**Exploratory Data Analysis**

**using**

**Power BI**

Contents

[**1.** **Introduction to Power BI** 6](#_Toc152346499)

[**1.1.** **Key features of Power BI** 6](#_Toc152346500)

[**1.1.** **Why Power BI?** 7](#_Toc152346501)

[**1.2.** **Two basic elements of Power BI** 7](#_Toc152346502)

[**1.2.1** **Power BI Desktop** 7](#_Toc152346509)

[**1.2.2** **Power BI Services** 7](#_Toc152346510)

[**1.3.** **Power BI Terminologies** 8](#_Toc152346511)

[**2.** **Data Analysis Expressions (DAX)** 10](#_Toc152346512)

[**2.1** **New DAX functions** 10](#_Toc152346514)

[**2.2** **Aggregation Functions** 11](#_Toc152346515)

[**2.3** **Date and Time Functions** 12](#_Toc152346516)

[**2.4** **Filter functions** 12](#_Toc152346517)

[**2.5** **Financial functions** 13](#_Toc152346518)

[**2.6** **Table manipulation functions** 13](#_Toc152346519)

[**3.** **Steps to create visualizations in Power BI** 14](#_Toc152346520)

[**3.1.** **Get Data** 14](#_Toc152346524)

[**3.2.** **Transform & Clean Data** 14](#_Toc152346525)

[**3.3.** **Inspect dataset loaded into Power BI** 15](#_Toc152346526)

[**3.4.** **Create visuals on report pages** 16](#_Toc152346527)

[**3.5.** **Share reports** 18](#_Toc152346528)

[**4.** **Use case: Iris Dataset visualization** 18](#_Toc152346529)

[**4.1.** **Introduction** 18](#_Toc152346530)

[**4.2.** **Data** 19](#_Toc152346531)

[**5.2.1.** **Data Dictionary** 19](#_Toc152346537)

[**5.2.2.** **Data understanding** 19](#_Toc152346538)

[**4.3.** **Visualizations using Power BI** 24](#_Toc152346539)

[**5.** **Appendix** 42](#_Toc152346540)

**Table of Figures**

|  |  |  |
| --- | --- | --- |
| 1 | *Fig. 1.1. Power BI Terminologies* |  |
| 2 | *Fig. 1.2. Visualization Pane Tabs* |  |
| 3 | Fig. 2.1(a) Creating a new table to show the working of the new function, NETWORKDAYS |  |
| 4 | Fig. 2.1(b). Illustration of the function, NETWORKDAYS |  |
| 5 | Fig. 3.1. Steps to create a visual in Power BI |  |
| 6 | Fig. 3.2. Get Data in Power BI |  |
| 7 | Fig. 3.3. Transform Data in Power BI |  |
| 8 | Fig. 3.4. Transform Data tab options in Power BI |  |
| 9 | Fig. 3.5. Table under Data Pane |  |
| 10 | Fig. 3.6. Table with fields under Data Pane |  |
| 11 | Fig. 3.7. Select the Don’t Summarize option for the chosen numeric column |  |
| 12 | Fig. 3.8 (a). Select a visualization in Power BI Desktop |  |
| 13 | Fig. 3.8 (b). Select a visualization in Power BI Desktop – continued |  |
| 14 | Fig. 3.9. Format your visualization in Power BI Desktop |  |
| 15 | Fig. 3.10. Format your visualization details in Power BI Desktop |  |
| 16 | Fig. 4.1. Iris Flower Species, Setosa, Virginica and Versicolor |  |
| 17 | Fig. 4.2. Transform data screen |  |
| 18 | Fig. 4.3. Power Query Editor screen |  |
| 19 | Fig. 4.4. Power Query Editor screen to select quantity of white spaces |  |
| 20 | Fig. 4.5. Quantity of white spaces in the iris dataset |  |
| 21 | Fig. 4.6. Steps to get Power Query Editor |  |
| 22 | Fig. 4.7. Group By summarization |  |
| 23 | Fig. 4.8. Group By options |  |
| 24 | Fig. 4.9. Group By Species to get average Sepal.Length |  |
| 25 | Fig. 4.10. Mean values of Petal.Length, Petal.Width, Sepal.Length and Sepal.Width grouped By Species |  |
| 26 | Fig. 4.11. Median values of Petal.Length, Petal.Width, Sepal.Length and Sepal.Width grouped By Species |  |
| 27 | Fig. 4.12. Create a new table in Power BI |  |
| 28 | Fig. 4.13 (a) Create a table with calculated columns in Power BI |  |
| 29 | Fig. 4.13 (b) Create a table, iris\_petal summary in Power BI with calculated columns and naming the table. |  |
| 30 | Fig. 4.14 Create a table, iris\_sepal summary in Power BI with calculated columns and naming the table. |  |
| 31 | Fig. 4.15. List of tables in Power BI |  |
| 32 | Fig. 4.16. Views in Power BI Desktop |  |
| 33 | Fig. 4.17(a). Page Pane before renaming Page 1 |  |
| 34 | Fig. 4.17(b). Page Pane after renaming Page 1 with Bar Charts |  |
| 35 | Fig. 4.18. Visualization Pane |  |
| 36 | Fig. 4.19. Place holder for Stacked Bar Charts |  |
| 37 | Fig. 4.20. Selecting values for X-axis and Legend for creating a Stacked Bar Chart |  |
| 38 | Fig. 4.21. Selecting median for X-axis and Legend for creating a Stacked Bar Chart |  |
| 39 | Fig. 4.22. Selecting Legend for creating a Stacked Bar Chart |  |
| 40 | Fig. 4.23. Stacked Bar Chart with default format |  |
| 41 | Fig. 4.24 (a). Formatting - Legend in the Stacked Bar Chart |  |
| 42 | Fig. 4.24 (b). Formatting - X-axis in the Stacked Bar Chart |  |
| 43 | Fig. 4.24 (c). Formatting - Data Labels in the Stacked Bar Chart |  |
| 44 | Fig. 4.24 (d). Formatting - Title in the Stacked Bar Chart |  |
| 45 | Fig. 4.25. Stacked Bar Chart for Petal.Length of iris flower species |  |
| 46 | Fig. 4.26(a). Page Pane before renaming Page 1 |  |
| 47 | Fig. 4.26(b). Page Pane after renaming Page 1 as Scatter Plots |  |
| 48 | Fig. 4.27. Select Scatter Plot from Visualization Pane |  |
| 49 | Fig. 4.28. Place holder for Scatter Plot |  |
| 50 | Fig. 4.29(a). Selecting values for X-axis for creating a Scatter Plot |  |
| 51 | Fig. 4.29(b). Selecting values for Y-axis for creating a Scatter Plot |  |
| 52 | Fig. 4.29(c). Selecting values for Legend for creating a Scatter Plot |  |
| 53 | Fig. 4.30. Scatter Plot with default format |  |
| 54 | Fig. 4.31. Scatter Plot showing how to format |  |
| 55 | Fig. 4.32. Scatter Plot for Petal Length and Petal Width |  |
| 56 | Fig. 4.33. Scatter Plot for pair of Length & Width of Petal and Sepal |  |
| 57 | Fig. 4.34 (a). Page Pane before renaming Page 1 |  |
| 58 | Fig. 4.34 (b). Page Pane after renaming Page 1 with Boxplots |  |
| 59 | Fig. 4.35. Visualization pane to select Python visual icon |  |
| 60 | Fig. 4.36(a). Place holder for Boxplot |  |
| 61 | Fig. 4.36(b). Selecting fields for Boxplot |  |
| 62 | Fig. 4.37. Python Script Editor |  |
| 63 | Fig. 4.38. Boxplots for Length & Width for Petal & Sepal |  |
| 64 | Fig. 4.39 (a). Page Pane before renaming Page 1 |  |
| 65 | Fig. 4.39 (b). Page Pane after renaming Page 1 with Histograms |  |
| 66 | Fig. 4.40(a). Selecting fields for Histogram |  |
| 67 | Fig. 4.40(b). Place holder for Histograms |  |
| 68 | Fig. 4.41. Histogram for Petal Length by Species |  |
| 69 | Fig. 4.42. Histogram for Length & Width for Petal & Sepal |  |
| 70 | Fig.5.1. Creating a new Python environment for python 3.6 |  |
| 71 | Fig.5.2. Activate the Python 3.6 environment |  |
| 72 | Fig.5.3. Conda environment list |  |
| 73 | Fig.5.4. Installing pandas package |  |
| 74 | Fig.5.5. Installing matplotlib package |  |
| 75 | Fig.5.6. Setting Python in Power BI - Select File Menu |  |
| 76 | Fig.5.7. Setting Python in Power BI - Options and Settings |  |
| 77 | Fig.5.8. Setting Python in Power BI - Select Python scripting |  |
| 78 | Fig.5.9. Setting Python Home directory |  |
| 79 | Fig.5.10(a). Get Data and More to create Pandas Dataframe |  |
| 80 | Fig.5.10(b). To select Python script to create Pandas Dataframe |  |
| 81 | Fig.5.10(c). To get a text area to key-in the Python script to create Pandas Dataframe |  |
| 82 | Fig.5.10(d). The Python script to create Pandas Dataframe |  |
| 83 | Fig.5.11. Dataframe created by Python script |  |

**Table of Tables**

|  |  |  |
| --- | --- | --- |
| 1 | Tab.2.1. New Functions (Partial list of New DAX functions) |  |
| 2 | Tab.2.2. Aggregation Functions (Partial list of Aggregation DAX functions) |  |
| 3 | Tab.2.3. Date and time Functions (Partial list of Date and Time DAX functions) |  |
| 4 | Tab.2.4. Filter Functions (Partial list) |  |
| 5 | Tab.2.5. Financial Functions (Partial list) |  |
| 6 | Tab.2.6. Table Manipulation Functions (Partial list) |  |
| 7 | Tab. 4.1. Data dictionary for iris dataset |  |
|  |  |  |

# **Introduction to Power BI**

Power BI[[1]](#footnote-1) is an interactive data visualization tool developed by Microsoft with a primary focus on Business Intelligence (BI). Power BI is a collection of software services, apps, and connectors that work together to convert various sources of data (such as database, webpage, PDF, spreadsheets, CSV, XML, JSON files) into static and interactive data visualization. Power BI was originally designed by Ron George[[2]](#footnote-2) in the summer of 2010 and named Project Crescent.

## **Key features of Power BI**

1. **Filtering data[[3]](#footnote-3)**

You can use FILTER to reduce the number of rows in the table that is displayed in a visual based on specific criteria. Various types of filters available in Power BI include the following:

* **Automatic filters:** They are automatically added to the visual level of the filter pane when you build a visual. These filters are based on the fields that made up your visual.
* **Manual filters:** They are the filters that you drag and drop in any section of the filter pane when you're editing a report.
* **Include and exclude filters:** They are automatically added to the filter pane when you use the include or exclude functionality for a visual.
* **Drill-down filters:** They are automatically added to the filter pane when you use the drill-down functionality for a visual in your report.

1. **Data Analytics Expressions (DAX)[[4]](#footnote-4)**

DAX is a formula expression language used in Analysis Services, Power BI and Power Pivot in Excel. DAX formulas include functions, operators, and constants to perform advanced calculations and queries on data in related tables and columns in tabular data models.

1. **Natural Language Query[[5]](#footnote-5)**

The Q&A feature in Power BI enables you to get answers about your data by asking questions using natural language.

1. **Key Performance Indicator (KPI)[[6]](#footnote-6)**

A Key Performance Indicator is a visual cue that communicates the amount of progress made toward a measurable goal. Using KPIs in Power BI helps business make more informed strategic decisions.

1. **Real-time dashboards[[7]](#footnote-7)**

Power BI with real-time streaming helps you stream data and update dashboards in real time. Any visual or dashboard created in Power BI can display and update real-time data and visuals.

## **Why Power BI?[[8]](#footnote-8)**

1. **Easy to use as Power BI follows an approach similar to that of Excel.**
2. **Easy to collaborate[[9]](#footnote-9)**

After you create a Power BI Desktop file, if you publish it to your Power BI app workspace, then your team members in your workspace can collaborate on it.

1. **Cost Effective**

Power BI Desktop is always free[[10]](#footnote-10). The Power BI service offers both free and paid license options. When compared to Power BI, Tableau, another interactive data visualization tool from Salesforce is expensive[[11]](#footnote-11).

1. **Good demand for Power BI skills[[12]](#footnote-12)**

Most organizations use Power BI because of the credibility of Microsoft Power BI and it is cost effective. Hence the demand for a Power BI programmer will be ever-increasing.

## **Two basic elements of Power BI**

Power BI provides[[13]](#footnote-13) cloud-based BI services, known as "Power BI Services", along with a desktop-based interface, known as "Power BI Desktop". Power BI Desktop is always **free**, while Power BI Service offers both free and paid license options.



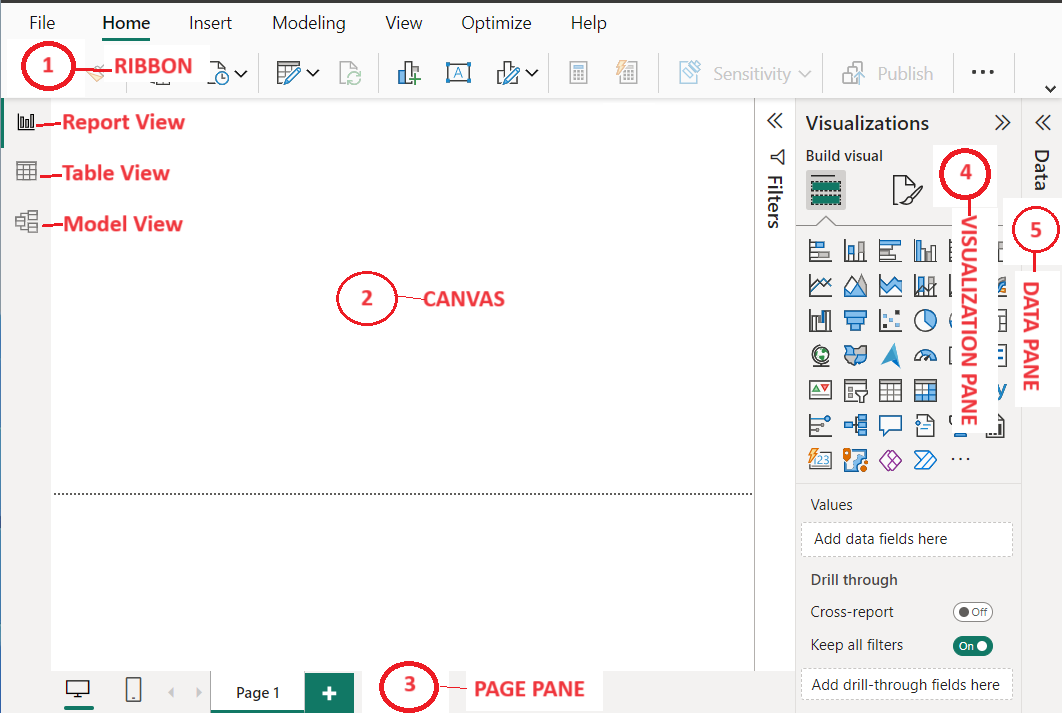
### **Power BI Desktop**

Power BI[[14]](#footnote-14) is a complete data analysis and report creation tool that is used to connect to, transform, visualize, and analyze the data. It includes the Query Editor, in which you can connect to many different sources of data, and combine them (or perform modeling) into a data model to be used to design a report. Reports can be shared by publishing to the Power BI service or directly with others.

### **Power BI Services**

The Power BI service is a cloud-based service, or software as a service (SaaS). It supports report editing and collaboration for teams and organizations. For more information on Power BI service, you may refer to <https://learn.microsoft.com/en-us/power-bi/fundamentals/power-bi-service-overview>.

## **Power BI Terminologies**

****

**Fig. 1.1.** Power BI Terminologies

Fig. 1.1 shows the basic terminologies used in Power BI and are explained below:

1. **Ribbon**

Ribbon [[15]](#footnote-15)in Power BI is where you can perform various actions such as get data etc. You may observe that the actions available from ribbon vary according to your selection.

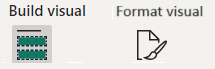
1. **Canvas**

The report canvas is where your visualizations are created and displayed.

1. **Page Pane**

Each tab at the bottom of the canvas represents a page in the report and you can rename the page to suit your needs.

1. **Visualization Pane**



**Fig. 1.2.** Visualization Pane Tabs

Visualizations pane as shown in Fig. 1.2 identifies the type of visual in use and has three tabs:

a) Build visual, where you add data to your visual

b) Format visual, where you format the report page

1. **Data Pane**

The Data pane lists all the available tables with fields in the data model.

1. **M Query Language**

M Query Language in Power BI [[16]](#footnote-16) is a mashup query language that can be used to query a lot of data from one or more supported sources.

There are three views[[17]](#footnote-17) in Power BI, namely Report view, Table view and Model view.

1. Report view: Here, you can create any number of pages with visualizations.
2. Table view: This helps you inspect, explore, and understand data in Power BI Desktop model.
3. Model view: This shows all of the table, columns, and relationships in the model.

# **Data Analysis Expressions (DAX)**

DAX[[18]](#footnote-18) is a library of functions and operators that can be combined to build formulas and expressions in Power BI and Power Pivot in Excel data models.



