Introduction to Business Analysis

Lecture 4: Data Preprocessing and Transformation. MS Excel and R

Iegor Vysnevskyi

Woosong University

March 21/27, 2023

Agenda

1. Introduction to Data Preprocessing and Transformation
2. Processing and Transformation Techniques in MS Excel
3. Process Data in R
4. Data Transformation in R
5. In-class Assignment
6. Home Assignment

1. Introduction to Data Preprocessing and Transformation

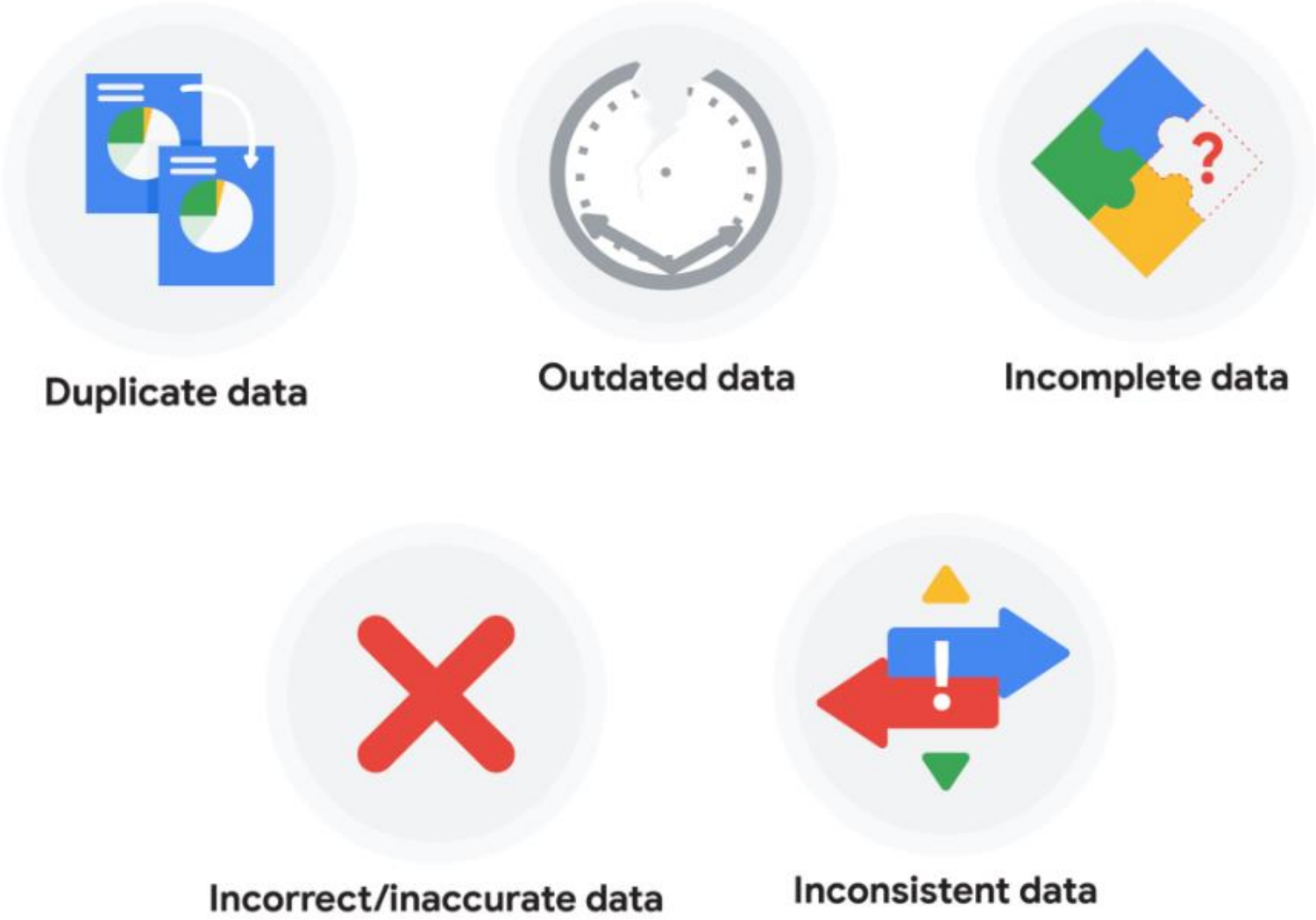
Data Processing is the overall process of manipulating, organizing, and structuring raw data into a more usable form.

It typically involves working with the ***dirty data*** such as cleaning data, removing duplicates, and formatting data for analysis.

Data Transformation is the process of converting data from one format or structure to another.

This may involve tasks such as splitting columns, merging data sets, or aggregating data.

Types of Dirty Data



Source: Coursera course "Process Data From Dirty to Clean"

Types of Dirty Data

- To deal with dirty data it is better to develop your own check list which you can refer to and improve in the future.

