NUMPY

```
import numpy as np
n1 = np.array([10, 20, 30, 40])
n1
```

```
array([10, 20, 30, 40])
```

```
import numpy as np
n2 = np.zeros((3,2))
n2
```

```
array([[0., 0.],
       [0., 0.],
       [0., 0.]])
```

```
n3 = np.full((4,5), 7)
n3
```

```
array([[7, 7, 7, 7, 7],
       [7, 7, 7, 7, 7],
       [7, 7, 7, 7, 7],
       [7, 7, 7, 7, 7]])
```

```
n4 = np.arange(10, 31)
n4
```

```
array([10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26,
       27, 28, 29, 30])
```

```
n5 = np.arange(10, 31, 2)
n5
```

```
array([10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30])
```

```
n5 = np.random.randint(1, 300, 7)
n5
```

```
array([186, 126, 242, 189, 138, 164,  26])
```

```
n6 = np.array([[1,2,3],[4,5,6]])
n6.shape
```

```
(2, 3)
```

```
n7 = np.array([10, 20, 30])
n8 = np.array([40, 50, 60])

np.vstack((n7, n8))
```

```
array([[10, 20, 30],
       [40, 50, 60]])
```

```
n9 = np.array([10, 20, 30])
n10 = np.array([40, 50, 60])
np.hstack((n7, n8))
```

```
array([10, 20, 30, 40, 50, 60])
```

```
n11 = np.array([10, 20, 30])
n12 = np.array([40, 50, 60])
np.column_stack((n7, n8))
```

```
array([[10, 40],
       [20, 50],
       [30, 60]])
```

```
import numpy as np
n1=np.array([1,2,3,4])
n2=np.array([4,3,2,1])
np.sum((n1,n2))
```

```
np.int64(20)
```

```
import numpy as np
n1=np.array([10,20,30,40])
np.mean(n1)
```

```
np.float64(25.0)
```

```
import numpy as np
n1=np.array([10,20,30,40])
np.std(n1)
```

```
np.float64(11.180339887498949)
```

```
import numpy as np
n1=np.array([10,20,30,40])
np.median(n1)
```

```
np.float64(25.0)
```

```
n1
```

```
array([10, 20, 30, 40])
```

```
n1.transpose()
```

```
array([10, 20, 30, 40])
```

PANDAS

```
import pandas as pd

s1=pd.Series([1,2,3,4,5])
s1
```

	0
0	1
1	2
2	3
3	4
4	5

dtype: int64

```
import pandas as pd

s1=pd.Series([1,2,3,4,5])
s2=pd.Series([10,20,30,40,50])
s1+s2
```

	0
0	11
1	22
2	33
3	44
4	55

dtype: int64

```
import pandas as pd
pd.DataFrame({"Name":['Bob','Sam','Anne'],"Marks":[76,25,92]})
```

1 to 3 of 3 entries Filter

index	Name	Marks
0	Bob	76
1	Sam	25
2	Anne	92

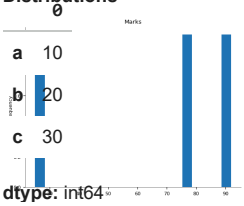
Show 25 per page

```
import pandas as pd
pd.Series({"a":10,"b":20,"c":30})
```

Distributions

	0
a	10
b	20
c	30

dtype: int64

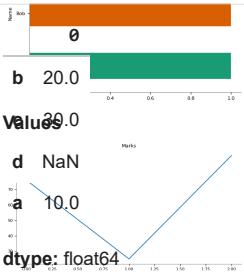


Categorical distributions

```
import pandas as pd
pd.Series({"a":10,"b":20,"c":30},index=["b","c","d","a"])
```

	0
b	20.0
c	30.0
d	NaN
a	10.0

dtype: float64



Values

Faceted distributions

```
s1=pd.Series([1,2,3,4,5,6,7,8,9])
s1[4]
```

```
np.int64(5)
```

```
s1=pd.Series([1,2,3,4,5,6,7,8,9])
s1[-4:]
```

	0
5	6
6	7
7	8
8	9

dtype: int64

```
s1=pd.Series([1,2,3,4,5,6,7,8,9])
s1[:4]
```

	0
0	1
1	2
2	3
3	4

dtype: int64

```
from sklearn.datasets import load_iris
data = load_iris()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['species'] = data.target

df.to_csv("iris.csv", index=False)
```

```
import pandas as pd
ds=pd.read_csv("/content/iris.csv")
```

```
ds.head()
```

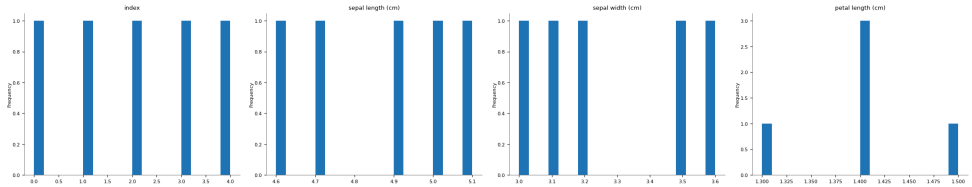
1 to 5 of 5 entries Filter

index	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	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

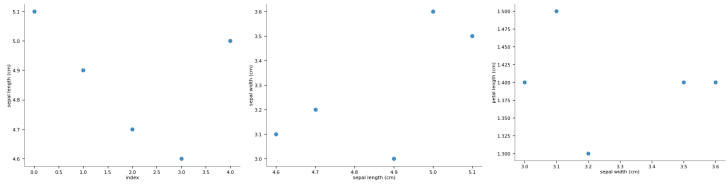
Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

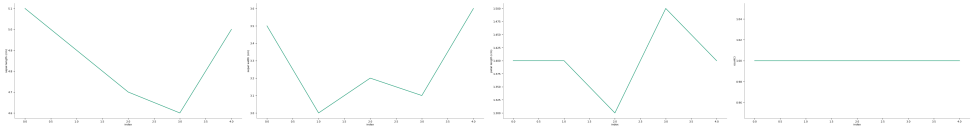
Distributions



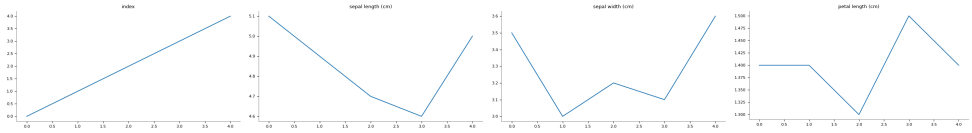
2-d distributions



Time series



Values



Next steps: Generate code with ds New interactive sheet

```
ds.tail()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	species
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

```
ds.shape
```

```
(150, 5)
```

```
ds.describe()
```

1 to 8 of 8 entries Filter

index	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	species
count	150.0	150.0	150.0	150.0	150.0
mean	5.843333333333334	3.0573333333333337	3.7580000000000005	1.1993333333333336	1.0
std	0.8280661279778629	0.435866284936698	1.7652982332594667	0.7622376689603465	0.8192319205190405
min	4.3	2.0	1.0	0.1	0.0
25%	5.1	2.8	1.6	0.3	0.0
max	7.9	4.4	6.9	2.5	2.0

```
ds.loc[10:20, ["sepal width (cm)"]]
```

1 to 11 of 11 entries Filter

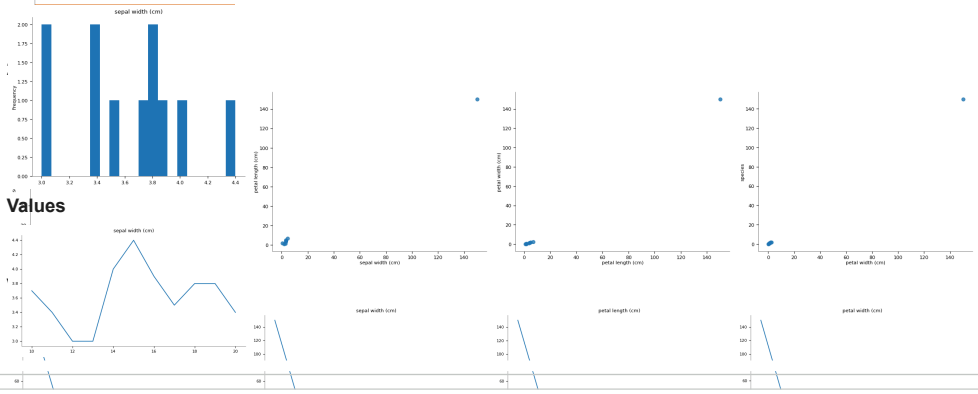
Show 100 per page

index	sepal width (cm)
10	3.7
11	3.4
12	3.0
13	3.0
14	4.0
15	4.4
16	3.9
17	3.5
18	3.8
19	3.8
20	3.4

Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

Distributions

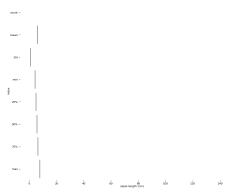


Values

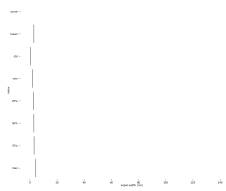

```
ds.iloc[0:3,0:2]
```

Faceted distributions

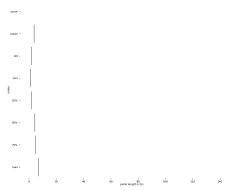
```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and
```



```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and
```



```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and
```



```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and
```

1 to 3 of 3 entries Filter

index	sepal length (cm)	sepal width (cm)

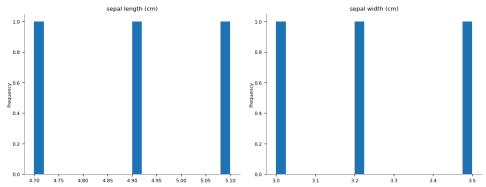
```
ds.drop([1,2,3],axis=0)
```

2	4.7	3.2

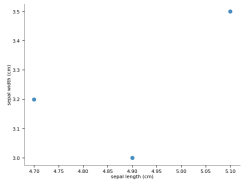
Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

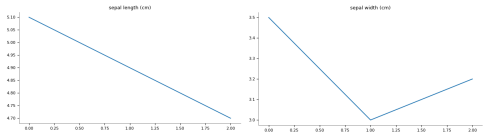
Distributions



2-d distributions



Values



1 to 25 of 147 entries Filter

index	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	species
0	5.1	3.5	1.4	0.2	0
4	5.0	3.6	1.4	0.2	0
5	5.4	3.9	1.7	0.4	0
6	4.6	3.4	1.4	0.3	0
7	5.0	3.4	1.5	0.2	0
8	4.4	2.9	1.4	0.2	0
9	4.9	3.1	1.5	0.1	0
10	5.4	3.7	1.5	0.2	0
11	4.8	3.4	1.6	0.2	0
12	4.8	3.0	1.4	0.1	0
13	4.3	3.0	1.1	0.1	0

