

```
In [85]: import pandas as pd
```

```
In [86]: df = pd.DataFrame()
```

```
In [87]: df = pd.read_csv('iris.csv')
```

```
In [88]: df
```

```
Out[88]:
```

	sepal.length	sepal.width	petal.length	petal.width	variety
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 × 5 columns

```
In [89]: df.head() # show first five
```

```
Out[89]:
```

	sepal.length	sepal.width	petal.length	petal.width	variety
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

```
In [90]: df.shape
```

```
Out[90]: (150, 5)
```

```
In [91]: 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   variety         150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

In [92]: `df.describe()`

```
Out[92]:
```

	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

In [93]: `df.mean()`

```
C:\Users\CSELAB1SYS10\AppData\Local\Temp\ipykernel_3416\3698961737.py:1: FutureWarning: The default value of numeric_only in DataFrame.mean is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.
df.mean()
```

```
Out[93]: sepal.length    5.843333
sepal.width      3.057333
petal.length     3.758000
petal.width      1.199333
dtype: float64
```

In [94]: `df.median()`

```
C:\Users\CSELAB1SYS10\AppData\Local\Temp\ipykernel_3416\530051474.py:1: FutureWarning: The default value of numeric_only in DataFrame.median is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.
df.median()
```

```
Out[94]: sepal.length    5.80  
        sepal.width     3.00  
        petal.length    4.35  
        petal.width     1.30  
        dtype: float64
```

```
In [95]: df.count()
```

```
Out[95]: sepal.length    150  
        sepal.width     150  
        petal.length    150  
        petal.width     150  
        variety        150  
        dtype: int64
```

```
In [96]: df.var()
```

C:\Users\CSELAB1SYS10\AppData\Local\Temp\ipykernel_3416\1568254755.py:1: FutureWarning: The default value of numeric_only in DataFrame.var is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.

```
df.var()
```

```
Out[96]: sepal.length    0.685694  
        sepal.width     0.189979  
        petal.length    3.116278  
        petal.width     0.581006  
        dtype: float64
```

```
In [97]: df.std()
```

C:\Users\CSELAB1SYS10\AppData\Local\Temp\ipykernel_3416\3390915376.py:1: FutureWarning: The default value of numeric_only in DataFrame.std is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.

```
df.std()
```

```
Out[97]: sepal.length    0.828066  
        sepal.width     0.435866  
        petal.length    1.765298  
        petal.width     0.762238  
        dtype: float64
```

```
In [98]: df.quantile(0,1)
```

C:\Users\CSELAB1SYS10\AppData\Local\Temp\ipykernel_3416\3399630545.py:1: FutureWarning: The default value of numeric_only in DataFrame.quantile is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.

```
df.quantile(0,1)
```

```
Out[98]: 0      0.2
         1      0.2
         2      0.2
         3      0.2
         4      0.2
         ...
        145     2.3
        146     1.9
        147     2.0
        148     2.3
        149     1.8
        Name: 0.0, Length: 150, dtype: float64
```

```
In [99]: df.quantile(0.95)
```

C:\Users\CSELAB1SYS10\AppData\Local\Temp\ipykernel_3416\1903517651.py:1: FutureWarning: The default value of numeric_only in DataFrame.quantile is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.

```
df.quantile(0.95)
```

```
Out[99]: sepal.length    7.255
         sepal.width     3.800
         petal.length    6.100
         petal.width     2.300
         Name: 0.95, dtype: float64
```

```
In [100]: df.quantile(0.75)
```

C:\Users\CSELAB1SYS10\AppData\Local\Temp\ipykernel_3416\3799946287.py:1: FutureWarning: The default value of numeric_only in DataFrame.quantile is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.

```
df.quantile(0.75)
```

```
Out[100]: sepal.length    6.4
          sepal.width     3.3
          petal.length    5.1
          petal.width     1.8
          Name: 0.75, dtype: float64
```

```
In [101]: df.quantile(0.25)
```

C:\Users\CSELAB1SYS10\AppData\Local\Temp\ipykernel_3416\3656653379.py:1: FutureWarning: The default value of numeric_only in DataFrame.quantile is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.

```
df.quantile(0.25)
```

```
Out[101]: sepal.length    5.1
          sepal.width     2.8
          petal.length    1.6
          petal.width     0.3
          Name: 0.25, dtype: float64
```

```
In [102]: df.quantile(0.25) - df.quantile(0.75)
```

C:\Users\CSELAB1SYS10\AppData\Local\Temp\ipykernel_3416\3938210929.py:1: FutureWarning: The default value of numeric_only in DataFrame.quantile is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.

```
df.quantile(0.25) - df.quantile(0.75)
```

```
Out[102]: sepal.length    -1.3
          sepal.width     -0.5
          petal.length    -3.5
          petal.width     -1.5
          dtype: float64
```

```
In [103... df.max()
```

```
Out[103]: sepal.length      7.9
          sepal.width       4.4
          petal.length      6.9
          petal.width       2.5
          variety           Virginica
          dtype: object
```

```
In [104... df.min()
```

```
Out[104]: sepal.length      4.3
          sepal.width       2.0
          petal.length      1.0
          petal.width       0.1
          variety           Setosa
          dtype: object
```

```
In [105... iris_group = df.groupby('variety')
```

```
In [106... print(iris_group)
```

```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x000000286C7746A0>
```

```
In [107... iris_group.mean()
```

```
Out[107]:
```

	sepal.length	sepal.width	petal.length	petal.width
variety				
Setosa	5.006	3.428	1.462	0.246
Versicolor	5.936	2.770	4.260	1.326
Virginica	6.588	2.974	5.552	2.026

```
In [108... iris_group.quantile(0.75)
```

```
Out[108]:
```

	sepal.length	sepal.width	petal.length	petal.width
variety				
Setosa	5.2	3.675	1.575	0.3
Versicolor	6.3	3.000	4.600	1.5
Virginica	6.9	3.175	5.875	2.3

```
In [109... iris_group.quantile(0.75) - iris_group.quantile(0.25)
```

Out[109]:

	sepal.length	sepal.width	petal.length	petal.width
variety				
Setosa	0.400	0.475	0.175	0.1
Versicolor	0.700	0.475	0.600	0.3
Virginica	0.675	0.375	0.775	0.5

```
In [110... iris_group.describe()
```

Out[110]:

	sepal.length								sepal.width			...	petal.length			...
	count	mean	std	min	25%	50%	75%	max	count	mean	std	...	75%	max	count	
variety																
Setosa	50.0	5.006	0.352490	4.3	4.800	5.0	5.2	5.8	50.0	3.428	0.762223	...	1.575	1.9	50.0	
Versicolor	50.0	5.936	0.516171	4.9	5.600	5.9	6.3	7.0	50.0	2.770	0.963634	...	4.600	5.1	50.0	
Virginica	50.0	6.588	0.635880	4.9	6.225	6.5	6.9	7.9	50.0	2.974	0.959302	...	5.875	6.9	50.0	

3 rows × 32 columns



```
In [111... #pima
```

```
In [112... df = pd.read_csv('pima-indians-diabetes.csv')
```

```
In [113... df
```

Out[113]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	6	148	72	35	0	33.6	0.627	51
1	1	85	66	29	0	26.6	0.351	33
2	8	183	64	0	0	23.3	0.672	33
3	1	89	66	23	94	28.1	0.167	24
4	0	137	40	35	168	43.1	2.288	33
...
763	10	101	76	48	180	32.9	0.171	63
764	2	122	70	27	0	36.8	0.340	24
765	5	121	72	23	112	26.2	0.245	33
766	1	126	60	0	0	30.1	0.349	44
767	1	93	70	31	0	30.4	0.315	24

768 rows × 9 columns

In [114]: `df.head()`

Out[114]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	6	148	72	35	0	33.6	0.627	50
1	1	85	66	29	0	26.6	0.351	31
2	8	183	64	0	0	23.3	0.672	32
3	1	89	66	23	94	28.1	0.167	21
4	0	137	40	35	168	43.1	2.288	33

In [115]: `df.shape`

Out[115]: (768, 9)

In [116]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Pregnancies           768 non-null    int64
1   Glucose               768 non-null    int64
2   BloodPressure         768 non-null    int64
3   SkinThickness         768 non-null    int64
4   Insulin               768 non-null    int64
5   BMI                   768 non-null    float64
6   DiabetesPedigreeFunction 768 non-null    float64
7   Age                   768 non-null    int64
8   Class                 768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

In [117]: `df.describe()`

Out[117]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	

```
In [118... df.count()
```

```
Out[118]: Pregnancies      768
          Glucose          768
          BloodPressure    768
          SkinThickness    768
          Insulin          768
          BMI              768
          DiabetesPedigreeFunction 768
          Age              768
          Class            768
          dtype: int64
```

```
In [119... df.mean()
```

```
Out[119]: Pregnancies      3.845052
          Glucose          120.894531
          BloodPressure    69.105469
          SkinThickness    20.536458
          Insulin          79.799479
          BMI              31.992578
          DiabetesPedigreeFunction 0.471876
          Age              33.240885
          Class            0.348958
          dtype: float64
```

```
In [120... df.median()
```

```
Out[120]: Pregnancies      3.0000
          Glucose          117.0000
          BloodPressure    72.0000
          SkinThickness    23.0000
          Insulin          30.5000
          BMI              32.0000
          DiabetesPedigreeFunction 0.3725
          Age              29.0000
          Class            0.0000
          dtype: float64
```

```
In [121... df.var()
```

```
Out[121]: Pregnancies      11.354056
          Glucose          1022.248314
          BloodPressure    374.647271
          SkinThickness    254.473245
          Insulin          13281.180078
          BMI              62.159984
          DiabetesPedigreeFunction 0.109779
          Age              138.303046
          Class            0.227483
          dtype: float64
```

```
In [122... df.std()
```



```
Out[122]: Pregnancies      3.369578
          Glucose          31.972618
          BloodPressure    19.355807
          SkinThickness    15.952218
          Insulin          115.244002
          BMI              7.884160
          DiabetesPedigreeFunction 0.331329
          Age              11.760232
          Class            0.476951
          dtype: float64
```

```
In [123... df.quantile(0.1)
```

```
Out[123]: Pregnancies      0.000
          Glucose          85.000
          BloodPressure    54.000
          SkinThickness    0.000
          Insulin          0.000
          BMI              23.600
          DiabetesPedigreeFunction 0.165
          Age              22.000
          Class            0.000
          Name: 0.1, dtype: float64
```

```
In [124... df.quantile(0.25)
```

```
Out[124]: Pregnancies      1.00000
          Glucose          99.00000
          BloodPressure    62.00000
          SkinThickness    0.00000
          Insulin          0.00000
          BMI              27.30000
          DiabetesPedigreeFunction 0.24375
          Age              24.00000
          Class            0.00000
          Name: 0.25, dtype: float64
```

```
In [125... df.quantile(0.75)
```

```
Out[125]: Pregnancies      6.00000
          Glucose          140.25000
          BloodPressure    80.00000
          SkinThickness    32.00000
          Insulin          127.25000
          BMI              36.60000
          DiabetesPedigreeFunction 0.62625
          Age              41.00000
          Class            1.00000
          Name: 0.75, dtype: float64
```

```
In [126... df.quantile(0.75)-df.quantile(0.25)
```

```
Out[126]: Pregnancies      5.0000
          Glucose         41.2500
          BloodPressure    18.0000
          SkinThickness    32.0000
          Insulin          127.2500
          BMI              9.3000
          DiabetesPedigreeFunction 0.3825
          Age              17.0000
          Class            1.0000
          dtype: float64
```

```
In [127... df.max()
```

```
Out[127]: Pregnancies      17.00
          Glucose         199.00
          BloodPressure    122.00
          SkinThickness    99.00
          Insulin          846.00
          BMI              67.10
          DiabetesPedigreeFunction 2.42
          Age              81.00
          Class            1.00
          dtype: float64
```

```
In [128... df.min()
```

```
Out[128]: Pregnancies      0.000
          Glucose          0.000
          BloodPressure     0.000
          SkinThickness     0.000
          Insulin           0.000
          BMI               0.000
          DiabetesPedigreeFunction 0.078
          Age               21.000
          Class             0.000
          dtype: float64
```

```
In [129... grp = df.groupby('Class')
```

```
In [130... grp.mean()
```

```
Out[130]:
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction
Class							
0	3.298000	109.980000	68.184000	19.664000	68.792000	30.304200	
1	4.865672	141.257463	70.824627	22.164179	100.335821	35.142537	

```
In [131... grp.quantile(0.25)
```

Out[131]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Class
0	1.00	93.0	62.0	0.0	0.0	25.4	0.22975	2
1	1.75	119.0	66.0	0.0	0.0	30.8	0.26250	2

In [132...]

grp.quantile(0.75) - grp.quantile(0.25)

