Introduction to Data Analysis in R

Derek Sollberger

2022-05-31

Contents


```
library("corrplot")
library("forcats")
library("janitor")
library("skimr")
library("tidyverse")
```

Today's Data Set

This survey is the result of a partnership between Axios and Harris Poll to gauge the reputation of the most visible brands in America, based on 20 years of Harris Poll research. From Trader Joe's to Disney, here's how this year's class stacks up.

"Methodology: The Axios Harris Poll 100 is based on a survey of 33,096 Americans in a nationally representative sample conducted March 11-April 3, 2022. The two-step process starts fresh each year by surveying the public's top-of-mind awareness of companies that either excel or falter in society.

"These 100 "most visible companies" are then ranked by a second group of Americans across the seven key dimensions of reputation to arrive at the ranking. If a company is not on the list, it did not reach a critical level of visibility to be measured."

Since we have CSV files (comma-separated values), we can use the `read_csv` function in the `readr` package to load the file into our current programming session.

```
reputation <- readr::read_csv("reputation.csv",
                              name_repair = janitor::make_clean_names)
reputation_wide <- readr::read_csv("reputation_wide.csv",
                                   name_repair = janitor::make_clean_names)
```

Exploring the Data

At this point, you can look at the `environment` pane (upper-right area in RStudio), and click on the name of a data frame (e.g. `reputation`) to open up a viewer to take a look at the data.

Another way to take a quick look at the data is with the `head` command to view the first few rows.

```
head(reputation)
```

```
## # A tibble: 6 x 5
##   company      industry name        score  rank
##   <chr>        <chr>    <chr>       <dbl> <dbl>
## 1 Trader Joe's Retail   TRUST        82.7     3
## 2 Trader Joe's Retail   ETHICS       82.5     2
## 3 Trader Joe's Retail   GROWTH       84.1     2
## 4 Trader Joe's Retail   P&S          83.5     9
## 5 Trader Joe's Retail   CITIZENSHIP  80       3
## 6 Trader Joe's Retail   VISION       81.9    13
```

```
head(reputation_wide)
```

```
## # A tibble: 6 x 9
##   company          industry trust ethics growth   p_s citizenship vision culture
##   <chr>            <chr>    <dbl>  <dbl>  <dbl> <dbl>       <dbl>  <dbl>   <dbl>
## 1 Trader Joe's     Retail    82.7   82.5   84.1  83.5        80     81.9    83.1
## 2 HEB Grocery      Retail    83.7   81.8   83.6  83.1        81     81.3    81
## 3 Patagonia        Retail    81.3   81.7   81.9  83.7        80.8   82.2    82.9
## 4 The Hershey Com~ Food & ~  79.9   79.8   82.3  81.4        75.2   81.8    79.1
## 5 Wegmans          Groceri~  80.7   81.4   83.1  81.1        78.6   80.9    81.7
## 6 Samsung          Tech      79.8   80.2   83.8  84.3        75     84.1    81.6
```

For our purposes, we should look at the structure of each data set. In R, this is processed with the `str` command. In particular, this view clarifies which columns are numerical and which are categorical.

```
str(reputation, give.attr = FALSE)
```

```
## spec_tbl_df [700 x 5] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ company : chr [1:700] "Trader Joe's" "Trader Joe's" "Trader Joe's" "Trader Joe's" ...
##  $ industry: chr [1:700] "Retail" "Retail" "Retail" "Retail" ...
##  $ name    : chr [1:700] "TRUST" "ETHICS" "GROWTH" "P&S" ...
##  $ score   : num [1:700] 82.7 82.5 84.1 83.5 80 81.9 83.1 83.7 81.8 83.6 ...
##  $ rank    : num [1:700] 3 2 2 9 3 13 1 1 4 4 ...
```

```
str(reputation_wide, give.attr = FALSE)
```

```
## spec_tbl_df [100 x 9] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ company    : chr [1:100] "Trader Joe's" "HEB Grocery" "Patagonia" "The Hershey Company" ...
##  $ industry   : chr [1:100] "Retail" "Retail" "Retail" "Food & Beverage" ...
##  $ trust      : num [1:100] 82.7 83.7 81.3 79.9 80.7 79.8 79.9 78.7 78.9 78.2 ...
##  $ ethics     : num [1:100] 82.5 81.8 81.7 79.8 81.4 80.2 78.9 79.7 79.7 77.2 ...
##  $ growth     : num [1:100] 84.1 83.6 81.9 82.3 83.1 83.8 82.9 83.5 83 81 ...
##  $ p_s        : num [1:100] 83.5 83.1 83.7 81.4 81.1 84.3 81.8 83.6 83.6 83.5 ...
##  $ citizenship: num [1:100] 80 81 80.8 75.2 78.6 75 77 76.6 74.2 73.5 ...
##  $ vision     : num [1:100] 81.9 81.3 82.2 81.8 80.9 84.1 84.1 82.4 82.4 80.1 ...
##  $ culture    : num [1:100] 83.1 81 82.9 79.1 81.7 81.6 79.1 80.2 81.5 79.5 ...
```

