Intro To Visualization In R - Exploratory Data Analysis - 1

One should look for what is and not what he thinks should be – Albert Einstein

Exploratory Data Analysis: topic introduction

In this part of the course, we will cover the following concepts:

- Exploratory data analysis use cases
- Perform EDA on data

Warm-up chat question

- Before we begin this module, let's start with a chat question
- Do you have experience making data visualizations? If so, what type?
- What tools do you use to make them?
- Share your responses in the chat

Module completion checklist

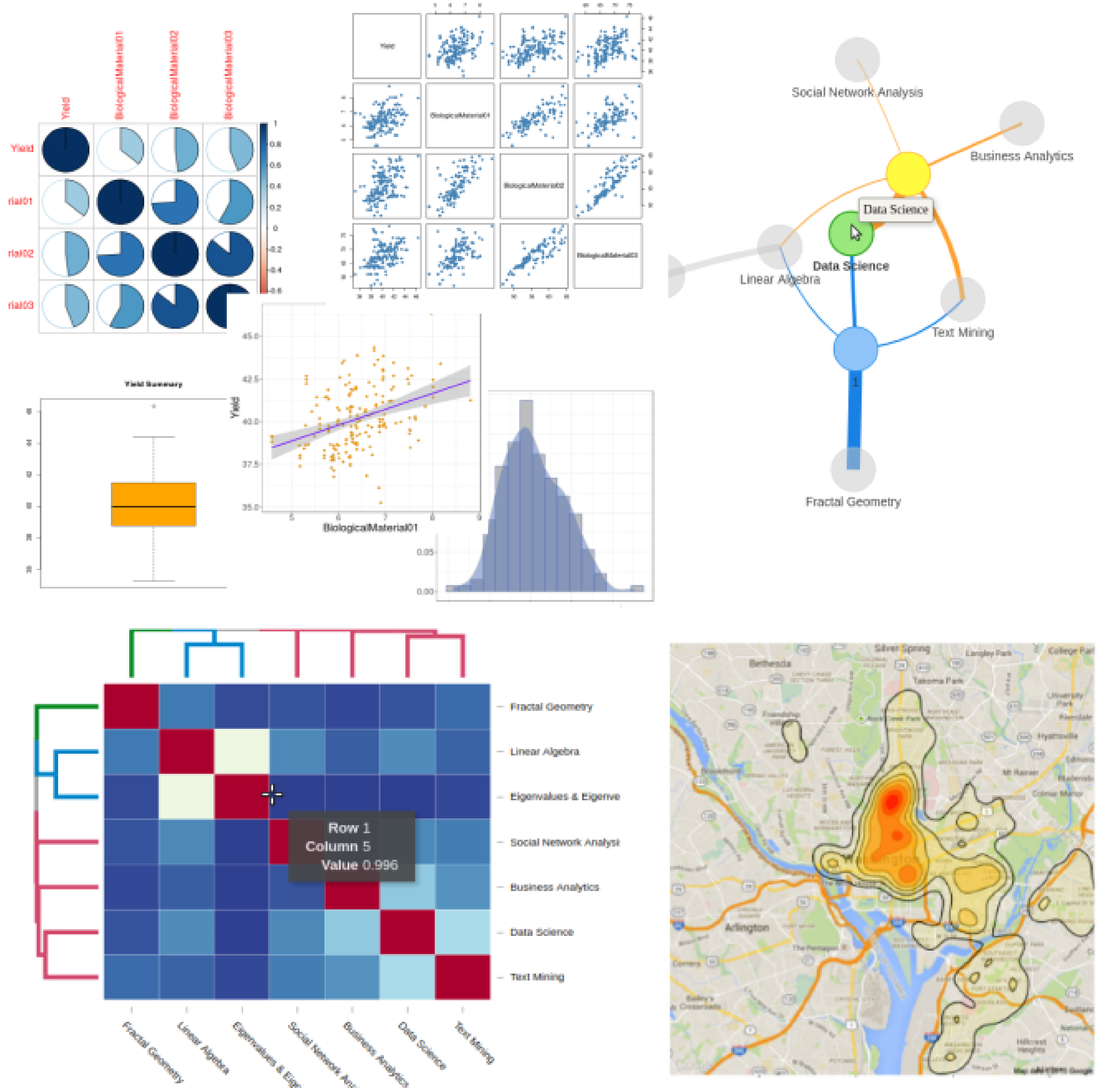
Objective	Complete
Define the exploratory data analysis (EDA) cycle	
Differentiate between static and interactive visualizations	

What is data visualization?

