Understanding Data Structure and Column Contents

using R

Dr. Austin Brown

Kennesaw State University

Understanding Column Contents

- **Objective**: Learn to differentiate between numeric and character data, and identify missing data.
- **Importance**: Essential for correct data processing and analysis.
- **Key Points**
  - **str()**: Displays structure of the dataframe, including data types.
  - **summary()**: Provides a summary of each column, helping to identify missing values.
  - **Example**: Display metadata about a dataset and its missing values.

Data Structure

- In any given dataset, the way the data are arranged is paramount for understanding what the data are and what information they contain.
- Generally, we want the way data are recorded within a dataset to be “tidy”.
- Tidy data is data that is well-organized and easy to work with. It has a specific structure:
  - Each variable is a column.
  - Each observation is a row.
  - Each cell has a single value.

Data Structure

- For instance, it the `cars` dataset below, the data are arranged in a tidy way:

	mpg	cyl	disp	hp	drat
Mazda RX4	21.0	6	160	110	3.90
Mazda RX4 Wag	21.0	6	160	110	3.90
Datsun 710	22.8	4	108	93	3.85
Hornet 4 Drive	21.4	6	258	110	3.08
Hornet Sportabout	18.7	8	360	175	3.15
Valiant	18.1	6	225	105	2.76

Data Structure

- Why are they considered tidy? Consider our above criteria:
  - Each variable is a column: `mpg`, `cyl`, `disp`, `hp`, `drat`.
  - Each observation is a row: Each row represents a different car.
  - Each cell has a single value: Each cell contains a single value for the variable it represents.

Data Types: Numeric vs. Character Data

- Now that we know how data should be structured, let's talk about the types of data we might encounter.
- In R, there are two main types of variables:
  - **Numeric or Quantitative**: These are variables which are naturally measured using numbers. Variables like age, height, weight, etc. are examples of numeric variables.
  - **Character or Qualitative**: These are variables which are naturally measured using non-quantitative qualities. Variables like hair color, favorite food, etc. are examples of character variables.

Data Types: Numeric vs. Character Data

- Note, when we read data into R, based on the values contained in each column, R will automatically assign a data type to each column.
- Thus, it is important for us to double-check to ensure that the data types as we understand them are also the way R has interpreted them.

Data Types: Numeric vs. Character Data

- For example, suppose we want to use the Cars.csv dataset. We have already learned how to read that data in. But how do we check to see what data types R has assigned to each column?
- To do this, we can use the `str()` function in R.
- The `str()` function provides a compact display of the internal structure of an R object.
    - For dataframes, it shows the data type of each column and the first few values in that column.

Data Types: Numeric vs. Character Data

```r
## Read in the Data ##
cars <- read.csv("Cars.csv")
## Examine the Data Structure ##
str(cars)
```

```
'data.frame':   32 obs. of  12 variables:
 $ X   : chr  "Mazda RX4" "Mazda RX4 Wag" "Datsun 710" "Hornet 4 Drive" ...
 $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
 $ cyl : int  6 6 4 6 8 6 8 4 4 6 ...
 $ disp: num  160 160 108 258 360 ...
 $ hp  : int  110 110 93 110 175 105 245 62 95 123 ...
 $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
 $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
 $ qsec: num  16.5 17 18.6 19.4 17 ...
 $ vs  : int  0 0 1 1 0 1 0 1 1 1 ...
 $ am  : int  1 1 1 0 0 0 0 0 0 0 ...
 $ gear: int  4 4 4 3 3 3 3 4 4 4 ...
 $ carb: int  4 4 1 1 2 1 4 2 2 4 ...
```

Data Types: Numeric vs. Character Data

- As we can see in the output above, we get a list of all the columns in the dataset, along with the data type of each column.
- Note, variables with `num` next to their name (e.g., mpg, disp, etc.) are numeric variables, while variables with `chr` next to their name (e.g., X) are character variables.
- However, notice next to the `cyl` variable it says `int`. This is because R has interpreted the `cyl` variable as an integer, which is a type of numeric variable.
- Character variables can also be referred to as `factor` variables in R.
    - Factors generally contain ordered levels whereas character variables do not.

