Massachusetts Bay Transportation Authority Exercise

Session 3

The Massachusetts Bay Transportation Authority ("MBTA") manages America's oldest subway, as well as Greater Boston's commuter rail, ferry, and bus systems. It's your first day on the job as the data analyst and you've been tasked with analyzing average ridership through time. The dataset is stored in this classes data subfolder called `mbta.xlsx`.

Create a .R script titled `session3_mbta.R` to complete the following data cleaning tasks. Be sure to use proper code styling and commenting throughout your script.

Import the data

1. What spreadsheets exist in the workbook?
2. Import mbta.xlsx[1]

[1] You may need to skip a row or two.

Examing the data

The first step when cleaning a dataset is to explore it a bit. Pay particular attention to how the rows and columns are organized and to the locations of missing values.

1. View the structure of mbta.
2. View the first 6 rows of mbta.
3. View a summary of mbta.

Removing unnecessary rows and columns

It appears that the data are organized with observations stored as columns rather than as rows. You can fix that.

First, though, you can address the missing data. All of the `NA` values are stored in the `All Modes by Qtr` row. This row really belongs in a different data frame; it is a quarterly average of weekday MBTA ridership. Since this dataset tracks monthly average ridership, you'll remove that row. Similarly, the 7th row (Pct Chg / Yr) and the 11th row (TOTAL) are not really observations as much as they are analysis. Go ahead and remove the 7th and 11th rows as well. The first column also needs to be removed because it's just listing the row numbers.

1. Remove the first, seventh, and eleventh rows of mbta.
2. Remove the first column.

Observations are stored in columns

In this data, variables are stored in rows instead of columns. The different modes of transportation (commuter rail, bus, subway, ferry, ...) are variables, providing information about each month's average ridership. The months themselves are observations[2]. You can tell which is which because as you go through time, the month changes, but the modes of transport offered by the T do not.

As is customary, you want to represent variables in columns rather than rows.

[2] Currently, months are listed as variable names; rather, they should be in their own column.

1. Load the `tidyr` package.
2. Gather the columns of the mbta data. Call your new columns `month` and `thou_riders` (for "thousand riders").[3]
3. View the head of this new data set.

[3] Use the `-` operator to omit the `mode` column from your gathering.

Type conversions

In a minute, you'll put variables where they belong (as column names). But first, take this opportunity to change the average weekday ridership column, thou_riders, into numeric values rather than character strings. That way, you'll be able to do things like compare values and do math.

- Coerce the ridership column, `mbta$thou_riders`, into a numeric data type.

Variables are stored in both rows and columns

Now, you can finish the job you started earlier: getting variables into columns. Right now, variables are stored as "keys" in the `mode` column. You'll use the `tidyr` function `spread()` to make them into columns containing average weekday ridership for the given month and mode of transport.

1. Use `spread()` to change values in the `mode` column of mbta into column names. The columns should contain the average weekday ridership values associated with that mode of transport.
2. View the head of this new mbta data set.

Separating columns

Your dataset is already looking much better! Your boss saw what a great job you're doing and now wants you to do an analysis of the T's ridership during certain months across all years.

Your dataset has months in it, so that analysis will be a piece of cake. There's only one small problem: if you want to look at ridership on the T during every January (for example), the month and year are together in the same column, which makes it a little tricky. You'll need to separate the `month` column into distinct `month` and `year` columns to make life easier.

1. Split the month column of mbta at the dash and create a new `month` column with only the month and a `year` column with only the year.
2. View the head of this new mbta data set.

Do your values seem reasonable?

Before you write up the analysis for your boss, it's a good idea to screen the data for any obvious mistakes and/or outliers.

1. View a `summary()` of the data. Pay particular attention to the `Boat` column stats.
2. Generate a histogram of the Commuter Boat ridership.[4]
3. Use a boxplot to identify the month and year of this outlier.

[4] This is the `Boat` variable.

Dealing with entry error

Every month, average weekday commuter boat ridership was on either side of 4,000. Then, one month it jumped to 40,000 without warning? Unless the Olympics were happening in Boston that month (they weren't), this value is certainly an error. You can assume that whoever was entering the data that month accidentally typed 40 instead of 4.

1. Locate the row and column of the incorrect value.
2. Replace the incorrect value with 4.
3. Now view the summary, histogram, and boxplots of the `Boat` variable.

Congratulations!

Congrats, your data is now clean and ready for analysis. When you are done, submit it to me in a direct message via Slack.