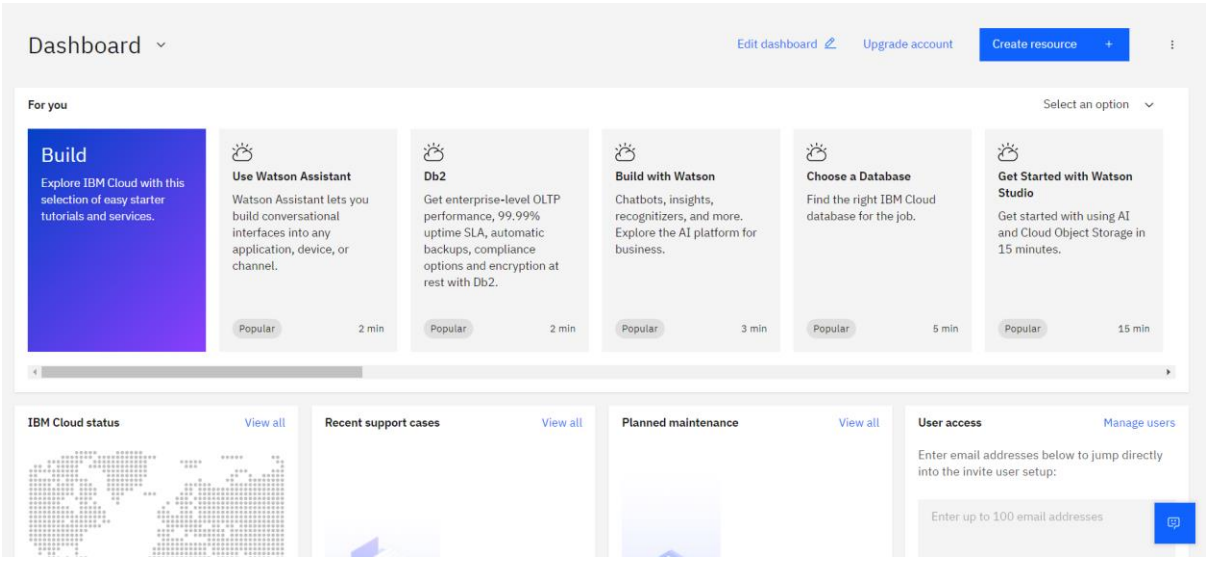
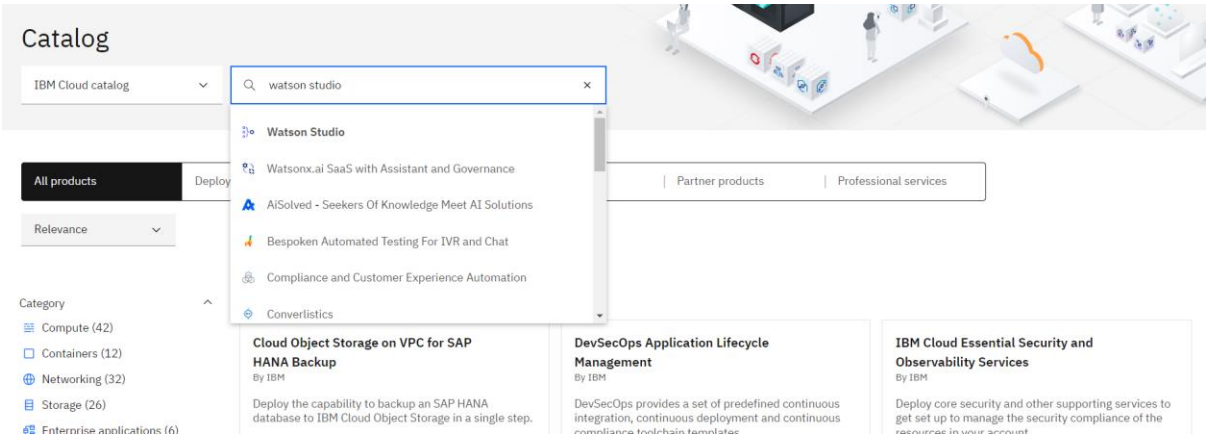
LAB 2- Advance Data Analytics and Data visualization using pandas/Watson Studio

