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

Colored Scatterplots

```
iris = pd.read_csv('/content/iris.csv')
iris.sample(5)
```


```
{"summary":"{\n  \"name\": \"iris\",\n  \"rows\": 5,\n  \"fields\": [\
n    {\n      \"column\": \"Id\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 50,\n        \"min\": 16,\n
\"max\": 119,\n        \"num_unique_values\": 5,\n        \"samples\":
[\n          17,\n          37,\n          109\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"SepalLengthCm\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.9848857801796105,\n        \"min\": 5.4,\n        \"max\": 7.7,\n
\"num_unique_values\": 5,\n        \"samples\": [\n          5.4,\n
5.5,\n          6.7\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"SepalWidthCm\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.8228000972289686,\n        \"min\":
2.5,\n        \"max\": 4.4,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          3.9,\n          3.5,\n          2.5\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"PetalLengthCm\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
2.7582603212894905,\n        \"min\": 1.3,\n        \"max\": 6.9,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          1.3,\n
1.5,\n          6.9\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"PetalWidthCm\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.9602083107326242,\n        \"min\":
0.2,\n        \"max\": 2.3,\n        \"num_unique_values\": 4,\n
\"samples\": [\n          0.4,\n          0.2,\n          2.3\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Species\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 2,\n        \"samples\": [\n          \"Iris-
setosa\",\n          \"Iris-virginica\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe"}
```


```
iris['Species'] = iris['Species'].replace({'Iris-setosa':0,'Iris-
versicolor':1,'Iris-virginica':2})
iris.sample(5)
```