```
ds.drop("sepal length (cm)", axis=1)
```

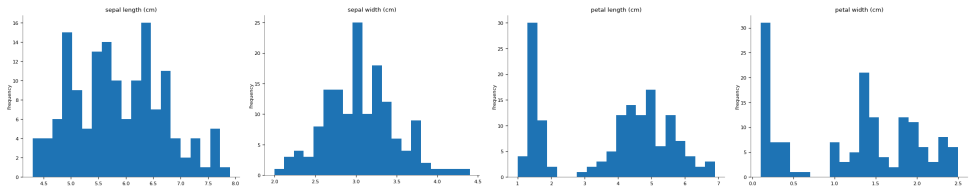
16	5.4	3.9	1.3	0.4	0
17	5.1	3.5	1.4	0.3	0
18	5.7	3.8	1.7	0.3	0
19	5.1	3.8	1.5	0.3	0
20	5.4	3.4	1.7	0.2	0
21	5.1	3.7	1.5	0.4	0
22	4.6	3.6	1.0	0.2	0
23	5.1	3.3	1.7	0.5	0
24	4.8	3.4	1.9	0.2	0
25	5.0	3.0	1.6	0.2	0
26	5.0	3.4	1.6	0.4	0
27	5.2	3.5	1.5	0.2	0

Show 25 per page

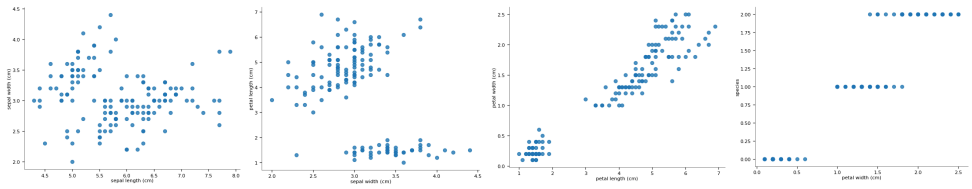
1 2 3 4 5 6

Like what you see? Visit the data table notebook to learn more about interactive tables.

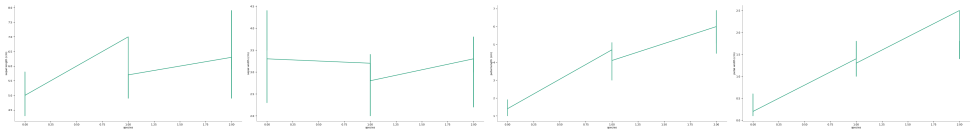
Distributions



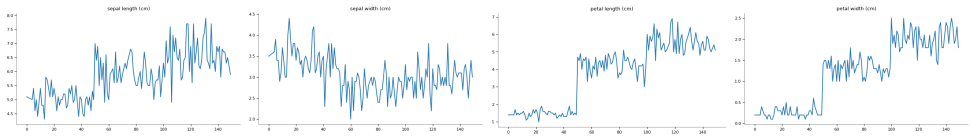
2-d distributions



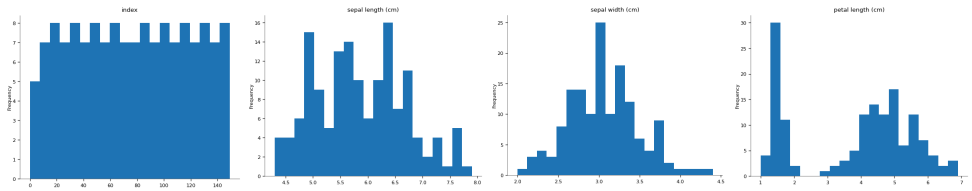
Time series



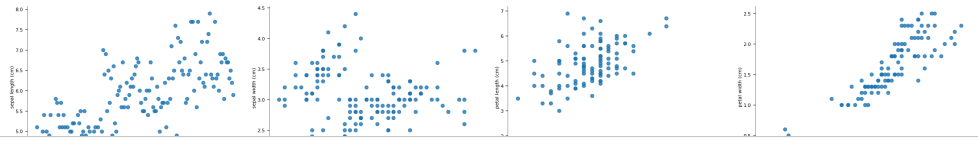
Values

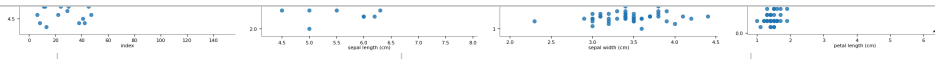


Distributions



2-d distributions





1 to 25 of 150 entries Filter

Time series

index	sepal width (cm)	petal length (cm)	petal width (cm)	species
0	3.5	1.4	0.2	0
1	3.0	1.4	0.2	0
2	3.2	1.3	0.2	0
3	3.1	1.5	0.2	0
4	3.6	1.4	0.2	0
5	3.9	1.7	0.4	0
6	3.4	1.4	0.3	0
7	3.4	1.5	0.2	0
8	2.9	1.4	0.2	0
9	3.1	1.5	0.1	0
10	3.7	1.5	0.2	0
11	3.4	1.6	0.2	0
12	3.0	1.4	0.1	0
13	3.0	1.1	0.1	0
14	4.0	1.2	0.2	0
15	4.4	1.5	0.4	0
16	3.9	1.3	0.4	0
17	3.5	1.4	0.3	0
18	3.8	1.7	0.3	0
19	3.8	1.5	0.3	0
20	3.4	1.7	0.2	0
21	3.7	1.5	0.4	0

Open in Colab

MATLIB

```
import numpy as np
from matplotlib import pyplot as plt
```

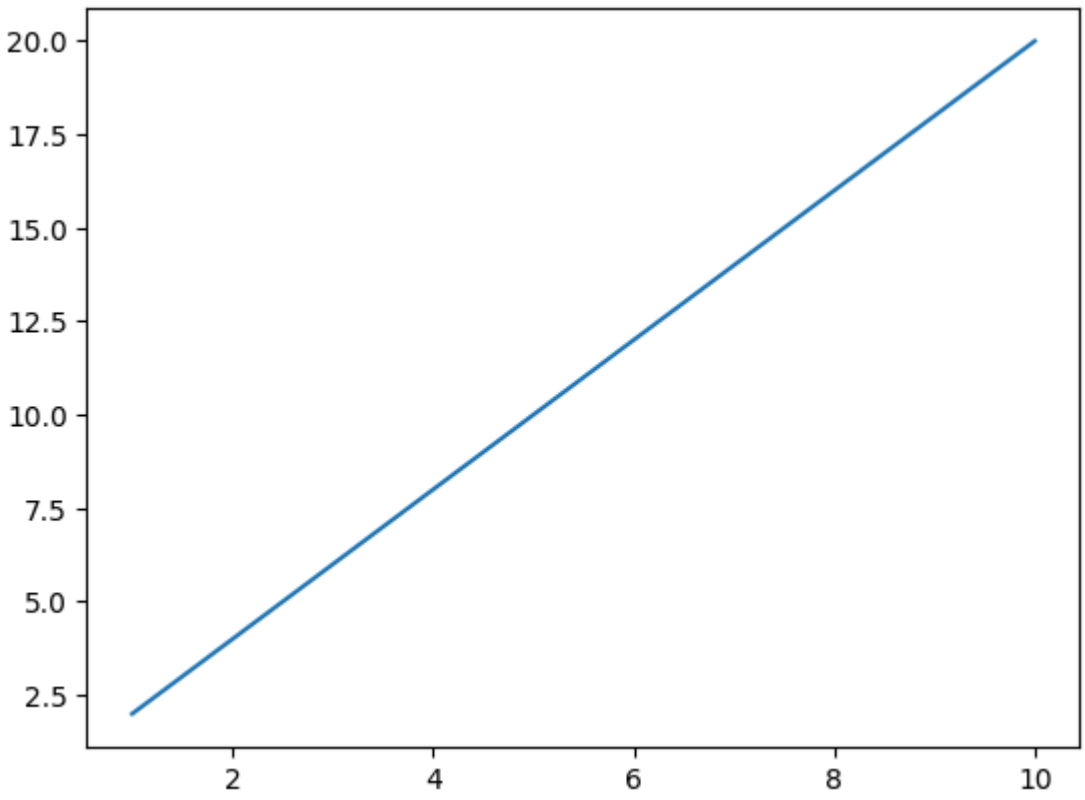
```
x=np.arange(1,11)
x
```