Step 1: Access Watson Studio

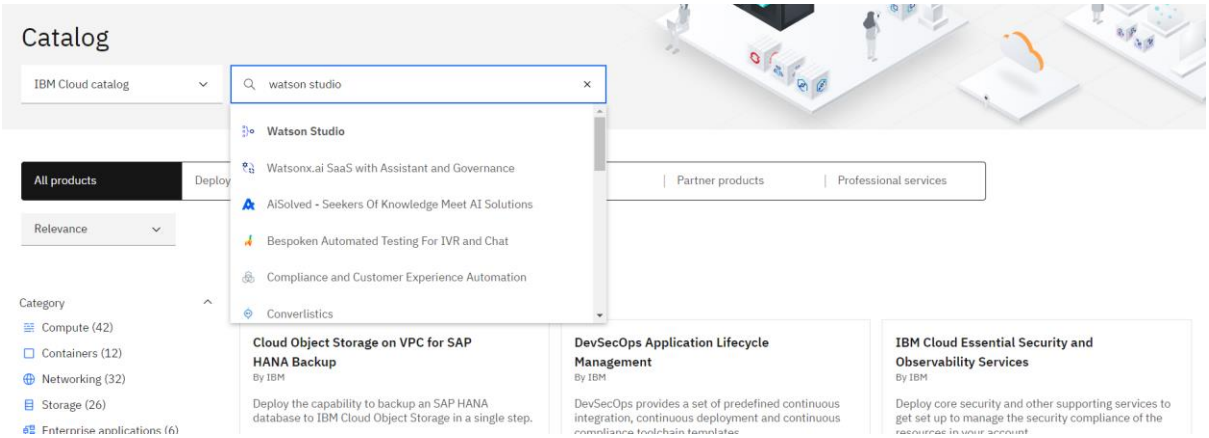
1. **Log in to IBM Cloud**: Go to IBM Cloud and sign in to your account.
2. **Search for Watson Studio**: In the IBM Cloud catalog, type "Watson Studio" in the search bar.





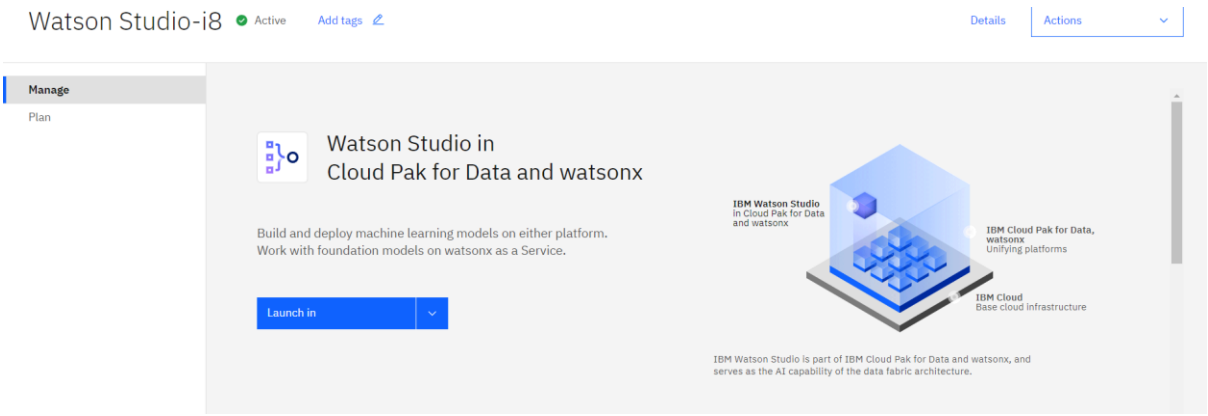
Step 2: Create a Service in Watson Studio

1. **Create Service**: Click on Watson Studio and follow the prompts to create a new Watson Studio service. Select the appropriate plan based on your needs.