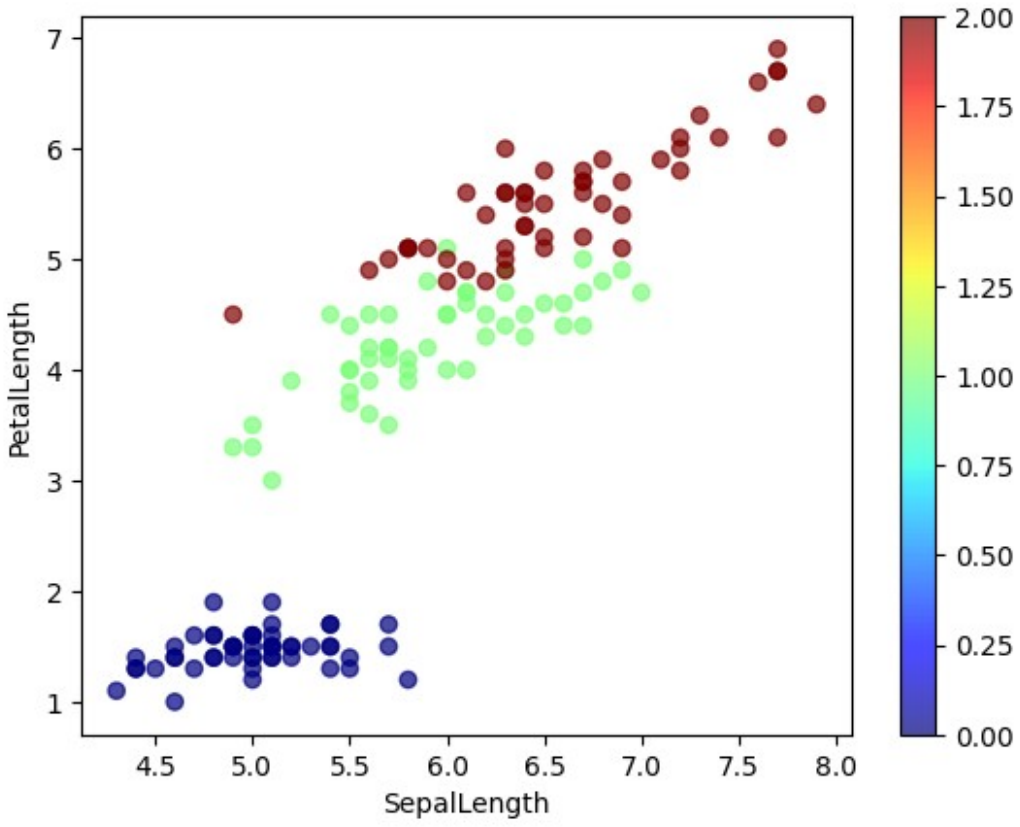
```
<ipython-input-3-c7d964f8b483>:1: FutureWarning: Downcasting behavior
in `replace` is deprecated and will be removed in a future version. To
```

```
retain the old behavior, explicitly call
`result.infer_objects(copy=False)`. To opt-in to the future behavior,
set `pd.set_option('future.no_silent_downcasting', True)`
  iris['Species'] = iris['Species'].replace({'Iris-setosa':0,'Iris-
versicolor':1,'Iris-virginica':2})
```


```
{"summary":"{\n  \"name\": \"iris\",\n  \"rows\": 5,\n  \"fields\": [\
n    {\n      \"column\": \"Id\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 56,\n        \"min\": 7,\n
\"max\": 140,\n        \"num_unique_values\": 5,\n        \"samples\":
[\n          7,\n          140,\n          41\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"SepalLengthCm\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1.2275992831539126,\n        \"min\": 4.6,\n        \"max\": 7.2,\n
\"num_unique_values\": 5,\n        \"samples\": [\n          4.6,\n
6.9,\n          5.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"SepalWidthCm\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.43243496620879307,\n        \"min\":
2.4,\n        \"max\": 3.5,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          3.4,\n          3.1,\n          3.5\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"PetalLengthCm\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
2.1879213879844954,\n        \"min\": 1.3,\n        \"max\": 6.0,\n
\"num_unique_values\": 5,\n        \"samples\": [\n          1.4,\n
5.4,\n          1.3\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"PetalWidthCm\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.8336666000266534,\n        \"min\":
0.3,\n        \"max\": 2.1,\n        \"num_unique_values\": 4,\n
\"samples\": [\n          0.3,\n          2.1,\n          1.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Species\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1,\n        \"min\": 0,\n        \"max\": 2,\n
\"num_unique_values\": 3,\n        \"samples\": [\n          1,\n
0,\n          2\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```

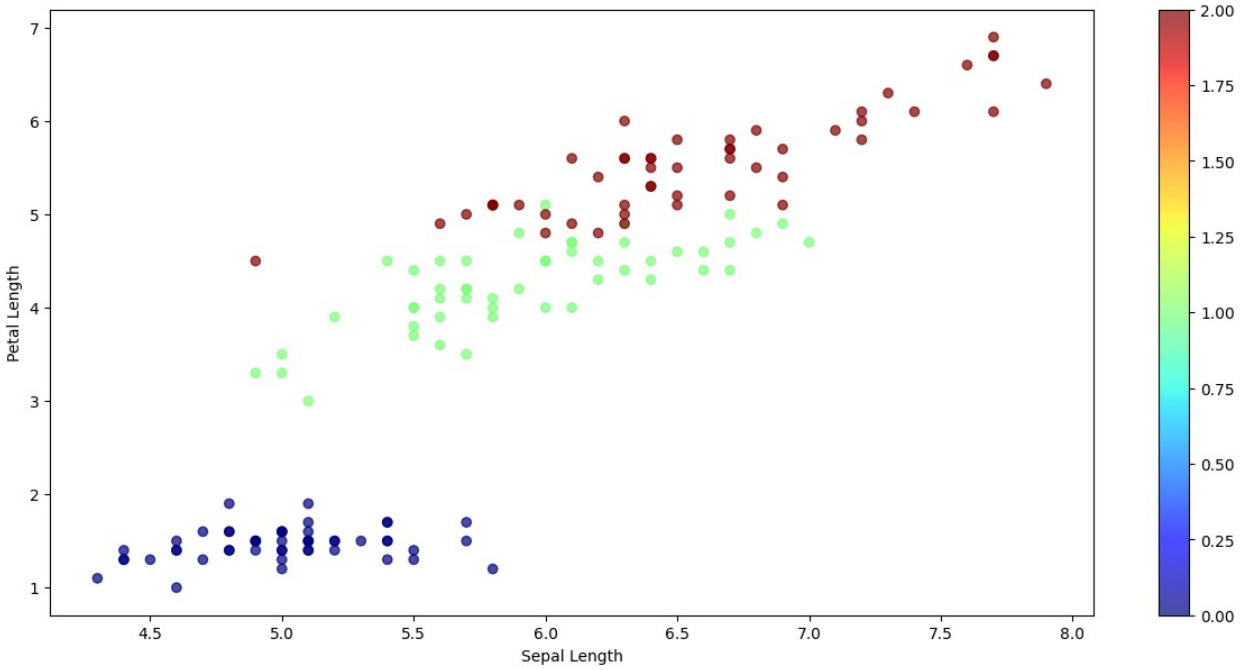

