

Introduction to Data Science Topic-4

- Instructor: Professor Henry Horng-Shing Lu,
Institute of Statistics, National Yang Ming Chiao Tung University, Taiwan
Email: henryhslu@nycu.edu.tw
- WWW: <http://misg.stat.nctu.edu.tw/hslu/course/DataScience.htm>
- Classroom: ED B27 (新竹市大學路1001號工程四館B27教室)
- References:
M. A. Pathak, Beginning Data Science with R, 2014, Springer-Verlag.
K.-T. Tsai, Machine Learning for Knowledge Discovery with R: Methodologies
for Modeling, Inference, and Prediction, 2021, Chapman and Hall/CRC.
- Evaluation: Homework: 70%, Term Project: 30%
- Office hours: By appointment

Course Outline

10 Topics and 10 Homeworks:

- Introduction of Data Science
- Introduction of R and Python
- Cleaning Data into R and Python
- **Data Visualization**
- Exploratory Data Analysis
- Regression (Supervised Learning)
- Classification (Supervised Learning)
- Text Mining
- Clustering (Unsupervised Learning)
- Neural Network and Deep Learning

Data Visualization with R

References:

Ch. 3, M. A. Pathak, Beginning Data Science with R, 2014, Springer-Verlag.

<https://www.kaggle.com/code/dongdongxzoez/r-topic-4/notebook>

<https://www.kaggle.com/learn>



why do we need data visualization?

“A picture is worth a thousand words”

Its eyes are round, with a triangular mouth under the little nose, a beautiful "eight" character on either side of the mouth, and two pointed ears erected on the round head, making it particularly airy.



<https://en.wiktionary.org/wiki/cat>



In the context of data visualization, the saying "A picture is worth a thousand words" emphasizes the idea that visual representations of data can convey a wealth of information and insights more effectively than a lengthy written or numerical description. Here's how it applies to data visualization:



1. **Efficiency of Communication:** Data visualizations, such as charts, graphs, and diagrams, can summarize complex datasets and trends in a concise and easily understandable manner. Instead of sifting through rows and columns of data, viewers can quickly grasp key points from a well-designed visualization.
2. **Clarity and Understanding:** Visualizations make it easier for people to understand the relationships and patterns within data. When data is presented visually, trends, anomalies, and comparisons become more apparent, reducing the need for lengthy explanations.
3. **Memory and Retention:** Visual information tends to be more memorable than textual or numerical data. People are more likely to remember and recall insights from a compelling chart or graph than from a lengthy report or spreadsheet.
4. **Universal Language:** Visualizations can transcend language barriers, making data accessible to a global audience. A well-designed chart can communicate information to individuals regardless of their language proficiency.
5. **Engagement:** Visualizations engage viewers and draw their attention to key points. They encourage exploration and interaction with the data, allowing users to dive deeper into specific aspects of the information presented.

6. **Decision-Making:** Visualizations support informed decision-making. Decision-makers can quickly assess the situation, identify trends, and make choices based on the visual representation of data.
7. **Storytelling:** Visualizations can be used to tell a compelling story about the data. They guide viewers through a narrative, highlighting important points and making the data more relatable and persuasive.

In essence, in the realm of data visualization, the saying emphasizes that a well-crafted chart or graph can convey complex data and insights in a way that is efficient, clear, and memorable. It underscores the value of using visuals to make data more accessible and actionable for a broader audience, from analysts and experts to non-technical stakeholders.



Data visualization is essential for several reasons, as it offers numerous benefits and advantages in various fields and industries:



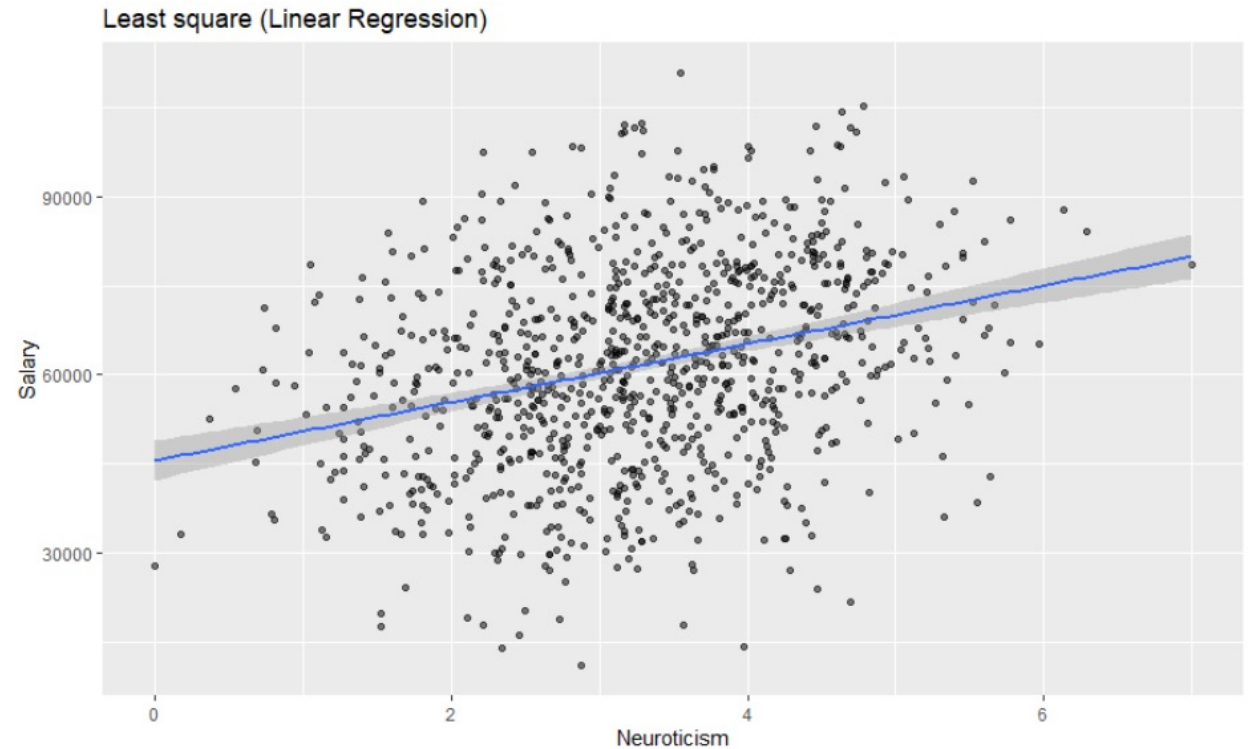
1. **Enhanced Understanding:** Human brains are highly adept at processing visual information. Data visualization makes complex data more understandable and accessible to a wider audience. Patterns, trends, and relationships in data become more apparent when presented visually.
2. **Insight Discovery:** By representing data graphically, it becomes easier to spot insights, outliers, and anomalies in the data. Visualizations can reveal hidden patterns and correlations that may not be apparent through raw data alone.
3. **Effective Communication:** Visualizations provide a powerful means of communicating data-driven insights. They enable data analysts, scientists, and decision-makers to convey their findings more effectively to colleagues, stakeholders, and the public.
4. **Data Exploration:** Interactive data visualizations allow users to explore data in a more interactive and intuitive manner. Users can drill down into details, filter data, and gain a deeper understanding of the information presented.
5. **Decision Making:** Data visualizations help in making informed decisions. Whether in business, healthcare, finance, or any other field, clear and visual data representations enable better decision-making processes by providing relevant information at a glance.

6. **Identifying Trends and Patterns:** Visualizations make it easier to identify trends over time or across different data dimensions. For example, line charts can show how a variable changes over time, helping in trend analysis.
7. **Comparisons:** Visualizations allow for easy comparisons between different data points or categories. Bar charts, pie charts, and scatter plots, for instance, enable quick comparisons between data elements.
8. **Storytelling:** Data visualizations can be used to tell a compelling data-driven story. They engage the audience and guide them through the data, helping them understand the narrative and the key takeaways.
9. **Error Detection:** Visualizations can also help in identifying data errors or inconsistencies. When data is visualized, unusual or incorrect data points can stand out more clearly.
10. **Efficiency:** Data visualization tools and software automate the process of creating charts and graphs, saving time and effort in data analysis and reporting.
11. **Predictive Analytics:** Data visualizations can aid in predictive modeling by visually representing historical data and model predictions. This is valuable in forecasting future trends and making proactive decisions.

In summary, data visualization is a crucial tool for anyone working with data. It simplifies complex data, supports decision-making, and improves communication, ultimately leading to better insights and more effective actions in various domains.

What's matter

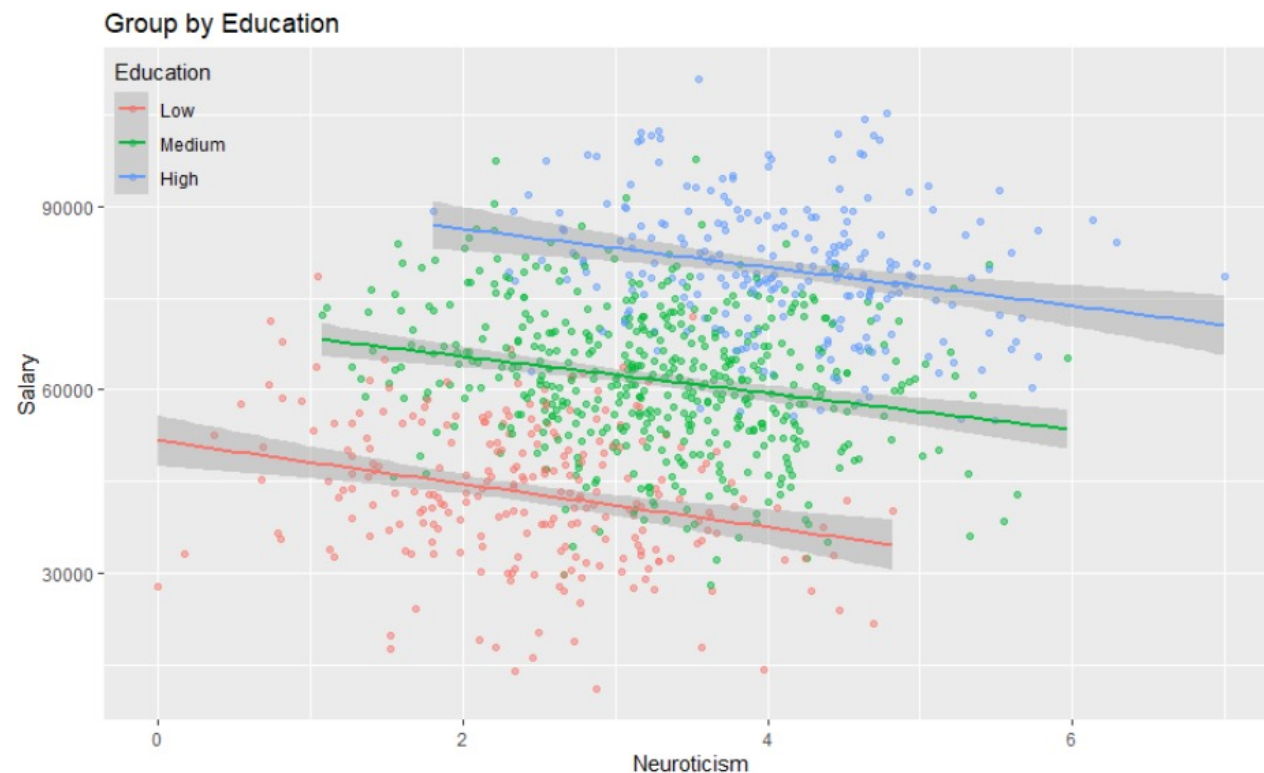
- Suppose we collected the salary data of employees from a company, and one of the variables is Neuroticism. At first glance, the more exaggerated the neuroticism, the higher the salary, but this does not conform to our cognition.



<https://paulvanderlaken.com/2017/09/27/simpsons-paradox-two-hr-examples-with-r-code/>

Simpson's paradox

- This is a very famous phenomenon - Simpson's paradox, because not only does Neuroticism influence salary, but other variables may be too. If we look at the education level of employees and divide them into three levels, we can find that salary is negatively correlated to neuroticism. In fact, the level of education determines the level of salary, which is why data visualization is very important!!



KS

what is Simpson's paradox?



Simpson's paradox, also known as the Yule-Simpson effect, is a statistical phenomenon in which a trend or relationship that appears in different groups of data disappears or even reverses when these groups are combined. In other words, when you analyze data separately for different subgroups, you might observe one trend, but when you combine those subgroups, the overall trend can be different.



This paradox can occur when there is a confounding variable, meaning a variable that affects both the independent and dependent variables in a study. When you don't account for the influence of this confounding variable, it can lead to misleading or counterintuitive results.

Here's a simplified example to illustrate Simpson's paradox:

Let's consider a hypothetical example to illustrate this paradox:

Example: Consider two hospitals, Hospital A and Hospital B. You want to determine which hospital has a better overall success rate for surgeries.

- **Hospital A:**

- Simple surgeries: 99 successes out of 100 surgeries (99% success rate)
- Complex surgeries: 1 success out of 10 surgeries (10% success rate)

- **Hospital B:**

- Simple surgeries: 490 successes out of 500 surgeries (98% success rate)
- Complex surgeries: 9 successes out of 10 surgeries (90% success rate)


If you look at each type of surgery separately, Hospital B has a lower success rate for simple surgeries and a higher success rate for complex surgeries compared to Hospital A.

However, if you combine the surgeries to look at overall success rates:

- **Hospital A:** 100 successes out of 110 surgeries (90.9% success rate)
- **Hospital B:** 499 successes out of 510 surgeries (97.8% success rate)

When aggregated, Hospital B appears to have a higher success rate overall, even though it was outperformed by Hospital A in each individual category.

This paradox can occur due to underlying factors or different group sizes. It serves as a cautionary tale about the importance of examining data at multiple levels before drawing conclusions.

 Regenerate



R basis plot function

Usage	Function	Argument
Scatter plot	plot(x, y)	(vec, vec)
Histogram	hist()	(vec)
Bar plot	barplot()	(vec or matrix)
Box Plot	boxplot()	(matrix or data)
KDE plot	density()	(vec)
Curve	curve()	(f(x), start, end)
QQ-plot	qqnorm()	(vec)
3D density	persp()	(vec, vec, matrix)
Contour and heat map	image()	(vec, vec, matrix)
Pie chart	pie()	(vec)
heatmap	heatmap()	(matrix)



In R, there are several basic plotting functions that are part of the base graphics system.