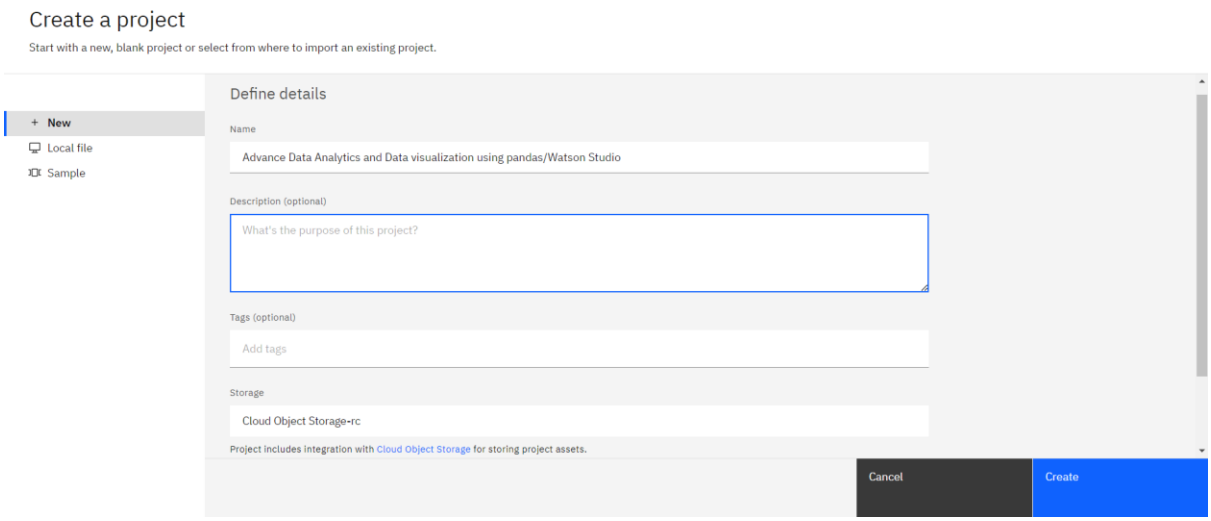
Step 3: Launch IBM Cloud Pak for Data

1. **Access Cloud Pak**: Navigate to IBM Cloud Pak for Data if it's part of your environment, or go directly to Watson Studio.
2. **Open Watson Studio**: Click on the Watson Studio service to launch it.