```
plt.scatter(iris['SepalLengthCm'],iris['PetalLengthCm'],c=iris['Specie
s'],cmap='jet',alpha=0.7)
plt.xlabel('SepalLength')
plt.ylabel('PetalLength')
plt.colorbar()
```

```
<matplotlib.colorbar.Colorbar at 0x7e4c973b22d0>
```




```
#plot size
plt.figure(figsize=(15,7))

plt.scatter(iris['SepalLengthCm'],iris['PetalLengthCm'],c=iris['Species'],cmap='jet',alpha=0.7)
plt.xlabel('Sepal Length')
plt.ylabel('Petal Length')
plt.colorbar()
```

```
<matplotlib.colorbar.Colorbar at 0x7e4c94ee4a10>
```

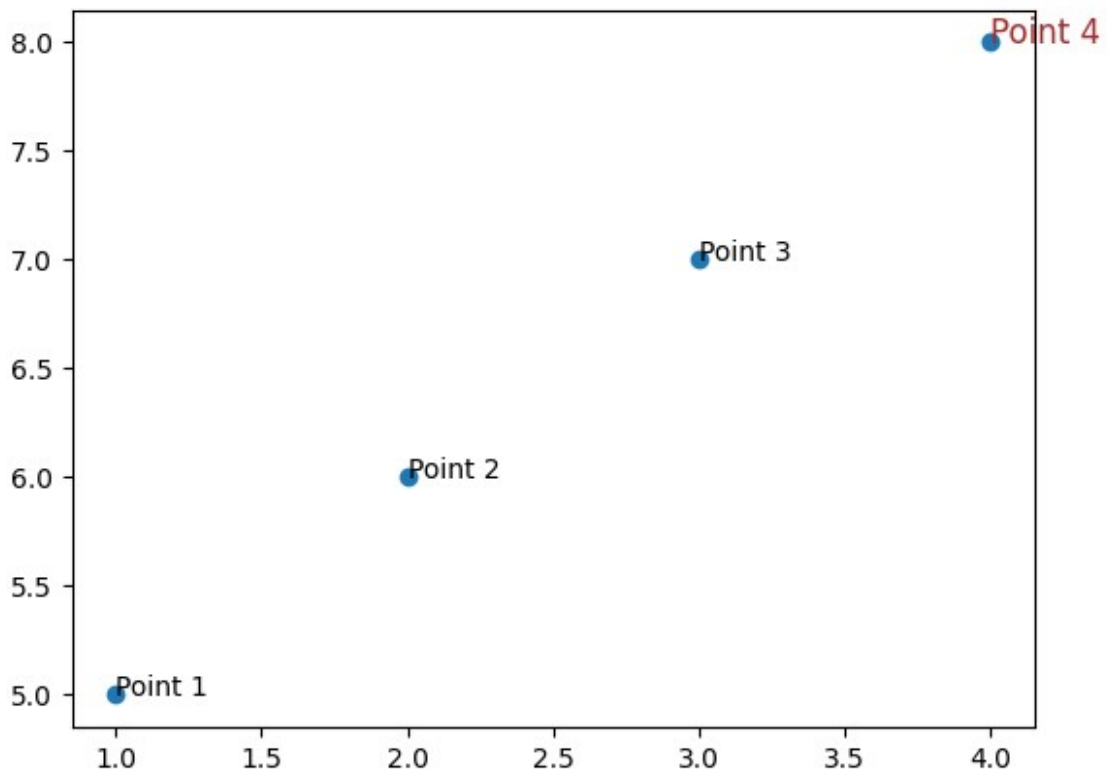



```
batters = pd.read_csv('/content/batter.csv')