Out[132]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Class
0	4.00	32.0	16.0	31.0	105.00	9.900	0.3320	
1	6.25	48.0	16.0	36.0	167.25	7.975	0.4655	

In [133...]

grp.describe()

Out[133]:

	Pregnancies								Glucose		...	DiabetesPedigr
	count	mean	std	min	25%	50%	75%	max	count	mean	...	75%
Class												
0	500.0	3.298000	3.017185	0.0	1.00	2.0	5.0	13.0	500.0	109.980000	...	0.56175
1	268.0	4.865672	3.741239	0.0	1.75	4.0	8.0	17.0	268.0	141.257463	...	0.72800

2 rows × 64 columns

In [134...]

df.value_counts("Class")

Out[134]:

Class
0 500
1 268
dtype: int64

In [135...]

df.skew()

Out[135]:

Pregnancies 0.901674
Glucose 0.173754
BloodPressure -1.843608
SkinThickness 0.109372
Insulin 2.272251
BMI -0.428982
DiabetesPedigreeFunction 1.919911
Age 1.129597
Class 0.635017
dtype: float64

In [136...]

df.kurt()

```
Out[136]: Pregnancies      0.159220
          Glucose          0.640780
          BloodPressure    5.180157
          SkinThickness    -0.520072
          Insulin          7.214260
          BMI              3.290443
          DiabetesPedigreeFunction  5.594954
          Age              0.643159
          Class            -1.600930
          dtype: float64
```

```
In [137]: df.Age
```

```
Out[137]: 0      50
          1      31
          2      32
          3      21
          4      33
          ..
          763    63
          764    27
          765    30
          766    47
          767    23
          Name: Age, Length: 768, dtype: int64
```

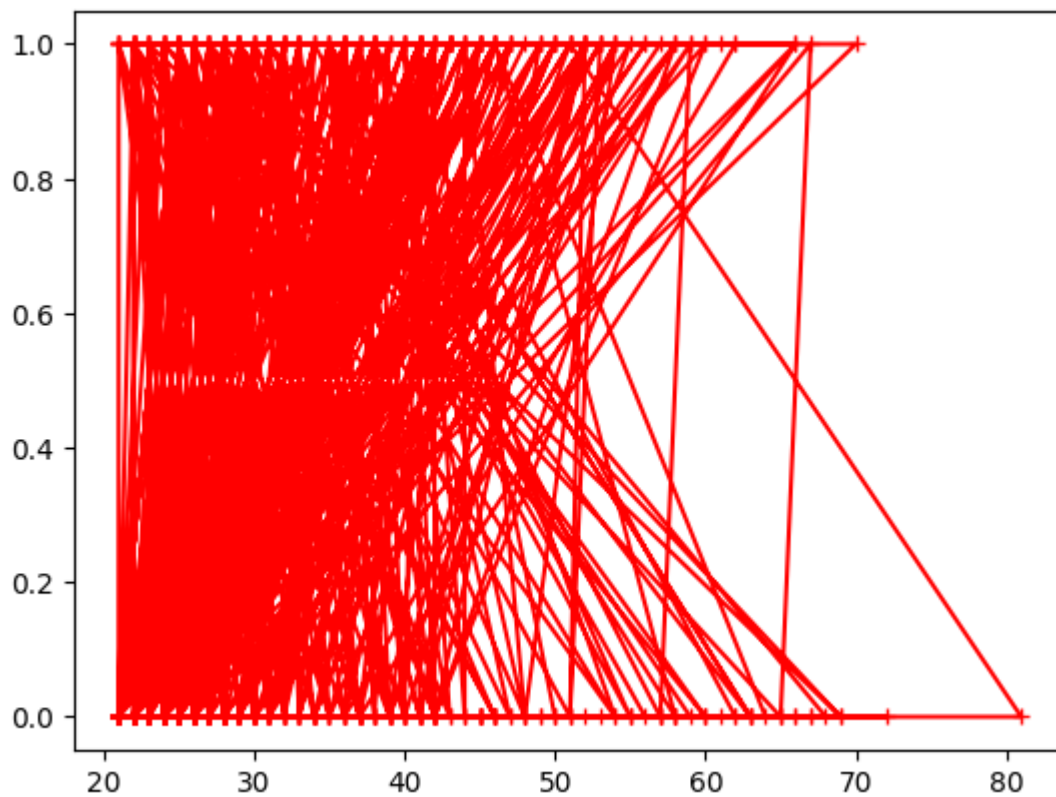
```
In [138]: df.Class
```

```
Out[138]: 0      1
          1      0
          2      1
          3      0
          4      1
          ..
          763    0
          764    0
          765    0
          766    1
          767    0
          Name: Class, Length: 768, dtype: int64
```

```
In [139]: from matplotlib import pyplot as plt
```

```
In [177]: plt.plot(df.Age, df.Class, marker='+', color='red')
```

```
Out[177]: [matplotlib.lines.Line2D at 0x286189d0c0]
```



```
In [141... from sklearn.model_selection import train_test_split
```

```
In [165... xtrain,xtest,ytrain,ytest = train_test_split(  
    df[['Age', 'Pregnancies']],  
    df.Class,  
    test_size=0.3,  
)
```

```
In [166... xtest
```

Out[166]:

	Age	Pregnancies
423	21	2
166	22	3
175	36	8
761	43	9
636	48	5
...
256	30	3
40	26	3
434	36	1
462	39	8
467	25	0

231 rows × 2 columns

In [167... xtrain

Out[167]:

	Age	Pregnancies
15	32	7
16	31	0
413	21	1
703	41	2
126	30	3
...
682	22	0
732	24	2
465	21	0
153	23	1
723	42	5

537 rows × 2 columns

In [148... `from sklearn.linear_model import LogisticRegression`In [170... `model = LogisticRegression()`In [171... `model.fit(xtrain,ytrain)`

Out[171]: **LogisticRegression**
 LogisticRegression()

In [172]: `x = model.predict(xtest)`

In [173]: `model.score(xtest,ytest)`

Out[173]: 0.6753246753246753

In [176]: `model.predict([[3,50]])`

C:\Users\CSELAB1SYS10\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LogisticRegression was fitted with feature names
 warnings.warn(

Out[176]: array([1], dtype=int64)

In [164]: `df[['Pregnancies','Class']]`

Out[164]:

	Pregnancies	Class
0	6	1
1	1	0
2	8	1
3	1	0
4	0	1
...
763	10	0
764	2	0
765	5	0
766	1	1
767	1	0

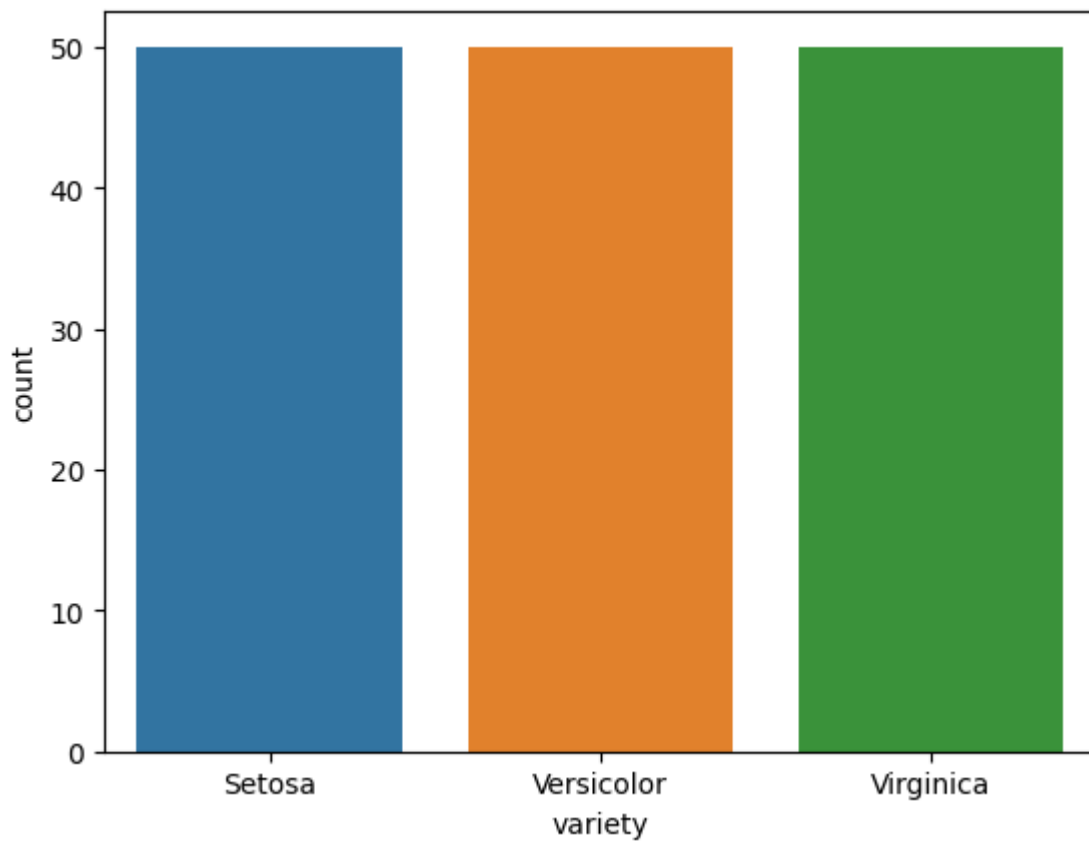
768 rows × 2 columns

In [2]: `import pandas as pd`
`import seaborn as sns`
`import matplotlib.pyplot as plt`

In [6]: `df = pd.read_csv('iris.csv')`

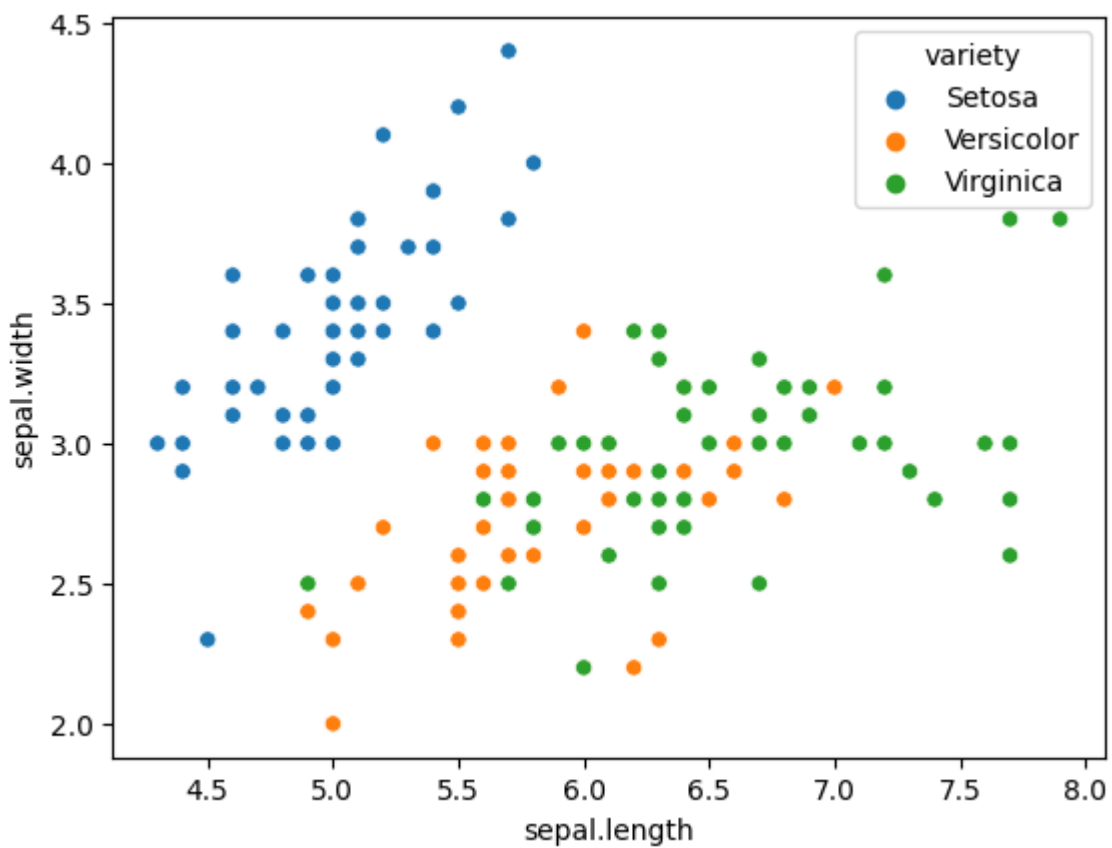
In [5]: `sns.countplot(x='variety', data= df) #7a`

