

pineapple.ipynb - JupyterLab

https://yellow-electrician-rlgpv.pwskills.app/lab/tree/work/pineapple.ipynb

File Edit View Run Kernel Tabs Settings Help

pineapple.ipynb

IRIS.csv

Python 3 (ipykernel)

[21]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import os
```

[22]:

```
df = pd.read_csv('IRIS.csv')
```

[23]:

```
df.head()
```

[23]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

[24]:

```
df = df.drop(columns = ['Id'])
```

[25]:

```
df.head()
```

[25]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa

Simple

0 1

Python 3 (ipykernel) | Idle

Mode: Command

Ln 5, Col 10

pineapple.ipynb

21°C Haze

Search

00:58 30-11-2023

pineapple.ipynb - JupyterLab

https://yellow-electrician-rlgpv.pwskills.app/lab/tree/work/pineapple.ipynb

90%

File Edit View Run Kernel Tabs Settings Help

pineapple.ipynb IRIS.csv

Python 3 (pykernel)

[21]:

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import os

[22]:

df = pd.read_csv('IRIS.csv')

[23]:

df.head()

[23]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

[24]:

df = df.drop(columns = ['Id'])

[25]:

df.head()

[25]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

Simple

0

1

Python 3 (ipykernel) | Idle

Mode: Command

Ln 5, Col 10

pineapple.ipynb

21°C
Haze

Search

ENG
IN

00:58
30-11-2023

pineapple.ipynb - JupyterLab

https://yellow-electrician-rlgqv.pwskills.app/lab/tree/work/pineapple.ipynb

90%

File Edit View Run Kernel Tabs Settings Help

pineapple.ipynb IRIS.csv

Python 3 (pykernel)

[26]: df.describe()

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
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