Data Cleaning Checklist	Preferred cleaning methods

2. Processing and Transformation Techniques in MS Excel

File to be used: *International-Logistics-Association-Memberships.csv*

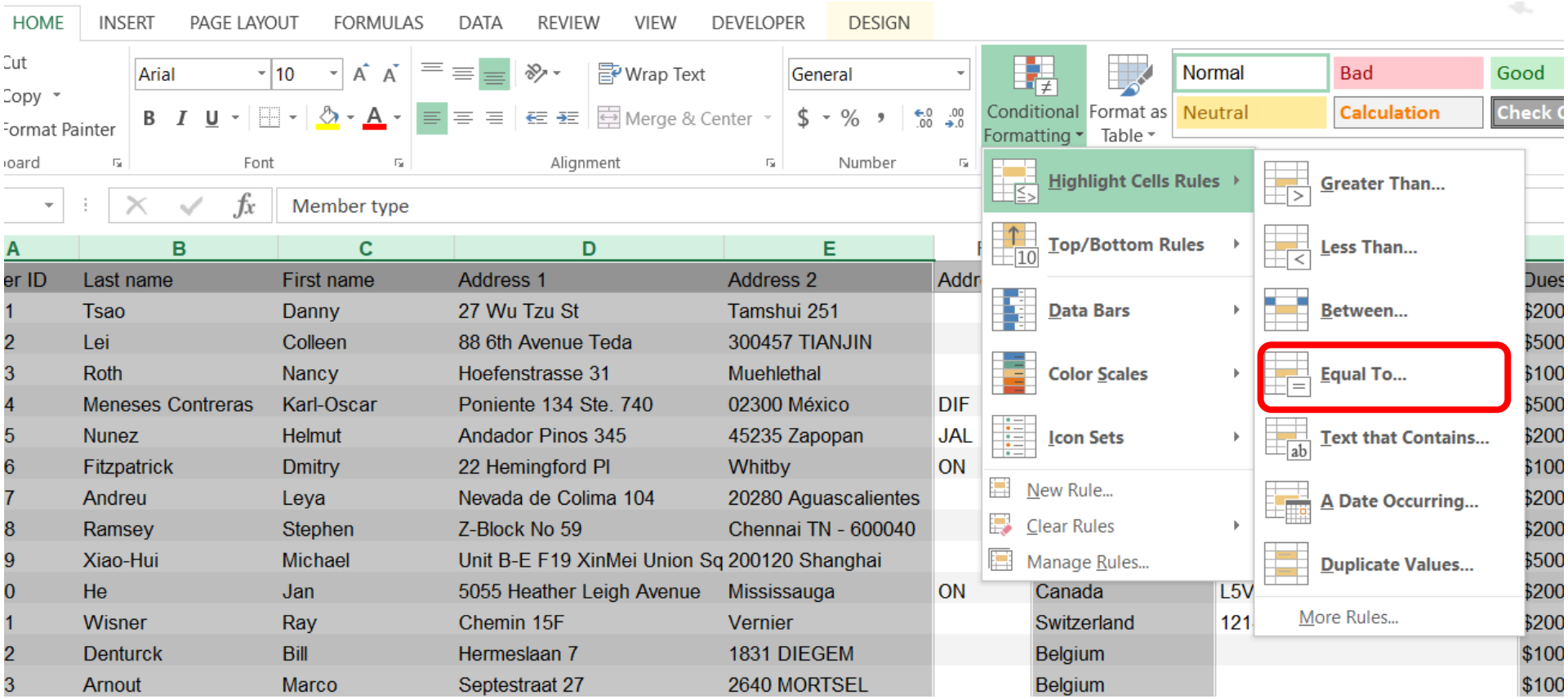
Conditional formatting

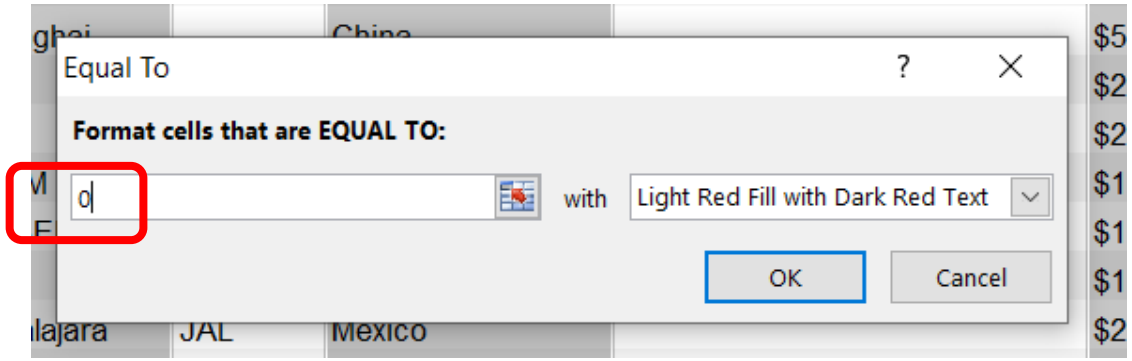
- **Identify missing values** by conditional formatting

Let’s apply conditional formatting to all columns in the table except for “Address 3”, “Address 5” and “Certification” columns.

General instructions:

1. Select the cells you want to apply conditional formatting to.
2. Go to the Home tab on the ribbon.
3. Click on the Conditional Formatting option.
4. Choose the type of formatting you want to apply (e.g. highlight cells rules, top/bottom rules, data bars, color scales, icon sets, etc.).
5. Choose the formatting options you want to apply (e.g. select the colors, the minimum/maximum values, the criteria for highlighting, etc.).
6. Click OK to apply the formatting.


HOME
INSERT
PAGE LAYOUT
FORMULAS
DATA
REVIEW
VIEW
DEVELOPER
DESIGN
Cut
Copy
Format Painter
Arial
10
Wrap Text
General
Merge & Center
Font
Alignment
Number
Conditional Formatting
Format as Table
Normal
Bad
Good
Neutral
Calculation
Member type
Highlight Cells Rules
Top/Bottom Rules
Data Bars
Color Scales
Icon Sets
New Rule...
Clear Rules
Manage Rules...
Greater Than...
Less Than...
Between...
Equal To...
Text that Contains...
A Date Occurring...
Duplicate Values...
More Rules...
er ID | Last name | First name | Address 1 | Address 2
1 | Tsao | Danny | 27 Wu Tzu St | Tamshui 251
2 | Lei | Colleen | 88 6th Avenue Teda | 300457 TIANJIN
3 | Roth | Nancy | Hoefenstrasse 31 | Muehlethal
4 | Meneses Contreras | Karl-Oscar | Poniente 134 Ste. 740 | 02300 México
5 | Nunez | Helmut | Andador Pinos 345 | 45235 Zapopan
6 | Fitzpatrick | Dmitry | 22 Hemingford Pl | Whitby
7 | Andreu | Leya | Nevada de Colima 104 | 20280 Aguascalientes
8 | Ramsey | Stephen | Z-Block No 59 | Chennai TN - 600040
9 | Xiao-Hui | Michael | Unit B-E F19 XinMei Union Sq | 200120 Shanghai
0 | He | Jan | 5055 Heather Leigh Avenue | Mississauga
1 | Wisner | Ray | Chemin 15F | Vernier
2 | Denturck | Bill | Hermeslaan 7 | 1831 DIEGEM
3 | Arnout | Marco | Septestraat 27 | 2640 MORTSEL