Out[5]: <AxesSubplot: xlabel='variety', ylabel='count'>



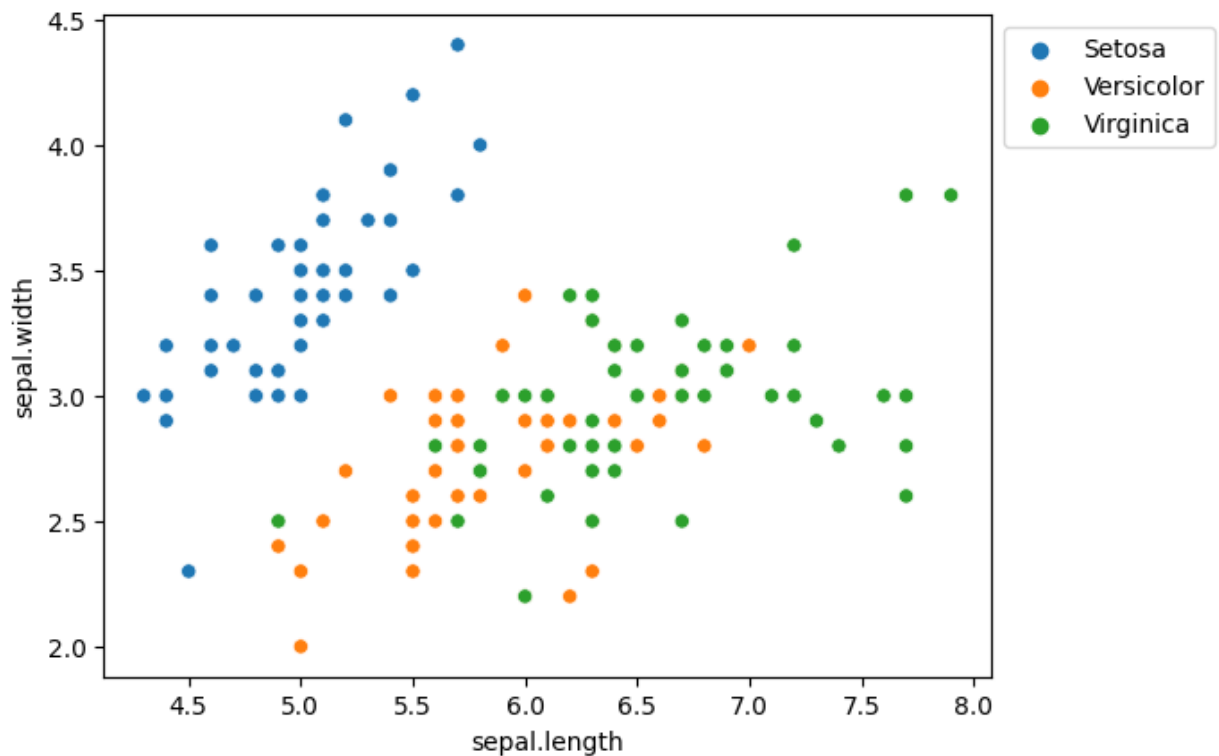
```
In [11]: sns.scatterplot(x='sepal.length', y='sepal.width', hue='variety', data=df)
```

```
Out[11]: <AxesSubplot: xlabel='sepal.length', ylabel='sepal.width'>
```



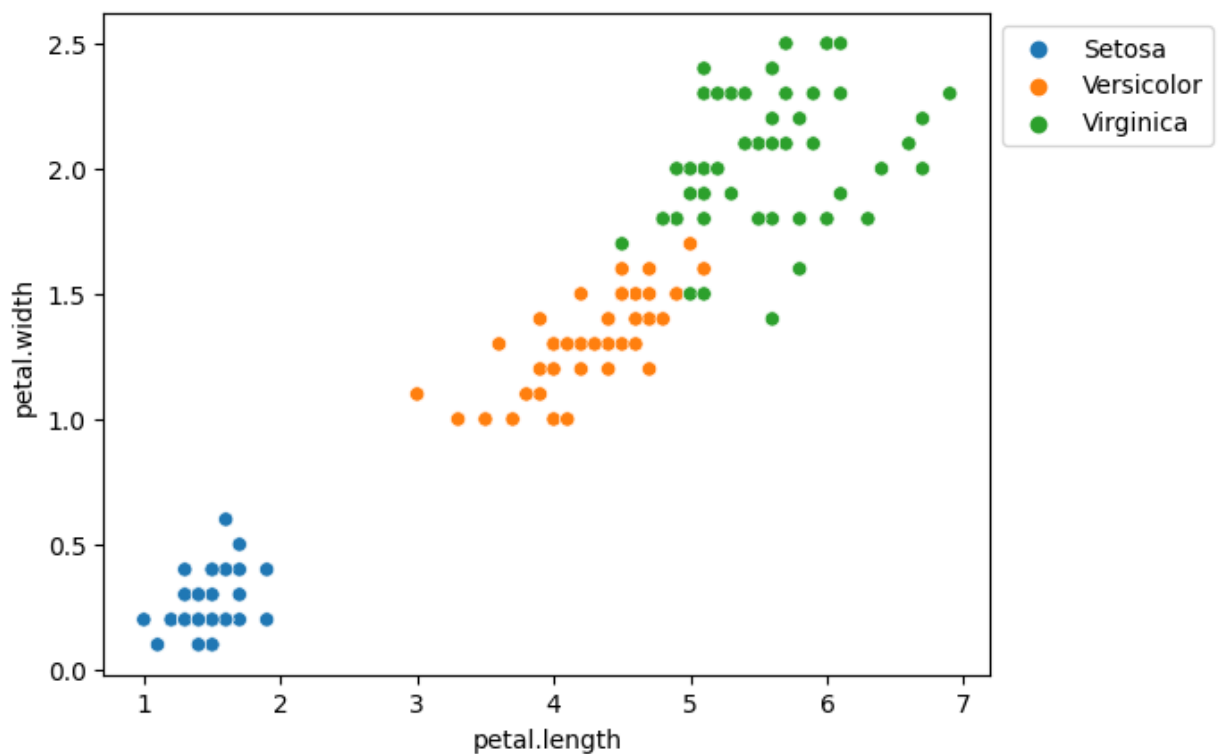

```
In [18]: sns.scatterplot(x='sepal.length', y='sepal.width', hue='variety',data=df) #7b  
plt.legend(bbox_to_anchor=(1,1), loc=2)
```

Out[18]: <matplotlib.legend.Legend at 0xcffed8da80>



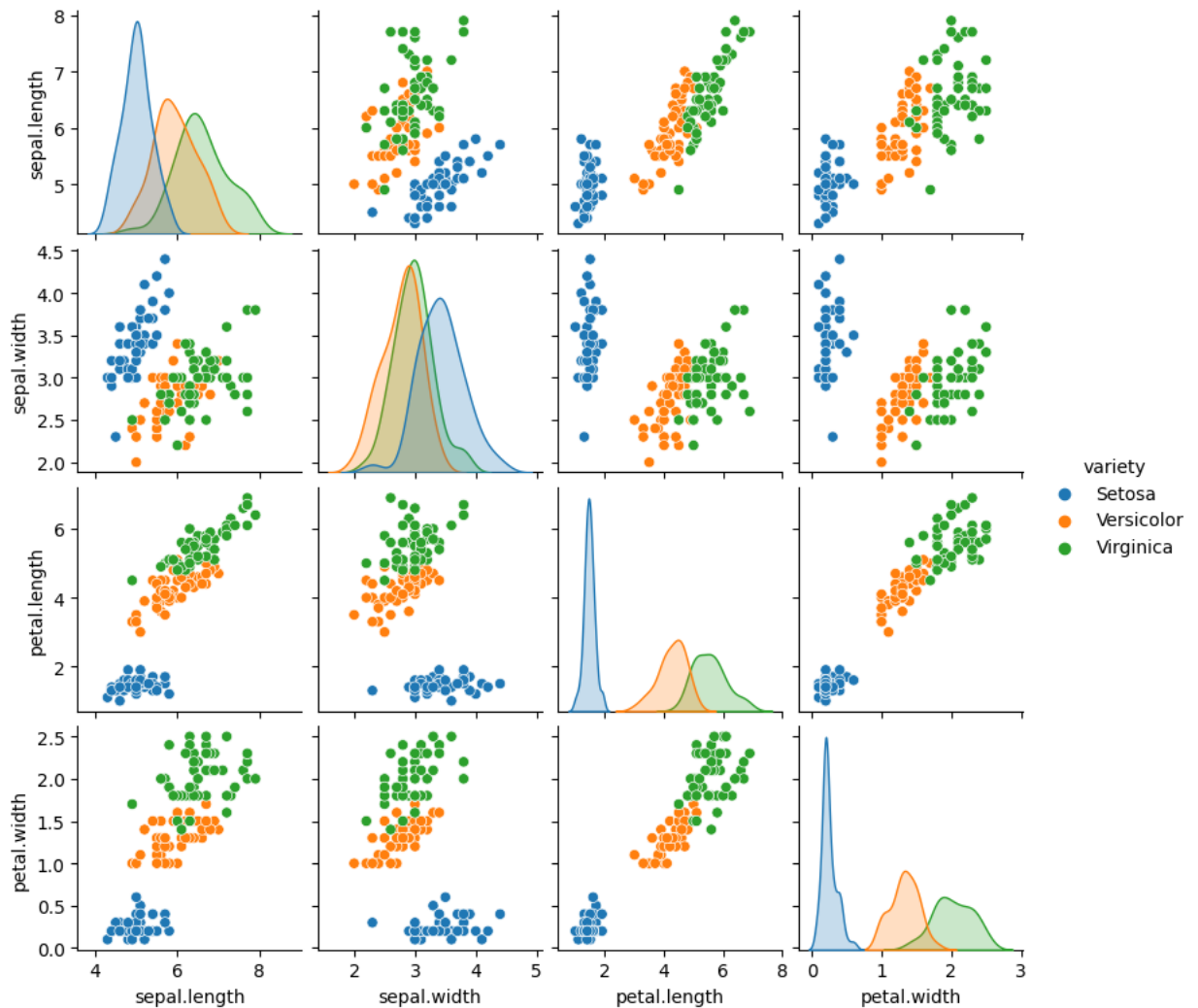
```
In [21]: sns.scatterplot(x='petal.length', y='petal.width', hue='variety',data=df)  
plt.legend(bbox_to_anchor=(1,1), loc=2)
```

Out[21]: <matplotlib.legend.Legend at 0xcf822841c0>



```
In [23]: sns.pairplot(df,hue='variety',height=2)
```

Out[23]: <seaborn.axisgrid.PairGrid at 0xcf84551b40>



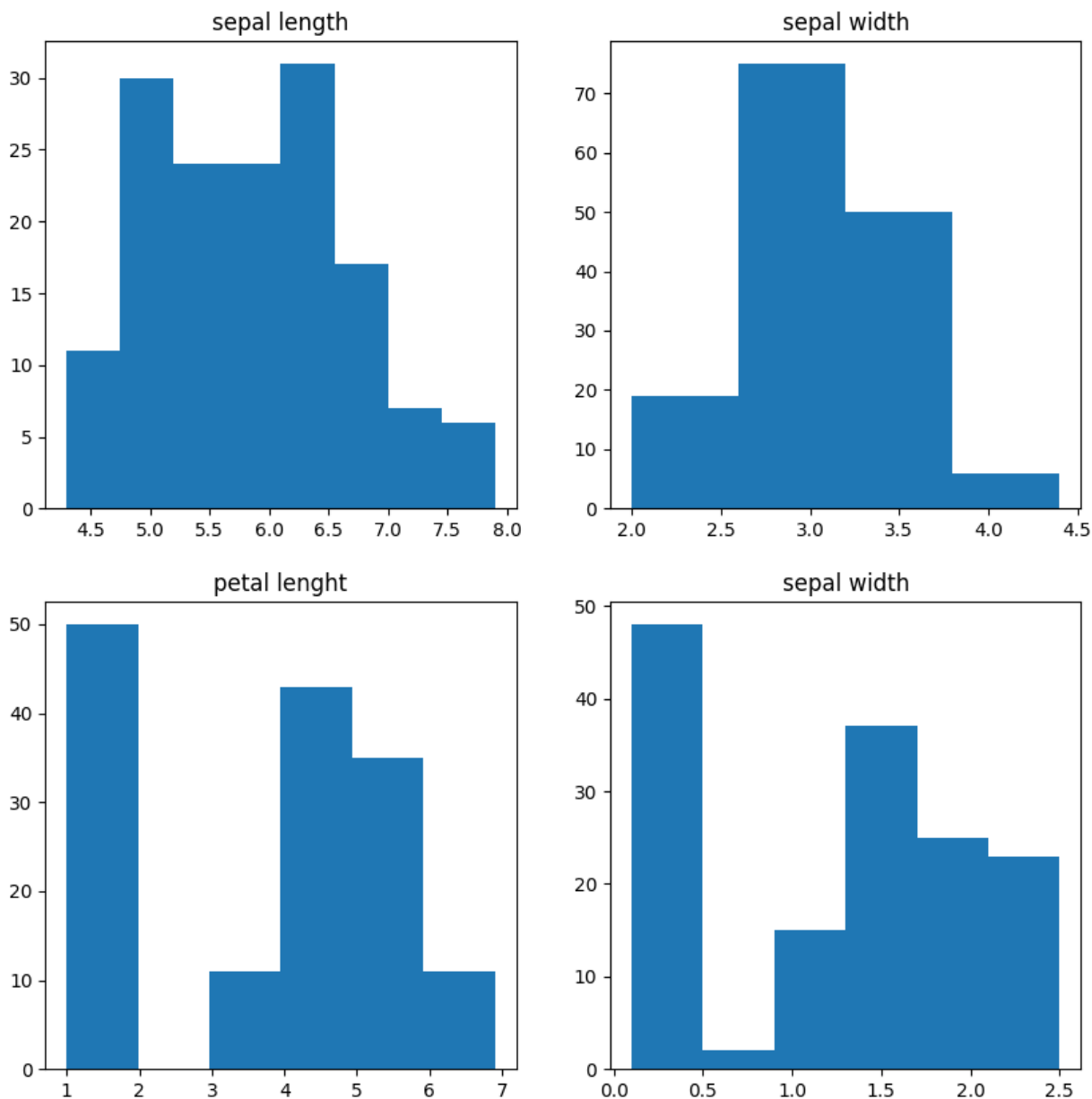
```
In [46]: fig, axes = plt.subplots(2, 2, figsize=(10, 10))
axes[0, 0].set_title("sepal length")
axes[0, 0].hist(df['sepal.length'], bins=8)

axes[0, 1].set_title("sepal width")
axes[0, 1].hist(df['sepal.width'], bins=4)

axes[1, 0].set_title("petal length")
axes[1, 0].hist(df['petal.length'], bins=6)

axes[1, 1].set_title("sepal width")
axes[1, 1].hist(df['petal.width'], bins=6)
```