[27]: df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   SepalLengthCm    150 non-null   float64
1   SepalWidthCm     150 non-null   float64
2   PetalLengthCm    150 non-null   float64
3   PetalWidthCm     150 non-null   float64
4   Species          150 non-null   object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

[28]: df["Species"].value_counts()

Iris-setosa	50
Iris-versicolor	50

Simple 0 1 Python 3 (ipykernel) | Idle

Mode: Command Ln 5, Col 10 pineapple.ipynb

21°C Haze

Search

🏠

📅

📧

📺

🔍

📊

🔧

🔥

🛡️

🎵

📁

ENG IN

📶

🔊

🌙

00:58 30-11-2023

pineapple.ipynb - JupyterLab

https://yellow-electrician-rlgqv.pwskills.app/lab/tree/work/pineapple.ipynb

90%

File Edit View Run Kernel Tabs Settings Help

pineapple.ipynb IRIS.csv

memory usage: 6.0+ KB

[28]: df["Species"].value_counts()

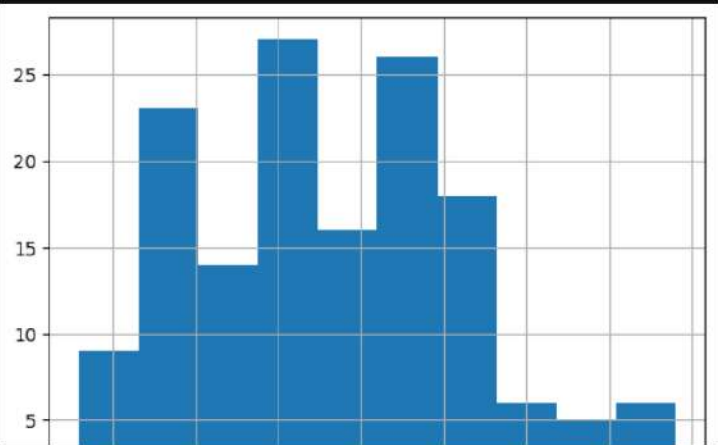
[28]: Iris-setosa 50
Iris-versicolor 50
Iris-virginica 50
Name: Species, dtype: int64

[29]: df.isnull().sum()

[29]: SepalLengthCm 0
SepalWidthCm 0
PetalLengthCm 0
PetalWidthCm 0
Species 0
dtype: int64

[30]: df["SepalLengthCm"].hist()

[30]: <AxesSubplot: >



The histogram displays the frequency distribution of sepal lengths (SepalLengthCm) for the Iris dataset. The x-axis represents the sepal length in centimeters, ranging from approximately 4.5 to 8.5. The y-axis represents the frequency, ranging from 0 to 25. The distribution is multimodal, with peaks around 5.0-5.5 cm and 6.5-7.0 cm. The bars are blue.

Bin Range (cm)	Frequency
4.5 - 5.0	9
5.0 - 5.5	23
5.5 - 6.0	14
6.0 - 6.5	27
6.5 - 7.0	16
7.0 - 7.5	26
7.5 - 8.0	18
8.0 - 8.5	6
8.5 - 9.0	6

Simple 0 1 Python 3 (ipykernel) | Idle

