Class 4 Data Wrangling with R Part I

Dr Wei Miao

UCL School of Management

October 11, 2023

Section 1

Overview

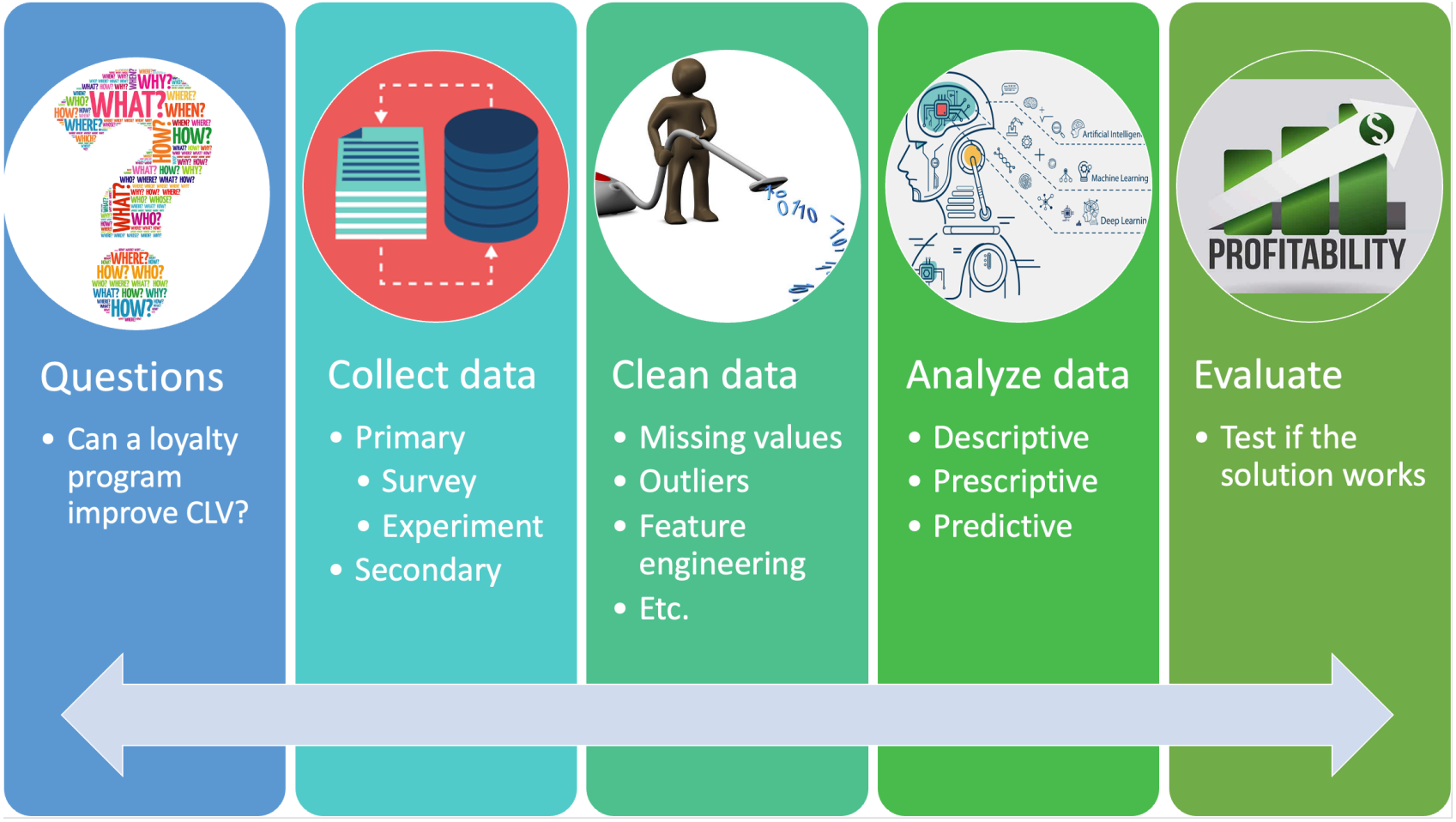
Class Objectives

- Understand the major steps to conduct data analytics
- **Data collection:** Learn how to collect first-hand data
- **Data cleaning:** Learn how to use the `dplyr` package to collect, load, and clean data
- **Data analysis:** Learn how to conduct descriptive analytics