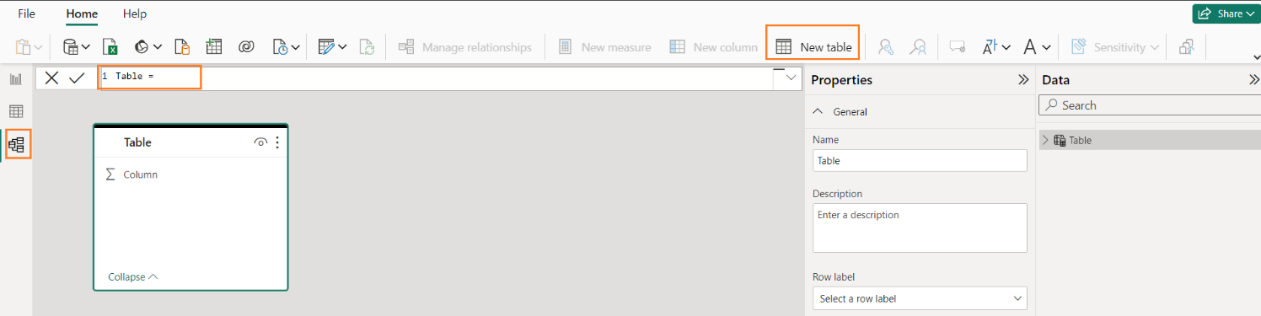
### **New DAX functions**

New and updated functions are typically first introduced in Power BI desktop, and then in Power Pivot in Excel, and tools.

New functions[[19]](#footnote-19) include the following as given in the Tab.2.1:

|  |  |
| --- | --- |
| **Function** | **Description** |
| MATCHBY | Defines the columns that are used to match data and identify the current row, in a window function operation |
| RANK | Returns the ranking for the current context within the specified partition, sorted by the specified order. |
| LINEST | Uses the Least Squares method to calculate a straight line that best fits the given data. |
| INDEX | Returns a row at an absolute position, specified by the position parameter, within the specified partition, sorted by the specified order or on the specified axis. |
| WINDOW | Returns multiple rows which are positioned within the given interval. |
| TOCSV | Returns a table as a string in CSV format. This function applies to Power BI Desktop only. |
| TOJSON | Returns a table as a string in JSON format. This function applies to Power BI Desktop only. |
| NETWORKDAYS | Returns the number of whole workdays between two dates. |

**Tab.2.1.** New Functions (Partial list of New DAX functions)



**Fig. 2.1(a)** Creating a new table to show the working of the new function, NETWORKDAYS

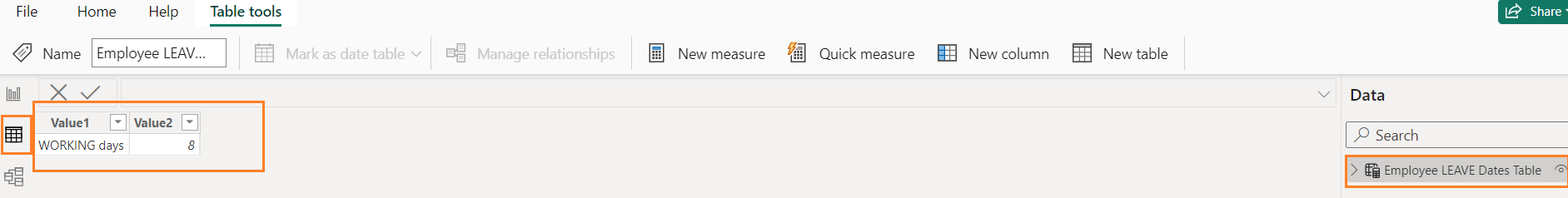
Paste the following function in the expression,1 Table = as follows:

Employee LEAVE Dates Table = {

("WORKING days", NETWORKDAYS(Date(2023,11,1), Date(2023,11,10),1))

}

Now, go to Table view as shown in Fig.2.1(b).



**Fig. 2.1(b).** Illustration of the function, NETWORKDAYS

Number of days between Wednesday. 1st Nov 2023 and Tuesday, 10th Nov 2023 is 8 excluding Saturdays and Sundays.

Syntax for NETWORKDAYS[[20]](#footnote-20)

**NETWORKDAYS (<start date>, <end date> [, <weekend>, <holidays>])**

**Weekend indicator values**

1 - Sat, Sun (Default)

2 - Sun, Mon

3 - Mon, Tue

4 - Tue, Wed

5 - Wed, Thu

6 - Thu, Fri

7 - Fri, Sat

11 - Sun

12 - Mon

13 - Tue

14 - Wed

15 - Thu

16 - Fri

17 - Sat

**Holidays**

A column table of one or more dates that are to be excluded from the working day calendar.

### **Aggregation Functions**

Aggregation functions[[21]](#footnote-21) include the following as given in the Tab.2.2:

|  |  |
| --- | --- |
| **Function** | **Description** |
| AVERAGE | Returns the average (arithmetic mean) of all the numbers in a column. |
| AVERAGEX | Returns the average (arithmetic mean) of a set of expressions evaluated over a table. |
| COUNT | Counts the number of rows in the specified column that contain non-blank values. |
| COUNTAX | Counts non-blank results when evaluating the result of an expression over a table. |
| COUNTROWS | Counts the number of rows in the specified table or in a table defined by an expression. |
| MAX | Returns the largest value in a column |
| MAXX | Evaluates an expression for each row of a table and returns the largest numeric value. |
| MIN | Returns the smallest numeric value in a column, or between two scalar expressions. |
| MINX | Returns the smallest numeric value that results from evaluating an expression for each row of a table. |
| PRODUCT | Returns the product of the number in a column. |
| PRODUCTX | Returns the product of an expression evaluated for each row in a table. |
| SUM | Add all the numbers in a column. |
| SUMX | Returns the sum of an expression evaluated for each row in a table. |

**Tab.2.2.** Aggregation Functions (Partial list of Aggregation DAX functions)

### **Date and Time Functions**

These DAX functions[[22]](#footnote-22) use a datetime data type to create calculations based on dates and time and include the following as given in the Tab.2.3:

|  |  |
| --- | --- |
| **Function** | **Description** |
| CALENDAR | Returns a table with a single column named "Date" that contains a contiguous set of dates. |
| DATE | Returns the specified date in datetime format. |
| NOW | Returns the current date and time in datetime format. |
| TODAY | Returns the current date. |
| YEARFRAC | Calculates the fraction of the year represented by the number of whole days between two dates. |

**Tab.2.3.** Date and time Functions (Partial list of Date and Time DAX functions)

### **Filter functions**

The filter functions[[23]](#footnote-23) in DAX are very powerful and work on Table and relationships to create dynamic calculations.

Some of these functions are given below in the Tab.2.4:

|  |  |
| --- | --- |
| **Function** | **Description** |
| Calculate | Evaluates an expression in a modified filter context. |
| Calculatetable | Evaluates a table expression in a modified filter context. |
| Filter | Returns a table that represents a subset of another table or expression. |
| Removefilters | Clears filters from the specified tables or columns. |
| All | Returns all the rows in a table, or all the values in a column, ignoring any filters that might have been applied. |

**Tab.2.4.** Filter Functions (Partial list)

Syntax for the function, **CALCULATE**

CALCULATE(<expression>[, <filter1>, [, <filter2>,[, ......]]])

where expression is the expression to be evaluated, filter1, filter2 (optional) Boolean expressions or table expressions that defines filters, or filter modifier functions.

### **Financial functions**

Financial functions[[24]](#footnote-24) in DAX are used in formulas that perform financial calculations, such as net present value and rate of return.

Some of these functions are given below in the Tab.2.5:

| **Function** | **Description** |
| --- | --- |
| ACCRINT | Returns the accrued interest for a security that pays periodic interest. |
| ACCRINTM | Returns the accrued interest for a security that pays interest at maturity. |
| DB | Returns the depreciation of an asset for a specific period using the fixed declining balance method |
| DDB | Returns the depreciation of an asset for a specific period using the double declining balance method or some other method you specify. |
| PMT | Calculates the payment for a loan based on constant payments and a constant interest rate. |
| PV | Calculates the present value of a loan or an investment, based on a constant interest rate. |
| YIELD | Returns the yield on a security that pays periodic interest |

**Tab.2.5.** Financial Functions (Partial list)

### **Table manipulation functions**

These functions[[25]](#footnote-25) return a table or manipulate existing tables.

Some of these functions are given below in the Tab.2.6:

|  |  |
| --- | --- |
| **Function** | **Description** |
| ADDCOLUMNS | Adds calculated columns to the given table or table expression. |
| ADDMISSINGITEMS | Adds combination of items from multiple columns to a table if they do not already exist. |
| DATATABLE | Provides a mechanism for declaring an inline set of data values. |
| ROLLUP | Modifies the behavior of SUMMARIZE by adding rollup rows to the result on columns defined by the groupBy\_columnName parameter. |
| SELECTCOLUMNS | Adds calculated columns to the given table or table expression. |
| SUMMARIZE | Returns a summary table for the requested totals over a set of groups. |
| SUMMARIZECOLUMNS | Returns a summary table over a set of groups. |
| VALUES | Returns a one-column table that contains the distinct values from the specified table or column. |

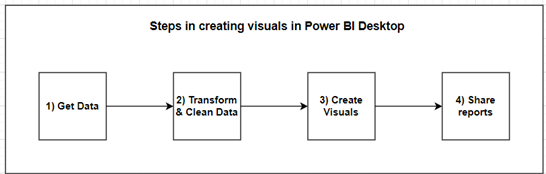
**Tab.2.6.** Table Manipulation Functions (Partial list)

.

.

# **Steps to create visualizations in Power BI**

To create a visual in Power BI, follow the steps below as shown in Fig. 3.1.



**Fig. 3.1.** Steps to create a visual in Power BI

**Steps:**

1) Get Data by connecting to data source.

2) Transform and clean that data, to create a data model.

3) Create visuals, such as charts or graphs, that provide visual representations of the data on one or more report pages.

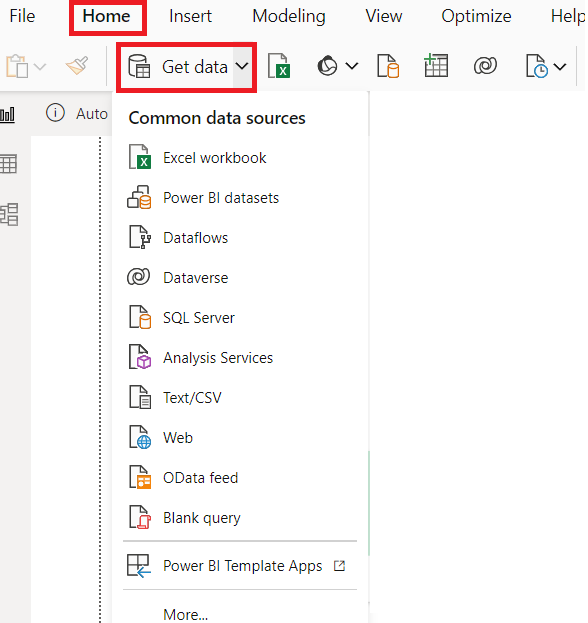
4) Share reports with others by sharing the pbix file or using the Power BI service.

Each of these steps are explained below:



## **Get Data**

To connect to data, from the Home ribbon, select Get data as shown in Fig. 3.2.

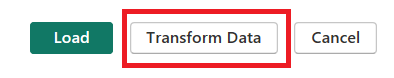


**Fig. 3.2.** Get Data in Power BI

From the Get Data window, you can choose from many different data sources such as Excel Workbook, Text / CSV, SQL Server or web as per your requirement.

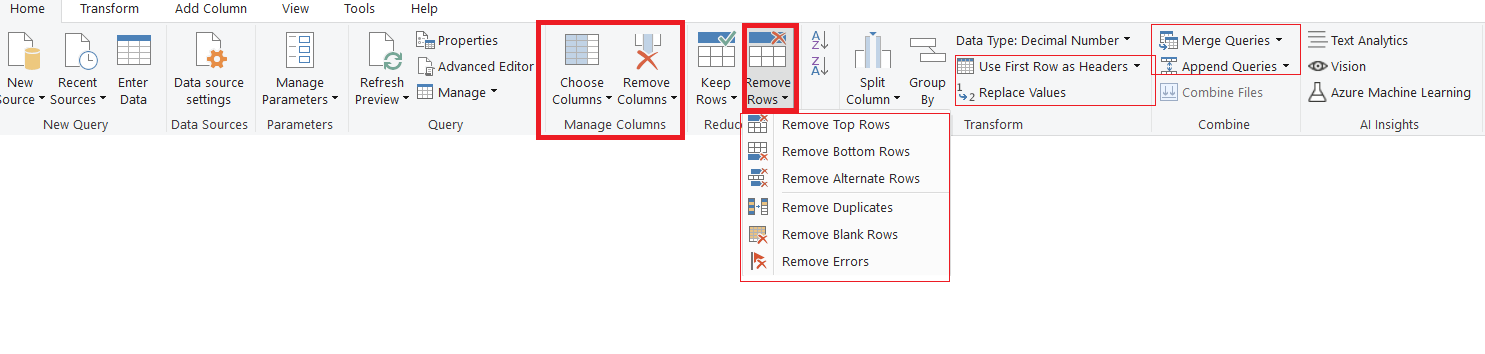
## **Transform & Clean Data**

Transformations include removing a column, filtering rows, treat the first row as table header, merge, append datasets, replace null or any value with another value or formula, perform group by, pivot, and unpivot operations on datasets. After connecting to the data source, you can clean and transform the data by clicking on **"Transform Data"** tab as shown In Fig.3.3.



**Fig. 3.3.** Transform Data in Power BI

Once you click on Transform Data tab, the following Fig. 3.4 appears on the screen:

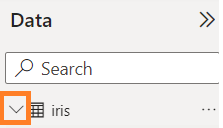


**Fig. 3.4.** Transform Data tab options in Power BI

After performing necessary transformations, click on **Load** tab to import the data into Power BI Desktop.

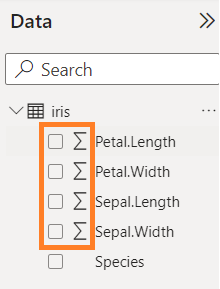
## **Inspect dataset loaded into Power BI**

Click on the Data Pane and click on the iris table to show all its fields as shown in Fig. 3.5.



**Fig. 3.5.** Table under Data Pane

The following image is displayed:

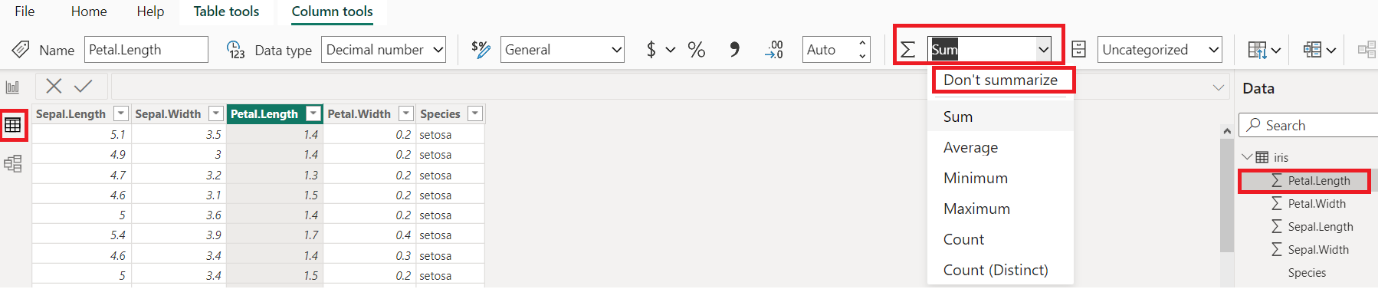


**Fig. 3.6.** Table with fields under Data Pane

You may observe the Sigma sign (Σ) before the numeric columns in Fig. 3.6. By default, **numeric columns** will **summarize** by summing values together[[26]](#footnote-26).

For these four numeric columns, we need click the drop-down on the field and choose **Don’t summarize**.

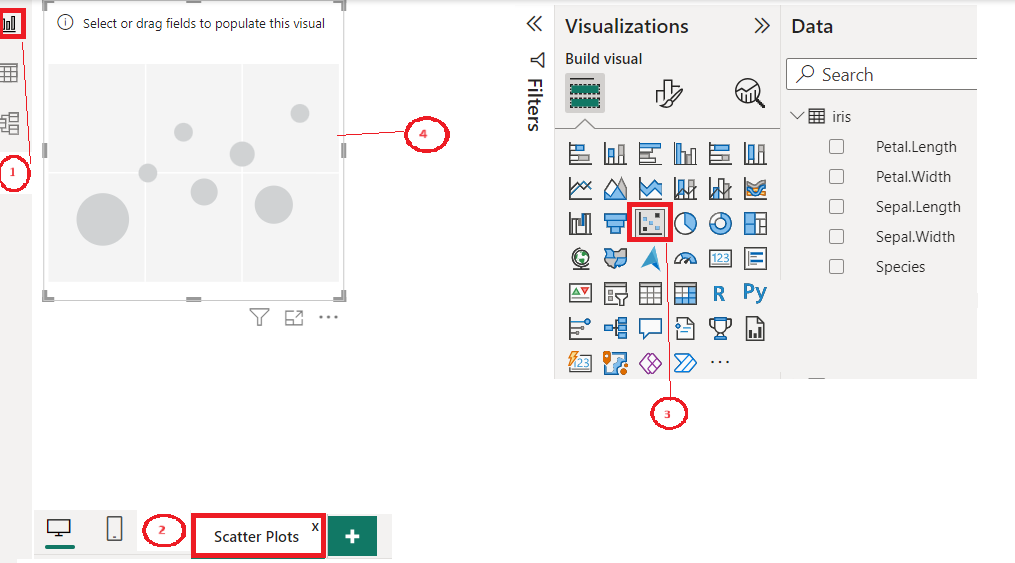
Go to “Table view”, click on the first numeric column,”Petal.Length” and choose the "Don't summarize" option as shown in Fig.3.7.