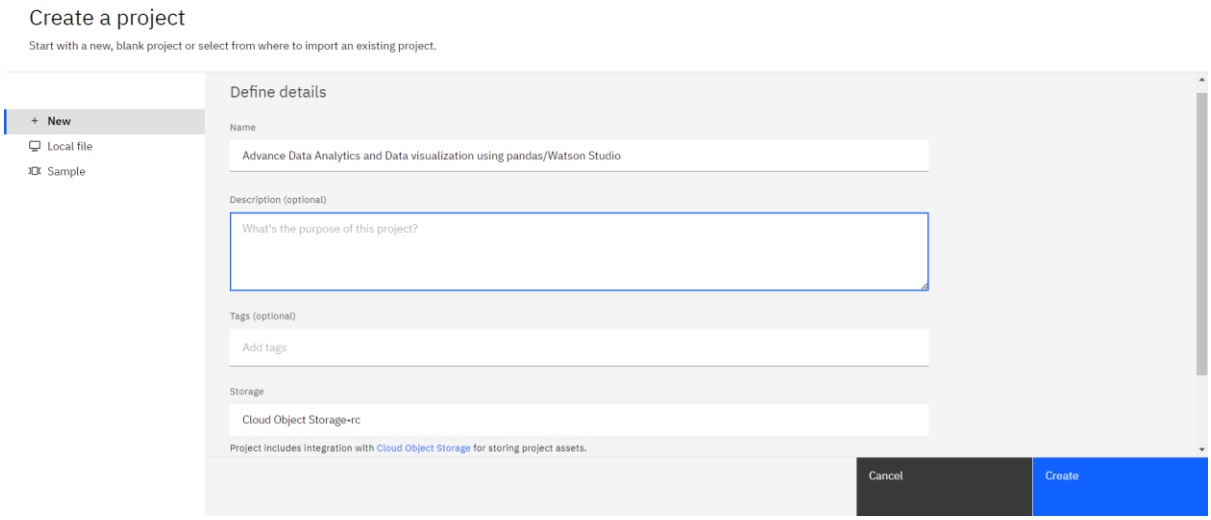
Step 4: Create a New Project

1. **Create Project**: In Watson Studio, click on "New Project."
2. **Select Project Type**: Choose "Default" for a standard project.
3. **Provide Project Details**: Enter a name and description for your project and click "Create."



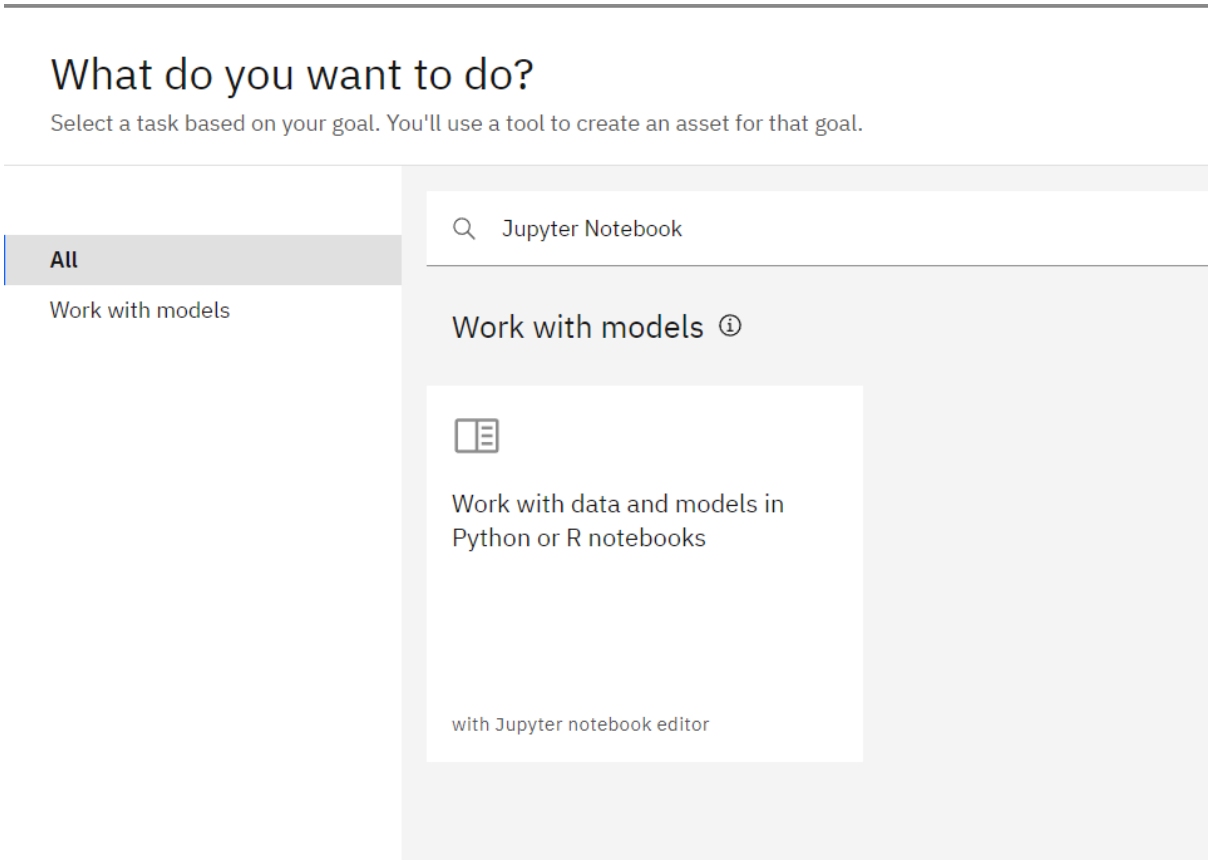
Step 5: Create a New Asset

1. **Navigate to Assets**: Within your project, click on the "Assets" tab.
2. **Create Asset**: Click on "Add Asset" or "Create New Asset."



