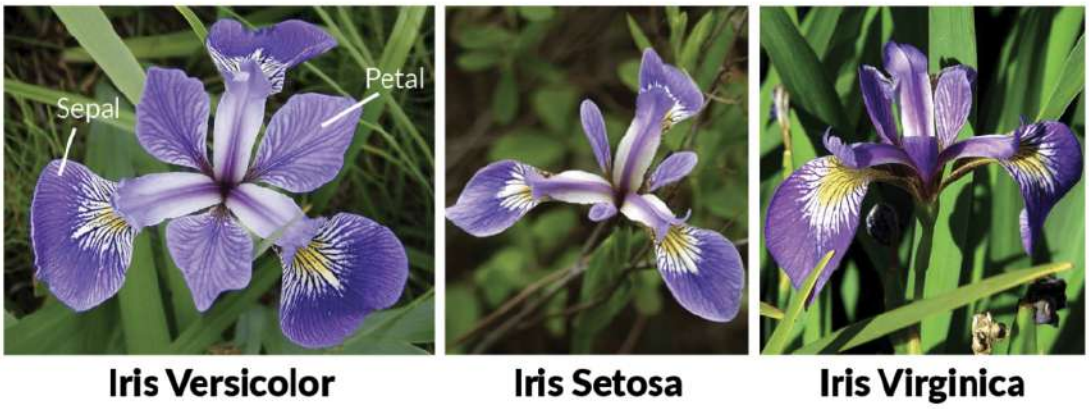
Dataset_Iris

April 2, 2025

[1]:
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
```

[2]:
```
from IPython.display import Image
Image('2018-10-17-classificacao-iris-01.png')
```

[2]:



[3]:
```
df = pd.read_csv('iris.csv')
```

[4]:
```
df.info()
```

```
<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
```

```
[5]: df.head(10)
```

```
[5]:    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
     5           5.4          3.9           1.7          0.4  Setosa
     6           4.6          3.4           1.4          0.3  Setosa
     7           5.0          3.4           1.5          0.2  Setosa
     8           4.4          2.9           1.4          0.2  Setosa
     9           4.9          3.1           1.5          0.1  Setosa
```

```
[6]: df.tail(10)
```

```
[6]:      sepal.length  sepal.width  petal.length  petal.width    species
     140           6.7          3.1           5.6          2.4  Virginica
     141           6.9          3.1           5.1          2.3  Virginica
     142           5.8          2.7           5.1          1.9  Virginica
     143           6.8          3.2           5.9          2.3  Virginica
     144           6.7          3.3           5.7          2.5  Virginica
     145           6.7          3.0           5.2          2.3  Virginica
     146           6.3          2.5           5.0          1.9  Virginica
     147           6.5          3.0           5.2          2.0  Virginica
     148           6.2          3.4           5.4          2.3  Virginica
     149           5.9          3.0           5.1          1.8  Virginica
```

```
[7]: df['species'].unique()
```

```
[7]: array(['Setosa', 'Versicolor', 'Virginica'], dtype=object)
```

```
[8]: df.describe()
```

```
[8]:        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
```

```
[9]: df.isnull()
```

```
[9]:      sepal.length  sepal.width  petal.length  petal.width  species
     0           False        False         False        False    False
```

```
1            False         False          False         False     False
2            False         False          False         False     False
3            False         False          False         False     False
4            False         False          False         False     False
..             …           …           …          …        …
145          False         False          False         False     False
146          False         False          False         False     False
147          False         False          False         False     False
148          False         False          False         False     False
149          False         False          False         False     False