```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

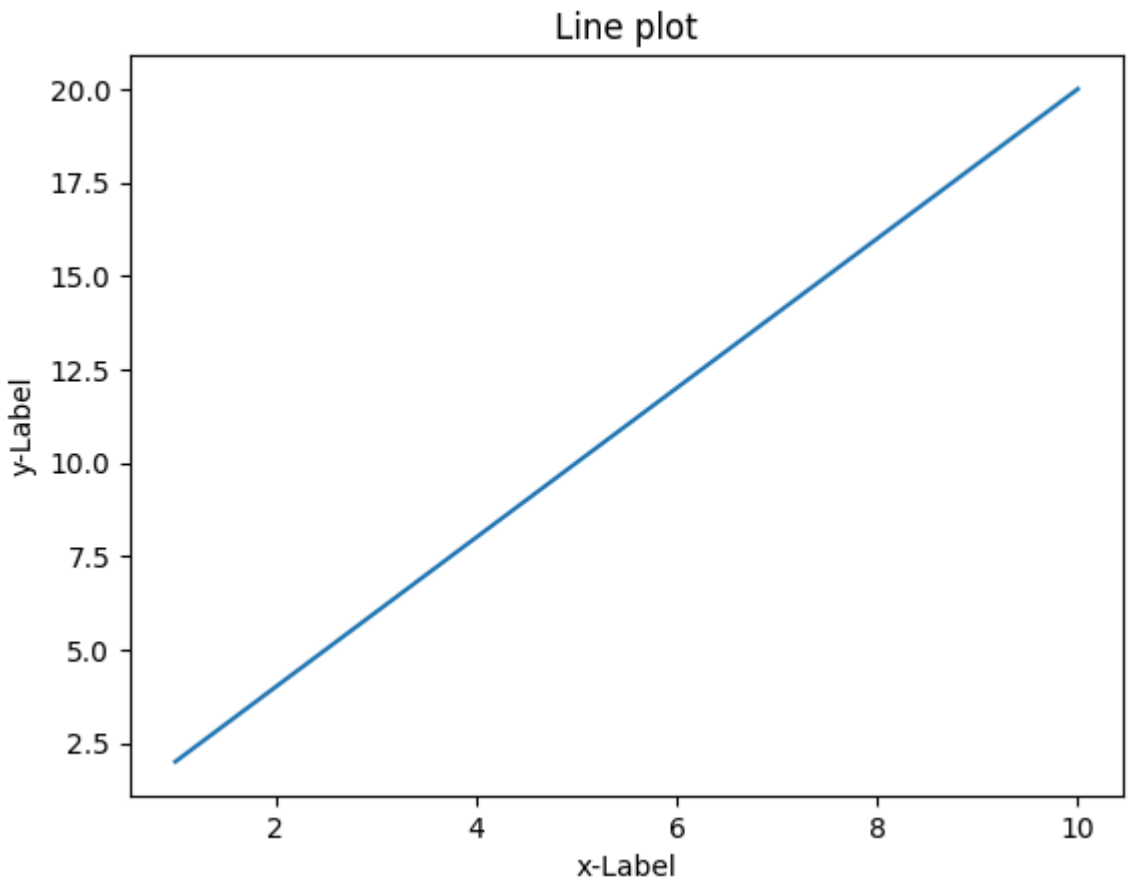
```
y=2*x
y
```

```
array([ 2,  4,  6,  8, 10, 12, 14, 16, 18, 20])
```

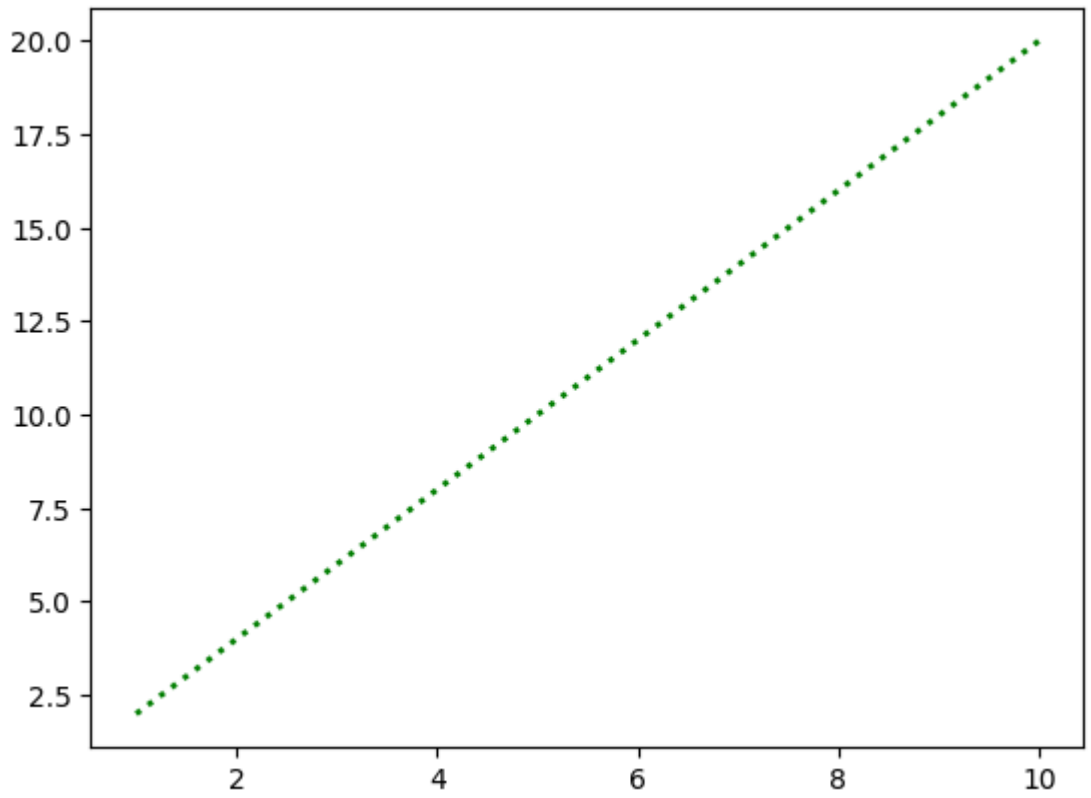
```
plt.plot(x,y)
plt.show()
```




```
plt.plot(x,y)
plt.title("Line plot")
plt.xlabel("x-Label")
plt.ylabel("y-Label")
plt.show()
```




```
plt.plot(x,y,color="g",linestyle=':',linewidth=2)
plt.show()
```




```
x=np.arange(1,11)
y1=2*x
```

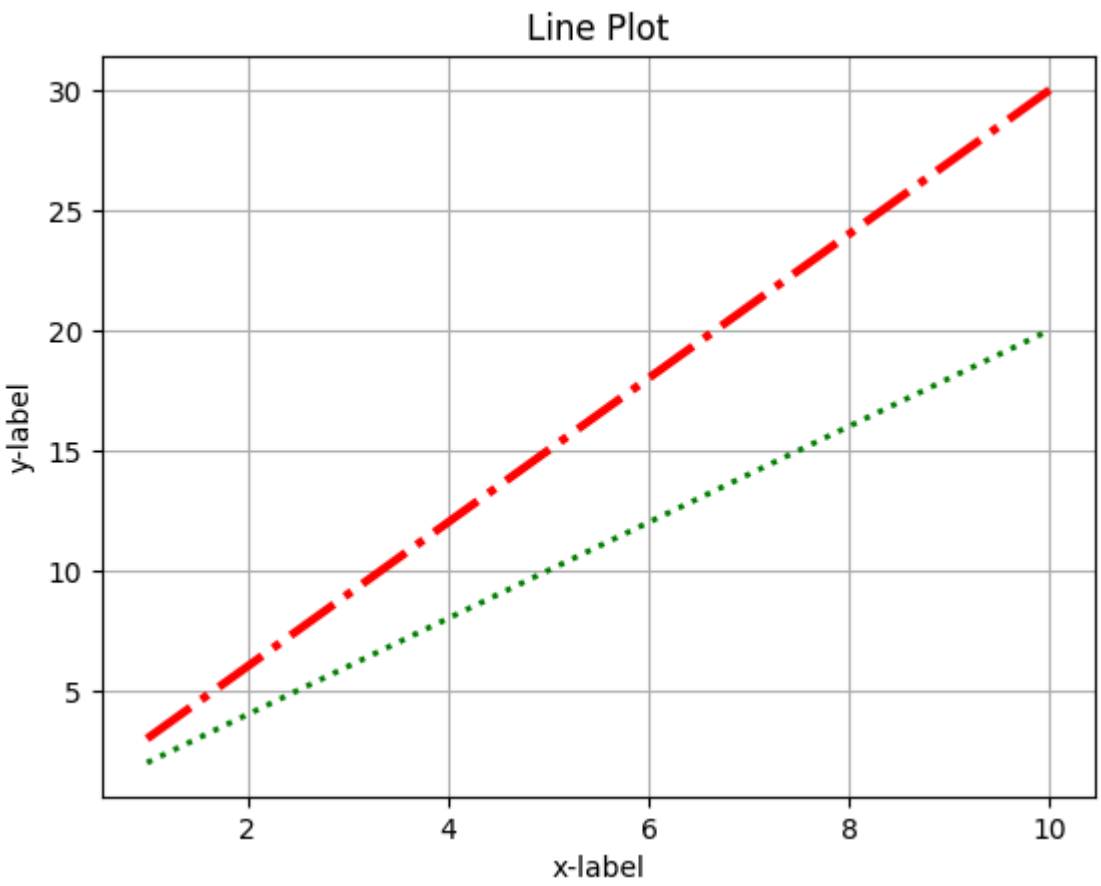
```
y2=3*x
```

```
plt.plot(x,y1,color='g',linestyle=':',linewidth=2)
plt.plot(x,y2,color='r',linestyle='-.',linewidth=3)
plt.title("Line Plot")
plt.xlabel("x-label")
plt.ylabel("y-label")
plt.grid(True)
plt.show
```