- Data visualizations are representations intended for communicating specific insights about datasets in actionable ways
- They may be created for a target audience, to communicate the relationship between variables or other essential statistics
- When created during data analysis, they serve an **exploratory** purpose for the analyst, helping them to draw conclusions and generate new hypotheses

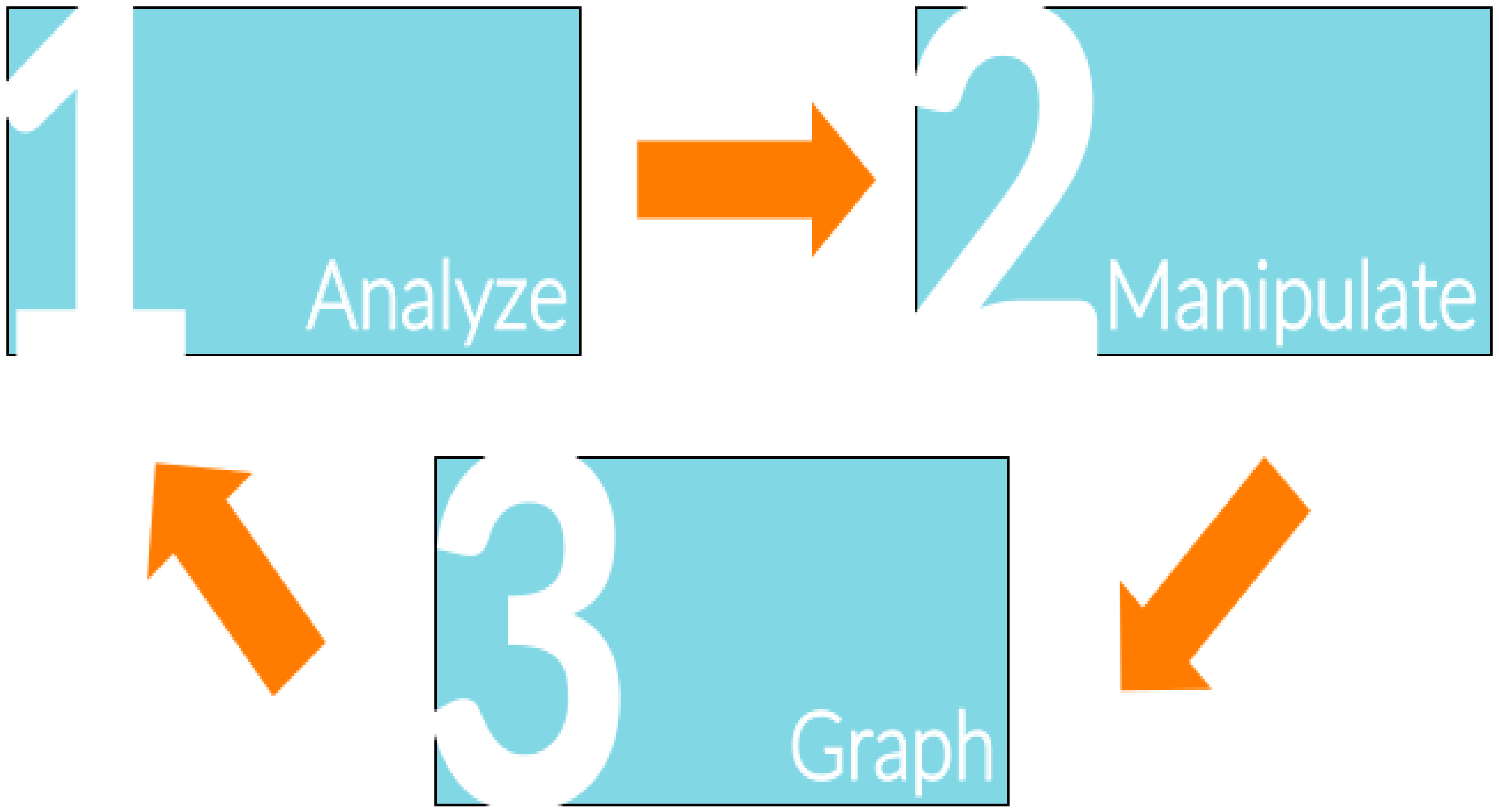
Why visualize data?