Out[46]: (array([48., 2., 15., 37., 25., 23.]),
array([0.1, 0.5, 0.9, 1.3, 1.7, 2.1, 2.5]),
<BarContainer object of 6 artists>)



In [37]: `plt.subplots??`

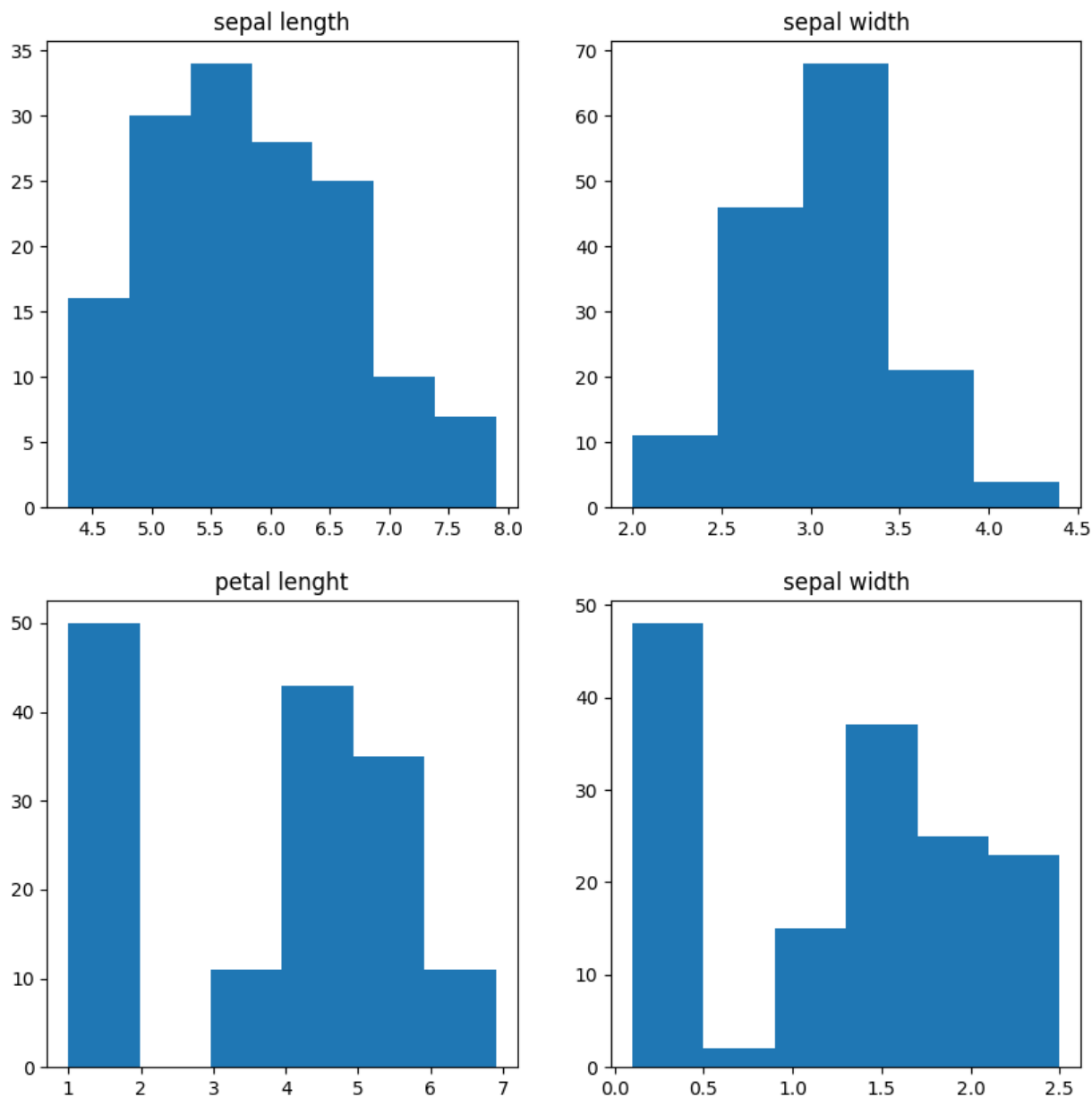
```
In [52]: fig, axes = plt.subplots(2, 2, figsize=(10, 10))
axes[0, 0].set_title("sepal length")
axes[0, 0].hist(df['sepal.length'], bins=7)

axes[0, 1].set_title("sepal width")
axes[0, 1].hist(df['sepal.width'], bins=5)

axes[1, 0].set_title("petal length")
axes[1, 0].hist(df['petal.length'], bins=6)

axes[1, 1].set_title("sepal width")
axes[1, 1].hist(df['petal.width'], bins=6)
```

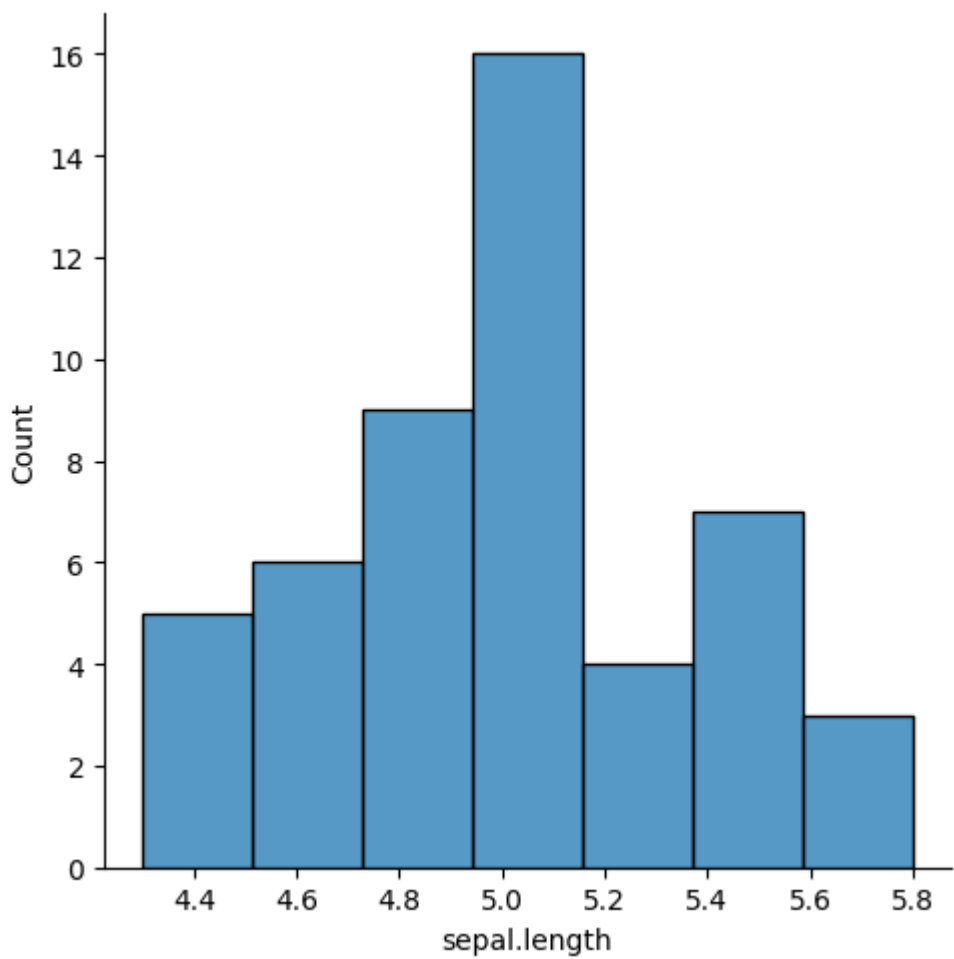
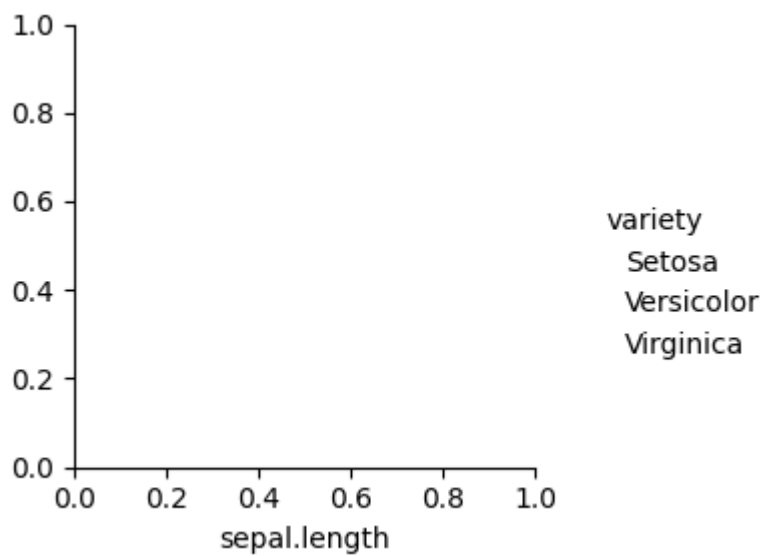
```
Out[52]: (array([48.,  2., 15., 37., 25., 23.]),
          array([0.1, 0.5, 0.9, 1.3, 1.7, 2.1, 2.5]),
          <BarContainer object of 6 artists>)
```