batters.shape
```

```
(605, 4)
```

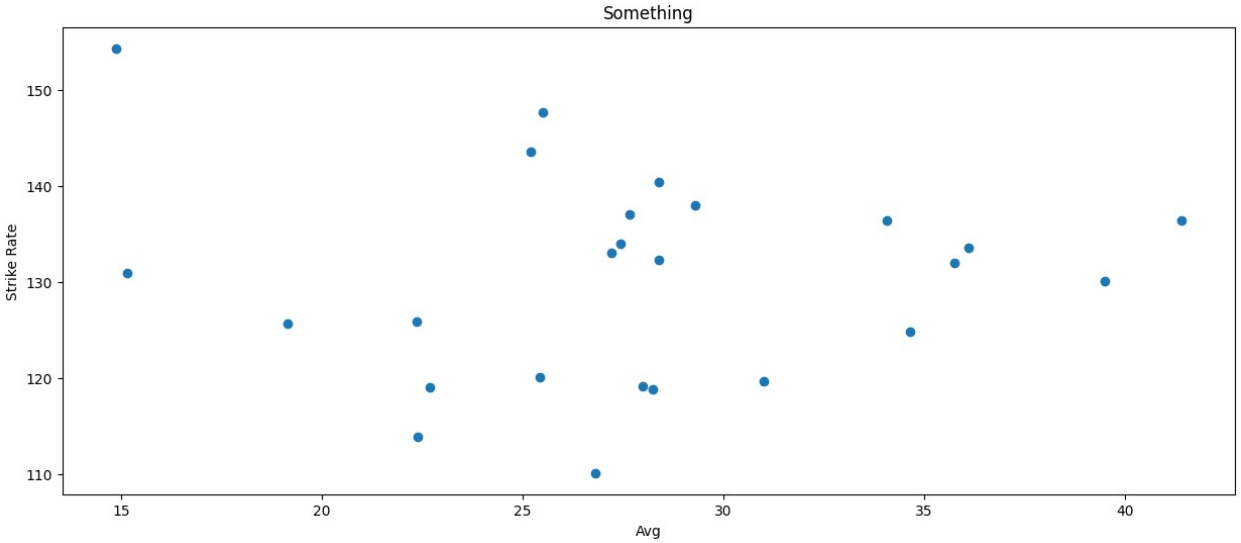
```
x = [1,2,3,4]
y = [5,6,7,8]
plt.scatter(x,y)
plt.text(1,5,'Point 1')
plt.text(2,6,'Point 2')
plt.text(3,7,'Point 3')
plt.text(4,8,'Point 4',fontdict={'size':12,'color':'brown'})
```

```
Text(4, 8, 'Point 4')
```

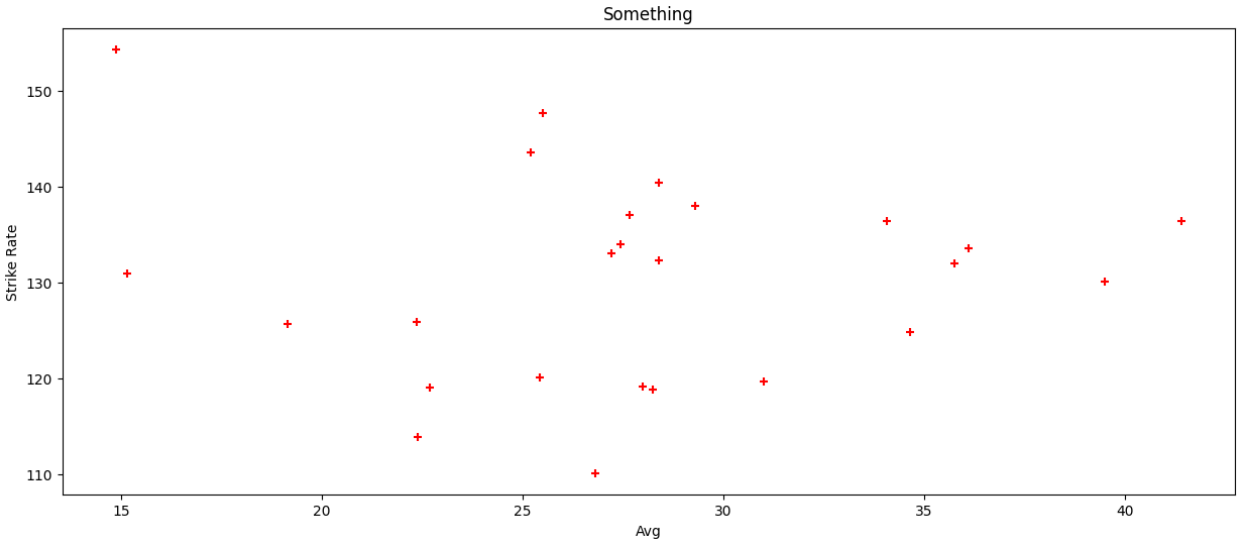


SubPlot