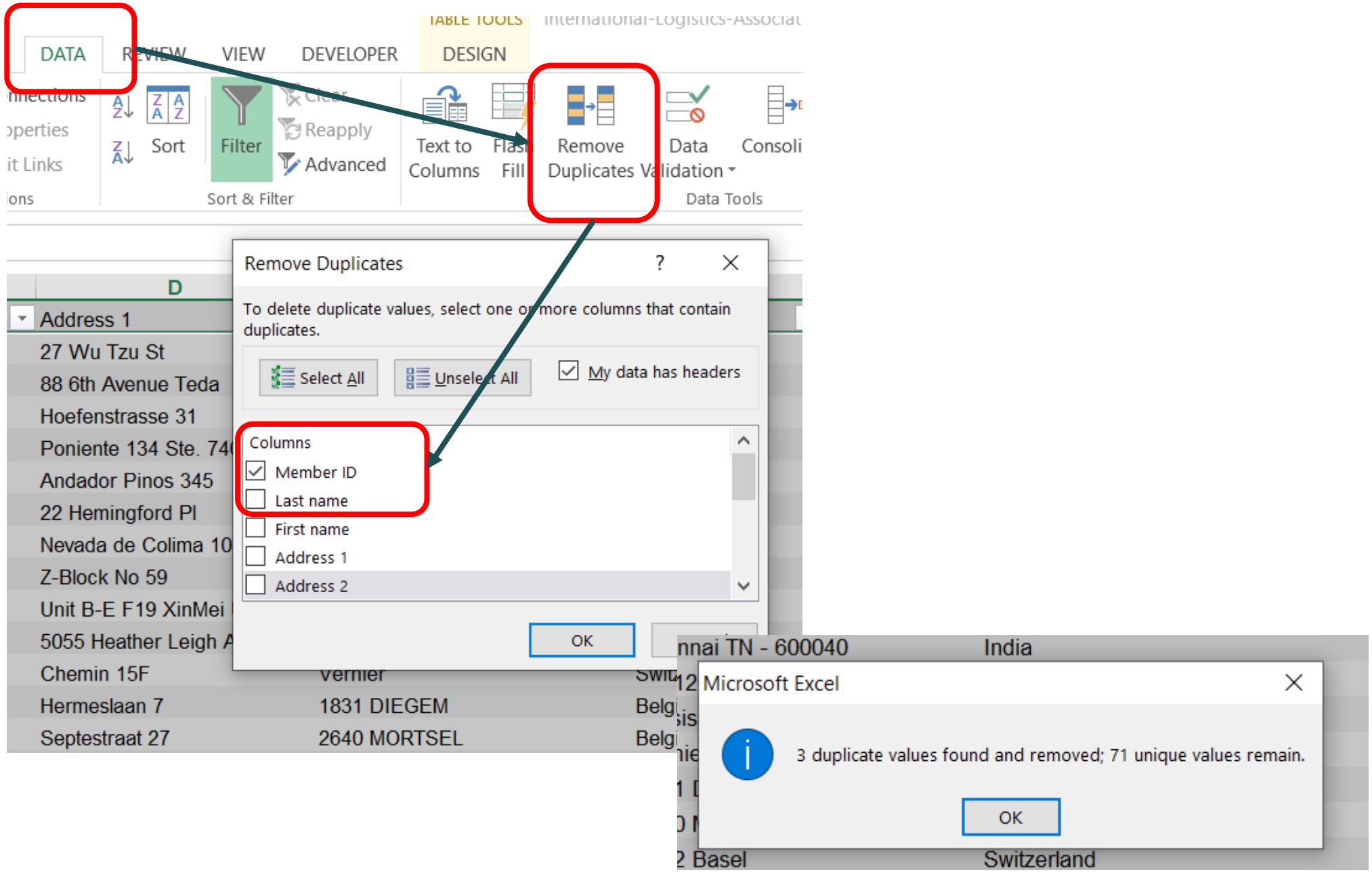

Equal To
Format cells that are EQUAL TO:
0
with
Light Red Fill with Dark Red Text
OK
Cancel

Remove duplicates

Let’s remove duplicates basing in Member ID

General instructions:

1. Click on the "Data" tab in the ribbon at the top of the screen.
2. Click on the "Remove Duplicates" button in the "Data Tools" section of the ribbon.
3. In the "Remove Duplicates" dialog box, select the columns that you want to check for duplicates.
4. Click the "OK" button.