Mode: Command Ln 5, Col 10 pineapple.ipynb

21°C Haze Search

00:59 30-11-2023

pineapple.ipynb - JupyterLab

90%

https://yellow-electrician-rlgvp.pwskills.app/lab/tree/work/pineapple.ipynb

File Edit View Run Kernel Tabs Settings Help

pineapple.ipynb

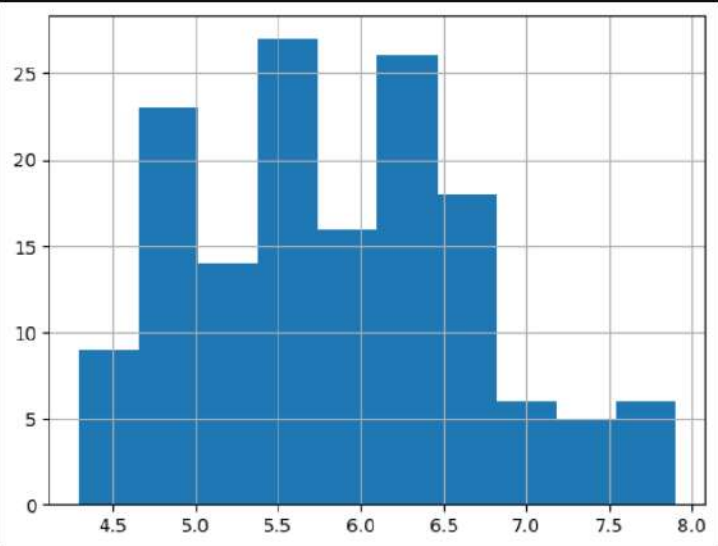
IRIS.csv

Python 3 (pykernel)

PetalLengthCm0
PetalWidthCm0
Species0
dtype: int64

[30]: df["SepalLengthCm"].hist()

[30]: <AxesSubplot: >

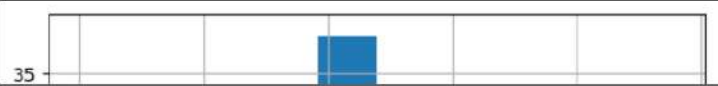


A histogram showing the frequency distribution of SepalLengthCm. The x-axis ranges from 4.5 to 8.0 with major ticks every 0.5 units. The y-axis ranges from 0 to 25 with major ticks every 5 units. The histogram consists of blue bars. The distribution is roughly bell-shaped but slightly skewed to the right. The highest frequency is around 5.5, with approximately 27 occurrences. Other notable peaks are around 4.8 (approx. 23), 6.2 (approx. 26), and 6.5 (approx. 18).

SepalLengthCm Range	Frequency
4.5 - 4.75	9
4.75 - 5.0	23
5.0 - 5.25	14
5.25 - 5.5	27
5.5 - 5.75	16
5.75 - 6.0	16
6.0 - 6.25	26
6.25 - 6.5	18
6.5 - 6.75	6
6.75 - 7.0	6
7.0 - 7.25	5
7.25 - 7.5	5
7.5 - 7.75	6

[31]: df["SepalWidthCm"].hist()

[31]: <AxesSubplot: >



A histogram showing the frequency distribution of SepalWidthCm. The x-axis is visible but mostly obscured. The y-axis has a label '35' at the bottom. Only one bar is clearly visible, centered around 6.0, with a frequency of approximately 35.

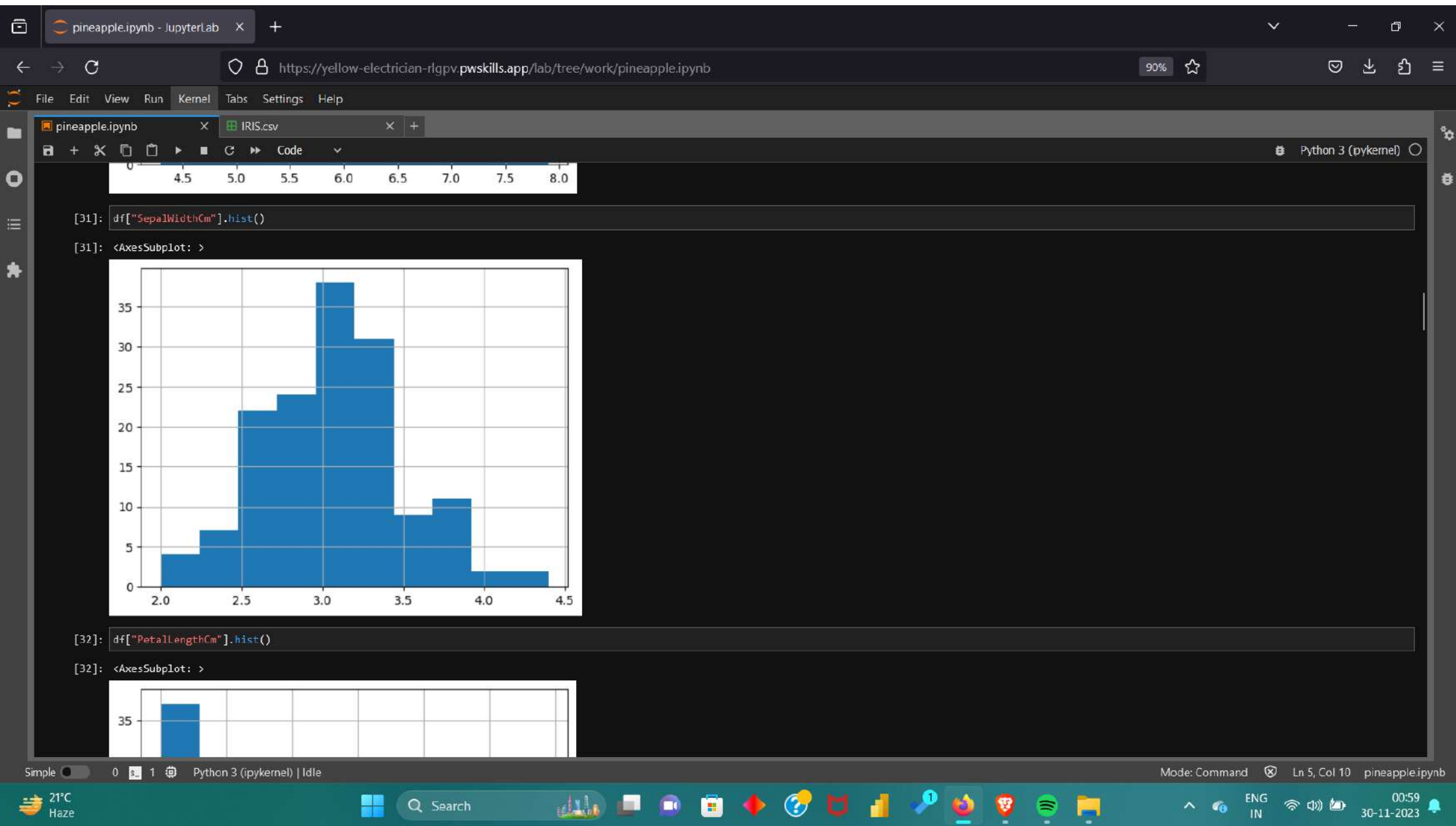
Simple 0 1 Python 3 (ipykernel) | Idle

Mode: Command Ln 5, Col 10 pineapple.ipynb

21°C Haze

Search

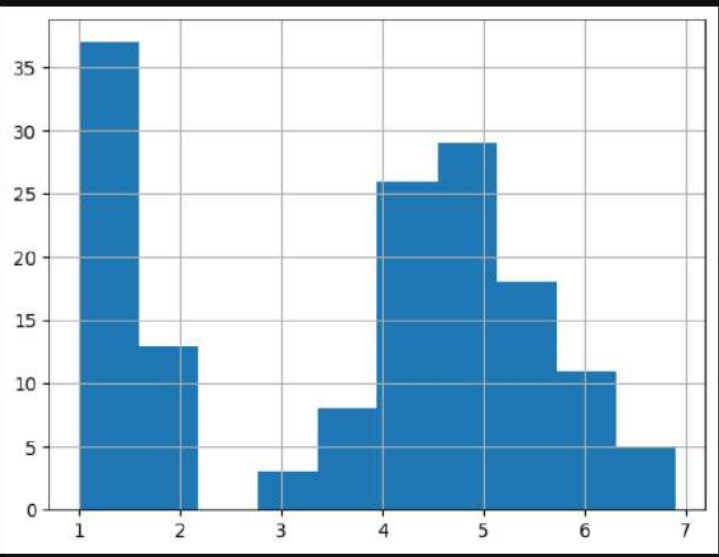
ENG IN 00:59 30-11-2023





