PS941 Practical Report

5637058

Challenge 1 - Data Scraping

Your task is to develop a short step-by-step tutorial demonstrating how to scrape data from the 'https://books.toscrape.com/' website.

In your tutorial, you should first describe potential ethical issues associated with scraping online resources. Make sure that you explain how someone could determine whether scraping data from the `books.toscrape` website is allowed (and possible).

In your tutorial you need to explain how to make custom GET requests to `books.toscrape` using the `httr2` package. Carefully annotate your code, explaining how to customize your request (e.g., by changing its header). Second, you will need to build a custom function that uses `rvest` and CSS selectors to extract data about each book's title, price, and star rating. Finally, you should create a crawler function for a given category (e.g., science fiction, romance) of books. In other words, you should write a wrapper function that sends a request and parses data received from the server. Ultimately, your objective should be to define a crawler function in a format similar to the template below:

```
scrape_book_data <- function(category){

    # Part 1
    # Make a request using httr2

    # Part 2
    # Extract data about each book's title, price, and star rating

    # Part 3
    # Perform the above steps for every page of results available for a given category of
    ↪  books

    # Return a data.frame object with the results
}
```

In order to obtain full mark for this challenge, your crawler must be functional. That is, I should be able to run your function to obtain a `data.frame` containing all results for a single category of books.

You can also achieve extra marks for explaining how to make your scraper more robust. For example, you may consider how your scraper should behave if it encounters a specific HTTP error, or how to avoid overloading the server with too many requests, or how your scraper function could prevent a person from scraping the same data twice.

Challenge 1 Answer

Potential Ethical Issues

In your tutorial, you should first describe potential ethical issues associated with scraping online resources. Make sure that you explain how someone could determine whether scraping data from the `books.toscrape` *website is allowed (and possible).*

All web scrapping endeavors must be ethical. Being ethical means adhering by some rules to keep the web scrapping process transparent and avoid potentially private data being scrapped. Some rules include utilising APIs when available, requesting data at a reasonable rate (i.e., not to be confused for a DDoS attack), only saving data that is absolutely required and ensuring data is being used to create new value not to duplicate it along with others.

To check if scraping is allowed or even possible on books.toscrape.com, look for its `robots.txt` file. This can be done by adding `robots.txt` to the homepage website link (e.g., https://books.toscrape.com/robots.txt). Looking at this website there does not appear to be an available `robots.txt` file; this file specifies certain rules concerning the behaviour of crawlers, bots, and scrapers. However, just as the presence of a `robots.txt` file does not prevent you from scrapping the absence of this file does not necessarily mean scrapping of this site is allowed. It is up to the scrapper (i.e., the user) to do the proper due diligence before beginning the scrapping process.

Tutorial

In your tutorial you need to explain how to make custom GET requests to `books.toscrape` *using the* `httr2` *package. Carefully annotate your code, explaining how to customize your request (e.g., by changing its header). Second, you will need to build a custom function that uses* `rvest` *and CSS selectors to extract data about each book's title, price, and star rating. Finally, you should create a crawler function for a given category (e.g., science fiction, romance) of books. In other words, you should write a wrapper function that sends a request and parses data received from the server. Ultimately, your objective should be to define a crawler function in a format similar to the template below:*

```
scrape_book_data <- function(category) {

  ### Step 1: Load the necessary libraries ###
  library(rvest)
  library(httr2)
  library(tidyverse)


  ### Step 2: Make an initial request from the homepage ###
  category <- category %>% tolower()   # standardise input to be lowercase

  base_website <- 'https://books.toscrape.com/'   # home page website

  # GET request (with custom headers)
  og_req <- request(base_website) %>%
    req_headers(
      Name = "5637058",
      Accept = "text/html"
    )

  # POST response
  og_resp <- og_req %>%
    req_perform()

  Sys.sleep(runif(1, 1, 3))   # random delay between 1 and 3 seconds to avoid overloading
  ↪  the server with too many requests

  # Extract data (i.e., categories and associated links)
  categories <- og_resp %>%
    resp_body_html() %>%
    html_elements(".nav-list ul a") %>%
    html_text2() %>%
    tolower()

  n <- length(categories)   # number of categories

  links <- og_resp %>%
    resp_body_html() %>%
    html_elements(".nav-list ul a") %>%
    html_attr("href")


  ### Step 3: Extracting data (i.e., prices, titles, ratings) from specific categories
  for (i in seq(1, n)) {
```

```r
  cat <- categories[i]

  ### Step 3a: Identify category from input ###
  if (cat == category) {

    # Extract full category website
    cat_website <- links[i]   # partial category website
    website <- paste0(base_website, cat_website)   # full category website

    # GET request
    req <- request(website) %>%
      req_headers(
        Name = "5637058",
        Accept = "text/html"
      )

    # POST response
    resp <- req %>%
      req_perform()

    Sys.sleep(runif(1, 1, 3))

    # Identify the number of pages for the category
    pages <- resp %>%
      resp_body_html() %>%
      html_elements(".current") %>%
      html_text2()

    page <- str_extract(pages, "\\d+$") %>% as.numeric()   # extract number of pages


    ### Step 3b: If there is more than one page, loop through each page and extract
    ↪  information ###
    if (length(page) > 0 ) {

      # Loop through each page
      for (j in seq(1, page)) {

        # Create website link for each page
        cat_website <- gsub("/index.html", "", cat_website)
        website3 <- paste0(base_website, cat_website, "/page-", j, ".html")

        # GET request
        req3 <- request(website3) %>%
          req_headers(
            Name = "5637058",
            Accept = "text/html"
          )

        # POST response
        resp3 <- req3 %>%
          req_perform()
```

```
    Sys.sleep(runif(1, 1, 3))

    # Extract information
    prices <- resp3 %>%
      resp_body_html() %>%
      html_elements(".price_color") %>%
      html_text2()

    titles <- resp3 %>%
      resp_body_html() %>%
      html_elements("h3 a") %>%
      html_attr("title")

    ratings <- resp3 %>%
      resp_body_html() %>%
      html_elements(".star-rating") %>%
      html_attr("class")

    # Rename ratings to follow format '#number of stars# stars'
    ratings <- ratings %>%
      gsub("star-rating ", "", .) %>%
      paste(., "stars")

    # Put all information in a table
    table <- data.frame(
      Title = titles,
      Price = prices,
      Rating = ratings
    )

    print(table)   # print table

  }

}


### Step 3b: If there is only one page, loop through that page and extract
↪  information ###
else {

  prices <- resp %>%
      resp_body_html() %>%
      html_elements(".price_color") %>%
      html_text2()

    titles <- resp %>%
      resp_body_html() %>%
      html_elements("h3 a") %>%
      html_attr("title")

    ratings <- resp %>%
      resp_body_html() %>%
```

```
            html_elements(".star-rating") %>%
            html_attr("class")

          ratings <- ratings %>% gsub("star-rating ", "", .) %>% paste(., "stars")

          table <- data.frame(
            Title = titles,
            Price = prices,
            Rating = ratings
          )

          print(table)

      }

    }

  }

}
```

```
scrape_book_data("science fiction")
```

```
## -- 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   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter()         masks stats::filter()
## x readr::guess_encoding() masks rvest::guess_encoding()
## x dplyr::lag()            masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
##                                                                                                  Title
## 1                                       Mesaerion: The Best Science Fiction Stories 1800-1849
## 2                                                                                                 Join
## 3  William Shakespeare's Star Wars: Verily, A New Hope (William Shakespeare's Star Wars #4)
## 4                                                                                          The Project
## 5                                                                                       Soft Apocalypse
## 6                                                                  Sleeping Giants (Themis Files #1)
## 7                                                                                              Arena
## 8                                                 Foundation (Foundation (Publication Order) #1)
## 9           The Restaurant at the End of the Universe (Hitchhiker's Guide to the Galaxy #2)
## 10                                                                                  Ready Player One
## 11                  Life, the Universe and Everything (Hitchhiker's Guide to the Galaxy #3)
## 12                                                                                    Dune (Dune #1)
## 13                                    Do Androids Dream of Electric Sheep? (Blade Runner #1)
## 14                                               Three Wishes (River of Time: California #1)
## 15                                                   The Last Girl (The Dominion Trilogy #1)
## 16                                 Having the Barbarian's Baby (Ice Planet Barbarians #7.5)
##    Price     Rating
## 1 £37.59  One stars
```

```
## 2  £35.67 Five stars
## 3  £43.30 Four stars
## 4  £10.65  One stars
## 5  £26.12  Two stars
## 6  £48.74  One stars
## 7  £21.36 Four stars
## 8  £32.42  One stars
## 9  £10.92  One stars
## 10 £19.07 Four stars
## 11 £33.26  Two stars
## 12 £54.86  One stars
## 13 £51.48  One stars
## 14 £44.18  Two stars
## 15 £36.26  Two stars
## 16 £34.96 Four stars
```

```
scrape_book_data("romance")
```

```
##                                           Title  Price      Rating
## 1                       Chase Me (Paris Nights #2) £25.27  Five stars
## 2                                    Black Dust £34.53  Five stars
## 3    Her Backup Boyfriend (The Sorensen Family #1) £33.97   One stars
## 4               First and First (Five Boroughs #3) £15.97  Four stars
## 5            Fifty Shades Darker (Fifty Shades #2) £21.96   One stars
## 6                           The Wedding Dress £24.12   One stars
## 7                Suddenly in Love (Lake Haven #1) £55.99   Two stars
## 8                        Something More Than This £16.24  Four stars
## 9              Doing It Over (Most Likely To #1) £35.61 Three stars
## 10            The Wedding Pact (The O'Malleys #2) £32.61 Three stars
## 11         Hold Your Breath (Search and Rescue #1) £28.82   One stars
## 12                          Dirty (Dive Bar #1) £40.83  Four stars
## 13     Take Me Home Tonight (Rock Star Romance #3) £53.98 Three stars
## 14           Off the Hook (Fishing for Trouble #1) £47.67 Three stars
## 15 A Gentleman's Position (Society of Gentlemen #3) £14.75  Five stars
## 16                               Sit, Stay, Love £20.90 Three stars
## 17  A Girl's Guide to Moving On (New Beginnings #2) £31.30   One stars
## 18              The Perfect Play (Play by Play #1) £59.99 Three stars
## 19        Dark Lover (Black Dagger Brotherhood #1) £12.87   One stars
## 20             Changing the Game (Play by Play #2) £13.38 Three stars
##                                           Title  Price      Rating
## 1                         A Walk to Remember £56.43   One stars
## 2   The Purest Hook (Second Circle Tattoos #3) £12.25   One stars
## 3                            The Obsession £45.43   One stars
## 4                      Reservations for Two £11.10 Three stars
## 5           Best of My Love (Fool's Gold #20) £27.41   Two stars
## 6  Where Lightning Strikes (Bleeding Stars #3) £39.77 Three stars
## 7          This One Moment (Pushing Limits #1) £48.71   One stars
## 8                   Rhythm, Chord & Malykhin £28.34   Two stars
## 9        My Perfect Mistake (Over the Top #1) £38.92   Two stars
## 10                  Listen to Me (Fusion #1) £58.99 Three stars
## 11                       Imperfect Harmony £34.74  Four stars
## 12               Fighting Fate (Fighting #6) £39.24 Three stars
## 13            Deep Under (Walker Security #1) £47.09  Five stars
## 14   Charity's Cross (Charles Towne Belles #4) £41.24   One stars
```

```
## 15              Bounty (Colorado Mountain #7) £37.26  Four stars
```

Challenge 2 - Fitting the Right Model

For your second challenge, you will need to demonstrate how underfitting and overfitting can impact the Mean Squared Error (MSE) of different regression models. For this challenge, you will work with a synthetic dataset generated by the R code provided below (but feel free to modify the data generation process if you wish). The 3D plot of these data are shown below.

