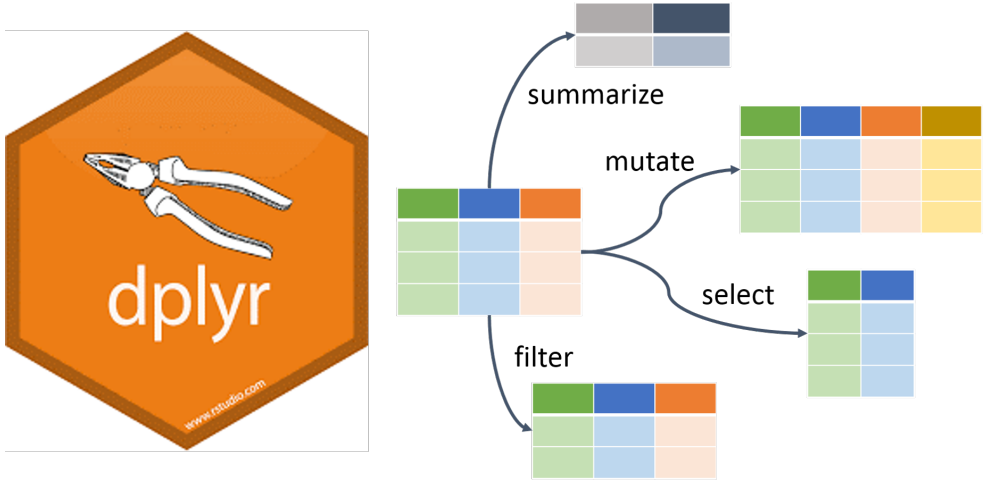
Transformation

Carolina A. de Lima Salge
Assistant Professor
Terry College of Business
University of Georgia

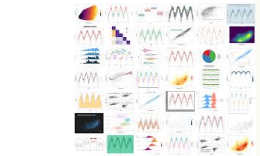
Business Intelligence
Spring 2021

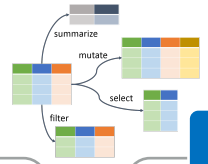


Data Value Creation Model

- ✓ Create Table Calculations
- ✓ Create Dual Axis Charts
- ✓ Create Calculated Fields in a Blend

- ✓ Create Barcharts
- ✓ Create Maps
- ✓ Create Piecharts
- ✓ Create Interactive Dashboards
- ✓ Create Area Charts
- ✓ Create Scatterplots
- ✓ Create Treemaps
- ✓ Create Storylines