matplotlib.pyplot.show
def show(*args, **kwargs) -> None

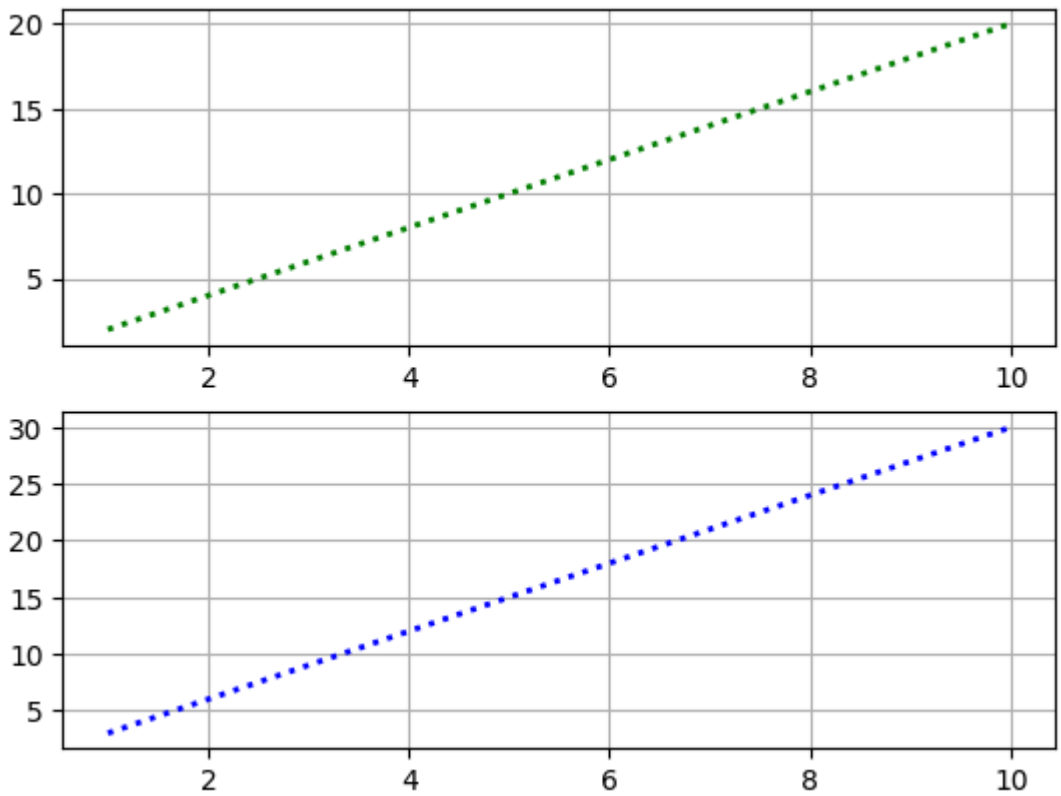
/usr/local/lib/python3.12/dist-packages/matplotlib/pyplot.py
Display all open figures.

Parameters
---
block : bool, optional




```
x=np.arange(1,11)
y1=2*x
y2=3*x

plt.subplot(2,1,1)
plt.plot(x,y1,color='g',linestyle=':',linewidth=2)
plt.grid(True)
plt.subplot(2,1,2)
plt.plot(x,y2,color='b',linestyle=':',linewidth=2)
plt.grid(True)
```

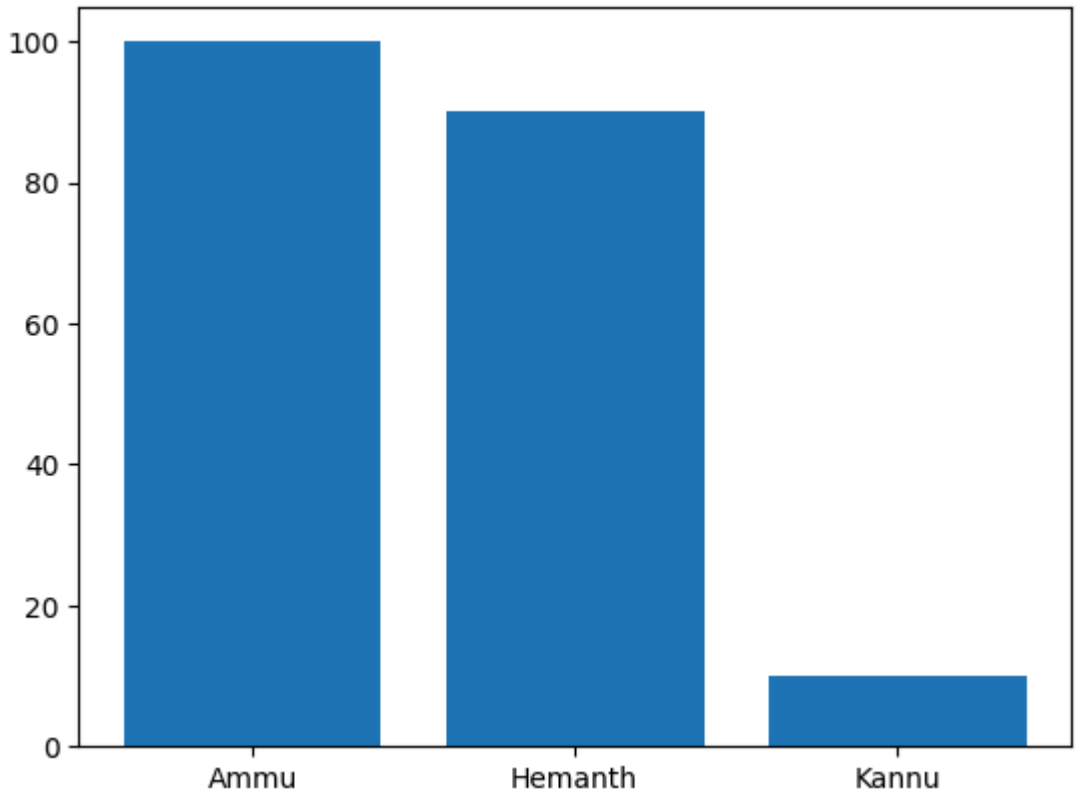
```
plt.show()
```



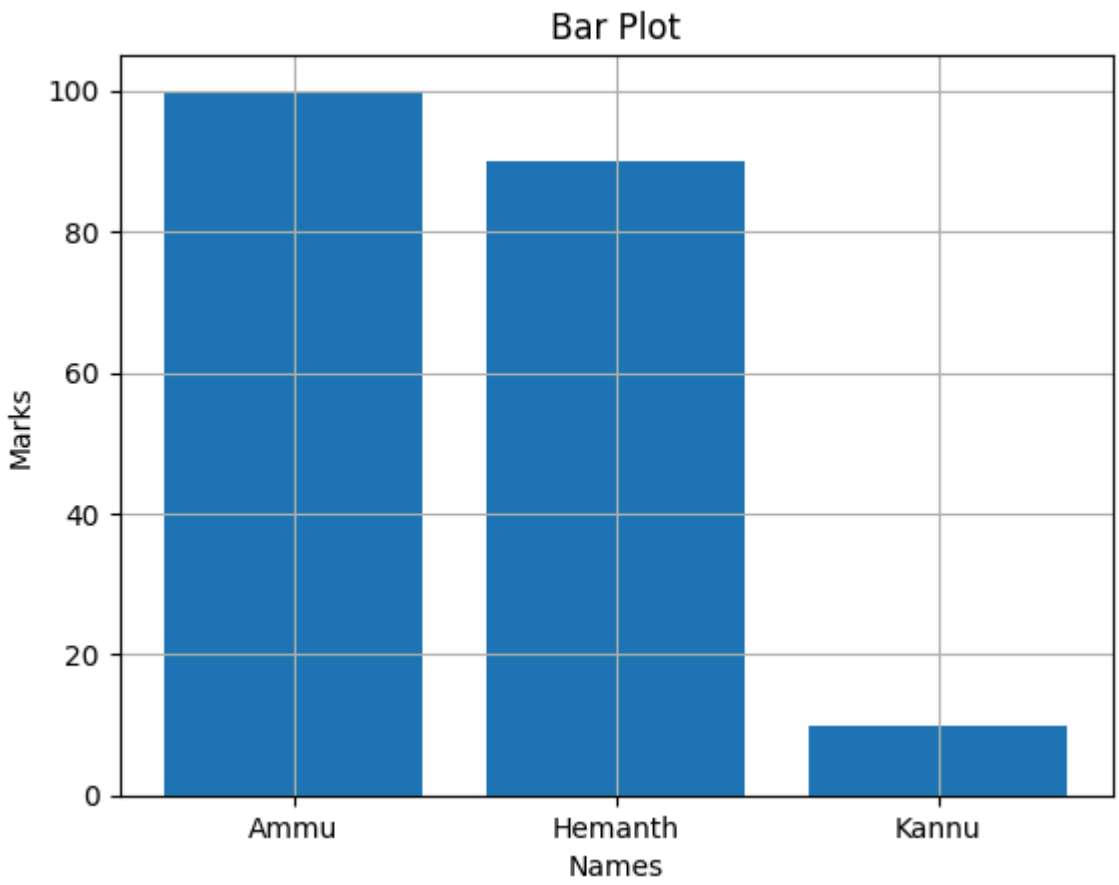

```
student = {"Ammu":100, "Hemanth":90,"Kannu":10}
```

```
names = list(student.keys())
values = list(student.values())
```

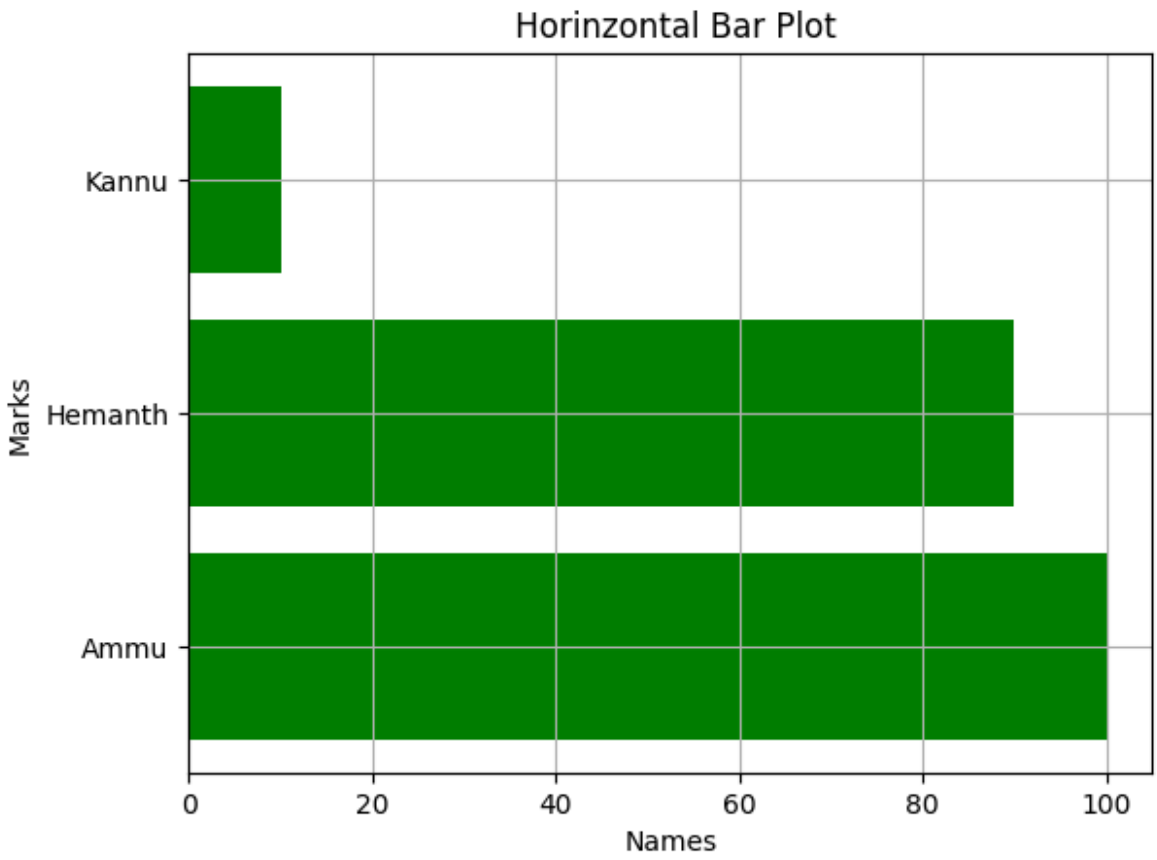
```
plt.bar(names,values)
plt.show()
```