Your response should include description and R code necessary to achieve the following:

Data Generation: Use the provided code to generate your dataset. You are welcome to adjust the code to create different data if you would like to experiment.

Model Fitting and Cross-Validation: Perform k-fold cross-validation (e.g., 5-fold or 10-fold) on your *entire* dataset.

Model Design: Within the cross-validation procedure, fit different versions of the linear regression models (varying their complexity). Design these models such that you can achieve the following outcomes:

- *Underfitting*: A model that is too simple to capture the underlying patterns in the data.
- *Good Fit*: A model that appropriately captures the underlying patterns without overfitting.
- *Overfitting*: A model that is too complex and fits the noise in the data, rather than the true underlying relationship.

Cross-validation Evaluation: For each model type (underfitting, good fit, overfitting), calculate the average MSE across all k folds of your cross-validation.

Reporting: Report the average MSE for each model type. Explain how the model complexity relates to the MSE. Discuss how underfitting and overfitting manifest themselves in the MSE values.

You can obtain extra marks if you can include a visualization of overfitting and underfitting.

Challenge 2 Answers

```
library(tidyverse)
library(caret)
library(glmnet)
```

Data Generation

Data Generation: *Use the provided code to generate your dataset. You are welcome to adjust the code to create different data if you would like to experiment.*

```
set.seed(123)

# Number of data points
n <- 1000

# Generate x values
x <- seq(0, 2*pi, length.out = n)

# Generate the true y values
y_true <- 3*sin(x) + 7

# Generate y values (y_true with random noise)
y <- y_true + rnorm(n, sd = 3)

# Combine everything into a single data frame
data <- data.frame(x = x, y = y, y_true = y_true)

# Rearrange columns
data <- data %>%
  mutate(id = seq_along(1:n)) %>%
  select(id, everything())

data %>% head()
```

```
##   id           x         y   y_true
## 1  1 0.000000000  5.318573 7.000000
## 2  2 0.006289475  6.328336 7.018868
## 3  3 0.012578950 11.713861 7.037736
## 4  4 0.018868424  7.268127 7.056602
## 5  5 0.025157899  7.463329 7.075466
## 6  6 0.031447374 12.239522 7.094327
```

Model Fitting and Cross-Validation

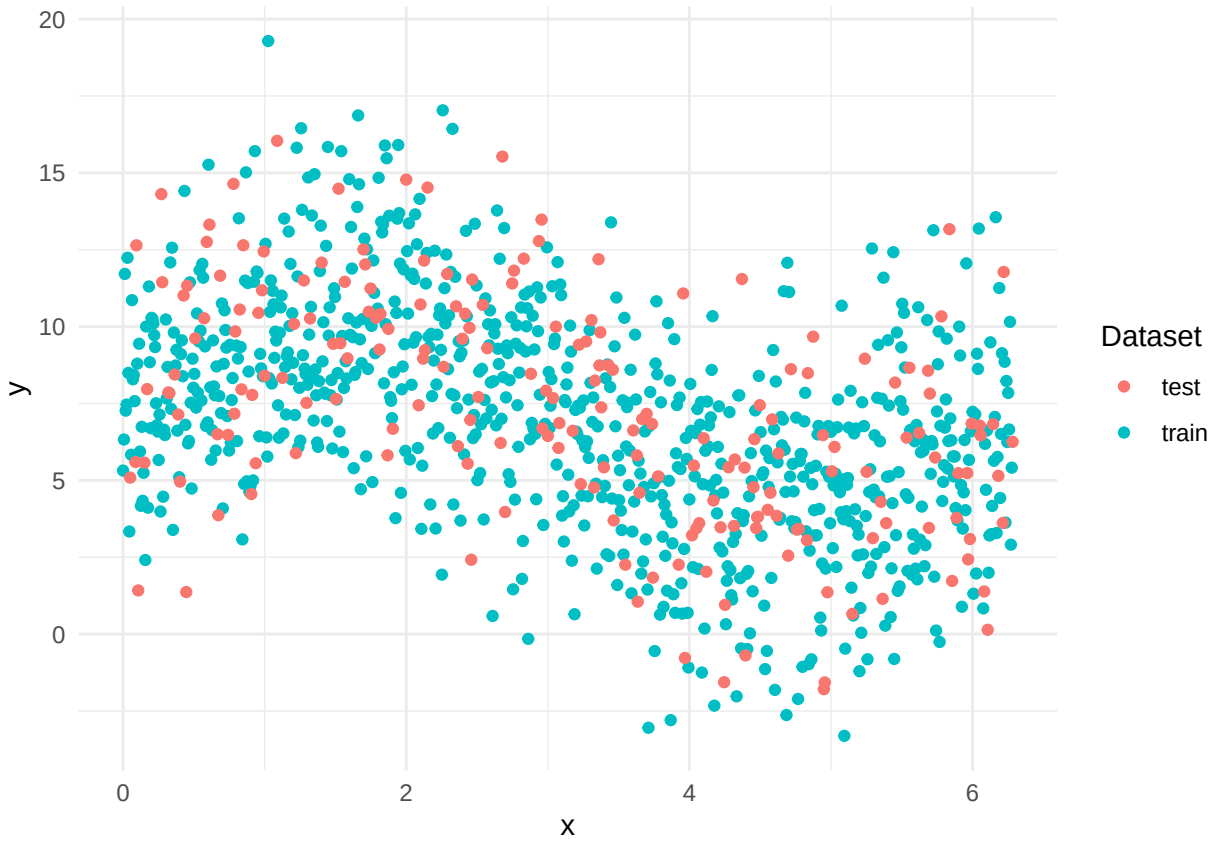
Model Fitting and Cross-Validation: *Perform k-fold cross-validation (e.g., 5-fold or 10-fold) on your entire dataset.*

```
# Split into four training and one test dataset (i.e., five-fold)
train1 <- sample(1:1000, 200, replace = FALSE)   # sample from 1 to 1000
remaining_nums <- setdiff(1:1000, train1)   # remove sampled numbers from original
↪  choices

train2 <- sample(remaining_nums, 200, replace = FALSE)   # sample from remaining numbers
remaining_nums <- setdiff(1:1000, c(train1, train2))   # remove sampled numbers from the
↪  remaining numbers
```

```r
train3 <- sample(remaining_nums, 200, replace = FALSE)
remaining_nums <- setdiff(1:1000, c(train1, train2, train3))

train4 <- sample(remaining_nums, 200, replace = FALSE)
remaining_nums <- setdiff(1:1000, c(train1, train2, train3, train4))

test <- sample(remaining_nums, 200, replace = FALSE)

train1_df <- data[train1, ]
train2_df <- data[train2, ]
train3_df <- data[train3, ]
train4_df <- data[train4, ]

train_index <- c(train1, train2, train3, train4)
train_df <- data[train_index, ]
test_df <- data[test, ]

train_df <- train_df %>%
  mutate(dataset = 'train')

test_df <- test_df %>%
  mutate(dataset = 'test')

train_test_df <- rbind(train_df, test_df)

train_test_df %>% head()
```

```
##      id        x         y   y_true dataset
## 225 225 1.408842  8.869770 9.960742   train
## 255 255 1.597527 14.794455 9.998928   train
## 561 561 3.522106  3.384278 5.885809   train
## 946 946 5.943554  6.311495 6.000581   train
## 554 554 3.478080  6.758955 6.009481   train
## 457 457 2.868001 11.141406 7.810575   train
```

```r
ggplot(train_test_df, aes(x = x, y = y, col = dataset)) +
  geom_point() +
  labs(col = "Dataset") +
  theme_minimal()
```



Model Design

***Model Design**: Within the cross-validation procedure, fit different versions of the linear regression models (varying their complexity). Design these models such that you can achieve the following outcomes:*

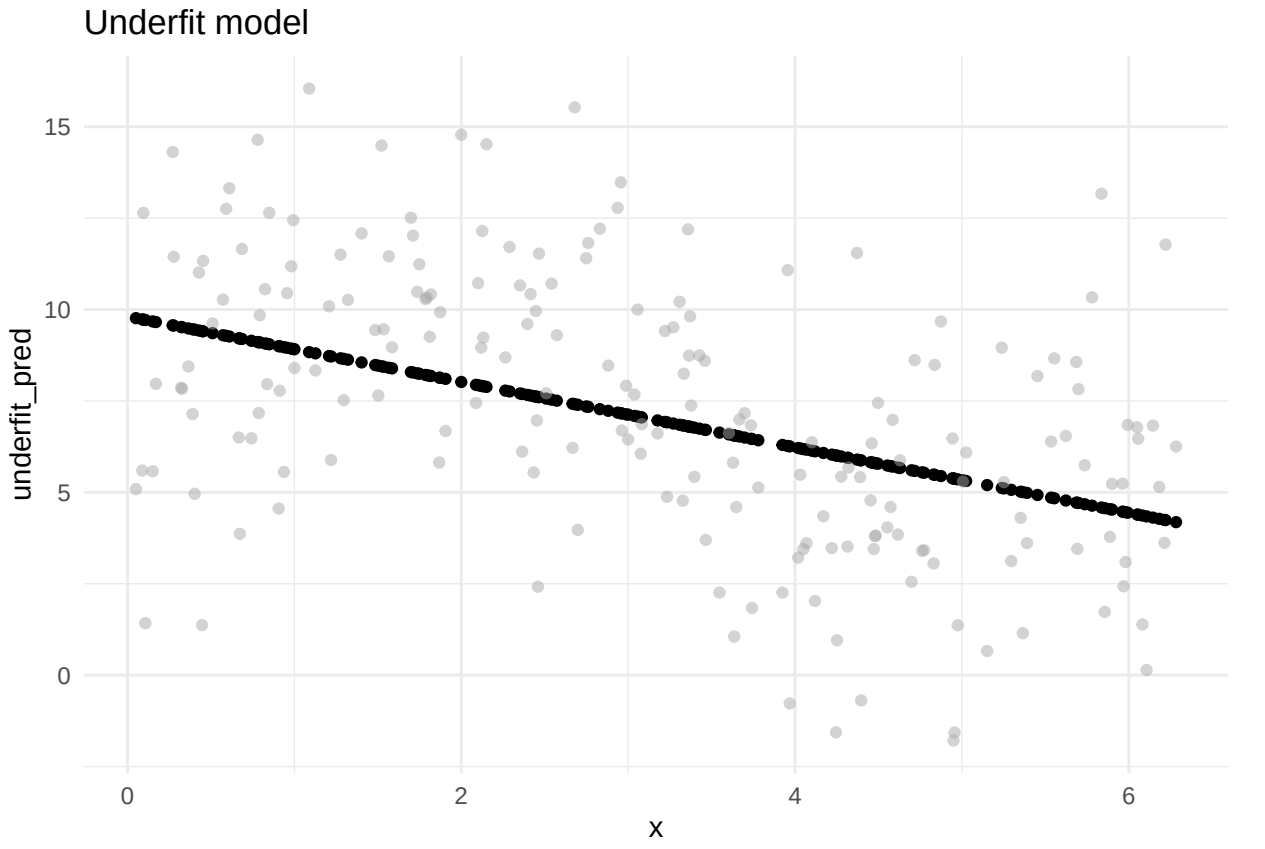
- *Underfitting: A model that is too simple to capture the underlying patterns in the data.*
- *Good Fit: A model that appropriately captures the underlying patterns without overfitting.*
- *Overfitting: A model that is too complex and fits the noise in the data, rather than the true underlying relationship.*

```
# Fit model to training data
underfit <- lm(y ~ poly(x, 1), data = train_df)
goodfit <- lm(y ~ poly(x, 3), data = train_df)
overfit <- lm(y ~ poly(x, 20), data = train_df)

# Predict using test data
test_df$underfit_pred <- predict(underfit, newdata = test_df)
test_df$goodfit_pred <- predict(goodfit, newdata = test_df)
test_df$overfit_pred <- predict(overfit, newdata = test_df)

test_df %>% head()
```