```
# A diff way to plot the graph
plt.figure(figsize=(15,6))
plt.scatter(batters['avg'],batters['strike_rate'])
plt.title('Something')
plt.xlabel('Avg')
plt.ylabel('Strike Rate')

plt.show()
```

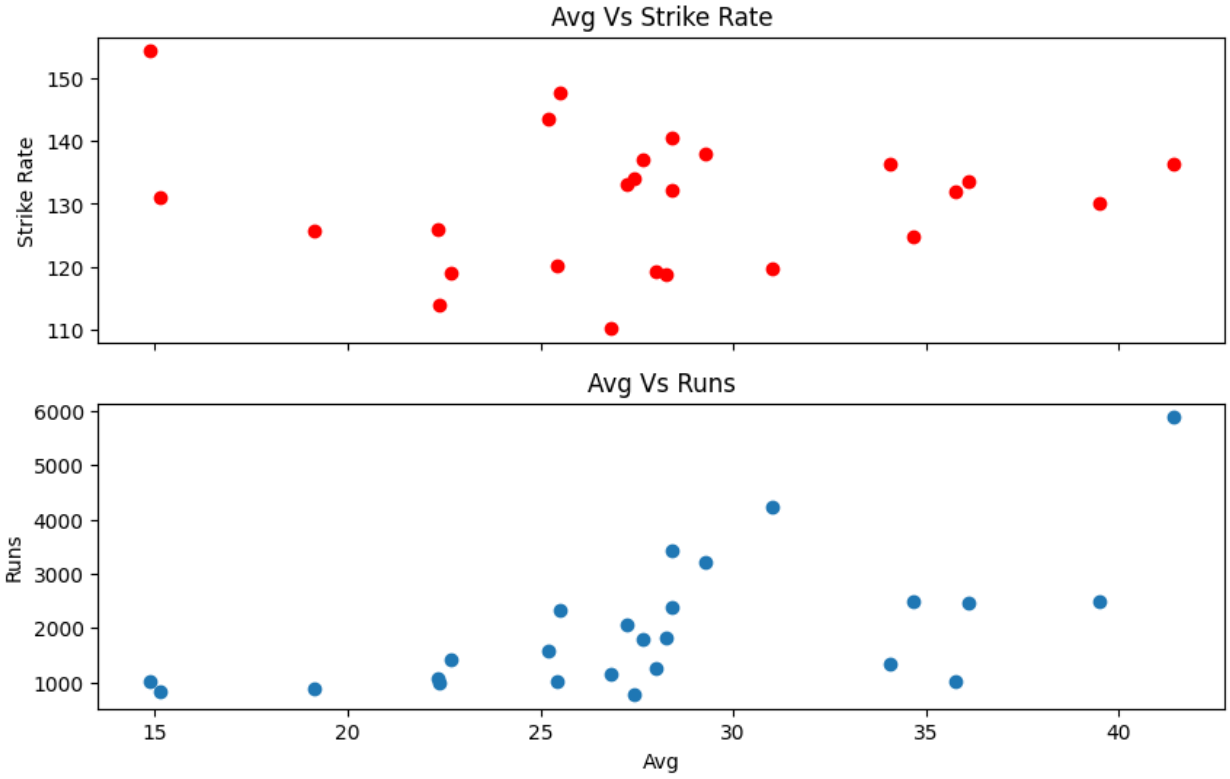



```
fig,ax = plt.subplots(figsize=(15,6))

ax.scatter(batters['avg'],batters['strike_rate'],color='red',marker='+
')
ax.set_title('Something')
ax.set_xlabel('Avg')
ax.set_ylabel('Strike Rate')