Sample Statistics

To get a sense of the numbers, we can compute sample statistics (such as the mean, median, and standard deviation) for a numerical variable.

```
reputation |>
  summarize(mean = mean(score, na.rm = TRUE),
            median = median(score, na.rm = TRUE),
            sd = sd(score, na.rm = TRUE))
```

```
## # A tibble: 1 x 3
##    mean median    sd
##   <dbl>  <dbl> <dbl>
## 1  74.8   75.9  6.00
```

The R programming language becomes really useful when we want to perform our tedious calculations across several categories. Notice how the inclusion of one line of code below helps us compute the sample statistics for each survey category.

```
reputation |>
  group_by(name) |>
  summarize(mean = mean(score, na.rm = TRUE),
            median = median(score, na.rm = TRUE),
            sd = sd(score, na.rm = TRUE))
```

```
## # A tibble: 7 x 4
##   name         mean median    sd
##   <chr>       <dbl>  <dbl> <dbl>
## 1 CITIZENSHIP  71.2   72.2  5.25
## 2 CULTURE      74.4   76    5.96
## 3 ETHICS       74.0   75.4  6.05
## 4 GROWTH       76.8   77.8  5.69
## 5 P&S          76.3   78    5.97
## 6 TRUST        74.1   75.8  5.96
## 7 VISION       77.0   78.4  5.02
```

In today's exploration, I am interested in the possible differences in the survey scores across the industries.

```
reputation |>
  group_by(industry) |>
  summarize(mean = mean(score, na.rm = TRUE),
            median = median(score, na.rm = TRUE),
            sd = sd(score, na.rm = TRUE))
```

```
## # A tibble: 19 x 4
##    industry            mean median    sd
##    <chr>              <dbl>  <dbl> <dbl>
##  1 Airline             69.3   70.1  5.10
##  2 Automotive          78.1   78.3  2.95
##  3 Consumer Goods      77.3   77.8  2.53
##  4 Ecommerce           70.1   69.4  7.17
##  5 Energy              70.3   71.1  4.26
##  6 Financial Services  74.3   74.8  4.70
##  7 Food & Beverage     75.7   75.4  4.70
##  8 Food Delivery       72.7   73    2.44
##  9 Groceries           71.6   79.1 12.6
## 10 Healthcare          79.1   79.4  2.16
## 11 Industrial          78.4   78.4  3.13
## 12 Insurance           75.3   75.8  2.59
## 13 Logistics           79.1   79.4  2.28
## 14 Media               67.8   67.5  5.91
## 15 Other               77.0   77.8  4.31
## 16 Pharma              74.6   74.6  3.48
## 17 Retail              75.8   76.5  5.24
## 18 Tech                75.0   76.9  7.11
## 19 Telecom             73.7   73.1  3.82
```