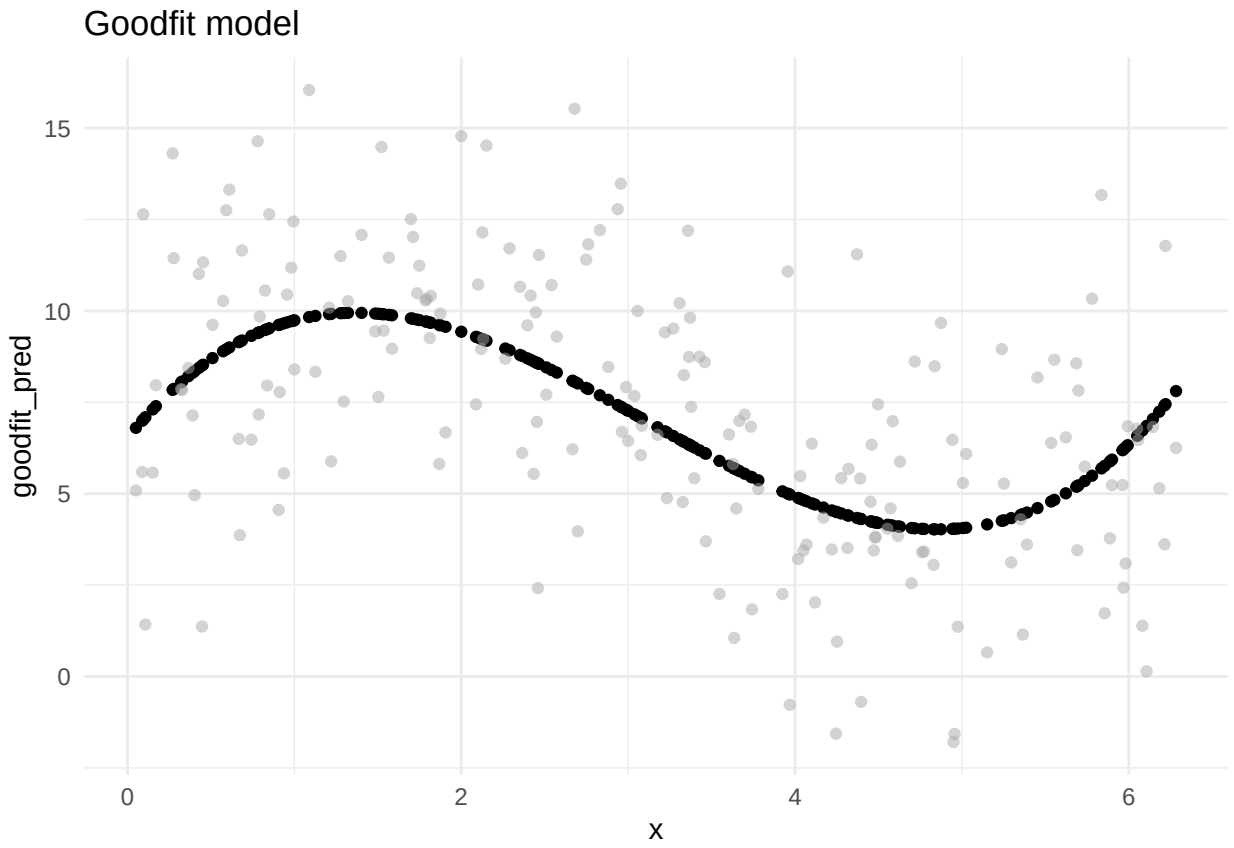
```
##      id         x         y   y_true dataset underfit_pred goodfit_pred
## 602 602 3.7799743  5.130269 5.212310    test      6.425140     5.366123
## 776 776 4.8743430  9.672850 4.039258    test      5.445394     4.026904
## 207 207 1.2956318  7.521333 9.887142    test      8.649276     9.941871
## 476 476 2.9875005  7.915491 7.460449    test      7.134611     7.298321
## 110 110 0.6855528 11.656293 8.899303    test      9.195456     9.190496
```

```
## 513 513 3.2202111  9.411783 6.764388    test      6.926274     6.707063
##     overfit_pred
## 602     4.782068
## 776     4.248781
## 207     9.712963
## 476     7.565222
## 110     8.174360
## 513     6.702291
```

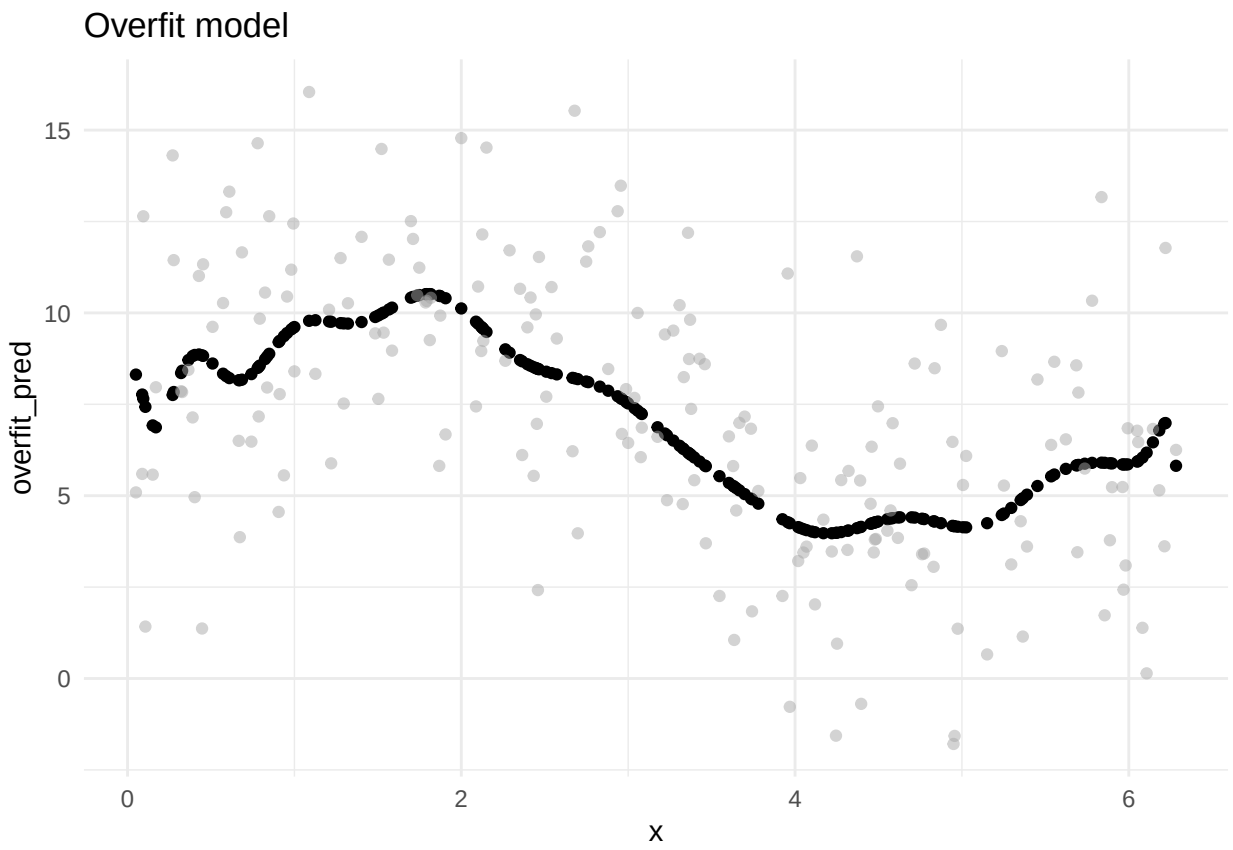
```
ggplot(test_df) +
  geom_point(aes(x = x, y = underfit_pred)) +
  geom_point(aes(x = x, y = y), color = 'darkgrey', alpha = 0.5) +
  labs(title = "Underfit model") +
  theme_minimal()
```




```
ggplot(test_df) +
  geom_point(aes(x = x, y = goodfit_pred)) +
  geom_point(aes(x = x, y = y), color = 'darkgrey', alpha = 0.5) +
  labs(title = "Goodfit model") +
  theme_minimal()
```




```
ggplot(test_df) +
  geom_point(aes(x = x, y = overfit_pred)) +
  geom_point(aes(x = x, y = y), color = 'darkgrey', alpha = 0.5) +
  labs(title = "Overfit model") +
  theme_minimal()
```