[150 rows x 5 columns]
```

[10]:
```
df.isnull().sum()
```

[10]:
```
sepal.length    0
sepal.width     0
petal.length    0
petal.width     0
species         0
dtype: int64
```

[11]:
```
data = df.drop_duplicates(subset='species')
```

[12]:
```
data
```

[12]:
```
     sepal.length  sepal.width  petal.length  petal.width     species
0             5.1          3.5           1.4          0.2      Setosa
50            7.0          3.2           4.7          1.4  Versicolor
100           6.3          3.3           6.0          2.5   Virginica
```

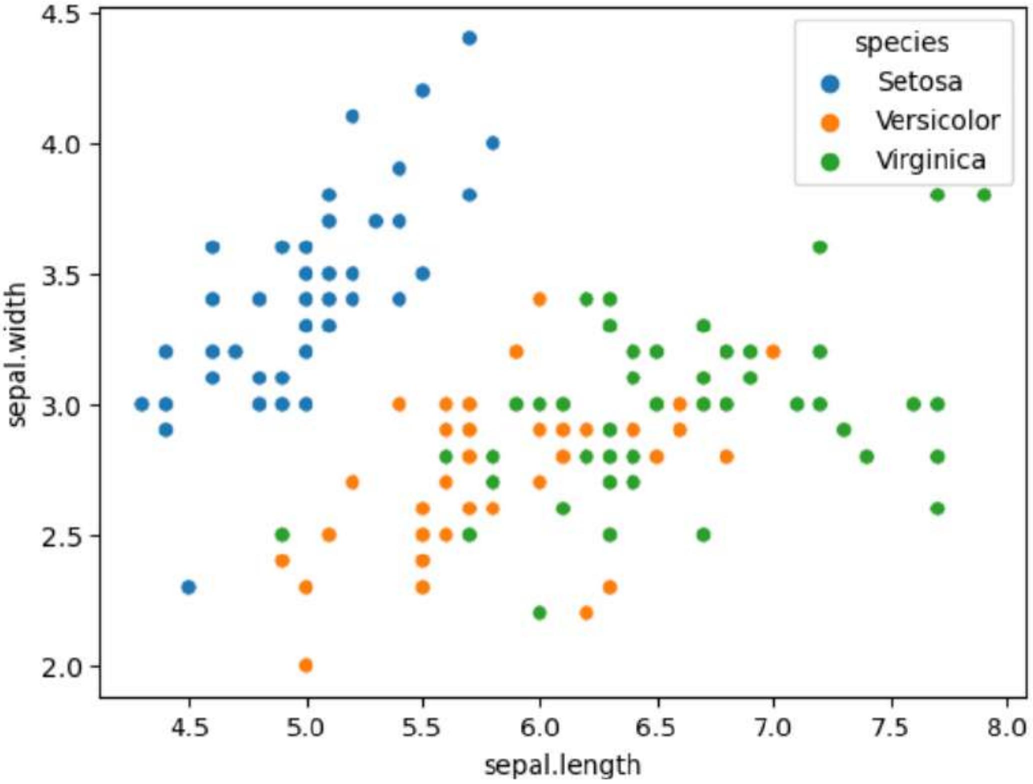
[13]:
```
df.value_counts('species')
```

[13]:
```
species
Setosa        50
Versicolor    50
Virginica     50
dtype: int64
```

[14]:
```
import seaborn as sns
```

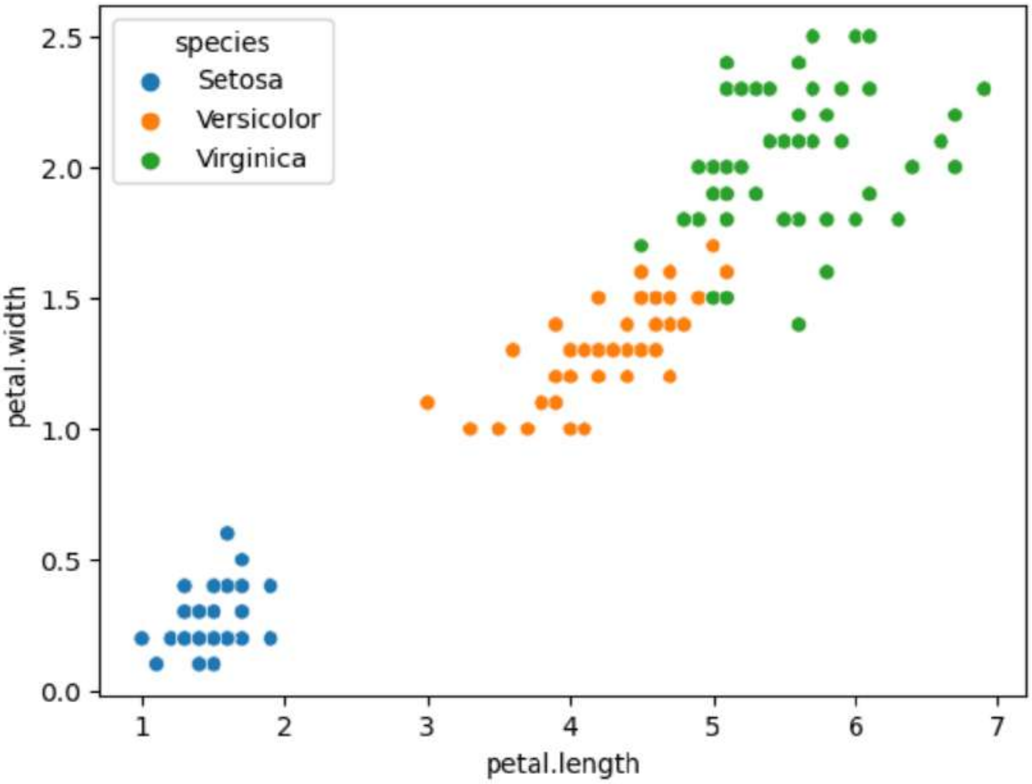
[15]:
```
sns.scatterplot(x='sepal.length', y='sepal.width',
                hue='species', data=df)
```

[15]:
```
<AxesSubplot: xlabel='sepal.length', ylabel='sepal.width'>
```



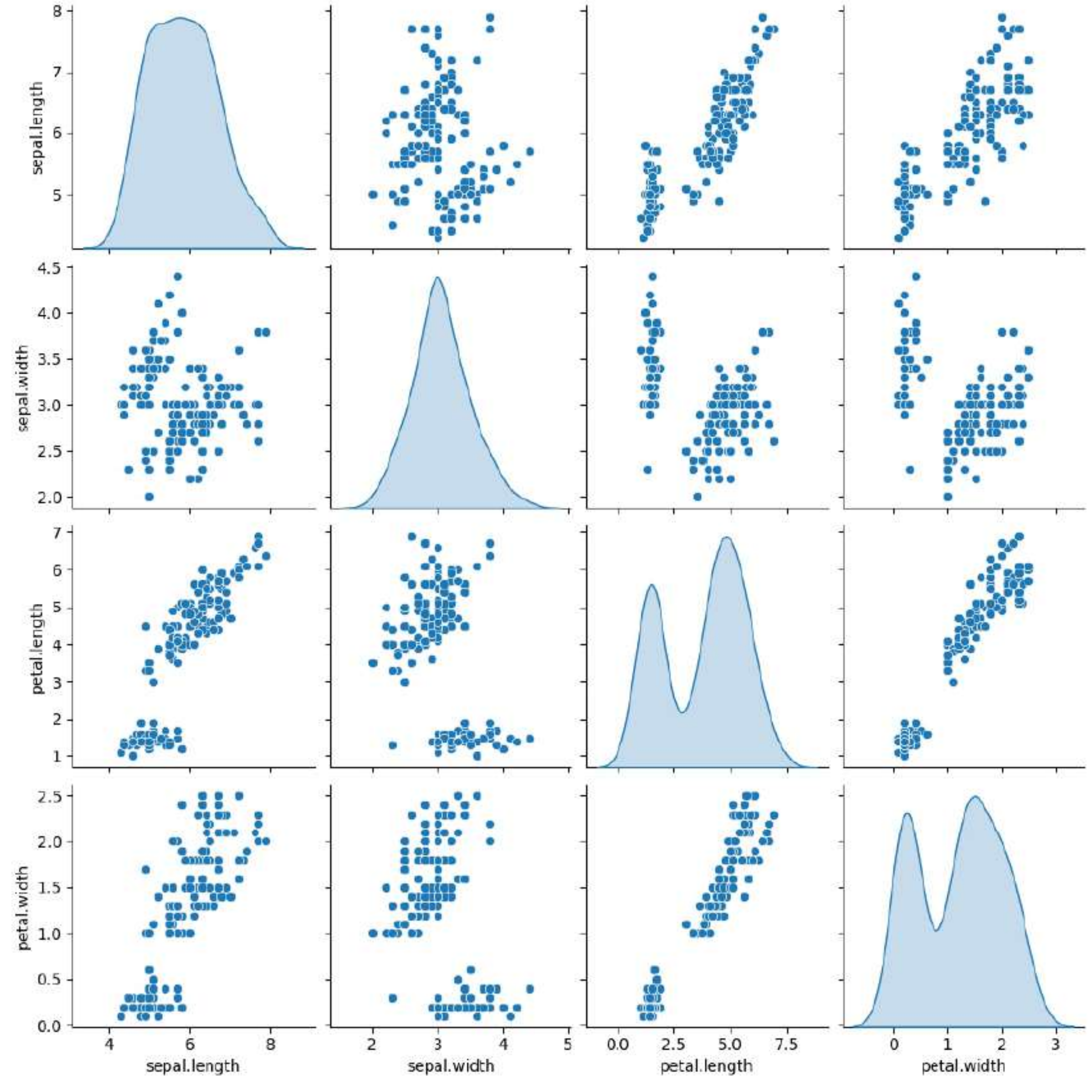
```
[16]: sns.scatterplot(x='petal.length', y='petal.width',
                      hue='species', data=df)
```