- ✓ Connect Tableau to various Datasets: Excel and CSV files

Import

Question

Wrangle

- ✓ Understand Types of Joins and how they work

Explore

Model

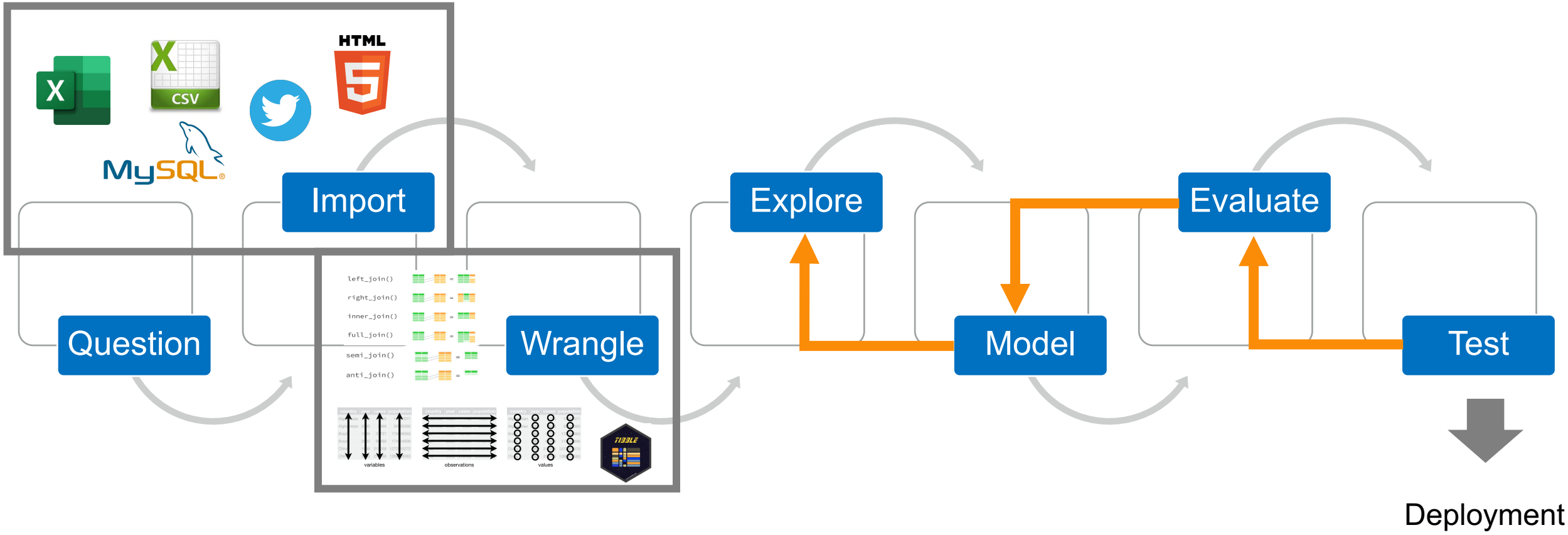
Evaluate

Test

Deployment

Database → Statistics → Machine Learning

Data Value Creation Model



Database → Statistics → Machine Learning

Import

Used the **readr** package to import CSV files & the **readxl** package to import Excel files. Used **RMySQL** and **DBI** packages to connect to and pull data from a relational database system

```
# Import data from CSV
library(readr)
CoffeeChain <- read_csv("CoffeeChain.csv")

# Import data from Excel
library(readxl)
CoffeeChain <- read_excel("CoffeeChain.xlsx")


# Save data as CSV
write_csv(CoffeeChain, "CoffeeChain.csv")
```

```
# Connect to database
library(RMySQL)
library(DBI)
conn1 <- dbConnect(MySQL(),
          host= "",
          dbname= "",
          user= "",
          password= "")

# Pull data from the database
products <- dbGetQuery(conn1, "select * from Products;")
```

Import

Used the **rvest** package to scrape data from HTML websites and the **rtweet** package to get data from Twitter

```
# Scrape data from IMDB
library(rvest)


lego_movie <- read_html("http://www.imdb.com/title/tt1490017/")

rating <- lego_movie %>%
  html_nodes("strong span") %>%
  html_text() %>%
  as.numeric()
```

```
# Scrape data from Twitter
library(rtweet)


big4 <- stream_tweets(
  q = "EY, PwC, Deloitte, KPMG",
  timeout = 30)


deloitee_timeline <- get_timeline("@Deloitte",
n = 3200)
```

Wrangling

Used the **tidyr** package to normalize untidy data and the **dplyr** package to join related tables

```
# Tidy data
library(tidyr)


# Gather the 1999 and 2000 columns into values
under the year variable and its corresponding
values as values under the cases variable
table4a %>%
         gather(`1999`, `2000`, key = "year",
value = "cases")
```