Cross-validation Evaluation

***Cross-validation Evaluation**: For each model type (underfitting, good fit, overfitting), calculate the average MSE across all k folds of your cross-validation.*

```
fit_predict <- function(train_set, test_set, degree) {

  # Fit model
  model <- lm(y ~ poly(x, degree), data = train_set)

  # Predict on test set
  predictions <- predict(model, newdata = test_set)

  # Calculate MSE for k-fold cv
  mse <- (sum((test_set$y - predictions)^2))/5

  return(mse)
}
```

```
# Underfit
degree = 1

mse1 <- fit_predict(train1_df, test_df, degree = degree)  # Train on fold 1
mse2 <- fit_predict(train2_df, test_df, degree = degree)  # Train on fold 2
mse3 <- fit_predict(train3_df, test_df, degree = degree)  # Train on fold 3
mse4 <- fit_predict(train4_df, test_df, degree = degree)  # Train on fold 4
```

```r
# MSE
mse_uf <- c(mse1, mse2, mse3, mse4)
mean_mse_uf <- mean(mse_uf)

print("Underfit")
```

```
## [1] "Underfit"
```

```r
for (i in seq(4)) {
  print(paste("MSE for fold", i, ":", mse_uf[i]))
}
```

```
## [1] "MSE for fold 1 : 428.626002475528"
## [1] "MSE for fold 2 : 434.343922187221"
## [1] "MSE for fold 3 : 426.401601866147"
## [1] "MSE for fold 4 : 432.145444958442"
```

```r
print(paste("Average MSE across all folds:", mean_mse_uf))
```

```
## [1] "Average MSE across all folds: 430.379242871835"
```

```r
# Goodfit
degree = 3

mse1 <- fit_predict(train1_df, test_df, degree = degree)
mse2 <- fit_predict(train2_df, test_df, degree = degree)
mse3 <- fit_predict(train3_df, test_df, degree = degree)
mse4 <- fit_predict(train4_df, test_df, degree = degree)

# MSE
mse_gf <- c(mse1, mse2, mse3, mse4)
mean_mse_gf <- mean(mse_gf)

print("Goodfit")
```

```
## [1] "Goodfit"
```

```r
for (i in seq(4)) {
  print(paste("MSE for fold", i, ":", mse_gf[i]))
}
```

```
## [1] "MSE for fold 1 : 366.930124217357"
## [1] "MSE for fold 2 : 375.260012497001"
## [1] "MSE for fold 3 : 359.603046497799"
## [1] "MSE for fold 4 : 382.562335589852"
```

```r
print(paste("Average MSE across all folds:", mean_mse_gf))
```

```
## [1] "Average MSE across all folds: 371.088879700502"
```

```r
# Overfit
degree = 20

mse1 <- fit_predict(train1_df, test_df, degree = degree)
mse2 <- fit_predict(train2_df, test_df, degree = degree)
mse3 <- fit_predict(train3_df, test_df, degree = degree)
```

```r
mse4 <- fit_predict(train4_df, test_df, degree = degree)

# MSE
mse_of <- c(mse1, mse2, mse3, mse4)
mean_mse_of <- mean(mse_of)

print("Overfit")
```

```
## [1] "Overfit"
```

```r
for (i in seq(4)) {
  print(paste("MSE for fold", i, ":", mse_of[i]))
}
```

```
## [1] "MSE for fold 1 : 363.298009058443"
## [1] "MSE for fold 2 : 454022.510913877"
## [1] "MSE for fold 3 : 383.250468050391"
## [1] "MSE for fold 4 : 420.609681918977"
```

```r
print(paste("Average MSE across all folds:", mean_mse_of))
```

```
## [1] "Average MSE across all folds: 113797.417268226"
```

Reporting

***Reporting**: Report the average MSE for each model type. Explain how the model complexity relates to the MSE. Discuss how underfitting and overfitting manifest themselves in the MSE values.*

1. Underfit: 430.38
2. Good fit: 371.09
3. Overfit: 113797.42

The greater the complexity the more likely for the MSE to be higher. This positive relationship between complexity and MSE can be attributed to *overfitting* whereby the model fits to not just the training data but the noise within this data as well. Therefore, this more complex model may have lower MSE with training data but greater MSE when predicting on test data. However, even extremely simple models can lead to greater MSE as they often *underfit* the model whereby they are less likely to capture important relationships within the data.

Challenge 3 - NLP and Classification

In your third challenge, your task is to build a language-based classification model.

In this challenge, you will be analysing a dataset containing social media posts on the topic of global warming. This dataset is provided to you (file called `ClimatePosts.csv` available on Moodle). This dataset includes messages posted by individuals who self-identify as those who believe that human activities are responsible for climate change (`believers`), as well as those who are skeptical about such claims (`sceptics`). User identity is provided in the column labeled `views`.

Your task is to construct classification models that could identify user type based on the content of their posts. There are two types of models that you need to construct and fit to this dataset.

1. **Top-down model** - to define your model, you must identify a psychological construct that you want to use in your prediction. You will need to define a list of tokens and then use this dictionary to obtain a score for each social media post. For example, you could calculate the relative frequency of particular term(s) appearing in each post, or use norms (e.g., like the norms developed by Warriner and colleagues) to score each post on some variable (e.g., dominance, valence). You are completely free to choose any psychological constructs (e.g., sentiment, certainty, emotional tone) as long as you provide some rationale for your choice.
2. **Bottom-up model** - to build this model, you will need to use a TF-IDF based on the entire vocabulary in the dataset. In order to fit this model, you will need to use a regularized regression, such as Ridge or Lasso.

The overall structure of your response to this challenge should be as follows:

1. Load the data into R and pre-process (e.g., remove punctuation, lowercase, etc.) it for the purpose of text analysis.
2. Define features of your top-down model. Clearly explain how your scoring algorithm works.
3. Construct a TF-IDF for the bottom-up model.
4. Fit both the top-down and bottom-up models to your data. Use k-fold cross-validation to tune your regularization parameter.
5. Inspect and evaluate your results. For assessing the models' overall performance, you could report the confusion matrix. For the top-down model, you may want to report (and visualize) the influence or importance of your chosen features. For the bottom-up model, you can report (and visualize) the most predictive tokens from your model (i.e., those with the largest absolute coefficient values).

Extra points will be awarded for creative use of graphs and plots to summarize your results (e.g., ROC curve to visualize model accuracy). You are also encouraged to introduce text pre-processing steps (e.g., removing stop words, lemmatizing text, etc.).

Challenge 3 Answers

```
library(tidyverse)
library(data.table)
library(tm)
library(quanteda)
library(quanteda.textstats)
library(quanteda.corpora)
library(glmnet)
```

```
data_cp <- read_csv("ClimatePosts.csv")

data_cp <- data_cp %>%
```

```
  select(id, post, views)   # rearrange columns

head(data_cp)
```

```
## # A tibble: 6 x 3
##      id post                                                          views
##   <dbl> <chr>                                                         <chr>
## 1     1 Global warming is a real threat. Rising temperatures are alarming~ beli~
## 2     2 The science is clear: rising temperatures are a crisis. We must r~ beli~
## 3     3 I'm worried about the future with these rising temperatures. We n~ beli~
## 4     4 Pollution is fueling global warming. We need stricter regulations~ beli~
## 5     5 Rising temperatures are impacting our ecosystems. We must act now~ beli~
## 6     6 The Paris Act is a start, but we need more action to combat globa~ scep~
```

```
dim(data_cp)
```

```
## [1] 242   3
```

```
# Remove non alpha numeric symbols
data_cp$post <- gsub("[^[:alnum:] ]", "", data_cp$post)

# Remove numbers
data_cp$post <- gsub("[0-9]", "", data_cp$post)

# Remove hyperlinks
data_cp$post <- gsub("http\\S+\\s*", "", data_cp$post)

# Remove stop words?
stopwords <- stopwords("en")   # get English stopwords (e.g., "the", "is", "and")
data_cp$post <- removeWords(data_cp$post, stopwords)

# Remove blank spaces
data_cp$post <- gsub("\\s+", " ", data_cp$post)
data_cp$post <- trimws(data_cp$post)

# Convert text to lowercase
data_cp$post <- tolower(data_cp$post)

# After preprocessing
data_cp$post[1:5]
```

```
## [1] "global warming real threat rising temperatures alarming we need action now climatechange"
## [2] "the science clear rising temperatures crisis we must reduce pollution protect planet globalwarmi
## [3] "im worried future rising temperatures we need take paris act seriously climateaction"
## [4] "pollution fueling global warming we need stricter regulations taxation polluters environment"
## [5] "rising temperatures impacting ecosystems we must act now protect planet climatechange"
```

```
# Convert each word into a token
cp_tokens <- tokens(data_cp$post, remove_punct = TRUE)
cp_tokens<- tokens_tolower(cp_tokens)

# Identifying colloquations
colqs <- tokens_select(
  cp_tokens,
  pattern = "^[a-z]",   # only selects words that start with a lowercase letter
```

```
  valuetype = "regex",    # `pattern` is a regular expression
  case_insensitive = TRUE
) %>%
  textstat_collocations(min_count = 15)    # identify colloquations that appear at least
  ↪  15 times in the text

print(colqs)
```

```
##           collocation count count_nested length    lambda         z
## 1      climate change    64            0      2  8.363222 11.695783
## 2             we need    18            0      2  5.520952 11.345985
## 3         the climate    26            0      2  3.105557 10.794374
## 4            im going    17            0      2  4.824719  8.833955
## 5 rising temperatures    28            0      2 10.617011  6.802917
## 6      global warming    24            0      2  9.129102  6.230868
## 7       change debate    16            0      2  7.044473  4.883316
```

```
post_colqs <- colqs$collocation

# Join all identified colloquations
for (colloquations in post_colqs) {

  joined_colqs <- gsub(" ", "_", colloquations)

  data_cp$post <- str_replace_all(
      data_cp$post,
      colloquations,
      joined_colqs
    )
}
```

```
# Code check
grep("_", data_cp$post, value = TRUE)[1:5]
```

```
## [1] "global_warming real threat rising_temperatures alarming we_need action now climatechange"
## [2] "the science clear rising_temperatures crisis we must reduce pollution protect planet globalwarmi
## [3] "im worried future rising_temperatures we_need take paris act seriously climateaction"
## [4] "pollution fueling global_warming we_need stricter regulations taxation polluters environment"
## [5] "rising_temperatures impacting ecosystems we must act now protect planet climatechange"
```

Top-down model

Top-down model *- to define your model, you must identify a psychological construct that you want to use in your prediction. You will need to define a list of tokens and then use this dictionary to obtain a score for each social media post. For example, you could calculate the relative frequency of particular term(s) appearing in each post, or use norms (e.g., like the norms developed by Warriner and colleagues) to score each post on some variable (e.g., dominance, valence). You are completely free to choose any psychological constructs (e.g., sentiment, certainty, emotional tone) as long as you provide some rationale for your choice.*

NOTE: Tabula would not allow `.zip` files as submission so the `.csv` file used in the code below is not included as part of the final submission. However, the data is available through the electronic supplementary material from Warriner, A. B., Kuperman, V., & Brysbaert, M. (2013). Norms of valence, arousal, and dominance for 13,915 English lemmas. *Behavior research methods*, *45*, 1191-1207.