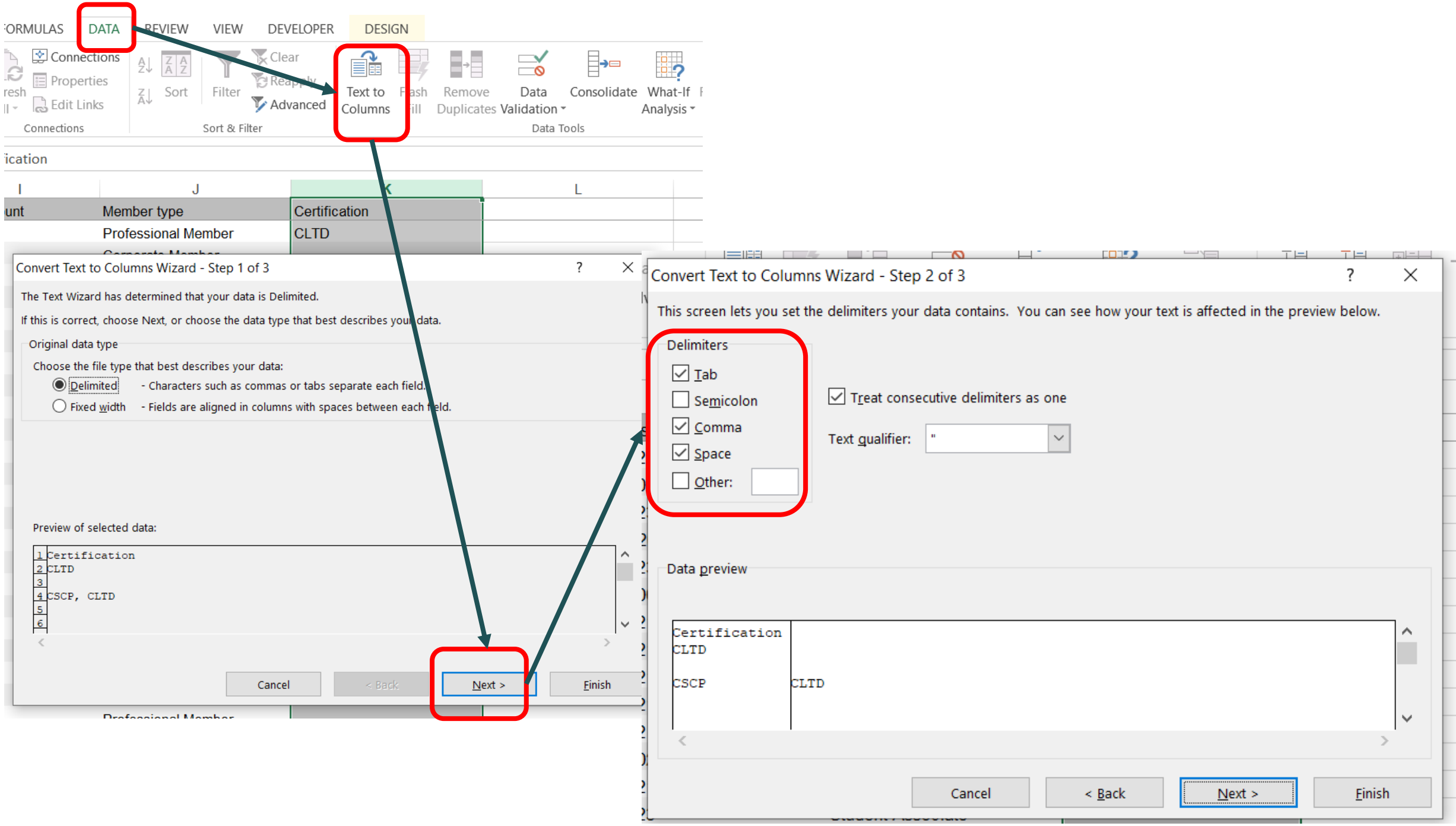

DATA
REVIEW
VIEW
DEVELOPER
DESIGN
Sort
Filter
Reapply
Advanced
Text to Columns
Remove Duplicates
Data Validation
Sort & Filter
Data Tools
Remove Duplicates
To delete duplicate values, select one or more columns that contain duplicates.
Select All
Unselect All
My data has headers
Columns
Member ID
Last name
First name
Address 1
Address 2
OK
Address 1
27 Wu Tzu St
88 6th Avenue Teda
Hoefenstrasse 31
Poniente 134 Ste. 74
Andador Pinos 345
22 Hemingford Pl
Nevada de Colima 10
Z-Block No 59
Unit B-E F19 XinMei
5055 Heather Leigh A
Chemin 15F
Hermeslaan 7
1831 DIEGEM
Septestraat 27
2640 MORTSEL
India
Switzerland
Microsoft Excel
3 duplicate values found and removed; 71 unique values remain.
OK

Splitting the data

Let’s split certification info into different columns

General instructions:

- Select the cell(s) containing the data you want to split.
- Go to the Data tab on the ribbon.
- Click on the "Text to Columns" button.
- In the "Convert Text to Columns Wizard" dialog box that appears, choose the type of data you want to split (e.g. delimited, fixed width, etc.) and click "Next".
- Depending on the type of data you selected, you may need to choose additional options (e.g. specify the delimiter character, set the column widths, etc.). Follow the on-screen instructions and click "Next" to proceed.
- Choose the format for each of the columns you want to create (e.g. general, text, date, etc.) and click "Finish".
- Excel will split the data in the original cell(s) into different cells based on the criteria you specified.