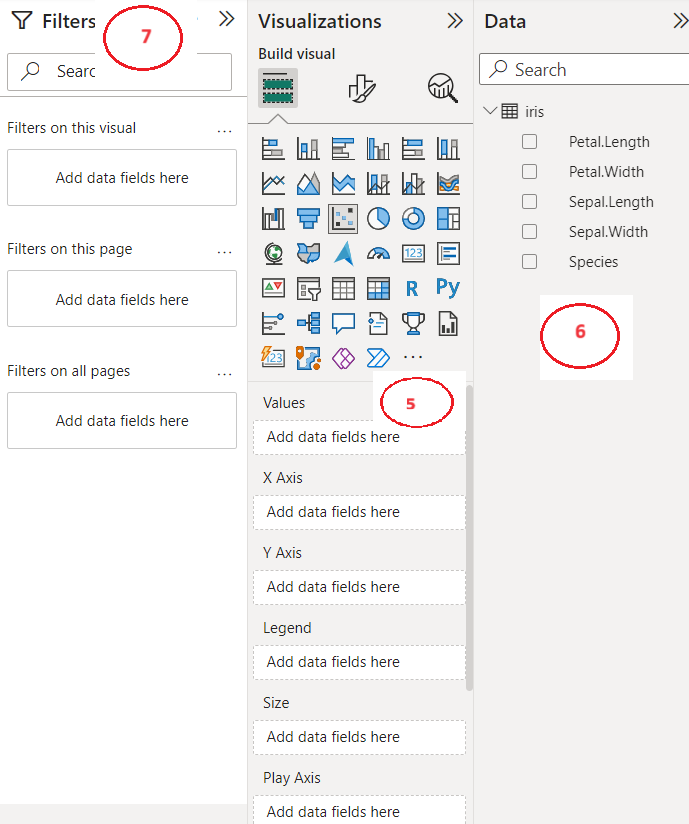
**Fig. 3.7.** Select the Don’t Summarize option for the chosen numeric column

Similarly, we need to turn off the Summarize option for other numeric columns such as Petal.Width, Sepal.Length and Sepal.Width.

## **Create visuals on report pages**



**Fig. 3.8 (a).** Select a visualization in Power BI Desktop



**Fig. 3.8 (b).** Select a visualization in Power BI Desktop - continued

**Fig. 3.8** (a) and (b) shows the steps involved in creating visuals on a report page in Power BI.

**Steps involved in creating visuals on a report page**

1) Click on the Report view on the left to invoke the Report editor.

2) Select the page to place the visualization to be created. Click either

a) on the + icon and rename the page name to a suitable name, say ‘Scatter Plots” by double clicking on Page 2 and overwriting Page 2.

**or**

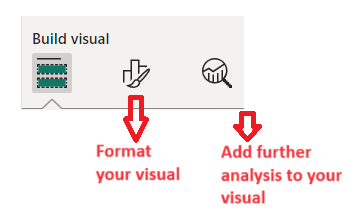
b) on “Page 1” and rename the page name suitably.

3) Select a visualization (say, Scatter Plot) from the list of standard list of visualization charts depending on the business requirements.

4) A placeholder of it is created on the canvas.

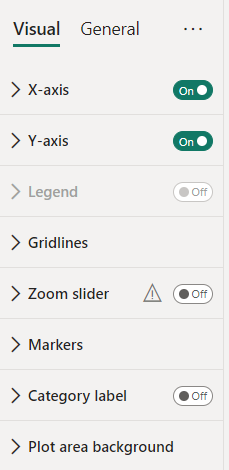
5 & 6) From the Data pane, select the dataset and fields for x-axis, values, y-axis, legend etc. based on the visualization chosen.

7) Filter pane is used to limit the data in the visualization.



**Fig. 3.9.** Format your visualization in Power BI Desktop

Click on the Format your visual icon in Fig.3.9 to get the following screen, Fig.3.10. You can change the font size, color, and display unit of Y-Axis / X-axis, set data type - continuous or categorical, range (min, max) etc.



**Fig. 3.10.** Format your visualization details in Power BI Desktop

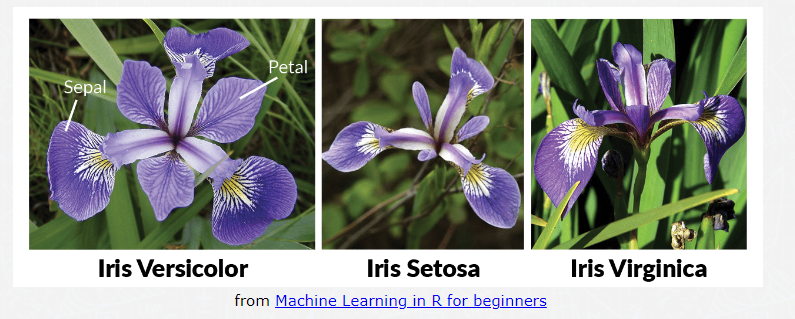
## **Share reports**

After successfully creating the visualizations report, you can share the PBIX file or publish your Power BI report to web using Power BI Service or embed your Power BI report in SharePoint.

# **Use case: Iris Dataset visualization**

# **Introduction**

The Iris dataset contains three classes of flowers, Versicolour, Setosa, Virginica, and each class contains 4 features (length and width of sepals and petals) of 50 samples[[27]](#footnote-27). We are required to predict the class of the flowers based on their specific features.

****

**Fig. 4.1.** Iris Flower Species, Setosa, Virginica and Versicolor

# **Data**

We shall use the Iris Dataset from the UCI Machine Learning Repository. Information about the original paper and usage of the dataset can be found in the UCI Machine Learning Repository -- Iris Dataset[[28]](#footnote-28). The dataset contains 150 rows and 5 columns. Table Tab.4.1 shows the data dictionary of the iris dataset.



### **Data Dictionary**

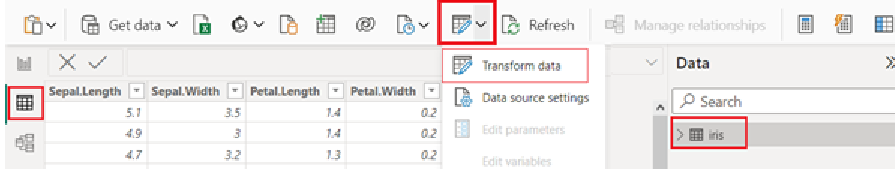
|  |  |  |
| --- | --- | --- |
| **Name** | **Data type** | **Description** |
| Sepal Length | Numeric | Independent variable -- Measured in cm. |
| Sepal Width | Numeric | Independent variable -- Measured in cm. |
| Petal Length | Numeric | Independent variable -- Measured in cm. |
| Petal Width | Numeric | Independent variable -- Measured in cm. |
| Species | Factor | w/3 levels, “Versicolour”, “Setosa” & “Virginica” |

**Tab. 4.1.** Data dictionary for iris dataset

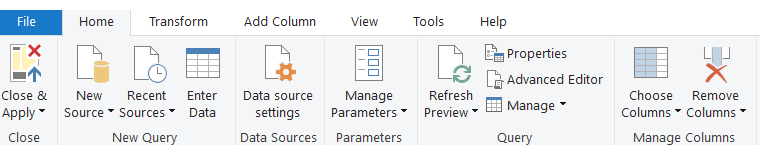
### **Data understanding**

**Presence of Null values:**

To get the Power Query Editor, select **Table** View and **Power Query icon** on the ribbon and select **Transform data** tab as shown in Fig. 4.2.

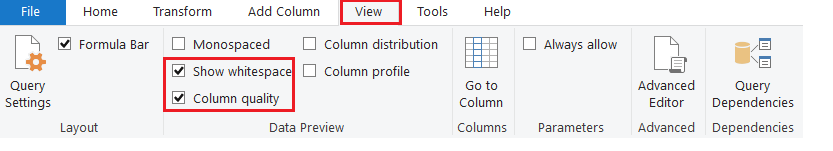


**Fig. 4.2.** Transform data screen

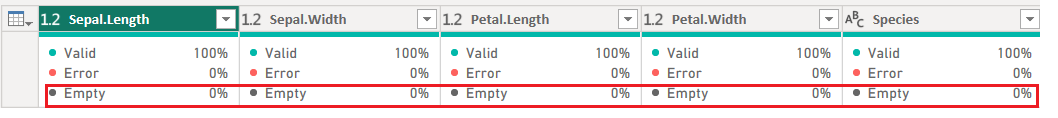


**Fig. 4.3.** Power Query Editor screen

Refer Fig. 4.3. To get the count of null values in the data set, use the Power Query Editor, choose **View** tab of the ribbon, and select by clicking on "**Show whitespace**", "**Column quantity**" as shown in Fig. 4.4.



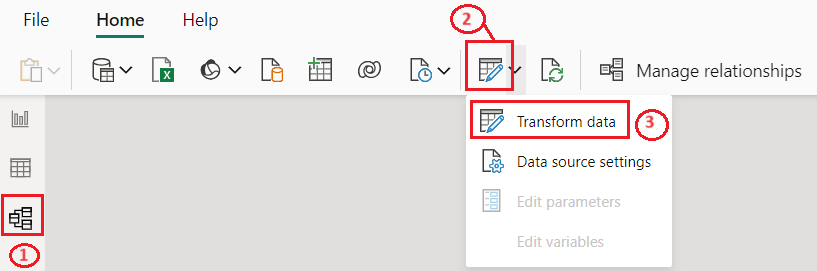
**Fig. 4.4.** Power Query Editor screen to select quantity of white spaces



**Fig. 4.5.** Quantity of white spaces in the iris dataset

**From Fig. 4.5, we observe that there are no missing values in the dataset since all the five columns have 0% as the percentage of empty values.**

We need to understand the location and dispersion of the data[[29]](#footnote-29). The measures such as mean and median, the measures of central tendency are used to estimate normal values of a dataset. Standard deviation, a measure of dispersion is important for describing the spread of the data or its variation around mean, a central value. We need to calculate the mean, median and standard deviation of the four features, Petal Length, Petal Length, Sepal Length and Sepal Width for each level of three species of Iris flower (Setosa, Virginica and Versicolor).



**Fig. 4.6.** Steps to get Power Query Editor

As shown in Fig. 4.6. you need to do the following:

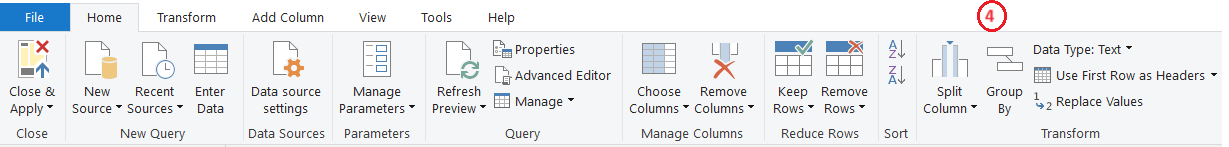
1) Click on the Model View tab.

2) Click on the Power Query Editor icon.

3) Click on Transform data tab

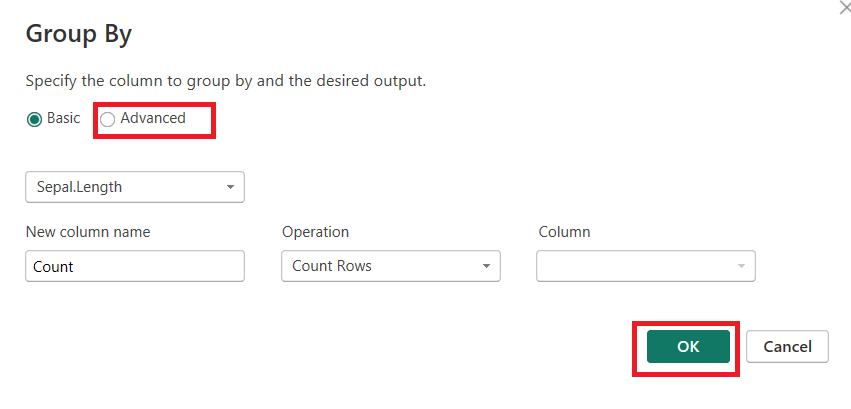
To get the location and dispersion of data, we can use “Group By” function in Power BI which summarizes the data based on one or more aggregate functions such as average (mean), median, standard deviation.

4) Click on the **Group By** icon as shown in Fig. 4.7.



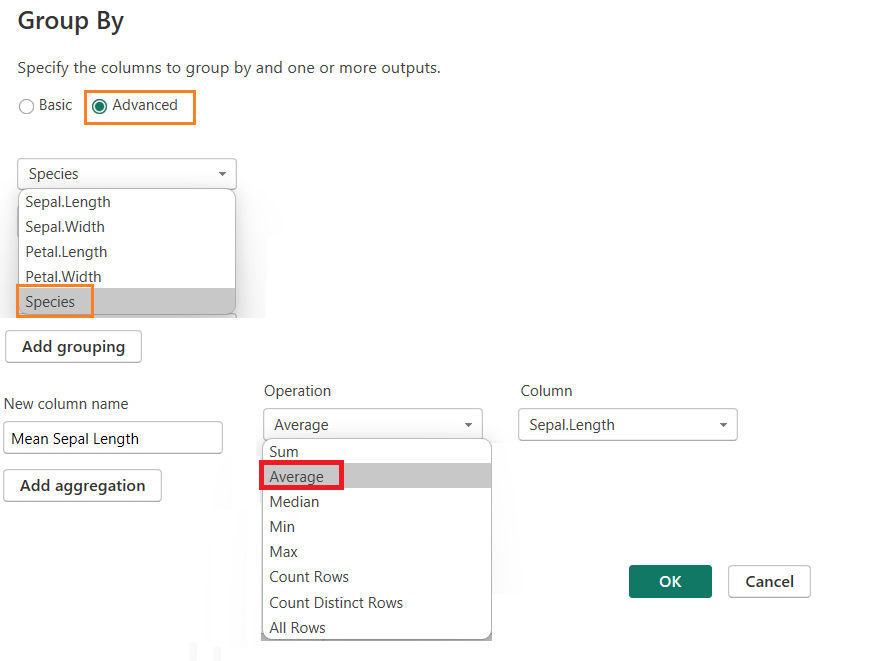
**Fig. 4.7.** Group By summarization

We will see the screen, Fig. 4.8. as shown below:



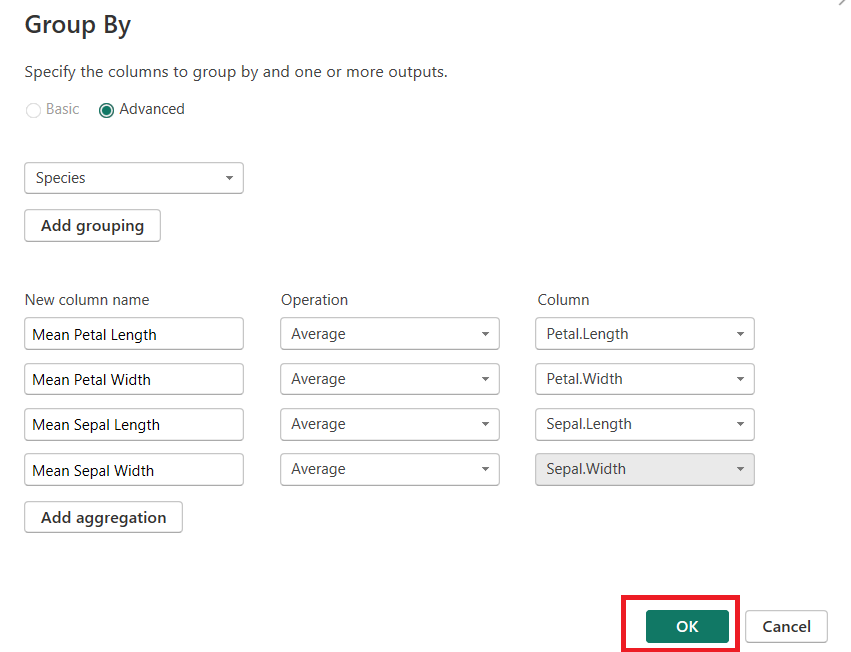
**Fig. 4.8.** Group By options

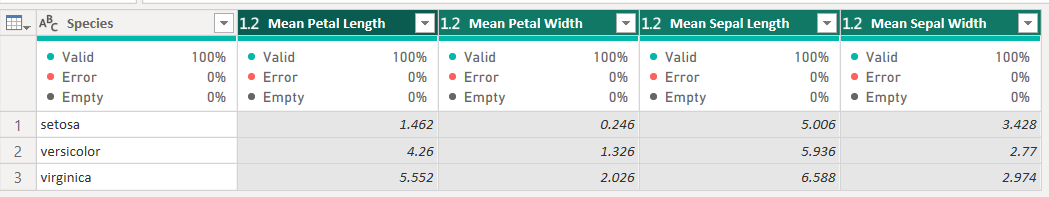
4) Click on the Advanced radio button, select Species in the dropdown box placed below the radio button option, select Species for grouping, type Average as the new column name, select Average from the dropdown box for operation and select Sepal.Length under column as shown in Fig.4.9.