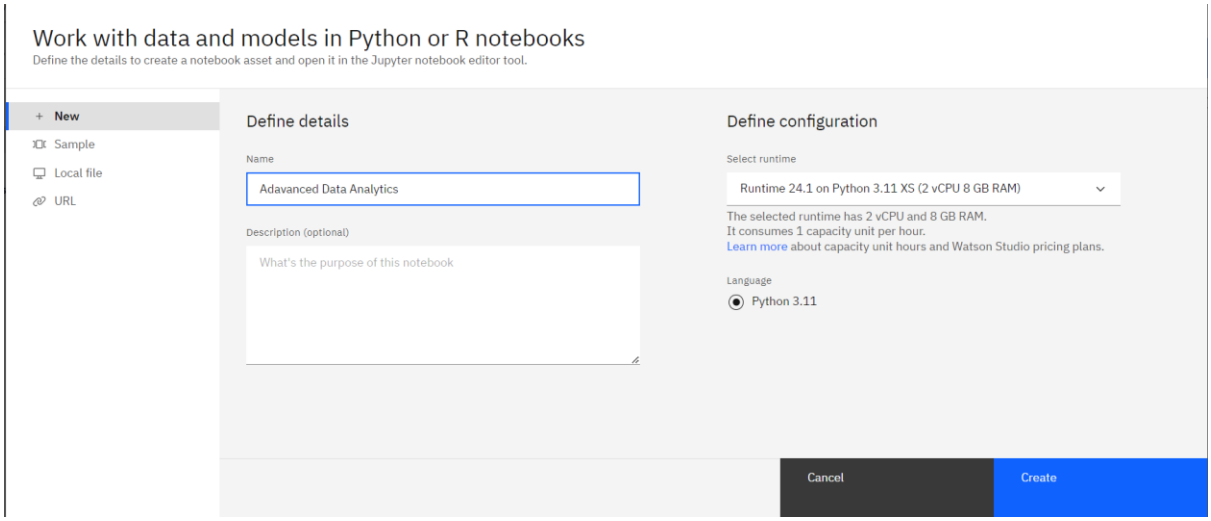
Step 6: Search for Jupyter Notebook

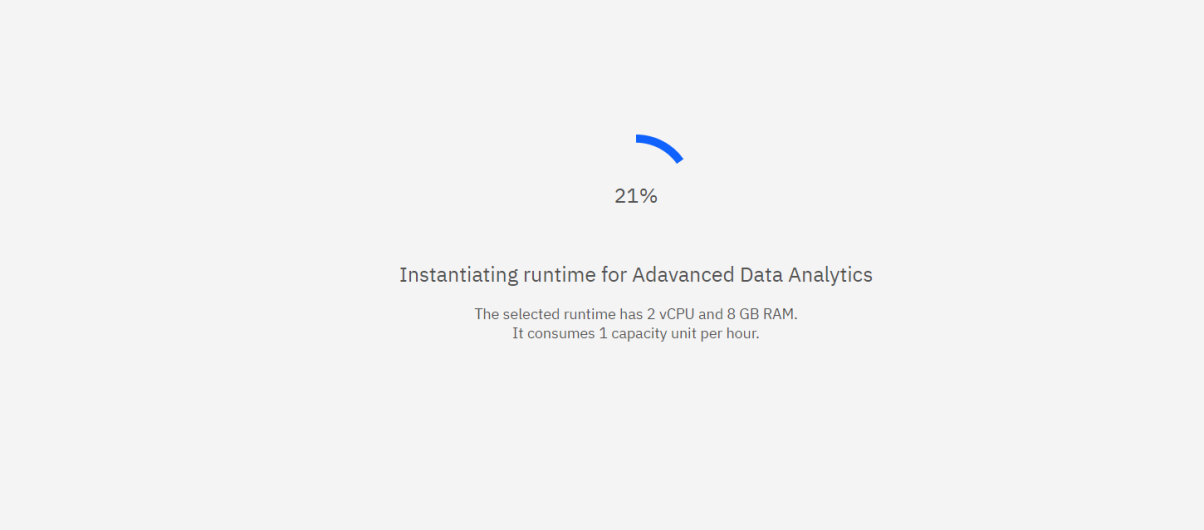
1. **Find Notebook**: In the asset creation menu, search for "Jupyter Notebook."
2. **Select Jupyter Notebook**: Choose the option to create a new Jupyter Notebook.