fig.show()
```

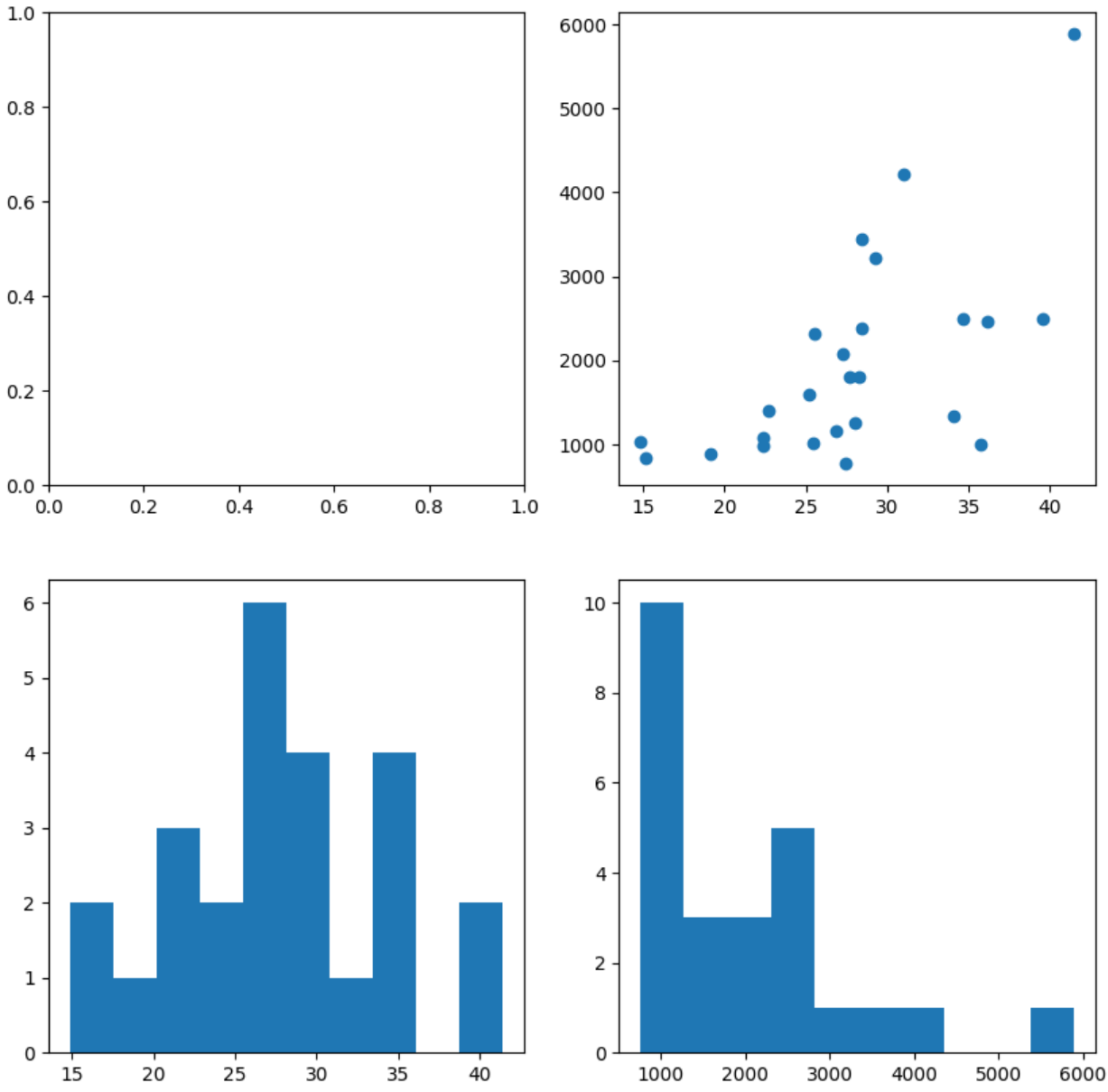



```
fig, ax = plt.subplots(nrows=2,ncols=1,sharex=True,figsize=(10,6))

ax[0].scatter(batters['avg'],batters['strike_rate'],color='red')
ax[1].scatter(batters['avg'],batters['runs'])
```

```
ax[0].set_title('Avg Vs Strike Rate')
ax[0].set_ylabel('Strike Rate')


ax[1].set_title('Avg Vs Runs')
ax[1].set_ylabel('Runs')
ax[1].set_xlabel('Avg')