- Visualizing data provides insights that are interpretable and relevant
- It visually represents data to see trends, outliers, and patterns
- It helps test hypotheses
- These actions are beneficial in **exploratory data analysis** (EDA)



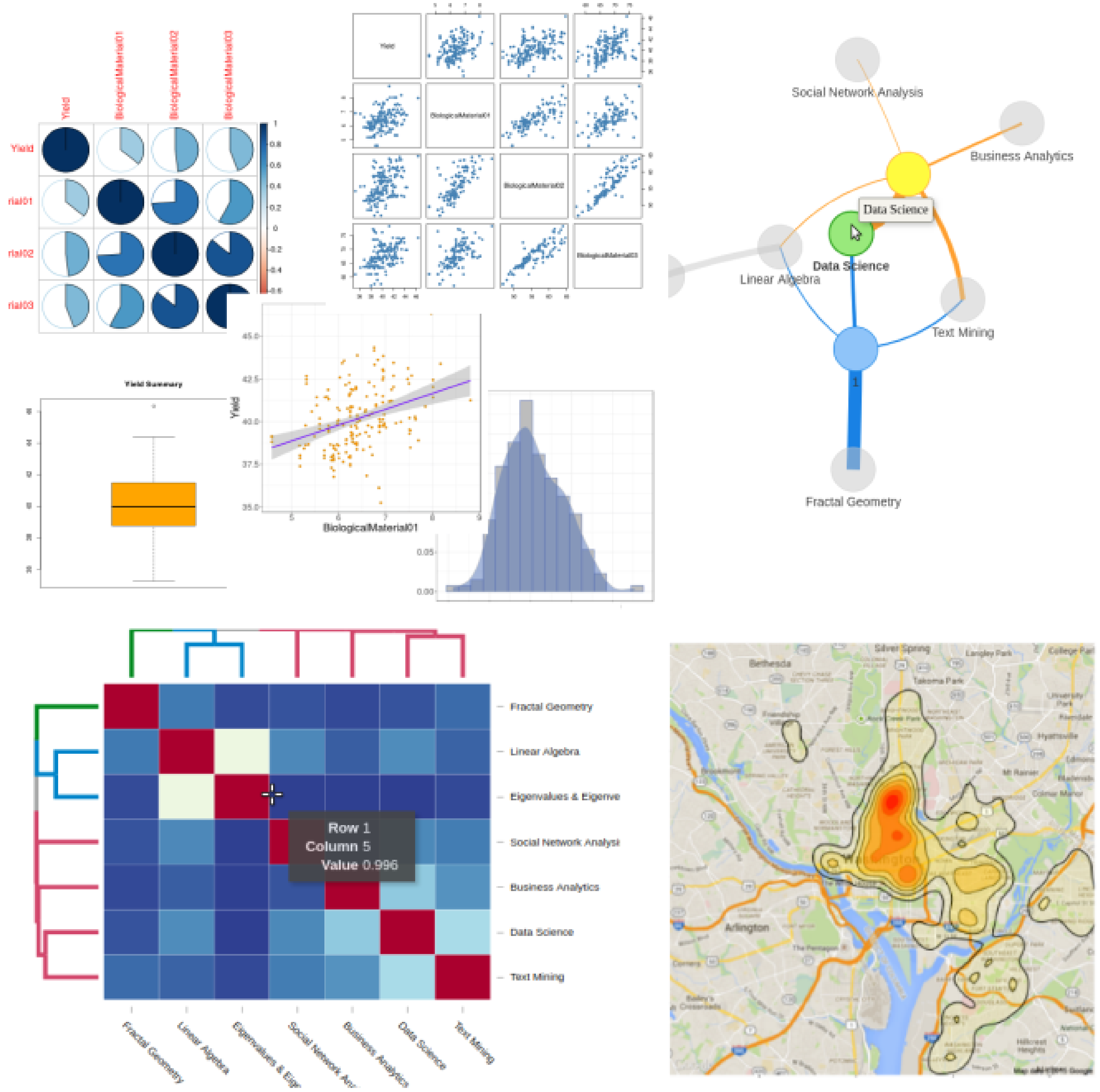
Exploratory data analysis

- **Exploratory Data Analysis** refers to the analysis process used to discover patterns, spot anomalies, test hypotheses, and check assumptions
- Visualization is an **iterative process** and consists of the following steps:
  - Analyze
  - Manipulate
  - Graph
  - Repeat