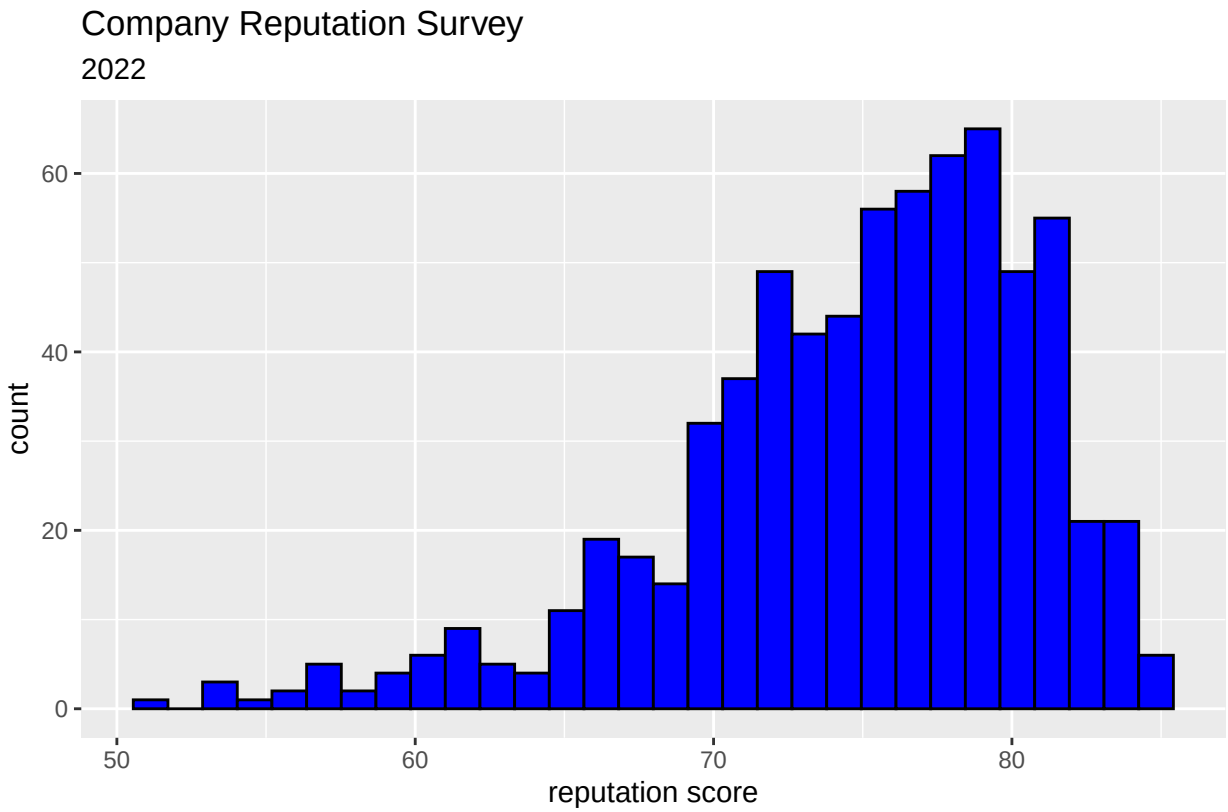
Data Visualization

Histogram

A histogram allows us to visualize the distribution of one numerical variable.

```
reputation |>
  ggplot(aes(x = score)) +
  geom_histogram(color = "black", fill = "blue") +
  labs(title = "Company Reputation Survey",
       subtitle = "2022",
       caption = "Source: Axios and Harris, TidyTuesday",
