labJournal2

Keiichiro Watanabe

2023-09-04

Part 0: Let's discuss the previous lab.

R has a unique ecosystem called Tidyverse that simplifies data manipulation and visualization, making it easier to work with data in a "tidy" format.

RStudio isn't just an IDE; it's a comprehensive toolkit for data science, featuring Git integration, R Markdown support, and much more.

R excels at vectorized operations, allowing for efficient data manipulation without explicit loops, which speeds up data analysis.

Part 1: Data Manipulation in R

1

getwd() : returns the current working directory in a string.

setwd(): set the current working environment.

2

Tibbles is a formatting tool.

3

The %<>% operator is often referred to as the compound assignment pipe-operator and comes from the magrittr package in R

```r
library(magrittr)

x <- c(1, 2, 3, 4, 5)

# Use %<>% to square the values and then subtract 1
x %<>% (function(y) y^2) %>% subtract(1)

print(x)
```

```
## [1]  0  3  8 15 24
```

4

The str function in R is used for compactly displaying the internal structure of an R object. This function provides a quick and convenient way to get an overview of the data types, dimensions, and contents of lists, vectors, matrices, data frames, and more.

Type of Object: Tells you whether it's a vector, list, data frame, etc.

Dimensions: For matrices and data frames, it shows the number of rows and columns.

Content: A preview of the actual data contained within the object.

Attributes: Any additional attributes that the object might have.

5

1

```r
my_data <- read.csv("/Users/keiichiro_watanabe/Desktop/CSC324/IndividualProject/data/OnlineNewsPopulari
```

2

Mutate

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
# Adding a new column "content_ratio" that is the ratio of n_tokens_content to n_tokens_title
my_data <- my_data %>%
  mutate(content_ratio = n_tokens_content / n_tokens_title)
```

Summarize

```r
# Summarizing to find the average number of tokens in the content
summary_result <- my_data %>%
  summarize(avg_n_tokens_content = mean(n_tokens_content))
```

Arrange

```r
# Sorting the data by the number of shares, in descending order
my_data_sorted <- my_data %>%
  arrange(desc(shares))
```

Filter

```r
# Filtering rows where data_channel_is_tech is 1.
my_data_tech <- my_data %>%
  filter(data_channel_is_tech == 1)

# Filtering rows where the number of shares is greater than 1000.
my_data_high_shares <- my_data %>%
  filter(shares > 1000)

# Filtering rows where the article was posted on a weekend.
my_data_weekend <- my_data %>%
  filter(is_weekend == 1)
```

Select

```r
# Selecting columns related to token counts
# -----------------------------------------------------
my_data_tokens <- my_data %>%
  select(n_tokens_title, n_tokens_content)
# -----------------------------------------------------

# Selecting columns related to data channels
# -----------------------------------------------------
my_data_channels <- my_data %>%
  select(starts_with("data_channel"))
# -----------------------------------------------------

# Selecting columns related to weekdays
# -----------------------------------------------------
my_data_weekdays <- my_data %>%
  select(weekday_is_monday:weekday_is_sunday)
# -----------------------------------------------------

# Selecting columns where the average token length is greater than 4
# ----------------------------------------------------------------------------
my_data_avg_token_length <- my_data %>%
  select(average_token_length) %>%
  filter(average_token_length > 4)
# ----------------------------------------------------------------------------

# Selecting columns with logical conditions on n_unique_tokens and n_non_stop_unique_tokens
# ----------------------------------------------------------------------------------------------------
my_data_logical <- my_data %>%
  select(n_unique_tokens, n_non_stop_unique_tokens) %>%
  filter(n_unique_tokens > 0.5 & n_non_stop_unique_tokens > 0.7)
```

3

```r
library(tidyr)
```

```
##
## Attaching package: 'tidyr'

## The following object is masked from 'package:magrittr':
##
##     extract
```

```r
# This will gather all the 'data_channel' columns into key-value pairs
long_data <- my_data %>%
  gather(key = "channel_type", value = "channel_value", starts_with("data_channel"))
# --------------------------------------------------------------

# This will spread the 'channel_type' and 'channel_value' back into separate columns
wide_data <- long_data %>%
  spread(key = "channel_type", value = "channel_value")
# --------------------------------------------------------------

# This will unite 'weekday_is_monday' to 'weekday_is_sunday' into a new column 'weekdays_combined'
united_data <- my_data %>%
  unite("weekdays_combined", weekday_is_monday:weekday_is_sunday, sep = ",")
# --------------------------------------------------------------------------

# This nests the data frame by 'num_keywords', storing the nested 'data' in a new column
nested_data <- my_data %>%
  nest(data = c(n_tokens_title, n_tokens_content, num_keywords))
```

Part 2: Reflection and questions on readings

1

Design is the third component in the life cycle od software development.Use design to understand the larger problem you are working with.

2

1. principle1: Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Example: I am working on an e-commerce platform. Instead of waiting for six months to a year to release a complete, feature-packed application, the team adopts an Agile approach.They prioritize building the most essential features first, such as user registration and item listings, and release it to a select group of users for testing within the first few weeks.

2. welcome changing requirements even late in development. Agile processes harness change for the customer's competitive advantage.

Example: I am building a health monitoring app that tracks physical activities, sleep, and dietary habits. Three months into development, a new type of wearable device gains popularity, and the client wants to incorporate compatibility with this device. In a non-Agile model, accommodating this new requirement could be problematic, costly, and time-consuming. However, in an Agile framework, the team would welcome this change.

3

1. Color-coded Map Markers Mark: The points on the U.S. map in the center of the visualization represent golf courses. Channel: Color (green for public courses, brown for private courses)

Benefits: The use of color helps viewers instantly differentiate between public and private courses without having to refer to a legend or click on each point for details. Colors are easy to understand and don't add visual clutter to the map.

2. Line Charts for Rankings Mark: Lines connecting bubbles on the chart. Channel: Position on a vertical axis (for rankings), and a horizontal axis (for years). The colors of the lines correspond to the upper-right filters for the magazines' rankings.

Benefits: Line charts are excellent for showing trends over time. The ranking of a course across various years can be easily interpreted.

Reference

Name(s) of all authors: Keiichiro Watanabe, Megan Bernacchi

Lab: Lab Journal 2

Due Monday, 11 September 2023

Written/online sources used: (enter "none" if no sources other than the textbook and course website used) : none

Help obtained (Acknowledgments): none

"I/we confirm that the above list of sources is complete AND that I/we have not talked to anyone else about the solution to this problem."