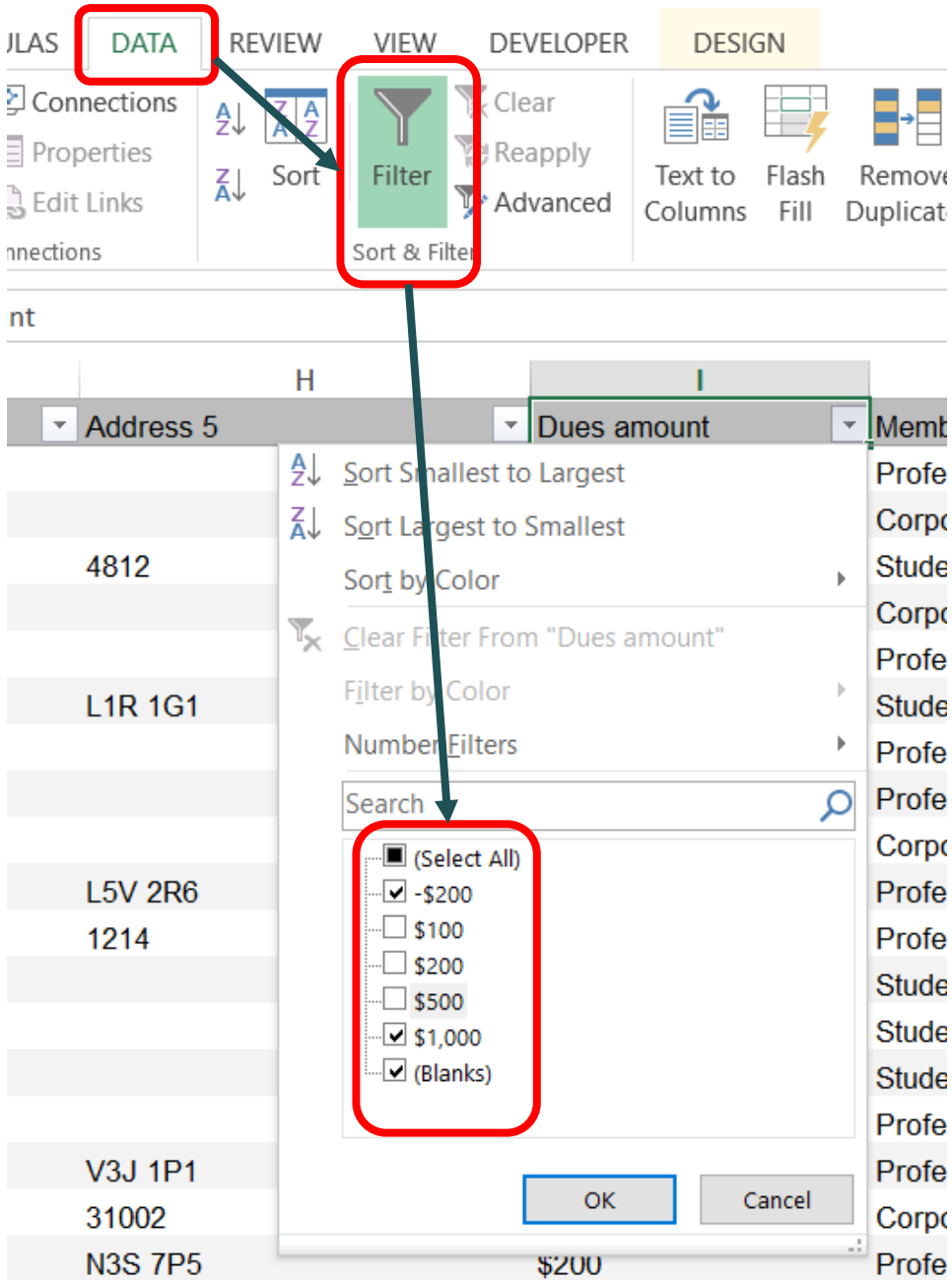

ORMULAS
DATA
REVIEW
VIEW
DEVELOPER
DESIGN
Connections
Properties
Edit Links
Connections
Sort
Filter
Clear
Reapply
Advanced
Sort & Filter
Text to Columns
Flash Fill
Remove Duplicates
Data Validation
Consolidate
What-If Analysis
Data Tools
ication
I
J
K
L
unt
Member type
Certification
Professional Member
CLTD
Convert Text to Columns Wizard - Step 1 of 3
The Text Wizard has determined that your data is Delimited.
If this is correct, choose Next, or choose the data type that best describes your data.
Original data type
Choose the file type that best describes your data:
Delimited - Characters such as commas or tabs separate each field.
Fixed width - Fields are aligned in columns with spaces between each field.
Preview of selected data:
1 Certification
2 CLTD
3
4 CSCP, CLTD
5
6
Cancel
< Back
Next >
Finish
Convert Text to Columns Wizard - Step 2 of 3
This screen lets you set the delimiters your data contains. You can see how your text is affected in the preview below.
Delimiters
Tab
Semicolon
Comma
Space
Other:
Treat consecutive delimiters as one
Text qualifier:
"
Data preview
Certification
CLTD
CSCP
CLTD
Cancel
< Back
Next >
Finish

Filtering data

Let’s check for inconsistent data

General instructions:

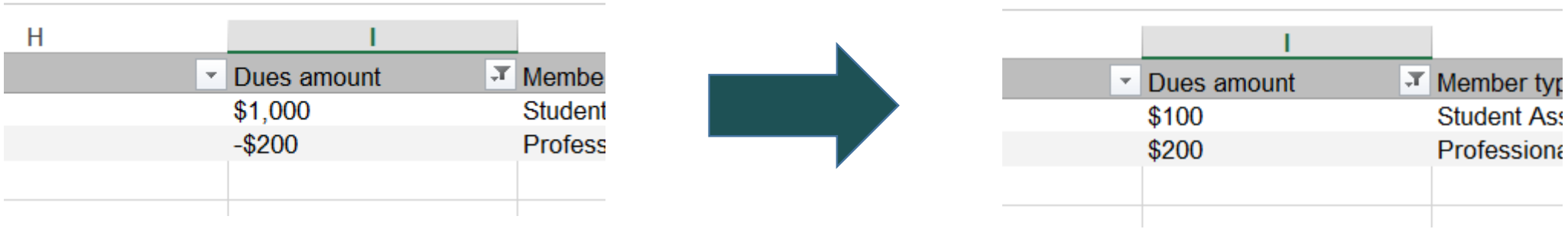
1. Select all range of cells that you want to filter.
2. Go to the "Data" tab in the ribbon menu at the top of the screen.
3. Click on the "Filter" button in the "Sort & Filter" group. This will add filter dropdowns to the header row of your data.
4. Click on the dropdown arrow in the header row of the column you want to filter. This will open the filter menu.



Let's say that we know that the dues range should be within $100-500.

The filter menu shows us two inconsistent data. Let's choose them to examine in more details.

Fix inaccurate data.



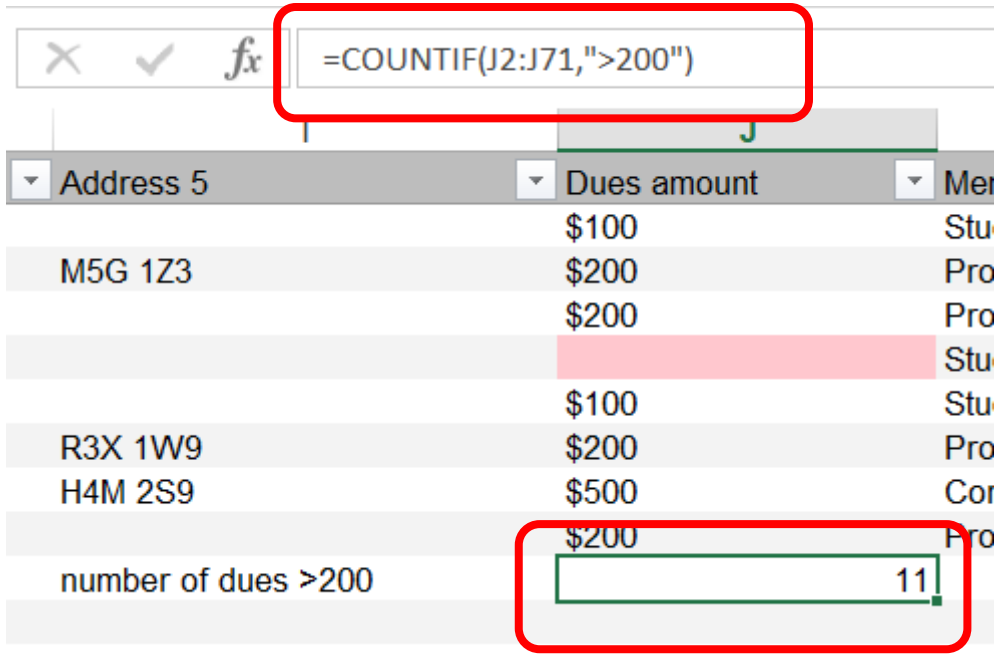
H	I	
	Dues amount	Membe
	$1,000	Student
	-$200	Profess

	I	
	Dues amount	Member typ
	$100	Student Ass
	$200	Professiona

COUNTIF()

- count the number of cells in a range that contain numeric values with certain condition

Let’s count the number of dues of the amount more than $200.

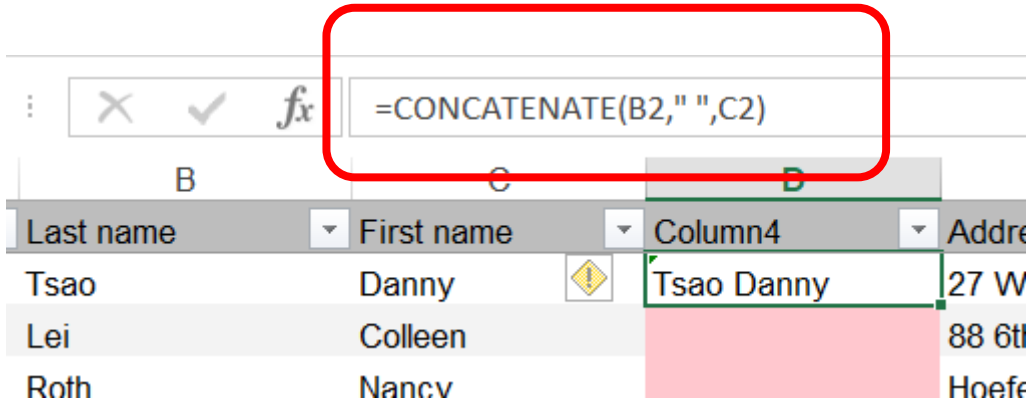


CONCATENATE()

- combine text from different cells into a single cell

Let’s combine last and first name of the members into one cell.