**Fig. 4.9.** Group By Species to get average Sepal.Length

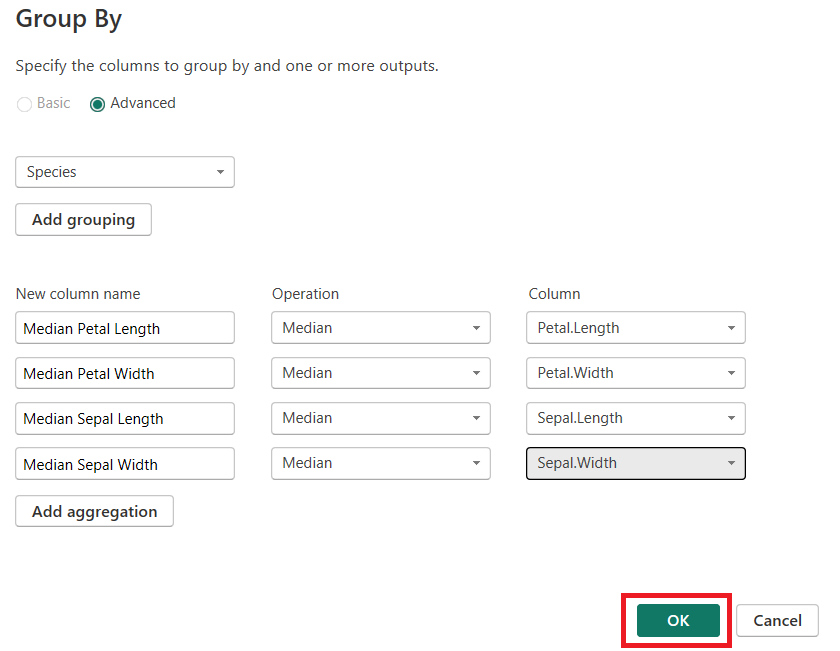
Similarly, we repeat the same process for other features, Petal.Length, Petal.Width and Sepal.Width and you will see the following screen as shown in Fig.4.10.

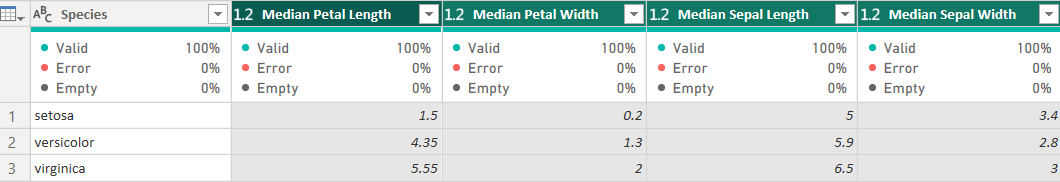




**Fig. 4.10.** Mean values of Petal.Length, Petal.Width, Sepal.Length and Sepal.Width grouped By Species

Similarly, we calculate median values for each of the species using the Group By function of Power BI by choosing the Median as the operation as shown in Fig. 4.11.





**Fig. 4.11.** Median values of Petal.Length, Petal.Width, Sepal.Length and Sepal.Width grouped By Species

We observe that the median length and width values of Petal and Sepal of the Iris flower are differently located for the Species, Setosa than others such as Versicolor and Virginica. We shall look at the standard deviation of length and width values of Petal and Sepal of the Iris flower.

**Creating a Power BI Table**

A Power BI Table is a grid with related data organized in a logical series of rows and columns. The header and row totals are included in the Power BI Table. Here, we are looking at measures of central tendency and dispersion such as mean, median and standard deviation of length and width of Petal and Sepal of three iris flowers, namely Setosa, Virginica and Versicolor.

Using SUMMARIZE function, we can create a summary table for the requested measure over a set of groups.

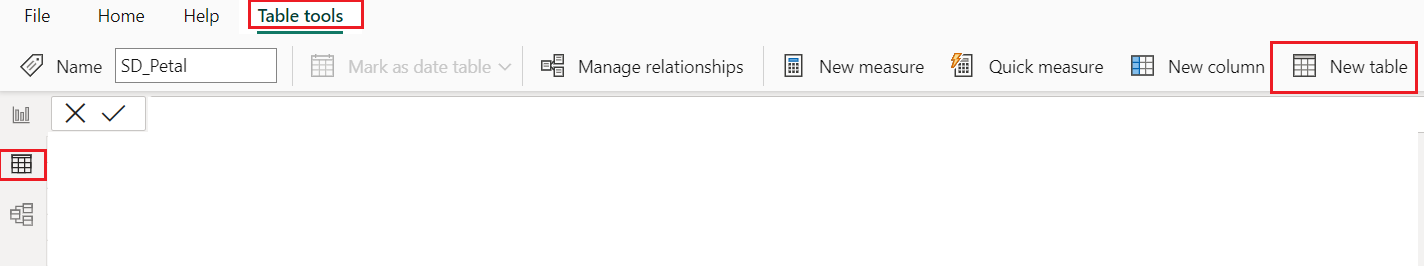
**Syntax**

SUMMARIZE(<table>, <group by column name>, <group by column name>,..., [, <name>, <expression>] ...)

Here expression = Any DAX expression that returns a single scalar value, where the expression is to be evaluated multiple times (for each row/context).

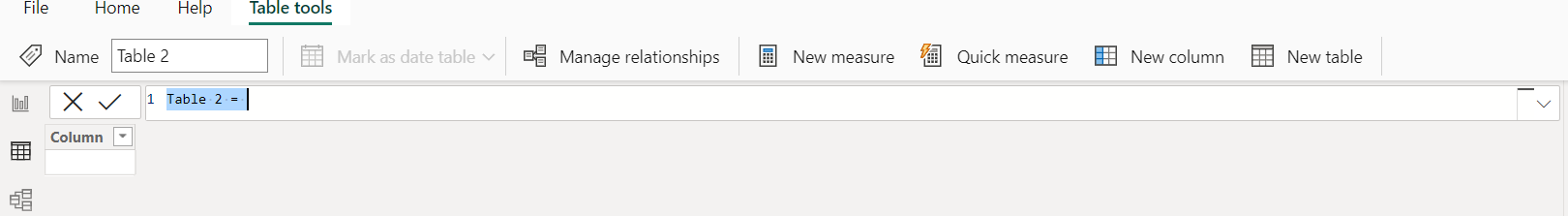
**Create a table, iris\_petal summary giving the measures such as mean, median and standard deviation of Petal.Length and Petal.Width**

As shown below in Fig. 4.12, click on the Table View icon on the Left; click on New Tools icon on the top; click on New Table located under the menu items.



**Fig. 4.12.** Create a new table in Power BI

You will see the following screen:



**Fig. 4.13** (a) Create a table in Power BI by giving required details

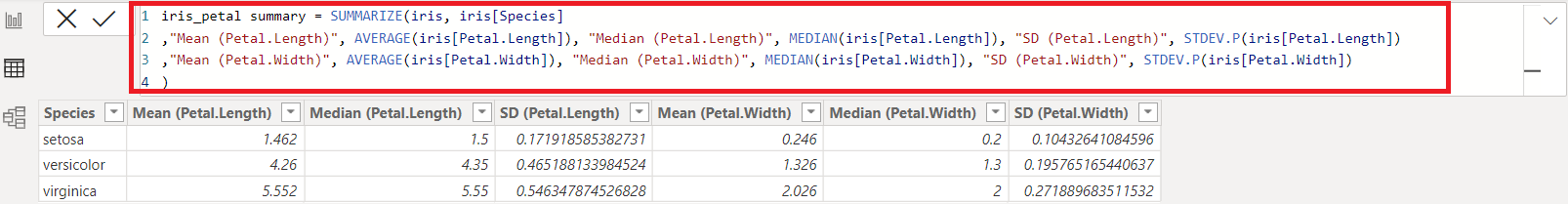
In Fig. 4.13 (a), replace the following text in the line, ”Table 2 = “:

iris\_petal summary = SUMMARIZE(iris, iris[Species]

,"Mean (Petal.Length)", AVERAGE(iris[Petal.Length]), "Median (Petal.Length)", MEDIAN(iris[Petal.Length]), "SD (Petal.Length)", STDEV.P(iris[Petal.Length])

,"Mean (Petal.Width)", AVERAGE(iris[Petal.Width]), "Median (Petal.Width)", MEDIAN(iris[Petal.Width]), "SD (Petal.Width)", STDEV.P(iris[Petal.Width])

)



**Fig. 4.13** (b) Create a table, iris\_petal summary in Power BI with calculated columns and naming the table.

**Create a table, iris\_sepal summary giving the measures such as mean, median and standard deviation of Sepal.Length and Sepal.Width**

Again, click on New Table located under the menu items as shown in Fig. 4.12.

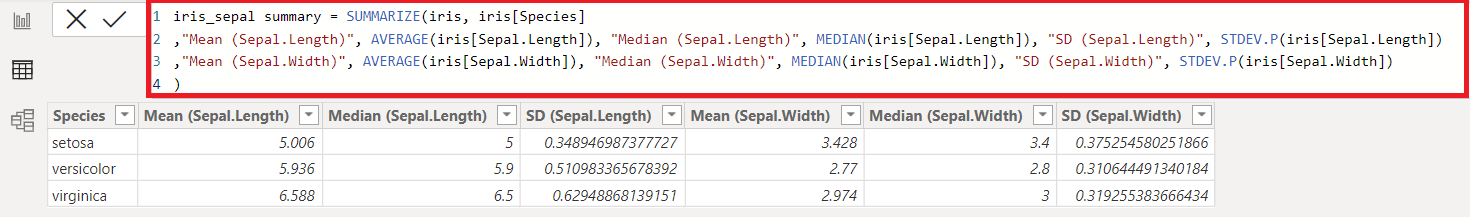
Replace the following text in the line, ”Table = “:

iris\_sepal summary = SUMMARIZE(iris, iris[Species]

,"Mean (Sepal.Length)", AVERAGE(iris[Sepal.Length]), "Median (Sepal.Length)", MEDIAN(iris[Sepal.Length]), "SD (Sepal.Length)", STDEV.P(iris[Sepal.Length])

,"Mean (Sepal.Width)", AVERAGE(iris[Sepal.Width]), "Median (Sepal.Width)", MEDIAN(iris[Sepal.Width]), "SD (Sepal.Width)", STDEV.P(iris[Sepal.Width])

)

****

**Fig. 4.14** Create a table, iris\_sepal summary in Power BI with calculated columns and naming the table.

|  |  |
| --- | --- |
| **You will see these two tables, listed as shown here:**  **“iris\_petal\_summary”, “iris\_sepal\_summary”** |  |

**Fig. 4.15.** List of tables in Power BI

# **Visualizations using Power BI**

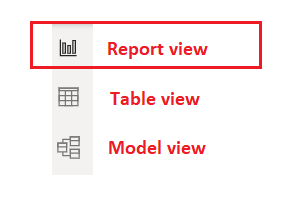
Visualizations display insights that have been discovered in the data. A Power BI report might have a single page with one visual or it might have pages full of visuals. We shall create separate pages for Boxplots, Scatter Plots, Histograms and Bar Charts.

Bar charts, boxplots, scatter plots, and histograms are different types of graphs that can be used to display data. Here, bar chart is used to represent measures such as count, means, medians, etc.) grouped by a categorical variable. Both histograms and box plots[[30]](#footnote-30) [[31]](#endnote-1)provide a visual means to describe the central tendency and dispersion, presence of gaps, outliers or unusual data points. Scatter plots show the relationship between two numeric variables.

Using Power BI Visualizations, we can create both Bar charts and scatter plots. Unfortunately, Power BI does not officially offer the box plot or histogram visualization tool. However, we can still acquire visualization extensions from third-party developers. We shall use Python Script to create Box plots. By acquiring visualization extension, we can create histogram.

**How to create a Report?**

1. Click on the Report view on the left to invoke the Report Editor as shown below in Fig. 4.16:



**Fig. 4.16.** Views in Power BI Desktop

Pages appear in the navigation area at the bottom of the Report view. You can create a new page by selecting the + icon next to the pages in the navigation area below the canvas.

**Bar chart**

A bar chart [[32]](#footnote-31)is a graph with rectangular bars. The graph usually compares different categories. Although the graphs can be plotted vertically (bars standing up) or horizontally (bars laying flat from left to right), the most usual type of bar graph is vertical.

The horizontal (x) axis represents the categories; The vertical (y) axis represents a value for those categories. In the graph below, the values are median of Petal Length of three species of iris flower.

Create a page for Bar charts by clicking on the + icon and rename the page name to “Bar charts” by double clicking on Page 1 and overwriting Page 1 as shown in Fig.4.17 (a) and (b).

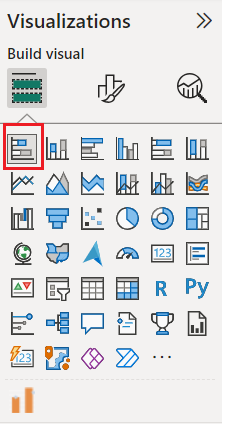


**Fig. 4.17 (**a). Page Pane before renaming Page 1



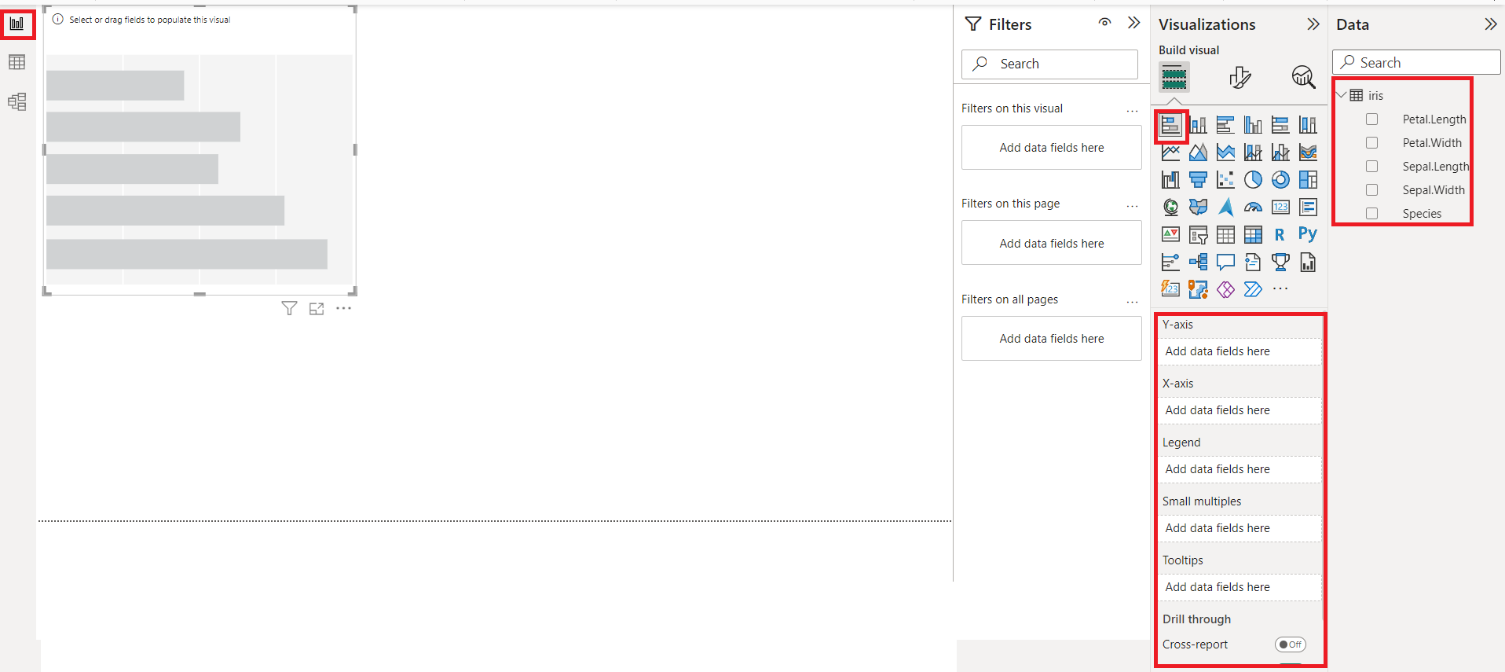
**Fig. 4.17** (b). Page Pane after renaming Page 1 with Bar Charts

1. With the report view selected, on page, “Bar Charts”, from visualizations, select the first chart, Stacked bar chart.



**Fig. 4.18.** Visualization Pane

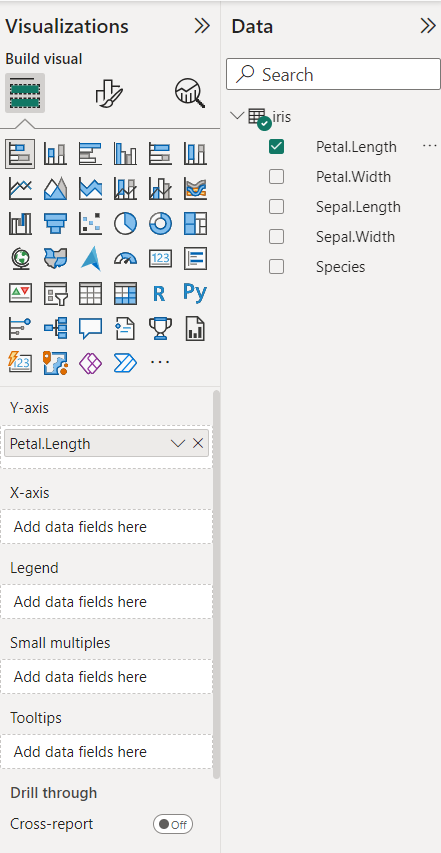
1. You will see the place holder for creating a stacked barcharts as shown in Fig. 4.19.



