**Data Visualization Portfolio – Iris Dataset**

**By: K.KAVIYA SHREE – Data Science Enthusiast**  
**Date: 19-07-2025**

**Project Objective**

The goal of this project is to explore and understand the relationships between different attributes of the famous Iris flower dataset. The dataset contains measurements of flower features such as petal length and sepal width and is often used for classification into species.

Through visual analysis, we aim to identify patterns, trends, and relationships using Python’s data visualization libraries.

**Dataset Overview**

* **Filename:** iris.csv
* **Records:** 150 flower samples
* **Features:**
  + Sepal Length (cm)
  + Sepal Width (cm)
  + Petal Length (cm)
  + Petal Width (cm)
  + Species (Setosa, Versicolor, Virginica)

**Tools & Technologies**

* **Python** – Core scripting and logic
* **Pandas** – Data manipulation and analysis
* **Seaborn & Matplotlib** – Data visualization and plotting

**Visualizations & Insights**

**Histogram Plots**

* **Sepal Length Distribution**  
  Normal distribution with slight skew.
* **Petal Width Distribution**  
  Reveals clear separability between species.

**Correlation Heatmap**

* **Petal Length and Width** are strongly positively correlated.
* **Sepal Width** has a weak or negative correlation with Petal Width.

**Scatter Plot**

* **Petal Length vs. Petal Width**  
  Strong clustering by species.  
  Setosa is distinctly separated from the other two species.

**Boxplot**

* **Sepal Length by Species**  
  Setosa has the smallest range.  
  Virginica has the longest sepal length.

**Bar Plot**

* **Average Petal Width by Species**
  + Setosa: Smallest average petal width
  + Virginica: Highest average petal width

**Pairplot**

* Visualizes all pairwise feature relationships.
* Clear species separation in petal dimensions.
* Setosa forms a distinct cluster in all plots.

**Key Learnings & Conclusions**

* Petal dimensions are key to distinguishing Iris species.
* Setosa species shows high separability from Versicolor and Virginica.
* Visualization reveals class boundaries and feature importance effectively.

**Next Steps**

* Train classification models (e.g., KNN, SVM) using the identified key features.
* Apply clustering (e.g., K-means) to verify natural groupings.
* Share visualizations on:
  + GitHub
  + Kaggle
  + Personal portfolio or blog