```
[16]: <AxesSubplot: xlabel='petal.length', ylabel='petal.width'>
```



```
[17]: sns.pairplot(df, diag_kind='kde')
```

```
[17]: <seaborn.axisgrid.PairGrid at 0x7f8ce0ce3c10>
```

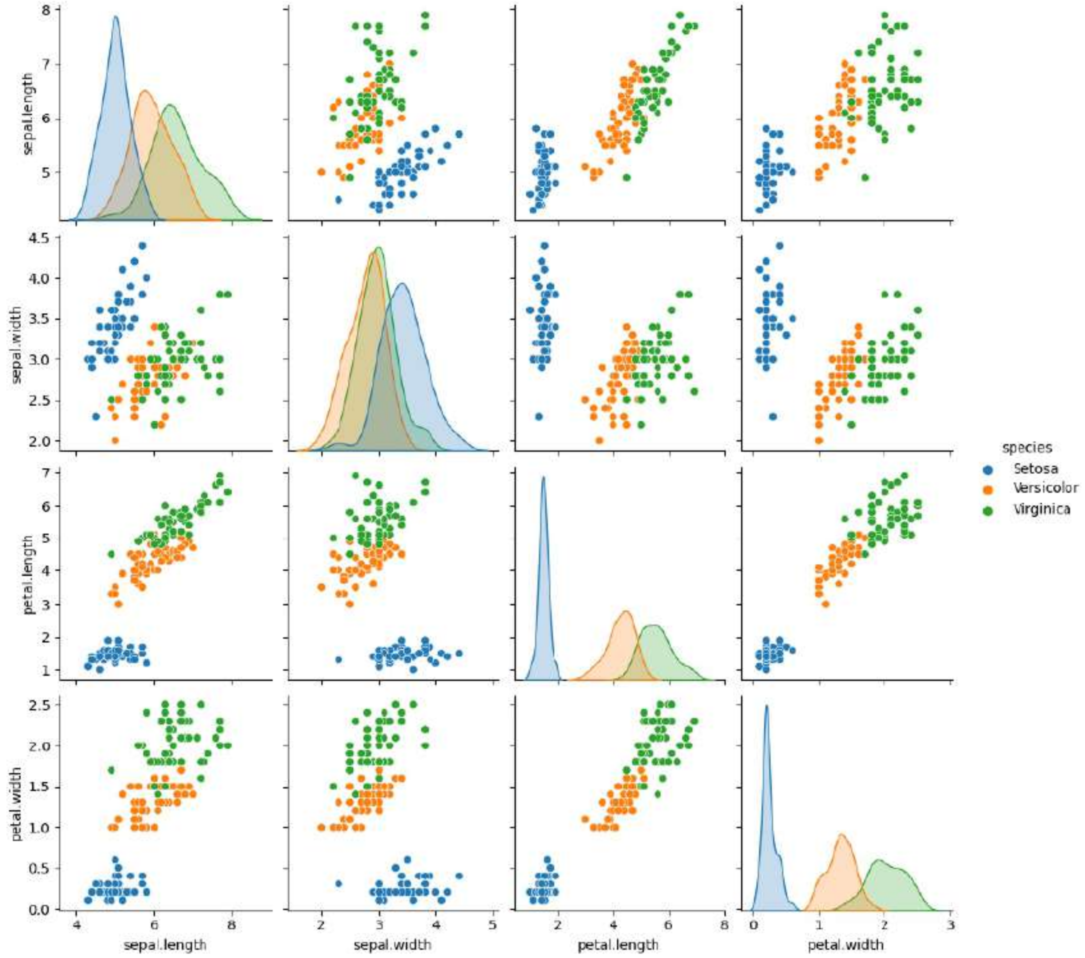


[18]:

```
sns.pairplot(df, hue='species')
```

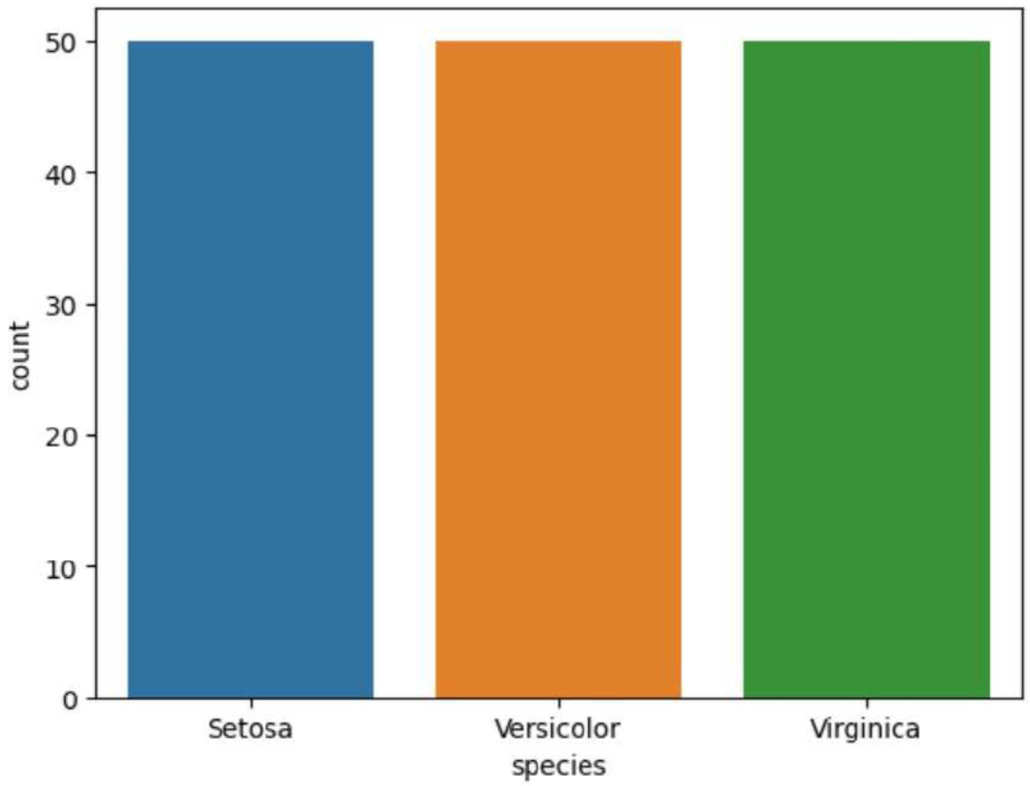
[18]:

```
<seaborn.axisgrid.PairGrid at 0x7f8ce0459950>
```



```
[19]: sns.countplot(x='species', data=df)
```

```
[19]: <AxesSubplot: xlabel='species', ylabel='count'>
```



[20]:
```
df
```

[20]:
```
     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
..            …            …             …            …        …
145           6.7          3.0           5.2          2.3  Virginica
146           6.3          2.5           5.0          1.9  Virginica
147           6.5          3.0           5.2          2.0  Virginica
148           6.2          3.4           5.4          2.3  Virginica
149           5.9          3.0           5.1          1.8  Virginica

[150 rows x 5 columns]
```

[21]:
```
fig, axes = plt.subplots(2,2, figsize=(8,8))
#titulo
axes[0,0].set_title('Sepal Length')
#elemento
```

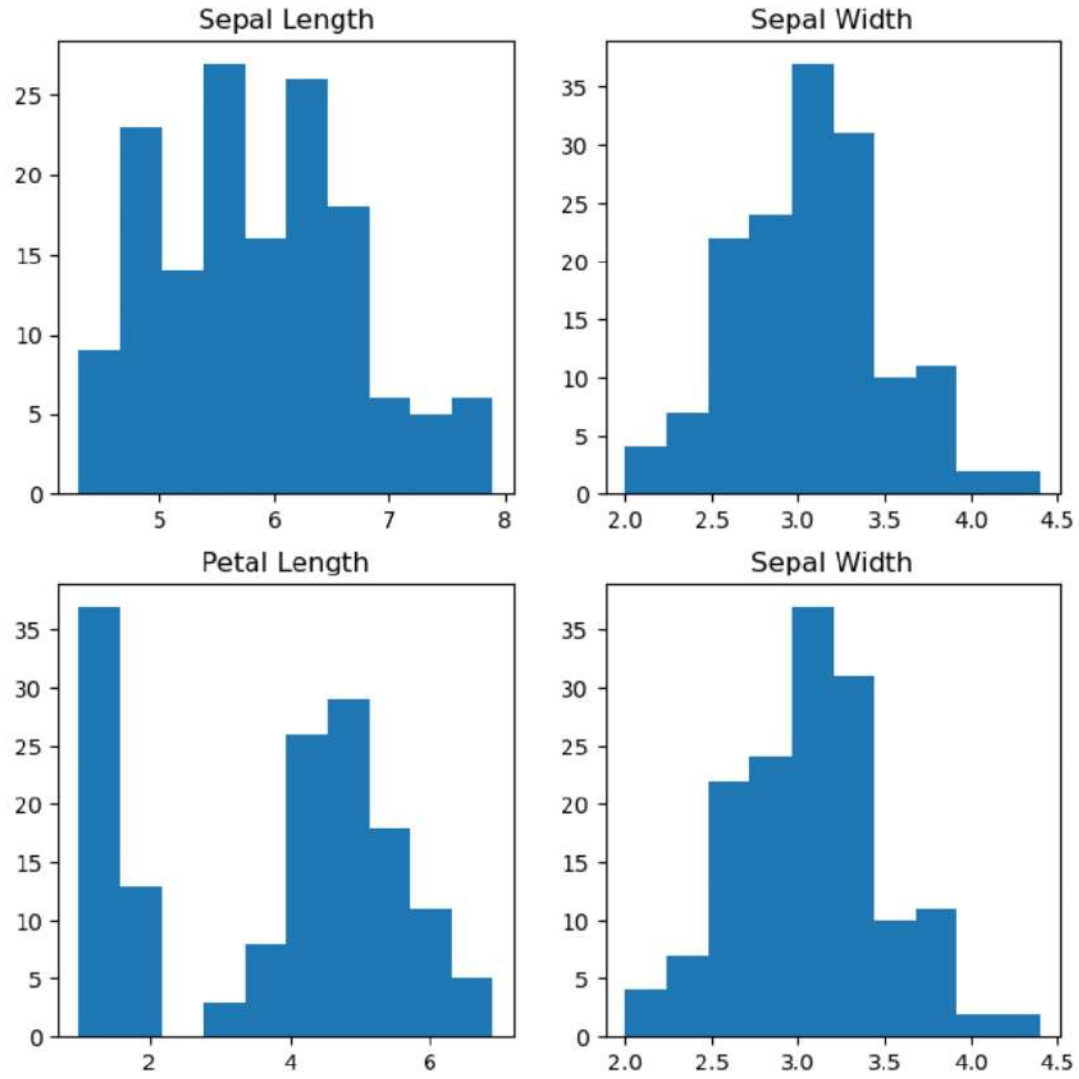
```
axes[0,0].hist(df['sepal.length'])