These functions allow you to create various types of plots and visualizations. Here's a list of some of the basic plotting functions in R:

1. **plot()**: This is a general-purpose function for creating scatter plots, line plots, and other types of basic plots.
2. **barplot()**: Used for creating vertical or horizontal bar charts.
3. **hist()**: Generates histograms to visualize the distribution of numeric data.
4. **boxplot()**: Creates box-and-whisker plots to display the distribution and spread of data.
5. **pie()**: Generates pie charts to represent the composition of categorical data.
6. **dotchart()**: Creates dot plots or Cleveland dot plots, which are used for visualizing small datasets.
7. **stripchart()**: Creates strip charts, also known as dot plots, for one-dimensional data.
8. **pairs()**: Generates a matrix of scatterplots for exploring pairwise relationships in multivariate data.
9. **heatmap()**: Produces heatmaps to visualize matrices or tables of data.
10. **mosaicplot()**: Creates mosaic plots, which are used for visualizing contingency tables.
11. **contour()**: Generates contour plots to visualize three-dimensional surfaces.
12. **image()**: Creates images and raster plots from two-dimensional data.
13. **persp()**: Used to create 3D perspective plots.
14. **plotly::plot_ly()**: While not a base graphics function, the `plot_ly()` function from the Plotly package allows for interactive and web-based data visualizations.

Low-level command for data visualization in R?

- `Lines()`: add a line
- `abline()`: add straight line
- `Points()`: add a point
- `Legend()`: add legend
- `Axes()`: change axis
- `Text()`: add text
- `Par()`: combine plots



In R, low-level graphics commands allow you to have fine-grained control over the appearance and layout of your plots. These commands give you more control compared to high-level functions like `plot()` or `hist()`. Low-level commands are typically used when you need to customize your plots extensively. Here are some common low-level graphics commands in R:



1. **plot.new()**: This function opens a new plotting window, allowing you to start a new plot from scratch.
2. **plot.window()**: It defines the plotting area within a plotting window, specifying the range of x and y values. You can set the limits of the plot using this function.
3. **points()**: Adds individual data points (e.g., scatter plot points) to an existing plot.
4. **lines()**: Draws lines between points on an existing plot. This is useful for creating line plots.
5. **abline()**: Adds lines to a plot, including horizontal, vertical, or diagonal lines. You can specify the intercept and slope of the line.
6. **segments()**: Draws line segments between specified endpoints on a plot.
7. **rect()**: Draws rectangles on the plot, often used for highlighting regions of interest.
8. **polygon()**: Draws polygons on the plot, which can be used for creating filled areas or irregular shapes.

8. **polygon()**: Draws polygons on the plot, which can be used for creating filled areas or irregular shapes.
9. **text()**: Adds text labels to the plot at specified coordinates.
10. **mtext()**: Adds text to the margins of the plot, allowing you to label axes or add titles to specific margins.
11. **axis()**: Customizes the axis labels, tick marks, and other axis properties.
12. **legend()**: Manually creates legends or keys for the plot to explain symbols or colors used.
13. **par()**: The `par()` function allows you to set and query graphical parameters, such as text size, line type, and more. It provides extensive control over the plot's appearance.
14. **lines()**: Draws lines on the plot, often used for creating line charts or connecting data points.
15. **symbols()**: Plots symbols at specified locations, allowing you to use different shapes and sizes for each point.

Low-level commands are particularly useful when you need to create highly customized or complex visualizations that cannot be easily achieved with high-level functions. They give you full control over the details of your plots, but they also require more manual configuration and customization.

Iris Data

The Iris Dataset

Collected by Ronald
Fisher in 1936



Iris Data Set:(iris) contains

- Sepal Length (mm)
- Sepal Width (mm)
- Petal Length (mm)
- Petal Width (mm)
- Species

iris setosa



petal

sepal

iris versicolor



petal

sepal

iris virginica



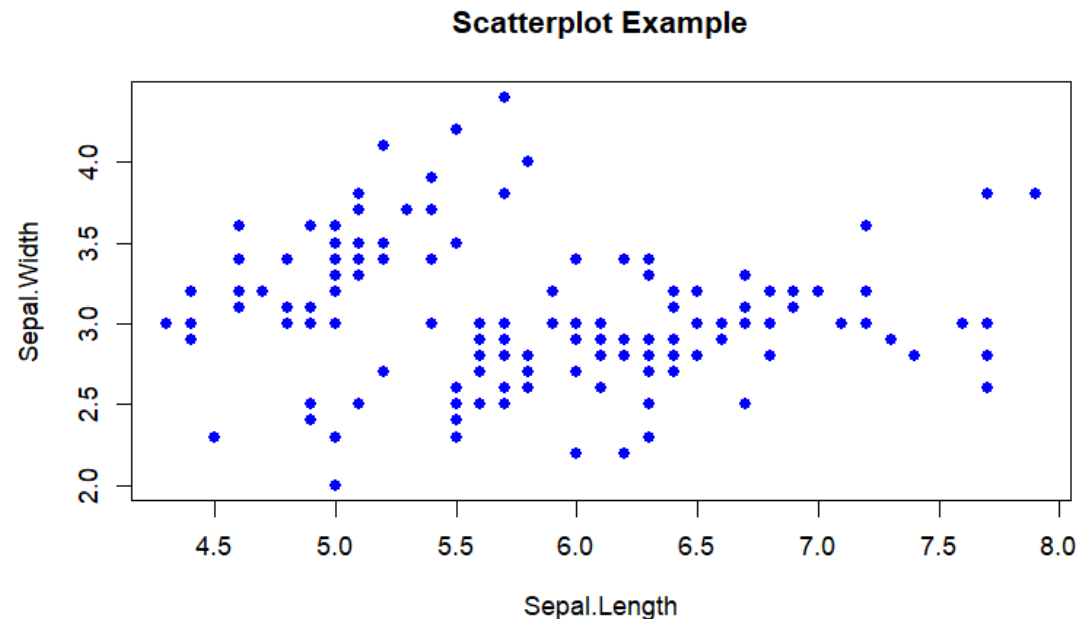
petal

sepal

Scatter plot

- A scatter plot is a type of data visualization that is used to display individual data points in a two-dimensional space, typically with one variable on the x-axis and another variable on the y-axis. Scatter plots are particularly useful for visualizing the relationship or correlation between two continuous variables.

```
{r}  
# Load the iris dataset  
data("iris")  
  
# Create the scatterplot and add labels  
plot(iris$Sepal.Length, iris$Sepal.Width, main = "Scatterplot Example", xlab = "Sepal.Length", ylab = "Sepal.Width", pch = 19, col="blue")
```





Scatter plots are a valuable tool in data visualization and analysis, and they serve several important purposes:



1. **Visualizing Relationships:** Scatter plots allow you to visualize the relationship between two continuous variables. You can quickly see how changes in one variable relate to changes in another, helping to identify patterns or trends.
2. **Detecting Correlations:** Scatter plots help in assessing the strength and direction of correlations between variables. If points tend to form a clear pattern, it suggests a correlation. A positive correlation means that as one variable increases, the other tends to increase, while a negative correlation indicates that as one variable increases, the other tends to decrease.
3. **Identifying Outliers:** Outliers, which are data points that deviate significantly from the general pattern of the data, can be easily identified in a scatter plot. They appear as data points that are far away from the main cluster of points.
4. **Checking for Linearity:** Scatter plots are used to determine whether a linear relationship exists between two variables. A linear relationship implies that the data points roughly follow a straight line pattern on the plot.
5. **Assessing Clusters or Groups:** Scatter plots can reveal the presence of clusters or distinct groups within the data. Clusters may indicate different subpopulations or categories within the dataset.
6. **Prediction:** In regression analysis, scatter plots are used to visualize how well a regression model fits the data. You can overlay the fitted regression line on the scatter plot to assess the model's accuracy in predicting one variable based on another.

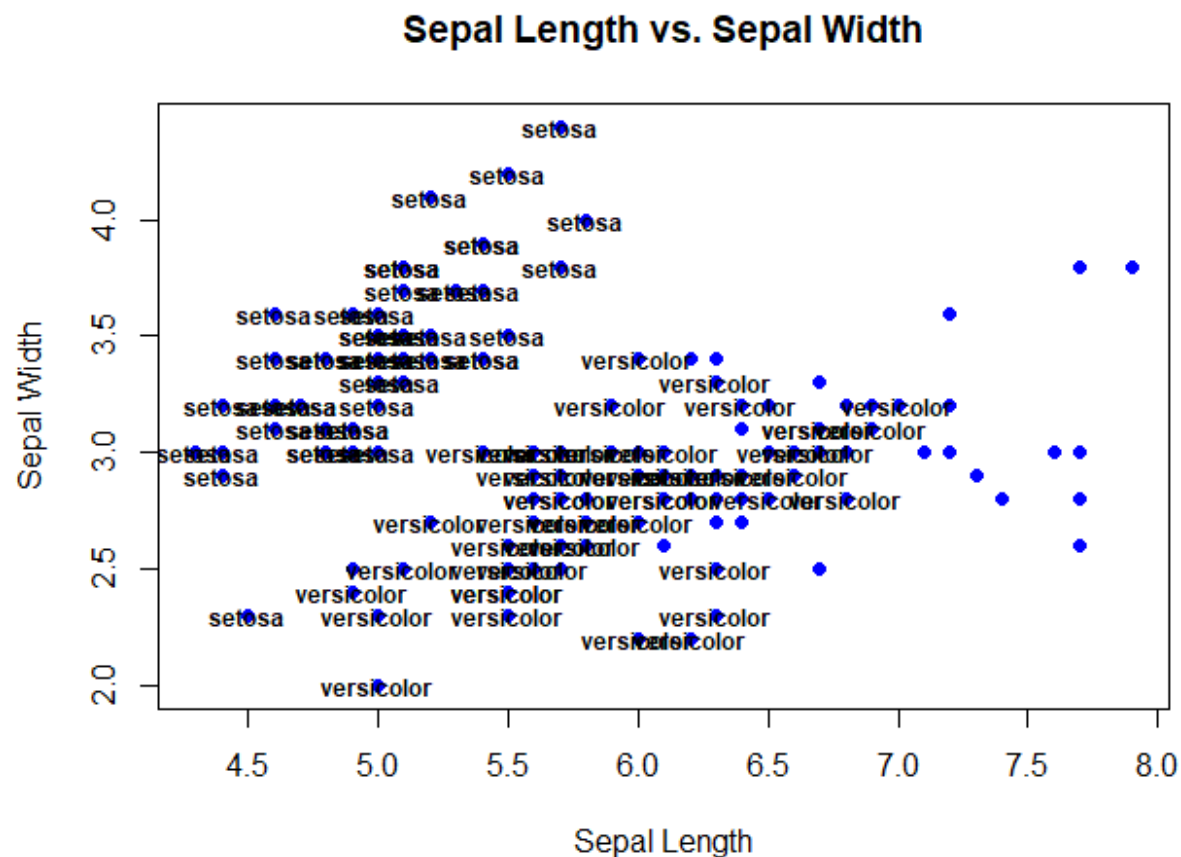
7. **Comparing Datasets:** Scatter plots are useful for comparing two or more datasets to see if they exhibit similar patterns or trends. Overlapping or distinct scatter plots can provide insights into similarities or differences between datasets.
8. **Data Exploration:** Scatter plots are an essential tool for exploring and understanding data before performing more advanced statistical analyses. They help in formulating hypotheses and making decisions about subsequent analyses.
9. **Communication:** Scatter plots are effective for communicating findings to others, such as colleagues or stakeholders. They provide a clear visual representation of data relationships that can be easily understood.

Overall, scatter plots are versatile and widely used in data analysis, statistics, and various fields like science, engineering, finance, and social sciences. They are a foundational tool for understanding and interpreting data patterns, relationships, and outliers.

Label Data Points in the Scatter Plot

```
## You can use the text() function to add labels or annotations to specific data points on the scatter plot.
##{r}
# Create a scatter plot
plot(iris$Sepal.Length, iris$Sepal.Width, main = "Sepal Length vs. Sepal Width", xlab = "Sepal Length", ylab = "Sepal Width",
     pch = 19, col = "blue")

# Add labels for the first 5 data points
text(iris$Sepal.Length[1:100], iris$Sepal.Width[1:100], labels = iris$Species[1:100], cex = 0.75, font = 2) #cex =character
expansion
##
```





Labeling data points in a scatter plot is important for several reasons:

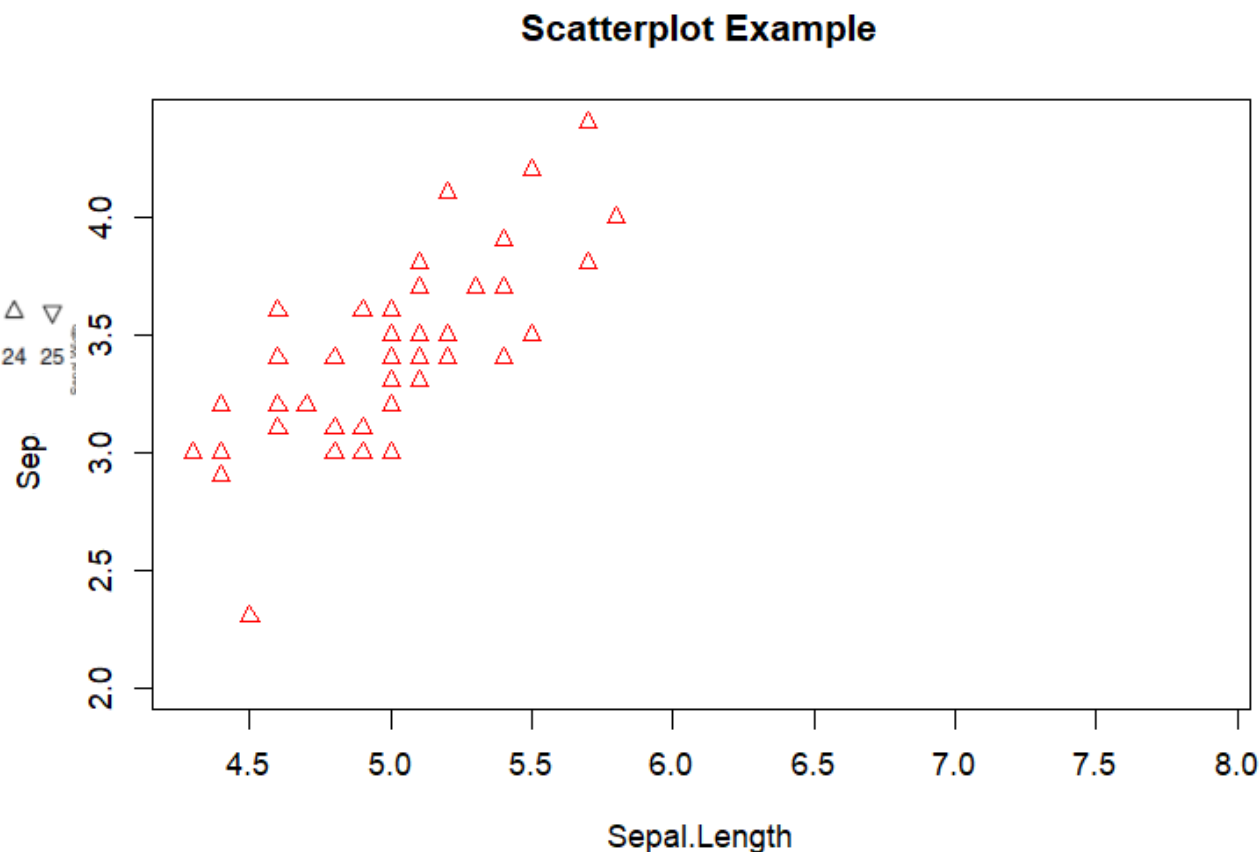


1. **Data Identification:** Labels allow you to identify specific data points within the plot. This is especially valuable when you have a large number of data points, and you want to pinpoint and understand individual observations.
2. **Contextual Information:** Labels provide context to the data points. They can include information such as names, categories, or identifiers associated with each point. For example, in a scatter plot of exam scores, labeling data points with student names or IDs makes it clear which student achieved which score.
3. **Highlighting Key Observations:** Labels help you highlight and draw attention to specific data points of interest. This can be important when you want to emphasize outliers, anomalies, or significant data points within your dataset.
4. **Enhanced Interpretation:** Labels make it easier to interpret the data. They provide additional information about what each data point represents, making it clear to viewers what the data means.
5. **Communication:** When sharing scatter plots with others, labels make the plot more informative and understandable. Viewers can quickly grasp the meaning of each data point and its relevance to the overall analysis.
6. **Quality Control:** In some cases, labeling data points can be part of a quality control process. You can use labels to verify that data points are correctly plotted in the scatter plot, helping to identify any data entry errors or inconsistencies.
7. **Interactive Exploration:** In interactive data visualization tools or applications, labeling data points can enable users to interactively explore the data. Users can hover over or click on points to access additional information provided by the labels.

Points and Lines in the Scatter Plot

```
```{r}
s1 = which(iris$Species == "setosa")
plot(iris$Sepal.Length, iris$Sepal.Width, main = "Scatterplot Example", xlab = "Sepal.Length", ylab = "Sepal.Width",
type = "n")
points(iris$Sepal.Length[s1], iris$Sepal.Width[s1], pch = 2,col="red")
```
```

o △ + × ◇ ▽ ▣ * ◇ ⊕ ⊗ ⊛ ⊞ ⊠ ⊡ ⊢ ⊣ ⊤ ⊥ ⊦ ⊧ ⊨ ⊩ ⊪ ⊫ ⊬ ⊭ ⊮ ⊯ ⊰ ⊱ ⊲ ⊳ ⊴ ⊵ ⊶ ⊷ ⊸ ⊹ ⊺ ⊻ ⊼ ⊽ ⊾ ⊿ ⊿
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25





In a scatter plot, you can control the appearance of points and lines to customize the visualization according to your needs. Here are some common options for configuring points and lines in a scatter plot:

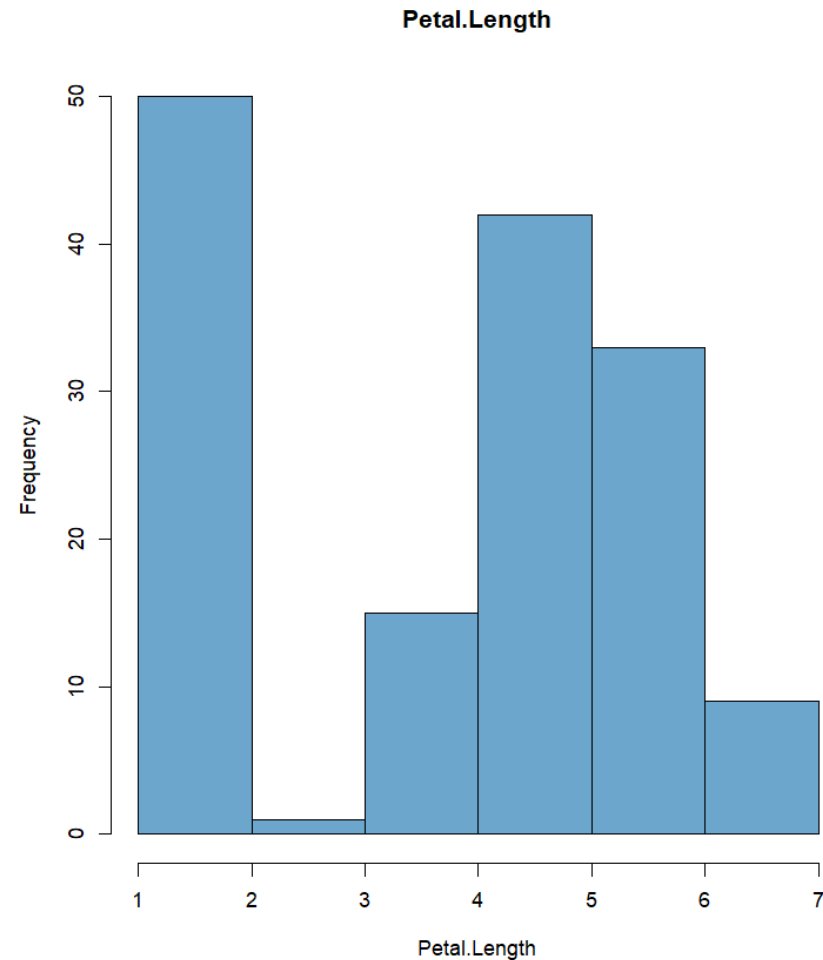


1. **Point Type (``pch``):** You can specify the type of symbol used for data points using the ``pch`` argument. Different values of ``pch`` represent various point symbols. For example:
 - * ``pch` = 1`: Solid circle (default)
 - * ``pch` = 2`: Open circle
 - * ``pch` = 3`: Cross
 - * ``pch` = 4`: X
 - * ``pch` = 5`: Diamond
 - * ...You can choose a value that represents the symbol you prefer for your data points.
2. **Point Size (``cex``):** The ``cex`` argument controls the size of data points. A ``cex`` value of 1 represents the default size, while values less than 1 make the points smaller, and values greater than 1 make them larger. For example, ``cex` = 0.5` makes the points half the default size, and ``cex` = 2` makes them twice as large.
3. **Point Color (``col``):** You can specify the color of data points using the ``col`` argument. You can use color names (e.g., "red," "blue") or numeric values representing colors. For instance, ``col` = "red"` sets the point color to red.
4. **Point Transparency (``alpha`` or ``bg``):** To control the transparency or fill color of data points, you can use the ``alpha`` argument (requires the ``scales`` package) or the ``bg`` argument. For example, ``alpha` = 0.5` makes the points semi-transparent, and ``bg` = "yellow"` sets the background color of the points to yellow.

[🔄 Regenerate](#)

Histogram

```
##{r}  
##### histogram  
hist(iris$Petal.Length, main='Histogram of Petal.Length', xlab='Petal.Length', col='skyblue3',  
      breaks=5) # try breaks = 15 or a vector
```





A histogram is a graphical representation of the distribution of a dataset. It is a type of data visualization that displays the frequency or count of data points within predefined intervals or bins. Histograms are widely used in data analysis and statistics for several purposes:



1. **Understanding Data Distribution:** Histograms provide a visual summary of the distribution of data. They show how data is spread across different values or ranges, allowing you to quickly assess whether data is symmetric, skewed, bimodal, or exhibits other patterns.
2. **Identifying Central Tendency:** Histograms help in identifying measures of central tendency such as the mean, median, and mode. The peak (mode) of a histogram often corresponds to the central value of the dataset.
3. **Detecting Skewness:** Skewness in data, where the distribution is asymmetric, can be visually detected in a histogram. Positive skewness means the tail of the distribution extends to the right, while negative skewness means the tail extends to the left.
4. **Identifying Outliers:** Outliers, which are extreme or unusual data points, can be seen as individual bars that are far from the bulk of the data in a histogram. This makes it easier to spot and investigate outliers.
5. **Choosing Appropriate Statistical Tests:** Histograms help in selecting appropriate statistical tests and models for data analysis. Understanding the data distribution is crucial when deciding on parametric or non-parametric methods.
6. **Data Preprocessing:** Histograms are useful in data preprocessing tasks, such as binning (grouping data into intervals), which can be helpful for feature engineering or data discretization.

8. **Decision Making:** When making decisions based on data, histograms can provide insights into the spread and characteristics of the data, helping decision-makers understand the implications of their choices.

Histograms are commonly used in various fields, including:

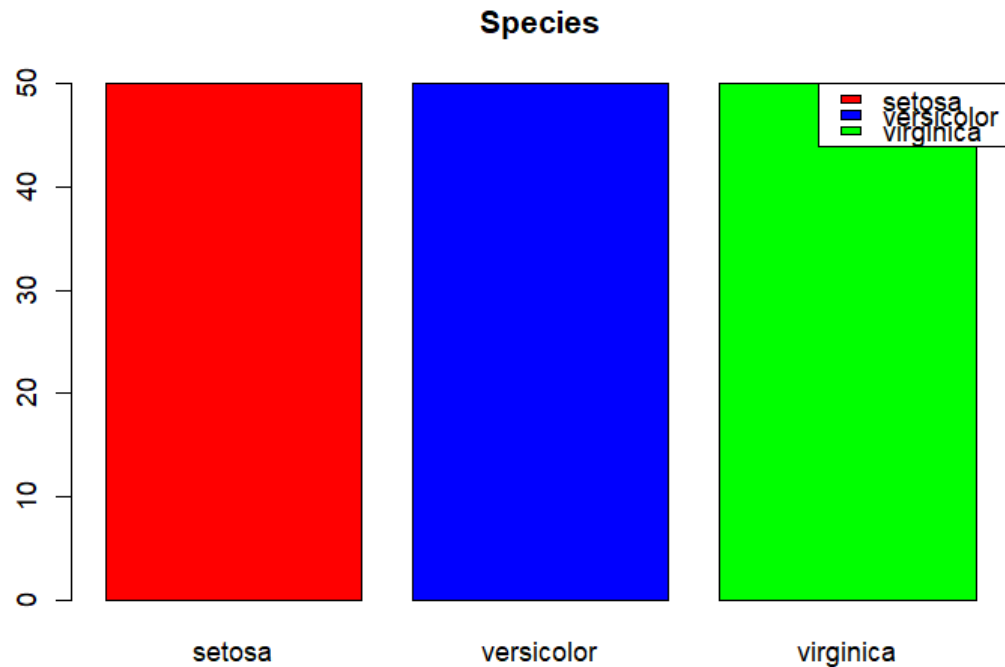
- **Statistics and Data Analysis:** To explore and describe the characteristics of datasets.
- **Finance:** For analyzing stock prices, market returns, and risk assessments.
- **Engineering:** In quality control and process analysis.
- **Biology:** To study characteristics like the distribution of species in ecology or the size distribution of cells in biology.
- **Social Sciences:** To analyze survey data, income distribution, and more.

Overall, histograms are a fundamental tool for exploring and summarizing data, making them an essential part of data analysis and visualization. They provide valuable insights into the underlying patterns and structures of datasets, aiding in informed decision-making and analysis.

Bar plot

- Bar plots are typically used to visualize and represent categorical or discrete data. These types of data are characterized by distinct categories, groups, or classes, with each data point falling into one of these categories.

```
```{r}
bar plot
barplot(table(iris$Species), main='Species',col=c("red","blue","green")) #table(iris$Species)It counts the number of
occurrences of each unique species (setosa, versicolor, and virginica).
legend("topright", legend = c("setosa", "versicolor", "virginica"),fill = c("red", "blue", "green"))
```
```





Bar plots are typically used to visualize and represent categorical or discrete data. These types of data are characterized by distinct categories, groups, or classes, with each data point falling into one of these categories. Bar plots are particularly well-suited for displaying the following types of data:



1. **Nominal Data:** Nominal data consists of categories or labels with no inherent order or ranking. Bar plots are used to show the counts or frequencies of different categories. Examples include:
 - Types of fruits in a fruit basket.
 - Colors of cars in a parking lot.
 - Species of animals in a wildlife reserve.
2. **Ordinal Data:** Ordinal data has categories with a specific order or ranking, but the intervals between them are not uniform or meaningful. Bar plots can display the frequency or distribution of ordinal categories. Examples include:
 - Survey responses such as "Strongly Disagree," "Disagree," "Neutral," "Agree," and "Strongly Agree."
 - Education levels such as "High School," "Bachelor's Degree," "Master's Degree," and "Ph.D."

3. **Count Data:** Count data represents the number of occurrences or events within specific categories. Bar plots are used to show the count or frequency of each category. Examples include:

- Number of customer complaints by product category.
- Frequency of disease cases by age group.
- Number of defects in manufacturing by defect type.

4. **Categorical Data:** Categorical data includes data points that belong to discrete categories or groups. Bar plots can be used to compare and visualize these categories. Examples include:

- Types of products sold by a retailer.
- Political party affiliations of registered voters.
- Marital status categories of survey respondents.

Bar plots are an effective way to visually summarize and communicate information about categorical or discrete data because they display the distribution, relationships, and comparisons between different categories or groups. They provide a clear representation of the data, making it easier to identify patterns, trends, and differences among the categories.