Step 7: Provide Notebook Details

1. **Name Your Notebook**: Enter a name for your Jupyter Notebook (e.g., "Iris Data Analysis").
2. **Select Environment**: Choose the appropriate Python environment if prompted.





Step 8: Install Required Libraries

1. **Install Libraries**: In the first cell of your notebook, run the following code to install necessary libraries:

```
pip install pandas matplotlib seaborn
```

```
Requirement already satisfied: pandas in /opt/conda/envs/Python-RT24.1/lib/python3.11/site-packages (2.1.4)
Requirement already satisfied: matplotlib in /opt/conda/envs/Python-RT24.1/lib/python3.11/site-packages (3.8.0)
Requirement already satisfied: seaborn in /opt/conda/envs/Python-RT24.1/lib/python3.11/site-packages (0.12.2)
Requirement already satisfied: numpy<2,>=1.23.2 in /opt/conda/envs/Python-RT24.1/lib/python3.11/site-packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in /opt/conda/envs/Python-RT24.1/lib/python3.11/site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /opt/conda/envs/Python-RT24.1/lib/python3.11/site-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.1 in /opt/conda/envs/Python-RT24.1/lib/python3.11/site-packages (from pandas) (2023.3)
Requirement already satisfied: contourpy>=1.0.1 in /opt/conda/envs/Python-RT24.1/lib/python3.11/site-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in /opt/conda/envs/Python-RT24.1/lib/python3.11/site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in /opt/conda/envs/Python-RT24.1/lib/python3.11/site-packages (from matplotlib) (4.51.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /opt/conda/envs/Python-RT24.1/lib/python3.11/site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: packaging>=20.0 in /opt/conda/envs/Python-RT24.1/lib/python3.11/site-packages (from matplotlib) (23.2)
Requirement already satisfied: pillow>=6.2.0 in /opt/conda/envs/Python-RT24.1/lib/python3.11/site-packages (from matplotlib) (10.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in /opt/conda/envs/Python-RT24.1/lib/python3.11/site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-RT24.1/lib/python3.11/site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

Step 9: Import Libraries

1. **Import Libraries**: In the next cell, import the libraries:

```
import seaborn as sns
import pandas as pd
```

Step 10: Load the Iris Dataset

1. **Fetch Dataset**: Load the Iris dataset using Seaborn:

```python
# Load the Iris dataset
iris = sns.load_dataset('iris')

# Display the first few rows of the dataset
print(iris.head())
```

```
   sepal_length  sepal_width  petal_length  petal_width species
0           5.1          3.5           1.4          0.2  setosa
1           4.9          3.0           1.4          0.2  setosa
2           4.7          3.2           1.3          0.2  setosa
3           4.6          3.1           1.5          0.2  setosa
4           5.0          3.6           1.4          0.2  setosa
```

Step 11: Explore the Data

1. **Basic Information**: Check the dataset's basic information and summary statistics:

```python
# Display basic information about the dataset
print(iris.info())

