Data Visualization III

Download the Iris flower dataset or any other dataset into a DataFrame. (e.g., https://archive.ics.uci.edu/ml/datasets/Iris). Scan the dataset and give the inference as:

1. List down the features and their types (e.g., numeric, nominal) available in the dataset.
2. Create a histogram for each feature in the dataset to illustrate the feature distributions.
3. Create a boxplot for each feature in the dataset.
4. Compare distributions and identify outliers.

In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
df = pd.read_csv("/home/mca01/Downloads/Iris.csv")
```

In [3]:
```
df.head()
```

Out[3]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Variety
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

In [4]:
```
df.shape
```

Out[4]:
```
(150, 6)
```

In [5]:
```
df.describe()
```

Out[5]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidt
count	150.000000	150.000000	150.000000	150.000000	150.00
mean	75.500000	5.843333	3.054000	3.758667	1.19
std	43.445368	0.828066	0.433594	1.764420	0.76
min	1.000000	4.300000	2.000000	1.000000	0.10
25%	38.250000	5.100000	2.800000	1.600000	0.30
50%	75.500000	5.800000	3.000000	4.350000	1.30
75%	112.750000	6.400000	3.300000	5.100000	1.80
max	150.000000	7.900000	4.400000	6.900000	2.50

In [6]:
```
df.tail()
```

Out[6]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Va
145	146	6.7	3.0	5.2	2.3	virg
146	147	6.3	2.5	5.0	1.9	virg
147	148	6.5	3.0	5.2	2.0	virg
148	149	6.2	3.4	5.4	2.3	virg
149	150	5.9	3.0	5.1	1.8	virg

In [7]:
```
df.mean()
```

```
/tmp/ipykernel_3465/3698961737.py:1: FutureWarning: Dropping of nuisance col
umns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a
future version this will raise TypeError.  Select only valid columns before
calling the reduction.
  df.mean()
```

Out[7]:
```
Id               75.500000
SepalLengthCm     5.843333
SepalWidthCm      3.054000
PetalLengthCm     3.758667
PetalWidthCm      1.198667
dtype: float64
```

In [8]:
```
df.std()
```

```
/tmp/ipykernel_3465/3390915376.py:1: FutureWarning: Dropping of nuisance col
umns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a
future version this will raise TypeError.  Select only valid columns before
calling the reduction.
  df.std()
```

Out[8]:
```
Id               43.445368
SepalLengthCm     0.828066
SepalWidthCm      0.433594
PetalLengthCm     1.764420
PetalWidthCm      0.763161
dtype: float64
```

In [9]:
```
df.mode()
```

Out[9]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Va
0	1	5.0	3.0	1.5	0.2	s
1	2	NaN	NaN	NaN	NaN	vers
2	3	NaN	NaN	NaN	NaN	vir
3	4	NaN	NaN	NaN	NaN	
4	5	NaN	NaN	NaN	NaN	
...	...	...	...	...	...	
145	146	NaN	NaN	NaN	NaN	
146	147	NaN	NaN	NaN	NaN	
147	148	NaN	NaN	NaN	NaN	
148	149	NaN	NaN	NaN	NaN	
149	150	NaN	NaN	NaN	NaN	

150 rows × 6 columns

In [10]:
```
df.cov()
```

Out[10]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm
Id	1887.500000	25.782886	-7.492282	67.667785
SepalLengthCm	25.782886	0.685694	-0.039268	1.273682
SepalWidthCm	-7.492282	-0.039268	0.188004	-0.321713
PetalLengthCm	67.667785	1.273682	-0.321713	3.113179
PetalWidthCm	29.832215	0.516904	-0.117981	1.296387

In [11]:
```
df.mode()
```

Out[11]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Va
0	1	5.0	3.0	1.5	0.2	s
1	2	NaN	NaN	NaN	NaN	vers
2	3	NaN	NaN	NaN	NaN	vir
3	4	NaN	NaN	NaN	NaN	
4	5	NaN	NaN	NaN	NaN	
...	...	...	...	...	...	
145	146	NaN	NaN	NaN	NaN	
146	147	NaN	NaN	NaN	NaN	
147	148	NaN	NaN	NaN	NaN	
148	149	NaN	NaN	NaN	NaN	
149	150	NaN	NaN	NaN	NaN	

150 rows × 6 columns

In [12]:
```
df.median()
```

