Class 5 Data Wrangling with R (Part II)

Dr Wei Miao

UCL School of Management

Thu, Oct 20 2022

Section 1

Data Wrangling

More Convenient Package Management Using `pacman`

```
1   pacman::p_load(dplyr,ggplot2)
```

- Please install `pacman` on your RStudio
- `pacman`'s functionality
  - Load all packages stated in the parantheses, seperated by commas
  - If the package is not downloaded yet, download it, and then load it
- R tip: if you want to use a function without loading the whole package, you can use two colons to call the function: `pacakge::function`

Recap: filter()

- `filter(dataset, criteria)`: pick observations by their values

- `arrange(dataset,variable)`: reorder the rows
- `mutate(dataset, newvariable = )`: create new variables with functions of existing variables

- pipe operator: **%>%**

Select Variables: `select`

- select() can select variables into a smaller dataset.

```
1  # Select two columns: hp and cyl
2  mtcars%>%
3    select(hp, cyl) %>%
4    head()
```

	hp	cyl
Mazda RX4	110	6
Mazda RX4 Wag	110	6
Datsun 710	93	4
Hornet 4 Drive	110	6
Hornet Sportabout	175	8
Valiant	105	6

Aggregation by Groups: `group_by`

- group_by() allows us to aggregate data by group and compute statistics for each group

```
1  # group by cyl
2  mtcars %>%
3    group_by(cyl)
```

- Although nothing seemingly happens to the dataset, internally, the dataset is already grouped based on the specified variable(s).

Aggregation by Groups: `group_by() + summarise()`

- summarise() creates a new data frame after aggregating data. The final dataset
    - has one row for each pair of grouping variables (for each `cyl` value)
    - contains one column for each grouping variable (`cyl`)
    - contains one column for each new summarised variable (`avg_mp`)

```
1  # compute the average mpg for each cyl group
2  mtcars %>%
3    group_by(cyl) %>% # group by cyl
4    summarise(avg_mp = mean(mpg)) %>% # compute the average mpg
5    ungroup()
```

cyl	avg_mp
4	26.66364
6	19.74286
8	15.10000

Aggregation by Groups: `group_by()` + `summarise()`

- We can have multiple group variables for `group_by`

```
1  # compute the average mpg for each cyl,vs group
2  mtcars %>%
3    group_by(cyl,vs) %>% # group by cyl
4    summarise(avg_mp = mean(mpg)) %>% # compute the average mpg
5    ungroup()
```

Aggregation by Groups: `group_by()` + `mutate()`

- Try the following code by replacing `summarise()` with `mutate()`, what do you get now?

```
1   # compute the average mpg for each cyl,vs group
2   mtcars %>%
3     group_by(cyl,vs) %>% # group by cyl
4     mutate(avg_mp = mean(mpg)) %>% # compute the average mpg
5     ungroup()
```

- A new column is added to the original dataset, the value of which is from the group-by aggregation.

Data Wrangling
○○○○○○○○●○○○

Data Cleaning
○○○○

Descriptive Analytics
○○○○○

Preliminary Customer Analysis
○○

Consolidate Multiple Data Frames

- When consolidating multiple data frames, we usually have 4 types of joining methods

left_join

- `left_join` keeps everything from the left data frame and matches as much as it can from the right data frame.
  - All IDs **in the left data frame** will be retained
  - If a match can be found, value from the right data frame will be filled in
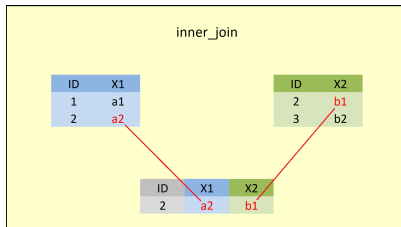 - If a match cannot be found, a missing value will be filled in

```
1  # Method 1 without pipe operator
2  left_join(df_left, df_right, by = 'ID')
3  # Method 2 with pipe operator
4  df_left %>%
5    left_join(df_right, by = 'ID')
```
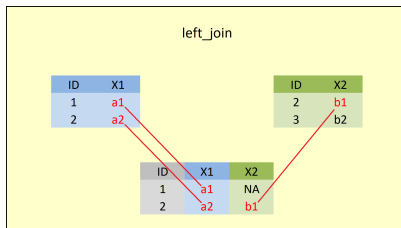
inner_join

- inner_join only keeps the observations that appear in both data frames
  - Only common IDs **in both data frames** will be retained
  - If a match can be found, values will be filled in from both data frames
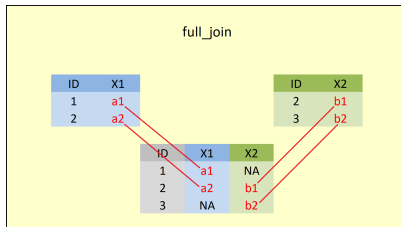
```
1  # Method 1 without pipe operator
2  inner_join(df_left, df_right, by = 'ID')
3  # Method 2 with pipe operator
4  df_left %>%
5    inner_join(df_right, by = 'ID')
6  # Method 3: order of data frames should not matter. Why?
7  df_right %>%
8    inner_join(df_left, by = 'ID')
```

full_join

- `full_join` keeps all observations from both data frames
    - All IDs **in either data frames** will be retained
    - If a match can be found, values will be filled in from both data frames

```
1  # Method 1 without pipe operator
2  full_join(df_left, df_right, by = 'ID')
3  # Method 2 with pipe operator
4  df_left %>%
5     full_join(df_right, by = 'ID')
6  # Method 3: order of data frames should not matter. Why?
7  df_right %>%
8     full_join(df_left, by = 'ID')
```