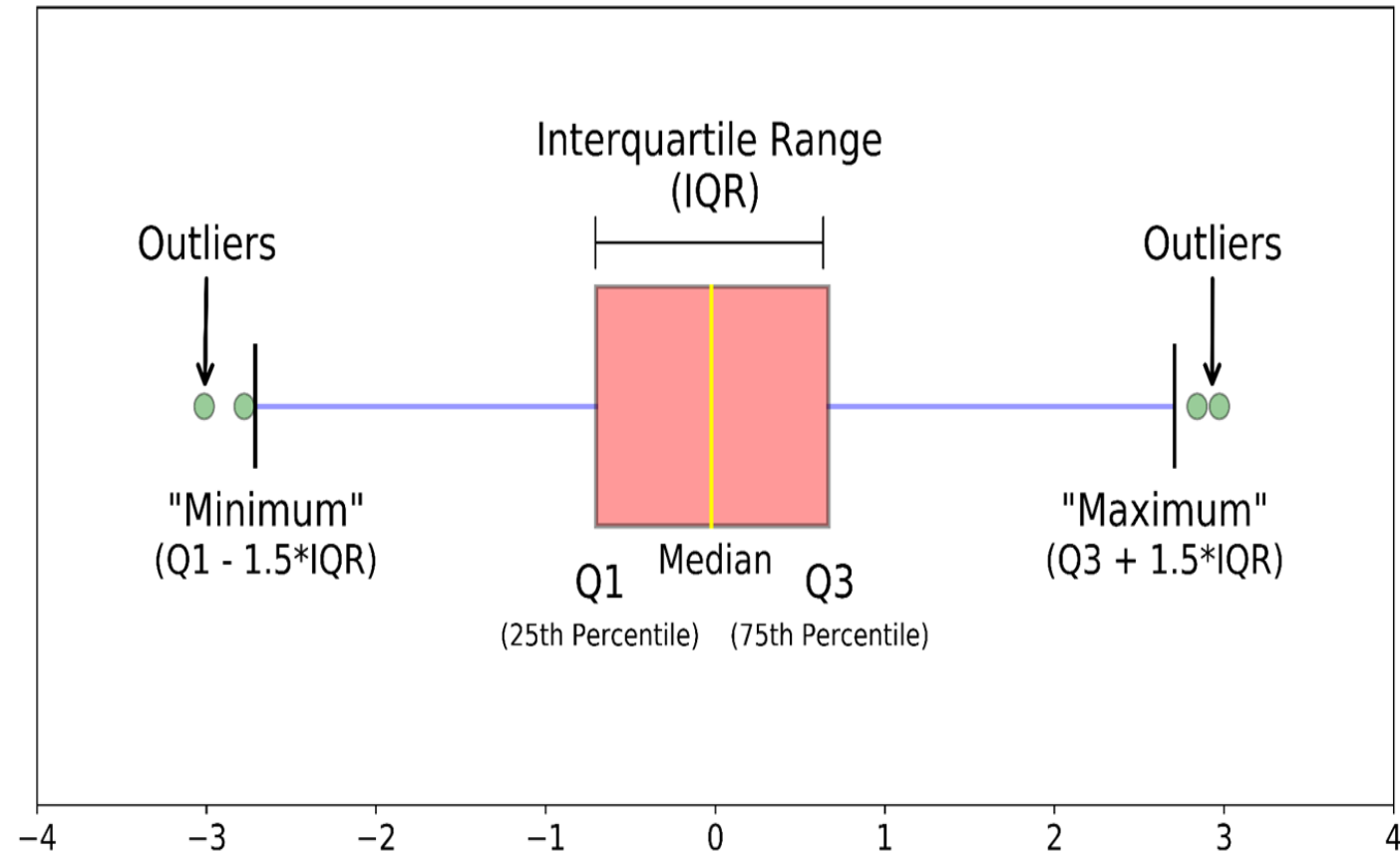
A bar plot, also known as a bar chart or bar graph, is a graphical representation of data using rectangular bars or columns. Each bar's length or height is proportional to the value it represents. Bar plots are used for visualizing categorical or discrete data and are valuable for several purposes:



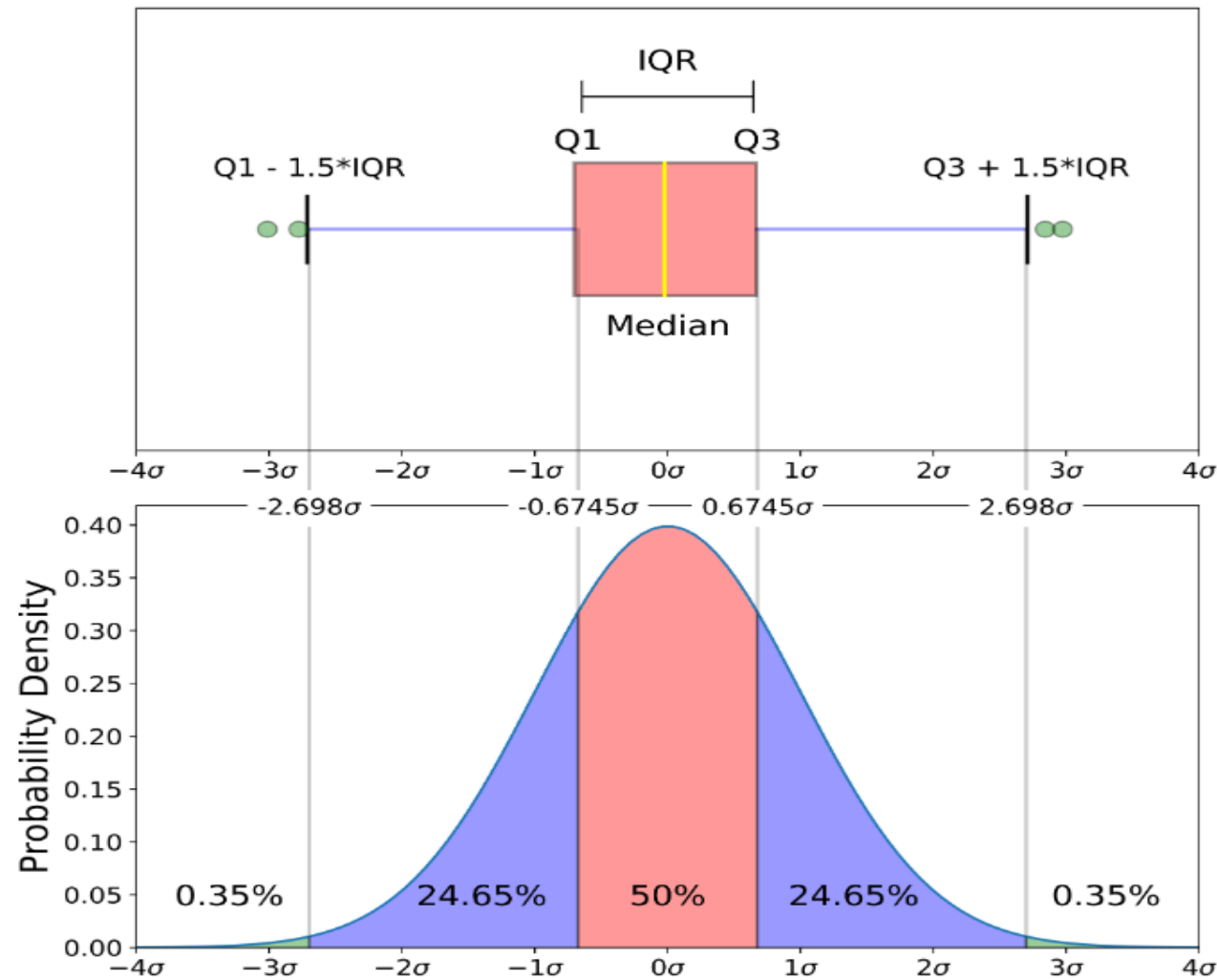
1. **Comparing Categories:** Bar plots are effective for comparing the values of different categories or groups. You can easily see which category has the highest or lowest values.
2. **Frequency Distribution:** Bar plots display the frequency or count of categories in a dataset. They are commonly used for displaying categorical data, such as the number of items sold in different product categories.
3. **Data Distribution:** You can use bar plots to visualize the distribution of data within discrete categories. This is helpful when you want to see how data is distributed across different classes or bins.
4. **Showing Trends:** Bar plots can be used to show trends over time or across different conditions. For example, you can create a bar plot to display monthly sales figures for a year to identify seasonal trends.
5. **Comparison of Multiple Groups:** Grouped or clustered bar plots allow you to compare multiple groups or subcategories within each main category. This is useful for displaying complex data comparisons.
6. **Categorical Relationships:** Bar plots can illustrate the relationships or associations between two or more categorical variables. For instance, you can create a stacked bar plot to show how the composition of one category varies within different subcategories.
7. **Ranking and Sorting:** Bar plots can help rank categories by their values, making it easy to identify the top or bottom performers.
8. **Visualizing Survey Data:** Bar plots are often used to visualize survey results, including responses to multiple-choice questions or preferences among different options.

Box plot

- A box plot, also known as a box-and-whisker plot, is a graphical representation of the distribution of a dataset. It provides a concise summary of key statistical measures and allows for a visual examination of the data's central tendency, spread, and presence of outliers.



Boxplot on a Normal Distribution



Comparison of a boxplot of a nearly normal distribution and a probability density function (pdf) for a normal distribution



A box plot, also known as a box-and-whisker plot, is a graphical representation of the distribution of a dataset. It displays a summary of key statistical measures, providing insights into the central tendency, spread, and presence of outliers within the data. Box plots are particularly useful for visualizing the distribution of numerical data, especially when comparing multiple groups or variables. Here's how to interpret and use a box plot:



1. **Box:** The main box in a box plot represents the interquartile range (IQR), which encompasses the middle 50% of the data. The top and bottom edges of the box represent the first quartile (Q1) and the third quartile (Q3), respectively. The length of the box (IQR) indicates the spread of the middle 50% of the data. The wider the box, the more spread out the data.
2. **Median (Line Inside the Box):** A horizontal line inside the box represents the median (Q2), which is the middle value when the data is ordered. It is a measure of central tendency and indicates the typical or central value of the data.
3. **Whiskers:** The "whiskers" extend from the box and represent the range of the data. Typically, the whiskers extend to the minimum and maximum values within a defined range, often determined by a formula (e.g., 1.5 times the IQR). Any data points beyond the whiskers are considered potential outliers.
4. **Outliers (Data Points Outside Whiskers):** Data points that fall outside the whiskers are plotted as individual points. These are potential outliers and may indicate data points that are significantly different from the rest of the data.

Box plots are used for various purposes, including:

- **Comparing Distributions:** Box plots are helpful for visually comparing the distributions of multiple groups or datasets. For example, you can compare the exam scores of students from different schools or the performance of different products.
- **Identifying Skewness:** By observing the position of the median within the box and the length of the whiskers, you can assess the skewness of the data. A skewed dataset will have a median that is not centered within the box.
- **Detecting Outliers:** Outliers can be easily identified in a box plot as individual points outside the whiskers. These outliers may warrant further investigation.
- **Summarizing Data:** Box plots provide a concise summary of the data's central tendency, spread, and presence of outliers, making them valuable in reports and presentations.
- **Comparing Variables:** When comparing multiple variables within a dataset, box plots allow you to visualize their distributions side by side, making it easy to identify differences and similarities.
- **Quality Control:** In industries like manufacturing, box plots are used to monitor and control the quality of products and processes. Deviations from expected values can be quickly detected.

Overall, box plots are a powerful tool in data analysis and visualization, particularly for exploring the distribution and characteristics of numerical data. They are widely used in various fields, including statistics, science, finance, and quality control.



Box plots are primarily used for visualizing and summarizing the distribution of numerical data, specifically continuous or interval data. They are particularly effective for the following types of data:



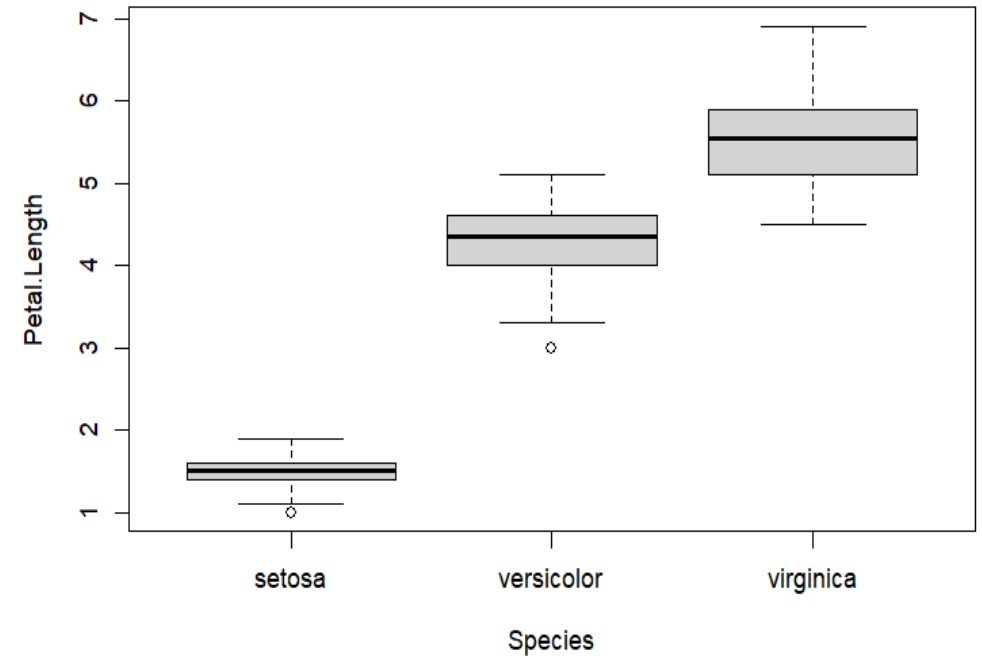
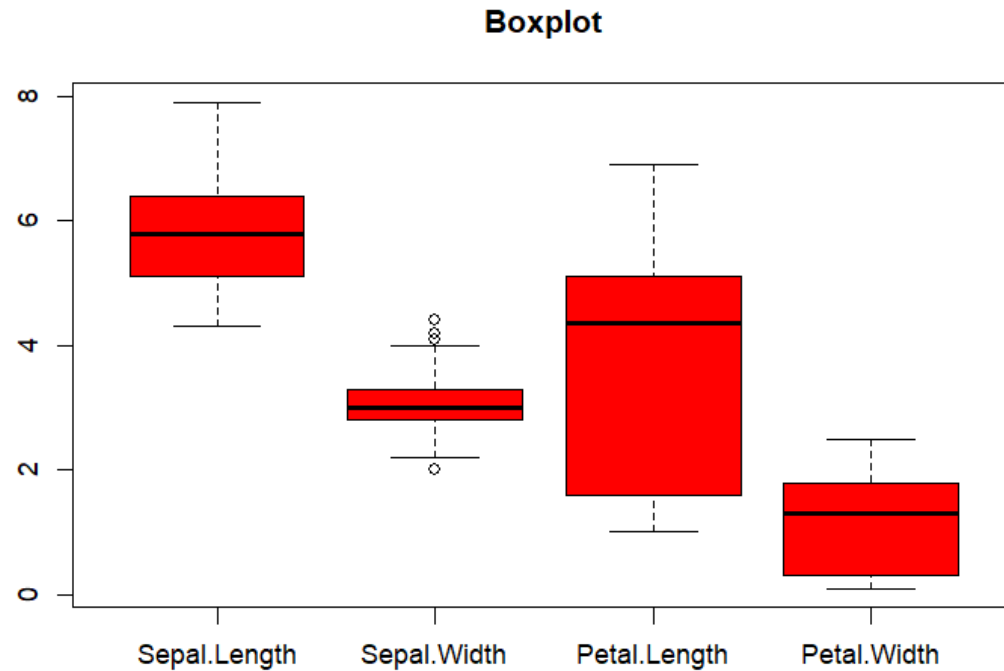
1. **Univariate Data:** Box plots are commonly used to analyze a single variable's distribution. They provide insights into the central tendency, spread, and presence of outliers within a single dataset.
2. **Comparing Multiple Groups:** Box plots are valuable when comparing the distributions of multiple groups or categories within a dataset. They allow you to visualize and compare the characteristics of different groups, making them useful for exploratory data analysis and hypothesis testing.
3. **Skewed Data:** Box plots are helpful for identifying skewness in data distributions. A skewed dataset will have a box plot where the median is not centered within the box, and one whisker may be longer than the other, indicating the direction of skewness.
4. **Outlier Detection:** Box plots make it easy to detect outliers, which are data points that deviate significantly from the rest of the data. Outliers are typically shown as individual points outside the whiskers of the box plot.
5. **Quantitative Data:** Box plots are suitable for quantitative data, which includes measurements with meaningful numeric values. Examples include test scores, income levels, temperature readings, and more.