```
[32]: df["PetalLengthCm"].hist()
```

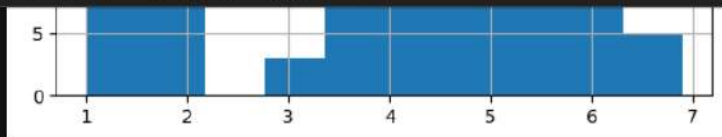
```
[32]: <AxesSubplot: >
```



```
[33]: df["PetalWidthCm"].hist()
```

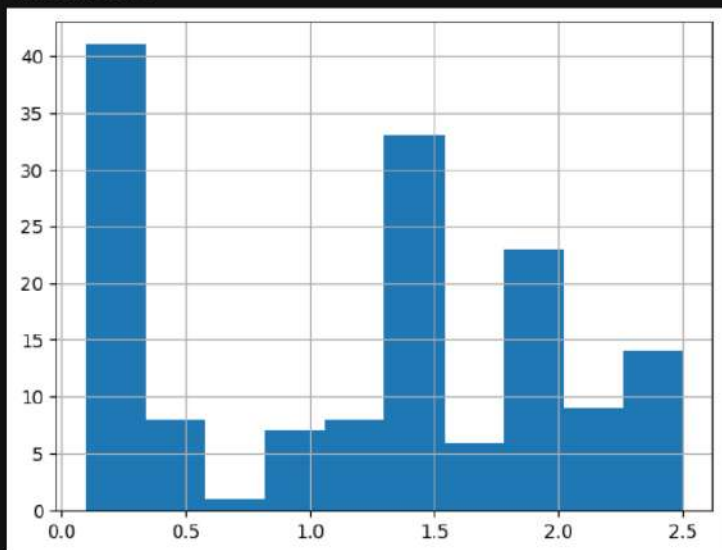
```
[33]: <AxesSubplot: >
```





```
[33]: df["PetalWidthCm"].hist()
```

```
[33]: <AxesSubplot: >
```



```
[34]: colours = ['red', 'green', 'purple']
species = ['Iris-virginica', 'Iris-versicolor', 'Iris-setosa']
```