Section 2

Data Analytics Workflow

Overview


Questions
• Can a loyalty program improve CLV?
Collect data
• Primary
• Survey
• Experiment
• Secondary
Clean data
• Missing values
• Outliers
• Feature engineering
• Etc.
Analyze data
• Descriptive
• Prescriptive
• Predictive
Evaluate
• Test if the solution works
Artificial Intelligen
Machine Learning
Deep Learnin
PROFITABILITY

Collect Data

- **Primary Data:** Data that are generated by the researcher himself/herself, surveys, interviews, experiments, specially designed for understanding and solving the research problem at hand.
- **Secondary Data:** Existing data generated by the company's or consumer's past activities, as part of organizational record keeping.

BASIS FOR COMPARISON	PRIMARY DATA	SECONDARY DATA
Meaning	Primary data refers to the first hand data gathered by the researcher himself.	Secondary data means data collected by someone else earlier.
Data	Real time data	Past data
Process	Very involved	Quick and easy
Source	Surveys, observations, experiments, questionnaire, personal interview, etc.	Government publications, websites, books, journal articles, internal records etc.
Cost effectiveness	Expensive	Economical
Collection time	Long	Short
Specific	Always specific to the researcher's needs.	May or may not be specific to the researcher's need.
Available in	Crude form	Refined form
Accuracy and Reliability	More	Relatively less

Collect Data: Marketing Surveys

- In a marketing survey, we typically would like to collect the following information from customers:
  - purchase intention
  - willingness to pay (WTP)
  - shopping basket
  - share of wallet (SoW)
  - demographics
- **Let's see a quick example of how to design a marketing survey!**
- Useful supplementary readings if you need to design marketing surveys for your term 3 dissertation.
  - The quick start guide on how to conduct market research

Section 3

Data Wrangling with R

Data Frame Basics

- Data Frame is the R object that we will deal with most of the time in the MSc program. You can think of `data.frame` as a spreadsheet in excel
- Each row stands for an `observation`
- Each column stands for a `variable`; each column should have a **unique** name.
- Each column must contain the same data type, but the different columns can store different data types.[1]

[1] Compared with matrix, is there any difference despite both being two-dimensional?

Install and Load the dplyr package

- In R, we will be using the dplyr package for data cleaning and manipulation.

```
install.packages("dplyr")
```

- Load the package

```
library(dplyr)
```

- Load a csv format dataset called data_demo using read.csv()

```
data_demo <- read.csv("https://www.dropbox.com/s/a0v38lpydls2emy/demogr
```

- To browse the whole dataset, we can simply click the dataset in the environment

First Look at the Dataset

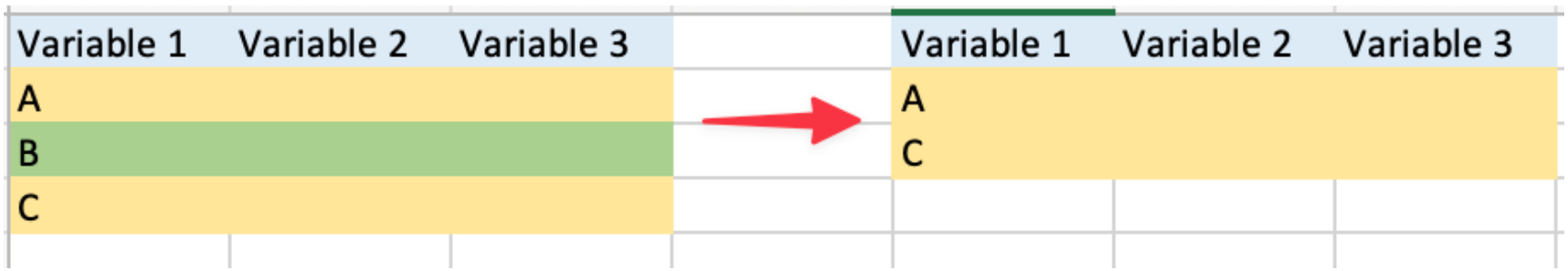
1. What variables do the data have?
2. What are the types of each variable?
   - Tip: We can use a function called `str()` short for structure.