       x = "reputation score")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



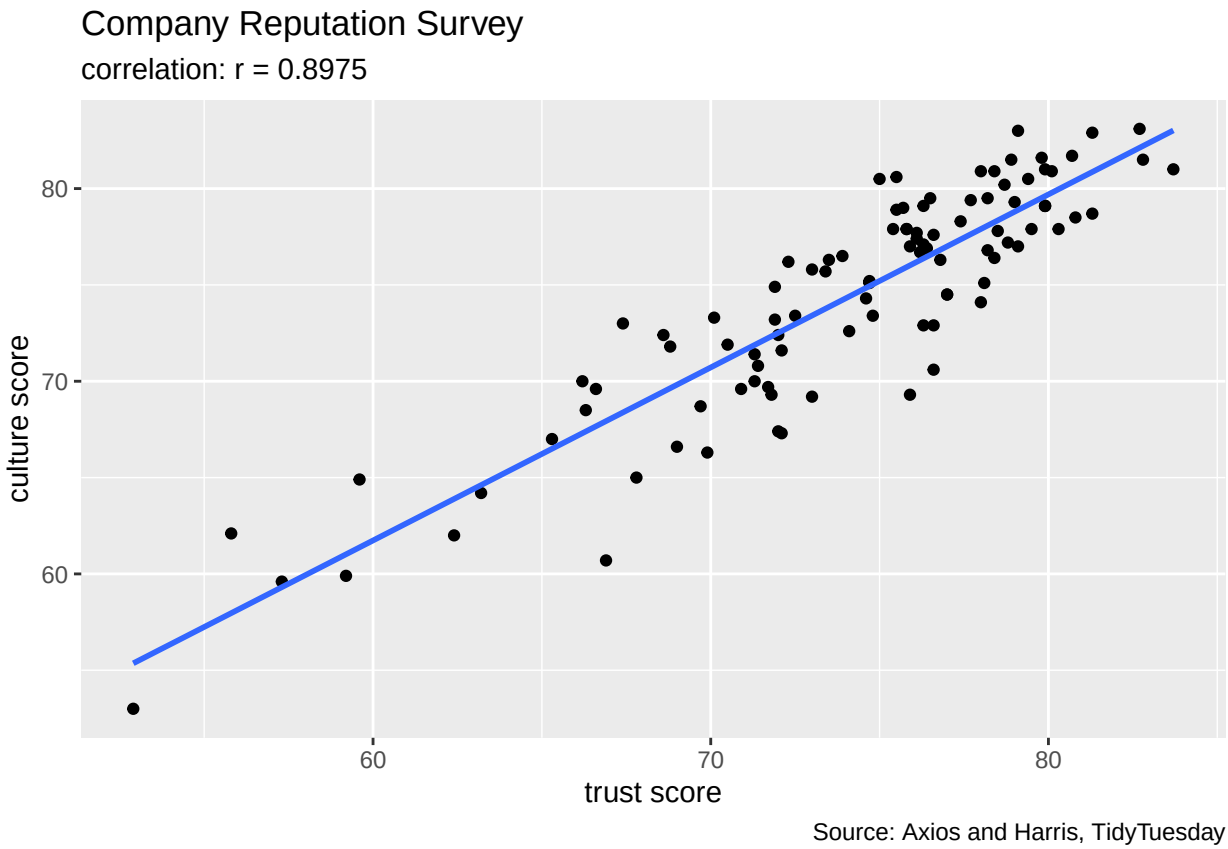
Scatterplot

A scatterplot allows us to visualize a pair of numerical variables.

```
correlation_value <- reputation_wide |>
  summarize(r = cor(trust, culture,
                    use = "pairwise.complete.obs")) |>
  unlist()