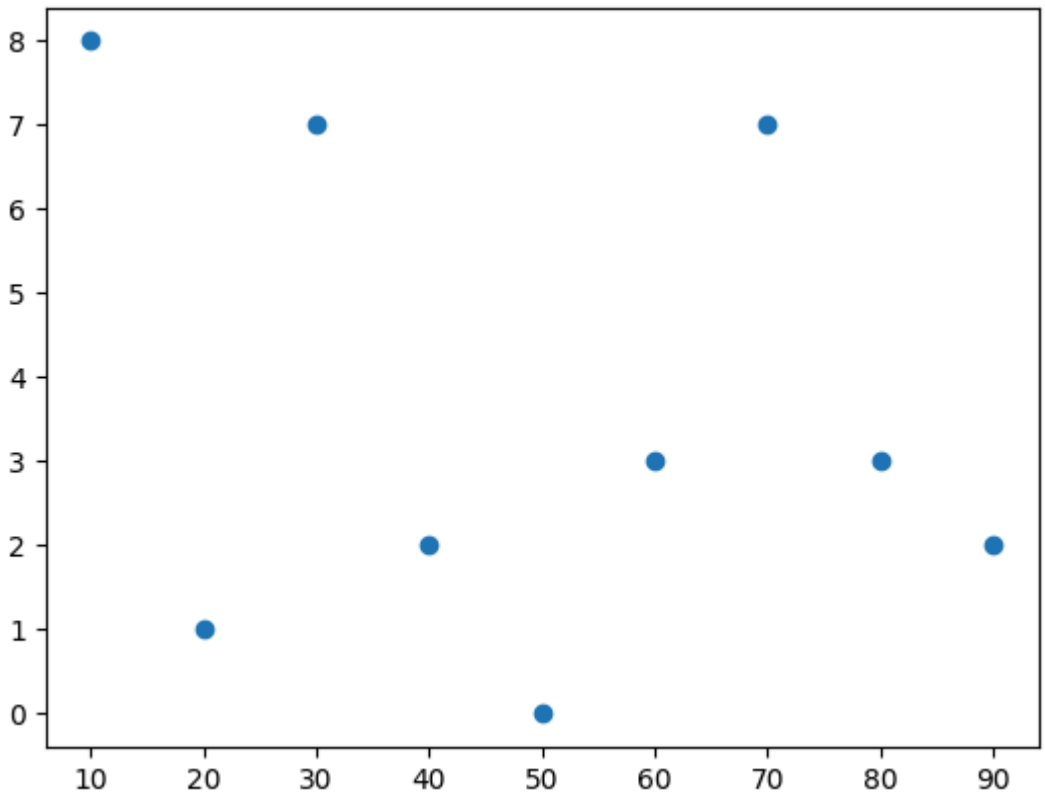

```
plt.bar(names,values)
plt.title("Bar Plot")
plt.xlabel("Names")
plt.ylabel("Marks")
plt.grid(True)
plt.show()
```



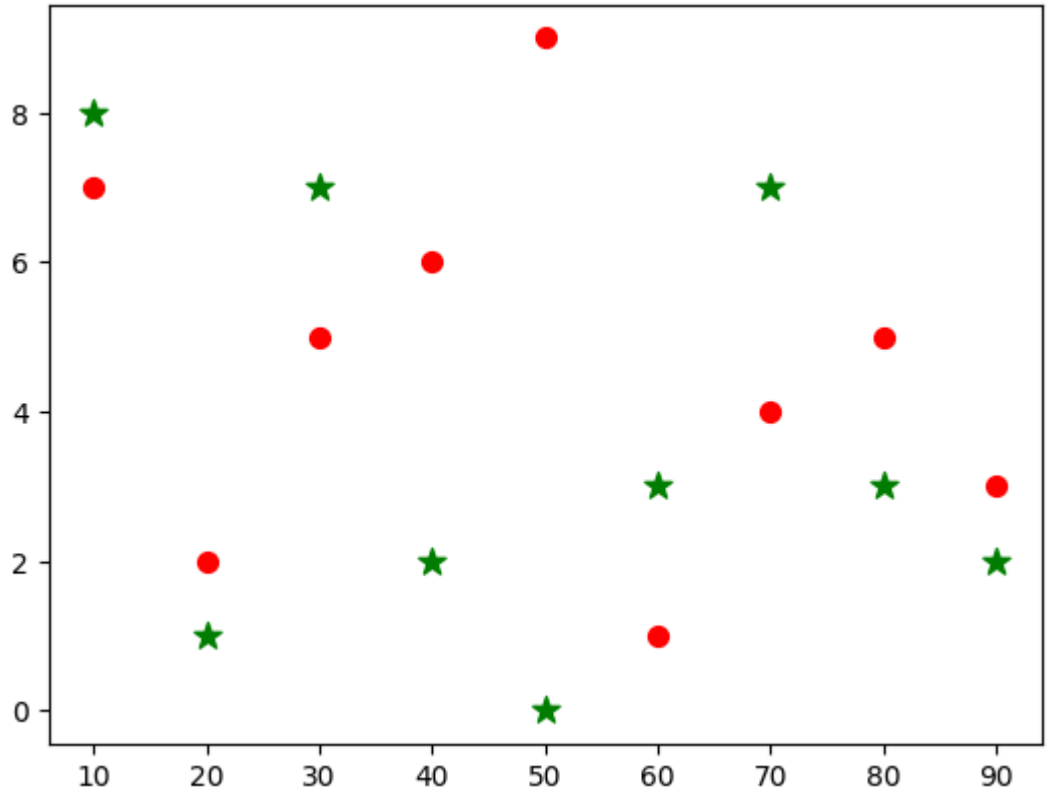

```
plt.barh(names,values,color="g")
plt.title(" Horinzontal Bar Plot")
plt.xlabel("Names")
plt.ylabel("Marks")
plt.grid(True)
plt.show()
```



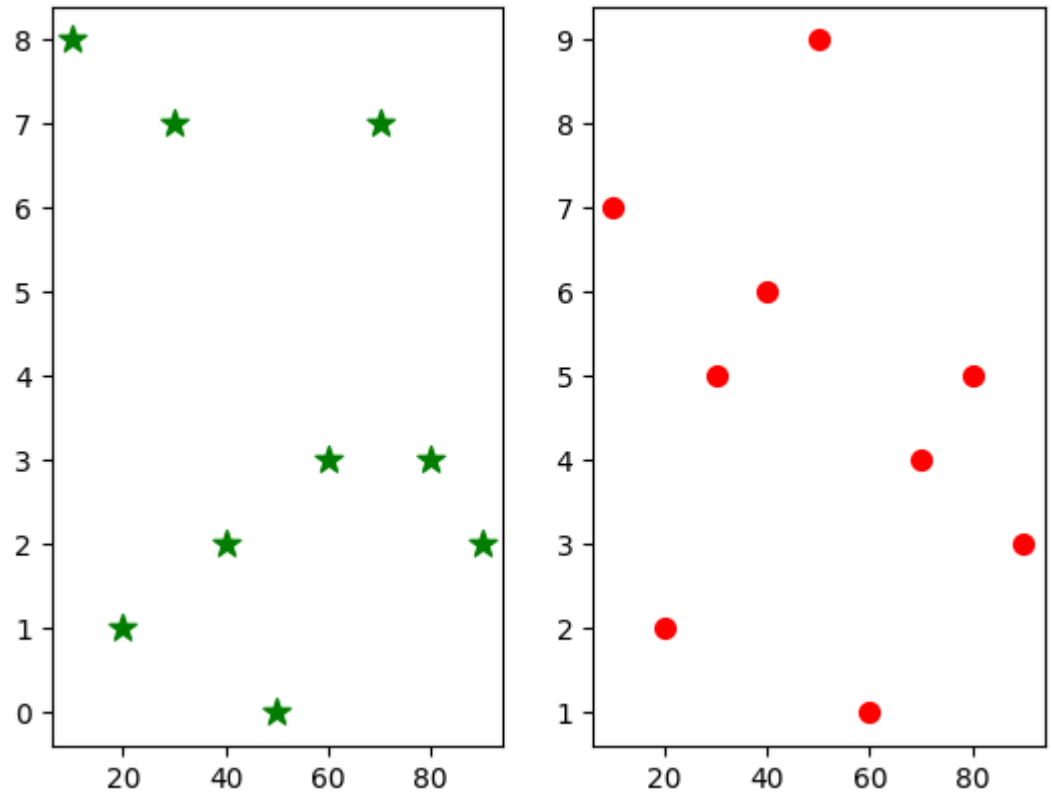

```
from matplotlib import pyplot as plt
x = [10, 20, 30, 40, 50, 60, 70, 80, 90]
a = [8,1,7,2,0,3,7,3,2]
plt.scatter(x,a)
plt.show()
```