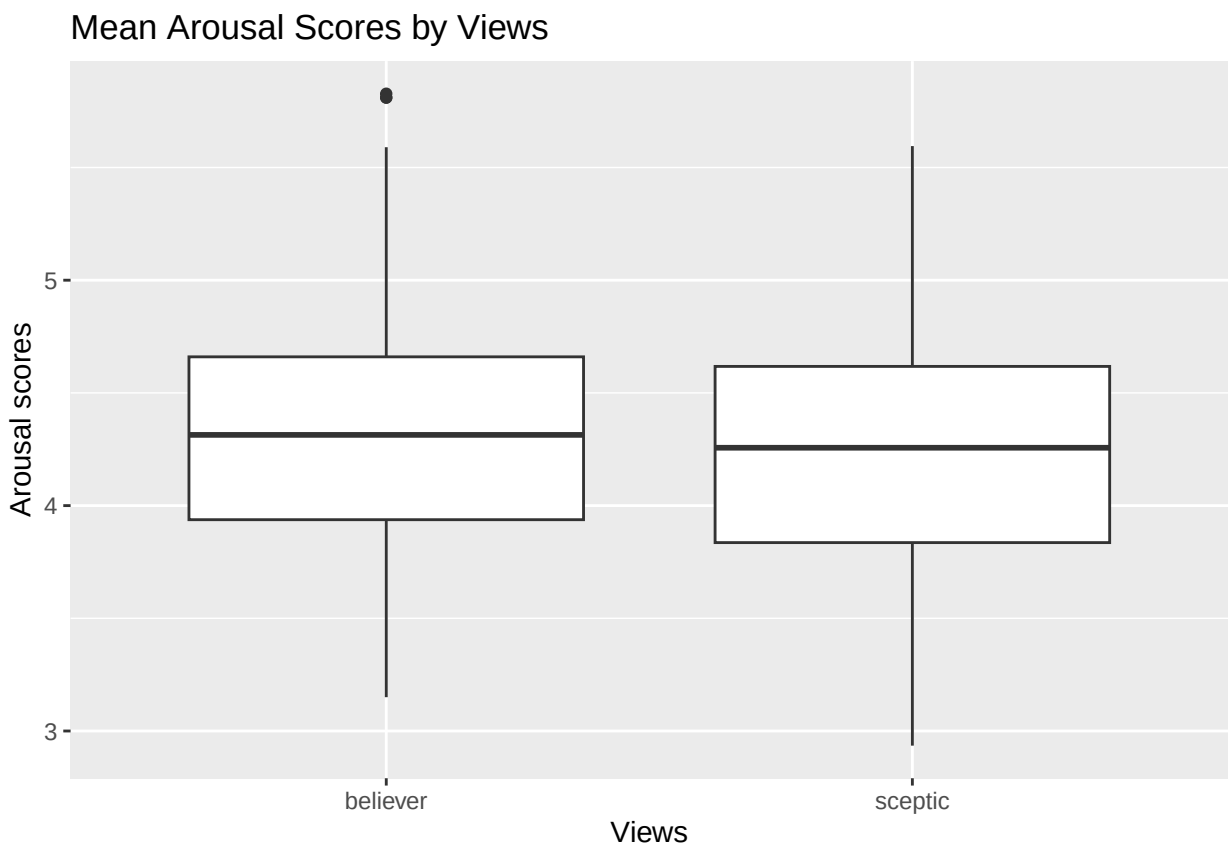
Warriner et al. (2013) defined *arousal* as "the intensity of emotion provoked by a stimulus" (page 1191). Arousal was chosen as the main psychological construct as climate change often elicits intense emotional responses from both skeptics and believers. Therefore, it would be interesting to understand whether there is a difference in the arousal scores of the posts between the two groups.

```
# Import norms
norms <- read_csv("BRM-emot-submit.csv")

# Identify different norms
norms_words <- norms$Word
norms_arousal <- norms$A.Mean.Sum

# Create dictionary
arousal_dict <- setNames(norms_arousal, norms_words)

# Create tokens
word_tokens <- tokens(data_cp$post)

# Function to calculate average arousal score for one tokenised text
arousal_scores <- function(tokens, dict) {

  score <- dict[tokens]    # retrieve scores for tokens
  score <- score[!is.na(score)]    # remove words not in dictionary
  if (length(score) == 0) return(NA)  # if no tokens have a score in the dictionary
  return(mean(score))    # return mean score

}

# Apply the function to each tokenised post in the corpus
cp_arousal <- data_cp

cp_arousal['arousal'] <- sapply(
  word_tokens,
  arousal_scores,
  dict = arousal_dict
)

head(cp_arousal)
```

```
## # A tibble: 6 x 4
##      id post                                                    views arousal
##   <dbl> <chr>                                                 <chr>   <dbl>
## 1     1 global_warming real threat rising_temperatures alarming w~ beli~    5.83
## 2     2 the science clear rising_temperatures crisis we must redu~ beli~    4.41
## 3     3 im worried future rising_temperatures we_need take paris ~ beli~    4.96
## 4     4 pollution fueling global_warming we_need stricter regulat~ beli~    4.32
## 5     5 rising_temperatures impacting ecosystems we must act now ~ beli~    4.08
## 6     6 the paris act start need action combat global_warming cli~ scep~    4.99
```

```
# Calculate mean
cp_arousal_mean <- cp_arousal %>%
  group_by(views) %>%
  summarise( arousal_mean = mean(arousal, na.rm = TRUE) )
```

```r
# Boxplot
ggplot(cp_arousal, aes(x = factor(views), y = arousal)) +
  geom_boxplot(na.rm = TRUE) +
  labs(x = "Views", y = "Arousal scores", title = "Mean Arousal Scores by Views")
```




```r
t.test(arousal ~ views, data = cp_arousal)
```

```
##
##  Welch Two Sample t-test
##
## data:  arousal by views
## t = 1.4149, df = 221.1, p-value = 0.1585
## alternative hypothesis: true difference in means between group believer and group sceptic is not equa
## 95 percent confidence interval:
##  -0.04218536  0.25692579
## sample estimates:
## mean in group believer  mean in group sceptic
##               4.341482               4.234112
```

```r
# Find the text with the lowest and highest scores for each norm
lowest_norm <- cp_arousal[which.min(cp_arousal$arousal), ]
highest_norm <- cp_arousal[which.max(cp_arousal$arousal), ]

cat("Text with the highest arousal score is a", highest_norm$views , "view scoring",
↪  highest_norm$arousal, ":\n\n", highest_norm$post, "\n\n\n")
```

```
## Text with the highest arousal score is a believer view scoring 5.8275 :
```

```
##
##  global_warming real threat rising_temperatures alarming we_need action now climatechange
```

```
cat("Text with the lowest arousal score is a", lowest_norm$views , "view scoring",
↪  lowest_norm$arousal, ":\n\n", lowest_norm$post)
```

```
## Text with the lowest arousal score is a sceptic view scoring 2.935 :
##
##  my nose frozen coldweather winter
```

Bottom-up approach

Bottom-up model *- to build this model, you will need to use a TF-IDF based on the entire vocabulary in the dataset. In order to fit this model, you will need to use a regularized regression, such as Ridge or Lasso.*

```
cp_new <- corpus(cp_arousal, text_field = "post")

summary(cp_new, 5)
```

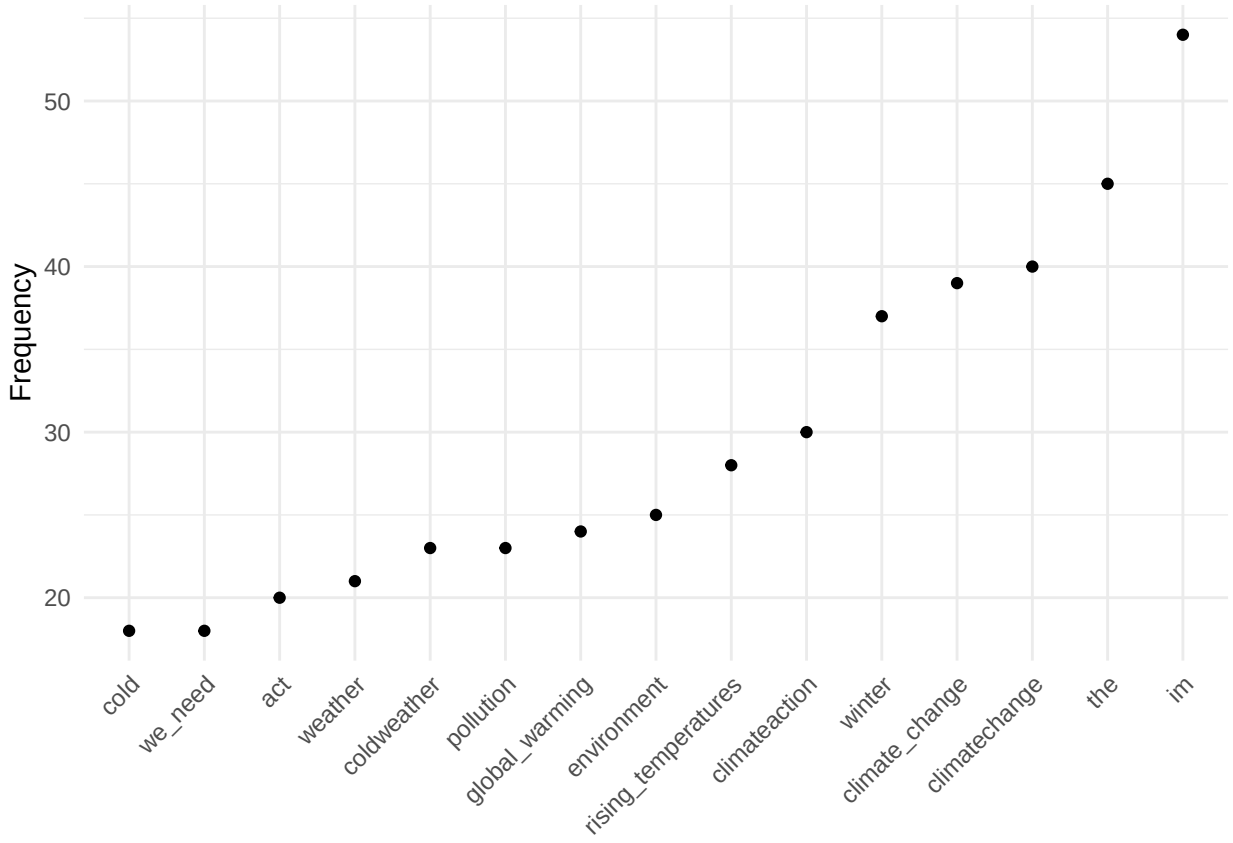
```
## Corpus consisting of 242 documents, showing 5 documents:
##
##   Text Types Tokens Sentences id    views arousal
##  text1     9      9         1  1 believer  5.8275
##  text2    12     12         1  2 believer  4.4075
##  text3    10     10         1  3 believer  4.9625
##  text4     9      9         1  4 believer  4.3200
##  text5    10     10         1  5 believer  4.0850
```

```
cp_new_tokens <- tokens(
  cp_new,
  remove_punct = TRUE,
  remove_symbols = TRUE,
  remove_numbers = TRUE,
  remove_url = TRUE
)

cp_new_dfm <- dfm(cp_new_tokens)   # create a document feature matrix

word_freq <- textstat_frequency(cp_new_dfm, n = 5)   # identifies top 5 most frequent
↪  words

head(word_freq)
```

```
##          feature frequency rank docfreq group
## 1             im        54    1      53   all
## 2            the        45    2      45   all
## 3  climatechange        40    3      40   all
## 4 climate_change        39    4      39   all
## 5         winter        37    5      33   all
```

```
# Visualise most frequent words
cp_new_dfm %>%
  textstat_frequency(n = 15) %>%
  ggplot(aes(x = reorder(feature, frequency), # sorts features based on ascending
  ↪  frequency
             y = frequency)) +
```