axes[0,1].set_title('Sepal Width')
axes[0,1].hist(df['sepal.width'])

axes[1,0].set_title('Petal Length')
axes[1,0].hist(df['petal.length'])

axes[1,1].set_title('Sepal Width')
axes[1,1].hist(df['sepal.width'])
```

```
[21]: (array([ 4.,  7., 22., 24., 37., 31., 10., 11.,  2.,  2.]),
       array([2.  , 2.24, 2.48, 2.72, 2.96, 3.2 , 3.44, 3.68, 3.92, 4.16, 4.4 ]),
       <BarContainer object of 10 artists>)
```

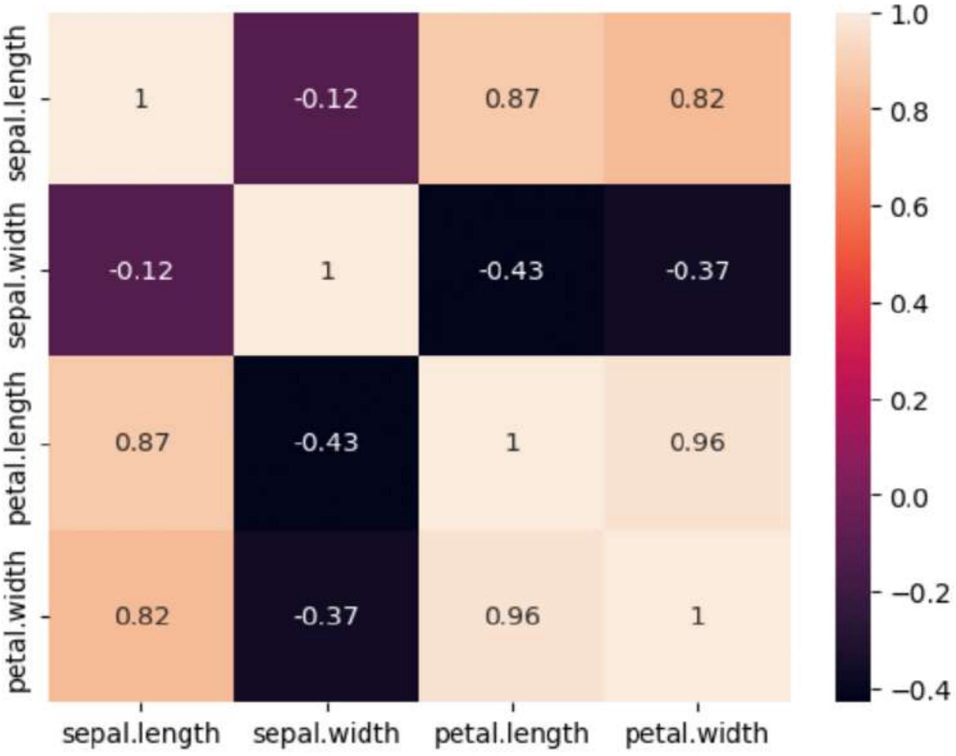


[22]:
```
df.corr(method='pearson', numeric_only=True)
```

[22]:
```
              sepal.length  sepal.width  petal.length  petal.width
sepal.length      1.000000    -0.117570      0.871754     0.817941