**Fig. 4.19**. Place holder for Stacked Bar Charts

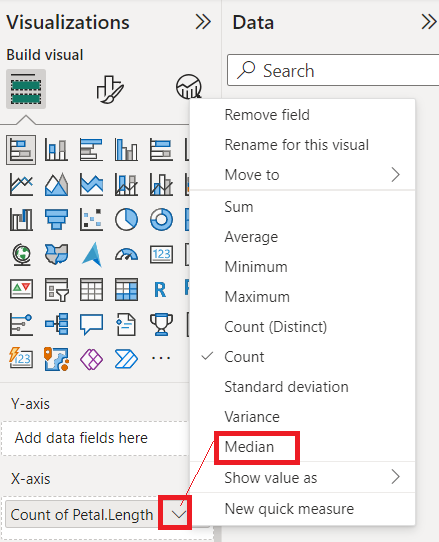
With the bar chart place holder selected on the canvas, select the following fields and make changes as shown below:

1. Select the field, Petal.Length from the iris table as shown in Fig. 4.20.



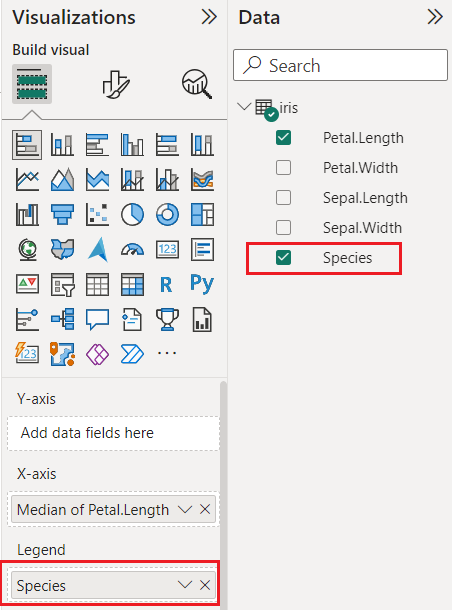
**Fig. 4.20.** Selecting values for X-axis and Legend for creating a Stacked Bar Chart

1. Push the Petal.Length from Y-axis to X-axis and choose Median from the list of values instead of Count as shown in Fig.4.21.

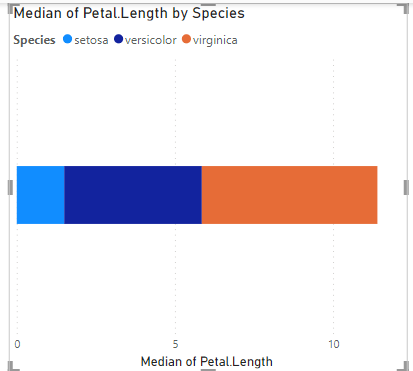


**Fig. 4.21.** Selecting median for X-axis

1. Select the field, Species from the iris table and push it from Y-axis to Legend as shown in Fig. 4.22 to have different colors of bars for each of the Species.



**Fig. 4.22.** Selecting Legend for creating a Stacked Bar Chart

****

**Fig. 4.23.** Stacked Bar Chart with default format

**A Stacked Bar Chart with the default formatting is shown as in Fig. 4.23.**

**Formatting: With the bar chart selected on the canvas, select “format your page” icon to set the formatting.**

1. Legend

|  |  |
| --- | --- |
|  | ***Make the following changes:***  ***i) Position: Top Right***  ***ii) Color: Black***  ***iii) Font: Trebuchet MS***  ***iv) Font Size: 12*** |

**Fig. 4.24 (a).** Formatting - Legend in the Stacked Bar Chart

1. X-Axis

|  |  |
| --- | --- |
|  | ***Make the following changes:***  ***i) Position: Top Right***  ***ii) Color: Black***  ***iii) Font: Trebuchet MS***  ***iv) Font Size: 12***  ***v) Turn Title Off by clicking on it*** |

**Fig. 4.24 (b).**  Formatting - X-axis in the Stacked Bar Chart

1. Data Labels

|  |  |
| --- | --- |
|  | ***Make the following changes:***  ***i) Data Labels: Turn ON by clicking on it***  ***ii) Color: Black***  ***iii) Font: Trebuchet MS***  ***iv) Font Size: 12*** |

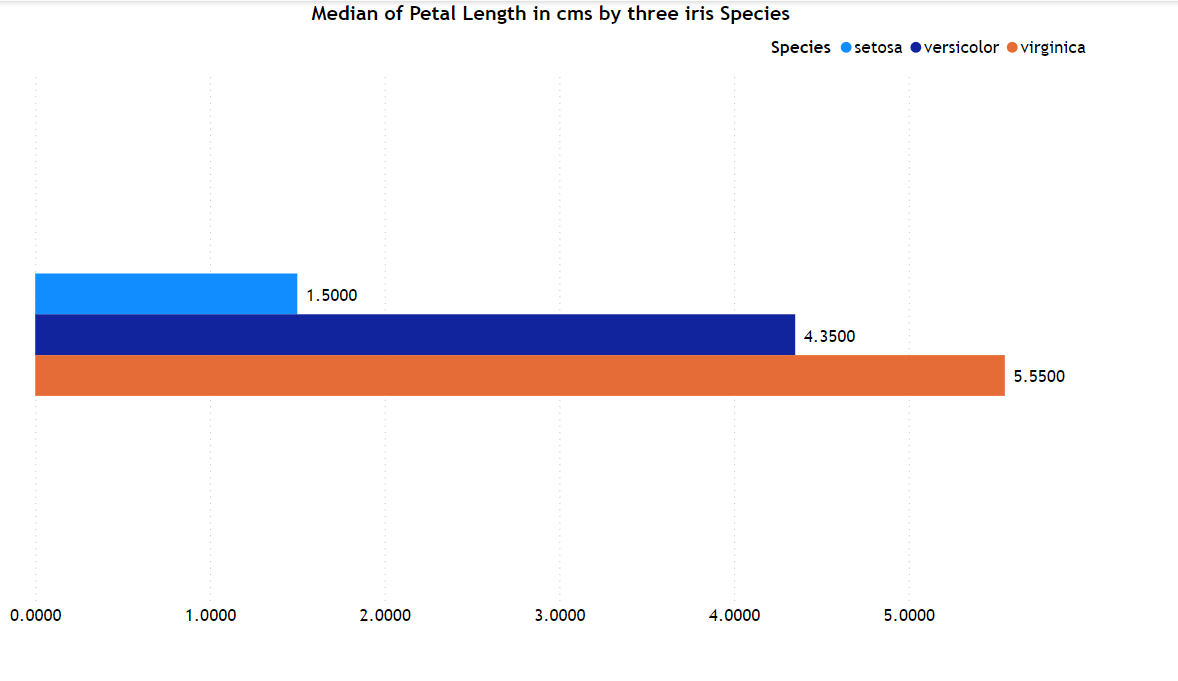
**Fig. 4.24 (c).** Formatting - Data Labels in the Stacked Bar Chart

1. Title:

|  |  |
| --- | --- |
|  | ***Make the following changes:***  ***i) Title Text: Median of Petal Length in cms by three iris Species***  ***ii) Heading: Heading 2***  ***ii) Color: Black***  ***iii) Font: Trebuchet MS***  ***iv) Font Size: 14 Bold***  ***v) Horizontal alignment: Center*** |

**Fig. 4.24 (d).** Formatting - Title in the Stacked Bar Chart

When you format as shown in Fig. 4.24 (a), (b), (c) and (d), you will see the bar chart as shown below as in Fig.4.25:

****

**Fig. 4.25.** Stacked Bar Chart for Petal.Length of iris flower species

We find that the median value or central value for iris-setosa, is differently placed than other species.

**Scatter Plots**

A scatter plot (aka scatter chart, scatter graph) uses dots to represent values for two different numeric variables. The position of each dot on the horizontal and vertical axis indicates values for an individual data point. Scatter plots are used to observe relationships between variables.

In the graph below, scatter plots of different pairs of length / width of length, width of petal / sepal of three species of iris flower (Setosa, Virginica, Versicolor):

1. Petal.Length and Petal.Width
2. Sepal.Length and Sepal.Width
3. Petal.Length and Sepal.Length
4. Petal.Width and Sepal.Width
5. Petal.Length and Sepal.Width
6. Petal.Width and Sepal.Length

We shall draw the Scatter Chart using the values of Petal.Length and Petal.Width.

Create a page for Scatter Charts by clicking on the + icon and rename the page name to “Scatter Plots” by double clicking on Page 1 and overwriting Page 1 as shown in Fig.4.26 (a) and (b).

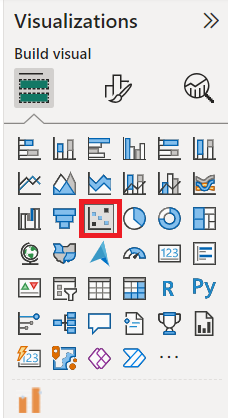


**Fig. 4.26(a).** Page Pane before renaming Page 1



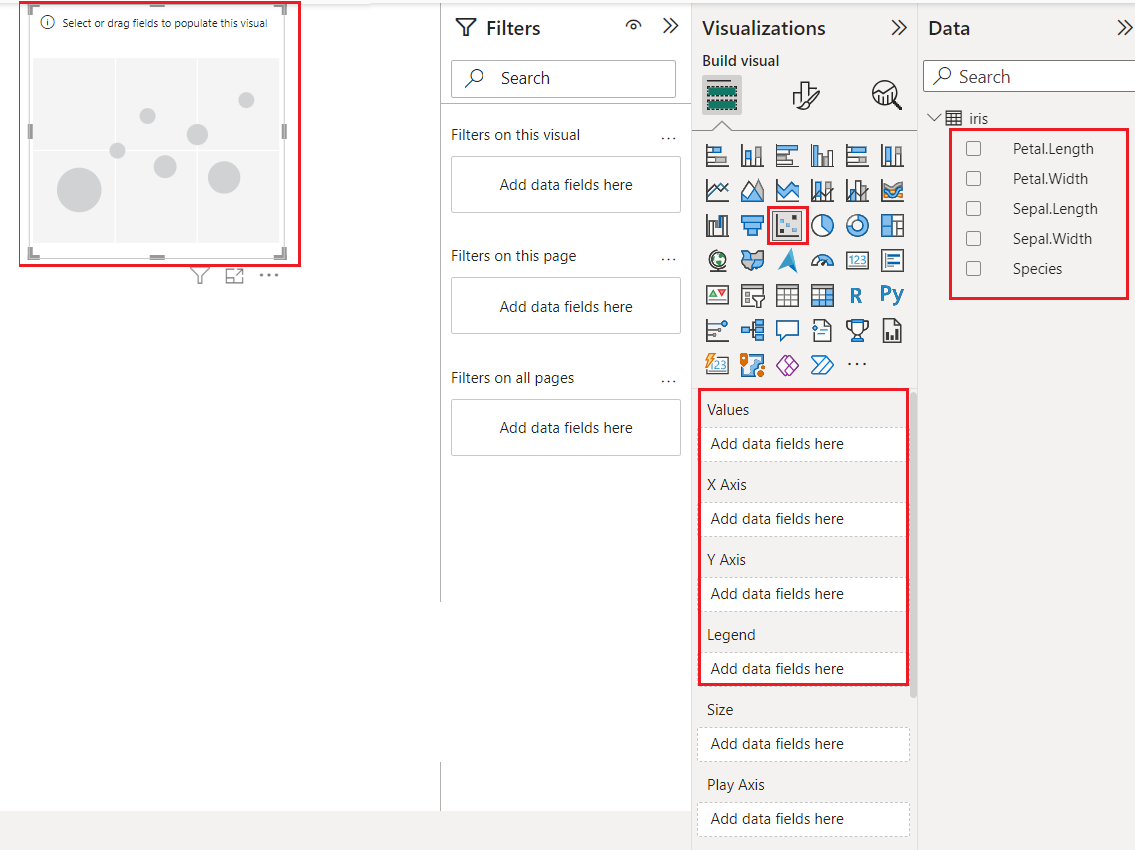
**Fig. 4.26(b).** Page Pane after renaming Page 1 as Scatter Plots

1. With the report view selected, on page, “Scatter Plots”, from visualizations, select the chart, Scatter chart as shown in Fig. 4.27.



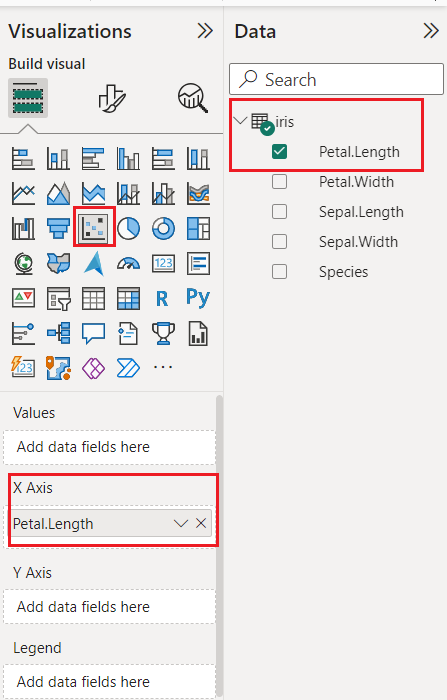
**Fig. 4.27.** Select Scatter Plot from Visualization Pane

1. You will see the place holder for creating a Scatter Chart as shown in Fig. 4.28.



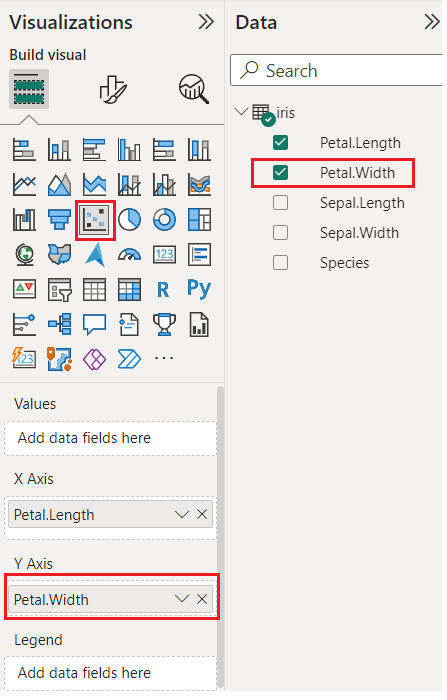
**Fig. 4.28.** Place holder for Scatter Plot

1. With the Scatter chart place holder selected on the canvas, select the following fields and make changes as shown below:
2. Select the field, Petal.Length from the iris table and push the field from Values to X-axis as shown in Fig. 4.29 (a).