```
[35]: for i in range(3):
    # df[df['Species'] == species[i]]
```


pineapple.ipynb - JupyterLab

https://yellow-electrician-rlgpv.pwskills.app/lab/tree/work/pineapple.ipynb

90%

File Edit View Run Kernel Tabs Settings Help

pineapple.ipynb IRIS.csv

Python 3 (pykernel)

0.0 0.5 1.0 1.5 2.0 2.5

[34]:

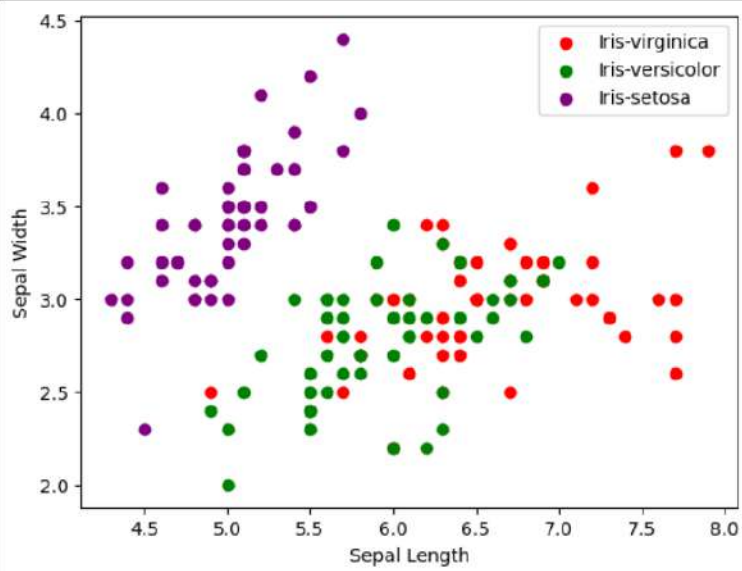
colours = ['red', 'green', 'purple']
species = ['Iris-virginica', 'Iris-versicolor', 'Iris-setosa']

[35]:

for i in range(3):
 x = df[df['Species'] == species[i]]
 plt.scatter(x['SepalLengthCm'], x['SepalWidthCm'], c = colours[i], label=species[i])
plt.xlabel("Sepal Length")
plt.ylabel("Sepal Width")
plt.legend()

[35]:

<matplotlib.legend.Legend at 0x7f34db03b6a0>



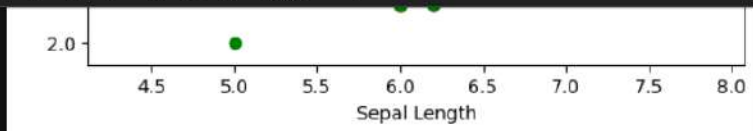
A scatter plot showing the relationship between Sepal Length (x-axis) and Sepal Width (y-axis) for three species of Iris: Iris-virginica (red), Iris-versicolor (green), and Iris-setosa (purple). The x-axis ranges from 4.5 to 8.0, and the y-axis ranges from 2.0 to 4.5. The plot shows distinct clusters for each species, with Iris-setosa having the highest Sepal Width values and Iris-virginica having the highest Sepal Length values. A legend in the top right corner identifies the species by color.

Simple 0 1 Python 3 (ipykernel) | Idle

Mode: Command Ln 5, Col 10 pineapple.ipynb

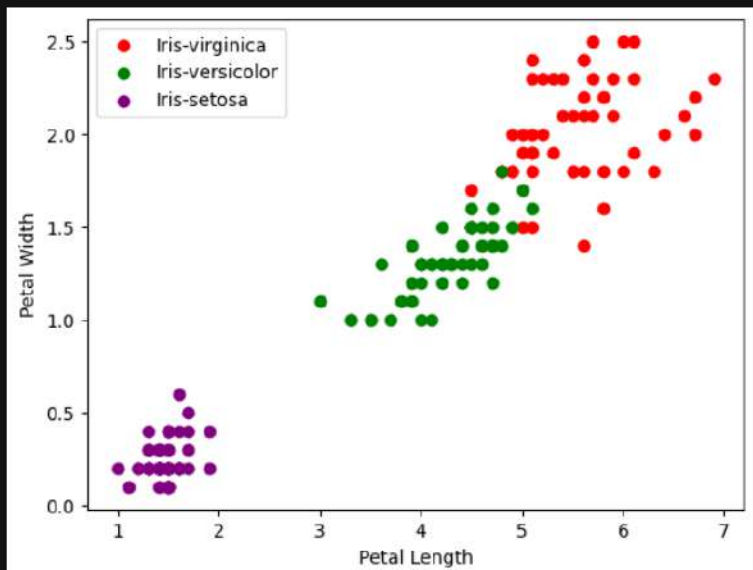
21°C Haze Search

00:59 30-11-2023



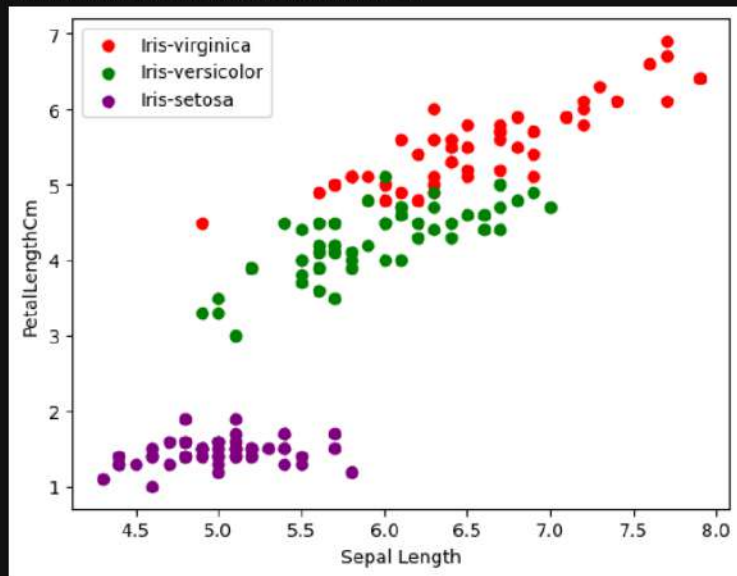
```
[36]: for i in range(3):
      x = df[df['Species'] == species[i]]
      plt.scatter(x['PetalLengthCm'], x['PetalWidthCm'], c = colours[i], label=species[i])
plt.xlabel("Petal Length")
plt.ylabel("Petal Width")
plt.legend()
```

[36]: <matplotlib.legend.Legend at 0x7f34daecf220>