Common Data Wrangling Operations

- Select rows (`filter`)
- Sort rows (`arrange`)
- Select columns (`select`)
- Generate new columns (`mutate`)
- Group aggregation (`group_by`)
- Merge datasets (`join`)

Subset Rows Based on Conditions: `filter`

- We can use `filter()` to select rows that meet certain logical criteria.
  - The filter operation results in a new dataset, which is a subset of the original dataset after filtering
  - The number of variables remains the same



- **Important**: To store the generated new subset of data in RStudio, we need to assign it to a new object.

Subset Rows Based on Conditions: `filter`

Example: From data_demo, find customers who are single

```
# keep only single customers
data_demo_single <- filter(data_demo, Marital_Status == "Single" )

# show the first 5 records using head()
head(data_demo_single,5)
```

ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency
5524	1957	Graduation	Single	58138	0	0	04/09/2012	58
2174	1954	Graduation	Single	46344	1	1	08/03/2014	38
2114	1946	PhD	Single	82800	0	0	24/11/2012	23
2278	1985	2n Cycle	Single	33812	1	0	03/11/2012	86
7892	1969	Graduation	Single	18589	0	0	02/01/2013	89

The Pipe Operator %>%

Pipe Operator

%>% passes the **object in front** as the **first argument** of the **subsequent function**.

input %>% functionName(, ...)

other inputs

functionName(input, ...)

other inputs

Example of the Pipe Operator %>%

```
1  # without using pipe
2  filter(data_demo, Marital_Status == 'Single')
3
4  # with pipe
5  data_demo %>% filter(Marital_Status == 'Single')
```

Why Do We Need Pipe Operator for Data Wrangling?

- **Exercise**: find out **single** customers who have a **PhD** without using pipe.

```
# based on data_demo, find out customers who are single
data_demo_single <-

# based on data_demo_single,
# further find out single customers who have a PhD
data_demo_single_PhD <-
```