sepal.width      -0.117570     1.000000     -0.428440    -0.366126
petal.length      0.871754    -0.428440      1.000000     0.962865
petal.width       0.817941    -0.366126      0.962865     1.000000
```

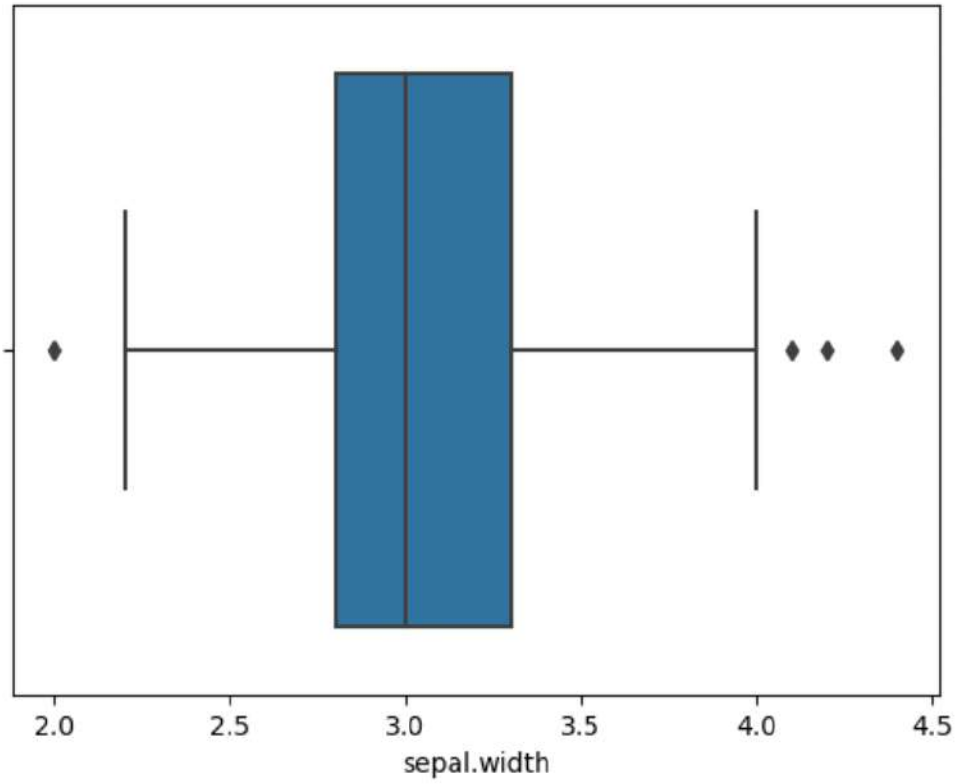
[23]:
```
sns.heatmap(df.corr(method='pearson', numeric_only=True), annot=True)
```

[23]:
```
<AxesSubplot: >
```



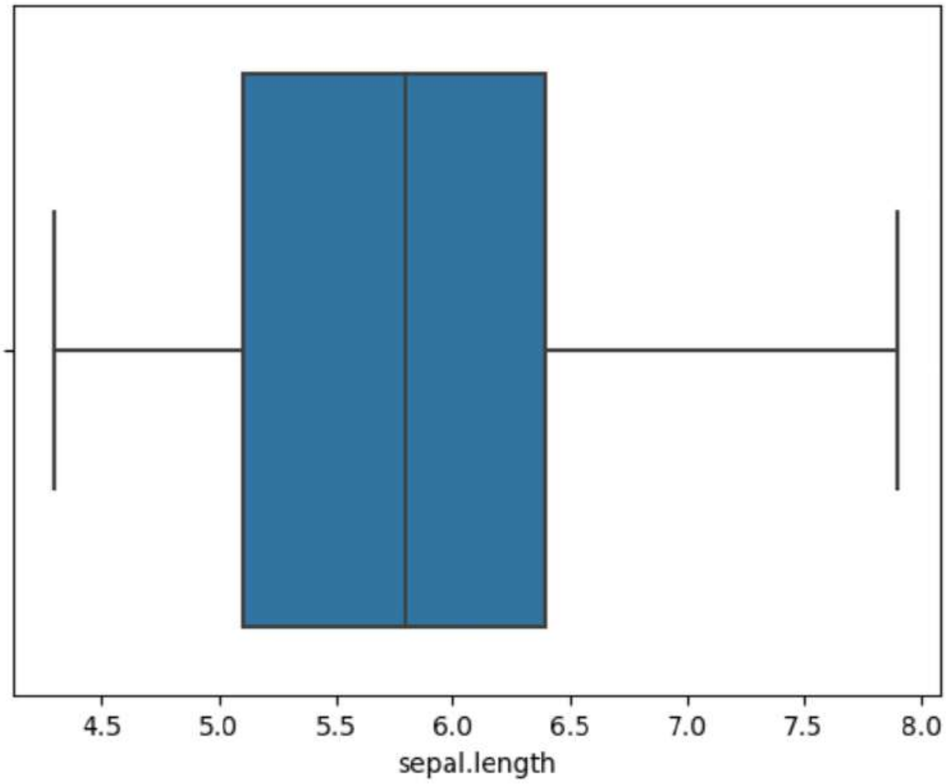
[24]:
```
sns.boxplot(x='sepal.width', data=df)
```

[24]:
```
<AxesSubplot: xlabel='sepal.width'>
```



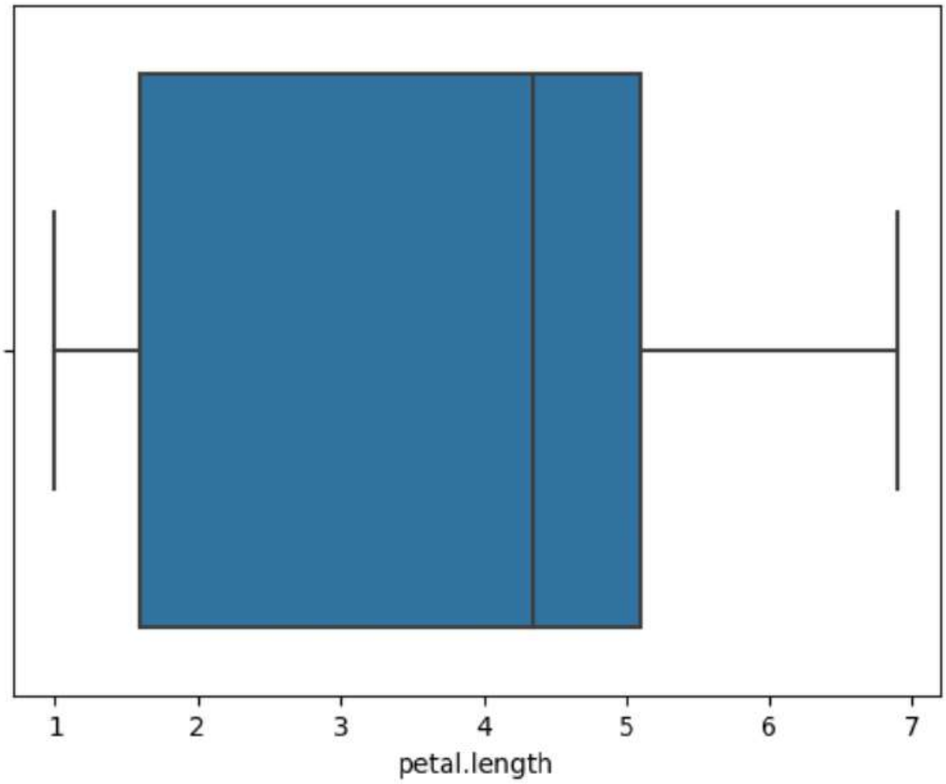
```
[25]: sns.boxplot(x='sepal.length', data=df)
```

