Lab 3: Descriptive Statistics

Jonathan Manzano

2024-02-14

Libraries

```
library(tidyverse)
library(Metrics)
```

A. Weather Forecast Data

1. For this lab, we'll be using data on weather forecasts gathered by student at Saint Louis University. You can read about the dataset here. Download the weather forecasts data using the following code:

```
weather_forecasts <- readr::read_csv(paste("https://raw.githubusercontent.",
                                           "com/rfordatascience/tidytuesday",
                                           "/master/data/2022/2022-12-20",
                                           "/weather_forecasts.csv", sep=""))
```

2. How many rows are in this dataset? How many columns?

```
cat("The number of rows are ", nrow(weather_forecasts))
```

```
The number of rows are  651968
```

```
cat("\nThe number of cols are ", ncol(weather_forecasts))
```

```
The number of cols are  10
```

3. How many cities are represented in this dataset?

```
weather_forecasts$city |>
  na.omit() |>
  unique() |>
  length()
```

```
[1] 160
```

There are 160 cities represented in this dataset. First I selected the cities column from the data set. I clean the column by pipping to `na.omit()` to remove any `NA` values. Next, the output of `na.omit()` is piped to `unique()` and returns a vector of unique cities. Last, the output of `unique()` is piped to `length()` function to output the the length of the vector.

4. Create a new data frame containing only the forecasts for San Jose. You may have to explore the values for the `city` variable.

```
forecasts_san_jose <- dplyr::filter(weather_forecasts, city=="SAN_JOSE")
```

To create a new dataframe specifically for the city of San Jose, the `weather_forcasts` dataframe and `city=="SAN_JOSE"` is passed to `dplyr::filter()` which will return a new dataframe. The returned dataframe is saved in `forcasts_san_jose`.

5. Compute the mean absolute error between `observed_temp` and `forecast_temp` for San Jose.

```
absolute_error <- abs(forecasts_san_jose$observed_temp
                                - forecasts_san_jose$forecast_temp)

mae <- mean(absolute_error, na.rm=TRUE)

mae
```

```
[1] 2.169762
```

The mean absolute error (MAE) is 2.171611. Compute the `absolute_error` by passing the the difference between the `observed_temp` column and `forecast_temp`. Compute the mean by passing `absolute_error` and setting the `na.rm` flag to `TRUE` to the `mean` function.

6. Compute the mean absolute error between `observed_temp` and `forecast_temp` for San Jose using only forecasts made 48 hours in advance.

```
forecasts_san_jose_48h <- subset(forecasts_san_jose,
                                 forecast_hours_before == 48)

absolute_error_48h <- abs(forecasts_san_jose_48h$observed_temp
                                - forecasts_san_jose_48h$forecast_temp)

mae_48h <- mean(absolute_error_48h, na.rm=TRUE)

mae_48h
```

```
[1] 2.262544
```

The mean absolute error (MAE) is 2.262544. Create a new dataframe called `forecasts_san_jose_48h` by passing `forecasts_san_jose` dataframe and passing the argument `forecast_hours_before == 48`. Compute the `absolute_error_48h` by passing the the difference between the `observed_temp` column and `forecast_temp`. Compute the mean by passing `absolute_error` and setting the `na.rm` flag to `TRUE` to the `mean` function.

7. Compute the mean absolute error between `observed_temp` and `forecast_temp` for San Jose using only forecasts made 12 hours in advance.

```
forecasts_san_jose_12h <- subset(forecasts_san_jose,
                                 forecast_hours_before == 12)

absolute_error_12h <- abs(forecasts_san_jose_12h$observed_temp
                          - forecasts_san_jose_12h$forecast_temp)

mae_12h <- mean(absolute_error_12h, na.rm=TRUE)

mae_12h
```

```
[1] 2.0553
```

The MAE is 2.0553

8. Compare your answers to 6 and 7. What do you notice? How does this compare to your expectation?

```
abs(mae_48h - mae_12h)
```

```
[1] 0.2072442
```

The statistical mean between questions 6 and 7 have a difference of 0.2072442. Question 7 has the lower MAE than question 6 by minuscule value. It seems that the forecasts predicted hours closer to the present have a lower MAE.