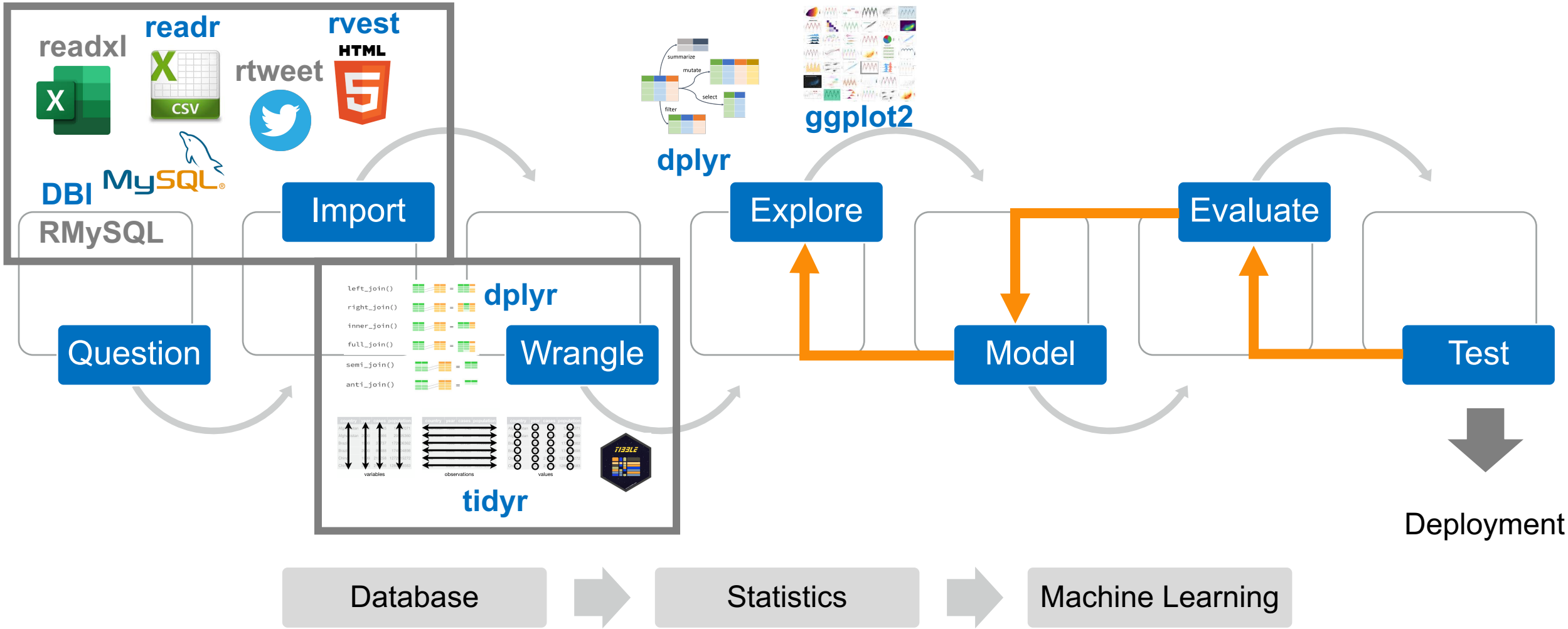
```
# Join tables
library(dplyr)


# Returns all rows from orderDetails where there are
matching values in orderList, and all columns
from orderDetails  and orderList
inner_join(orderDetails, orderList, by = "Order ID")
```

Packages & Resources

Package	Task	Tidyverse	Resource
readr	Import CSV files	Yes	https://evoldyn.gitlab.io/evomics-2018/ref-sheets/R_data-import.pdf
readxl	Import Excel files	*No*	
DBI	Connect to SQL database	Yes	https://blog.rsquaredacademy.com/working-with-databases-using-r/
RMySQL	Connect to SQL database	*No*	
rvest	Scrape data from HTML websites	Yes	https://github.com/yusuzech/r-web-scraping-cheat-sheet
rtweet	Scrape data from Twitter	*No*	https://github.com/ropensci/rtweet
tidyr	Fix untidy data	Yes	https://tidyr.tidyverse.org
dplyr	Join many related tables	Yes	https://dplyr.tidyverse.org

For R Markdown, see: https://rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf

Data Value Creation Model



Transformation

Wrangling is important for getting the data in a standard and useful format for analysis, but it is rare that you work with the data “as-is”

- Filter rows to work with certain segments
- Rename or reorder variables and observations
- Create new variables or grouped summaries