Text(0.5, 0, 'Avg')
```



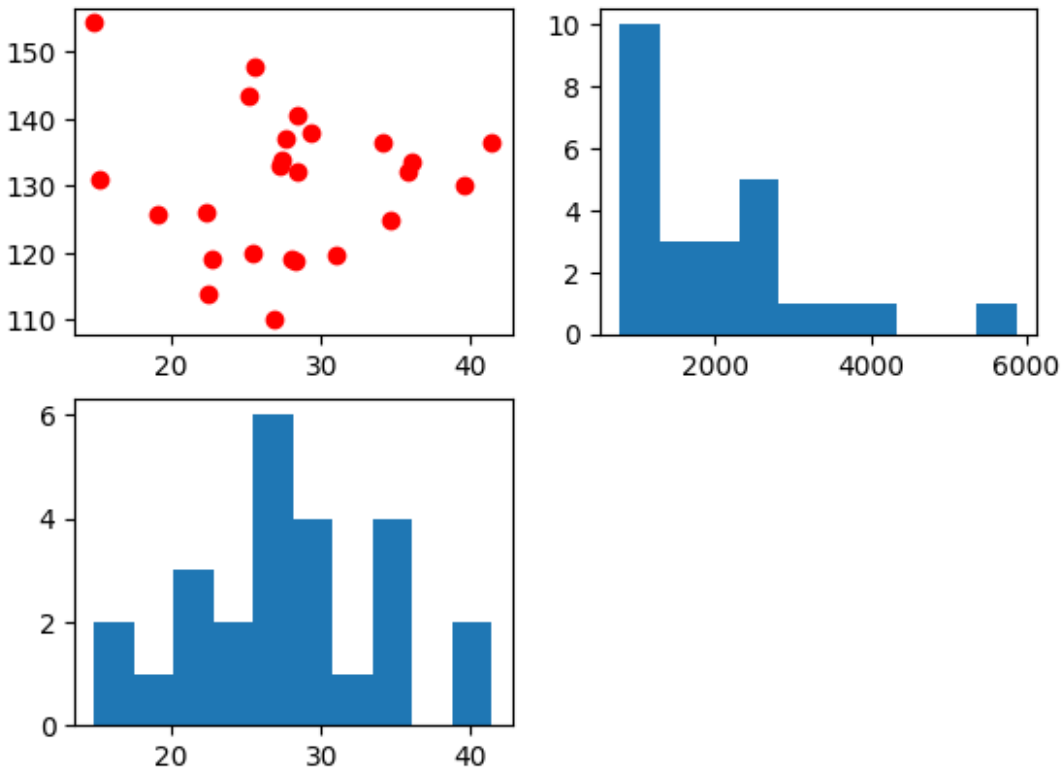

```
fig, ax = plt.subplots(nrows=2,ncols=2,figsize=(10,10))

ax[0,0]
ax[0,1].scatter(batters['avg'],batters['runs'])
ax[1,0].hist(batters['avg'])
ax[1,1].hist(batters['runs'])

(array([10.,  3.,  3.,  5.,  1.,  1.,  1.,  0.,  0.,  1.]),
 array([ 768. , 1279.5, 1791. , 2302.5, 2814. , 3325.5, 3837. ,
4348.5,
        4860. , 5371.5, 5883. ]),
 <BarContainer object of 10 artists>)
```



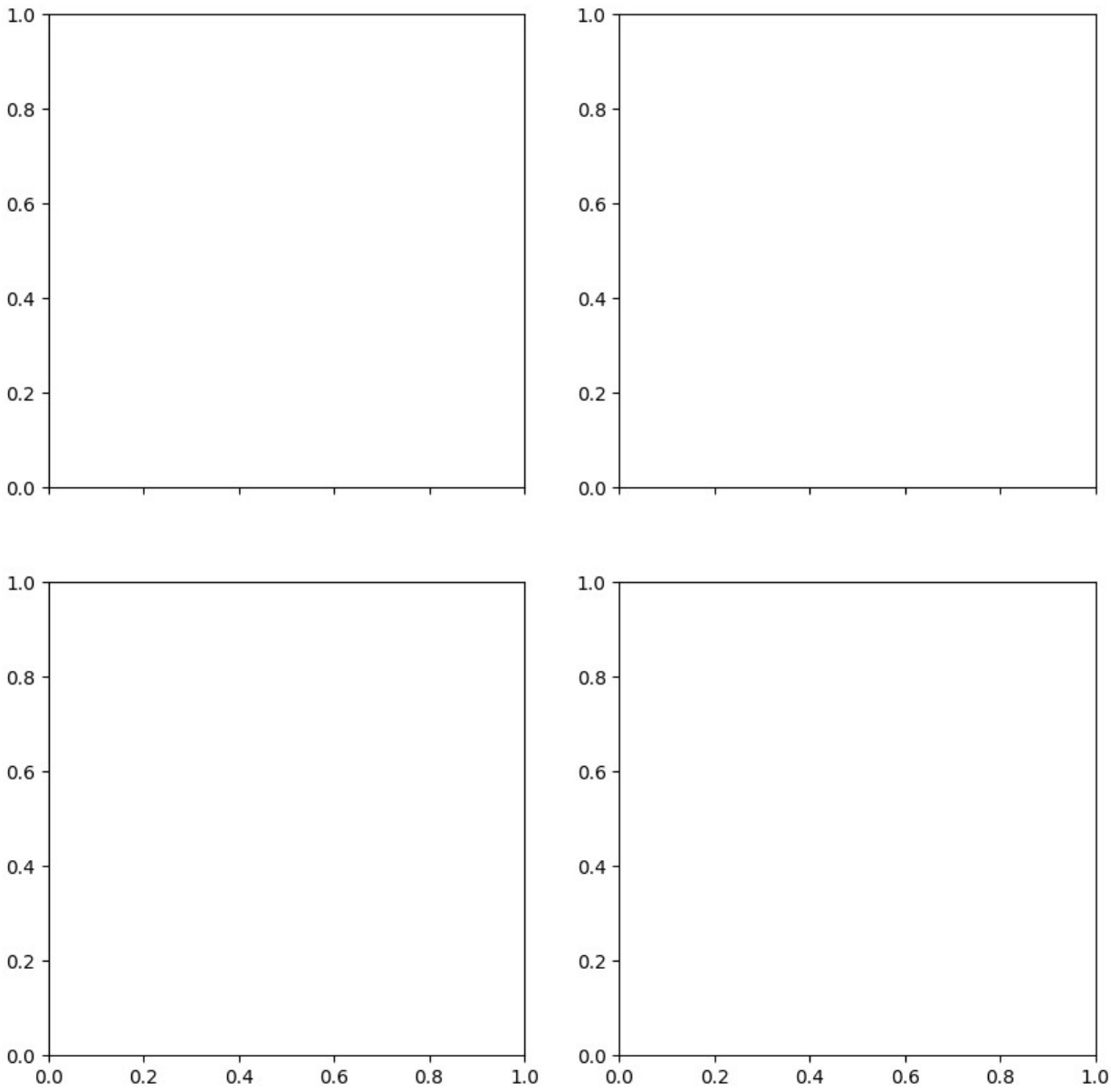