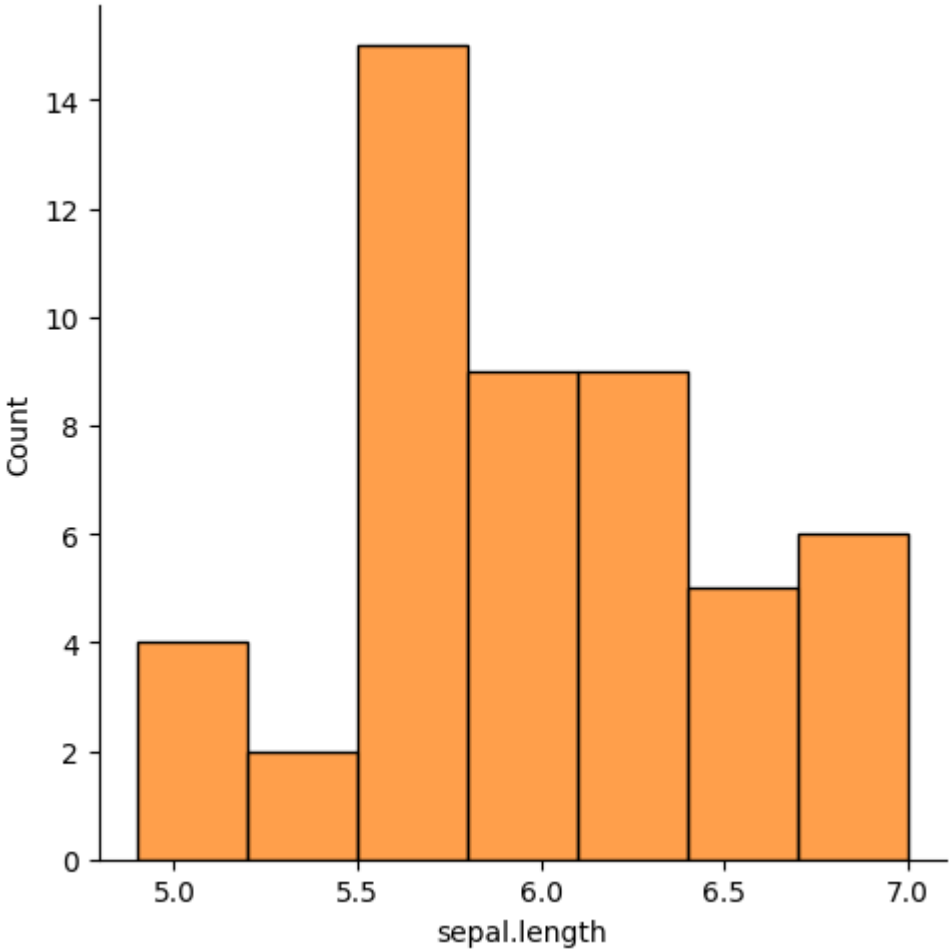


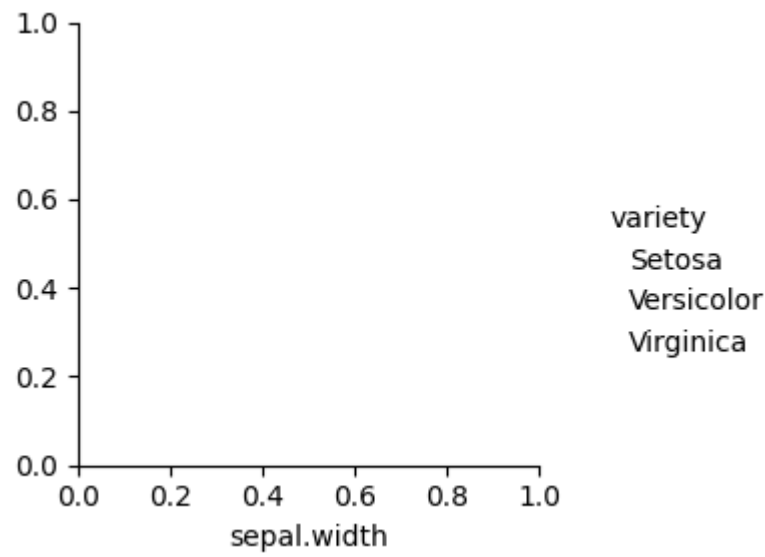
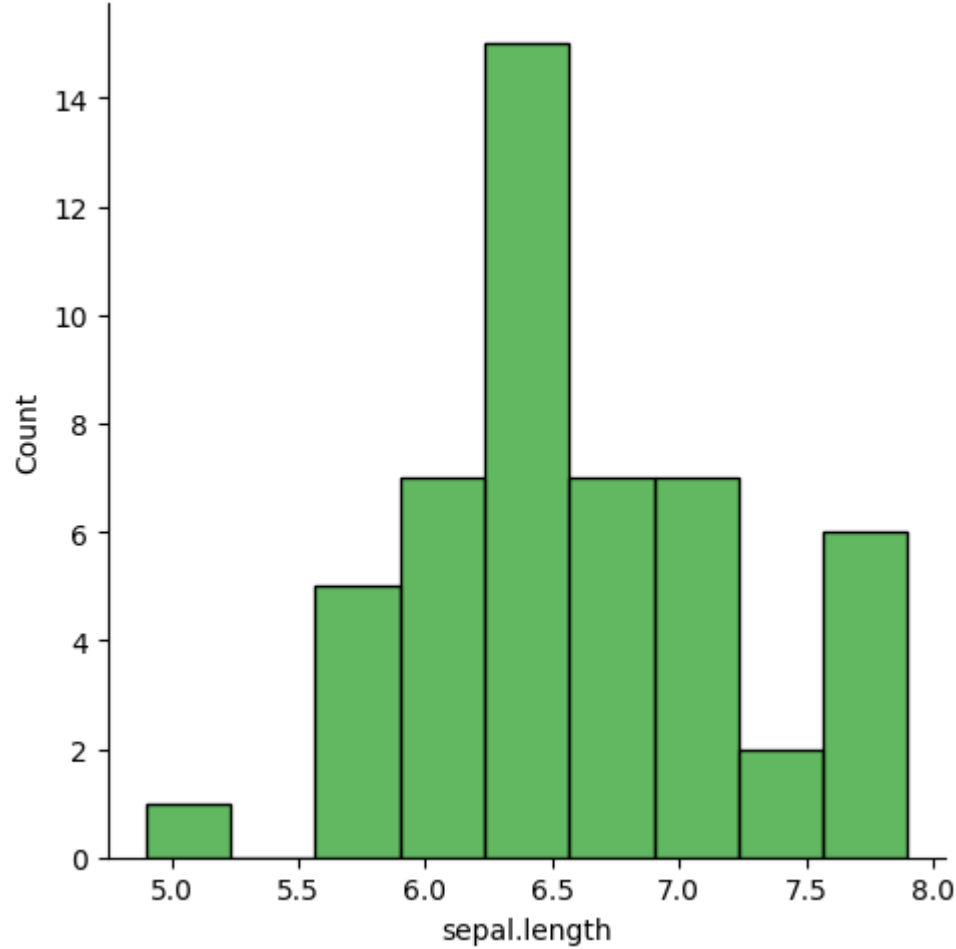
```
In [58]: plot = sns.FacetGrid(df, hue="variety")
plot.map(sns.displot, 'sepal.length').add_legend()
plot = sns.FacetGrid(df, hue="variety")
plot.map(sns.displot, 'sepal.width').add_legend()
plot = sns.FacetGrid(df, hue="variety")
plot.map(sns.displot, 'petal.length').add_legend()
plot = sns.FacetGrid(df, hue="variety")
plot.map(sns.displot, 'petal.width').add_legend()
```

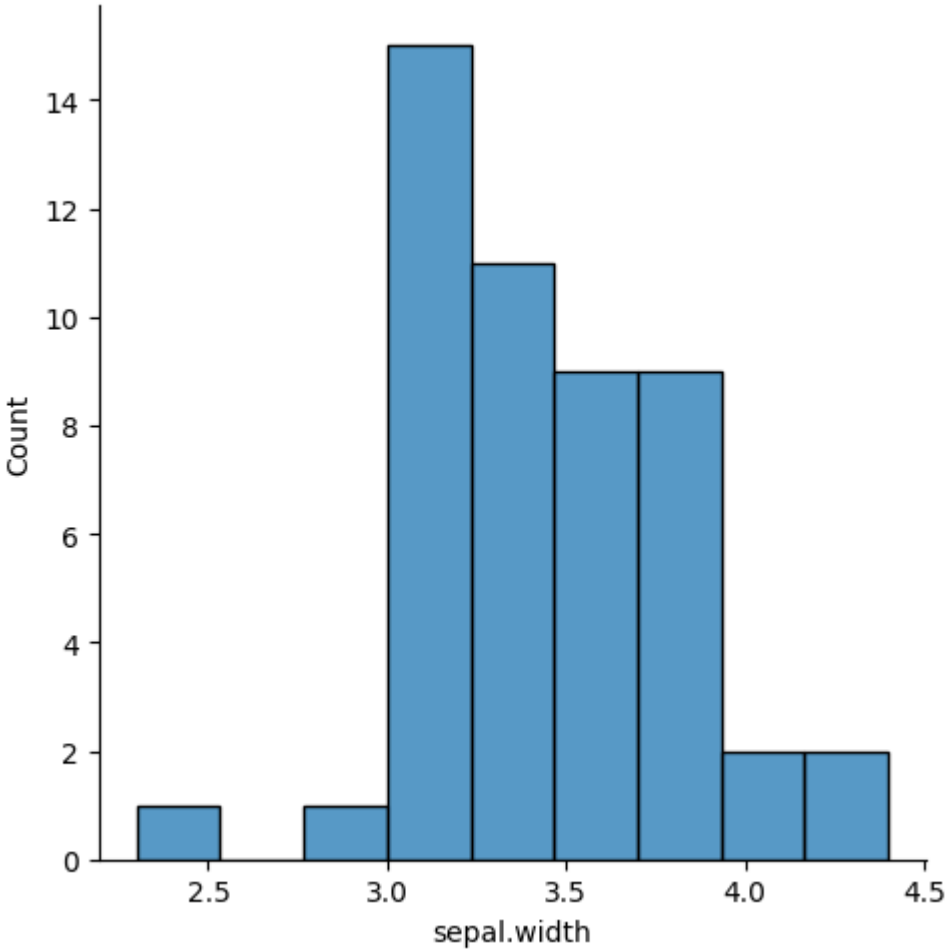
```
C:\Users\CSELAB1SYS10\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\distributions.py:2142: UserWarning: `displot` is a figure-level function and does not accept the ax= parameter. You may wish to try histplot.
  warnings.warn(msg, UserWarning)
C:\Users\CSELAB1SYS10\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\distributions.py:2142: UserWarning: `displot` is a figure-level function and does not accept the ax= parameter. You may wish to try histplot.
  warnings.warn(msg, UserWarning)
C:\Users\CSELAB1SYS10\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\distributions.py:2142: UserWarning: `displot` is a figure-level function and does not accept the ax= parameter. You may wish to try histplot.
  warnings.warn(msg, UserWarning)
C:\Users\CSELAB1SYS10\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\distributions.py:2142: UserWarning: `displot` is a figure-level function and does not accept the ax= parameter. You may wish to try histplot.
  warnings.warn(msg, UserWarning)
C:\Users\CSELAB1SYS10\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\distributions.py:2142: UserWarning: `displot` is a figure-level function and does not accept the ax= parameter. You may wish to try histplot.
  warnings.warn(msg, UserWarning)
C:\Users\CSELAB1SYS10\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\distributions.py:2142: UserWarning: `displot` is a figure-level function and does not accept the ax= parameter. You may wish to try histplot.
  warnings.warn(msg, UserWarning)
C:\Users\CSELAB1SYS10\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\distributions.py:2142: UserWarning: `displot` is a figure-level function and does not accept the ax= parameter. You may wish to try histplot.
  warnings.warn(msg, UserWarning)
C:\Users\CSELAB1SYS10\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\distributions.py:2142: UserWarning: `displot` is a figure-level function and does not accept the ax= parameter. You may wish to try histplot.
  warnings.warn(msg, UserWarning)
C:\Users\CSELAB1SYS10\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\distributions.py:2142: UserWarning: `displot` is a figure-level function and does not accept the ax= parameter. You may wish to try histplot.
  warnings.warn(msg, UserWarning)
C:\Users\CSELAB1SYS10\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\distributions.py:2142: UserWarning: `displot` is a figure-level function and does not accept the ax= parameter. You may wish to try histplot.
  warnings.warn(msg, UserWarning)
C:\Users\CSELAB1SYS10\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\distributions.py:2142: UserWarning: `displot` is a figure-level function and does not accept the ax= parameter. You may wish to try histplot.
  warnings.warn(msg, UserWarning)
C:\Users\CSELAB1SYS10\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\distributions.py:2142: UserWarning: `displot` is a figure-level function and does not accept the ax= parameter. You may wish to try histplot.
  warnings.warn(msg, UserWarning)
C:\Users\CSELAB1SYS10\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\distributions.py:2142: UserWarning: `displot` is a figure-level function and does not accept the ax= parameter. You may wish to try histplot.
  warnings.warn(msg, UserWarning)
```