```
[25]: <AxesSubplot: xlabel='sepal.length'>
```



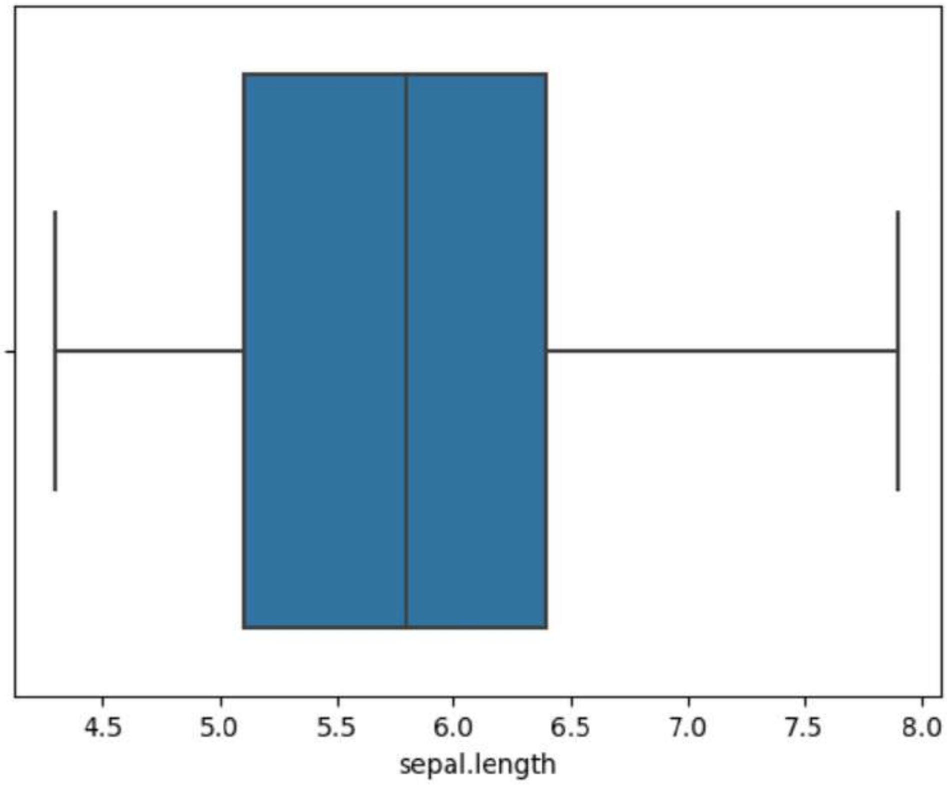
[26]:
```
sns.boxplot(x='petal.length', data=df)
```

[26]:
```
<AxesSubplot: xlabel='petal.length'>
```



```
[27]: sns.boxplot(x='sepal.length', data=df)
```

```
[27]: <AxesSubplot: xlabel='sepal.length'>
```



[29]:
```
df
```

[29]:
```
     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
..            …            …             …            …        …
145           6.7          3.0           5.2          2.3  Virginica
146           6.3          2.5           5.0          1.9  Virginica
147           6.5          3.0           5.2          2.0  Virginica
148           6.2          3.4           5.4          2.3  Virginica
149           5.9          3.0           5.1          1.8  Virginica

[150 rows x 5 columns]
```

[30]:
```
## calculando intervalo interquartil

Q1=np.percentile(df['sepal.width'],25, interpolation='midpoint')
```

```
Q3=np.percentile(df['sepal.width'],75, interpolation='midpoint')
```

```
/tmp/ipykernel_43689/1525703056.py:3: DeprecationWarning: the `interpolation=`
argument to percentile was renamed to `method=`, which has additional options.
Users of the modes 'nearest', 'lower', 'higher', or 'midpoint' are encouraged to
review the method they used. (Deprecated NumPy 1.22)
  Q1=np.percentile(df['sepal.width'],25, interpolation='midpoint')
/tmp/ipykernel_43689/1525703056.py:5: DeprecationWarning: the `interpolation=`
argument to percentile was renamed to `method=`, which has additional options.
Users of the modes 'nearest', 'lower', 'higher', or 'midpoint' are encouraged to
review the method they used. (Deprecated NumPy 1.22)
  Q3=np.percentile(df['sepal.width'],75, interpolation='midpoint')
```

[31]:
```
Q1
```

[31]:
```
2.8
```

[32]:
```
Q3
```

[32]:
```
3.3
```

[33]:
```
IRQ=Q3-Q1
```

[34]:
```
IRQ
```

[34]:
```
0.5
```

[36]:
```
superior = np.where(df['sepal.width']>=(Q3+1.5*IRQ))