Types of visualization in R

- We can create several types of visualizations, such as:
  - Basic plots & composite graphs
  - Maps
  - Dynamic visualizations
  - Interactive charts & dashboards
  - 3D graphics
- *Visit the R graph gallery* to see a list of help pages, vignettes, and code demos

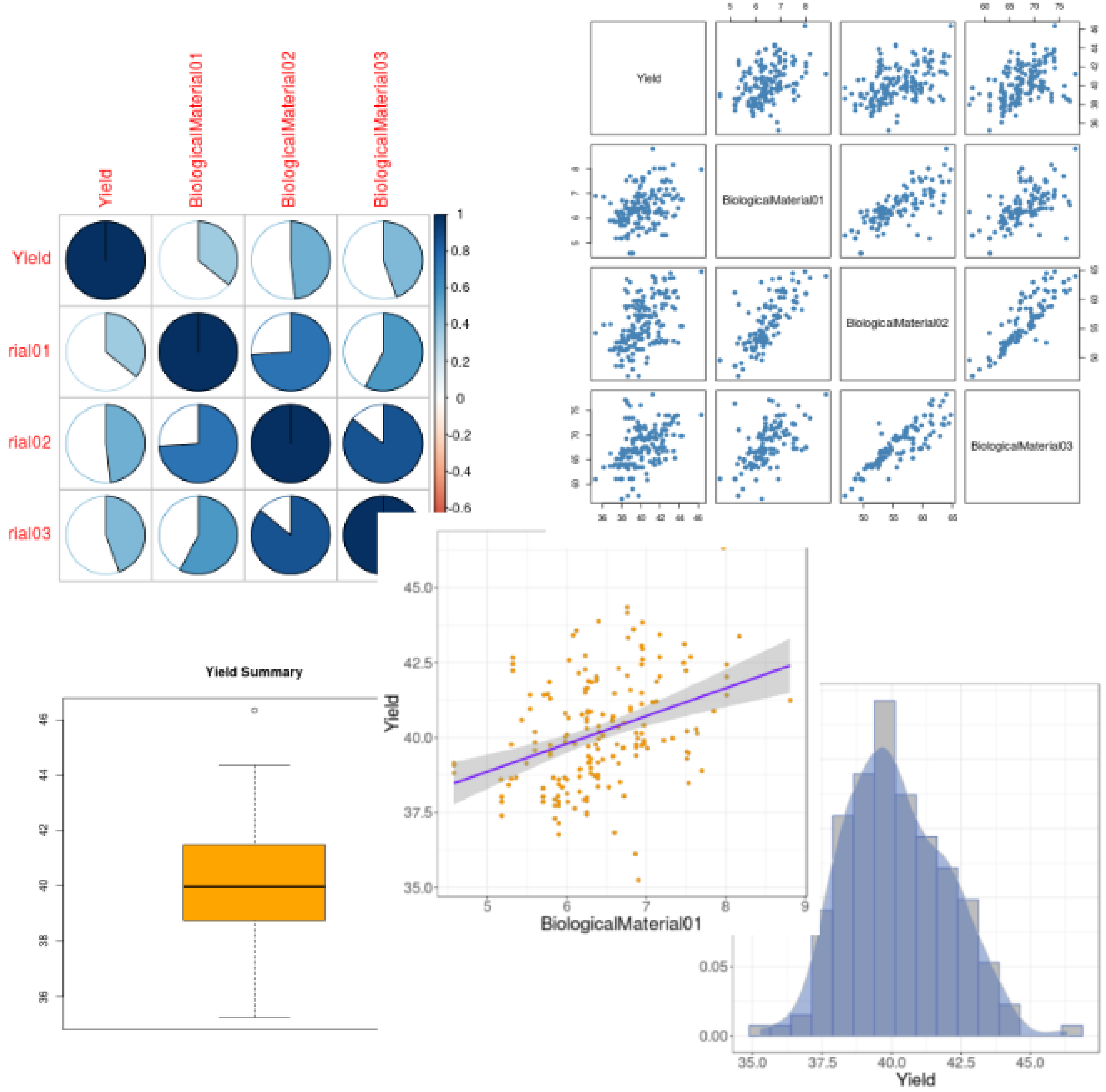


Module completion checklist

Objective	Complete
Define the exploratory data analysis (EDA) cycle	✔
Differentiate between static and interactive visualizations	

