Homework 6

Ocean Fan

```r
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.2
v ggplot2   4.0.0     v tibble    3.3.0
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.1.0
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

1.

a.

```r
launches <- readr::read_csv("https://raw.githubusercontent.com/rfordatascience/tidytuesday/re
```

```
Rows: 5726 Columns: 11
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (8): tag, type, variant, mission, agency, state_code, category, agency_...
dbl  (2): JD, launch_year
date (1): launch_date

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
launches <- launches |>
  mutate(agency_type = case_when(
    agency_type == "startup" ~ "private",
    agency_type == "state" ~ "state",
    agency_type == "private" ~ "private"
    )
  )
```

b.

```r
launches <- launches |>
  mutate(state_code = case_when(
    state_code %in% c("RU", "SU") ~ "Russia",
    TRUE ~ state_code
    )
  ) |>
  mutate(state_code = case_when(
    state_code %in% c("F", "I", "I-ELDO", "I-ESA") ~ "Europe",
    TRUE ~ state_code
    )
  )
```

c.

```r
launches |>
  count(state_code, sort = TRUE)
```

```
# A tibble: 13 x 2
   state_code     n
   <chr>      <int>
 1 Russia      3178
 2 US          1716
 3 Europe       316
 4 CN           302
 5 J            115
 6 IN            65
 7 IL            10
 8 IR             8
 9 KP             5
10 CYM            4
11 KR             3
12 BR             2
```

```
13 UK              2
```

```
#the top 6 is Russia, US, Europe, CN, J, IN
launches <- launches |>
  mutate(state_code = case_when(
    state_code %in% c("Russia", "US", "Europe", "CN", "J", "IN") ~ state_code,
    FALSE ~ "other"
    )
  )
```

d.

```
launches <- launches |>
  mutate(state_code = case_when(
    state_code == "CN" ~ "China",
    state_code == "US" ~ "USA",
    state_code == "J" ~ "Japan",
    state_code == "IN" ~ "India",
    TRUE ~ state_code
    )
  )
```

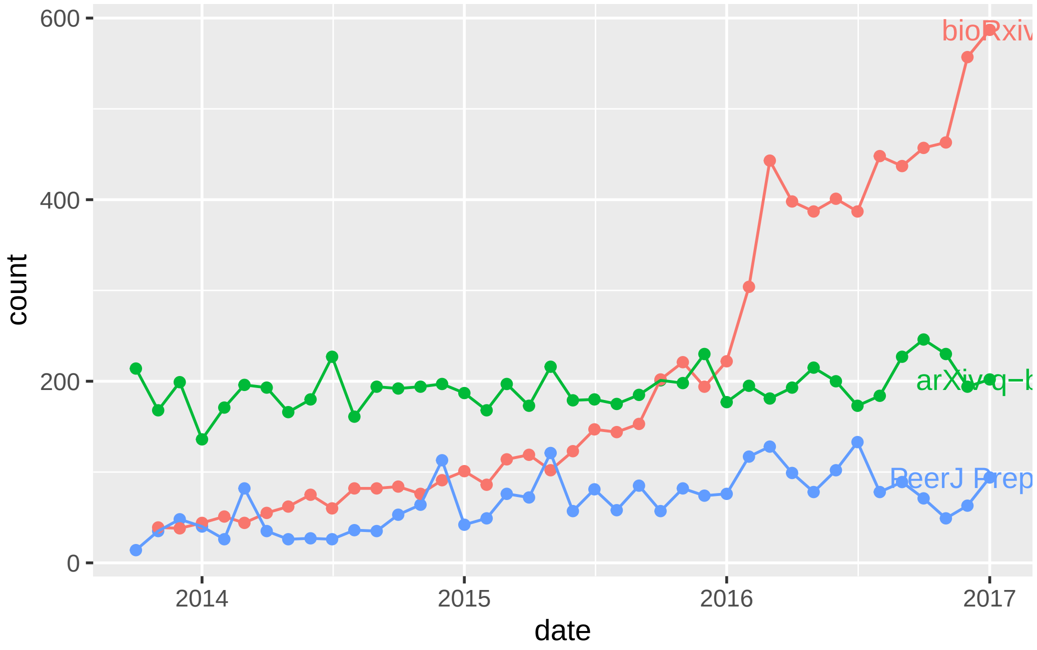
2.