9. Pick two cities in this dataset. Investigate whether the forecast accuracy is better for one city than for the other, using an appropriate statistic. Discuss your findings.

```
forecasts_seattle <- dplyr::filter(weather_forecasts, city=="SEATTLE")

absolute_error_seattle <- abs(forecasts_seattle$observed_temp
                              - forecasts_seattle$forecast_temp)

mae_seattle <- mean(absolute_error, na.rm=TRUE)

mae_seattle
```

```
[1] 2.169762
```

```r
forecasts_seattle_48h <- subset(forecasts_seattle,
                                forecast_hours_before == 48)

absolute_error_seattle_48h <- abs(forecasts_seattle_48h$observed_temp
                                - forecasts_seattle_48h$forecast_temp)

mae_seattle_48h <- mean(absolute_error_seattle_48h, na.rm=TRUE)

mae_seattle_48h
```

```
[1] 1.903376
```

```r
forecasts_seattle_12h <- subset(forecasts_seattle,
                                forecast_hours_before == 12)

absolute_error_seattle_12h <- abs(forecasts_seattle_12h$observed_temp
                                - forecasts_seattle_12h$forecast_temp)

mae_seattle_12h <- mean(absolute_error_seattle_12h, na.rm=TRUE)

mae_seattle_12h
```

```
[1] 1.622989
```

```r
forecasts_miami_beach <- dplyr::filter(weather_forecasts, city=="MIAMI_BEACH")

absolute_error_miami_beach <- abs(forecasts_miami_beach$observed_temp
                                - forecasts_miami_beach$forecast_temp)

mae_miami_beach <- mean(absolute_error_miami_beach, na.rm=TRUE)

mae_miami_beach
```

```
[1] 1.689895
```

```r
forecasts_miami_beach_48h <- subset(forecasts_miami_beach,
                                forecast_hours_before == 48)

absolute_error_miami_beach_48h <- abs(forecasts_miami_beach_48h$observed_temp
                                - forecasts_miami_beach_48h$forecast_temp)
```

```
mae_miami_beach_48h <- mean(absolute_error_miami_beach_48h, na.rm=TRUE)

mae_miami_beach_48h
```

```
[1] 1.779463
```

```
forecasts_miami_beach_12h <- subset(forecasts_miami_beach,
                                    forecast_hours_before == 12)

absolute_error_miami_beach_12h <- abs(forecasts_miami_beach_12h$observed_temp
                                    - forecasts_miami_beach_12h$forecast_temp)

mae_miami_beach_12h <- mean(absolute_error_miami_beach_12h, na.rm=TRUE)

mae_miami_beach_12h
```

```
[1] 1.579008
```

I computed the MAE when the forecast is predicted 12 hours before for both Seattle, WA and Miami Beach, FL. The MAE forecast 12 hours before for Seattle is 1.622989 and for Miami Beach is 1.579008. Since Miami Beach's MAE is smaller comparable to Seattle, we can infer that the forecast for Miami Beach 12 hours before is more accurate that Seattle.

B. Find your own data

For this component, pick a Tidy Tuesday dataset and complete the following activity.

10. Provide a brief description of your dataset. Identify at least two questions you could try to answer using this dataset.

```
age_gaps <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/ma
```

```r
head(age_gaps)
```

```
# A tibble: 6 x 13
  movie_name     release_year director age_difference couple_number actor_1_name
  <chr>                 <dbl> <chr>             <dbl>         <dbl> <chr>
1 Harold and Ma~         1971 Hal Ash~             52             1 Ruth Gordon
2 Venus                  2006 Roger M~             50             1 Peter O'Too~
3 The Quiet Ame~         2002 Phillip~             49             1 Michael Cai~
4 The Big Lebow~         1998 Joel Co~             45             1 David Huddl~
5 Beginners              2010 Mike Mi~             43             1 Christopher~
6 Poison Ivy             1992 Katt Sh~             42             1 Tom Skerritt
# i 7 more variables: actor_2_name <chr>, character_1_gender <chr>,
#   character_2_gender <chr>, actor_1_birthdate <date>,
#   actor_2_birthdate <date>, actor_1_age <dbl>, actor_2_age <dbl>
```

The Hollywood Age Gap data set comprises of Hollywood Love Movies' love interest of two actors and the age gap between them. Each record has:

- movie_name
- release_year
- director
- age_difference
- couple_number
- actor_1_name
- actor_2_name
- character_1_gender
- character_2_gender
- actor_1_birthdate
- actor_2_birthdate
- actor_1_age
- actor_2_age

(1) On average, between same gender and opposite gender love interests, which has the bigger age gap?
(2) What is the standard deviation of the age gaps?

11. Open your dataset in R and compute one or more descriptive statistics that shed light on your questions. Discuss your findings.

```
same_gender <- filter(age_gaps, character_1_gender == character_2_gender)
mean(same_gender$age_difference, na.rm = TRUE)
```

```
[1] 13.30435
```

```
different_gender <- filter(age_gaps, character_1_gender != character_2_gender)
mean(different_gender$age_difference, na.rm = TRUE)
```

```
[1] 10.36572
```

The mean of the same gender character love interest is 13.30435. The mean of the different gender is 10.36572. The average age gap is bigger for same gender love interest versus different gender love interests.

12. Are there any limitations of your analysis? Could additional data or more complicated methods improve your analysis? Discuss.

```
nrow(same_gender)
```

```
[1] 23
```

```
nrow(different_gender)
```

```
[1] 1132
```

A limitation would be the about of records being used to generate the mean and compare. There are 23 movies that have characters with the same gender while there are 1132 movies that have love interest characters of different genders. More movies with love interest characters of the same gender would give more significance to the statistics.