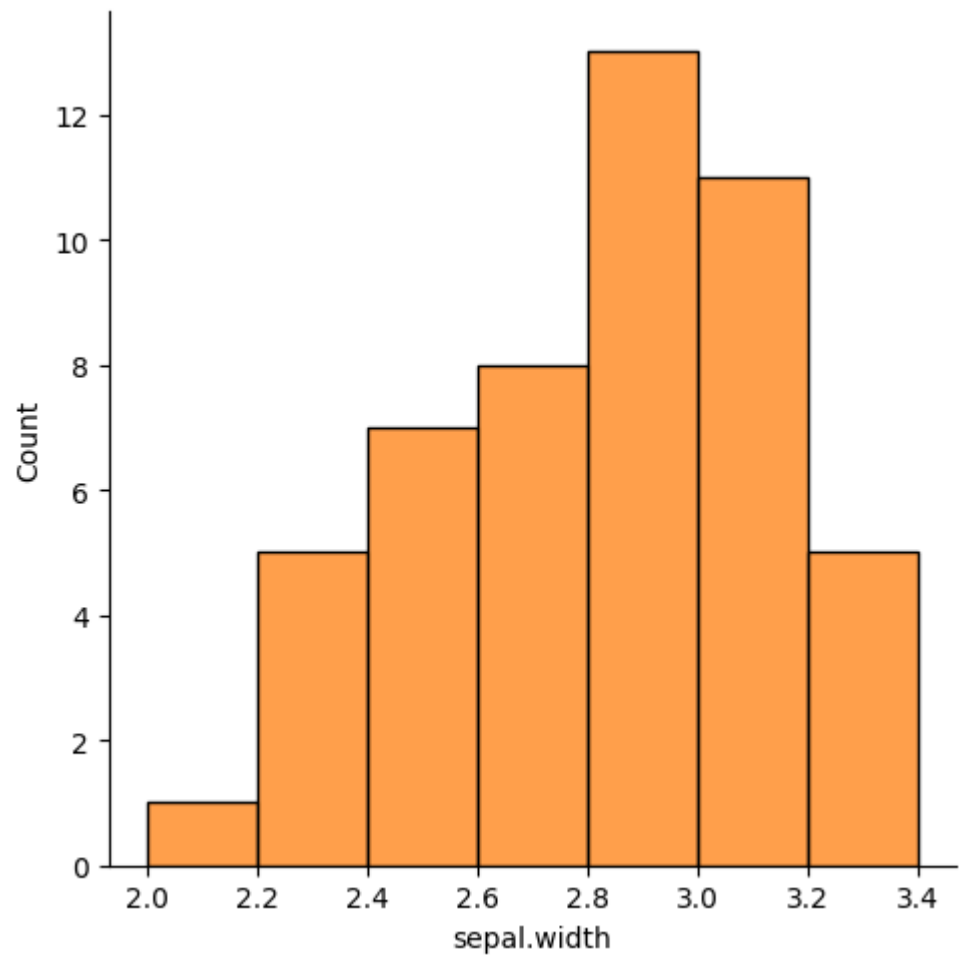
```
Out[58]: <seaborn.axisgrid.FacetGrid at 0xc92dc5990>
```

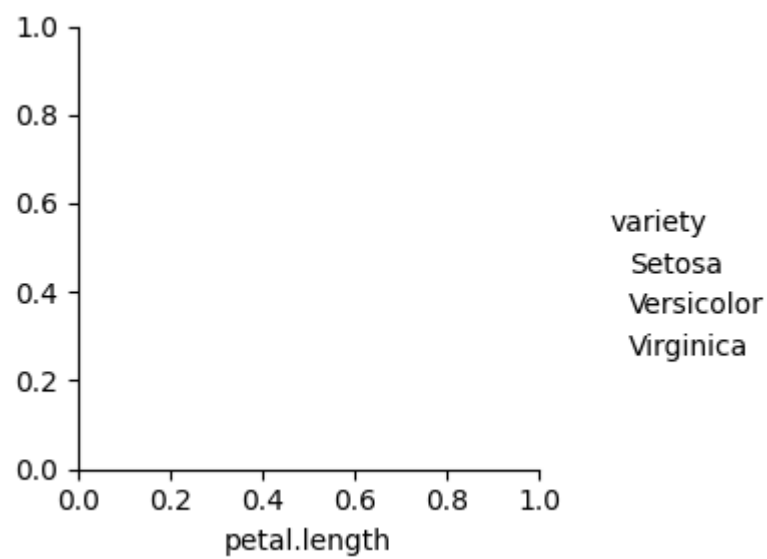
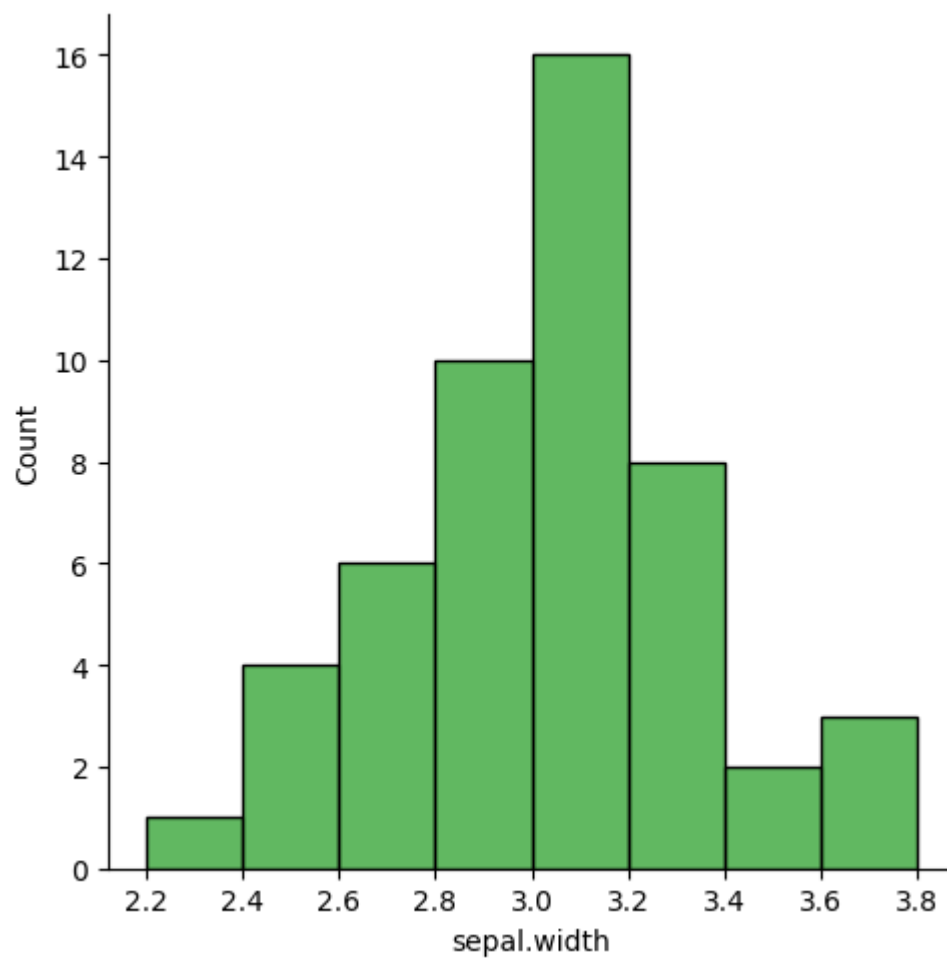