For this purpose create new column and write corresponding function.



3. Process Data in R

install.packages() and library()

- install packages from CRAN (Comprehensive R Archive Network), which is a repository of R packages maintained by the R community
- load packages that have been installed on your system into your R workspace so that you can use their functions

```r
#basic package
install.packages("tidyverse")
library(tidyverse)

#useful for data processing and transformation
install.packages("dplyr")
library(dplyr)
install.packages("tidyr")
library(tidyr)

#to load the csv file
install.packages("utils")
library(utils)
```

read.csv()

- read a CSV (Comma-Separated Values) file and returns its contents as a data frame

```
#load data set on data stocks prcies
data_stocks <- read.csv(file = "https://raw.githubusercontent.com/singh1985/rforanalytics/master/data/us_stocks.csv",
                        header = TRUE)
```

```
#load a dataset on video games sales
vg_data <- read.csv(file = "https://gist.githubusercontent.com/Ironraptor3/34f3938c7
                        header = TRUE) #the file is without header
```

head()

- view the first few rows of a data frame

```
#view first several rows of the table
head(data_stocks)
```

Result:

```
       Date  MSFT    IBM  AAPL   MCD    PG GOOG
1 2/01/2002 33.52 121.50 11.65 26.49 40.00   NA
2 3/01/2002 34.62 123.66 11.79 26.79 39.62   NA
3 4/01/2002 34.45 125.60 11.84 26.99 39.22   NA
4 7/01/2002 34.28 124.05 11.45 27.20 38.78   NA
5 8/01/2002 34.69 124.70 11.30 27.36 38.88   NA
6 9/01/2002 34.36 124.49 10.82 26.88 38.60   NA
```

```
#to view the exact number of first rows of the table
head(data_stocks, 3)
```

Result:

```
       Date  MSFT    IBM  AAPL   MCD    PG GOOG
1 2/01/2002 33.52 121.50 11.65 26.49 40.00   NA
2 3/01/2002 34.62 123.66 11.79 26.79 39.62   NA
3 4/01/2002 34.45 125.60 11.84 26.99 39.22   NA
```

tail()

- view the last few rows of a data frame

```
#to view the tail of the table
tail(data_stocks)
tail(data_stocks, 2)
```

Result:

```
> tail(data_stocks)
           Date  MSFT    IBM   AAPL   MCD    PG   GOOG
2779 21/12/2012 27.45 193.42 519.33 90.18 68.72 715.63
2780 24/12/2012 27.06 192.40 520.17 89.29 68.52 709.50
2781 26/12/2012 26.86 191.95 513.00 88.74 68.00 708.87
2782 27/12/2012 26.96 192.71 515.06 88.72 67.97 706.29
2783 28/12/2012 26.55 189.83 509.59 87.58 67.15 700.01
2784 31/12/2012 26.71 191.55 532.17 88.21 67.89 707.38
> tail(data_stocks, 2)
           Date  MSFT    IBM   AAPL   MCD    PG   GOOG
2783 28/12/2012 26.55 189.83 509.59 87.58 67.15 700.01
2784 31/12/2012 26.71 191.55 532.17 88.21 67.89 707.38
```

Extract a specific column

```r
#to return MSFT data as a vector
data_stocks$MSFT

#to return MSFT data as column
data_stocks["MSFT"]
```

Result:

```
 [1] 33.52 34.62 34.45 34.28 34.69 34.36 34.64 34.30 34.2
[20] 31.42 31.86 31.33 30.56 30.58 30.20 29.90 30.32 30.5
[39] 29.20 29.17 30.68 31.60 31.54 31.82 31.36 31.98 32.1
```

Result:

	MSFT
1	33.52
2	34.62
3	34.45
4	34.28
5	34.69
6	34.36
7	34.64
8	34.30
9	34.24
10	34.78

Two options to return the specific value

Number of the row

```
#return the price of AAPL stock in 3th row
data_stocks[3,"AAPL"]
data_stocks[3,4]

#return the values of AAPL stock in 3-5 rows
data_stocks[3:5,"AAPL"]
data_stocks[3:5, 4]
```

Number of the column

- Create a table with MSFT daily returns.

Calculate daily returns of MSFT stocks

```
msft_ret <- diff(data_stocks$MSFT)
```

Assign the name of new data frame

length()

- returns the number of elements in a vector or the number of columns or rows in a matrix or data frame

```
#check the number of values for MSFT columns and MSFT returns
length(data_stocks$MSFT)
length(msft_ret)
```

Result:

```
> length(data_stocks$MSFT)
[1] 2784
> length(msft_ret)
[1] 2783
```

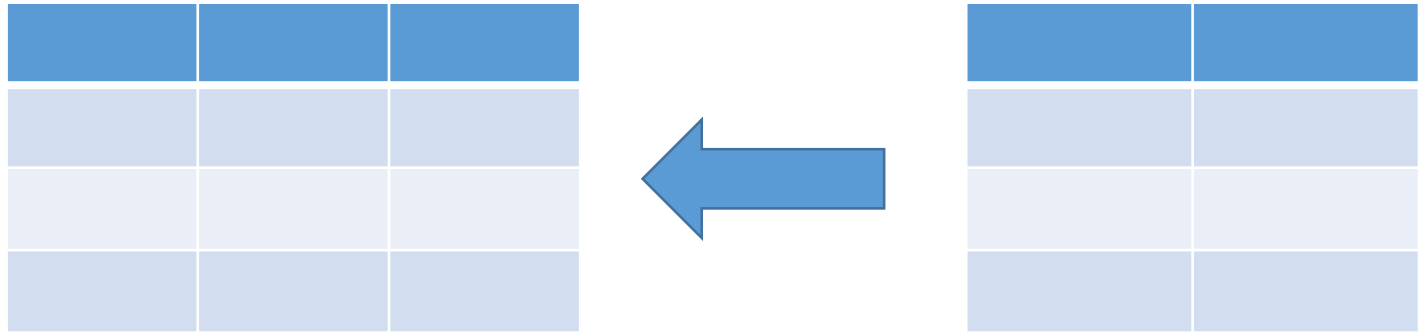
```
#add a zero value to the beginning of the vector
msft_ret <- c(0, msft_ret)
#check the length
length(msft_ret)
```

Result:

```
> length(msft_ret)
[1] 2784
```

cbind()