reputation_wide |>
  ggplot(aes(x = trust, y = culture)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Company Reputation Survey",
       subtitle = paste0("correlation: r = ",
                         round(correlation_value, 4)),
       caption = "Source: Axios and Harris, TidyTuesday",
       x = "trust score",
       y = "culture score")
```

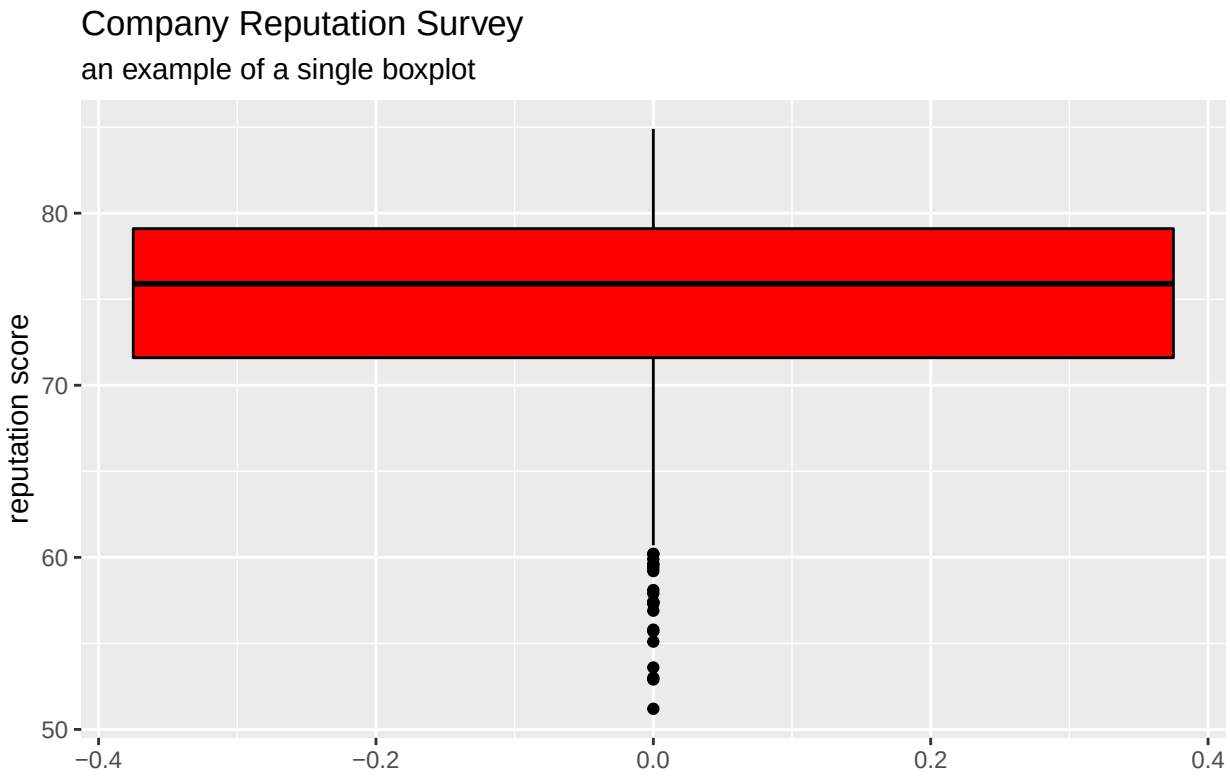
```
## `geom_smooth()` using formula 'y ~ x'
```



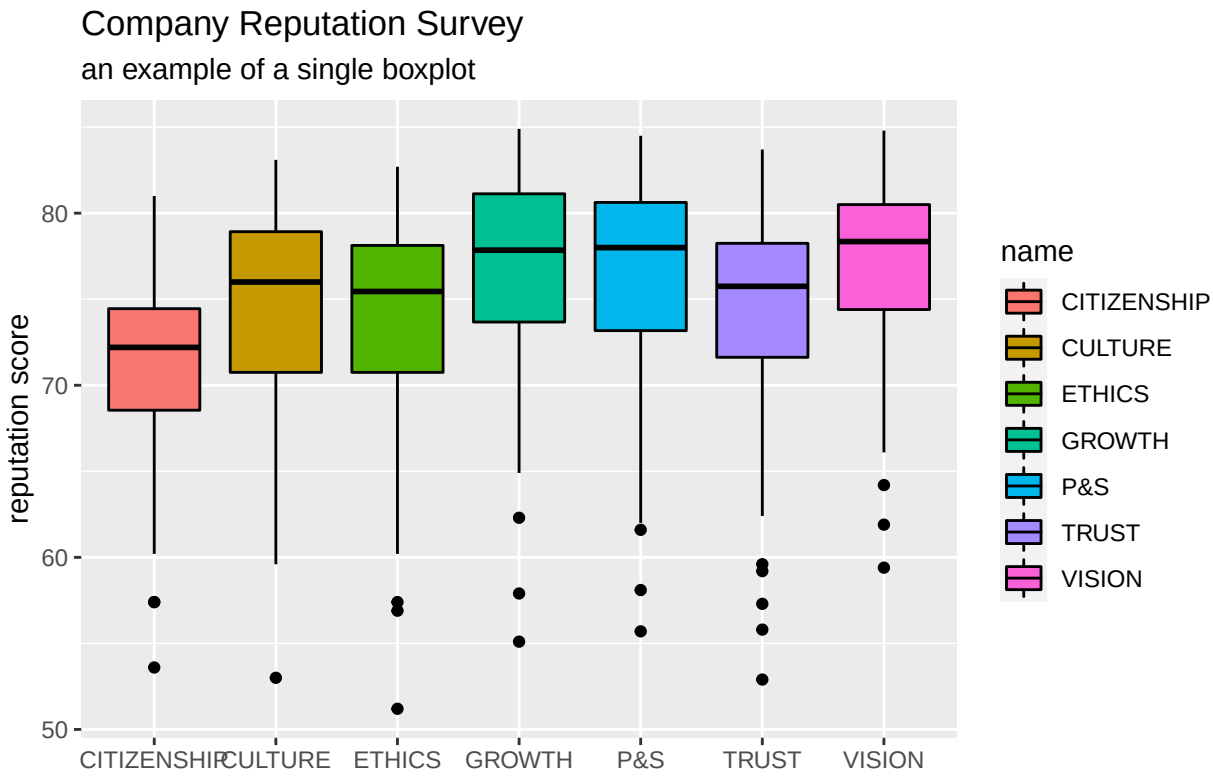
Boxplot

A boxplot allows us to visualize numerical distributions across a categorical variable.

