Iris Data Visualization Project

# 1. Introduction

This project demonstrates data visualization techniques using Python. Various types of plots are created to gain insights from the dataset, and popular libraries like Matplotlib and Seaborn are used.

# 2. Tools and Libraries Used

- Python  
- Pandas  
- Matplotlib  
- Seaborn  
- Jupyter Notebook

# 3. Code and Visualizations

Code:

# import Libraries   
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns  
sns.set\_palette('husl')  
import warnings  
import math

Code:

#Read data from csv  
iris\_data= pd.read\_csv("C:\\Users\\jahna\\OneDrive\\Desktop\\Data Visualization\\IRIS - IRIS.csv")  
  
#show top 10 rows  
iris\_data.head(10)

Code:

#Get detailed information of data  
iris\_data.info()

Code:

#Shape of data, No of columns  
iris\_data.shape[1]

Code:

#No of rows in the data  
iris\_data.shape[0]

Code:

# No of rows and columns  
iris\_data.shape

Code:

iris\_data.species.unique()

Code:

iris\_data.tail(10)

Code:

iris\_data.count(axis=0)

Code:

iris\_data.isnull()

Code:

iris\_data.describe()

Code:

iris\_data["species"].value\_counts()

Code:

iris\_data.dropna()

Code:

def missing\_values(x):  
 return sum(x.isnull())  
  
print("Missing values in each column:")  
print(iris\_data.apply(missing\_values,axis=0))

Code:

type(iris\_data)

Code:

# Data Visualization  
iris\_data.describe().plot(kind="area", fontsize=20, figsize=(20,8), table=False, colormap="rainbow")  
plt.xlabel("Statistics")  
plt.ylabel("Value")  
plt.title("Statistics of IRIS dataset")

Code:

sns.countplot(x="species", data=iris\_data)

Code:

# Draw a Pie Chart  
iris\_data["species"].value\_counts().plot.pie(explode=[0.1,0.1,0.1],autopct='%1.1f%%', shadow=True, figsize=(5,3))  
plt.show()

Code:

# Draw a scatterplot with marginal histogram  
figure= sns.jointplot(x="sepal-length", y="sepal-width", data=iris\_data, color="purple")

Code:

# Replace the scatterplots and histograms with density estimates  
figure= sns.jointplot(x="sepal-length", y="sepal-width", data=iris\_data, color="orange", kind="kde")

Code:

# Add regression  
figure= sns.jointplot(x="sepal-length", y="sepal-width", data=iris\_data, color="green", kind="reg")

Code:

# Replace the scatterplot with a joint histogram using hexagonal bins  
figure= sns.jointplot(x="sepal-length", y="sepal-width", data=iris\_data, color="blue", kind="hex")

Code:

#Draw a scatterplot, then add a joint density estimate  
figure= (sns.jointplot(x="sepal-length",y="sepal-width", data=iris\_data,color="red").plot\_joint(sns.kdeplot,zorder=0,n\_levels=6))

Code:

sns.FacetGrid(iris\_data, hue="species", height=7).map(plt.scatter,"sepal-length","sepal-width").add\_legend()

Code:

#Box Plot  
fig=plt.gcf()  
fig.set\_size\_inches(8,5)  
fig=sns.boxplot(x="species", y="petal-width", data=iris\_data, hue="species",\  
 order=["Iris-setosa","Iris-versicolor","Iris-virginical"],\  
 linewidth=2.5, orient='v', dodge=False)

Code:

#Draw a categorical scatterplot with non-overlapping points.  
sns.swarmplot(x="species", y="petal-width", data=iris\_data, color=".5")

Code:

#Draw boxplot by species  
iris\_data.boxplot(by="species", figsize=(8, 5))

Code:

#combine stripplot and boxplot  
fig=plt.gcf()  
fig.set\_size\_inches(8, 5)  
fig= sns.boxplot(x="species", y="petal-width", data=iris\_data)  
fig= sns.stripplot(x="species", y="petal-width", data=iris\_data, jitter=True, edgecolor="black", hue="species"\  
 ,linewidth=1.0)

Code:

# Draw Violin Plot  
fig=plt.gcf()  
fig.set\_size\_inches(8,5)  
fig=sns.violinplot(x="species", y="petal-width", data=iris\_data, hue="species", saturation=0.8, palette= "Set3")

Code:

# Plot subplot for different columns in the data set  
plt.figure(figsize=(10,5))  
plt.subplot(2,2,1)  
sns.violinplot(x="species", y="sepal-length", data=iris\_data, hue="species", saturation=0.8, palette="summer")  
plt.subplot(2,2,2)  
sns.violinplot(x="species", y="sepal-length", data=iris\_data, hue="species", saturation=0.8, palette="summer")  
plt.subplot(2,2,3)  
sns.violinplot(x="species", y="sepal-length", data=iris\_data, hue="species", saturation=0.8, palette="summer")  
plt.subplot(2,2,4)  
sns.violinplot(x="species", y="sepal-length", data=iris\_data, hue="species", saturation=0.8, palette="summer")

Code:

# Pair plot  
sns.pairplot(data=iris\_data, kind="scatter", hue="species", dropna=True, palette="winter")

Code:

# Heat Map  
fig = plt.gcf()  
fig.set\_size\_inches(8, 5)  
  
sns.heatmap(iris\_data.select\_dtypes(include='number').corr(),   
 vmin=-1, vmax=1, cmap="cubehelix", linewidths=1, linecolor="blue",   
 cbar=True, cbar\_kws={'orientation': 'vertical'},   
 square=True, annot=True, mask=False)

Code:

# Distribution Plot  
fig=plt.gcf()  
fig.set\_size\_inches(8,5)  
iris\_data.hist(bins=10, grid=True, linewidth=1, edgecolor="black")  
iris\_data.hist(by="species", bins=10, grid=True, linewidth=1, edgecolor="black")

Code:

# LMplot  
fig=sns.lmplot(x="sepal-length", y="sepal-width", data=iris\_data, hue="species", markers='o', palette="winter")

Code:

# FacetGrid  
sns.FacetGrid(iris\_data, hue="species", height=5)\  
 .map(sns.kdeplot,"sepal-length")\  
 .add\_legend()  
plt.ioff()

Code:

# 4. Conclusion

The project successfully explores data using various visualization techniques. Charts like bar plots, scatter plots, histograms, and heatmaps help in understanding the distribution and relationships within the dataset.

# 5. GitHub Repository

GitHub Link: [Add your GitHub project link here]