```
[37]: for i in range(3):
      x = df[df['Species'] == species[i]]
      plt.scatter(x['SepalLengthCm'], x['PetalLengthCm'], c = colours[i], label=species[i])
      plt.xlabel("Sepal Length")
      plt.ylabel("PetalLengthCm")
      plt.legend()
```

[37]: <matplotlib.legend.Legend at 0x7F34daf5b940>

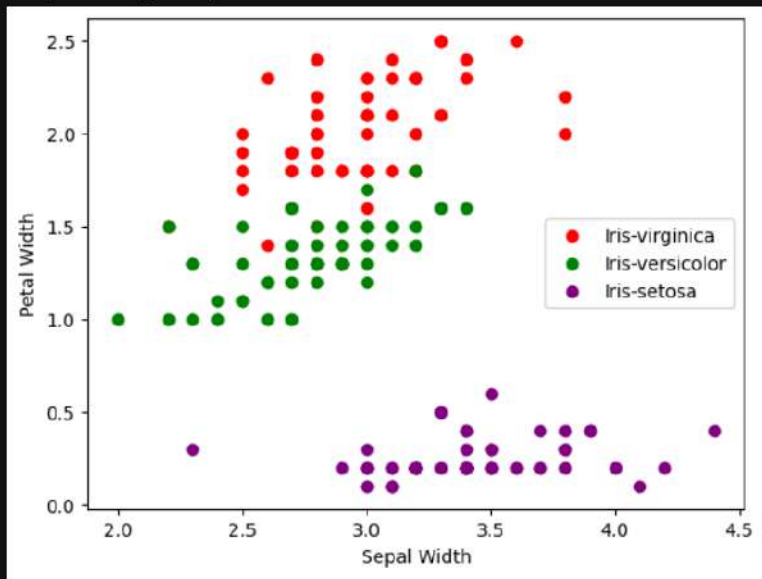


```
[38]: for i in range(3):
      x = df[df['Species'] == species[i]]
      plt.scatter(x['SepalWidthCm'], x['PetalWidthCm'], c = colours[i], label=species[i])
      plt.xlabel("Sepal Width")
```

Sepal Length