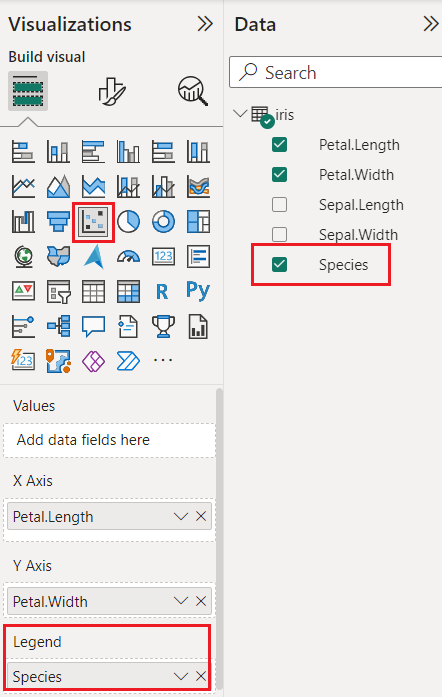
**Fig. 4.29(a).** Selecting values for X-axis for creating a Scatter Plot

1. Select the field, Petal.Width from the iris table and push the field from Values to Y-axis as shown in Fig. 4.29 (b).



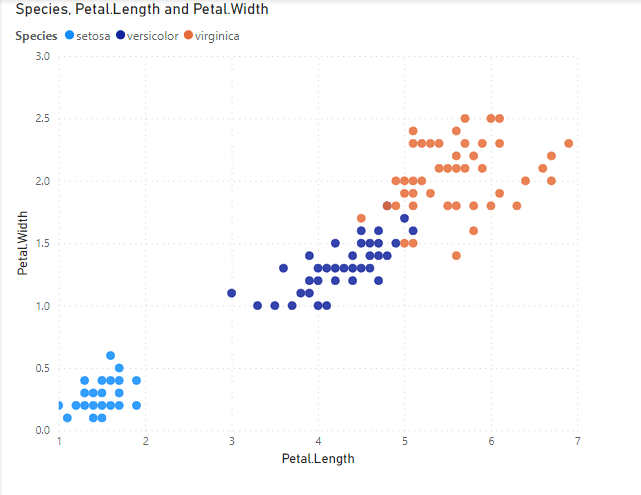
**Fig. 4.29(b).** Selecting values for Y-axis for creating a Scatter Plot

1. Select the field, Species from the iris table and push the field from Values to Legend as shown in Fig. 4.29 (c).



**Fig. 4.29(c).** Selecting values for Legend for creating a Scatter Plot

1. You will see the scatter plot with default format as shown in Fig.4.30.



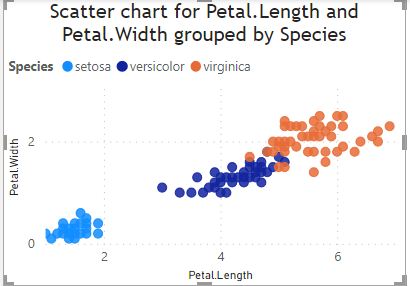
**Fig. 4.30.** Scatter Plot with default format

**Formatting: With the bar chart selected on the canvas, select format your page icon to set the formatting as shown in Fig. 4.31.**

|  |  |
| --- | --- |
|  | **Title Text: Scatter chart for Petal.Length and Petal.Width grouped by Species**  **Heading: Heading 2**  **Font: Trebuchet MS**  **Font size: 14**  **Text color: Black**  **Horizontal alignment: Center** |

**Fig. 4.31.** Scatter Plot showing how to format

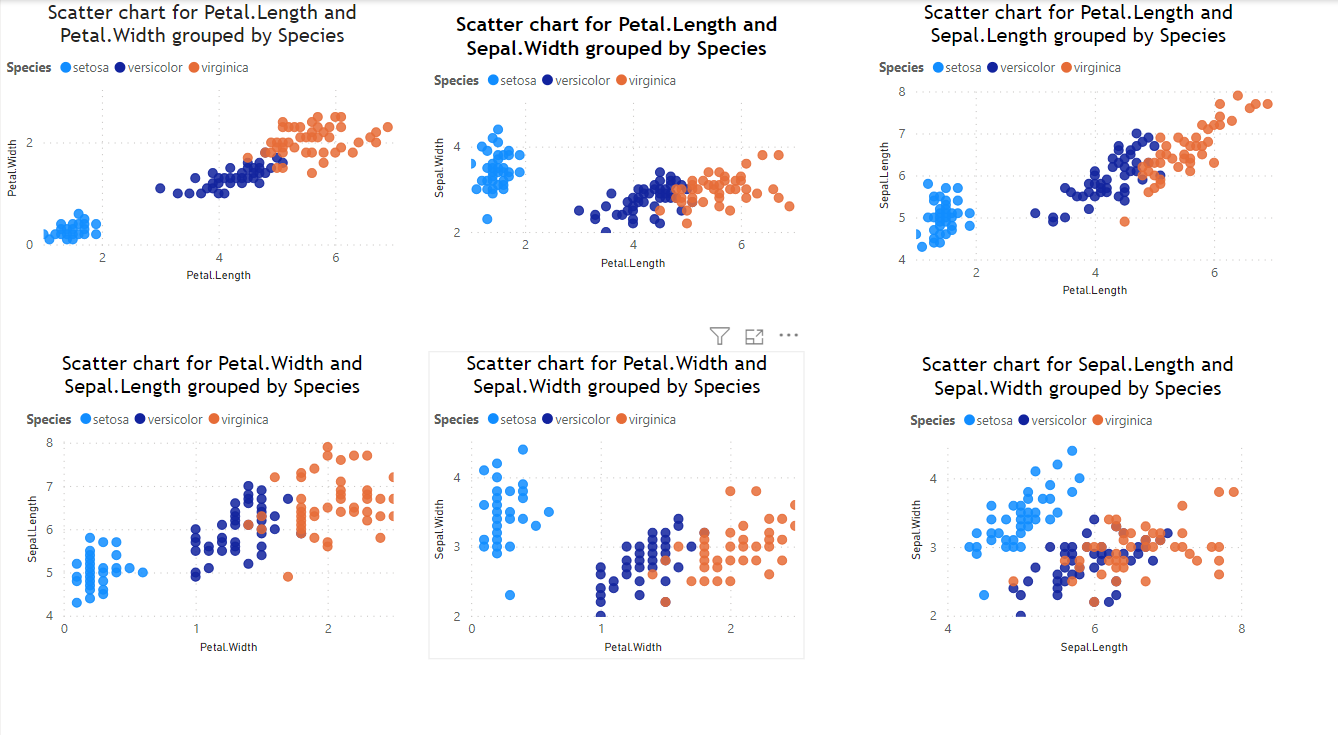
You will see the scatter chart of **Petal.Length and Petal.Width** as shown below in Fig. 4.32.:



**Fig. 4.32.** Scatter Plot for Petal Length and Petal Width

In the same way, you can proceed to draw scatter plots for the following in the same page:

1. Sepal.Length and Sepal.Width
2. Petal.Length and Sepal.Length
3. Petal.Width and Sepal.Width
4. Petal.Length and Sepal.Width
5. Petal.Width and Sepal.Length



**Fig. 4.33.** Scatter Plot for pair of Length & Width of Petal and Sepal

From these scatter plot graphs shown in Fig. 4.33, we observe that

a) iris-setosa is well separated from the other two flowers, iris-versicolor and iris-virginica.

b) Petal Length and Petal Width show the maximum separation in classes (Species of the iris flower).

**Boxplots**

A Box and Whisker plot summarizes data using five-number data summary and includes minimum & maximum value, Quartiles 1, 2 and 3 and also show outliers. We shall use boxplots to see the data distribution. We shall use Python scripts to draw boxplots. By integrating with Power BI, use can step up visualization techniques beyond what Power BI can currently do.

Before integrating Python script with Power BI[[33]](#footnote-32), you need to

1. install Python on your computer, install pip, a package installer for python, and libraries such as pandas and matplotlib.
2. enable Python scripting on your Power BI desktop

For more details, please refer to Appendix.

We shall draw the Boxplot using the values of Petal.Length, Petal.Width, Sepal. Length, Sepal.Width grouped by Species using Python script.

Create a page for Boxplots by clicking on the + icon and rename the page name to “Boxplots” by double clicking on Page 1 and overwriting Page 1 in Fig. 4.34 (a) and (b).

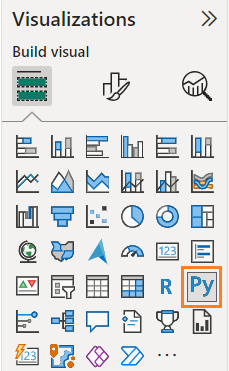


**Fig. 4.34 (a).** Page Pane before renaming Page 1



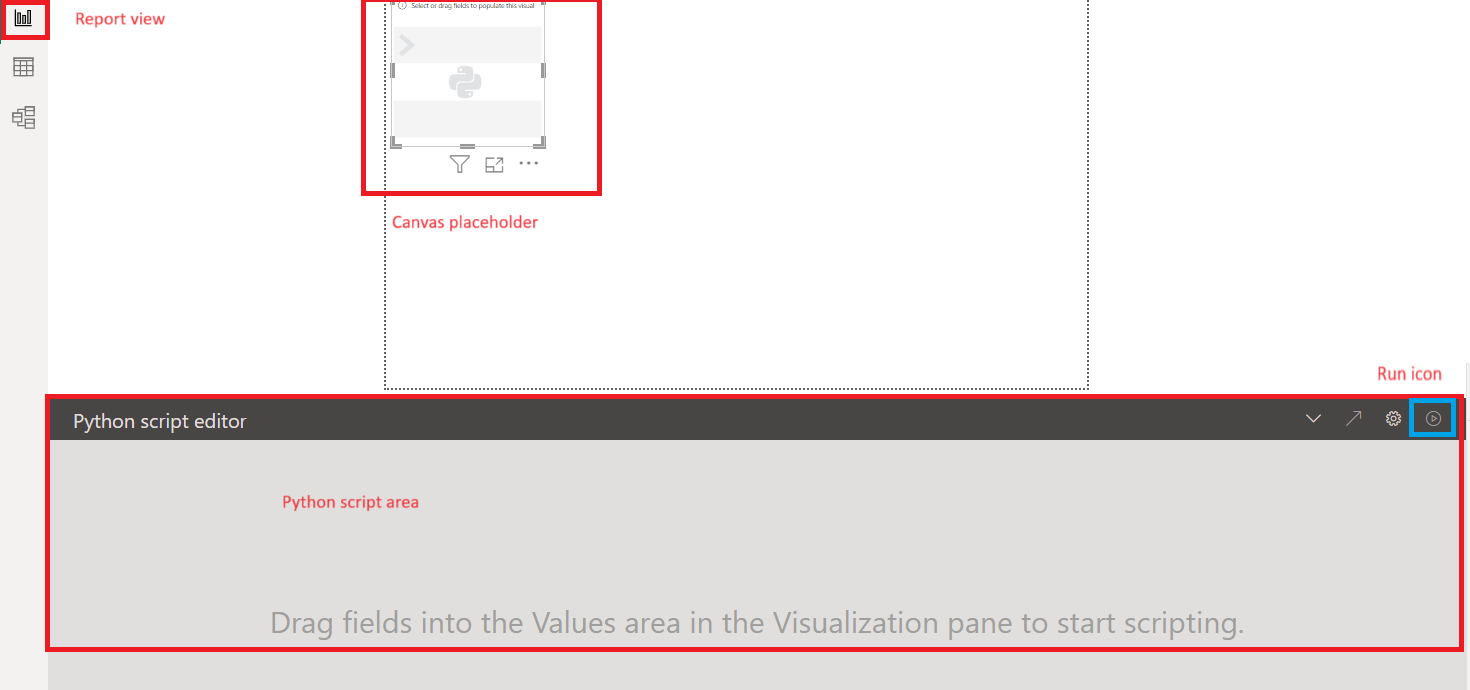
**Fig. 4.34 (b).** Page Pane after renaming Page 1 with Boxplots

1. With the report view selected, on page, “Boxplots”, from visualizations, select the Python Visual as shown here in Fig. 4.35.

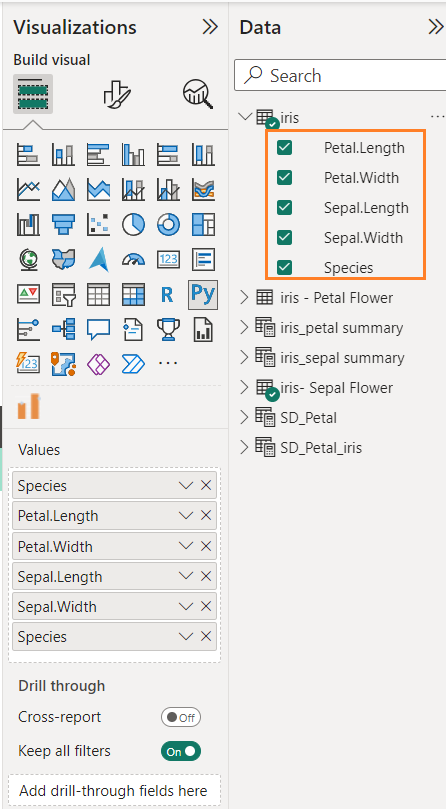


**Fig. 4.35.** Visualization pane to select Python visual icon

1. With the Python visualization placeholder selected on the canvas, select the required fields in the Data Panel and make changes as suggested in Fig. 4.36 (a) and (b).



**Fig. 4.36 (a).** Place holder for Boxplot



**Fig. 4.36(b).** Selecting fields for Boxplot

1. In the Python Script Editor panel, type the code as shown below to get the graph:

# The following code to create a dataframe and remove duplicated rows is always executed and acts as a preamble for your script:

import pandas

# dataset = pandas.DataFrame(petal.length, petal.width, sepal.length, sepal.width, Species)

# dataset = dataset.drop\_duplicates()

# Paste or type your script code here:

cols = ['Petal.Length', 'Petal.Width', 'Sepal.Length', 'Sepal.Width']

import matplotlib.pyplot as plt

dataset.boxplot(column= cols, by='Species', layout=(2, 2))

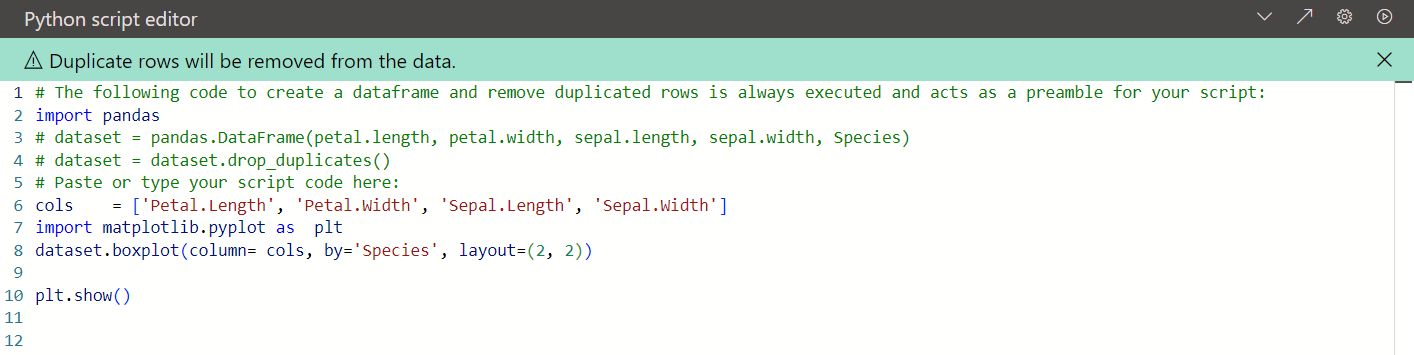
plt.show()

Please **do not remove** **#** symbol, a leading symbol in the following sentences:

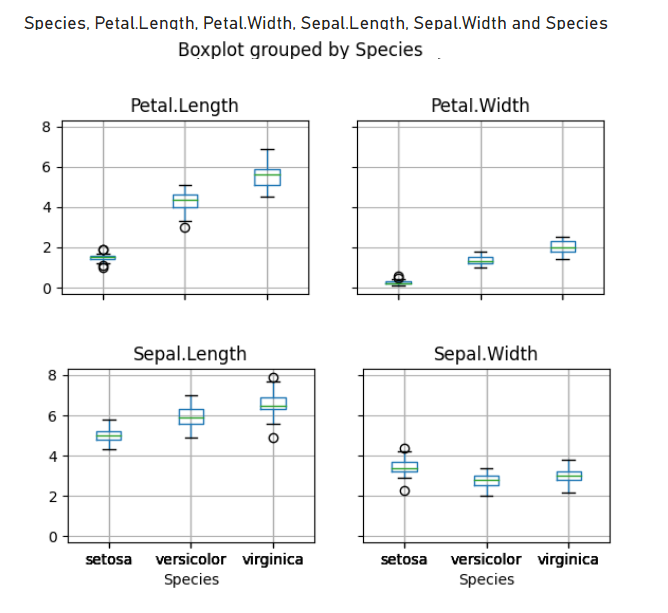
# dataset = pandas.DataFrame(petal.length, petal.width, sepal.length, sepal.width, Species)

# dataset = dataset.drop\_duplicates()

Your Python script editor will look like Fig.4.37. and you need to click on the Run icon to generate the Boxplot.



**Fig. 4.37.** Python Script Editor



**Fig. 4.38.** Boxplots for Length & Width for Petal & Sepal

From these boxplots in Fig. 4.38, we observe that

a) iris-setosa is well separated from the other two flowers, iris-versicolor and iris-virginica.

b) Petal Length and Petal Width show the maximum separation in classes (Species of the iris flower).

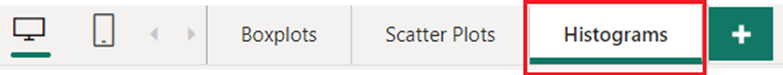
**Histograms**

A histogram[[34]](#footnote-33) is an approximate representation of the distribution of numerical data.

We shall draw the histogram for the variables Petal.Length grouped by Species using Python script. Create a page for Histograms by clicking on the + icon and rename the page name to “Histograms” by double clicking on Page 1 and overwriting Page 1 in Fig. 4.39 (a) and (b).