6. **Comparing Variables:** Box plots can be used to compare the distributions of multiple variables within a dataset. This is useful when assessing how different variables behave in relation to one another.
7. **Quality Control:** In manufacturing and quality control processes, box plots are used to monitor product quality by tracking key metrics. Deviations from expected values can be quickly identified using box plots.
8. **Statistical Analysis:** Box plots are often used in statistical analyses, such as ANOVA (analysis of variance) and non-parametric tests, to visualize and compare data distributions across different groups or conditions.
9. **Scientific Research:** Scientists use box plots to visualize data in various fields, including biology, environmental science, and social sciences, to understand and communicate data distributions.

In summary, box plots are a versatile tool for visualizing and summarizing numerical data, making them valuable in data exploration, analysis, and presentation across a wide range of domains and disciplines. They provide a clear and informative way to assess the distributional characteristics of data and detect potential anomalies.

Box plot

```
##{r}  
##### box plot  
boxplot(iris[,-5], main='Boxplot',col="red")  
boxplot(Petal.Length~Species, data=iris)
```

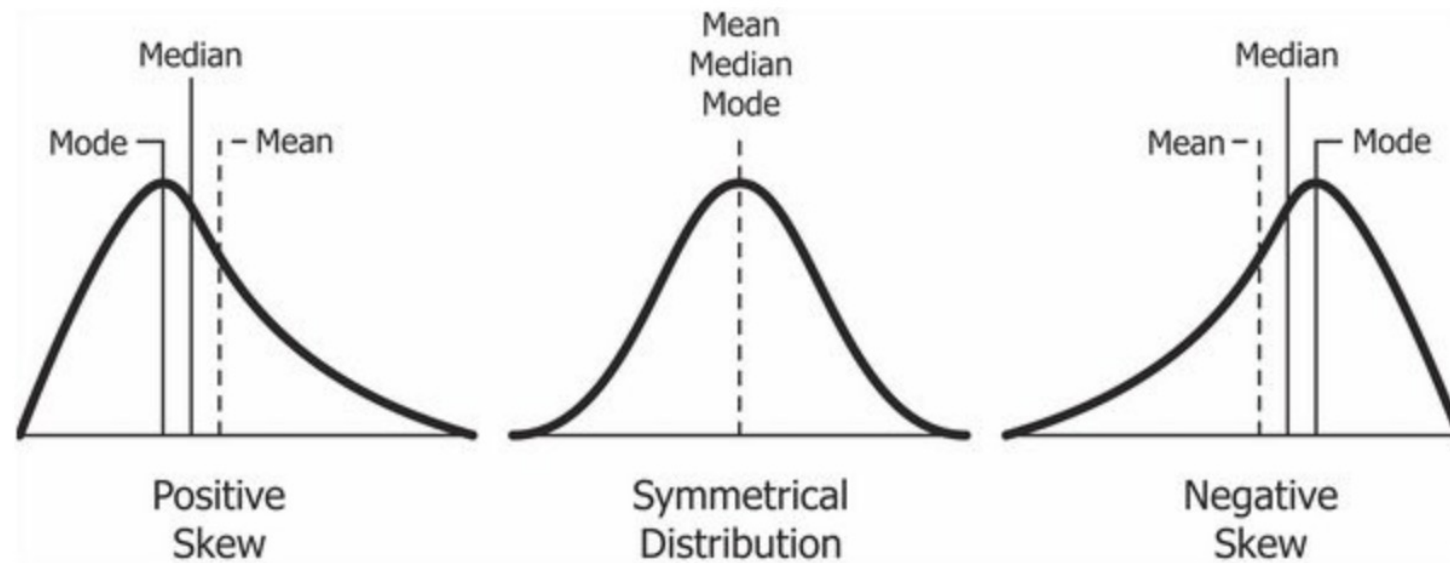


Skewness

- In probability theory and statistics, **skewness** is a measure of the asymmetry of the probability distribution of a real-valued random variable about its mean. The skewness value can be positive, zero, negative, or undefined.

There are two main types of skewness:


- 1. Positive Skew (Right Skew):** In a positively skewed distribution, the tail extends to the right, indicating that there are a few unusually large values or outliers on the right side of the distribution. The bulk of the data is concentrated on the left side.
- 2. Negative Skew (Left Skew):** In a negatively skewed distribution, the tail extends to the left, indicating that there are a few unusually small values or outliers on the left side of the distribution. The bulk of the data is concentrated on the right side.



Here's what happens when you have skewed data:

1. **Central Tendency:** The choice of central tendency measure (mean, median, or mode) becomes important. In positively skewed data, the mean tends to be greater than the median, while in negatively skewed data, the mean tends to be less than the median. The mode may not accurately represent the central value.
2. **Spread:** Skewed data can affect the spread or dispersion of the data. In a positively skewed distribution, the spread may appear larger because of the presence of outliers on the right. In a negatively skewed distribution, the spread may appear smaller because of outliers on the left.
3. **Data Analysis:** When performing statistical analysis or modeling on skewed data, it's important to consider the impact of skewness on the results. For example, in a positively skewed distribution, the mean may be inflated due to the presence of outliers.
4. **Visualization:** Visualizing skewed data with histograms or box plots can help identify the direction and degree of skewness. The shape of these plots can provide insights into the distribution's characteristics.
5. **Data Transformation:** Sometimes, data transformation techniques like logarithmic transformation or Box-Cox transformation are applied to mitigate the effects of skewness. These transformations can make the data more symmetric and improve the performance of statistical models.
6. **Outlier Detection:** Skewed data often results from the presence of outliers. Identifying and handling these outliers may be necessary to analyze the data effectively.

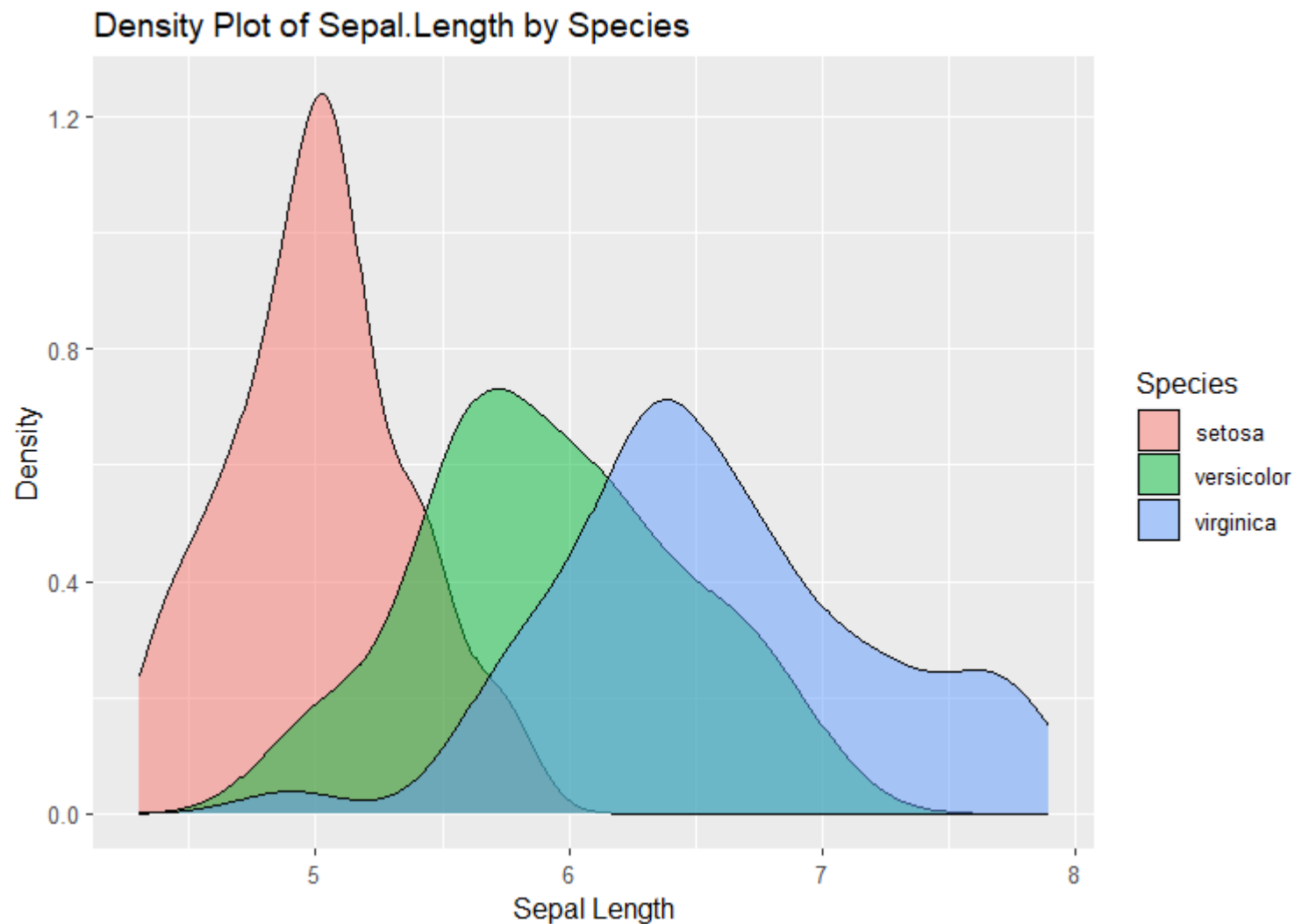
In summary, recognizing and addressing skewness in data is essential for accurate analysis and modeling. Skewed data can affect central tendency measures, spread, and the validity of statistical assumptions. Understanding the direction and degree of skewness is crucial for

 Regenerate

Density plot

```
```{r}
Load the ggplot2 library for creating density plots
library(ggplot2)

Create a density plot for Sepal.Length
ggplot(iris, aes(x = Sepal.Length, fill = Species)) +
 geom_density(alpha = 0.5) +
 labs(
 title = "Density Plot of Sepal.Length by Species",
 x = "Sepal Length",
 y = "Density"
)
```
```

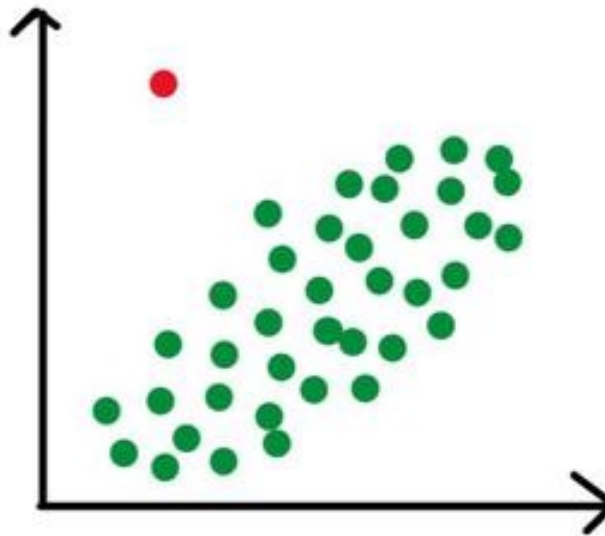


Outliers

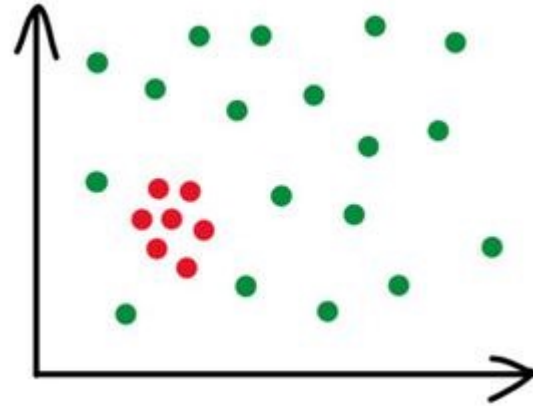
- In data science, outliers are data points or observations that significantly differ from the majority of the data in a dataset. They are values that are unusually high (positive outliers) or unusually low (negative outliers) when compared to the central tendency and spread of the rest of the data.
- Outliers can arise for various reasons, including data entry errors, measurement errors, natural variability, or extreme events.

Outliers are of three types, namely –

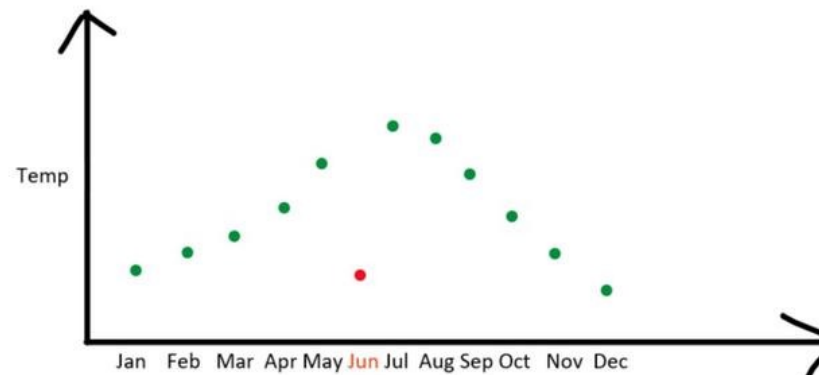
- 1. Global Outliers:** - Global outliers are data points that deviate significantly from the overall distribution of a dataset.