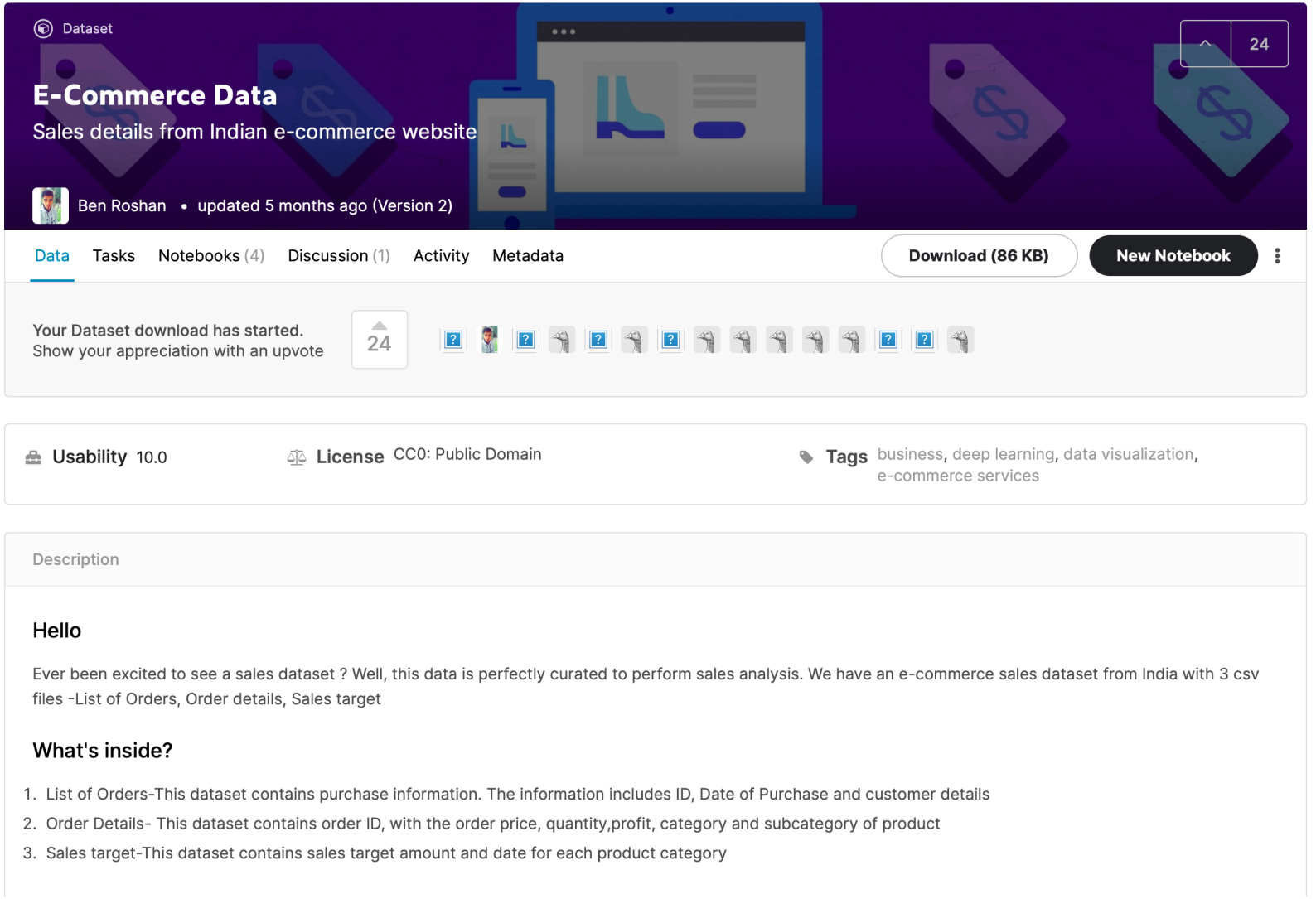
dplyr

One of the packages in the tidyverse that enables the transformation of data

- Look at a subset of the rows—**filter()**
- Reorder rows—**arrange()**
- Rename variables—**rename()**
- Create new variables—**mutate()**
- Collapse values down to a summary—**summarise()**

Data

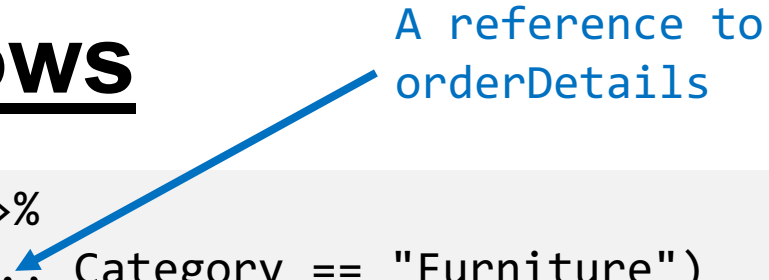
Rely on e-commerce data from Kaggle



Data Import

```
library(tidyverse)

orderList <- read_csv("List of Orders.csv")
orderDetails <- read_csv("Order Details.csv")
salesTarget <- read_csv("Sales target.csv")
```

Filter Rows




```
orderDetails %>%
        filter(., Category == "Furniture")
# A tibble: 243 x 6
   `Order ID` Amount Profit Quantity Category  `Sub-Category`
   <chr>       <dbl>  <dbl>    <dbl> <chr>     <chr>
 1 B-25601      1275  -1148        7 Furniture Bookcases
 2 B-25603        24    -30        1 Furniture Chairs
 3 B-25608      1364  -1864        5 Furniture Tables
 4 B-25608       476      0        3 Furniture Chairs
 5 B-25610        30     -5        2 Furniture Furnishings
 6 B-25612       259    -55        2 Furniture Chairs
 7 B-25614       494     54        4 Furniture Bookcases
 8 B-25618       362    127        1 Furniture Bookcases
 9 B-25626      1103   -276        3 Furniture Chairs
10 B-25628        35     -8        2 Furniture Furnishings
# … with 233 more rows
```