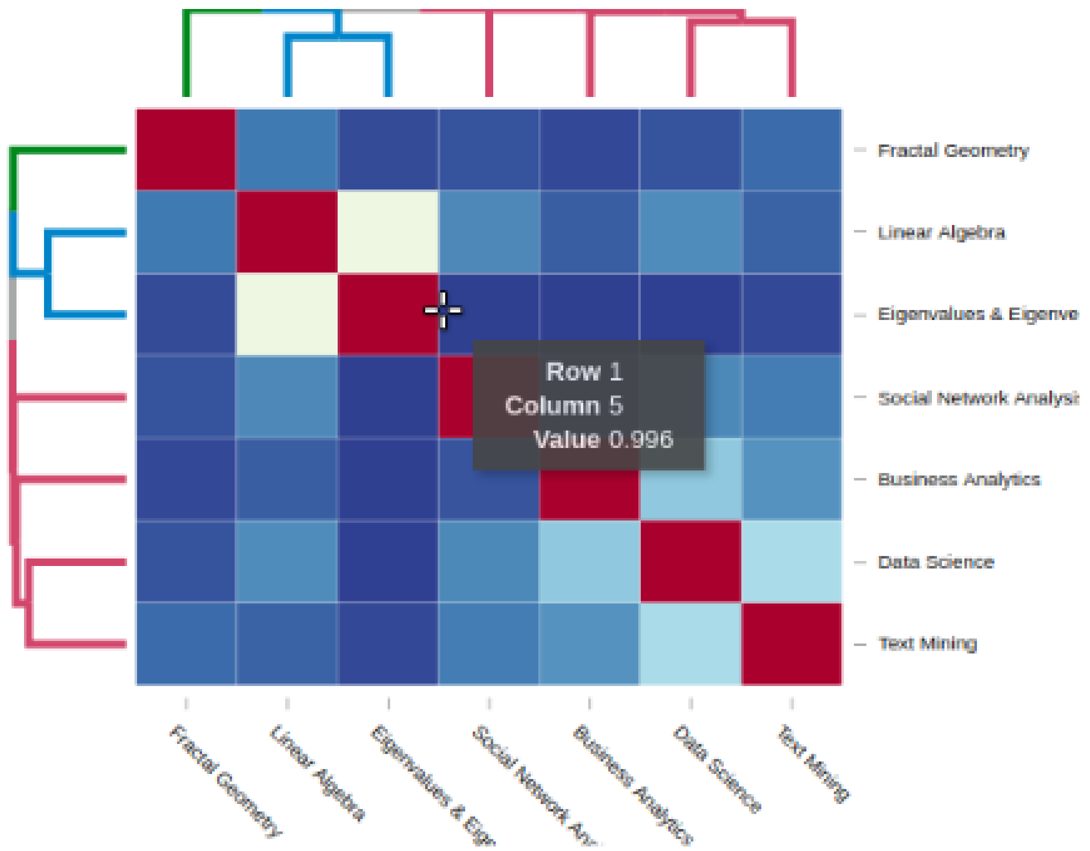
Static visualizations in R

- **Static visualizations** are for the **display** of data only, without interactivity for users
- They are best suited to display patterns in data for **print media**
- Static visualizations can be created using base R and packages like `ggplot2` and `corrplot`



Interactive visualizations in R

- **Interactive visualizations** let users **click, drag, and zoom** through data
- They are best displayed as components of a **web-based media** like websites or apps
- Creating them requires packages like `highcharter`, `plotly`, and `htmlwidgets`



Directory settings

- In order to maximize the efficiency of your workflow, you may want to use the `box` package and encode your directory structure into `variables`

```
install.packages(box)
```

- Let the `main_dir` be the variable corresponding to your materials folder

```
# Set `main_dir` to the location of your materials folder.

path = box::file()
main_dir = dirname(dirname(path))
```

Directory settings (cont'd)

- We will store all datasets in the `data` directory inside the materials folder in your environment, so we'll save its path to a `data_dir` variable
- We will save all of the plots in the `plots` directory corresponding to `plot_dir` variable

- To append a string to another string, use `paste0` command and pass the strings you would like to paste together

```
# Make `data_dir` from the `main_dir` and
# remainder of the path to data directory.
data_dir = paste0(main_dir, "/data")
# Make `plots_dir` from the `main_dir` and
# remainder of the path to plots directory.
plot_dir = paste0(main_dir, "/plots")
```

