Homework (Answers) - Week 5

Vikram Viswanath

2025-09-26

Preface

The goal of this assignment is to introduce you to string manipulation and regular expressions. These tools enable you to identify patterns and extract information from text data. As in class, we'll stick to the functions provided by the **stringr** package (part of the **tidyverse**). Regular expressions, or `regex`, are supported by many functions in R, including those in **stringr**. You may find it helpful to refer to the R stringr cheat sheet and the R for Data Science chapter on strings.

We will work with multiple data sets:

1. Headlines and sentiment analysis of news articles mentioning Apple Inc. from 2018 to 2024 (`aapl_news.csv`).
    a. Each row corresponds to a news article
    b. The data frame contains the variable names: id, ticker, headline, publish_year, publish_month, publish_quarter, publish_datetime, source, url, archived_url, compound_score, positive_score, negative_score, neutral_score, sentiment, encoded_source
2. Release dates of all iPhone models up until 2024 (`iphone_release_dates.csv`).
    a. Each row corresponds to an iPhone model (e.g., "iPhone 16")
    b. The data frame contains the variable names: iphone_model, release_date

1. Import the data in `aapl_news.csv` and assign it to a concise name. Each article is assigned to one of three sentiment categories: positive, neutral, or negative. To warm up, create a data frame that summarizes the number of articles (i.e., rows) that are classified into each sentiment category, then use that to compute the share of articles in each sentiment category. Print the resulting data frame below. Which of the three categories is the most common type of news coverage for Apple, according to these data?

```
aapl_news <- read_csv("aapl_news.csv")

aapl_news |>
  count(sentiment, sort = TRUE) |>
  mutate(share = n / sum(n) * 100)
```

```
# A tibble: 3 x 3
  sentiment     n share
  <chr>     <int> <dbl>
1 neutral    8458  42.5
2 positive   7015  35.3
3 negative   4421  22.2
```

The most common type of news coverage for Apple is **positive**.

2. While every article mentions Apple at some point, not all of them explicitly mention it in the headline. How many articles feature the word "Apple" in the headline, with that specific capitalization?

```
aapl_news |>
  filter(str_detect(headline, "Apple")) |>
  count()
```

```
# A tibble: 1 x 1
      n
  <int>
1  9352
```

3. How many articles feature the word "Apple" in the headline, with *any* capitalization?

```
aapl_news |>
  filter(str_detect(headline, regex("Apple", ignore_case = TRUE))) |>
  count()
```

```
# A tibble: 1 x 1
      n
  <int>
1  9353
```

4. Some headlines mention tech companies other than Apple. Create a new column `other_tech` that counts the number of occurrences of the words "Google", "Alphabet", "Amazon", "Microsoft", "Meta", and "Nvidia" in each headline, with any capitalization. For each value of `other_tech`, compute the share of articles that have a positive sentiment, and present the results in a data frame. Does the share of positive articles increase or decrease as the number of competitor firms mentioned in the headline increase?

```
aapl_news |>
  mutate(other_tech = str_count(
    headline,
    regex("Google|Alphabet|Amazon|Microsoft|Meta|Nvidia", ignore_case = TRUE)
  )) |>
  group_by(other_tech) |>
  summarize(share_positive = mean(sentiment == "positive") * 100)
```

```
# A tibble: 5 x 2
  other_tech share_positive
       <int>          <dbl>
1          0           34.1
2          1           41.0
3          2           46.0
4          3           76.4
5          4           97.0
```

The share of positive articles **decreases** as the number of competitor firms mentioned in the headline increases.

5. Import the data in `iphone_release_dates.csv` and assign it to a concise name. Use `str_extract()` to find components of the `release_date` variable that match certain patterns. First, create a new variable `release_year`, which contains the year of release as an integer. Then, create a new variable `release_month`, which contains the month of release as a string. Once you have both variables, use `count()` to count the number of models (i.e., rows) released in each `release_year-release_month` combination. Assign the resulting data frame to a name (recycling the original name is okay). Use the data frame to determine which month is the most common release month for new iPhone models.

```
iphone_releases <- read_csv("iphone_release_dates.csv")

# extract date components using regular expressions
# there are many acceptable ways to do this!
iphone_releases <- iphone_releases |>
  mutate(
    release_month = str_extract(release_date, "[A-Za-z]+"),
    release_year = as.integer(str_extract(release_date, "[0-9]{4}$")),
  ) |>
  # count the number of releases in each year-month combination
  count(release_year, release_month)

# find the most common month (regardless of year)
most_common_month <- iphone_releases |>
  group_by(release_month) |>
  summarize(n = sum(n)) |>
  slice_max(n) |>
  pull(release_month)
```

New iPhone models are most often released in September.

6. Join the data frame you created above with `aapl_news` by year and month, making sure you preserve all news headlines. Filter the joined data so that you have only headlines that mention the word "iPhone", keeping in mind that capitalization might vary (e.g., "iphone", "IPhone", etc.). Create a new column `was_iphone_released` that takes value `TRUE` if any number of iPhones was released in the same year-month as the article, and `FALSE` otherwise. Present a table showing the share of articles in each sentiment category, with sentiments in the columns and whether the articles came out in months with and without an iPhone release in the rows. How does sentiment compare between articles published in months when a new iPhone model was released versus not?

```
# Step 1: join and filter the data
joined_data <- aapl_news |>
  filter(str_detect(headline, regex("iPhone", ignore_case = TRUE))) |>
  left_join(iphone_releases, by = c("publish_year" = "release_year",
  "publish_month" = "release_month"))

# Step 2: create logical variable was_iphone_released
joined_data <- joined_data |>
  mutate(was_iphone_released = !is.na(n))

# Step 3: count headlines per was_iphone_released and sentiment category
sentiment_counts <- joined_data |>
  count(was_iphone_released, sentiment)

# Step 4: compute share of articles with each sentiment by was_iphone_released
sentiment_shares <- joined_data |>
  group_by(was_iphone_released) |>
  count(sentiment) |>
  mutate(share = n / sum(n) * 100) |>
  select(-n)

# Step 5: pivot the data to get the desired format
sentiment_table <- sentiment_shares |>
  pivot_wider(
    names_from = sentiment,
    values_from = share,
    names_prefix = "share_"
  ) |>
  mutate(
    was_iphone_released = ifelse(was_iphone_released,
```

```
    "iPhone Released", "No iPhone Release")
  )

sentiment_table
```

```
# A tibble: 2 x 4
# Groups:   was_iphone_released [2]
  was_iphone_released share_negative share_neutral share_positive
  <chr>                        <dbl>         <dbl>          <dbl>
1 No iPhone Release             30.7          41.7           27.6
2 iPhone Released               19.1          49.6           31.3
```

The sentiment of articles that mention the iPhone is **more positive** in months when a new iPhone is released, compared to other months.