Filter Rows

and

```
orderDetails %>%
        filter(., Category == "Furniture", Quantity > 1)
# A tibble: 223 x 6
   `Order ID` Amount Profit Quantity Category  `Sub-Category`
   <chr>       <dbl>  <dbl>    <dbl> <chr>     <chr>
 1 B-25601      1275  -1148        7 Furniture Bookcases
 2 B-25608      1364  -1864        5 Furniture Tables
 3 B-25608       476      0        3 Furniture Chairs
 4 B-25610        30     -5        2 Furniture Furnishings
 5 B-25612       259    -55        2 Furniture Chairs
 6 B-25614       494     54        4 Furniture Bookcases
 7 B-25626      1103   -276        3 Furniture Chairs
 8 B-25628        35     -8        2 Furniture Furnishings
 9 B-25631        89    -89        2 Furniture Furnishings
10 B-25634       389    -83        3 Furniture Chairs
# … with 213 more rows
```

Filter Rows

and

```
orderDetails %>%
        filter(., Category == "Furniture" & Quantity > 1)
# A tibble: 223 x 6
   `Order ID` Amount Profit Quantity Category  `Sub-Category`
   <chr>       <dbl>  <dbl>    <dbl> <chr>     <chr>
 1 B-25601      1275  -1148        7 Furniture Bookcases
 2 B-25608      1364  -1864        5 Furniture Tables
 3 B-25608       476      0        3 Furniture Chairs
 4 B-25610        30     -5        2 Furniture Furnishings
 5 B-25612       259    -55        2 Furniture Chairs
 6 B-25614       494     54        4 Furniture Bookcases
 7 B-25626      1103   -276        3 Furniture Chairs
 8 B-25628        35     -8        2 Furniture Furnishings
 9 B-25631        89    -89        2 Furniture Furnishings
10 B-25634       389    -83        3 Furniture Chairs
# … with 213 more rows
```

Filter Rows

or

```
orderDetails %>%
        filter(., Category == "Furniture" | Quantity > 1)
# A tibble: 1,388 x 6
   `Order ID` Amount Profit Quantity Category    `Sub-Category`
   <chr>       <dbl>  <dbl>    <dbl> <chr>       <chr>
 1 B-25601      1275  -1148        7 Furniture   Bookcases
 2 B-25601        66    -12        5 Clothing    Stole
 3 B-25601         8     -2        3 Clothing    Hankerchief
 4 B-25601        80    -56        4 Electronics Electronic Games
 5 B-25602       168   -111        2 Electronics Phones
 6 B-25602       424   -272        5 Electronics Phones
 7 B-25602      2617   1151        4 Electronics Phones
 8 B-25602       561    212        3 Clothing    Saree
 9 B-25602       119     -5        8 Clothing    Saree
10 B-25603      1355    -60        5 Clothing    Trousers
# … with 1,378 more rows
```

Quiz “Question”

John is trying to create a new table called **posBAprofit** that filters the orderDetails table to show observations where profit is above zero but below the average. He has written the below code in R to achieve this goal. Is John’s code correct?

```
posBAprofit <- orderDetails %>%
       filter(., Profit < mean(Profit), Profit > 0)
```

Arrange Rows

```
orderDetails %>%
        arrange(., desc(Profit))
# A tibble: 1,500 x 6
   `Order ID` Amount Profit Quantity Category    `Sub-Category`
   <chr>       <dbl>  <dbl>    <dbl> <chr>       <chr>
 1 B-25973      4141   1698       13 Electronics Printers
 2 B-25602      2617   1151        4 Electronics Phones
 3 B-25761      2188   1050        5 Furniture   Bookcases
 4 B-25923      3873    891        6 Electronics Phones
 5 B-25830      1954    782        3 Electronics Phones
 6 B-26073      1514    742        4 Electronics Printers
 7 B-25853      2093    721        5 Furniture   Chairs
 8 B-26093      2847    712        8 Electronics Printers
 9 B-25862      2061    701        5 Furniture   Bookcases
10 B-25656      1389    680        7 Clothing    Saree
# … with 1,490 more rows
```