Case study: stroke survey

- According to the World Health Organization (WHO), stroke is the 2nd leading cause of death globally
- *Click here* for a dataset showing the results of a stroke drug survey clinical trial on a sample of adults in the U.S
- Each row in the data provides relevant information about the adult, including if they had a stroke

Load the dataset for EDA

- Let's load the stroke dataset from the data directory into R's environment

```
# Read CSV file called "healthcare-dataset-stroke-data.csv"
health_data = read.csv(file = file.path(data_dir,"healthcare-dataset-stroke-data.csv"), #<- provide
file path
               header = TRUE,             #<- if file has header set to TRUE
               stringsAsFactors = FALSE) #<- read strings as characters, not as factors
```

`Stroke` Dataset: attribute information

- *id*: unique identifier
- *gender*: “Male”, “Female” or “Other”
- *age*: age of the patient
- *hypertension*: 0 if the patient doesn’t have hypertension, 1 if the patient has hypertension
- *heart_disease*: 0 if the patient doesn’t have any heart diseases, 1 if the patient has a heart disease
- *ever_married*: “No” or “Yes”
- *work_type*: “children”, “Govt_job”, “Never_worked”, “Private” or “Self-employed”
- *Residence_type*: “Rural” or “Urban”
- *avg_glucose_level*: average glucose level in blood
- *bmi*: body mass index
- *smoking_status*: “formerly smoked”, “never smoked”, “smokes” or “Unknown”*
- *stroke*: 1 if the patient had a stroke or 0 if not

View data types

- Let's examine the data types of the columns in the dataset and handle the missing data, if there are any

```
str(health_data)
```

```
'data.frame':	5110 obs. of  12 variables:
 $ id               : int  9046 51676 31112 60182 1665 56669 53882 10434 27419 60491 ...
 $ gender           : chr  "Male" "Female" "Male" "Female" ...
 $ age              : num  67 61 80 49 79 81 74 69 59 78 ...
 $ hypertension     : int  0 0 0 0 1 0 1 0 0 0 ...
 $ heart_disease    : int  1 0 1 0 0 0 1 0 0 0 ...
 $ ever_married     : chr  "Yes" "Yes" "Yes" "Yes" ...
 $ work_type        : chr  "Private" "Self-employed" "Private" "Private" ...
 $ Residence_type   : chr  "Urban" "Rural" "Rural" "Urban" ...
 $ avg_glucose_level: num  229 202 106 171 174 ...
 $ bmi              : num  36.6 NA 32.5 34.4 24 29 27.4 22.8 NA 24.2 ...
 $ smoking_status   : chr  "formerly smoked" "never smoked" "never smoked" "smokes" ...
 $ stroke           : int  1 1 1 1 1 1 1 1 1 1 ...
```

Impute missing data

- We will now impute missing values in the `bmi` column with the mean

```
# Convert BMI to numeric
health_data$bmi <- as.numeric(health_data$bmi)
# Replace N/A's in BMI column with mean
health_data$bmi[is.na(health_data$bmi)] <- mean(health_data$bmi,na.rm=TRUE)
# Display data
str(health_data)
```

Subsetting data

- For visualization, let's restructure our data by taking a **subset** of the data with all the observations of the following variables:
  - `age`
  - `avg_glucose_level`
  - `bmi`

```
health_subset <- health_data[, c("age", "avg_glucose_level", "bmi")]
str(health_subset)
```

```
'data.frame':	5110 obs. of  3 variables:
 $ age              : num  67 61 80 49 79 81 74 69 59 78 ...
 $ avg_glucose_level: num  229 202 106 171 174 ...
 $ bmi              : num  36.6 28.9 32.5 34.4 24 ...
```