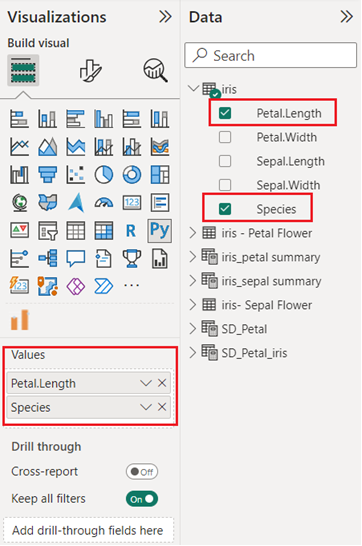


**Fig. 4.39 (a).** Page Pane before renaming Page 1

****

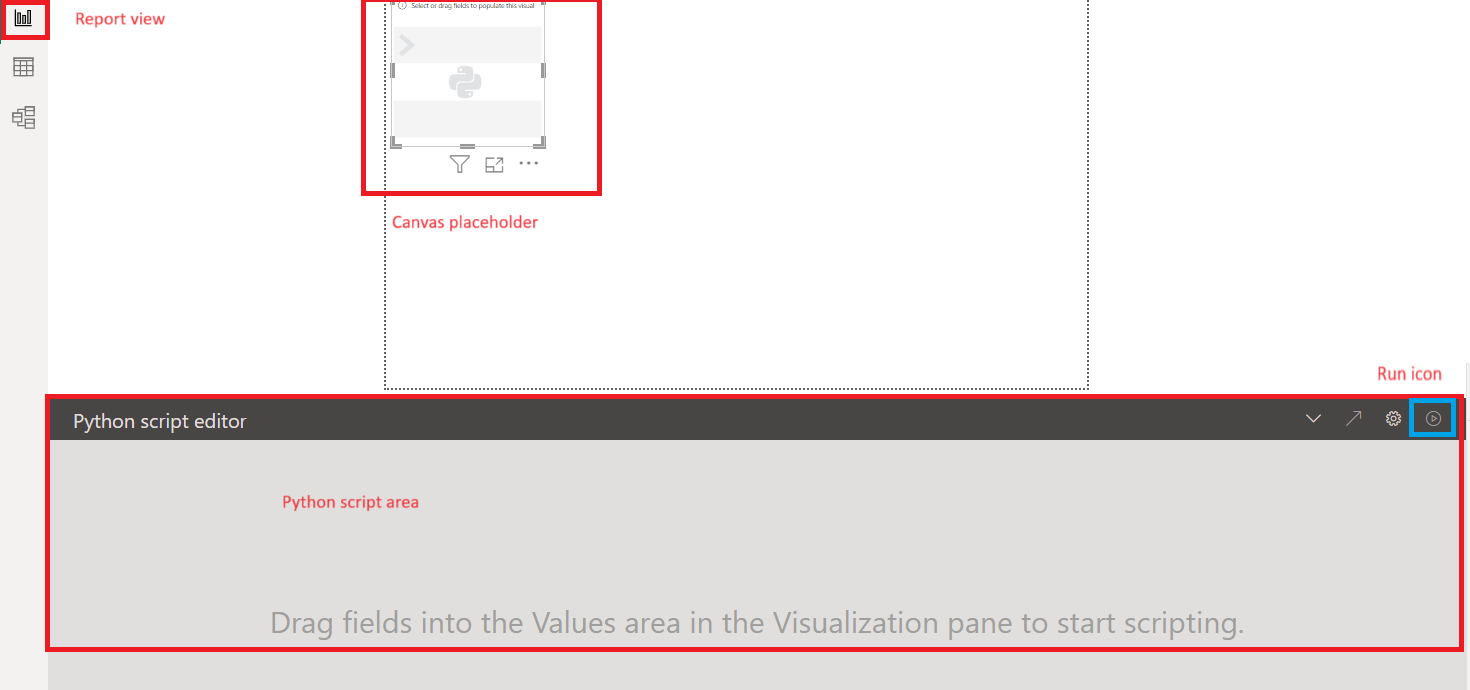
**Fig. 4.39 (b). Page Pane after renaming Page 1 with Histograms**

1. With the report view selected, on page, “Histograms”, from visualizations, select the Python Visual as shown in Fig. 4.35.
2. You will see a place holder for Histograms in Fig. 4.40 (b). With the Python visualization placeholder selected on the canvas, select the required fields, Petal.Length and Species in the Data Panel and make changes as suggested as shown in Fig.4.40(a).



**Fig. 4.40(a).** Selecting fields for Histogram

In the Python Script Editor panel, type the code as shown below and click on Run icon to get the graph shown below in Fig.4.40(b):



**Fig. 4.40 (b).** Place holder for Histograms

# The following code to create a dataframe and remove duplicated rows is always executed and acts as a preamble for your script:

# dataset = pandas.DataFrame(Petal.Length, Petal.Width, Sepal.Length, Sepal.Width, Species)

# dataset = dataset.drop\_duplicates()

import matplotlib.pyplot as  plt

setosa\_x     =  dataset.loc[dataset['Species'] == 'setosa', 'Petal.Length']

versicolor\_x =  dataset.loc[dataset['Species'] == 'versicolor', 'Petal.Length']

virginica\_x  =  dataset.loc[dataset['Species'] == 'virginica', 'Petal.Length']

kwargs = dict(bins = 5, density=True, stacked=True)

plt.hist(setosa\_x, \*\*kwargs, color='b', label = 'setosa')

plt.hist(versicolor\_x, \*\*kwargs, color='r', label = 'versicolor')

plt.hist(virginica\_x, \*\*kwargs, color='g', label='virginica')

plt.xlabel('Petal Length in cms')

plt.legend()

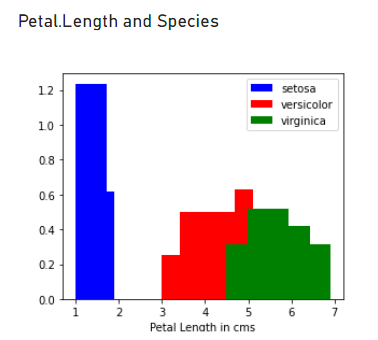
plt.show()

Please **do not remove** **#** symbol, a leading symbol in the following sentences:

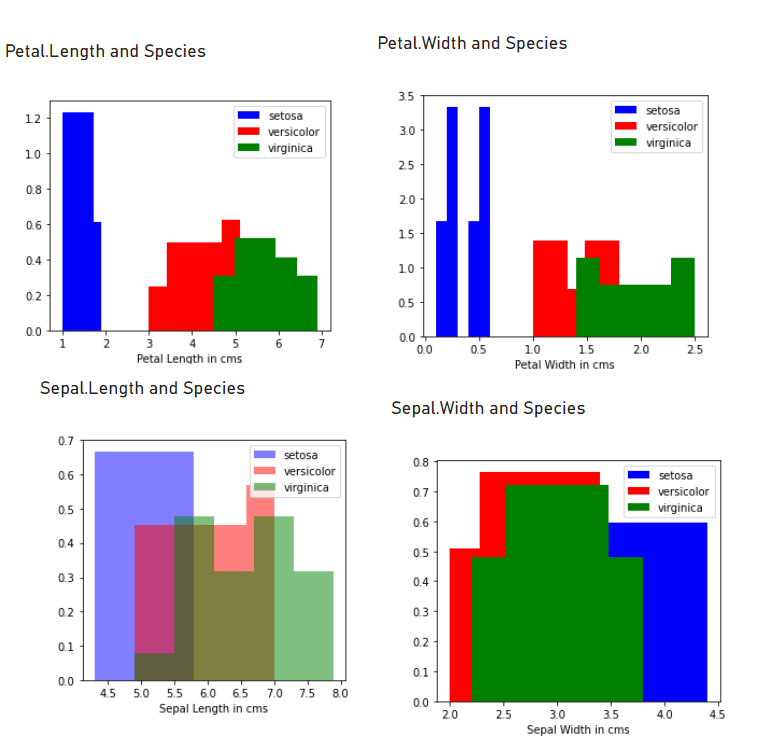
# dataset = pandas.DataFrame(Petal.Length, Petal.Width, Sepal.Length, Sepal.Width, Species)

# dataset = dataset.drop\_duplicates()

1. Histogram for Petal.Length as shown in Fig.4.41 will appear. Similarly, we can draw histograms for each of the fields, Petal.Width, Sepal. Length, Sepal.Width grouped by Species as shown in Fig.4.42.

****

**Fig. 4.41.** Histogram for Petal Length by Species

****

**Fig. 4.42.** Histogram for Length & Width for Petal & Sepal

From the histograms, we observe that the

i) Maximum Petal Length of the flower, iris-setosa is less than 2 cms (1.9 cms) and the minimum of Petal Length of other flowers, iris-versicolor and iris-virginica are 3 cms and 4.5 cms respectively.

ii) Maximum Petal Width of the flower, iris-setosa is less than 1 cm (0.6 cms) and the minimum of Petal Width of other flowers, iris-versicolor and iris-virginica are 1 cms and 1.4 cms respectively.

iii) iris-setosa is well separated from the other two flowers, iris-versicolor and iris-virginica.

iv) Petal Length and Petal Width show the maximum separation in classes (Species of the iris flower).

# **Appendix**

**Include Python scripts in Power BI.**

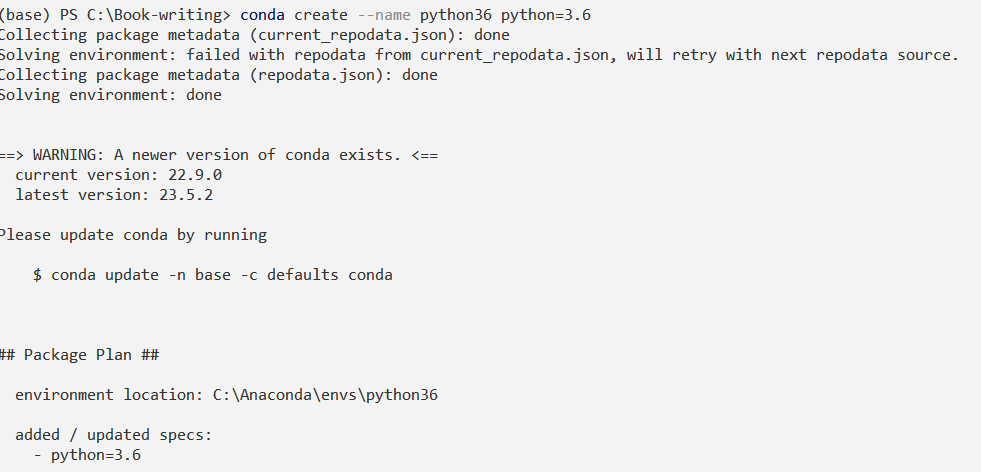
Ref: https://www.freecodecamp.org/news/python-in-powerbi/

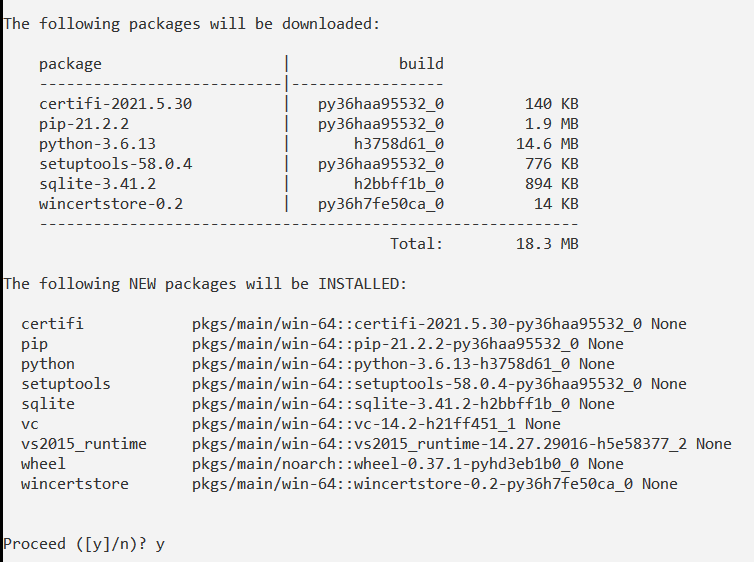
Power BI have issues when working with versions higher than 3.6. So, it is advisable to create a virtual environment for Python 3.6 and it does not affect other projects.

In the Anaconda Prompt, create a new environment by typing the code

**conda create --name python36 python=3.6**

You will see the following screen as shown in Fig.5.1.

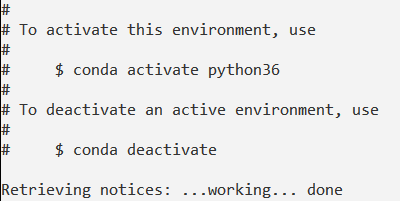




**Fig.5.1.** Creating a new Python environment for python 3.6

Then, when you see the Proceed ([y]/n), type y.

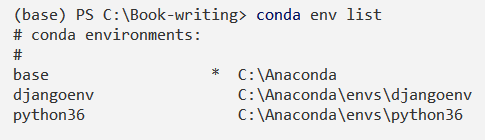
Wait until it completes running. You will see the following screen as shown in Fig.5.2.



**Fig.5.2.** Activate the Python 3.6 environment

To see the list of Anaconda environments as shown in Fig.5.3, type

**conda env list**

****

**Fig.5.3.** Conda environment list

Note the path to python36 from the line

**python36 C:\Anaconda\envs\python36**

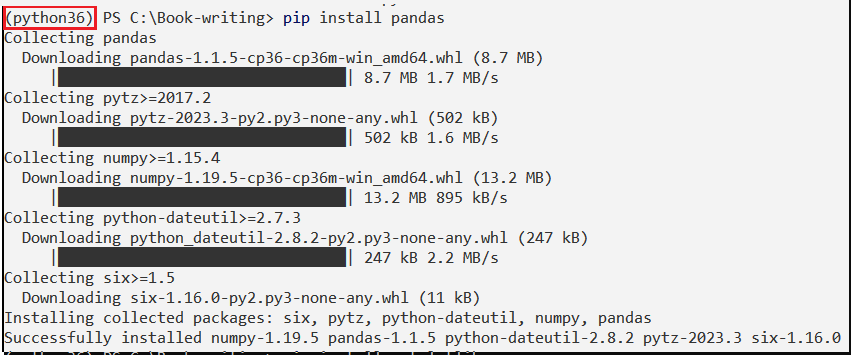
Activate the new environment, before installing pandas and matplotlib libraries. Go to the folder, where you would like to install pandas and other packages.

**conda activate python36**

You will observe the change of environment name in the parenthesis as (python36).

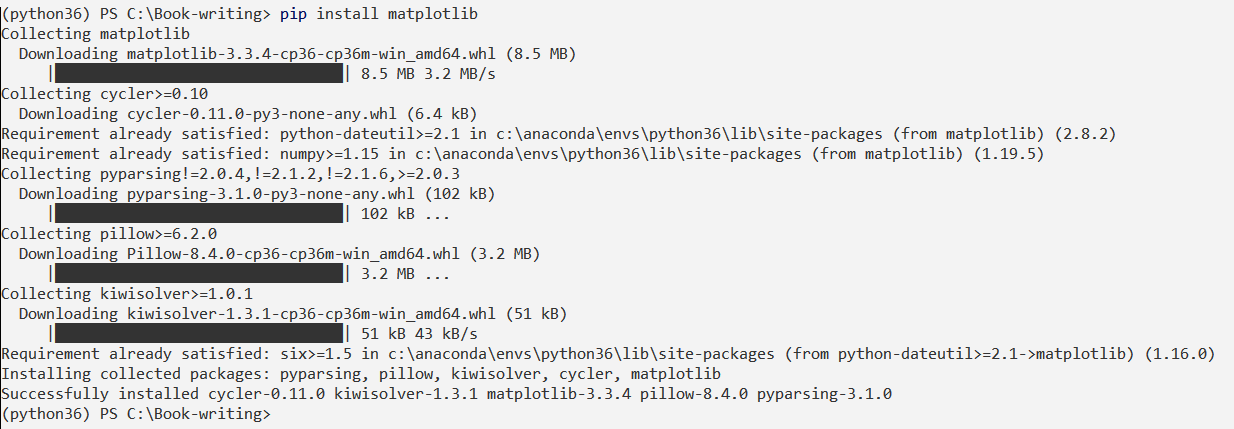
You can install the python libraries using the pip install command as shown in Fig.5.4 and Fig.5.5.

**pip install pandas**

****

**Fig.5.4.** Installing pandas package

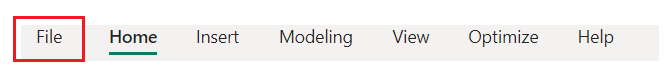
**pip install matplotlib**



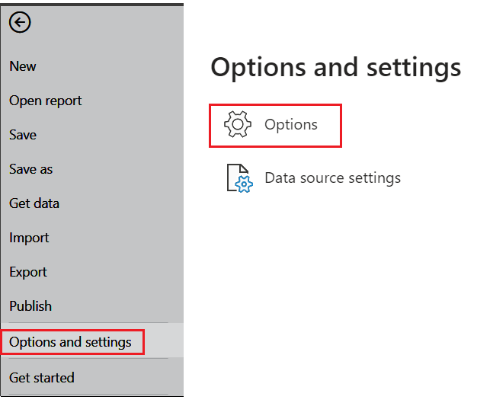
**Fig.5.5.** Installing matplotlib package

Set Python in Power BI

1) Click on the File menu in the upper left-hand corner as shown in Fig. 5.6.