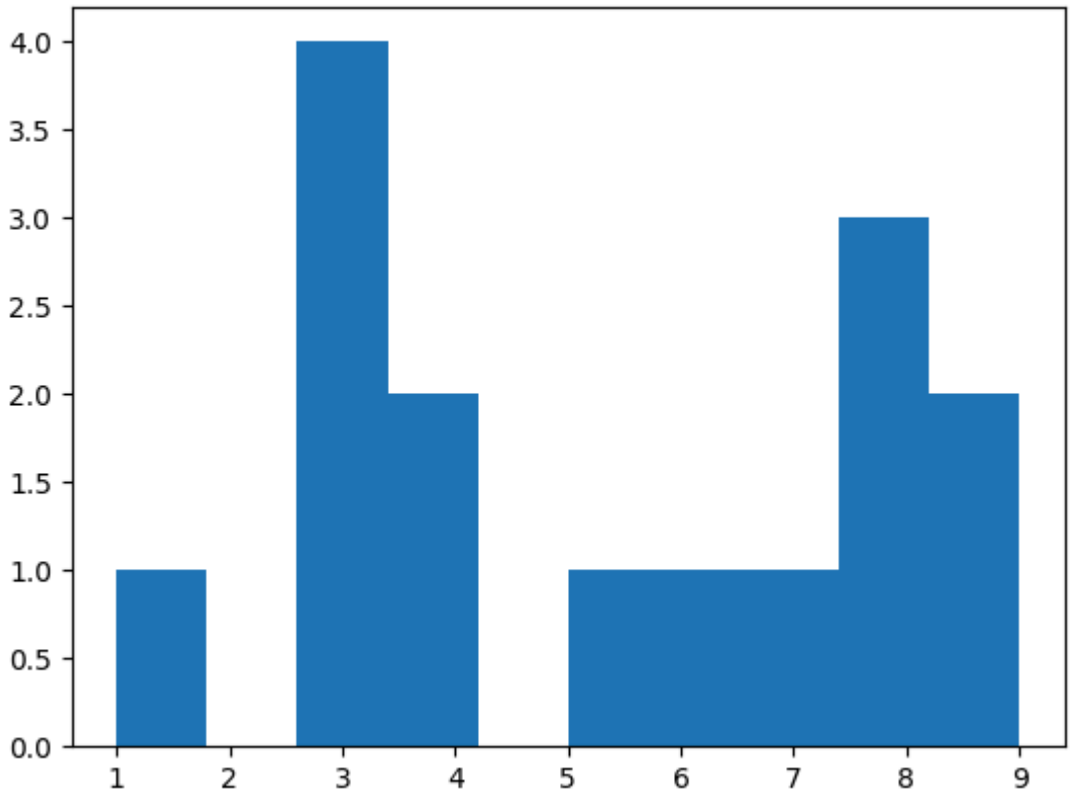
```
x = [10, 20, 30, 40, 50, 60, 70, 80, 90]
a = [8,1,7,2,0,3,7,3,2]
b = [7,2,5,6,9,1,4,5,3]
plt.scatter(x,a,marker="*",c="g",s=100)
plt.scatter(x,b,marker=".",c="r",s=200)
plt.show()
```




```
x = [10, 20, 30, 40, 50, 60, 70, 80, 90]
a = [8,1,7,2,0,3,7,3,2]
b = [7,2,5,6,9,1,4,5,3]

plt.subplot(1,2,1)
plt.scatter(x,a,marker="*",c="g",s=100)

plt.subplot(1,2,2)
plt.scatter(x,b,marker=".",c="r",s=200)
plt.show()
```

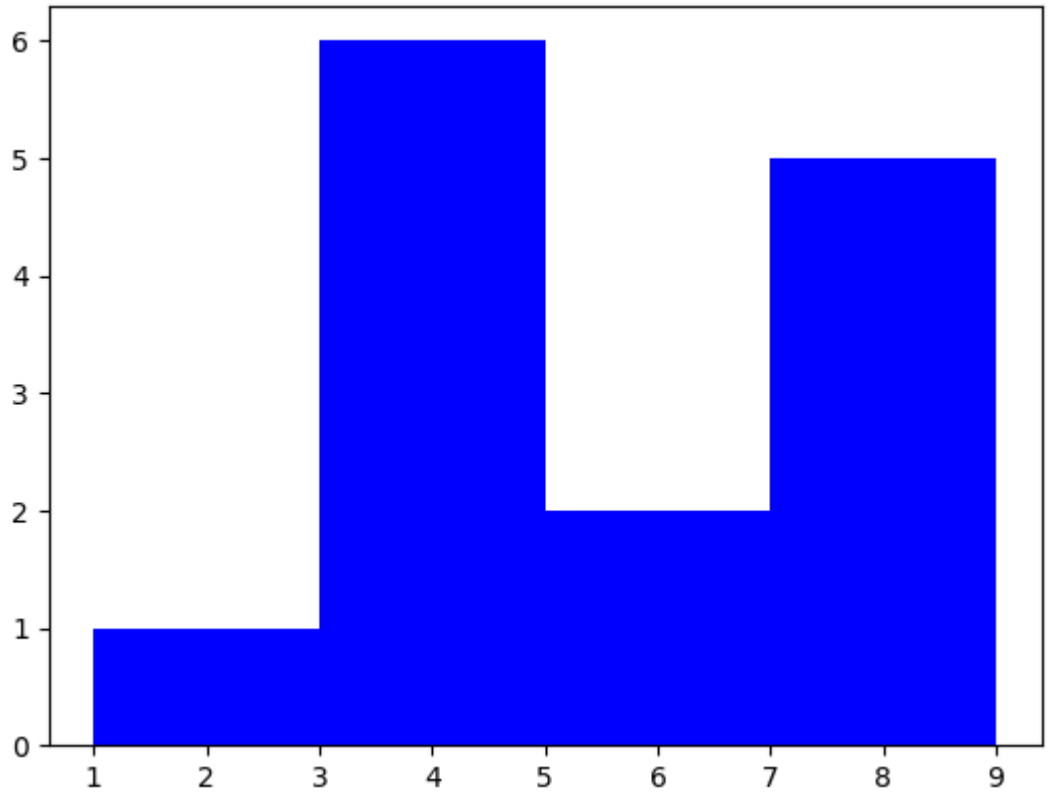



```
from matplotlib import pyplot as plt
data = [1,3,3,3,3,9,9,5,4,4,8,8,8,6,7]
plt.hist(data)
plt.show()
```



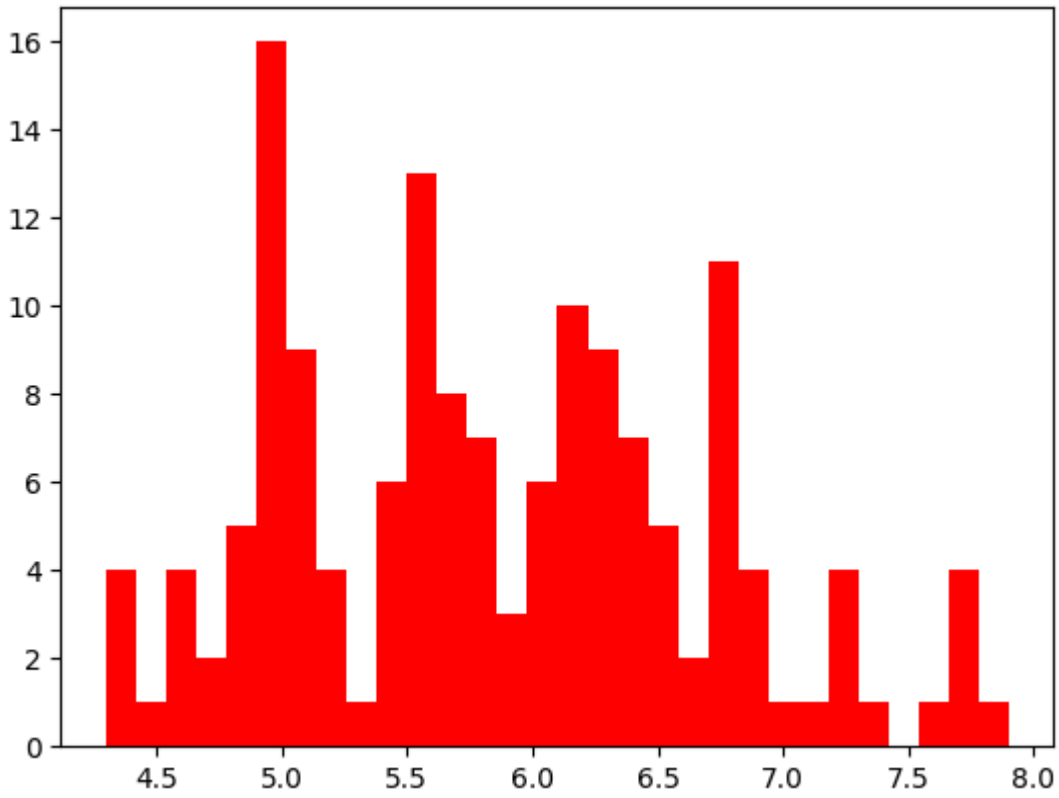
Double-click (or enter) to edit

```
from matplotlib import pyplot as plt
data = [1,3,3,3,3,9,9,5,4,4,8,8,6,7]
plt.hist(data, color='b',bins =4)
plt.show()
```