```
/tmp/ipykernel_3465/222071786.py:1: FutureWarning: Dropping of nuisance colu
mns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a f
uture version this will raise TypeError.  Select only valid columns before c
alling the reduction.
  df.median()
```

Out[12]:
```
Id               75.50
SepalLengthCm     5.80
SepalWidthCm      3.00
PetalLengthCm     4.35
PetalWidthCm      1.30
dtype: float64
```

In [13]:
```
df.var()
```

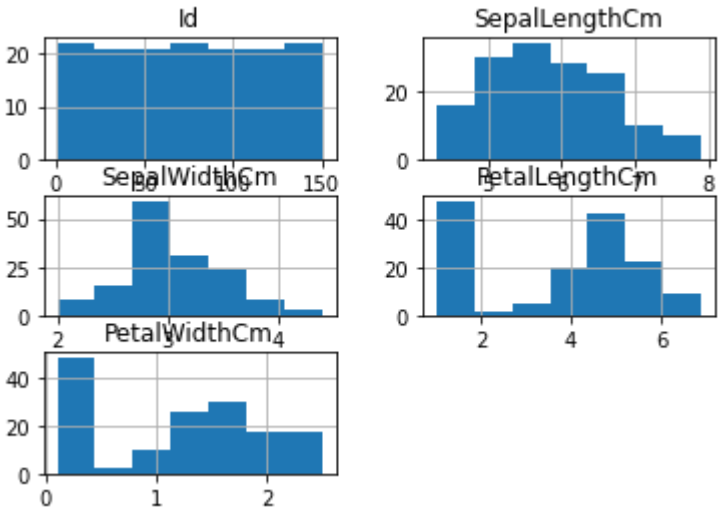
```
/tmp/ipykernel_3465/1568254755.py:1: FutureWarning: Dropping of nuisance col
umns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a
future version this will raise TypeError.  Select only valid columns before
calling the reduction.
  df.var()
```

Out[13]:
```
Id               1887.500000
SepalLengthCm       0.685694
SepalWidthCm        0.188004
PetalLengthCm       3.113179
PetalWidthCm        0.582414
dtype: float64
```

In [14]:
```
import seaborn as sns
import matplotlib.pyplot as plt
```

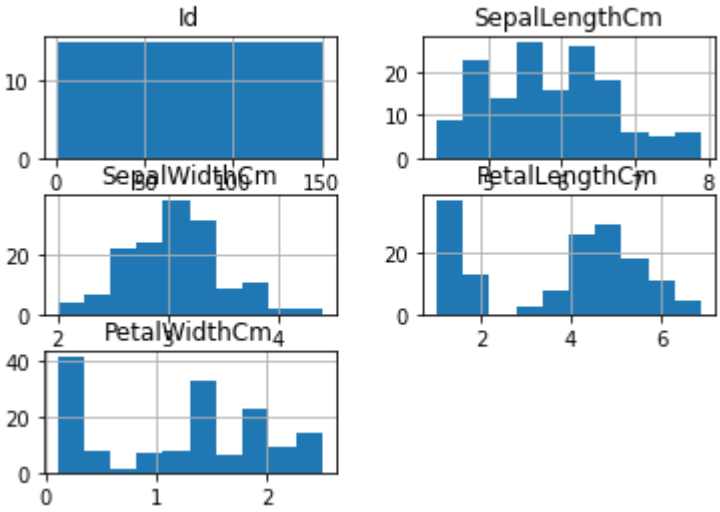
In [16]:
```
df.hist(bins=7)
```

Out[16]:
```
array([[<AxesSubplot:title={'center':'Id'}>,
        <AxesSubplot:title={'center':'SepalLengthCm'}>],
       [<AxesSubplot:title={'center':'SepalWidthCm'}>,
        <AxesSubplot:title={'center':'PetalLengthCm'}>],
       [<AxesSubplot:title={'center':'PetalWidthCm'}>, <AxesSubplot:>]],
      dtype=object)
```



In [17]:
```
df.hist()
```

Out[17]:
```
array([[<AxesSubplot:title={'center':'Id'}>,
        <AxesSubplot:title={'center':'SepalLengthCm'}>],
       [<AxesSubplot:title={'center':'SepalWidthCm'}>,
        <AxesSubplot:title={'center':'PetalLengthCm'}>],
       [<AxesSubplot:title={'center':'PetalWidthCm'}>, <AxesSubplot:>]],
      dtype=object)
```



In [18]:
```
df.columns
```

Out[18]:
```
Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthC
m',
       'Variety'],
      dtype='object')