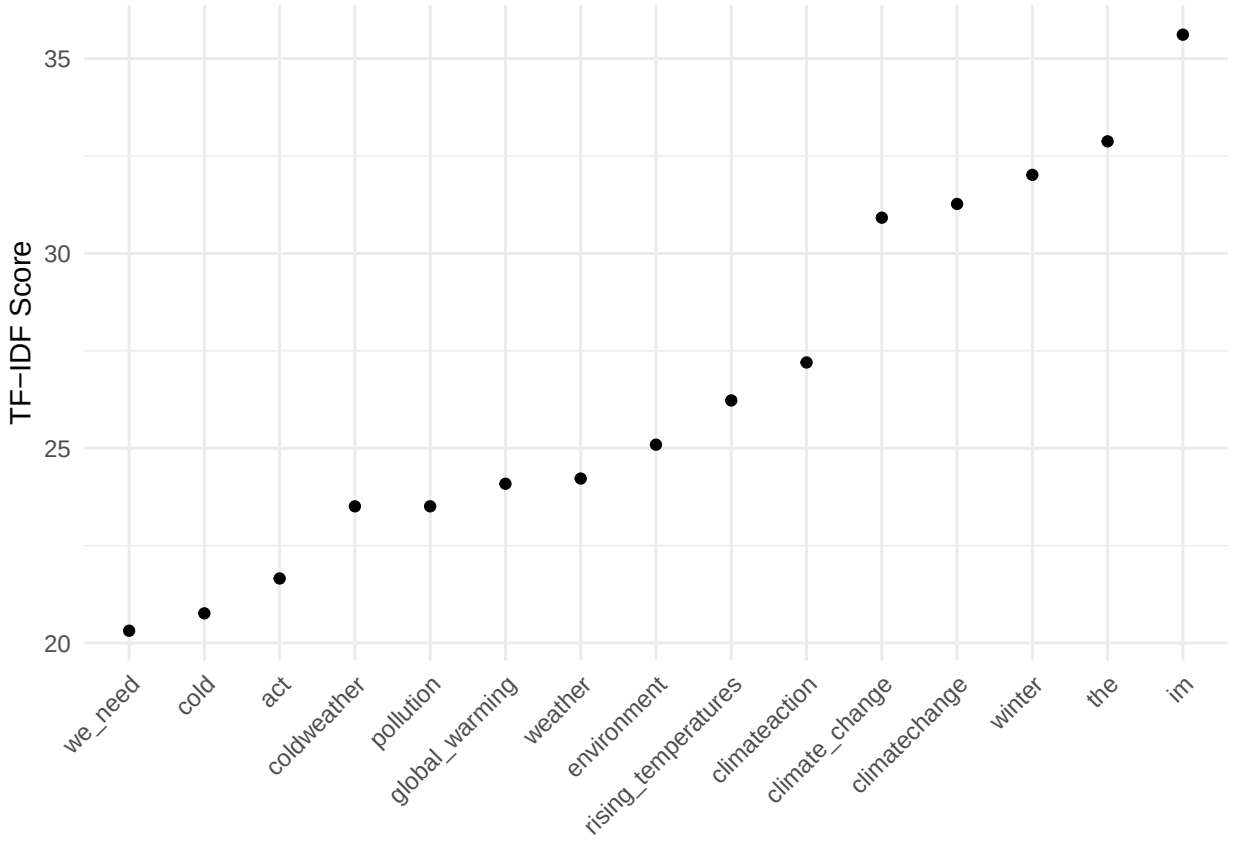
```
  geom_point() +
  labs(x = NULL, y = "Frequency") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
dfm_tfidf <- dfm_tfidf(cp_new_dfm)
print(dfm_tfidf)
```

```
## Document-feature matrix of: 242 documents, 791 features (99.05% sparse) and 3 docvars.
##        features
## docs    global_warming     real   threat rising_temperatures alarming  we_need
##   text1       1.003604 2.082785 1.538717           0.9366573 2.082785 1.128543
##   text2       0        0        0                  0.9366573 0        0
##   text3       0        0        0                  0.9366573 0        1.128543
##   text4       1.003604 0        0                  0         0        1.128543
##   text5       0        0        0                  0.9366573 0        0
##   text6       1.003604 0        0                  0         0        0
##        features
## docs      action      now climatechange       the
##   text1 1.304634 1.480725     0.7817554 0
##   text2 0        0            0         0.7306029
##   text3 0        0            0         0
##   text4 0        0            0         0
##   text5 0        1.480725     0.7817554 0
##   text6 1.304634 0            0         0.7306029
## [ reached max_ndoc ... 236 more documents, reached max_nfeat ... 781 more features ]
```

```r
top_features <- topfeatures(dfm_tfidf, n = 15)   # extracts the 15 most important words
↪  based on their TF-IDF scores

important_words <- data.frame(
  feature = names(top_features),
  score = top_features
)

ggplot(important_words, aes(x = reorder(feature, score), y = score)) +
  geom_point() +
  labs(x = NULL, y = "TF-IDF Score") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```r
# Ridge regression
cp_new_dfm <- cp_new_dfm[!is.na(cp_new_dfm$arousal), ]   # remove na

X <- as.matrix(cp_new_dfm)
y <- cp_new_dfm$arousal

cp_ridge <- cv.glmnet(X, y, alpha = 0)
cp_ridge <- glmnet(X, y, alpha = 0, lambda = cp_ridge$lambda.min)
cp_ridge_coef <- coef(cp_ridge)

cp_ridge_coef %>% max()   # the result is the intercept
```

```
## [1] 4.295177
```

```r
colnames(cp_ridge_coef)
```

```
## [1] "s0"
```

```r
sort(cp_ridge_coef, decreasing = TRUE)[2]
```

```
## [1] 0.2399228
```

```r
cp_ridge_coef
```

```
## 792 x 1 sparse Matrix of class "dgCMatrix"
##                                      s0
## (Intercept)                4.295177e+00
## global_warming             3.283090e-02
## real                       1.569652e-01
## threat                     1.283416e-01
## rising_temperatures        3.164861e-02
## alarming                   1.533479e-01
## we_need                    2.490588e-02
## action                     6.430663e-02
## now                        5.102104e-02
## climatechange              6.899779e-03
## the                       -1.911867e-02
## science                    7.143863e-03
## clear                     -2.302705e-02
## crisis                     2.732397e-02
## we                         3.459504e-05
## must                      -1.077922e-02
## reduce                    -1.801530e-02
## pollution                  1.154800e-02
## protect                   -2.932102e-02
## planet                     1.287880e-02
## globalwarming              6.301582e-02
## im                         3.087228e-03
## worried                    1.100050e-01
## future                     8.192700e-02
## take                       6.408773e-02
## paris                      1.378479e-03
## act                        4.859127e-03
## seriously                  8.201790e-02
## climateaction              4.115009e-02
## fueling                   -5.470012e-03
## stricter                  -5.492208e-03
## regulations               -5.533367e-03
## taxation                  -5.578609e-03
## polluters                 -5.613488e-03
## environment                1.148684e-03
## impacting                 -6.081131e-02
## ecosystems                -2.197393e-02
## start                      8.030645e-02
## need                      -1.960127e-02
## combat                     5.990284e-02
## concerned                 -5.125774e-02
## rising                    -4.283407e-02
```

```
## sea                  -4.285291e-02
## levels               -4.287486e-02
## due                  -1.979774e-02
## coastal              -4.291748e-02
## communities           4.175846e-02
## major                 1.453020e-03
## contributor          -1.178162e-03
## cleaner              -5.101808e-02
## energy                2.877540e-02
## solutions            -4.550087e-03
## undeniable            3.399078e-02
## happening             3.395128e-02
## address              -6.306549e-02
## urgently              3.390575e-02
## causing               4.777558e-02
## extreme               1.225197e-01
## weather               1.060451e-02
## events                1.224657e-01
## prepare              -1.663067e-02
## impact                6.762718e-02
## generations           9.608014e-02
## harming              -9.979439e-02
## health               -3.037675e-02
## contributing          1.477752e-02
## air                  -6.958516e-02
## crucial              -7.067115e-03
## step                  1.690400e-03
## fight                 5.230350e-02
## uphold               -7.178646e-03
## goals                -7.209323e-03
## global               -3.622934e-03
## emergency            -1.958707e-02
## international        -5.969741e-02
## cooperation          -5.973563e-02
## loss                  1.851348e-02
## biodiversity          1.847743e-02
## just                  3.355187e-02
## environmental         7.748311e-02
## issue                 2.176806e-02
## social                8.702282e-02
## justice               8.514072e-02
## transition            8.751808e-03
## renewable             5.370174e-02
## affecting            -1.438919e-02
## food                 -4.148384e-02
## security             -3.480018e-02
## sustainable          -2.976642e-02
## agriculture          -5.025258e-02
## practices            -5.028984e-02
## economic             -1.775719e-02
## invest                1.410756e-02
## green                -1.657345e-02
## problem               1.737683e-02
## requires             -1.047875e-02
```

```
## testament               -3.318757e-02
## commitment              -4.614855e-02
## fighting                -4.057851e-02
## way                     -2.308991e-02
## life                     2.005347e-02
## slow                    -4.778517e-02
## pace                     6.710745e-03
## urgency                  6.551573e-02
## moral                   -7.289629e-03
## responsibility           4.181252e-02
## carbon                  -1.577984e-02
## emissions               -1.580466e-02
## call                    -7.024957e-04
## part                    -7.960831e-02
## dont                     3.902163e-02
## form                     4.837647e-02
## violence                 1.817248e-02
## stop                     1.812555e-02
## symbol                   2.873735e-02
## hope                     1.157471e-01
## reminder                -1.059480e-02
## serious                 -1.366524e-02
## demands                  5.106855e-02
## immediate               -1.372700e-02
## attention               -1.376559e-02
## sign                    -5.428736e-02
## urgent                   7.273093e-02
## climate                 -5.318227e-03
## consequences             2.050297e-01
## inaction                 9.776853e-02
## destroying               3.413488e-02
## threatening              3.409040e-02
## indication              -3.277203e-02
## provides                -1.525135e-02
## framework               -8.660965e-02
## human                   -4.134809e-02
## irrefutable              3.707752e-02
## humancaused              1.435289e-02
## frequent                 1.507732e-01
## intense                  1.507411e-01
## heatwaves                1.606796e-01
## longterm                 6.281775e-02
## effects                  1.242767e-01
## contaminating           -6.754806e-02
## water                    2.240986e-02
## important                6.468815e-03
## towards                  1.041879e-02
## melting                  2.992063e-02
## glaciers                 2.990157e-02
## ice                     -4.534435e-02
## caps                     2.986253e-02
## rate                     2.982372e-02
## habitat                  2.742826e-02
## destruction              2.740962e-02
```

```
## climate_change             2.822089e-02
## lowcarbon                 -3.632456e-02
## economy                    1.619800e-02
## agricultural              -1.914328e-02
## yields                    -1.917696e-02
## potential                  1.912088e-02
## climaterelated             1.910679e-02
## conflicts                  1.907239e-02
## displacement               1.903388e-02
## respiratory               -4.056022e-02
## illnesses                 -4.058483e-02
## problems                  -4.061218e-02
## healthier                 -5.871052e-02
## disrupting                 4.928669e-02
## patterns                   1.196011e-02
## injustice                  7.680779e-02
## disproportionately         7.678021e-02
## vulnerable                 5.464144e-02
## avoid                     -6.910516e-02
## worst                     -6.132764e-02
## impacts                   -6.135052e-02
## change                    -4.479186e-02
## ways                      -1.376719e-01
## legacy                     1.225173e-01
## leaving                    1.224818e-01
## behind                     3.216146e-02
## stain                     -3.167072e-02
## conscience                -3.170868e-02
## clean                     -6.934212e-02
## promise                   -4.841595e-02
## made                      -4.844311e-02
## world                      2.180521e-02
## keep                      -4.847085e-02
## test                      -4.857362e-02
## humanity                   4.789087e-02
## rise                       7.344744e-02
## challenge                  2.917120e-02
## lack                      -2.109217e-01
## political                 -6.168698e-02
## betrayal                   6.747172e-02
## late                      -6.028896e-03
## accelerating              -7.498577e-02
## faster                    -7.501846e-02
## wildfires                  1.394659e-01
## like                       3.578164e-03
## coral                      3.204628e-02
## reefs                      3.200888e-02
## complex                    1.621231e-03
## multifaceted               3.972895e-02
## approach                   2.266664e-02
## resources                  4.238240e-02
## agreement                 -3.388591e-02
## success                    3.440912e-02
## depends                    3.438073e-02
```

```
## national                 3.435616e-02
## costs                   -8.688073e-02
## technologies             3.248686e-02
## distribution            -6.441543e-02
## plant                   -6.443669e-02
## animal                  -6.446963e-02
## species                 -6.450448e-02
## children                 1.180295e-01
## affected                 1.179791e-01
## making                  -7.196200e-03
## destructive              2.399228e-01
## progress                -1.019425e-01
## implementing            -1.019723e-01
## source                   3.339906e-02
## contamination            3.337301e-02
## reliance                -5.632380e-02
## fossil                  -5.633750e-02
## fuels                   -5.635808e-02
## symptom                  1.987650e-03
## deeper                   6.453698e-03
## unsustainable           -9.560134e-03
## violation                5.483090e-02
## beacon                   1.339731e-01
## stark                    1.237572e-01
## ethical                 -8.674628e-02
## dimensions              -8.678680e-02
## consequence             -2.549384e-02
## consumption             -2.552708e-02
## fundamental             -7.938893e-02
## shift                   -7.939568e-02
## values                  -5.517888e-03
## priorities              -2.686479e-02
## psychological            1.736575e-01
## individuals              1.736156e-01
## reflection              -3.264517e-02
## disregard               -3.267016e-02
## inhabitants             -3.270499e-02
## collective              -6.587127e-02
## civilization            -1.072911e-03
## face                    -1.115622e-03
## skeptical                9.177440e-02
## manmade                  6.325535e-02
## the_climate              7.272567e-02
## always                   1.245867e-01
## changed                  1.244897e-01
## naturally                1.243948e-01
## naturalvariation         1.243343e-01
## socalled                -7.541363e-02
## scientific              -7.539494e-02
## consensus               -7.539305e-02
## politically             -7.541214e-02
## motivated               -7.545165e-02
## lackofconsensus         -7.549942e-02
## alarmism                -2.818912e-02
```