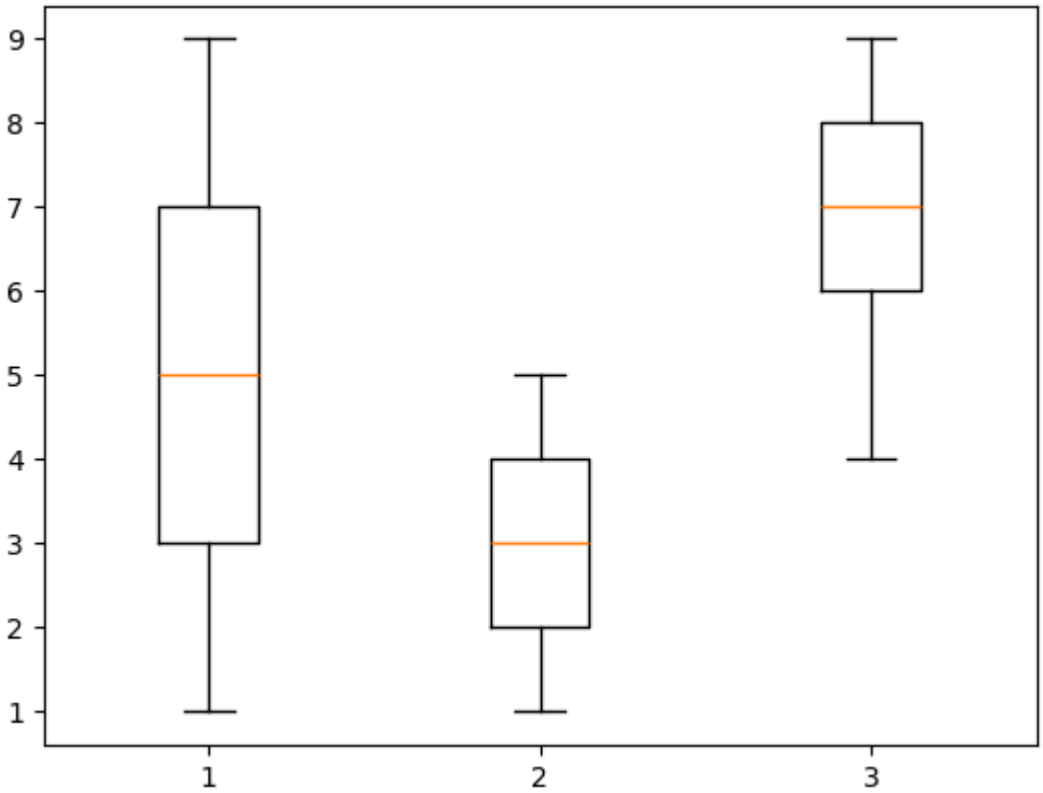



```
from sklearn.datasets import load_iris
import pandas as pd

data = load_iris()
df = pd.DataFrame(data.data, columns=["SepalLengthCm","SepalWidthCm","PetalL
df['Species'] = data.target

df.to_csv("Iris.csv", index=False)
```

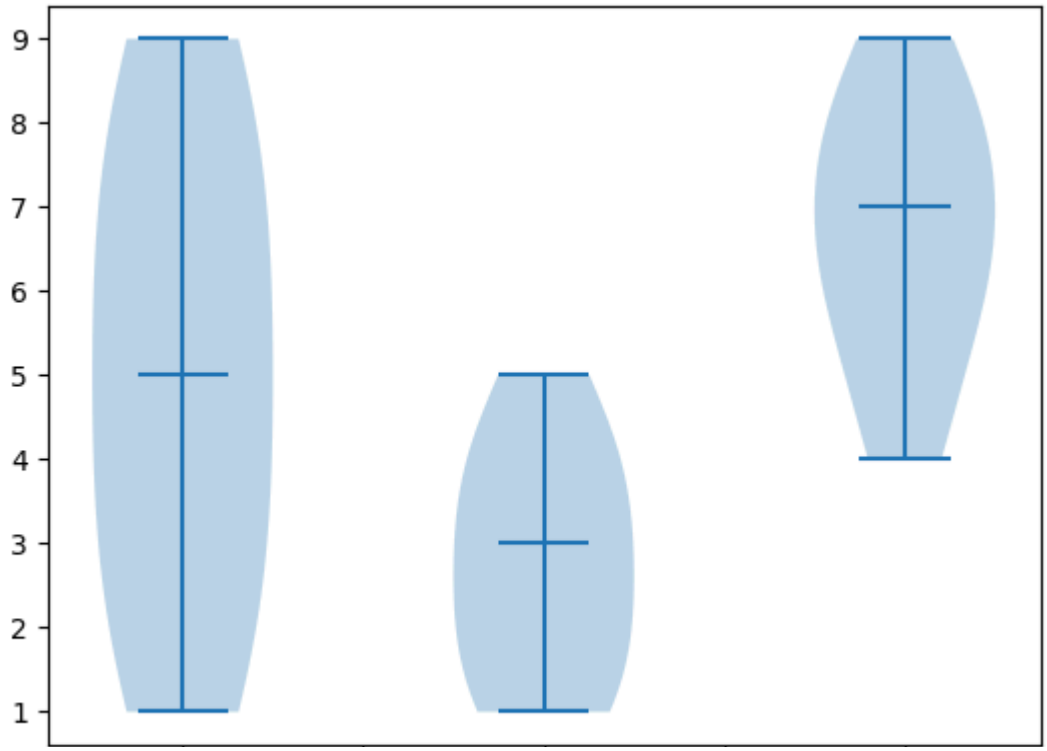
```
import pandas as pd
Iris = pd.read_csv('Iris.csv')
Iris.head()
plt.hist(Iris['SepalLengthCm'],bins=30,color="r")
plt.show()
```




```
one = [1,2,3,4,5,6,7,8,9]
two = [1,2,3,4,5,4,3,2,1]
three = [6,7,8,9,8,7,6,5,4]

data = list([one,two,three])
plt.boxplot(data)
plt.show()
```



```
plt.violinplot(data,showmedians=True)
plt.show()
```



StreamlitButton.py

```python
import streamlit as st
import pandas as pd
from io import StringIO

uploaded_file = st.file_uploader("Choose a file")
if uploaded_file is not None:
    bytes_data = uploaded_file.getvalue()
    st.write(bytes_data)

    stringio = StringIO(uploaded_file.getvalue().decode("utf-8"))
    st.write(stringio)

    string_data = stringio.read()
    st.write(string_data)

    dataframe = pd.read_csv(uploaded_file)
    st.write(dataframe)
```

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

```
ds = sns.load_dataset('tips')
```

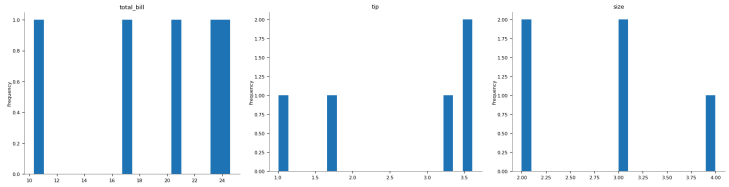
```
ds.head()
```

None

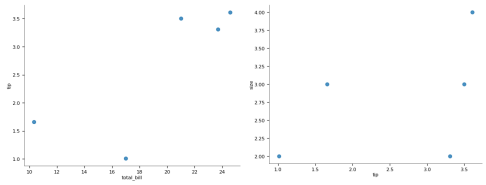
Like what you see? Visit the [data table notebook](#) to learn more about interactive tables.

```
WARNING:root:Quickchart encountered unexpected dtypes in columns: "(['sex'],)"
```

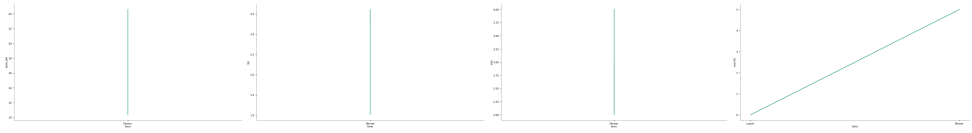
Distributions



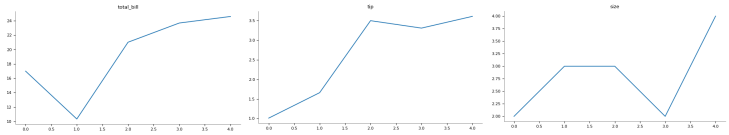
2-d distributions



Time series



Values




```
Error: Runtime no longer has a reference to this dataframe, please re-run this cell and try again.
```

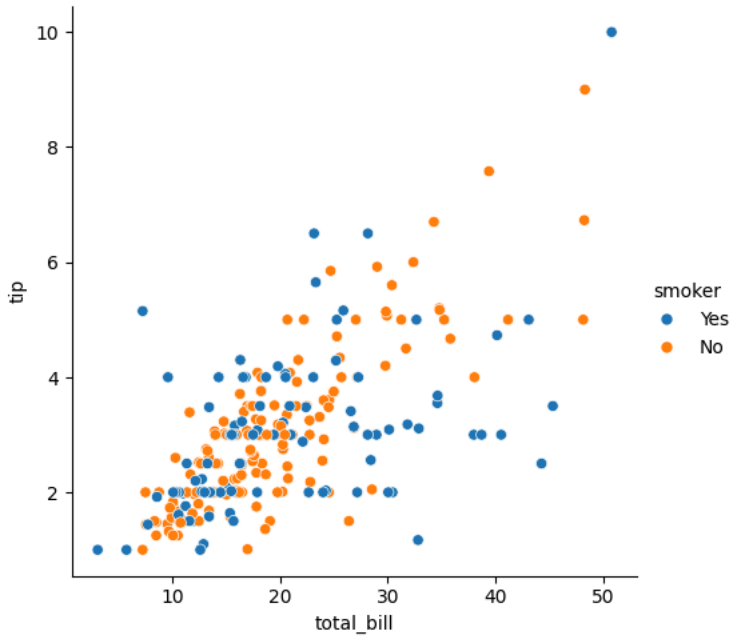
Next steps: Generate code with ds | New interactive sheet

```
ds.shape
```

```
(244, 7)
```