- combine vectors, matrices, or data frames by column



```
#combine the tables adding the column msft_ret to the data_stocks table
data_stocks_r <- cbind(data_stocks, msft_ret)
head(data_stocks_r)
```

Result:

```
      Date  MSFT    IBM  AAPL   MCD    PG GOOG msft_ret
1 2/01/2002 33.52 121.50 11.65 26.49 40.00   NA     0.00
2 3/01/2002 34.62 123.66 11.79 26.79 39.62   NA     1.10
3 4/01/2002 34.45 125.60 11.84 26.99 39.22   NA    -0.17
4 7/01/2002 34.28 124.05 11.45 27.20 38.78   NA    -0.17
5 8/01/2002 34.69 124.70 11.30 27.36 38.88   NA     0.41
6 9/01/2002 34.36 124.49 10.82 26.88 38.60   NA    -0.33
```

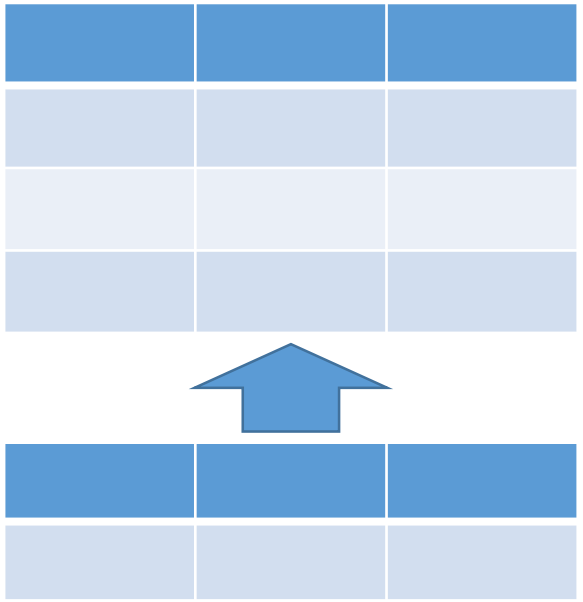
rbind()

- combine vectors, matrices, or data frames by row

```
#create two dataframes from data_stocks
data_r1 <- data_stocks_r[1:10,]  #first 10 rows
data_r2 <- data_stocks_r[2775:2784,]  #last 10 rows

#combine the tables adding one under another
data_stocks_rbind <- rbind(data_r1, data_r2)

#check the results
head(data_stocks_rbind)
tail(data_stocks_rbind)
```



Result:

```
       Date  MSFT    IBM  AAPL   MCD    PG GOOG msft_ret
1 2/01/2002 33.52 121.50 11.65 26.49 40.00   NA     0.00
2 3/01/2002 34.62 123.66 11.79 26.79 39.62   NA     1.10
3 4/01/2002 34.45 125.60 11.84 26.99 39.22   NA    -0.17
4 7/01/2002 34.28 124.05 11.45 27.20 38.78   NA    -0.17
5 8/01/2002 34.69 124.70 11.30 27.36 38.88   NA     0.41
6 9/01/2002 34.36 124.49 10.82 26.88 38.60   NA    -0.33
> tail(data_stocks_rbind)
           Date  MSFT    IBM   AAPL   MCD    PG   GOOG msft_ret
2779 21/12/2012 27.45 193.42 519.33 90.18 68.72 715.63    -0.23
2780 24/12/2012 27.06 192.40 520.17 89.29 68.52 709.50    -0.39
2781 26/12/2012 26.86 191.95 513.00 88.74 68.00 708.87    -0.20
2782 27/12/2012 26.96 192.71 515.06 88.72 67.97 706.29     0.10
2783 28/12/2012 26.55 189.83 509.59 87.58 67.15 700.01    -0.41
2784 31/12/2012 26.71 191.55 532.17 88.21 67.89 707.38     0.16
```

summary()

- summary of the central tendency, dispersion, and shape of a distribution for a given vector or data frame

```
#to examine the table and check for missing values
summary(data_stocks)
```

Result:

```
     Date                MSFT             IBM              AAPL              MCD              PG              GOOG       
 Length:2784        Min.   :15.15   Min.   : 55.07   Min.   :  6.56   Min.   : 12.38   Min.   :37.23   Min.   :100.0  
 Class :character   1st Qu.:25.27   1st Qu.: 84.43   1st Qu.: 19.46   1st Qu.: 29.16   1st Qu.:52.61   1st Qu.:385.1  
 Mode  :character   Median :26.92   Median : 99.34   Median : 94.00   Median : 50.80   Median :59.91   Median :486.5  
                    Mean   :26.87   Mean   :113.72   Mean   :160.85   Mean   : 50.48   Mean   :57.79   Mean   :469.9  
                    3rd Qu.:28.79   3rd Qu.:128.25   3rd Qu.:235.91   3rd Qu.: 66.71   3rd Qu.:63.85   3rd Qu.:579.8  
                    Max.   :37.06   Max.   :211.00   Max.   :702.10   Max.   :101.74   Max.   :74.67   Max.   :768.0  
                    NA's   :1                        NA's   :1                                         NA's   :663
```

na.omit()

- remove any rows with missing values from a data frame

```
#remove all rows with NA from data_stocks
data_stocks_naomit <- na.omit(data_stocks)

head(data_stocks)
head(data_stocks_naomit)
```

Result:

```
> head(data_stocks)
      Date  MSFT    IBM  AAPL   MCD    PG GOOG
1 2/01/2002 33.52 121.50 11.65 26.49 40.00   NA
2 3/01/2002 34.62 123.66 11.79 26.79 39.62   NA
3 4/01/2002 34.45 125.60 11.84 26.99 39.22   NA
4 7/01/2002 34.28 124.05 11.45 27.20 38.78   NA
5 8/01/2002 34.69 124.70 11.30 27.36 38.88   NA
6 9/01/2002 34.36 124.49 10.82 26.88 38.60   NA
```

```
> head(data_stocks_naomit)
          Date  MSFT    IBM  AAPL   MCD    PG   GOOG
663 19/08/2004 27.12 84.89 15.36 26.60 54.48 100.34
664 20/08/2004 27.20 85.25 15.40 27.07 54.85 108.31
665 23/08/2004 27.24 84.65 15.54 26.64 54.75 109.40
666 24/08/2004 27.24 84.71 15.98 26.87 54.95 104.87
667 25/08/2004 27.55 85.07 16.52 26.95 55.30 106.00
668 26/08/2004 27.44 84.69 17.33 27.10 55.70 107.91
```

4. Data Transformation in R

subset()