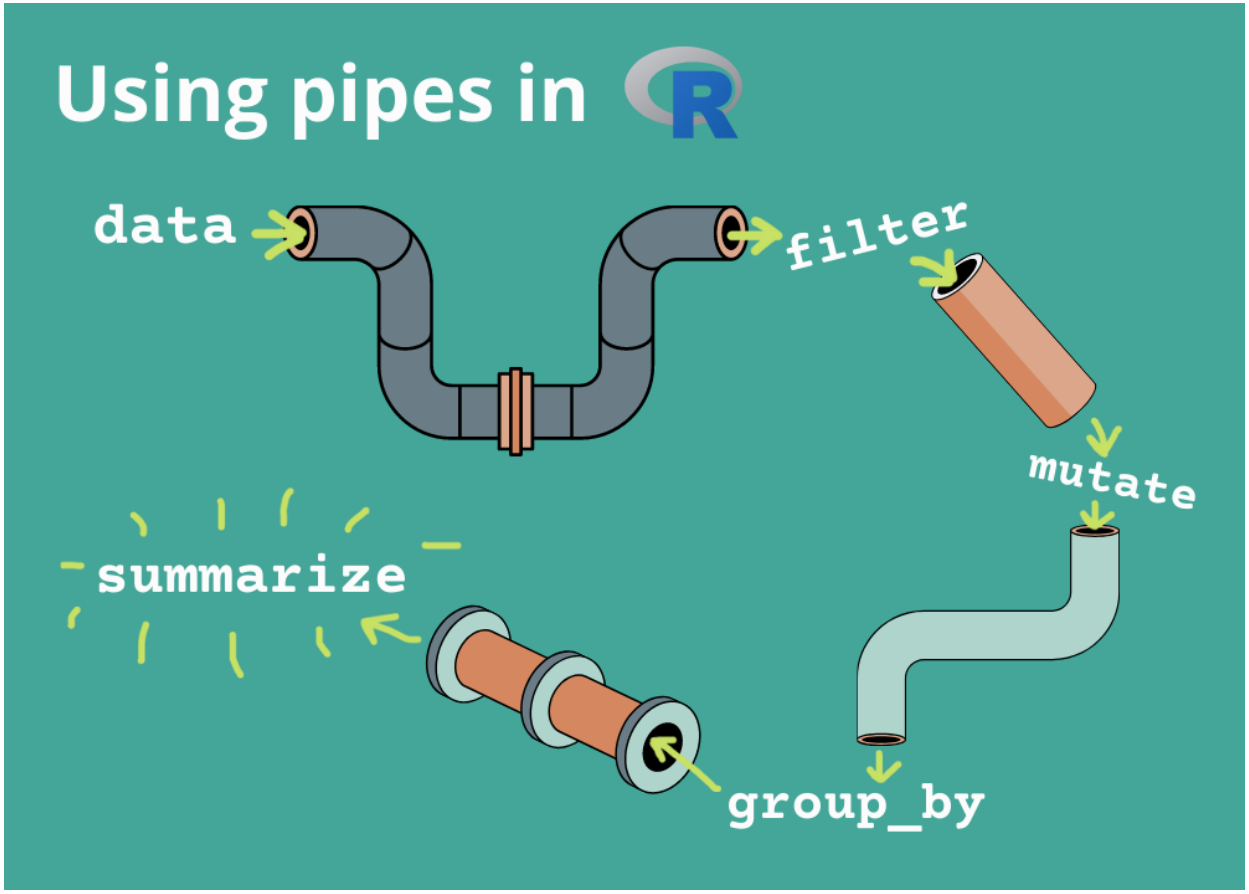
Why Do We Need Pipe Operator for Data Wrangling?

- **Exercise**: find out **single** customers who have a **PhD** using pipe.

```
data_demo_single_PhD <- data_demo %>%
  filter(Marital_Status == 'Single') %>%
  filter(Education == 'PhD') %>%
  head() ## You can even continue with more filter steps
```

Why Do We Need Pipe Operator for Data Wrangling?

- The pipe works like a conveyor belt in a factory, passing the intermediate outputs from the previous data wrangling step to the next step for further processing until you finish your data wrangling task.



Subset Rows Based on Multiple Conditions: `filter`

- We can also add multiple criteria using &, |, and ! to represent and, or, and not (induction week)

```
data_demo %>%
  filter(Marital_Status == 'Single' &
           Education == 'PhD') %>%
  head()
```

ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency
2114	1946	PhD	Single	82800	0	0	24/11/2012	23
7281	1959	PhD	Single	NA	0	0	05/11/2013	80
1016	1959	PhD	Single	34554	0	1	30/03/2014	43
7431	1991	PhD	Single	68126	0	0	10/11/2012	40
2937	1974	PhD	Single	68352	0	1	28/08/2013	47
5823	1970	PhD	Single	32303	0	1	08/03/2014	63

Sort Rows: `arrange`

- `arrange()` orders the rows by the values of selected columns.
  - ascending order by default; for descending order, put a minus sign.
  - allows multiple sorting variables separated by comma.
- **Example**: sort customers based on marital status in ascending order and number of teens in descending order.

```
data_demo %>%
  arrange(Marital_Status, -Teenhome)
```

- **Exercise:** sort customers based on income in descending order.

Generate New Variables: `mutate`

- `mutate()` generates new variables in the dataset while preserving existing variables
- **Example**: create a new variable named `Age` from `Year_Birth`.

```
data_demo %>%
  mutate(Age = 2023 - Year_Birth)
```

- **Exercise**: create a new variable named `totalkids`, which is the sum of `Kidhome` and `Teenhome`.

After-Class Exercise

- Data camp dplyr exercise
- Read "Preliminary Customer Analyses" dataset, and try to solve the case questions using the techniques learned today