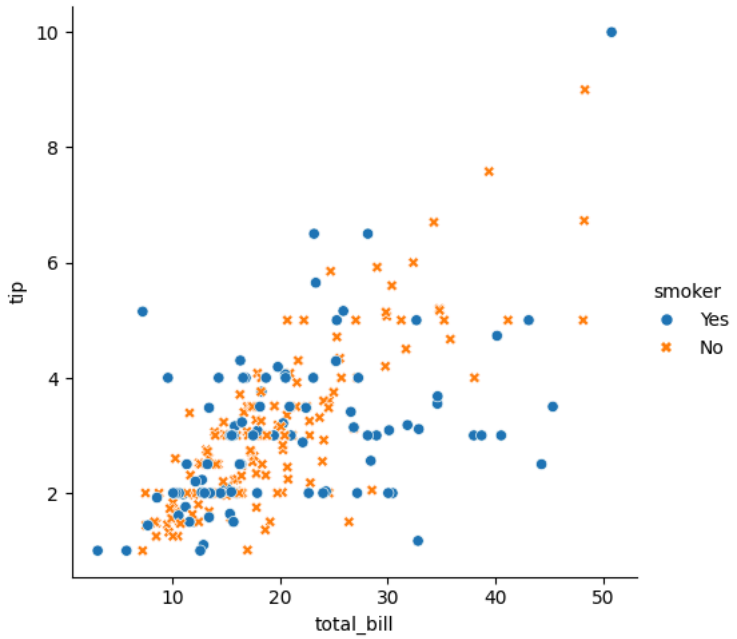
```
sns.relplot(data=ds,x='total_bill',y='tip',hue='smoker')
```

```
<seaborn.axisgrid.FacetGrid at 0x7c14d5432900>
```



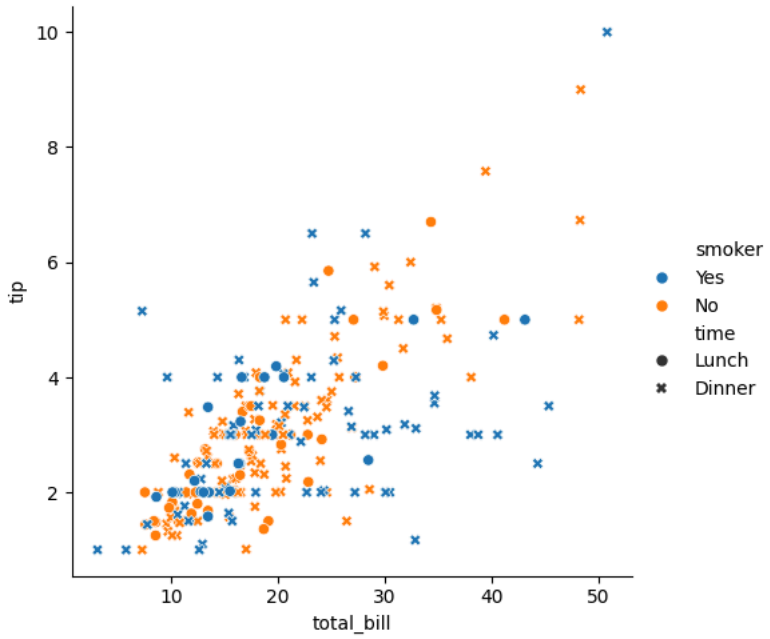

```
sns.relplot(data=ds,x='total_bill',y='tip',hue='smoker', style='smoker')
```

```
<seaborn.axisgrid.FacetGrid at 0x7c14d5437ad0>
```



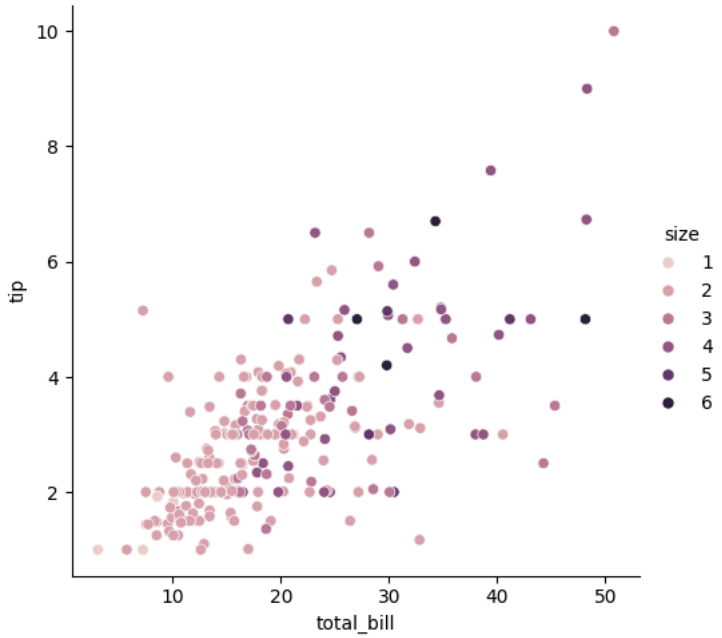

```
sns.relplot(
data=ds,
x="total_bill", y="tip", hue="smoker", style="time"
)
```

```
<seaborn.axisgrid.FacetGrid at 0x7c14d55858b0>
```



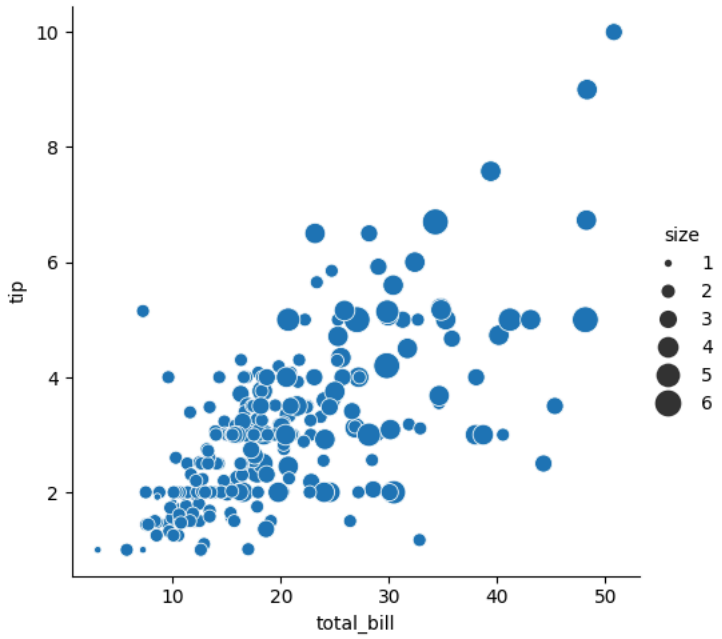

```
sns.relplot(
data=ds, x="total_bill", y="tip", hue="size",

)
```

```
<seaborn.axisgrid.FacetGrid at 0x7c14d2c218e0>
```



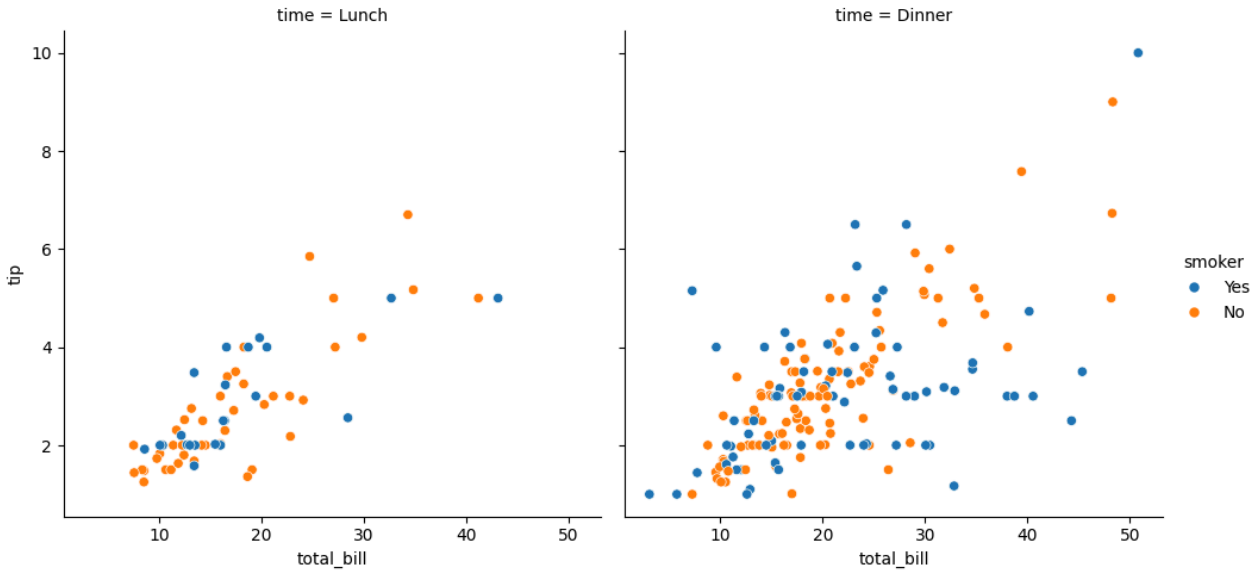

```
sns.relplot(
data=ds, x="total_bill", y="tip",
size="size", sizes=(15, 200)
)
```

```
<seaborn.axisgrid.FacetGrid at 0x7c14d2b42a80>
```




```
sns.relplot(
    data=ds,
    x="total_bill",
    y="tip",
    hue="smoker",
    col="time"
)
```

```
<seaborn.axisgrid.FacetGrid at 0x7c14d2b40230>
```




```
fmri = sns.load_dataset('fmri')
```

```
fmri.head()
```

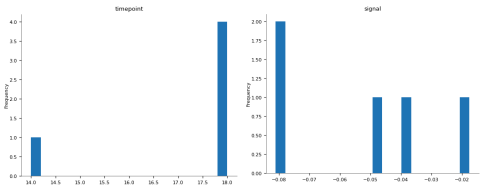
1 to 5 of 5 entries Filter

index	subject	timepoint	event	region	signal
0	s13	18	stim	parietal	-0.017551581538
1	s5	14	stim	parietal	-0.0808829319505
2	s12	18	stim	parietal	-0.0810330187333
3	s11	18	stim	parietal	-0.0461343901751999
4	s10	18	stim	parietal	-0.0379702032642

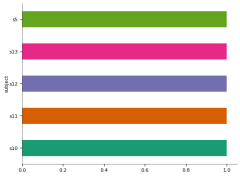
Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

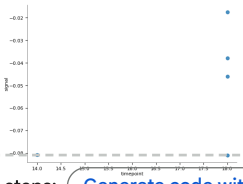
Distributions



Categorical distributions



2-d distributions



Next steps: Generate code with fmri New interactive sheet

Time series

```
fmri.shape
```

```
(1064, 5)
```