```
reputation |>
  ggplot(aes(y = score)) +
  geom_boxplot(color = "black", fill = "red") +
  labs(title = "Company Reputation Survey",
       subtitle = "an example of a single boxplot",
       caption = "Source: Axios and Harris, TidyTuesday",
       x = "",
       y = "reputation score")
```

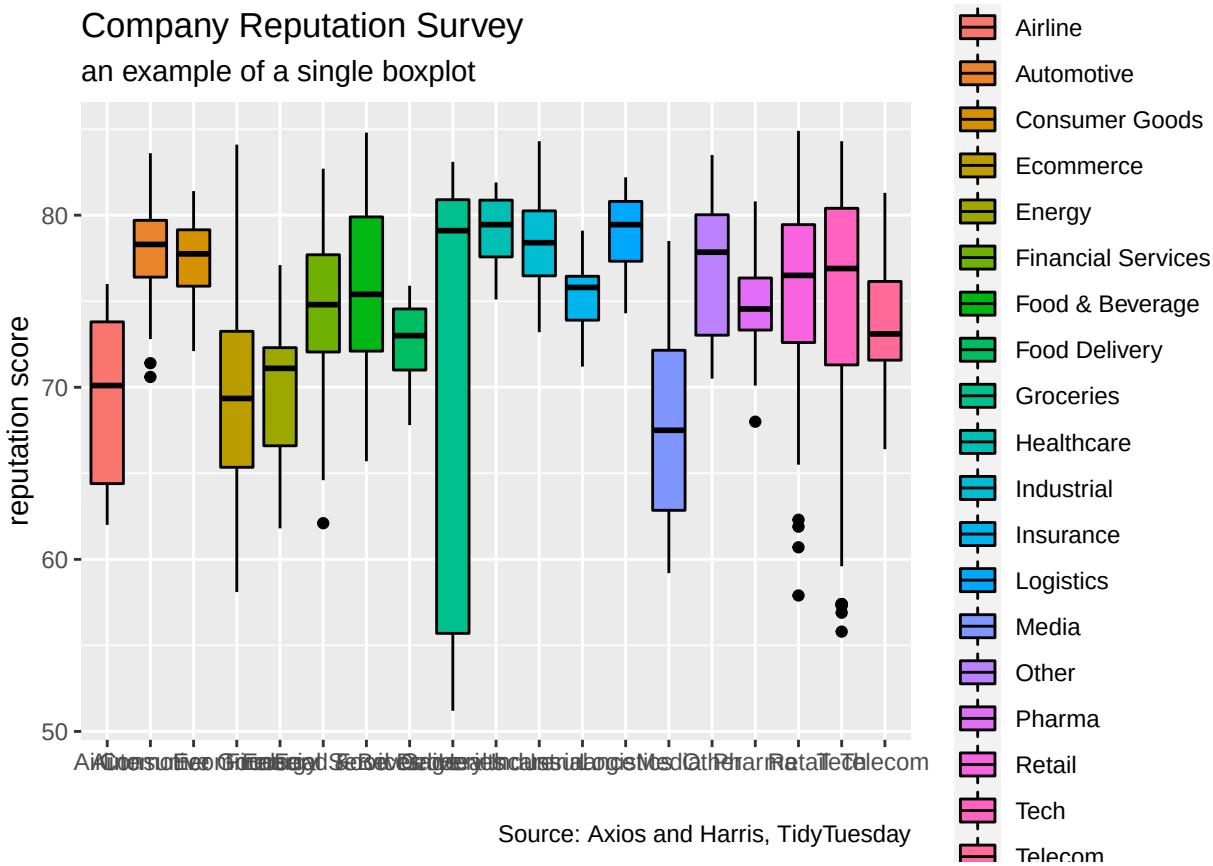