```
## control                 5.375247e-02
## people                 -3.671001e-02
## raise                  -2.814584e-02
## taxes                  -2.819250e-02
## climateskepticism      -4.348134e-02
## earth                   7.148447e-03
## warmer                  7.145557e-03
## past                    7.120850e-03
## this                   -2.616004e-02
## natural                -1.149908e-02
## variation               7.087761e-03
## i                      -6.831671e-02
## believe                -2.974377e-02
## its                    -3.945082e-03
## full                   -5.842759e-02
## holes                  -5.847007e-02
## hoax                    8.496892e-02
## perpetrated             8.499472e-02
## liberal                 8.496836e-02
## media                   8.491597e-02
## politicians             3.027158e-02
## fakenews                8.486684e-02
## where                  -7.530013e-02
## evidence                1.288789e-02
## humans                 -5.962470e-02
## convinced              -4.035655e-02
## showmetheproof         -7.532375e-02
## models                  2.025438e-02
## flawed                  2.024457e-02
## unreliable              2.020393e-02
## they                    2.161806e-02
## cant                   -3.661733e-02
## predict                 2.013999e-02
## accurately              2.008959e-02
## climatemodels           2.006217e-02
## co                     -2.538562e-03
## trace                  -5.036116e-03
## gas                    -5.026992e-03
## it                     -5.047540e-03
## possibly               -5.086982e-03
## big                    -5.129720e-03
## sun                     3.434591e-02
## main                   -3.689461e-02
## driver                 -2.999655e-02
## activity               -3.002727e-02
## solaractivity          -3.006789e-02
## cycle                  -6.075153e-02
## weve                   -6.075750e-02
## ages                   -6.078460e-02
## warm                   -8.272270e-02
## periods                -6.082188e-02
## naturalcycles          -6.085967e-02
## scientists             -2.181760e-02
## disagree               -2.182464e-02
```

```
## narrative                -5.050806e-02
## silenced                  8.570145e-03
## censorship               -2.189637e-02
## well                      8.262718e-04
## adapt                     7.956000e-04
## adaptation                7.582046e-04
## high                     -4.754066e-02
## worth                    -4.756502e-02
## economicimpact           -4.759926e-02
## policies                  2.784418e-02
## hurt                      2.789931e-02
## poor                      4.190797e-02
## developing                5.668501e-02
## countries                 4.183242e-02
## unfairpolicies            4.177062e-02
## focus                     7.097587e-02
## new                       7.092374e-02
## restricting               7.086700e-02
## innovation                7.082692e-02
## the_climate_change_debate 8.253303e-03
## politicized              -1.498982e-02
## balanced                 -1.499513e-02
## discussion               -1.502458e-02
## politics                 -1.506399e-02
## denying                  -4.351301e-02
## changing                 -4.576371e-02
## cause                    -4.353151e-02
## naturalcauses            -4.355440e-02
## circumstantial            1.220389e-01
## theres                    1.220035e-01
## smoking                   1.219324e-01
## gun                       1.218648e-01
## lackofevidence            1.218244e-01
## open                     -1.456818e-03
## possibility              -8.368344e-03
## convincing               -8.409296e-03
## showmethedata            -8.456862e-03
## the_climate_change        4.246864e-02
## movement                  4.301967e-02
## driven                    9.771802e-02
## fearmongering             9.769790e-02
## exaggeration              9.766955e-02
## hysteria                  1.436672e-01
## tired                     2.289150e-02
## doomsday                  2.290236e-02
## predictions               2.289026e-02
## never                    -5.293406e-02
## come                      4.099069e-03
## true                      2.286841e-02
## failedpredictions         2.282436e-02
## average                  -3.648690e-02
## person                   -3.648610e-02
## understand               -3.650983e-02
## confusing                -3.654585e-02
```

```
## scientist                -7.873989e-02
## common                   -8.357340e-02
## sense                    -3.934287e-02
## doesnt                   -7.868518e-02
## make                     -7.871527e-02
## commonsense              -7.876418e-02
## alarmists                 4.260683e-02
## using                     6.694932e-02
## scare                     6.696195e-02
## tactics                   6.691291e-02
## push                      6.683823e-02
## agenda                    6.677223e-02
## manipulation              6.673404e-02
## funded                    5.833371e-02
## government                5.831120e-02
## grants                    5.826876e-02
## biasedscience             5.822626e-02
## power                     1.491741e-01
## politicalagenda           1.491402e-01
## im_going                  3.279732e-02
## lifestyle                 4.847559e-02
## unproven                  4.845834e-02
## theory                    4.842315e-02
## mychoice                  4.838644e-02
## religion                  6.967254e-02
## climatechangecult         6.963839e-02
## trust                     5.455046e-02
## technology                5.454326e-02
## solve                     5.451529e-02
## technologicalsolutions    5.448083e-02
## distraction               2.470925e-03
## issues                    2.458263e-03
## realproblems              2.429263e-03
## existential               1.943273e-01
## overblown                 1.942958e-01
## trying                    9.199327e-03
## away                      9.174103e-03
## freedoms                  9.132121e-03
## liberty                   9.090642e-03
## lectured                 -2.455981e-02
## celebrities              -2.454462e-02
## fly                      -2.455233e-02
## private                  -2.457903e-02
## jets                     -2.461724e-02
## hypocrisy                -2.465543e-02
## elitist                  -4.491861e-02
## touch                    -4.493901e-02
## ordinary                 -4.497370e-02
## ivorytower               -4.500987e-02
## sacrifice                 5.368225e-02
## standard                  5.368713e-02
## living                    5.366091e-02
## hypothetical              5.362421e-02
## qualityoflife             5.359213e-02
```

```
## based                      .
## emotions                   .
## facts                      .
## emotionalarguments         .
## bullied                    1.522519e-01
## believing                  1.522285e-01
## skepticism                 1.521931e-01
## antihuman                 -3.657801e-02
## antiprogress              -7.318967e-02
## support                    1.367763e-02
## cost                       1.362348e-02
## jobs                       1.356112e-02
## economicgrowth             1.351198e-02
## decadent                  -2.487169e-03
## society                   -1.988275e-02
## decadence                 -2.500884e-03
## let                        4.633396e-02
## ruin                       5.325469e-02
## enjoyment                  5.321459e-02
## enjoylife                  5.317422e-02
## insanity                   1.128637e-01
## madness                    1.128327e-01
## victim                     1.884876e-01
## rationality                1.884539e-01
## waste                     -1.071183e-01
## time                      -6.363929e-02
## wastedresources           -1.071201e-01
## freedom                    3.894365e-02
## democracy                  1.341741e-01
## authoritarianism           1.341420e-01
## freespeech                 3.892213e-02
## battle                     1.431278e-01
## good                       2.544091e-03
## evil                       1.431275e-01
## moralbattle                1.430965e-01
## give                       9.254486e-02
## optimism                   9.250927e-02
## force                     -2.849236e-03
## positivechange            -2.884769e-03
## regardless                -1.201112e-01
## beliefs                   -1.201306e-01
## environmentalism          -1.201600e-01
## opportunity                9.407823e-02
## us                         4.191088e-02
## together                  -1.470024e-02
## find                      -1.472623e-02
## unity                     -1.475739e-02
## educate                   -1.669514e-02
## opinion                   -1.671948e-02
## criticalthinking          -1.675273e-02
## connected                 -8.770179e-02
## share                     -8.769110e-02
## fate                      -8.770839e-02
## interconnectedness        -8.774429e-02
```

```
## faith                     5.214253e-02
## humanitys                 5.215141e-02
## ability                   5.212714e-02
## overcome                  5.208850e-02
## faithinhumanity           5.205226e-02
## better                    1.091187e-01
## stewards                  1.522395e-02
## stewardship               1.520623e-02
## climate_change_debate     5.528502e-03
## mind                      5.534382e-03
## willingness               5.510880e-03
## listen                    5.471535e-03
## openmindedness            5.431915e-03
## chance                   -3.272410e-02
## learn                    -3.707598e-02
## grow                     -3.710097e-02
## learning                 -3.713605e-02
## solution                 -2.700587e-02
## solutionoriented         -2.703527e-02
## live                     -4.872323e-02
## respectful               -4.873093e-02
## sustainableliving        -4.875420e-02
## something                -1.278923e-01
## bigger                   -1.278949e-01
## bigpicture               -1.279196e-01
## grateful                  1.949399e-02
## beauty                    1.948863e-02
## wonder                    1.946623e-02
## gratitude                 1.943683e-02
## create                    2.020562e-01
## enjoying                 -1.359840e-02
## cup                      -7.501786e-02
## hot                       9.252264e-03
## cocoa                    -7.494732e-02
## cold                     -6.867573e-02
## winter                   -5.397061e-02
## night                    -7.493044e-02
## hotcocoa                 -7.498196e-02
## can                      -1.233471e-01
## barely                   -9.366173e-02
## type                     -1.382491e-01
## coldweather              -4.183665e-02
## typing                   -1.382572e-01
## spending                  2.145086e-01
## money                     2.145068e-01
## drinks                    2.144765e-01
## hotdrinks                 2.144380e-01
## cloudy                   -2.806911e-02
## sunshine                  2.549588e-03
## typical                  -2.806112e-02
## spring                    6.083958e-03
## springweather            -2.808384e-02
## clouds                   -6.040223e-02
## officially               -7.642307e-03
```