```
preprints <- readr::read_csv("https://stat220-w23.github.io/materials/data/preprints.csv") %>
  filter(between(date, as.Date("2013-10-01"), as.Date("2017-01-01")))
```

```
Rows: 248 Columns: 3
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (1): archive
dbl  (1): count
date (1): date

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
preprints$archive <- factor(preprints$archive, levels = c("bioRxiv", "arXiv q-bio", "PeerJ P
archive_label <- preprints |>
  group_by(archive) |>
  filter(date == "2017-01-01")
preprints |>
  ggplot(aes(x = date, y = count, color = archive)) +
  geom_point() +
  geom_line() +
  geom_text(data = archive_label, aes(label = archive)) +
  theme(legend.position = "none")
```



3.

```
approval <- read_csv("https://stat220-w23.github.io/materials/data/approval.csv")
```

```
Rows: 1020 Columns: 11
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (3): country, president, president_gender
dbl (8): year, quarter, net_approval, gdp, corruption, population, unemploym...
```

```
i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

a.

```
approval <- approval[, c("country", "president", "net_approval", "gdp", "corruption", "popula
```

b.

```
approval <- approval |>
  group_by(country)
approval <- approval[order(approval$corruption),]
head(approval, 8)
```

```
# A tibble: 8 x 6
# Groups:   country [1]
  country   president      net_approval   gdp   corruption population
  <chr>     <chr>                 <dbl> <dbl>        <dbl>      <dbl>
1 Nicaragua Arnoldo Alemán        7.60   85.7 17373017702.    5026796
2 Nicaragua Arnoldo Alemán        7.57   85.7 17373017702.    5026796
3 Nicaragua Arnoldo Alemán        3.87   85.7 17373017702.    5026796
4 Nicaragua Arnoldo Alemán        1.04   85.7 17373017702.    5026796
5 Nicaragua Arnoldo Alemán        0.305  74.4 17887405572.    5100750
6 Nicaragua Arnoldo Alemán       -7.91   74.4 17887405572.    5100750
7 Nicaragua Arnoldo Alemán       -5.13   74.4 17887405572.    5100750
8 Nicaragua Arnoldo Alemán        3.00   74.4 17887405572.    5100750
```

c.

```
smallest_corruption <- approval[1, ]
```

d.

```
approval <- approval[order(approval$net_approval, decreasing = TRUE), ]
largest_net_approval <- approval[1, ]
```

###4.

```
approval$GDP_per_capita <- approval$gdp/approval$population
approval$pop_mil <- approval$population/1000000
```

###5.

```r
load("data/FrontRange.rda")
load("data/FrontRange2.rda")
```

a.

```r
precip <- FR[[1]]
```

FR is stored in list of lists, all values are seperated into certain data types such as double or int; there is also more information in the FR dataset compare to FR2 (which only have station and column). FR2 is stored in a table.

b.

```r
FR2 <- FR2 |>
  filter(station == "st051401")
length(FR2$station)
```

```
[1] 9399
```

```r
info <- FR[["info"]]
nOBs <- info[[4]]
count <- nOBs[[7]]
```

we can see that both dataset returns the value 9399. FR2 is easy to process because we can just distinguish between columns; for FR we have to parse through layers of list and identify the index of the station.

c.

```r
load("data/FrontRange2.rda")
FR2 <- FR2 |>
  filter(station == "st051401")|>
  filter(date <= 1949.999 & date >= 1949.000)|>
  summarize(a = sum(rain))
```

I used FR2 here because we can select the data that we wanted just by doing filter twice.

d.

```
info <- FR[["info"]]
stations1 <- info
station <- names(FR[[1]])
station_factor <- factor(station)
stations1$station <- station_factor
```