- extract a subset of a data frame based on certain conditions

```r
#Subset the data to include only the columns for Microsoft (MSFT) and IBM (IBM)
subset_data <- subset(data_stocks,
                      select=c("Date", "MSFT", "IBM"))

head(subset_data)
```

Result:

```
        Date  MSFT    IBM
1 2/01/2002 33.52 121.50
2 3/01/2002 34.62 123.66
3 4/01/2002 34.45 125.60
4 7/01/2002 34.28 124.05
5 8/01/2002 34.69 124.70
6 9/01/2002 34.36 124.49
```

pivot_longer()

- convert table from wide to long format

```
# Convert data_stocks table from wide to long format
data_stocks_l <- pivot_longer(data_stocks,
                      # tells pivot_longer() to use all columns in data_stocks
                      #except for Date as the values to reshape
                      cols = -Date,
                      #specifies that the column headers in the original
                      #data_stocks table should become the values in
                      #the new Stock column of data_stocks_l
                      names_to = "Stock",
                      #tells pivot_longer() to stack the data_stocks values
                      #in a single column named Price
                      values_to = "Price")

head(data_stocks_l)
```

Result:

	Date	Stock	Price
	<chr>	<chr>	<dbl>
1	2/01/2002	MSFT	33.5
2	2/01/2002	IBM	122.
3	2/01/2002	AAPL	11.6
4	2/01/2002	MCD	26.5
5	2/01/2002	PG	40
6	2/01/2002	GOOG	NA

select()

- select specific columns from a data frame

```
# select only the columns Rank, Name, Year, and Global_Sales
vg_data_select <- select(vg_data, Rank, Name, Year, Global_Sales)
head(vg_data_select)
```

Name of dataset

Columns to be selected

Result:

	Rank	Name	Year	Global_Sales
1	259	Asteroids	1980	4.31
2	545	Missile Command	1980	2.76
3	1768	Kaboom!	1980	1.15
4	1971	Defender	1980	1.05
5	2671	Boxing	1980	0.77
6	4027	Ice Hockey	1980	0.49

filter()

- filter rows based on conditions

```
# filter the data to include only games released in or after 2010
vg_data_filtered <- filter(vg_data, Year >= 2010)
head(vg_data_filtered)
```

Name of dataset Filter condition

Result:

```
  Rank                      Name Platform Year         Genre                 Publisher NA_Sales EU_Sales JP_Sales
1   16          Kinect Adventures!     X360 2010          Misc     Microsoft Game Studios    14.97     4.94     0.24
2   27 Pokemon Black/Pokemon White       DS 2010 Role-Playing                  Nintendo     5.57     3.28     5.65
3   32     Call of Duty: Black Ops     X360 2010       Shooter                Activision     9.67     3.73     0.11
4   41     Call of Duty: Black Ops      PS3 2010       Shooter                Activision     5.98     4.44     0.48
5   55                Gran Turismo 5      PS3 2010        Racing Sony Computer Entertainment     2.96     4.88     0.81
6   63                   Halo: Reach     X360 2010       Shooter     Microsoft Game Studios     7.03     1.98     0.08
  Other_Sales Global_Sales
1        1.67        21.82
2        0.82        15.32
3        1.13        14.64
4        1.83        12.73
5        2.12        10.77
6        0.78         9.88
```

mutate()

- create new columns based on existing columns

```
# add a new column to the data frame that calculates the total sales for each game
vg_data_mutated <- mutate(vg_data,
                          Total_Sales = NA_Sales + EU_Sales + JP_Sales + Other_Sales + Global_Sales)
head(vg_data_mutated)
```

Result:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Total_Sales
1	259	Asteroids	2600	1980	Shooter	Atari	4.00	0.26	0	0.05	4.31	8.62
2	545	Missile Command	2600	1980	Shooter	Atari	2.56	0.17	0	0.03	2.76	5.52
3	1768	Kaboom!	2600	1980	Misc	Activision	1.07	0.07	0	0.01	1.15	2.30
4	1971	Defender	2600	1980	Misc	Atari	0.99	0.05	0	0.01	1.05	2.10
5	2671	Boxing	2600	1980	Fighting	Activision	0.72	0.04	0	0.01	0.77	1.54
6	4027	Ice Hockey	2600	1980	Sports	Activision	0.46	0.03	0	0.01	0.49	0.99

Original table

New column

group_by () %>% summarize()

- group data by one or more columns
- compute summary statistics of grouped data

```
#group the data by genre and calculate the sum of global sales for each genre
vg_data_grouped <- group_by(vg_data, Genre) %>% summarize(sum(Global_Sales))
head(vg_data_grouped)
```

The pipe operator, which is used to chain together multiple functions in a single line of code

Result:

Genre	`sum(Global_Sales)`
<chr>	<dbl>
Action	1723.
Adventure	235.
Fighting	444.
Misc	798.
Platform	829.
Puzzle	242.

5. In-class Assignment

For questions related to R:

you will be working with the datasets built-in in R for practice*:

- ***longley***
- ***iris***
- ***AirPassengers***

Use head() function to get an idea what this data set is about.

* For the more examples of datasets built-in in R use function data().

Longley

A macroeconomic data set which contains data on the US economy in the 1947-1962 period

```
> head(longley)
     GNP.deflator     GNP Unemployed Armed.Forces Population Year Employed
1947         83.0 234.289      235.6        159.0    107.608 1947   60.323
1948         88.5 259.426      232.5        145.6    108.632 1948   61.122
1949         88.2 258.054      368.2        161.6    109.773 1949   60.171
1950         89.5 284.599      335.1        165.0    110.929 1950   61.187
1951         96.2 328.975      209.9        309.9    112.075 1951   63.221
1952         98.1 346.999      193.2        359.4    113.270 1952   63.639
```

- **GNP.deflator**: GNP deflator (implicit price deflator for GNP)
- **GNP**: Gross National Product (in millions of dollars)
- **Unemployed**: Number of unemployed (in thousands)
- **Armed.Forces**: Size of armed forces (in thousands)
- **Population**: Population (in thousands)
- **Year**: Year (1947-1962)
- **Employed**: Number of employed (in thousands)

iris

A multivariate dataset which contains measurements for iris flowers from different specie.

```
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

- Sepal.Length: The length of the sepal (in cm).
- Sepal.Width: The width of the sepal (in cm).
- Petal.Length: The length of the petal (in cm).
- Petal.Width: The width of the petal (in cm).
- Species: The species of the iris flower.

AirPassengers

A dataset which contains monthly totals of international airline passengers

```
> head(AirPassengers)
[1] 112 118 132 129 121 135
```

6. Home Assignment

- Create an account in https://public.tableau.com/