Data Wrangling
○○○○○○○○○○○○○

Data Cleaning
●○○○

Descriptive Analytics
○○○○○

Preliminary Customer Analysis
○○

Section 2

Data Cleaning

Data Wrangling
○○○○○○○○○○○○○○

Data Cleaning
○●○○

Descriptive Analytics
○○○○○

Preliminary Customer Analysis
○○

Variable Types

- Non-metric
  - Categorical (gender, region, brand, religion)
  - Ordinal (Business Week rankings, NCAA rankings)
- Metric
  - Continuous (age, height, sales, rainfall)
- Different types of variables are handled in different ways in statistics
  - Can talk about an average age, but not an average color
  - Some statistical techniques only work with one type of variable
- We need to make sure the variables are of the correct data types. Or we may need to convert them to the correct types.
  - e.g., from character to date time using `lubridate` package

Data Wrangling
000000000000

Data Cleaning
000●0

Descriptive Analytics
00000

Preliminary Customer Analysis
00

Missing Values

- In R, missing values are represented by the symbol `NA` (i.e., *not available*).

- Most statistical models cannot handle missing values, so we need to deal with them in R.

  - Few missing values: remove them from analysis.

  - Many missing values: need to replace them with appropriate values: mean/median/imputation

Data Wrangling
○○○○○○○○○○○○○

Data Cleaning
○○○●

Descriptive Analytics
○○○○○

Preliminary Customer Analysis
○○

Outliers

- In statistics, an outlier is a data point that differs significantly from other observations.

  - Few outliers: remove them from analysis

  - Many outliers: winsorize data

  - **If the distribution of a variable is not normal distribution, we often log transform variables to mitigate outlier issues**

Section 3

Descriptive Analytics

Two Major Tasks of Descriptive Analytics

1. **Describe data depending on your business purposes**
   - "How much do our customers spend each month on average?"
   - "What percentage of our customers are unprofitable?"
   - "What is the difference between the retention rates of men and women?"

2. **Make statistical inferences from data**
   - "Based on our sample, does the difference between the spendings of men and women indicate that men and women respond differently in the customer base at large?"
   - "Based on our sample, can we conclude that customers who sign up for online banking are more profitable than customers who do not?"
   - "Based on our test mailing, can we conclude that ad-copy A works better than ad-copy B?"

- You can think of descriptive analytics as **creating a dashboard** to display the key information you would like to know for your business.

Summary Statistics

- In descriptive analytics, **summary statistics** are used to summarize a set of observations, in order to communicate the largest amount of information as simply as possible.

- There are two main types of summary statistics used in evaluation: **measures of central tendency** and **measures of dispersion**.

  - Measures of central tendency provide different versions of the average, including the mean, the median, 25 percentile, 75 percentile, the mode, etc.

  - Measures of dispersion provide information about how much variation there is in the data, including the range and the standard deviation.

- It's good to include summary statistics table in your dissertation before any statistical analysis!

  - Commonly reported summary statistics include *mean*, *standard deviation*, *number of observations*, *min*, *25 percentile*, *median*, *75 percentile*, and *max*.

  - Then describe the distribution of the variable, dispersion of the variable, etc.

Summary Statistics with R

- In R, a nice package to report summary statistics is `modelsummary`. `datasummary_skim()` is a shortcut to conduct basic summary statistics

- For more features, refer to the package tutorial here, especially `datasummary()` function.
  - `datasummary_skim()` is a special case of more general `datasummary()`, which outputs a pre-determined set of summary statistics

```
1   pacman::p_load(modelsummary)
2   mtcars %>%
3     datasummary_skim()
```

Correlation Matrix

- Correlation matrix helps us understand the co-movement of any two variables in the data
- datasummary_correlation() reports the pairwise correlation coefficient
- In general, in a statistical model, variables of high correlation should not be included together, which leads instability

```
1   mtcars %>%
2     datasummary_correlation()
```

Data Wrangling
○○○○○○○○○○○○○

Data Cleaning
○○○○

Descriptive Analytics
○○○○○

Preliminary Customer Analysis
●○

Section 4

Preliminary Customer Analysis

Data Wrangling
000000000000

Data Cleaning
0000

Descriptive Analytics
00000

Preliminary Customer Analysis
0●

Preliminary Customer Analysis

- Spend 30 min-ish to work on the case study in a group.

- There are 7 questions in total. At the end of the discussion, each group selects a group leader to answer one question.

- To show your codes, join the Zoom link under "Module Overview" on Moodle.

- The group that did not answer correctly needs to do a performance next week!