Correlation between variables

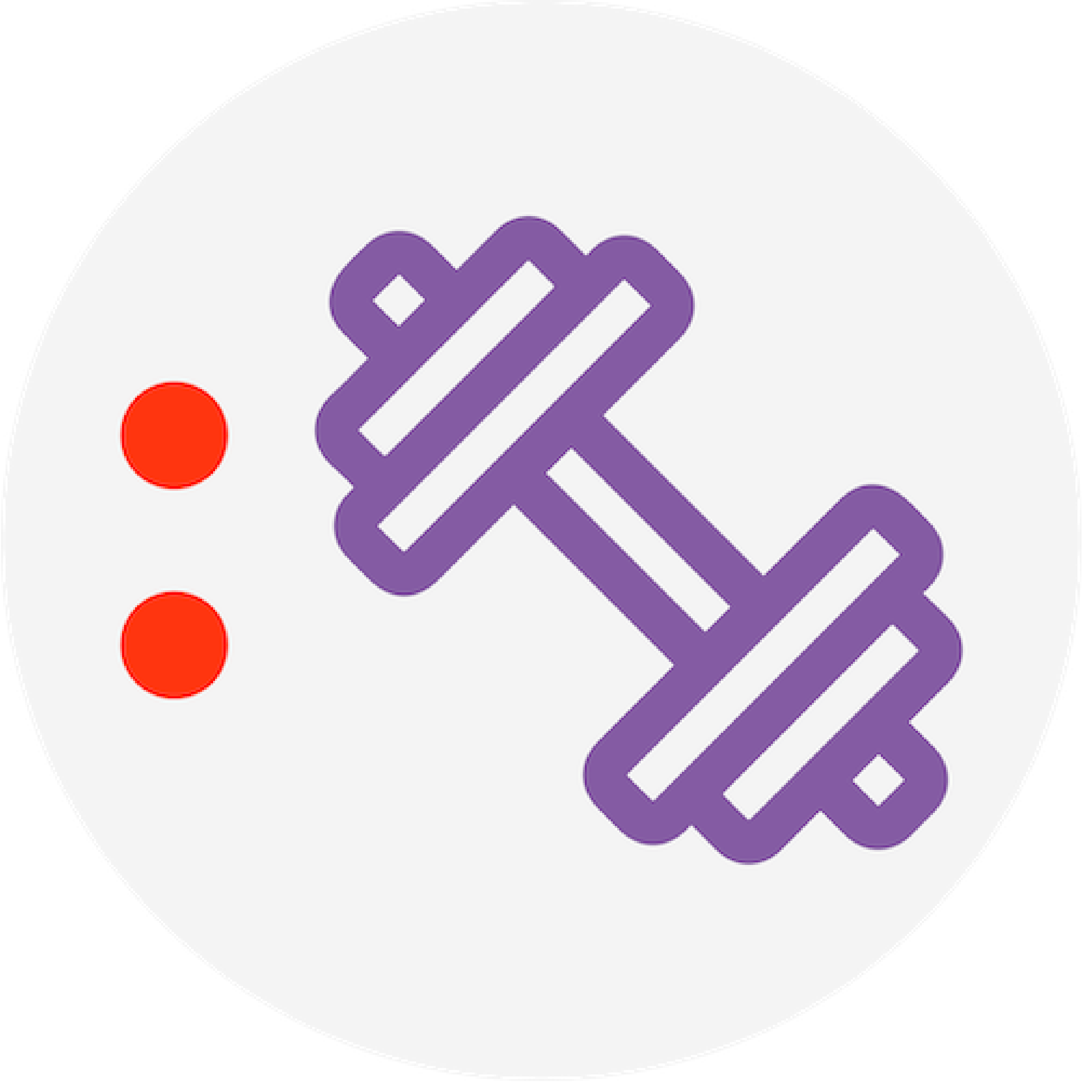
- Let's visualize the relationship between the variables with a **correlation matrix**
  - Each value in the matrix is a **correlation coefficient**, which is a value between $[-1, 1]$
  - The matrix is **square**, as the number of rows is the same as the number of columns
  - The matrix is **symmetric**, as the values on opposite sides of the diagonal are mirrored $value_{row_i,col_j} = value_{row_j,col_i}$
  - Values on the **diagonal** are equal to 1

```
# Compute a correlation matrix of 3 variables using `cor` function.
health_cor = cor(health_subset)
health_cor
```

```
                        age avg_glucose_level       bmi
age               1.0000000         0.2381711 0.3259425
avg_glucose_level 0.2381711         1.0000000 0.1687514
bmi               0.3259425         0.1687514 1.0000000
```

Knowledge check

Exercise



You are now ready to try tasks 1-2 in the exercise for this topic

Module completion checklist

Objective	Complete
Define the exploratory data analysis (EDA) cycle	✔
Differentiate between static and interactive visualizations	✔

Exploratory Data Analysis: Topic Summary

In this part of the course, we have covered the following concepts:

- Defining Exploratory Data Analysis
- Performing EDA on data

Congratulations on completing this module!