```

In [20]:
```
numeric_cols=['Id','SepalLengthCm','SepalWidthCm','PetalLengthCm','PetalWidt
np.min(df[numeric_cols])
```

Out[20]:
```
Id                          1
SepalLengthCm             4.3
SepalWidthCm              2.0
PetalLengthCm             1.0
PetalWidthCm              0.1
Variety           Iris-setosa
dtype: object
```

In [21]:
```
np.max(df[numeric_cols])
```

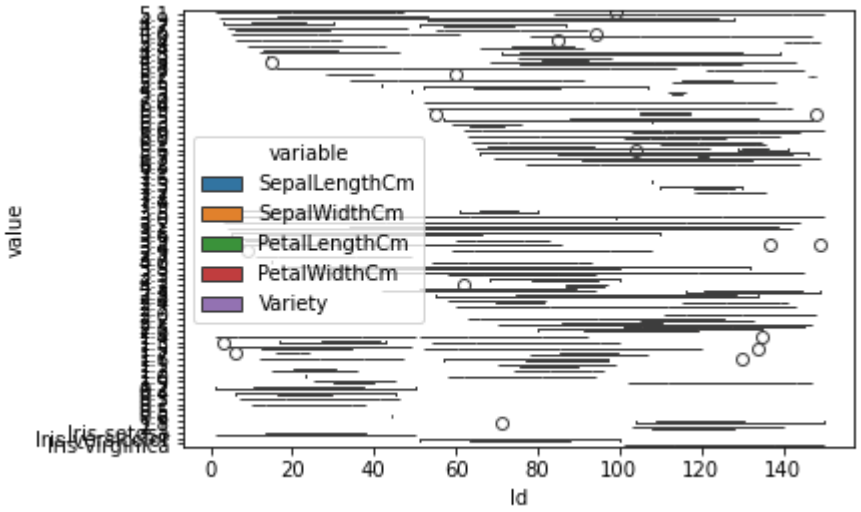
Out[21]:
```
Id                          150
SepalLengthCm               7.9
SepalWidthCm                4.4
PetalLengthCm               6.9
PetalWidthCm                2.5
Variety           Iris-virginica
dtype: object
```

In [22]:
```
df.quantile([0.0,0.1,0.5,1.0],numeric_only=True)
```

Out[22]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0.0	1.0	4.3	2.0	1.00	0.1
0.1	15.9	4.8	2.5	1.40	0.2
0.5	75.5	5.8	3.0	4.35	1.3
1.0	150.0	7.9	4.4	6.90	2.5

In [24]:
```
iris_long = pd.melt(df, id_vars='Id')
ax = sns.boxplot(x="Id", y="value",hue="variable", data=iris_long)
plt.show()
```

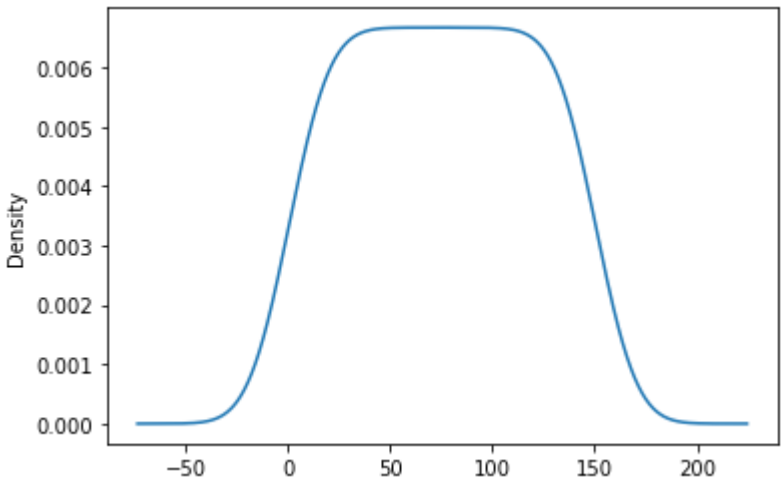


In [26]:
```
df['Id'].value_counts()
```

Out[26]:
```
1      1
95     1
97     1
98     1
99     1
      ..
51     1
52     1
53     1
54     1
150    1
Name: Id, Length: 150, dtype: int64
```

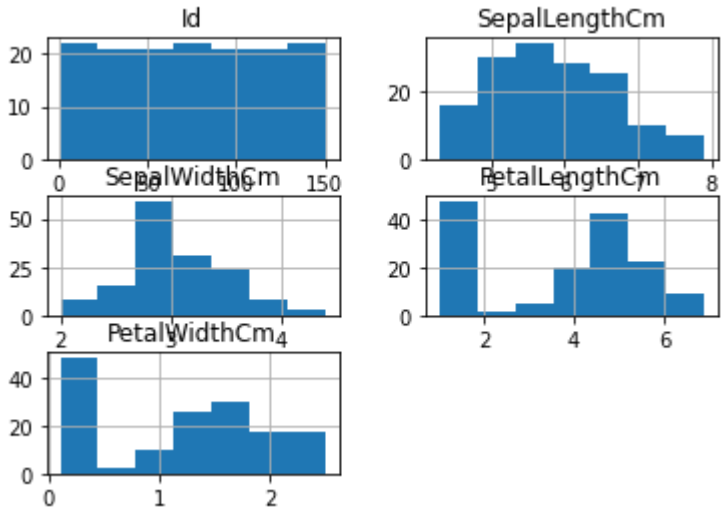
In [27]:
```
df['Id'].plot.density()
```

Out[27]: <AxesSubplot:ylabel='Density'>



In [29]:
```
df.hist(bins=7)
```

Out[29]:
```
array([[<AxesSubplot:title={'center':'Id'}>,
        <AxesSubplot:title={'center':'SepalLengthCm'}>],
       [<AxesSubplot:title={'center':'SepalWidthCm'}>,
        <AxesSubplot:title={'center':'PetalLengthCm'}>],
       [<AxesSubplot:title={'center':'PetalWidthCm'}>, <AxesSubplot:>]],
      dtype=object)
```



In [30]:
```
sns.heatmap(df.corr(), annot=True) #Correlation is feature to feature relati
```

Out[30]:
```
<AxesSubplot:>
```