# Describe the numerical features
print(iris.describe())

# Check for missing values
print(iris.isnull().sum())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
None
```

```
       sepal_length  sepal_width  petal_length  petal_width
count    150.000000   150.000000    150.000000   150.000000
mean       5.843333     3.057333      3.758000     1.199333
std        0.828066     0.435866      1.765298     0.762238
min        4.300000     2.000000      1.000000     0.100000
25%        5.100000     2.800000      1.600000     0.300000
50%        5.800000     3.000000      4.350000     1.300000
75%        6.400000     3.300000      5.100000     1.800000
max        7.900000     4.400000      6.900000     2.500000
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

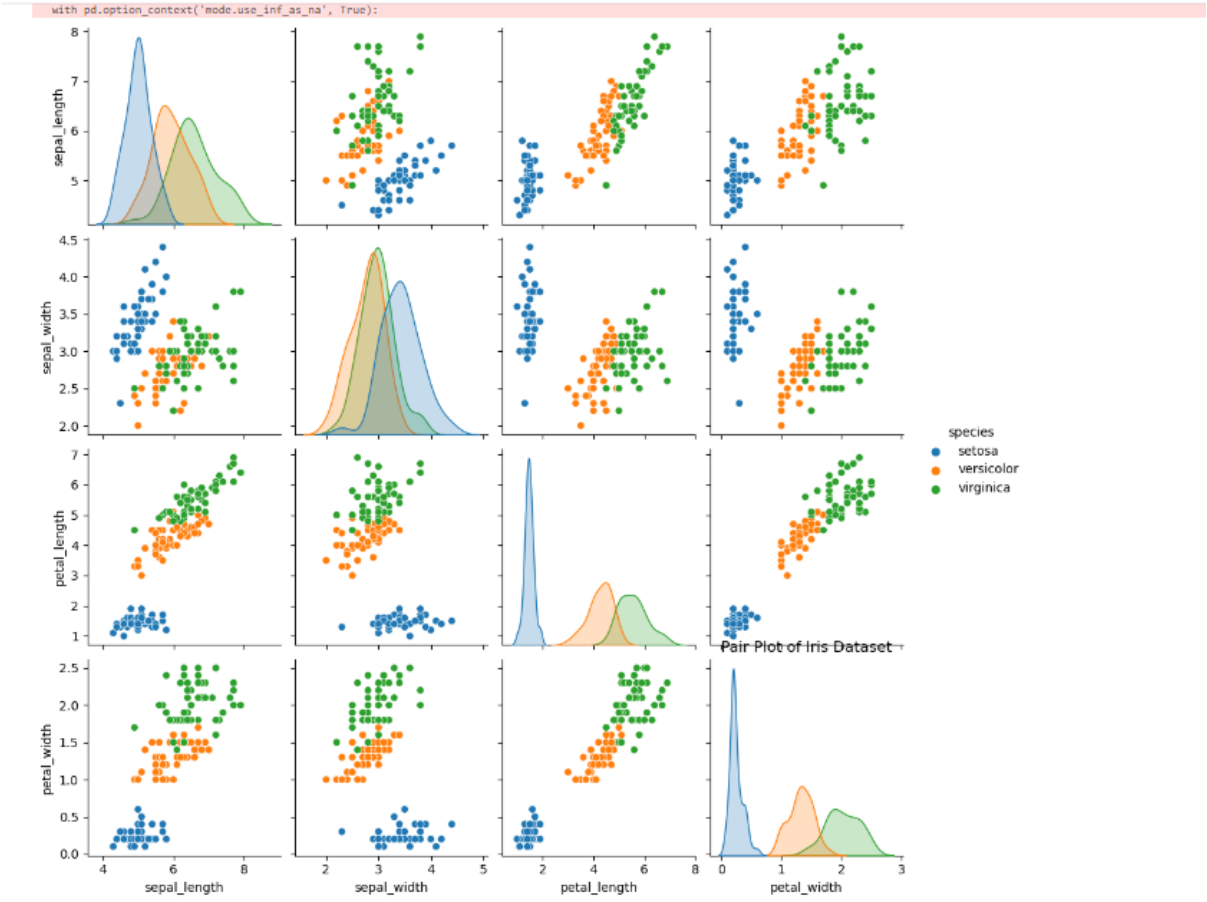
Step 12: Visualize the Data

1. **Create Visualizations**

- Pair plot

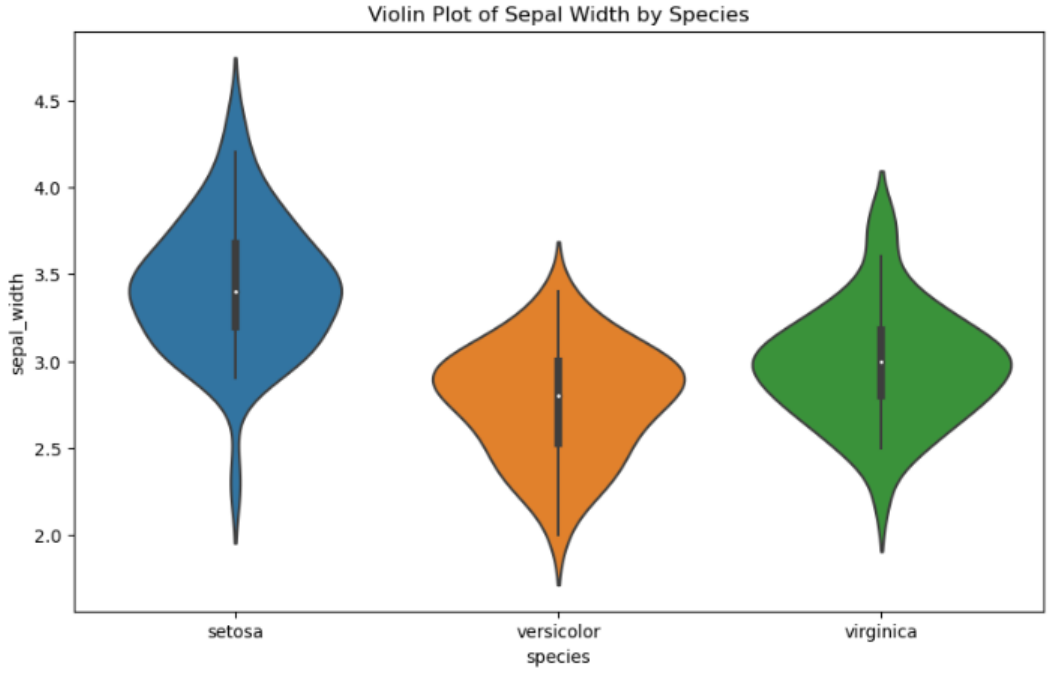
```
import matplotlib.pyplot as plt

# Pair plot to visualize relationships between features
sns.pairplot(iris, hue='species')
plt.title('Pair Plot of Iris Dataset')
plt.show()
```



- Violin plot:

```
# Violin plot for Sepal Width by Species
plt.figure(figsize=(10, 6))
sns.violinplot(x='species', y='sepal_width', data=iris)
plt.title('Violin Plot of Sepal Width by Species')
plt.show()
```



- Boxplot:

```
# Box plot for Sepal Length by Species
plt.figure(figsize=(10, 6))
sns.boxplot(x='species', y='sepal_length', data=iris)
plt.title('Box Plot of Sepal Length by Species')
plt.show()
```

Step 13: Advanced Data Analysis

1. **Group Analysis**: Conduct group analysis to find averages by species

```
# Group by species and calculate mean of each feature
grouped_data = iris.groupby('species').mean().reset_index()
print(grouped_data)
```

```
      species  sepal_length  sepal_width  petal_length  petal_width
0      setosa         5.006        3.428         1.462        0.246
1  versicolor         5.936        2.770         4.260        1.326
2   virginica         6.588        2.974         5.552        2.026
```