Rename Variables

New name Old name

```
orderDetails %>%
        rename(., profit = Profit)
# A tibble: 1,500 x 6
   `Order ID` Amount profit Quantity Category    `Sub-Category`
   <chr>       <dbl>  <dbl>    <dbl> <chr>       <chr>
 1 B-25601      1275  -1148        7 Furniture   Bookcases
 2 B-25601        66    -12        5 Clothing    Stole
 3 B-25601         8     -2        3 Clothing    Hankerchief
 4 B-25601        80    -56        4 Electronics Electronic Games
 5 B-25602       168   -111        2 Electronics Phones
 6 B-25602       424   -272        5 Electronics Phones
 7 B-25602      2617   1151        4 Electronics Phones
 8 B-25602       561    212        3 Clothing    Saree
 9 B-25602       119     -5        8 Clothing    Saree
10 B-25603      1355    -60        5 Clothing    Trousers
# … with 1,490 more rows
```

Add New Variables

Name of the new variable

Value of the new variable

```
orderDetails %>%
       mutate(., ProfitN = (Profit - min(Profit)) / (max(Profit) - min(Profit)))
# A tibble: 1,500 x 7
   `Order ID` Amount Profit Quantity Category    `Sub-Category`   ProfitN
   <chr>       <dbl>  <dbl>    <dbl> <chr>       <chr>              <dbl>
 1 B-25601      1275  -1148        7 Furniture   Bookcases          0.226
 2 B-25601        66    -12        5 Clothing    Stole              0.535
 3 B-25601         8     -2        3 Clothing    Hankerchief        0.538
 4 B-25601        80    -56        4 Electronics Electronic Games   0.523
 5 B-25602       168   -111        2 Electronics Phones             0.508
 6 B-25602       424   -272        5 Electronics Phones             0.465
 7 B-25602      2617   1151        4 Electronics Phones             0.851
 8 B-25602       561    212        3 Clothing    Saree              0.596
 9 B-25602       119     -5        8 Clothing    Saree              0.537
10 B-25603      1355    -60        5 Clothing    Trousers           0.522
# … with 1,490 more rows
```

Grouped Summaries

Grouped variables

Summarized function and variable

Name of the new summarized variable

```
orderDetails %>%
  group_by(Category, `Sub-Category`) %>%
  summarize(`Average Profit` = mean(Profit, na.rm = TRUE)) %>%
  arrange(desc(`Average Profit`))
# A tibble: 17 x 3
# Groups:   Category [3]
   Category    `Sub-Category`   `Average Profit`
   <chr>       <chr>                      <dbl>
 1 Electronics Printers                   80.6
 2 Clothing    Trousers                   73
 3 Furniture   Bookcases                  61.9
 4 Electronics Accessories                49.4
 5 Electronics Phones                     26.6
 6 Clothing    T-shirt                    19.5
 7 Clothing    Shirt                      16.4
 8 Clothing    Stole                      13.3
 9 Furniture   Furnishings                11.6
10 Clothing    Hankerchief                10.6
11 Furniture   Chairs                      7.80
12 Clothing    Leggings                    4.91
13 Clothing    Kurti                       3.85
14 Clothing    Skirt                       3.67
15 Clothing    Saree                       1.68
16 Electronics Electronic Games          -15.6
17 Furniture   Tables                   -236.
```

The Pipe

The `%>%` focuses on the transformations, not what is being transformed, which makes the code easier to read

- Take the `orderDetails` dataset **then** `group_by()` **then** `summarise()` **then** `arrange()`

```
orderDetails %>%
  group_by(Category, `Sub-Category`) %>%
  summarize(`Average Profit` = mean(Profit, na.rm = TRUE)) %>%
  arrange(desc(`Average Profit`))
```

Useful Summary Functions

- Location
  - **mean(x)** and **median(x)**
- Spread
  - **sd(x)** and **IQR(x)**
- Rank
  - **min(x)**, **quantile(x, 0.25)**, and **max(x)**
- Count
  - **n(x)**, **sum(!is.na(x))**, and **n_distinct(x)**

At-Home Exercises

Copy and paste the code (not the results of running the code, which are in blue) from the slides (starting on slide 12) into an R Markdown document. Execute the code in R, line by line. Are your results like the ones in the slides? If yes, try to knit the code to PDF

Read chapter 5 and do the exercises of the R for Data Science book

Check out the dplyr cheat sheet: https://github.com/rstudio/cheatsheets/raw/master/data-transformation.pdf

Open the orderDetails dataset in Tableau. Next, try to use the software to filter and arrange rows, rename and add new variables, and execute grouped summaries (very similar to how we used R to do it). How long did it take you? Which functionalities did you use?

Thank You!