```
## declaring                 6.488041e-02
## war                       3.242839e-02
## makes                    -5.069776e-02
## want                     -2.300793e-02
## stay                     -3.909372e-02
## bed                      -5.065619e-02
## day                      -9.448576e-03
## lazyday                  -5.066887e-02
## my                       -4.201829e-02
## dog                       8.876587e-02
## refuses                   8.877793e-02
## go                        8.876183e-02
## outside                  -2.760322e-02
## dogproblems               8.872188e-02
## anyone                   -7.917139e-02
## elses                    -3.807711e-02
## car                      -3.478542e-02
## struggling               -3.805493e-02
## coldweatherproblems      -8.136596e-02
## carproblems              -3.809360e-02
## commute                  -1.145297e-02
## going                    -4.854012e-02
## fun                      -1.345384e-03
## sarcasm                   2.700930e-03
## hard                     -1.958921e-02
## decide                   -1.957959e-02
## wear                     -6.859319e-02
## fashion                  -1.958943e-02
## grumpy                    4.745367e-02
## apartment                -1.437216e-01
## colder                   -1.437263e-01
## coldapartment            -1.437447e-01
## foggy                    -6.248778e-02
## morning                  -6.246516e-02
## drive                    -6.244832e-02
## carefully                -6.244755e-02
## everyone                 -4.303392e-02
## fog                      -4.671972e-04
## foggyweather             -6.245210e-02
## soaking                   1.348519e-01
## last                      1.348473e-01
## rays                      1.348317e-01
## summer                    3.088776e-02
## beautiful                 2.367020e-02
## sunset                    2.083275e-03
## what                      2.571144e-02
## end                      -6.328917e-02
## beautifulsky              4.136650e-03
## appreciate               -1.026321e-01
## warmth                   -2.294637e-02
## home                     -5.132102e-02
## forever                  -1.302438e-01
## is                       -7.996424e-02
## ever                     -1.406104e-01
```

```
## bottle                -6.880034e-02
## lifeline              -6.881562e-02
## hotwaterbottle        -6.884623e-02
## chocolate              2.459929e-02
## thing                  2.462809e-02
## keeping                2.462090e-02
## alive                  2.458988e-02
## right                  2.063080e-03
## hotchocolate           2.452530e-02
## starting              -6.075953e-02
## myth                  -8.459905e-04
## unpredictable          1.997521e-02
## days                   3.994144e-02
## humid                 -4.957491e-02
## today                 -2.580555e-03
## hydrated              -4.955121e-02
## summerheat            -4.958574e-02
## humidity              -4.963048e-02
## shining                5.572310e-02
## birds                  3.291838e-02
## chirping               5.570902e-02
## sunnyday               2.257498e-02
## pretty                 2.390405e-02
## sure                   2.394113e-02
## blood                  4.053612e-02
## turned                 4.050883e-02
## iceblood               4.047523e-02
## vacation               1.913538e-02
## beach                  1.856909e-02
## stat                   1.917115e-02
## vacationneeded         1.914626e-02
## wearing               -4.825577e-02
## many                  -4.825402e-02
## layers                -3.118391e-03
## move                  -2.383810e-02
## curl                  -6.535330e-02
## book                  -3.615084e-02
## leave                 -5.003248e-02
## house                 -5.006027e-02
## hibernating           -6.532905e-02
## hibernation           -2.825576e-02
## perfect               -1.044818e-02
## soak                   1.785409e-02
## enjoy                  1.784652e-02
## waves                  1.782103e-02
## beachday               1.779075e-02
## fresh                 -1.682257e-01
## rain                  -5.613590e-02
## freshair              -1.682046e-01
## wind                  -1.530216e-02
## picking               -6.409451e-02
## looks                 -6.408787e-02
## might                 -6.410842e-02
## get                   -6.414310e-02
```

```
## electric                -1.097764e-01
## blanket                 -7.362496e-02
## best                    -1.097520e-01
## friend                  -1.097680e-01
## electricblanket         -1.097983e-01
## crave                   -5.413053e-02
## comfort                 -5.412587e-02
## comfortfood             -5.413718e-02
## question                 1.591953e-02
## choices                  1.592008e-02
## existentialcrisis        1.590153e-02
## cutting                 -2.044096e-02
## windy                    1.865008e-02
## hibernate                9.047487e-03
## ugh                     -6.036086e-02
## brutal                  -6.038157e-02
## first                   -1.212963e-01
## frost                   -6.065569e-02
## season                  -8.480808e-02
## definitely              -1.213064e-01
## flowers                  9.851768e-03
## blooming                 9.863550e-03
## singing                  9.860528e-03
## newbeginnings            9.847112e-03
## howling                  5.762819e-02
## tonight                  5.766826e-02
## sounds                   5.766515e-02
## storm                    5.255830e-02
## brewing                  5.761759e-02
## stormyweather            5.757694e-02
## turning                 -1.799025e-01
## cube                    -1.799175e-01
## sunny                   -1.072098e-02
## walk                    -1.073552e-02
## park                    -5.199878e-02
## rainy                   -8.008159e-02
## blues                   -8.007810e-02
## rainyday                -8.009668e-02
## cozy                    -8.012835e-02
## loving                  -8.490383e-02
## crisp                   -8.489129e-02
## autumn                  -4.359925e-02
## leaves                  -6.644104e-02
## colorful                -8.477839e-02
## fallcolors              -8.480333e-02
## brrr                    -1.206537e-01
## getting                 -1.206630e-01
## coat                    -1.206888e-01
## forecast                -1.594082e-02
## calling                 -1.590036e-02
## thunderstorms           -1.588082e-02
## safe                    -1.588706e-02
## thunderstorm            -1.591462e-02
## weatherforecast         -1.595107e-02
```

```
## snow                  1.224697e-02
## falling               2.448860e-02
## winterwonderland      2.446964e-02
## whats                -8.962076e-03
## globalweather        -8.966372e-03
## color                -4.747982e-02
## quickly              -4.744311e-02
## favorite             -4.744059e-02
## autumnleaves         -4.747013e-02
## fall                 -4.751532e-02
## coming               -1.264021e-01
## winteriscoming       -1.080746e-01
## chilly                5.691512e-02
## winterdays            5.689282e-02
## buckets              -1.436877e-01
## heavyrain            -1.437028e-01
## gorgeous              4.702628e-02
## rainbow               2.351250e-02
## afterthestorm         4.701338e-02
## picnic               -4.640871e-02
## outdoors             -2.429424e-02
## love                 -1.364356e-01
## watching             -1.364123e-01
## drift                -1.364077e-01
## sky                  -1.364238e-01
## peacefulness         -3.963049e-02
## snowy                -3.961964e-02
## snowday              -3.962933e-02
## peaceful             -3.965326e-02
## rolling               3.005411e-02
## mysterious            3.005396e-02
## mystery               3.004369e-02
## bike                  4.442764e-02
## ride                  4.441157e-02
## bikeride              4.439063e-02
## everything           -6.733580e-02
## lush                 -6.732416e-02
## greenery             -6.733071e-02
## bones                -4.592522e-03
## aching               -4.501816e-03
## chill                -4.441276e-03
## send                 -4.420224e-03
## help                 -4.436411e-03
## blankets             -4.479258e-03
## freezing             -4.533639e-03
## winterwoes           -4.583294e-03
## feel                 -1.239897e-01
## fingers              -1.239984e-01
## heating               7.636403e-02
## bill                  7.636535e-02
## insane                7.634831e-02
## expensive             7.632461e-02
## nose                 -2.350050e-01
## frozen               -2.350266e-01
```

```
## layering                 8.099224e-02
## onion                    8.098245e-02
## unproductive            -1.156071e-01
## procrastination         -1.156157e-01
## ive                      7.159527e-03
## developed                7.151905e-03
## permanent                7.125610e-03
## shiver                   7.094455e-03
## shivering                7.068860e-03
## early                    1.974953e-02
## dreaming                -2.183121e-02
## summerdreaming           1.975214e-02
## sweaters                 .
## miss                    -3.334198e-02
## socially                -1.170189e-01
## acceptable              -1.170052e-01
## snowsuit                -5.850360e-02
## everywhere              -1.170239e-01
## lips                    -8.368740e-02
## chapped                 -8.366665e-02
## skin                    -8.366128e-02
## dry                     -8.367502e-02
## dryskin                 -8.370194e-02
## ready                    2.615595e-02
## please                   1.996224e-02
## hug                     -1.843486e-02
## and                     -3.687878e-02
## covered                 -3.121768e-02
## times                   -3.122712e-02
## icedcar                 -3.124683e-02
## walking                 -7.629081e-02
## penguin                 -7.628275e-02
## slipping                -7.629241e-02
## tropical                 8.127332e-04
## island                   8.215142e-04
## tropicalisland           8.148890e-04
## does                    -1.195574e-01
## tips                    -5.977856e-02
## staying                 -1.195744e-01
## beaches                 -6.323878e-02
## warmweather             -6.325101e-02
## see                     -1.803547e-01
## breath                  -9.018489e-02
## think                   -1.201699e-01
## moved                   -1.201680e-01
## south                   -6.009041e-02
## already                  3.210720e-02
## bring                   -2.232320e-02
```

Inspect & Evaluate Results

Inspect and evaluate your results. For assessing the models' overall performance, you could report the confusion matrix. For the top-down model, you may want to report (and visualize) the influence or importance of your chosen features. For the bottom-up model, you can report (and visualize) the most predictive tokens from your

model (i.e., those with the largest absolute coefficient values).

For the top-down model, a Welch two-sample t-test was conducted and found no significant difference in the group arousal means between the believer ($M = 4.34$) and sceptic ($M = 4.23$) groups. Text with the highest arousal score (5.83) is a believer view: "global_warming real threat rising_temperatures alarming we_need action now climatechange". Text with the lowest arousal score (2.94) is a sceptic view: "my nose frozen coldweather winter".

For the bottom-up model, the most predictive token was destructive with a coefficient of 0.240.