2. Collective Outliers:- Collective outliers are groups of data points that collectively deviate significantly from the overall distribution of a dataset.

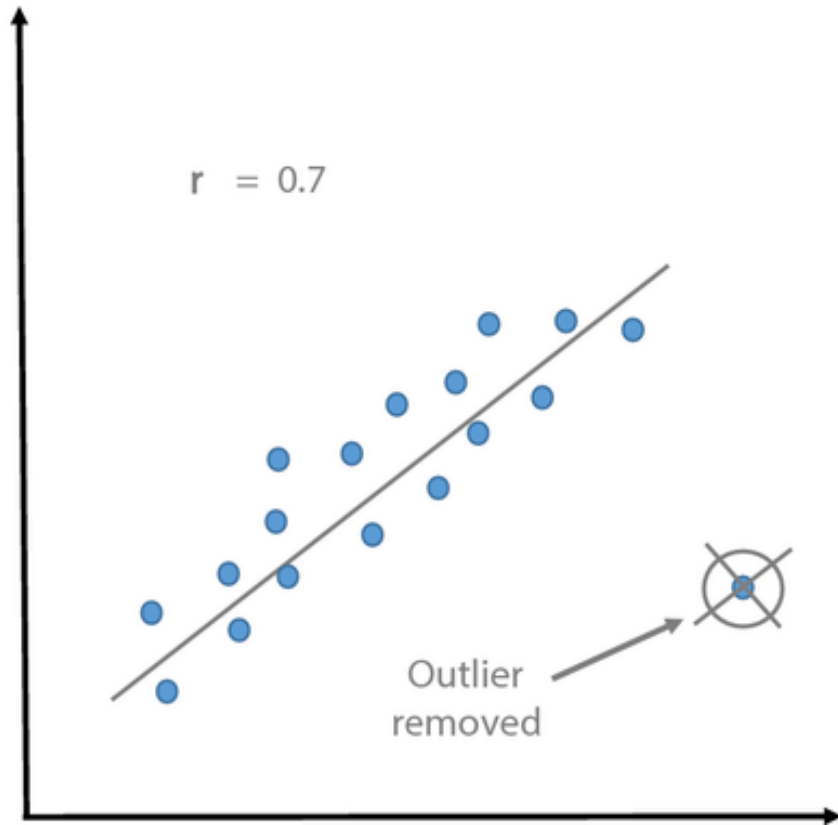
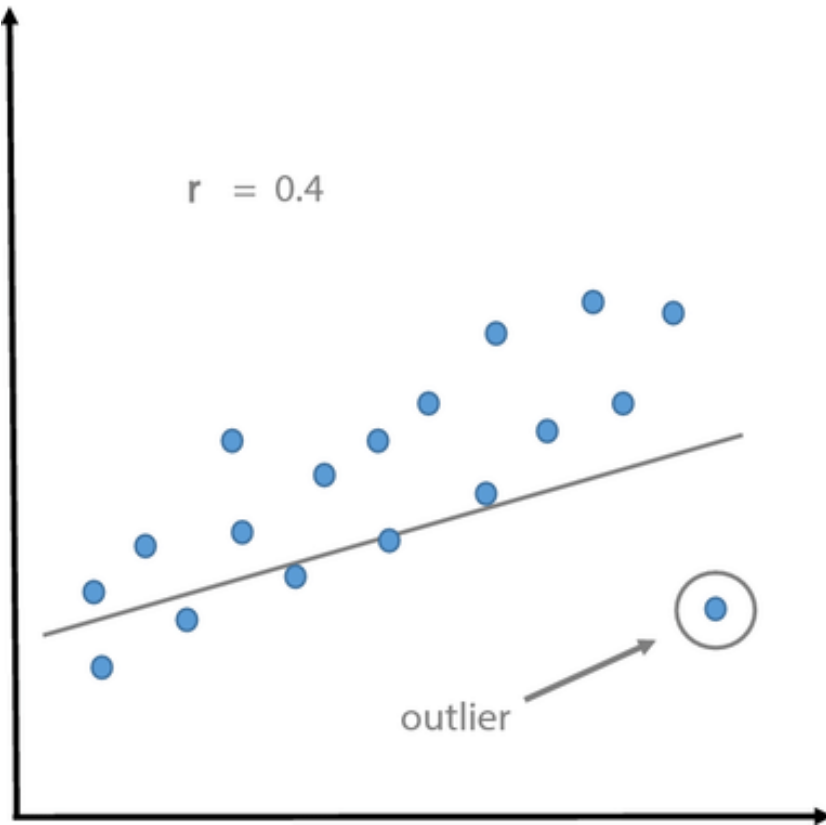


3 . Contextual Outliers: - Contextual outliers are data points that deviate significantly from the expected behavior within a specific context or subgroup.



Outliers

Example:-



what are the ways we can detect the outliers in a data?



Detecting outliers in data is a crucial step in data analysis and can be done using various methods. Here are some common techniques and approaches for detecting outliers:



1. Visual Inspection:

- **Box Plots:** Box plots visually display the distribution of data and identify potential outliers as data points outside the whiskers (typically defined as 1.5 times the interquartile range beyond the upper or lower quartile).
- **Scatter Plots:** Scatter plots can reveal outliers by showing data points that are far from the main cluster of points.

2. Summary Statistics:

- **Z-Score:** Calculate the z-score for each data point, which measures how many standard deviations a point is away from the mean. Points with high absolute z-scores (e.g., greater than 3 or -3) are considered outliers.
- **IQR (Interquartile Range):** Compute the interquartile range (IQR) and identify data points that fall below $Q1 - 1.5 * IQR$ or above $Q3 + 1.5 * IQR$, where $Q1$ and $Q3$ are the first and third quartiles, respectively.

3. Density-Based Methods:

- **Kernel Density Estimation (KDE):** KDE estimates the probability density function of data and can identify regions with low density as potential outlier regions.
- **Local Outlier Factor (LOF):** LOF calculates the local density of data points relative to their neighbors. Points with a significantly lower density are considered outliers.

🔄 Regenerate

4. Distance-Based Methods:

- **Euclidean Distance:** Calculate the Euclidean distance of each data point from the centroid or mean of the data. Points with large distances can be considered outliers.
- **Mahalanobis Distance:** This distance metric accounts for the correlations between variables and is particularly useful for multivariate data.

5. Machine Learning Algorithms:

- Some machine learning algorithms, such as isolation forests and one-class SVM (Support Vector Machines), are specifically designed for outlier detection. They learn to separate normal data from outliers.

6. Statistical Tests:

- Use statistical tests, such as the Grubbs' test or Dixon's test, to identify single outliers in univariate datasets.

7. Domain Knowledge:

- Expert knowledge of the data and the domain can be valuable for identifying outliers that are not evident from statistical methods alone.

8. Visualization Tools:

- Interactive data visualization tools like Tableau, Power BI, or Python libraries like Seaborn and Matplotlib can help visually identify outliers.

9. Time-Series Analysis:

- For time-series data, methods like the detection of anomalies using seasonality and trend decomposition (STL decomposition) can be effective.

10. Ensemble Methods:

- Combine multiple outlier detection methods to improve detection accuracy and robustness.

It's essential to choose the outlier detection method that best suits the characteristics of your data and the goals of your analysis. Additionally, consider the potential impact of outliers on your analysis and whether they should be removed, transformed, or handled differently based on the context.

🔄 Regenerate



Dealing with outliers in data is a crucial step in data preprocessing to ensure that they don't unduly influence statistical analyses or machine learning models. Here are several techniques for handling outliers:



1. **Data Truncation or Censoring:**

- Truncate or censor extreme values by setting a threshold beyond which data points are considered outliers and are either removed or replaced with a less extreme value.
- Example: Replace values above a certain percentile (e.g., 99th percentile) with the threshold value.

2. **Winsorization:**

- Winsorization is a technique that replaces extreme values with less extreme values, often at a specific percentile (e.g., 99th percentile).
- This approach reduces the impact of outliers while preserving the overall distribution of the data.

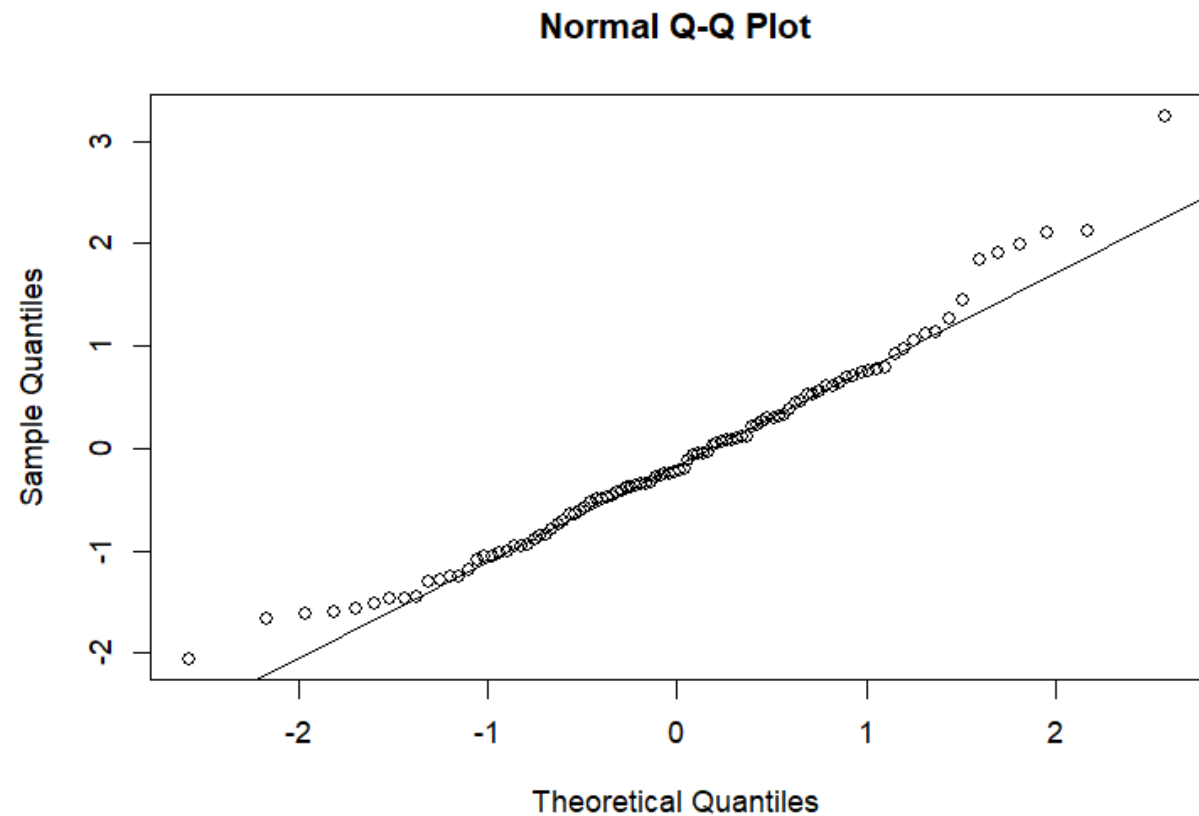
3. **Transformation:**

- Transform the data using mathematical functions to make it less sensitive to outliers.
- Common transformations include the log transformation, square root transformation, or Box-Cox transformation.
- Transformed data may be less skewed and better suited for analysis.

4. **Robust Statistical Methods:**
 - Use statistical methods that are less sensitive to outliers, such as robust regression techniques (e.g., robust linear regression or robust regression with M-estimators), which downweight the influence of outliers during model fitting.
5. **Imputation:**
 - For missing values resulting from outlier removal, impute new values based on the distribution of the remaining data.
 - Imputation methods can include mean imputation, median imputation, or regression-based imputation.
6. **Model-Based Approaches:**
 - Develop models that explicitly account for outliers. For example, robust statistical models, such as the Huber loss in machine learning, can handle outliers more gracefully.
7. **Data Segmentation:**
 - Divide the dataset into subsets based on certain criteria or features. Analyze each subset separately to address outliers within a more specific context.
8. **Data Clustering:**
 - Use clustering techniques to identify and isolate outliers in separate clusters. This allows you to treat outliers differently from the main data.
9. **Trimming:**
 - Remove a fixed percentage of data points from the tails of the distribution, effectively eliminating extreme values.
 - Be cautious with this approach, as it may lead to a significant loss of data.
10. **Machine Learning Models:**
 - Some machine learning models, such as tree-based models or support vector machines with appropriate kernels, can handle outliers naturally.
 - Train models on datasets that have undergone outlier handling to improve model performance.

QQ-plot

```
##{r}  
# Generate example data  
data <- rnorm(100) # Generating normally distributed data  
  
# Create a QQ plot against the normal distribution  
qqnorm(data)  
qqline(data)
```





A QQ plot, or quantile-quantile plot, is a graphical tool used to assess whether a dataset follows a particular theoretical distribution or if it exhibits departures from that distribution. It is particularly useful for checking the normality of data, but it can also be applied to other distributions.



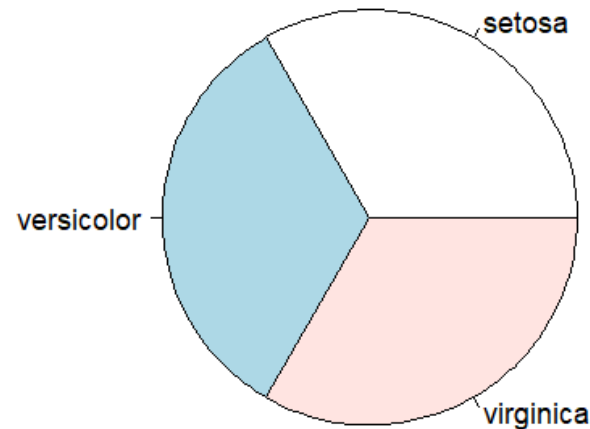
Here's how a QQ plot works:

1. **Basic Idea:** In a QQ plot, the quantiles of the observed data are compared to the quantiles of a theoretical distribution (e.g., the normal distribution). If the data closely follows the theoretical distribution, the points in the QQ plot will approximately fall along a straight line (the diagonal).
3. **Interpretation:**
 - If the QQ plot forms a straight line, it suggests that your data is approximately normally distributed (for a QQ plot against the normal distribution).
 - If the QQ plot deviates from a straight line, it indicates departures from normality. For example, if the points curve upward, it suggests positive skewness, while a downward curve suggests negative skewness.
4. **Outliers:** Outliers in the data may appear as points far away from the expected straight line in the QQ plot, indicating deviations from the assumed distribution.
5. **Customization:** QQ plots can be customized to compare against various theoretical distributions, not just the normal distribution. You can also add a reference line to help assess deviations more easily.