```
reputation |>
  ggplot(aes(x = name, y = score,
             fill = name, group = name)) +
  geom_boxplot(color = "black") +
  labs(title = "Company Reputation Survey",
       subtitle = "an example of a single boxplot",
       caption = "Source: Axios and Harris, TidyTuesday",
       x = "",
       y = "reputation score")
```



Let us see what happens if we explore the `industry` categorical variable.

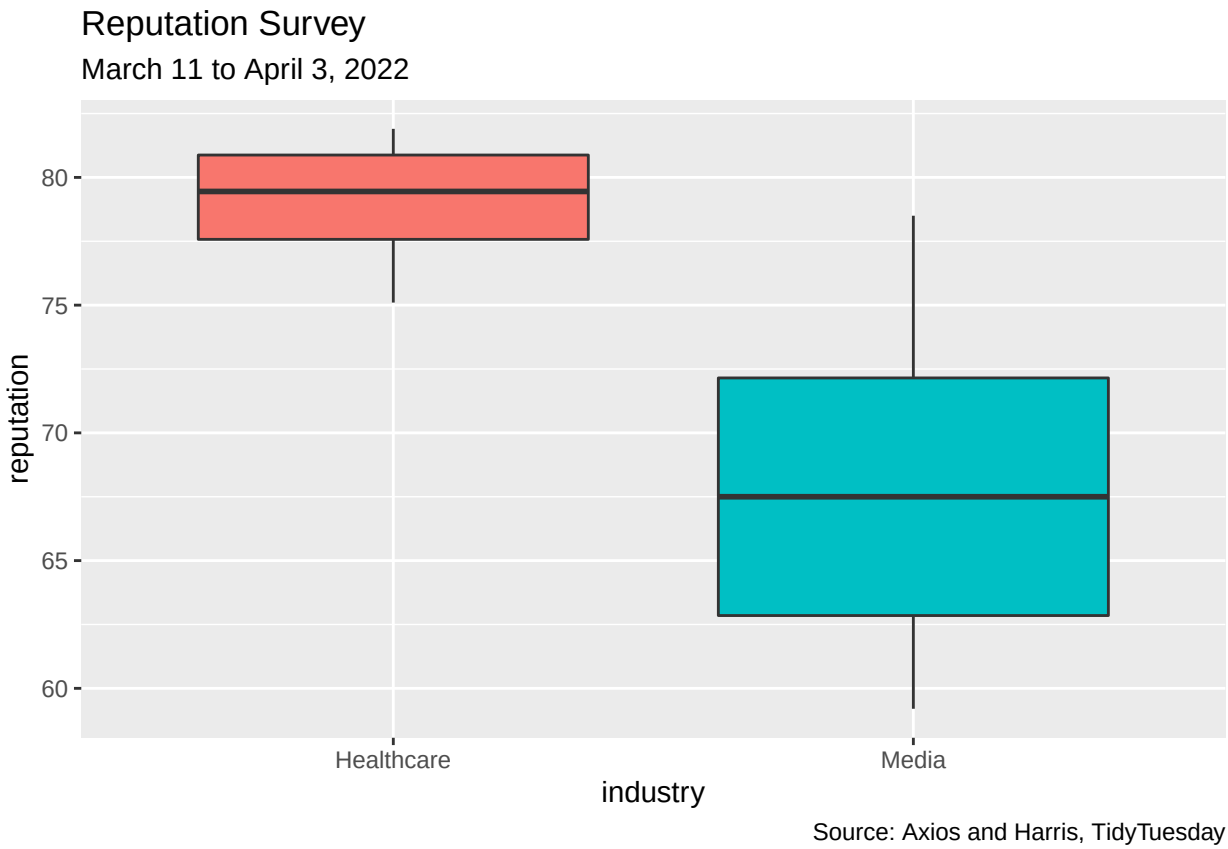
```
reputation |>
  ggplot(aes(x = industry, y = score,
             fill = industry, group = industry)) +
  geom_boxplot(color = "black") +
  labs(title = "Company Reputation Survey",
       subtitle = "an example of a single boxplot",
       caption = "Source: Axios and Harris, TidyTuesday",
       x = "",
       y = "reputation score")
```



To start to ease the complexity, we can (for example)

- focus on two of the categories
- remove the legend

```
reputation |>
  filter(industry %in% c("Healthcare", "Media")) |>
  ggplot(aes(x = industry, y = score,
             fill = industry)) +
  geom_boxplot() +
  labs(title = "Reputation Survey",
       subtitle = "March 11 to April 3, 2022",
       caption = "Source: Axios and Harris, TidyTuesday",
       x = "industry",
       y = "reputation") +
  theme(legend.position = "none")
```


Reputation Survey
March 11 to April 3, 2022
reputation
80
75
70
65
60
Healthcare
Media
industry
Source: Axios and Harris, TidyTuesday