```
[38]: for i in range(3):
      x = df[df['Species'] == species[i]]
      plt.scatter(x['SepalWidthCm'], x['PetalWidthCm'], c = colours[i], label=species[i])
      plt.xlabel("Sepal Width")
      plt.ylabel("Petal Width")
      plt.legend()
```

[38]: <matplotlib.legend.Legend at 0x7f34dadf1ea0>



```
[83]: corr = df.corr()
fig, ax = plt.subplots(figsize=(6,6))
sns.heatmap(corr, annot=True, ax=ax, cmap = 'coolwarm')
```

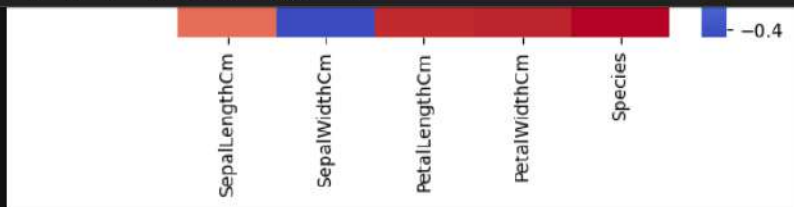
The screenshot displays a JupyterLab environment with a notebook titled 'pineapple.ipynb'. The code cell [87] contains the following Python code:

```
[87]: corr = df.corr()
fig, ax = plt.subplots(figsize=(6,5))
sns.heatmap(corr, annot=True, ax=ax, cmap = 'coolwarm')
```

The output of the code is a heatmap visualization of the correlation matrix for the Iris dataset. The x and y axes are labeled with the features: SepalLengthCm, SepalWidthCm, PetalLengthCm, PetalWidthCm, and Species. The color scale ranges from -0.4 (blue) to 1.0 (red). The diagonal elements are all 1.0. The off-diagonal elements show the following correlation values:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
SepalLengthCm	1	-0.11	0.87	0.82	0.78
SepalWidthCm	-0.11	1	-0.42	-0.36	-0.42
PetalLengthCm	0.87	-0.42	1	0.96	0.95
PetalWidthCm	0.82	-0.36	0.96	1	0.96
Species	0.78	-0.42	0.95	0.96	1

The bottom status bar shows the current mode is 'Command', the cursor is at line 1, column 10, and the notebook is titled 'pineapple.ipynb'.



[84]:

```
df.corr()
```

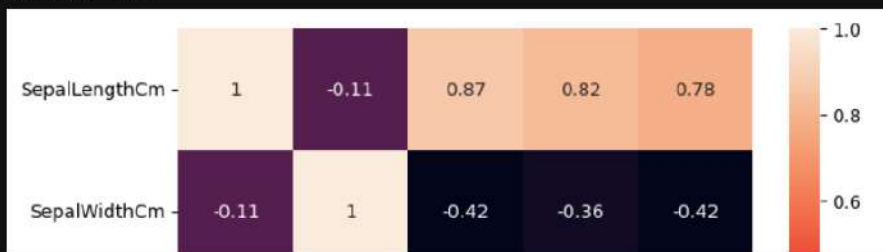
[84]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
SepalLengthCm	1.000000	-0.109369	0.871754	0.817954	0.782561
SepalWidthCm	-0.109369	1.000000	-0.420516	-0.356544	-0.419446
PetalLengthCm	0.871754	-0.420516	1.000000	0.962757	0.949043
PetalWidthCm	0.817954	-0.356544	0.962757	1.000000	0.956464
Species	0.782561	-0.419446	0.949043	0.956464	1.000000

[89]:

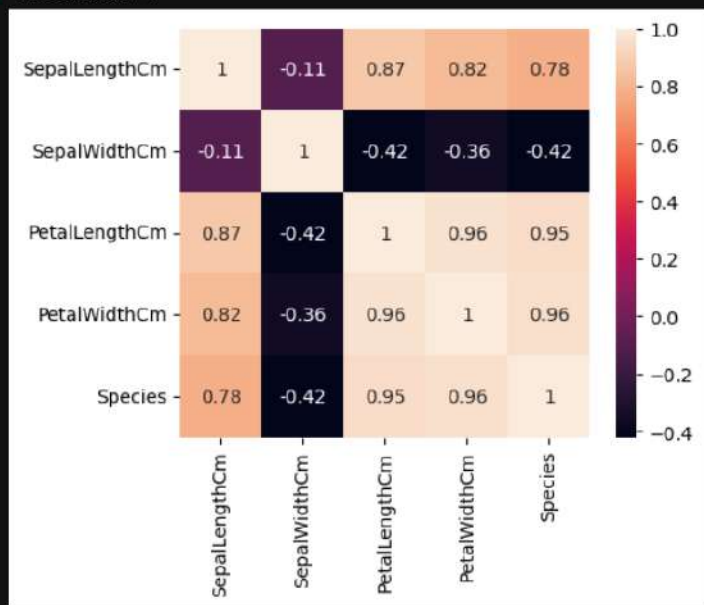
```
corr = df.corr()  
fig, ax = plt.subplots(figsize=(7,6))  
sns.heatmap(corr, annot=True)
```

[89]: <AxesSubplot: >