inferior = np.where(df['sepal.width']<=(Q1-1.5*IRQ))
```

[37]:
```
superior
```

[37]:
```
(array([15, 32, 33]),)
```

[38]:
```
inferior
```

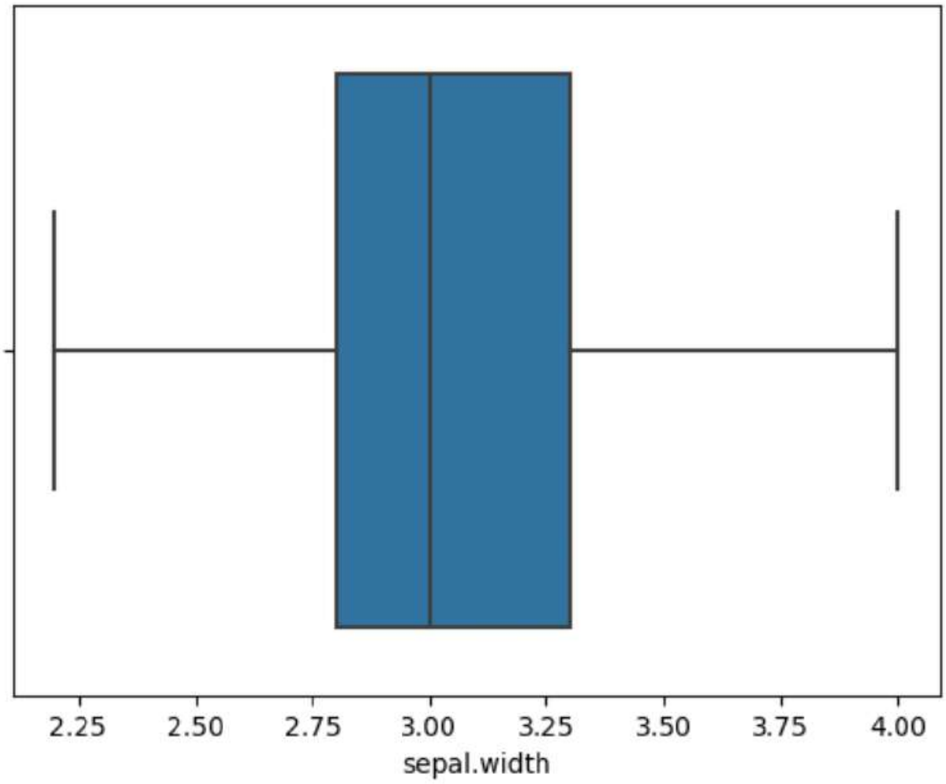
[38]:
```
(array([60]),)
```

[40]:
```
df.drop(superior[0], inplace=True)
```

[41]:
```
df.drop(inferior[0], inplace=True)
```

[42]:
```
sns.boxplot(x='sepal.width', data=df)
```

[42]:
```
<AxesSubplot: xlabel='sepal.width'>
```



```
[43]: df
```

```
[43]:      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
      ..            …            …             …            …         …
      145           6.7          3.0           5.2          2.3  Virginica
      146           6.3          2.5           5.0          1.9  Virginica
      147           6.5          3.0           5.2          2.0  Virginica
      148           6.2          3.4           5.4          2.3  Virginica
      149           5.9          3.0           5.1          1.8  Virginica

      [146 rows x 5 columns]
```

```
[45]: df['species'].unique()
```

```
[45]: array(['Setosa', 'Versicolor', 'Virginica'], dtype=object)
```

[46]:
```
from sklearn import preprocessing
```

[48]:
```
label_encoder=preprocessing.LabelEncoder()
```

[49]:
```
#label_encoder.fit

#label_encoder.transform

df['species']=label_encoder.fit_transform(df['species'])
```

[50]:
```
df['species'].unique()
```

[50]:
```
array([0, 1, 2])
```

[51]:
```
data = {'Employee id': [10, 20, 15, 25, 30, 45, 78, 56, 12, 7, 8, 57, 14, 27,␣
 ↪35],
        'Gender': ['M', 'F', 'F', 'M', 'F','M', 'F', 'F', 'M', 'F','M', 'F',␣
 ↪'F', 'M', 'F'],
        'Remarks': ['Good', 'Nice', 'Good', 'Great', 'Nice','Good', 'Nice',␣
 ↪'Good', 'Great', 'Nice','Good', 'Nice', 'Good', 'Great', 'Nice'],
        }
```

[52]:
```
data
```

[52]:
```
{'Employee id': [10, 20, 15, 25, 30, 45, 78, 56, 12, 7, 8, 57, 14, 27, 35],
 'Gender': ['M',
  'F',
  'F',
  'M',
  'F',
  'M',
  'F',
  'F',
  'M',
  'F',
  'M',
  'F',
  'F',
  'M',
  'F'],
 'Remarks': ['Good',
  'Nice',
  'Good',
  'Great',
  'Nice',
  'Good',
  'Nice',
 'Good',
 'Great',
 'Nice',
 'Good',
 'Nice',
 'Good',
 'Great',
 'Nice']}
```

[53]:
```
df=pd.DataFrame(data)
```

[54]:
```
df
```

[54]:
```
    Employee id Gender Remarks
0            10      M    Good
1            20      F    Nice
2            15      F    Good
3            25      M   Great
4            30      F    Nice
5            45      M    Good
6            78      F    Nice
7            56      F    Good
8            12      M   Great
9             7      F    Nice
10            8      M    Good
11           57      F    Nice
12           14      F    Good
13           27      M   Great
14           35      F    Nice
```

[55]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15 entries, 0 to 14
Data columns (total 3 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Employee id  15 non-null     int64
 1   Gender       15 non-null     object
 2   Remarks      15 non-null     object
dtypes: int64(1), object(2)
memory usage: 492.0+ bytes
```

[56]:
```
df['Gender'].unique()
```

[56]:
```
array(['M', 'F'], dtype=object)
```

[57]:
```
df['Remarks'].unique()
```

```
[57]: array(['Good', 'Nice', 'Great'], dtype=object)
```

```
[58]: df['Gender'].value_counts()
```

```
[58]: F    9
      M    6
      Name: Gender, dtype: int64
```

```
[59]: df['Remarks'].value_counts()
```

```
[59]: Good     6
      Nice     6
      Great    3
      Name: Remarks, dtype: int64
```

```
[60]: one_hot_encoded_data = pd.get_dummies(df, columns=['Remarks', 'Gender'])
```

```
[61]: one_hot_encoded_data
```

```
[61]:     Employee id  Remarks_Good  Remarks_Great  Remarks_Nice  Gender_F  Gender_M
      0            10             1              0             0         0         1
      1            20             0              0             1         1         0
      2            15             1              0             0         1         0
      3            25             0              1             0         0         1
      4            30             0              0             1         1         0
      5            45             1              0             0         0         1
      6            78             0              0             1         1         0
      7            56             1              0             0         1         0
      8            12             0              1             0         0         1
      9             7             0              0             1         1         0
      10            8             1              0             0         0         1
      11           57             0              0             1         1         0
      12           14             1              0             0         1         0
      13           27             0              1             0         0         1
      14           35             0              0             1         1         0
```

```
[ ]:
```