Pie chart

- Pie charts are a type of data visualization that presents data in a circular graph, divided into slices to represent the proportion of each category within a dataset.
- Pie charts are typically used to visualize categorical data, particularly when you want to show the distribution or proportions of different categories within a whole

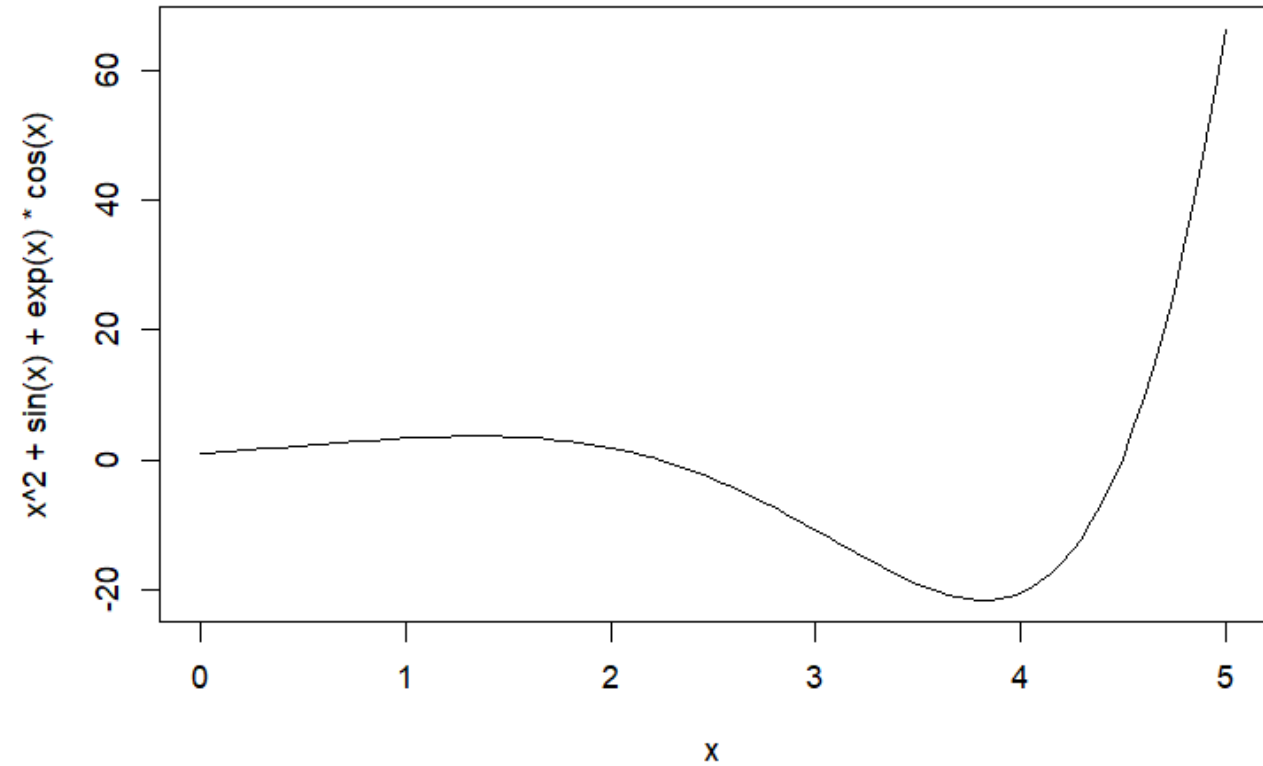
```
##### pie chart##  
{r}  
pie(table(iris$Species))|
```



Curve

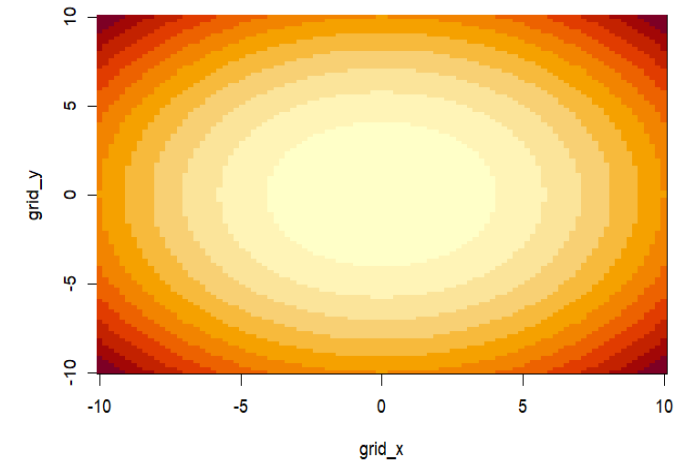
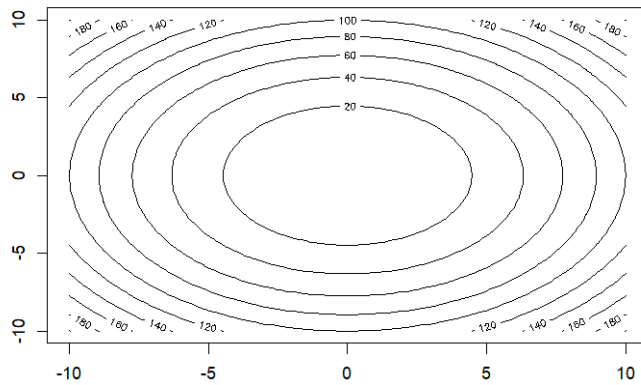
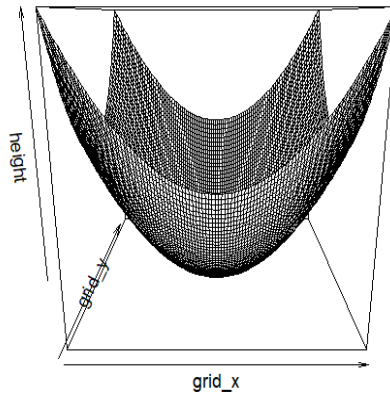
```
**Curve**
```

```
{r}  
##### curve  
curve(x**2+sin(x)+exp(x)*cos(x), 0,5)|
```



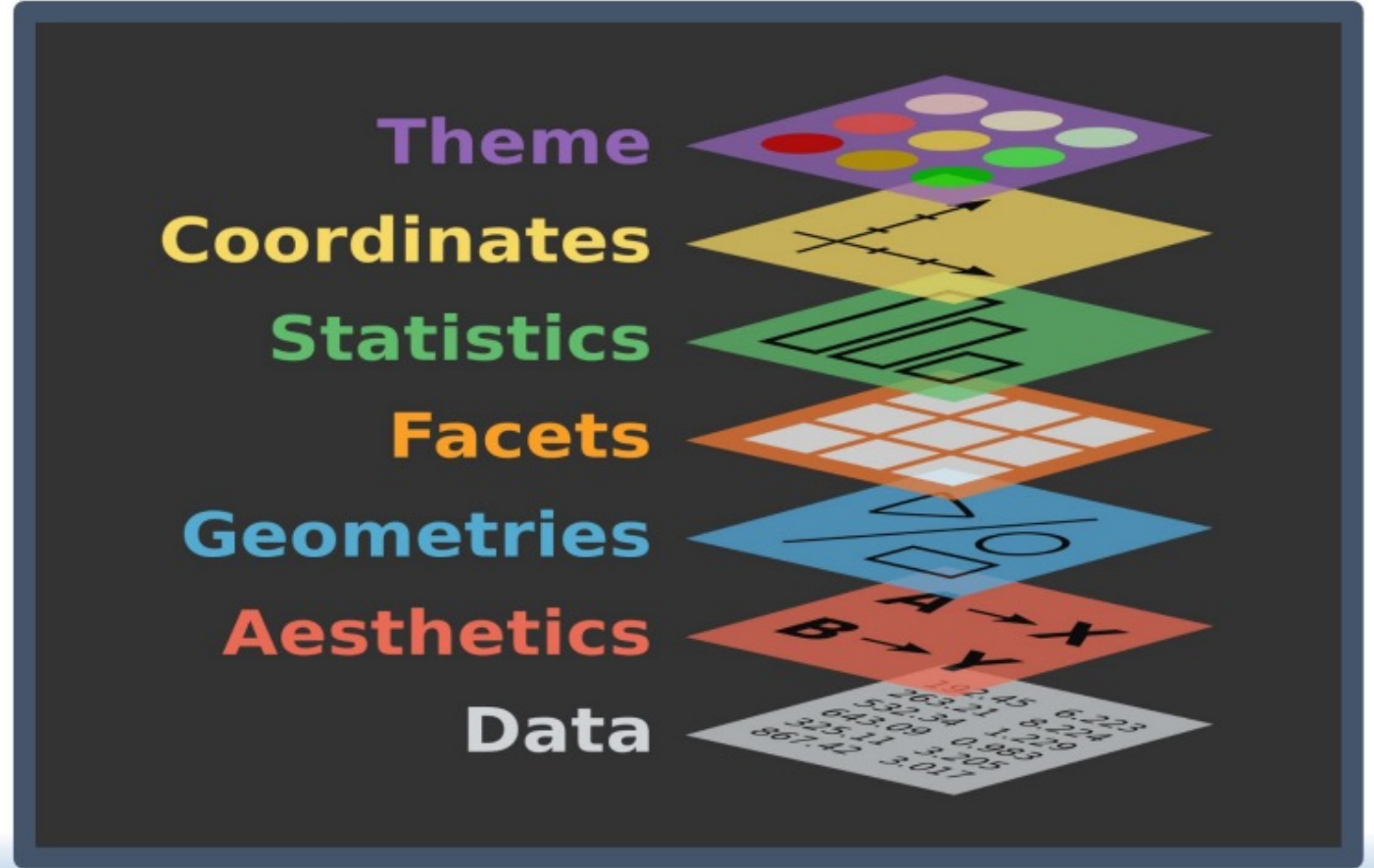
3D Density

```
**3D Density**
```{r}
grid_x <- seq(-10,10,length=100)
grid_y <- grid_x
height <- matrix(0, nrow=length(grid_x), ncol=length(grid_y))
for(i in 1:length(grid_x)){
 for(j in 1:length(grid_y)){
 height[i,j] <- grid_x[i]**2 + grid_y[j]**2
 }
}
persp(grid_x, grid_y, height)
contour(grid_x, grid_y, height)
image(grid_x, grid_y, height)|
```
```



ggplot2 - most powerful R graphic tools

ggplot is a graph formed by stacking each layer. The way of stacking is achieved by "+"
When constructing a ggplot graph, the first three layers are necessary (Data, Aesthetics, Geometries). For more information, please refer to [ggplot2 API](#)





``ggplot2`` is a widely used data visualization package in the R programming language. It is part of the ``tidyverse`` ecosystem and provides a flexible and powerful framework for creating a wide range of high-quality statistical and data visualizations. Here are the key features and concepts associated with ``ggplot2``:



1. **Grammar of Graphics:** ``ggplot2`` is based on the grammar of graphics, a structured approach to data visualization. This grammar consists of a set of building blocks that you assemble to create a plot. The core components are data, aesthetics (variables mapped to visual properties), and layers.
2. **Data:** You start by specifying the dataset you want to visualize.
3. **Aesthetics (aes):** Aesthetics refer to how data variables are mapped to visual properties of the plot, such as x and y coordinates, colors, shapes, and sizes.
4. **Layers:** In ``ggplot2``, you add layers to your plot using the ``geom_`` functions. Each ``geom_`` function defines a specific type of geometric object or visualization element (e.g., points, lines, bars, histograms).
5. **Facets:** You can use facets to create small multiples or subplots based on one or more categorical variables. This is useful for visualizing data across multiple dimensions.
6. **Themes:** ``ggplot2`` allows you to customize the appearance of your plots using themes, including fonts, colors, backgrounds, and more.
7. **Statistical Transformations:** You can apply statistical transformations to your data using functions like ``stat_summary``, ``stat_smooth``, and others to generate summary statistics or smooth curves.

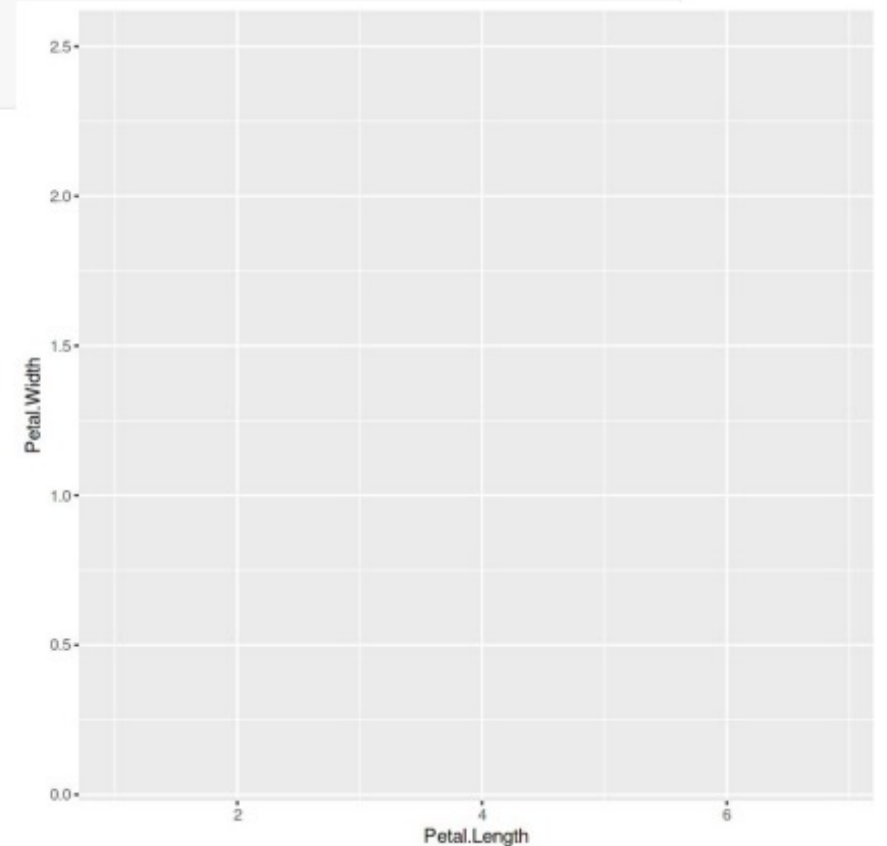
1. **Data:** We first build an empty canvas and input a df (data.frame)

```
# Two ways to build an empty graph  
library(tidyverse)  
library(ggplot2)  
iris %>% ggplot()  
ggplot(iris)
```

