

Dataframes in R: Takeaways

by Dataquest Labs, Inc. - All rights reserved © 2020

Concepts

- Tabular data is organized into rows, where one row represents a single entity and columns represent different characteristics of this row.
- Microsoft Excel, Google Sheets, and CSV files are common ways that we see tabular data.
- Tibbles are a data structure that implements tabular data in R and the `tidyverse`.
- Piping enables us to create pipelines with all of the functions we learned, allowing us to convert raw data in tibbles to more refined datasets.

Syntax

- Import a dataset:

```
library(readr)
data <- read_csv("name_of_file_with_data.csv")
```

- Learn about a tibble's columns, types and dimensions:

```
> glimpse(recent_grads)
Observations:  173
Variables:    18
$ Rank  1, 2...
$ Major_code 2419, 2416...
$ Major  "PETROLEUM ENGINEERING" "MINING AND MINERAL ENGINEERING"...
```

- Return the number of rows or columns from a tibble:

```
nrow(data) # returns the number of rows in `data`  
ncol(data) # returns the number of rows in `data`
```

- Pick columns to keep or remove from your data:

```
# Keeping data  
filtered_data <- select(recent_grads, Rank, Major)  
# Removing data  
filtered_data <- select(recent_grads, - College_jobs)
```

- Filter rows based on conditions:

```
top_100_majors <- filter(recent_grads, Rank < 100)
```

- Chain together tidyverse functions into a pipeline:

```
library(dplyr)  
low_total_ranked_majors <- recent_grads %>%  
  select(., Rank, Major, Total) %>%  
  filter(., ranked_majors, Total < 2000)
```

- Create new columns:

```
new_recent_grads <- recent_grads %>%  
  mutate(  
    prop_male = Men/ Total  
  )
```

- Sort data by a particular or multiple columns:

```
new_recent_grads <- recent_grads %>%
  mutate(
    prop_male = Men/ Total
  ) %>%
  arrange( -prop_male)
```

- Use `head()` to return just the first few rows of a tibble

```
> head(new_recent_grads)
# A tibble: 6 x 3
  Total    Menprop_male
1   124    124         1
2  4790   4419       0.923
3 18498  16820       0.909
4   756    679       0.898
5  1258   1123       0.893
6 91227  80320       0.880
```

- Use `summarize()` to calculate some summary values based on entire columns:

```
summary_table <- recent_grads %>%
  summarize(
    avg_unemp= mean(Unemployment_rate),
    min_unemp= min(Unemployment_rate),
    max_unemp= max(Unemployment_rate)
  )
```