Data Types: Numeric vs. Character Data

- Now, let's read in the NYC Airplanes 2013 Excel dataset and perform the same operation.

```r
## Load readxl package ##
library(readxl)
## Read in Airplanes Data ##
planes <- read_xlsx("NYC Airplanes 2013.xlsx")
## Examine the Data Structure ##
str(planes)
```

```
tibble [3,322 x 9] (S3: tbl_df/tbl/data.frame)
 $ tailnum     : chr [1:3322] "N10156" "N102UW" "N103US" "N104UW" ...
 $ year        : num [1:3322] 2004 1998 1999 1999 2002 ...
 $ type        : chr [1:3322] "Fixed wing multi engine" "Fixed wing multi engin
 $ manufacturer: chr [1:3322] "EMBRAER" "AIRBUS INDUSTRIE" "AIRBUS INDUSTRIE" "
 $ model       : chr [1:3322] "EMB-145XR" "A320-214" "A320-214" "A320-214" ...
 $ engines     : num [1:3322] 2 2 2 2 2 2 2 2 2 2 ...
 $ seats       : num [1:3322] 55 182 182 182 55 182 182 182 182 182 ...
 $ speed       : num [1:3322] NA NA NA NA NA NA NA NA NA NA ...
 $ engine      : chr [1:3322] "Turbo-fan" "Turbo-fan" "Turbo-fan" "Turbo-fan" .
```

Data Types: Numeric vs. Character Data

- As we can see from the output, we have five character columns and four numeric columns.
- We also have 3322 observations (or rows) and 9 columns (or variables)

Producing Column Summaries: Missing Data

- In addition to understanding the data types of each column, it is also important to understand the contents of each column.
- One very common issue that arises when working with data is missing data.
- Missing data can be problematic for many reasons, including:
  - It can lead to biased results.
  - It can lead to incorrect conclusions.
  - It can lead to incorrect inferences.
- Thus, it is important to identify and address missing data in our datasets before proceeding with any analysis.

Producing Column Summaries: Missing Data

- How do we do this? One way is to use the `summary()` function in R.
- The `summary()` function is like the `str()` function in some ways, but it also provides a summary of each column in the dataset, including the number of missing values in each column.
- Let's see how this works with the `cars` dataset.

Producing Column Summaries: Missing Data

```r
## Examine the Data Summary ##
summary(cars)
```

```
      X                  mpg             cyl             disp      
 Length:32          Min.   :10.40   Min.   :4.000   Min.   : 71.1  
 Class :character   1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8  
 Mode  :character   Median :19.20   Median :6.000   Median :196.3  
                    Mean   :20.09   Mean   :6.188   Mean   :230.7  
                    3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0  
                    Max.   :33.90   Max.   :8.000   Max.   :472.0  
       hp             drat             wt             qsec      
 Min.   : 52.0   Min.   :2.760   Min.   :1.513   Min.   :14.50  
 1st Qu.: 96.5   1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89  
 Median :123.0   Median :3.695   Median :3.325   Median :17.71  
 Mean   :146.7   Mean   :3.597   Mean   :3.217   Mean   :17.85  
 3rd Qu.:180.0   3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90  
 Max.   :335.0   Max.   :4.930   Max.   :5.424   Max.   :22.90  
       vs               am              gear            carb      
 Min.   :0.0000   Min.   :0.0000   Min.   :3.000   Min.   :1.000  
 1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000  
 Median :0.0000   Median :0.0000   Median :4.000   Median :2.000  
 Mean   :0.4375   Mean   :0.4062   Mean   :3.688   Mean   :2.812  
 3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000  
 Max.   :1.0000   Max.   :1.0000   Max.   :5.000   Max.   :8.000  
```

Producing Column Summaries: Missing Data

- As we can see from the output above, the `summary()` function provides a summary of each column in the dataset.
- For numeric variables, it provides the minimum, 1st quartile, median, mean, 3rd quartile, and maximum values.
- For character variables, it provides the number of unique values in the column.
- It also provides the number of missing values in each column.

Producing Column Summaries: Missing Data

- In the case of the cars data, we see that there are no missing values in any of the columns.
- However, this is not always the case. Let's see what happens when we use the `summary()` function on the `planes` dataset.

Producing Column Summaries: Missing Data

```r
## Examine the Data Summary ##
summary(planes)
```

```
   tailnum               year          type           manufacturer
 Length:3322        Min.   :1956   Length:3322        Length:3322
 Class :character   1st Qu.:1997   Class :character   Class :character
 Mode  :character   Median :2001   Mode  :character   Mode  :character
                    Mean   :2000
                    3rd Qu.:2005
                    Max.   :2013
                    NA's   :70
    model              engines          seats           speed
 Length:3322        Min.   :1.000   Min.   :  2.0   Min.   : 90.0
 Class :character   1st Qu.:2.000   1st Qu.:140.0   1st Qu.:107.5
 Mode  :character   Median :2.000   Median :149.0   Median :162.0
                    Mean   :1.995   Mean   :154.3   Mean   :236.8
                    3rd Qu.:2.000   3rd Qu.:182.0   3rd Qu.:432.0
                    Max.   :4.000   Max.   :450.0   Max.   :432.0
                                                    NA's   :3299
    engine
 Length:3322
 Class :character
 Mode  :character
```

Producing Column Summaries: Missing Data

- Here we can see that two of our variables, `year` and `speed` contain missing values.
- The `year` variable has 70 missing values, while the `speed` variable has 3299 missing values.
- This is important information to know, as it will help us to decide how to proceed with our analysis.