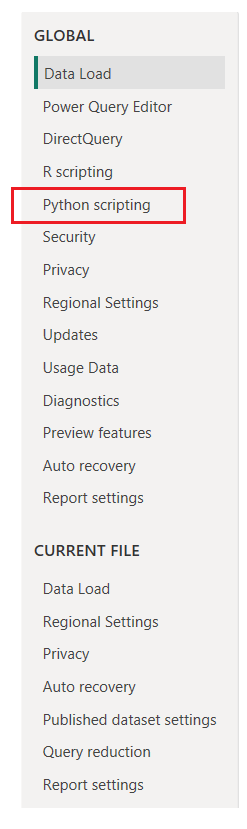
 **Fig.5.6.** Setting Python in Power BI - Select File Menu

2) Click on “Options and settings” and then Options as shown in Fig.5.7.



**Fig.5.7.** Setting Python in Power BI - Options and Settings

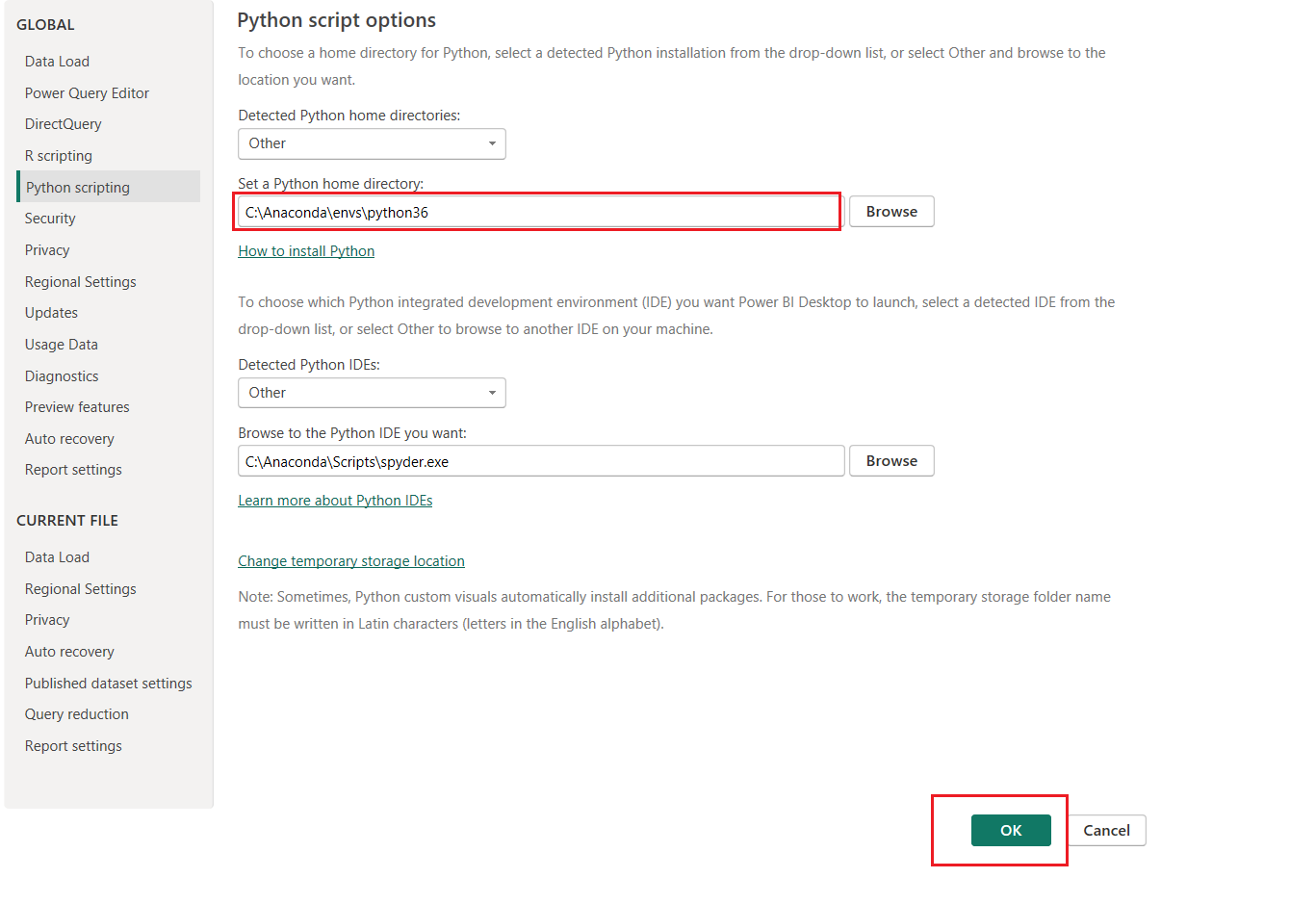
3) Click on Python Scripting as shown in Fig.5.8.



**Fig.5.8.** Setting Python in Power BI - Select Python scripting

4) Change directories and Navigate to your Python Environment.

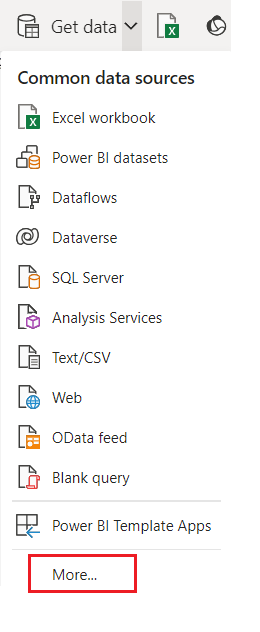
Change the detected Python home directory to "Other" and browse for your Python environment created above as shown in Fig.5.9.



**Fig.5.9.** Setting Python Home directory

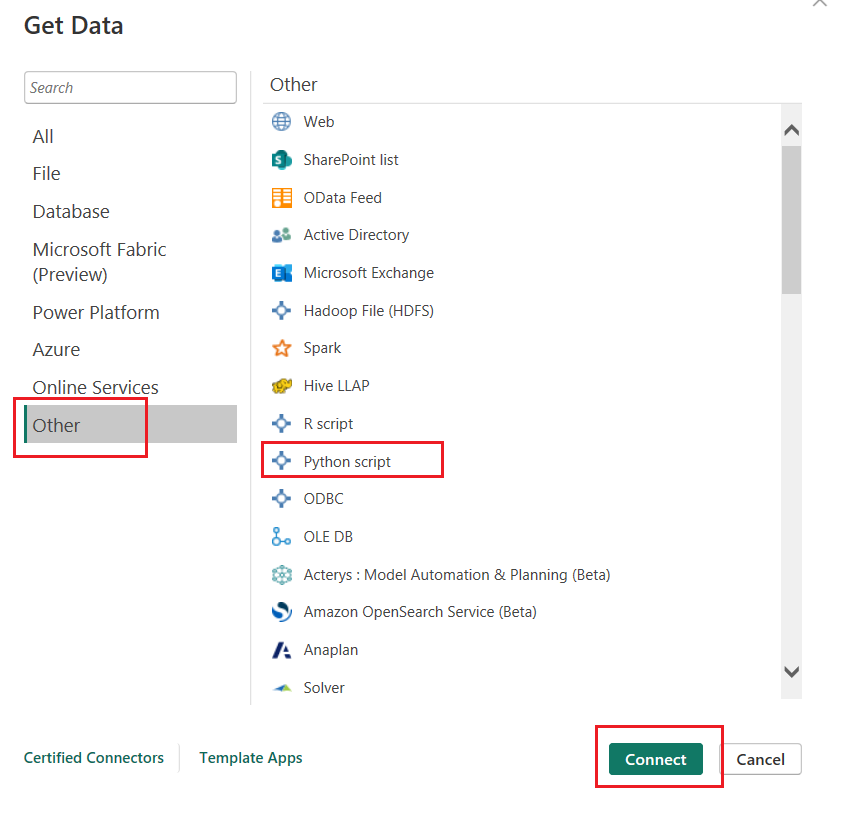
In this example, I will create a Pandas DataFrame as shown in Fig.5.10(a) , (b), (c) and (d).

Step 1: Click on Get Data and then More



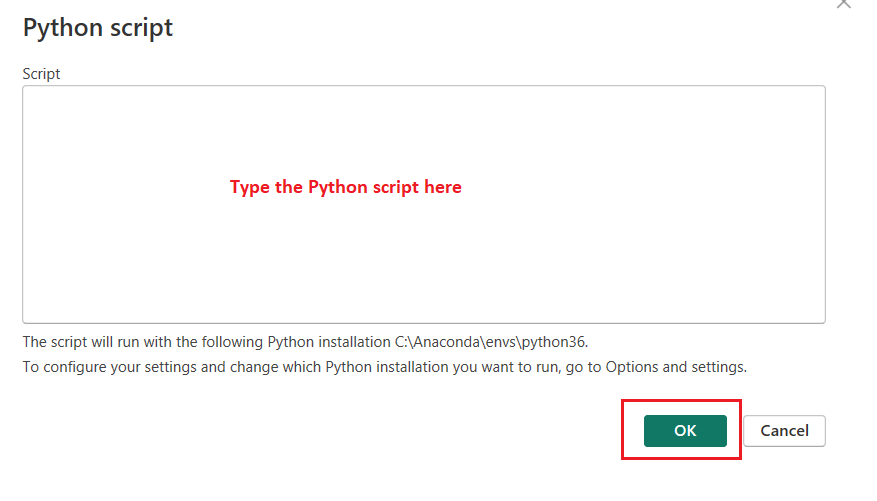
**Fig.5.10(a).** Get Data and More to create Pandas Dataframe

Step 2: Click on “Other”, “Python Script” and “Connect” as shown below:



**Fig.5.10(b).** To select Python script to create Pandas Dataframe

Step 3: The following screen appears for you to enter your Python script.



**Fig.5.10(c).** To get a text area to key-in the Python script to create Pandas Dataframe

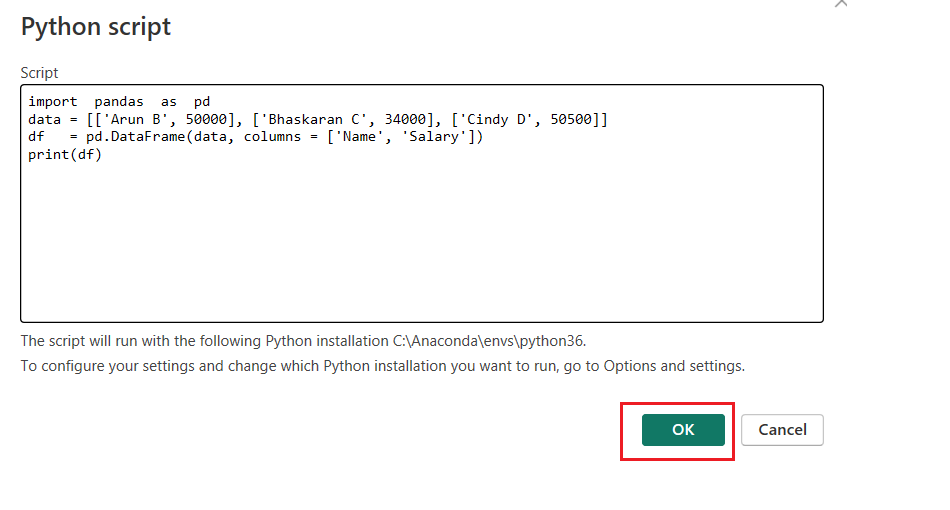
Type the Python script in the box provided.

import pandas as pd

data = [['Arun B', 50000], ['Bhaskaran C', 34000], ['Cindy D', 50500]]

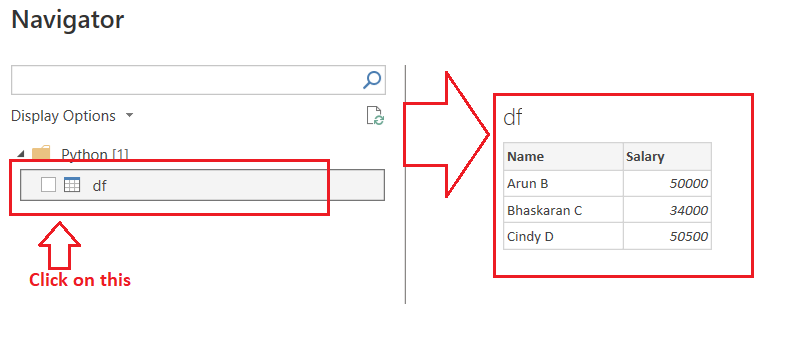
df = pd.DataFrame(data, columns = ['Name', 'Salary'])

print(df)



**Fig.5.10(d).** The Python script to create Pandas Dataframe

It will take some time to establish the connection to Python and execute the script for showing the screen shown below. Click on the df, DataFrame to see the results as shown in Fig.5.11.



**Fig.5.11.** Dataframe created by Python script

1. https://en.wikipedia.org/wiki/Microsoft\_Power\_BI [↑](#footnote-ref-1)
2. https://learn.microsoft.com/en-gb/archive/blogs/bi/data-visualization-done-right-project-crescent [↑](#footnote-ref-2)
3. https://learn.microsoft.com/en-us/power-bi/create-reports/power-bi-report-filter-types [↑](#footnote-ref-3)
4. https://learn.microsoft.com/en-us/dax/dax-overview [↑](#footnote-ref-4)
5. https://learn.microsoft.com/en-us/power-bi/natural-language/q-and-a-data-sources [↑](#footnote-ref-5)
6. https://learn.microsoft.com/en-us/power-bi/visuals/power-bi-visualization-kpi?tabs=powerbi-desktop [↑](#footnote-ref-6)
7. https://learn.microsoft.com/en-us/power-bi/connect-data/service-real-time-streaming [↑](#footnote-ref-7)
8. https://www.clariontech.com/blog/6-reasons-to-use-power-bi-as-your-business-intelligence-solution [↑](#footnote-ref-8)
9. https://community.fabric.microsoft.com/t5/Desktop/Best-practices-to-create-PowerBI-reports-with-collaboration-in/m-p/192827#:~:text=Power%20BI%20app%20workspaces%20are,view%2Fedit)%20on%20it. [↑](#footnote-ref-9)
10. https://learn.microsoft.com/en-us/power-bi/fundamentals/service-self-service-signup-for-power-bi [↑](#footnote-ref-10)
11. https://www.biconnector.com/blog/tableau-vs-power-bi/#:~:text=Power%20BI%20tends%20to%20drag,of%20data%20points%20in%20visualization. [↑](#footnote-ref-11)
12. https://www.knowledgehut.com/blog/business-intelligence-and-visualization/power-bi-developer-future#demand-and-future-scope-of-power-bi [↑](#footnote-ref-12)
13. https://learn.microsoft.com/en-us/power-bi/fundamentals/service-self-service-signup-for-power-bi [↑](#footnote-ref-13)
14. https://learn.microsoft.com/en-us/power-bi/fundamentals/service-service-vs-desktop [↑](#footnote-ref-14)
15. https://learn.microsoft.com/en-us/power-bi/create-reports/service-the-report-editor-take-a-tour [↑](#footnote-ref-15)
16. https://learn.microsoft.com/en-us/powerquery-m/ [↑](#footnote-ref-16)
17. https://learn.microsoft.com/en-us/power-bi/ [↑](#footnote-ref-17)
18. https://learn.microsoft.com/en-us/dax/ [↑](#footnote-ref-18)
19. https://learn.microsoft.com/en-us/dax/new-dax-functions [↑](#footnote-ref-19)
20. https://learn.microsoft.com/en-us/dax/networkdays-dax [↑](#footnote-ref-20)
21. https://learn.microsoft.com/en-us/dax/aggregation-functions-dax [↑](#footnote-ref-21)
22. https://learn.microsoft.com/en-us/dax/date-and-time-functions-dax [↑](#footnote-ref-22)
23. https://learn.microsoft.com/en-us/dax/filter-functions-dax [↑](#footnote-ref-23)
24. https://learn.microsoft.com/en-us/dax/financial-functions-dax [↑](#footnote-ref-24)
25. https://learn.microsoft.com/en-us/dax/table-manipulation-functions-dax [↑](#footnote-ref-25)
26. https://microsoftlearning.github.io/PL-300-Microsoft-Power-BI-Data-Analyst/Instructions/03-configure-data-model-in-power-bi-desktop.html [↑](#footnote-ref-26)
27. [https://www.datacamp.com/tutorial/machine-learning-in-r](https://www.datacamp.com/tutorial/machine-learning-in-r%20) [↑](#footnote-ref-27)
28. <http://archive.ics.uci.edu/ml/datasets/Iris> [↑](#footnote-ref-28)
29. <https://iridl.ldeo.columbia.edu/dochelp/StatTutorial/> [↑](#footnote-ref-29)
30. https://citoolkit.com/articles/histograms-and-boxplots/ [↑](#footnote-ref-30)
31. [↑](#endnote-ref-1)
32. <https://www.statisticshowto.com/probability-and-statistics/descriptive-statistics/bar-chart-bar-graph-examples/> [↑](#footnote-ref-31)
33. https://blog.enterprisedna.co/how-to-use-python-script-in-power-bi/ [↑](#footnote-ref-32)
34. https://en.wikipedia.org/wiki/Histogram [↑](#footnote-ref-33)