```
[91]: corr = df.corr()  
fig, ax = plt.subplots(figsize=(5,4))  
sns.heatmap(corr, annot=True)
```

[91]: <AxesSubplot: >



```
[45]: from sklearn.preprocessing import LabelEncoder  
le = LabelEncoder()
```

```
[56]: df["Species"] = le.fit_transform(df["Species"])  
df.head()
```


pineapple.ipynb - JupyterLab

https://yellow-electrician-rlgvp.pwskills.app/lab/tree/work/pineapple.ipynb

90%

File Edit View Run Kernel Tabs Settings Help

pineapple.ipynb IRIS.csv

Python 3 (pykernel)

[45]:

from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()

[56]:

df["Species"] = le.fit_transform(df["Species"])
df.head()

[56]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

[70]:

from sklearn.model_selection import train_test_split
train - 70
test - 30
X = df.drop(columns=['Species'])
Y = df['Species']
x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.30)

[71]:

from sklearn.linear_model import LogisticRegression
model = LogisticRegression()

[86]:

model.fit(x_train, y_train)

[86]:

DecisionTreeClassifier
DecisionTreeClassifier()

[73]:

print("Accuracy: ", model.score(x_test, y_test))

Simple

0 1 Python 3 (ipykernel) | Idle

Mode: Command Ln 2, Col 35 pineapple.ipynb

21°C Haze

Search

ENG IN

01:01 30-11-2023

pineapple.ipynb - JupyterLab

https://yellow-electrician-rlgpv.pwskills.app/lab/tree/work/pineapple.ipynb

90%

☆

🔒

📄

🔗

☰

File Edit View Run Kernel Tabs Settings Help

pineapple.ipynb IRIS.csv

Python 3 (pykernel)

```
[73]: print("Accuracy: ", model.score(x_test, y_test))
Accuracy: 0.9333333333333333

[74]: print("Accuracy: ", model.score(x_test, y_test) * 100)
Accuracy: 93.33333333333333

[ ]:

[75]: from sklearn.neighbors import KNeighborsClassifier
model = KNeighborsClassifier()

[76]: model.fit(x_train, y_train)

[76]: KNeighborsClassifier
KNeighborsClassifier()

[77]: print("Accuracy: ", model.score(x_test, y_test) * 100)
Accuracy: 97.77777777777777

[78]: from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier()

[79]: model.fit(x_train, y_train)

[79]: DecisionTreeClassifier
DecisionTreeClassifier()

[80]: print("Accuracy: ", model.score(x_test, y_test) * 100)
Accuracy: 95.55555555555556

[ ]:
```

Simple 0 1 Python 3 (ipykernel) | Idle Mode: Command Ln 2, Col 35 pineapple.ipynb

21°C Clear 🔍 Search 🏠 📅 🔄 🔍 📊 📌 🔒 🎧 📁 ⬆️ 📶 🔊 🗓️ 01:02 30-11-2023 🔔