2. Aesthetics : Specify the mapping manner

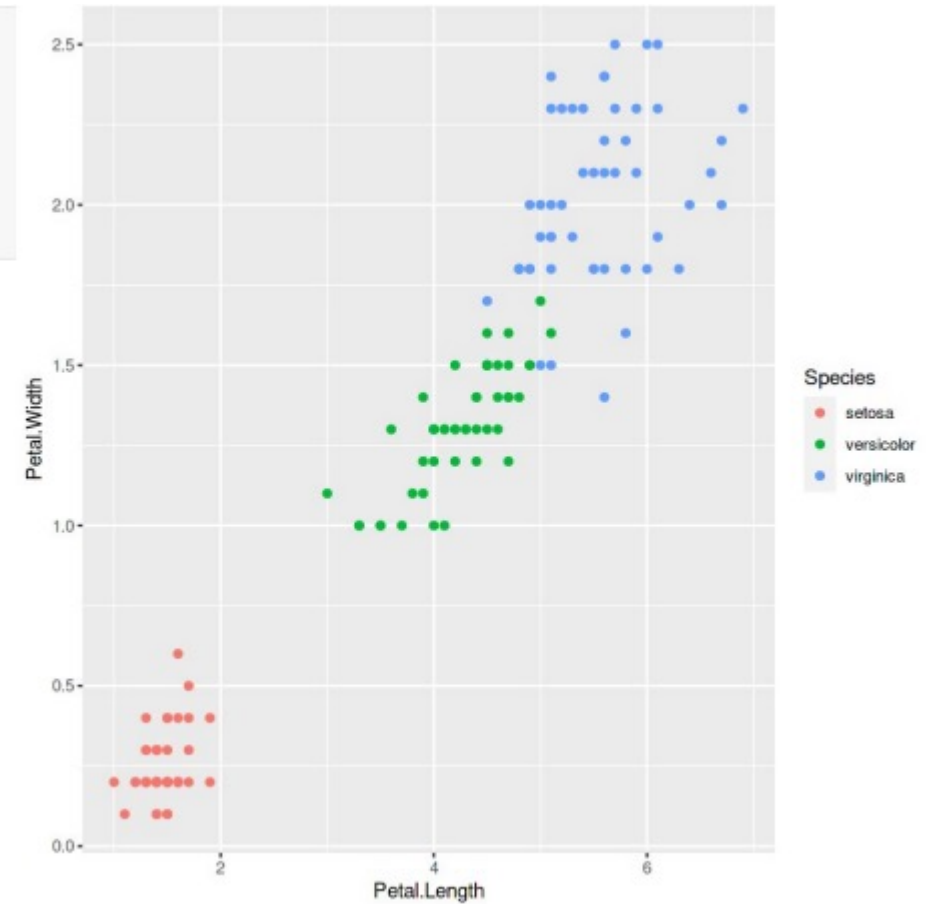
```
iris %>% ggplot() +  
  aes(x=Petal.Length, y=Petal.Width, color=Species)
```

- **aes(x, y, color, size, shape, alpha)**
 - x: x axis data
 - y: y axis data
 - color: color function
 - size: geometric attribute size
 - shape: geometric attribute shape
 - alpha: transparency



3. Geometries : Specify the geometric attribute

```
iris %>% ggplot() +  
  aes(x=Petal.Length, y=Petal.Width, color=Species) +  
  geom_point()
```



Other commands

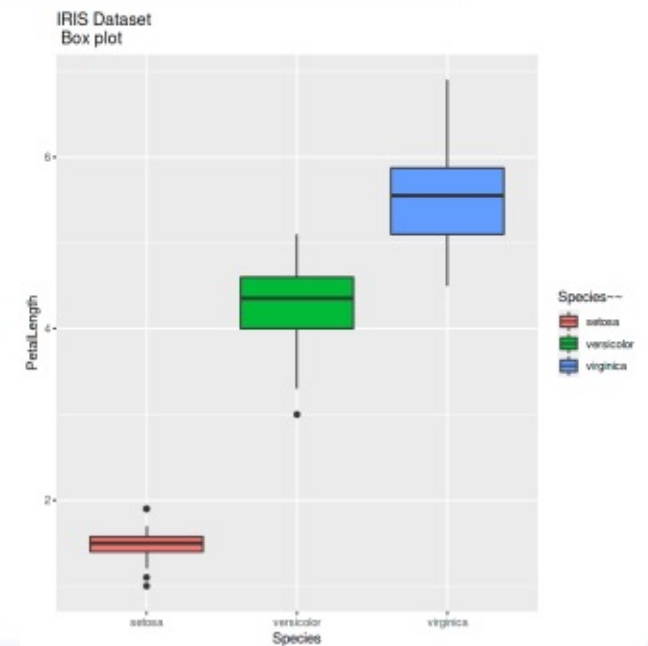
```
ggplot(iris) + aes(x=Species, y=Petal.Length, fill=Species) +  
  geom_boxplot() + labs(x='Species', y='Petal.Length',  
    fill='Species~~', title='IRIS Dataset\n Box plot')
```

Other commonly used function

- `ggtitle()` : set title
- `xlab()` : set x axis title
- `ylab()` : set y axis title

Or we can just simply use the following function

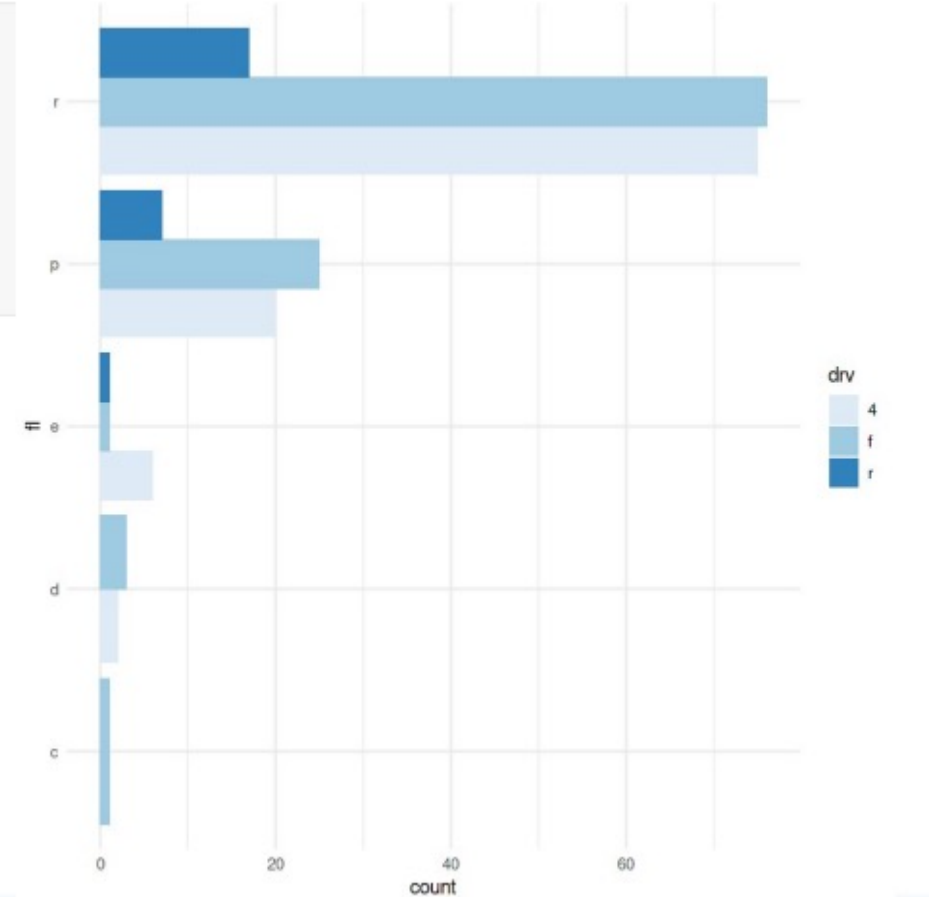
- `labs(x,y,title,fill)`



geom_Bar Plot

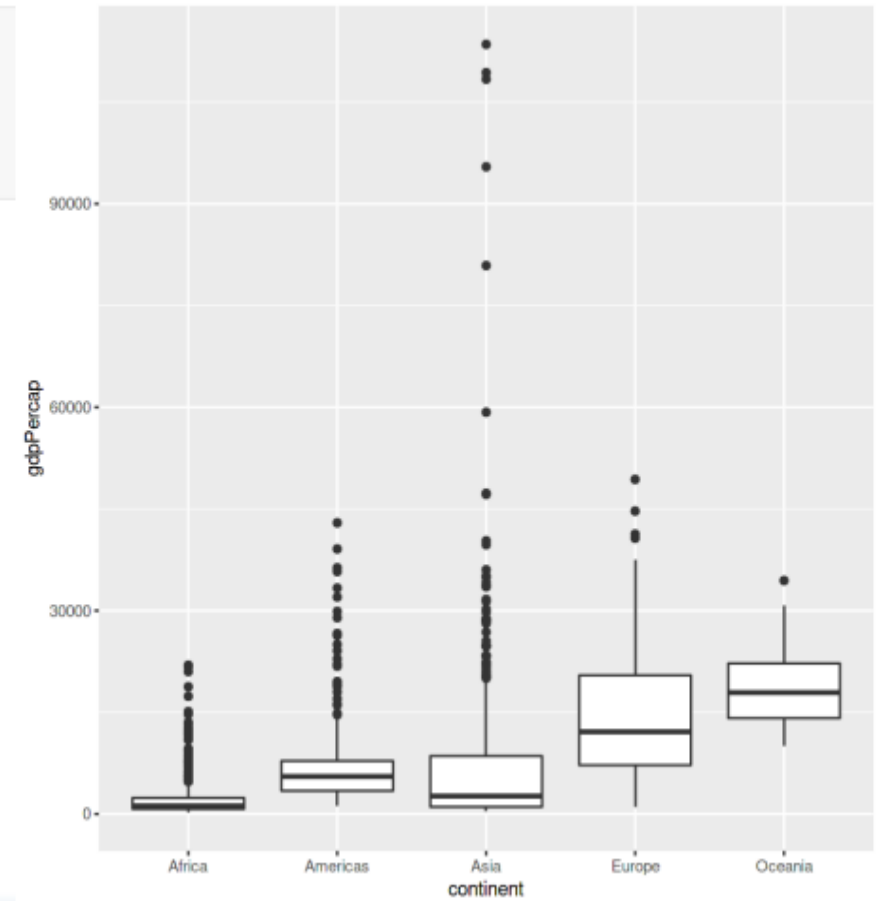
```
ggplot(mpg) + aes(x=fl,fill=drv) +  
  geom_bar(position='dodge') + # change to fill or stack  
  coord_flip() + theme_minimal() +  
  scale_fill_brewer(palette="Blues")
```

- `geom_bar(position)`
 - `dodge`: dodge bars
 - `fill`: fill bars
 - `stack`: stack bars



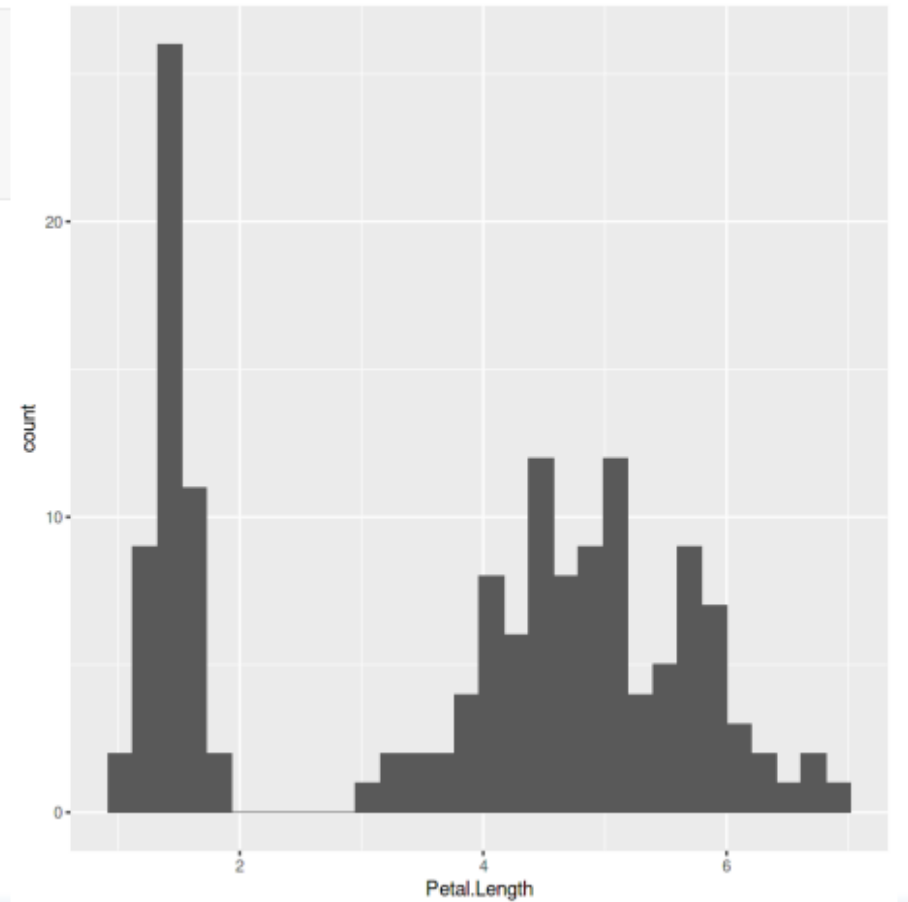
geom_Boxplot

```
ggplot(gapminder, aes(x = continent, y = gdpPercap)) +  
  geom_boxplot()
```



geom_histogram

```
ggplot(iris) + aes(x=Petal.Length) + stat_bin(bins=30) +  
  geom_histogram()
```



Homework 4 (submitted to e3.nycu.edu.tw before Oct 18, 2023)

- Use R, Python, and suitable computer packages to conduct visualization, including Boxplot, Bar plot, scatter plot, QQ plot, Density plot, pie chart, and histogram....
- Check whether **Simpson's paradox** exists in your dataset and explain what you find and why you choose these visualization methods.
- Do Visualization with and without outliers and compare the results.
- Discuss possible problems you plan to investigate for future studies

Possible sources of open datasets:

- UCI Machine Learning Repository
(<https://archive.ics.uci.edu/ml/datasets.php>)
- Kaggle Datasets (<https://www.kaggle.com/datasets>)