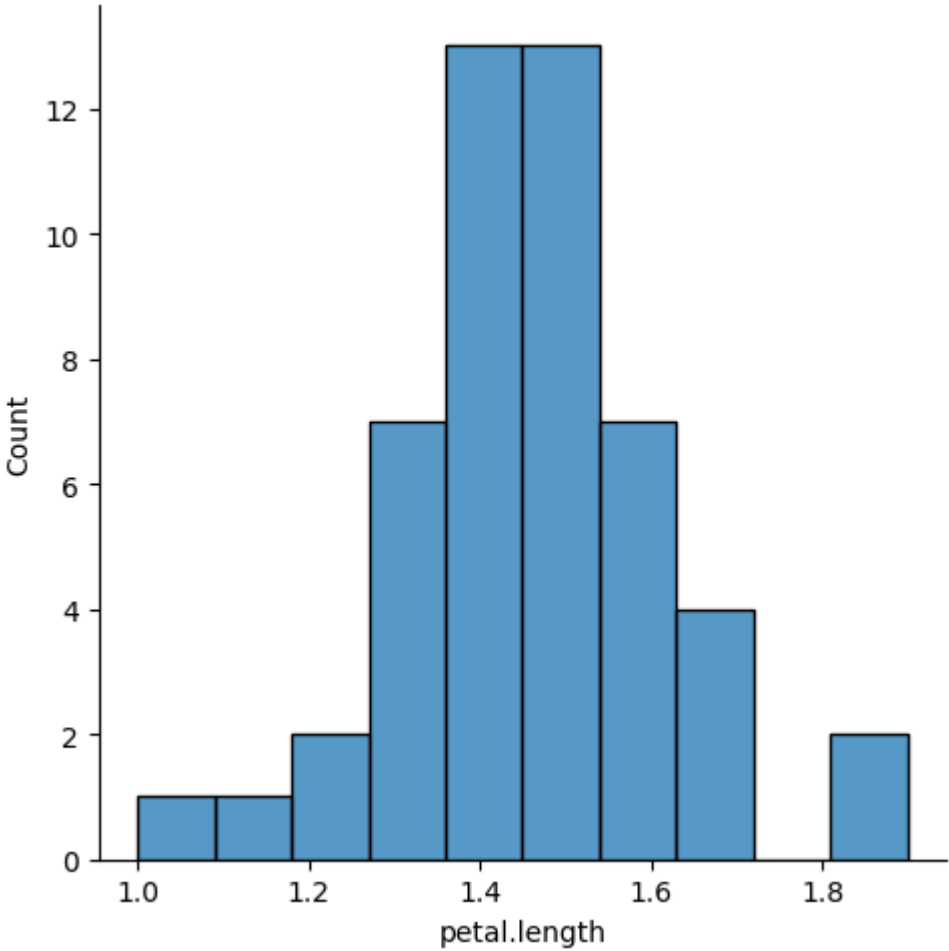


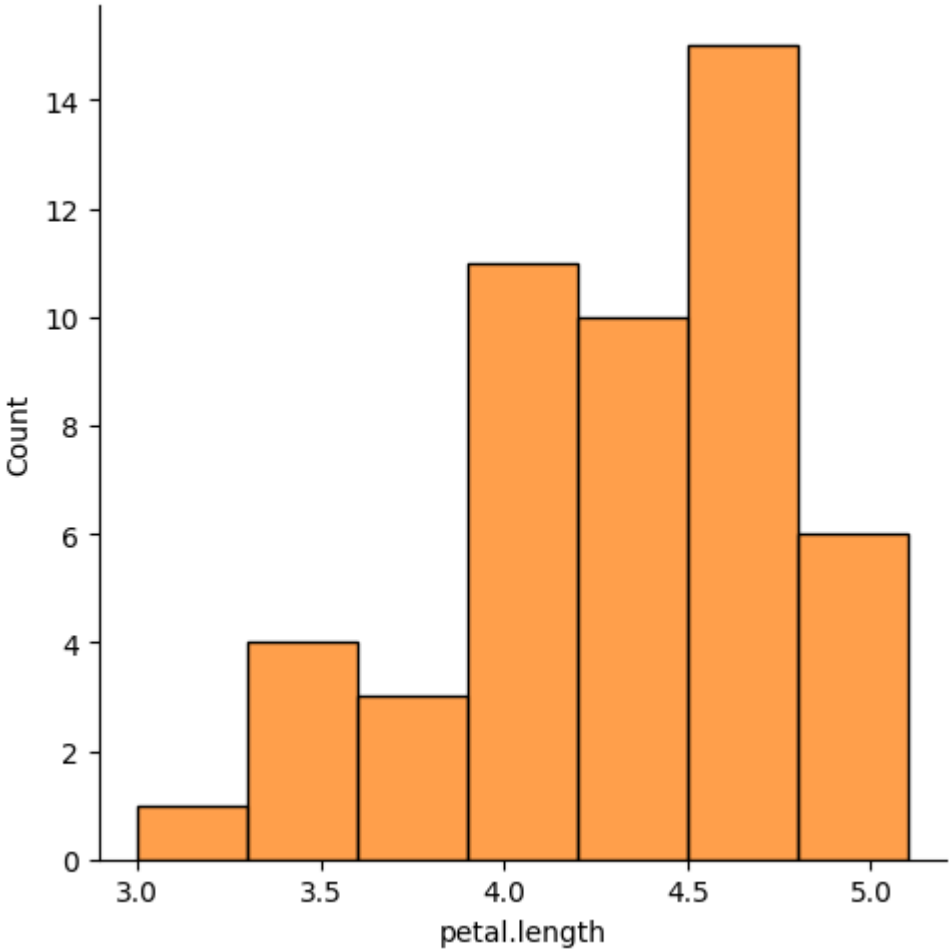


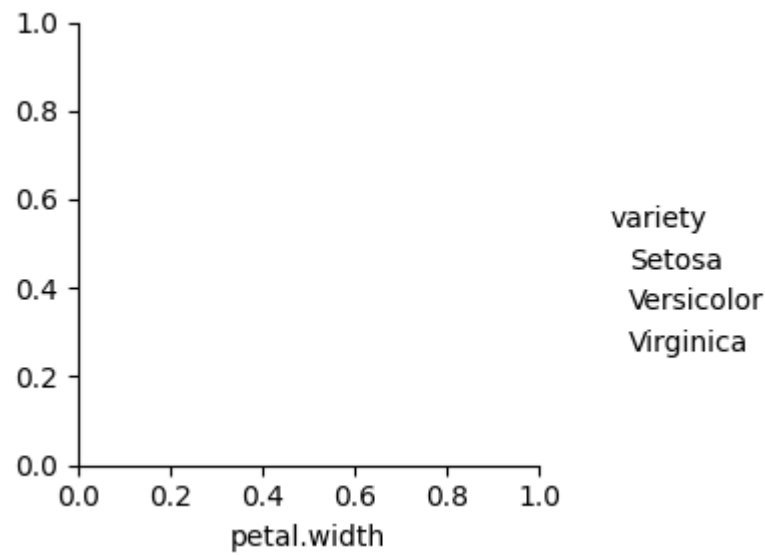
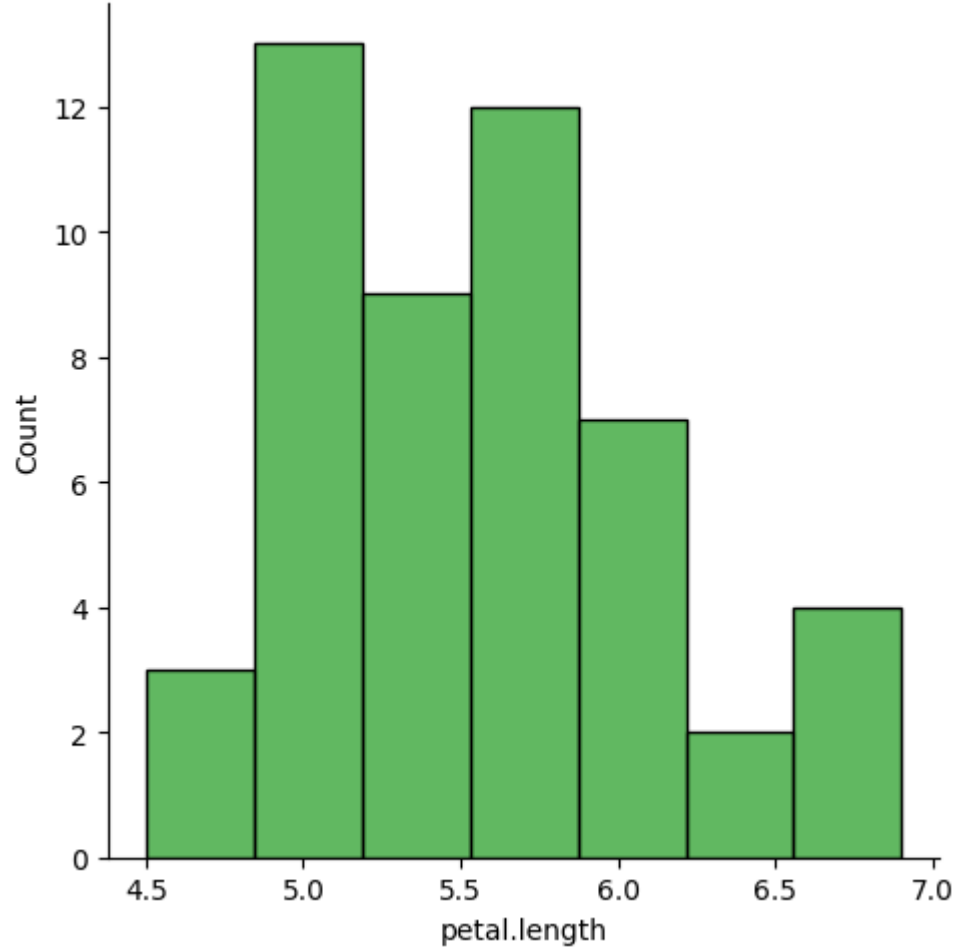


