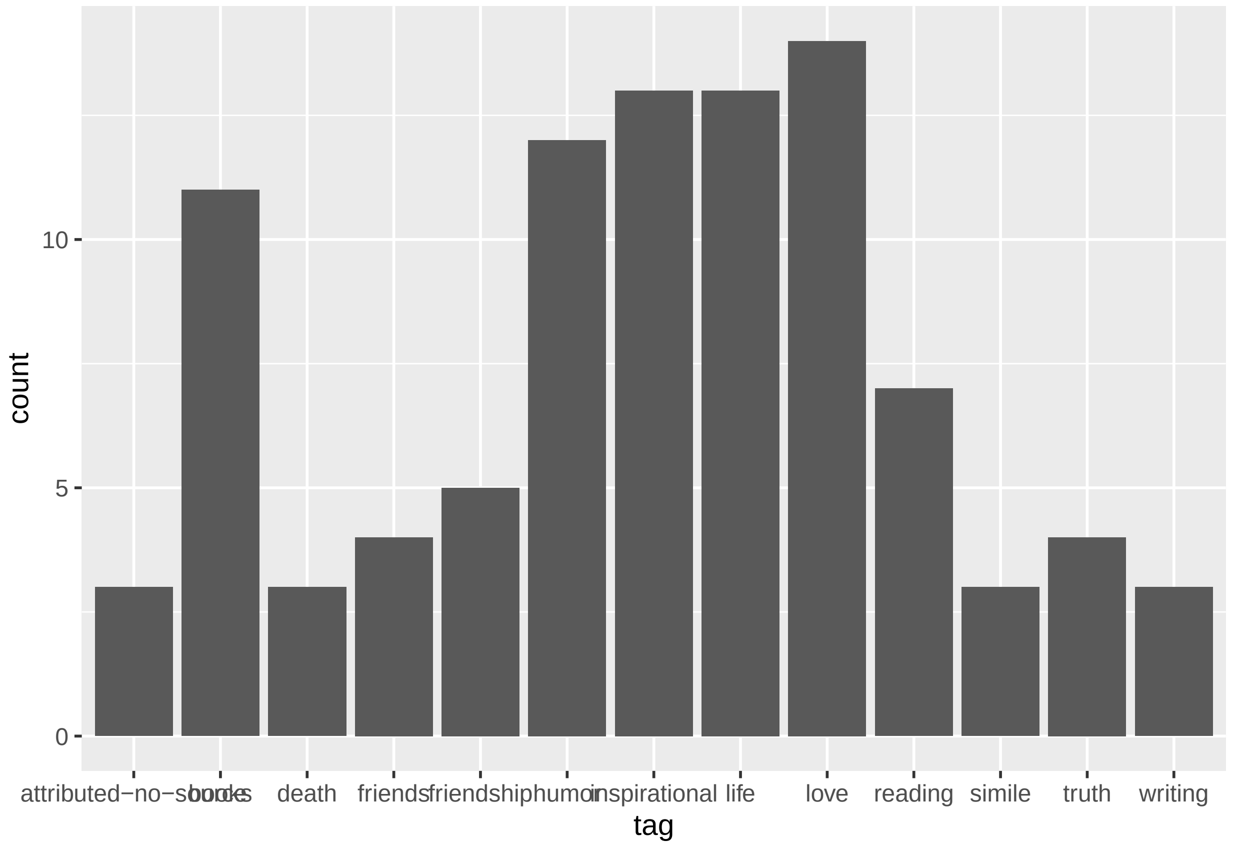
```r
tags <- str_split(df_quotes$tags, '\\s', simplify = TRUE)
df_quotes <- cbind.data.frame(df_quotes, tags)

df_quotes <- df_quotes %>% pivot_longer(cols=!c(1:3), values_to = "tag", values_drop_na = TRUE)

df_quotes <- df_quotes %>%
  filter(tag!="") %>%
  select(-tags, -name)
glimpse(df_quotes)
```

```
## Rows: 232
## Columns: 3
## $ quotes  <chr> ""The world as we have created it is a process of our thinking~
## $ authors <chr> "Albert Einstein", "Albert Einstein", "Albert Einstein", "Albe~
## $ tag     <chr> "change", "deep-thoughts", "thinking", "world", "abilities", "~
```

Step 4: Now that our dataset is ready, produce a visualization to show:

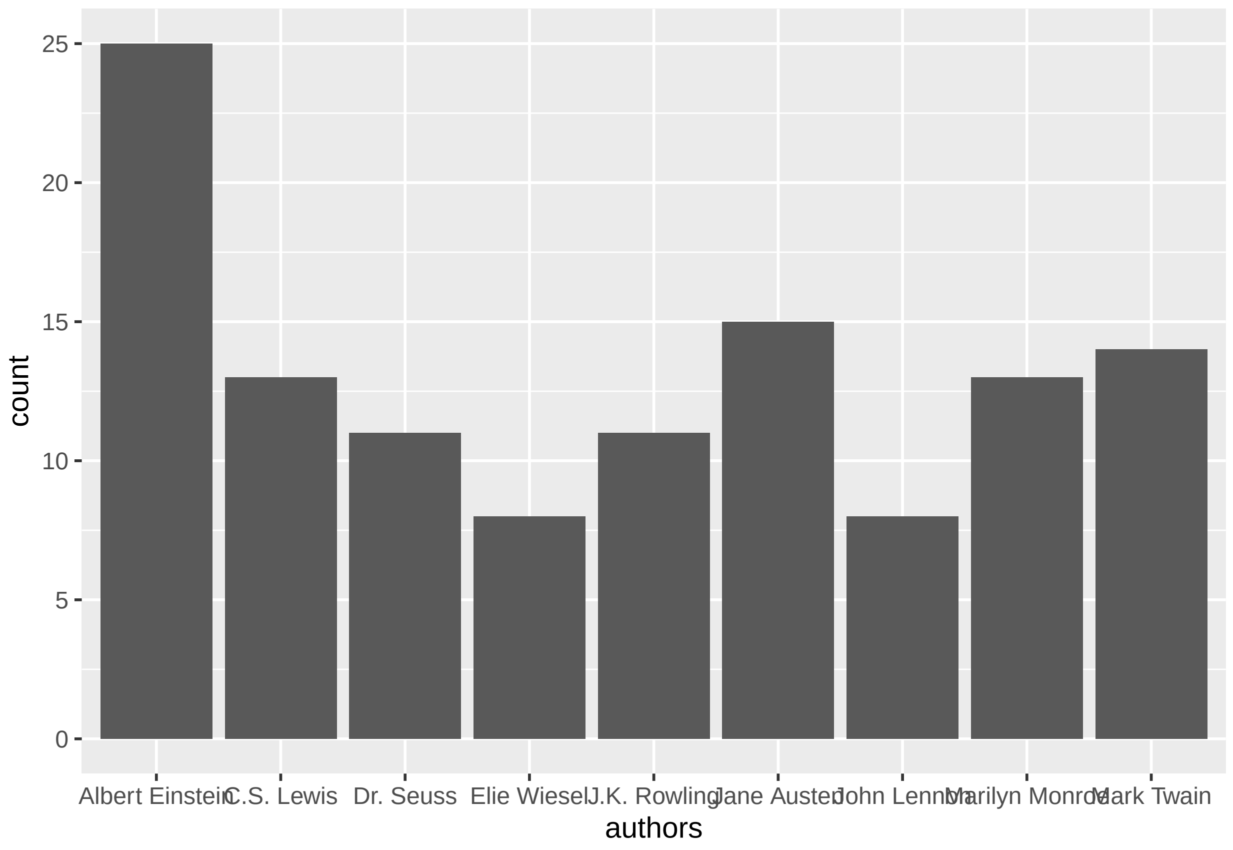
1. the frequency of tags that are mentioned more than twice; *Tips:* grouping by variable using `dplyr` library may be useful here!

```r
filter_tag <- df_quotes %>% group_by(tag) %>% filter(n() > 2) %>% select(tag)

ggplot(filter_tag, aes(x=tag)) +
  geom_bar()
```



2. the frequency of authors that has more than seven quotes; *Tips:* grouping by variable using `dplyr` library may be useful here!

```r
filter_author <- df_quotes %>% group_by(authors) %>% filter(n() > 7) %>% select(authors)

ggplot(filter_author, aes(x=authors)) +
  geom_bar()
```


count
Albert Einstein
C.S. Lewis
Dr. Seuss
Elie Wiesel
J.K. Rowling
Jane Austen
John Lennon
Marilyn Monroe
Mark Twain
authors