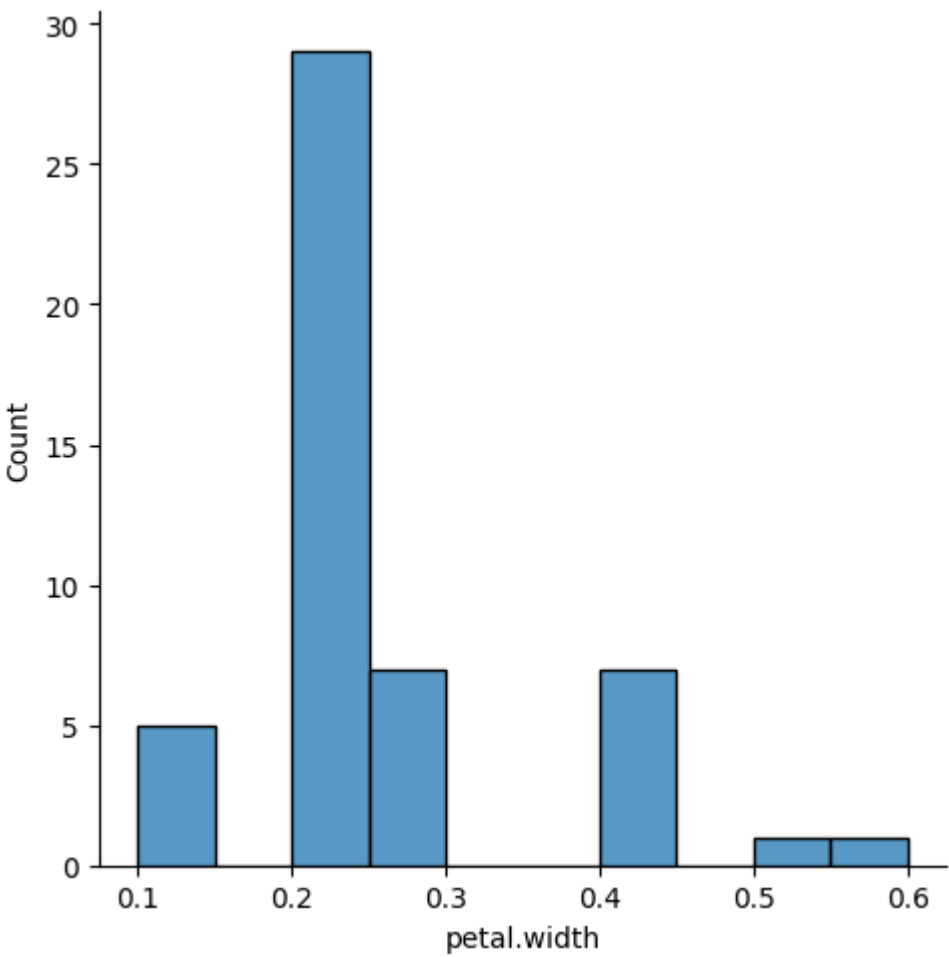


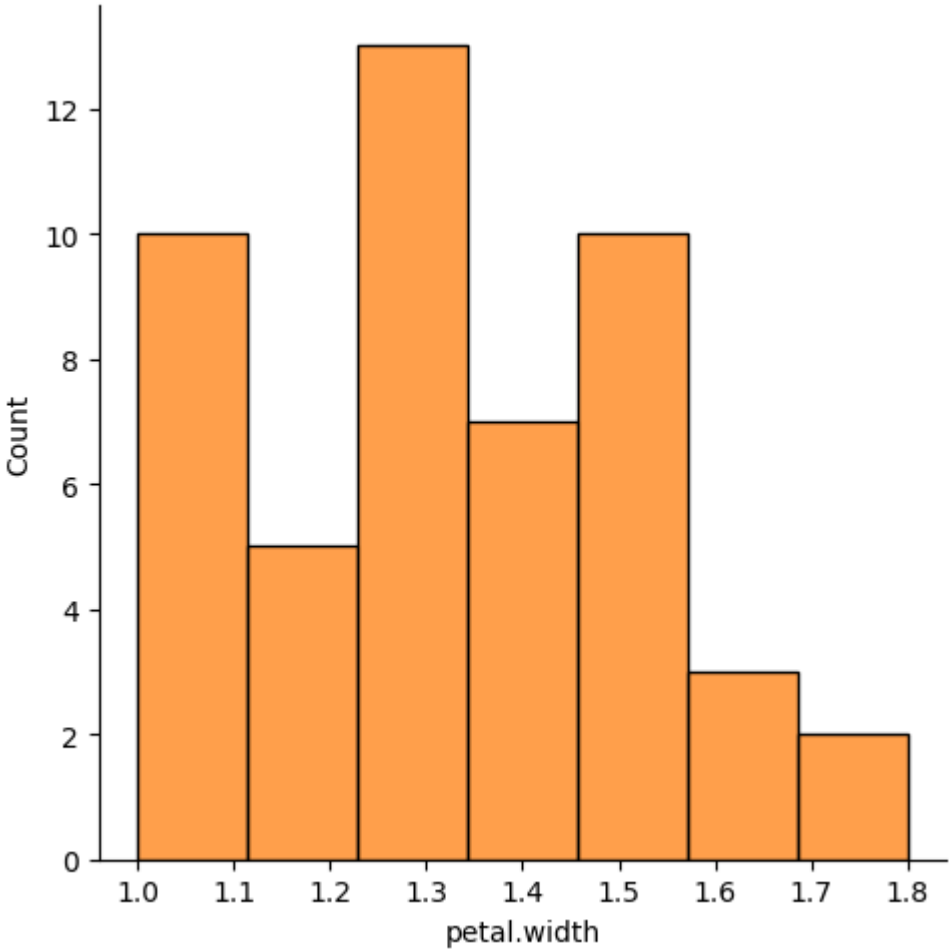


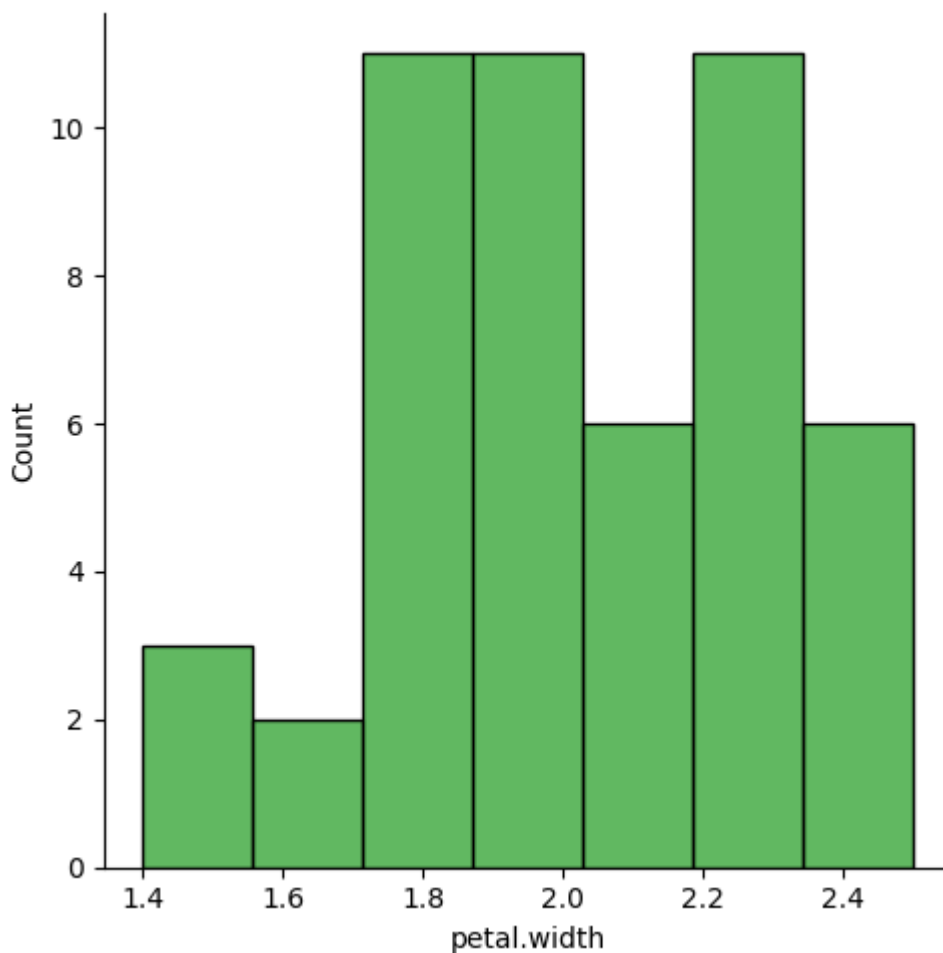






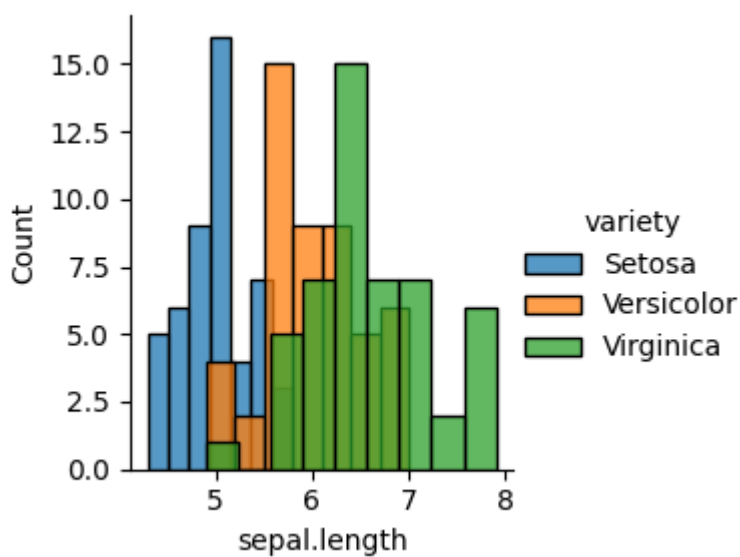


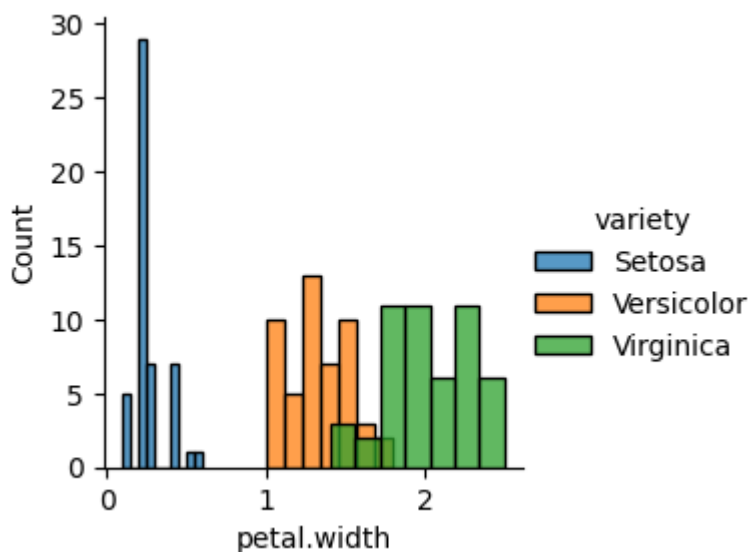
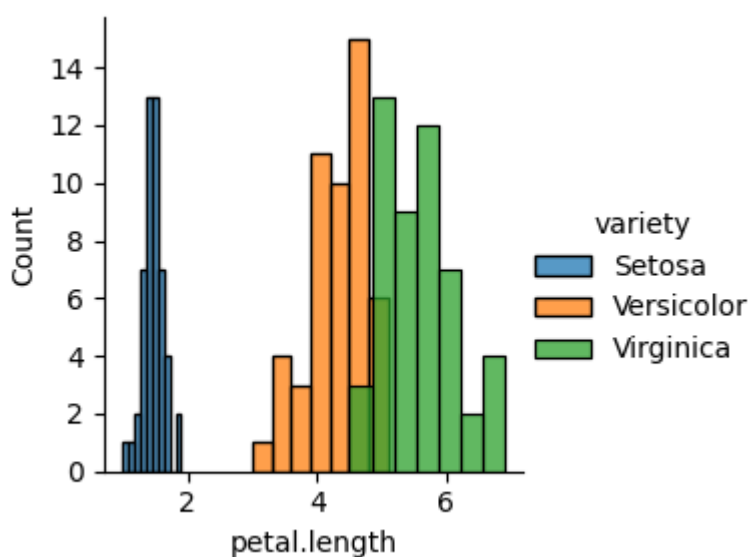
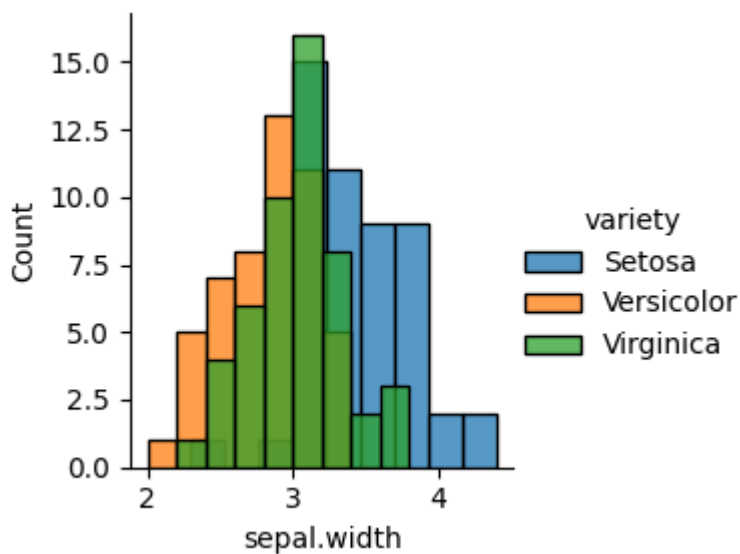




```
In [59]: plot = sns.FacetGrid(df, hue="variety")
plot.map(sns.histplot, 'sepal.length').add_legend()
plot = sns.FacetGrid(df, hue="variety")
plot.map(sns.histplot, 'sepal.width').add_legend()
plot = sns.FacetGrid(df, hue="variety")
plot.map(sns.histplot, 'petal.length').add_legend()
plot = sns.FacetGrid(df, hue="variety")
plot.map(sns.histplot, 'petal.width').add_legend()
```

Out[59]: <seaborn.axisgrid.FacetGrid at 0xcf97eddea0>



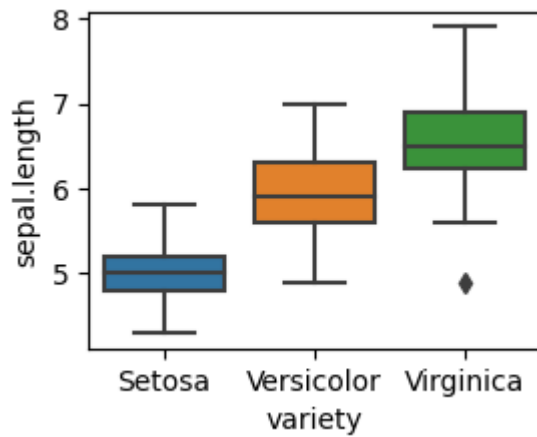


```
In [61]: def graph(y):  
         sns.boxplot(x='variety', y=y, data=df)
```

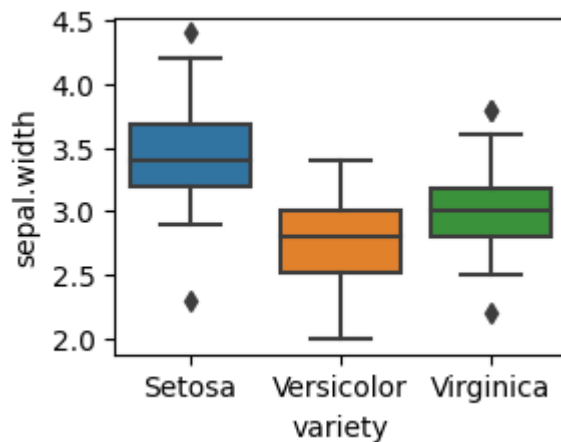
```
In [65]: plt.figure(figsize=(10,10))
```

Out[65]: <Figure size 1000x1000 with 0 Axes>
<Figure size 1000x1000 with 0 Axes>

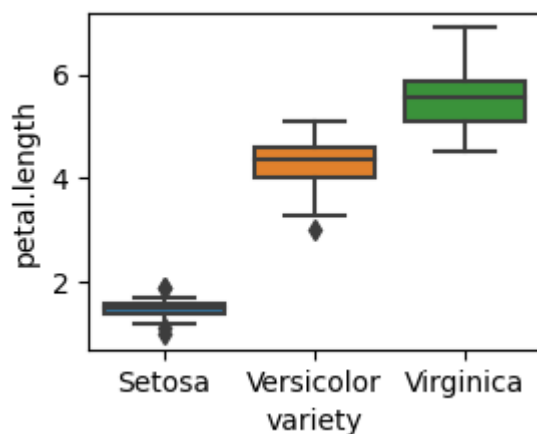
```
In [66]: plt.subplot(221)  
graph('sepal.length')
```



```
In [67]: plt.subplot(222)  
graph('sepal.width')
```

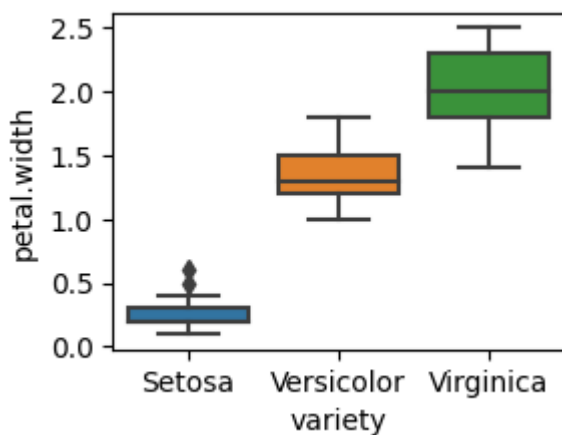


```
In [68]: plt.subplot(223)  
graph('petal.length')
```



```
In [70]: plt.subplot(224)
```

```
graph('petal.width')
```



```
In [71]: data = df.drop_duplicates(subset='variety')
```

```
In [72]: data.corr(method='pearson')
```

C:\Users\CSELAB1SYS10\AppData\Local\Temp\ipykernel_1172\2721894934.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.

```
data.corr(method='pearson')
```

```
Out[72]:
```

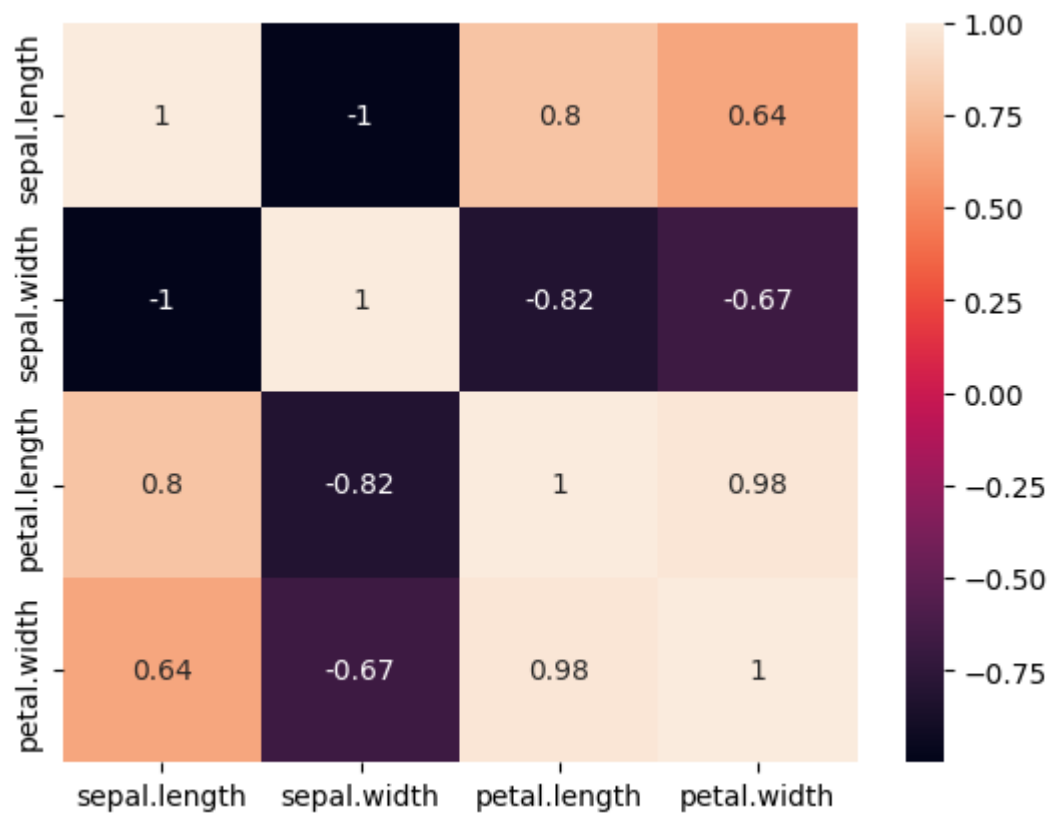
	sepal.length	sepal.width	petal.length	petal.width
sepal.length	1.000000	-0.999226	0.795795	0.643817
sepal.width	-0.999226	1.000000	-0.818999	-0.673417
petal.length	0.795795	-0.818999	1.000000	0.975713
petal.width	0.643817	-0.673417	0.975713	1.000000

```
In [75]: sns.heatmap(data.corr(method='pearson'),annot = True)
```

C:\Users\CSELAB1SYS10\AppData\Local\Temp\ipykernel_1172\1271692383.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.

```
sns.heatmap(data.corr(method='pearson'),annot = True)
```

```
Out[75]: <AxesSubplot: >
```



```
In [ ]:
```