```
fig = plt.figure()

ax1 = fig.add_subplot(2,2,1)
ax1.scatter(batters['avg'],batters['strike_rate'],color='red')

ax2 = fig.add_subplot(2,2,2)
ax2.hist(batters['runs'])

ax3 = fig.add_subplot(2,2,3)
ax3.hist(batters['avg'])
```

```
(array([2., 1., 3., 2., 6., 4., 1., 4., 0., 2.]),
 array([14.85507246, 17.51252296, 20.16997346, 22.82742396,
```

```
25.48487446,
        28.14232496, 30.79977546, 33.45722596, 36.11467646,
38.77212696,
        41.42957746]),
 <BarContainer object of 10 artists>)
```




```
fig, ax = plt.subplots(nrows=2,ncols=2,sharex=True,figsize=(10,10))

ax[1,1]

<Axes: >
```

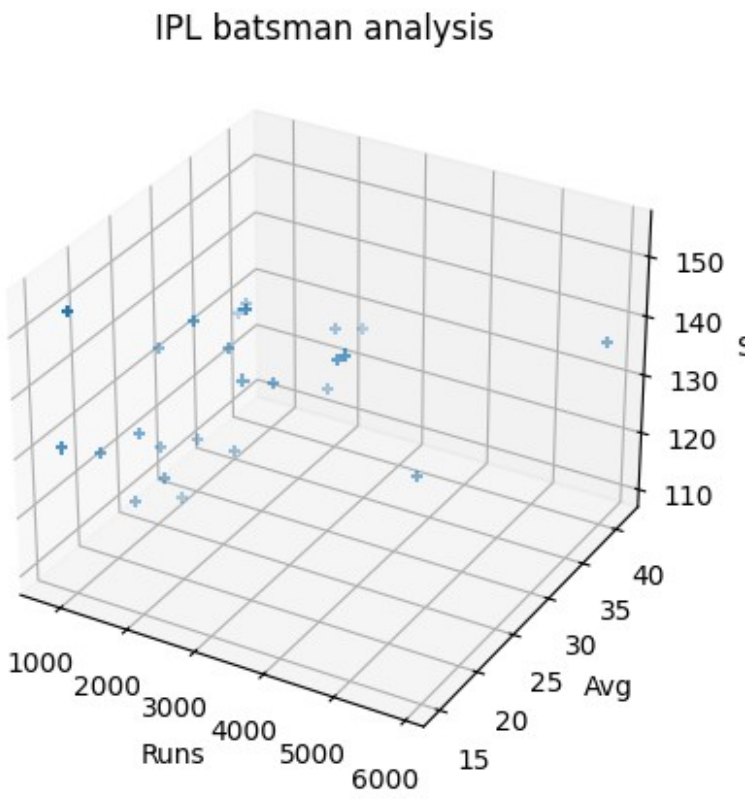


3D Scatter Plots

```
batters

fig = plt.figure()

ax = plt.subplot(projection='3d')

ax.scatter3D(batters['runs'],batters['avg'],batters['strike_rate'],mar
ker='+')
ax.set_title('IPL batsman analysis')

ax.set_xlabel('Runs')
```

```
ax.set_ylabel('Avg')
ax.set_zlabel('SR')
```

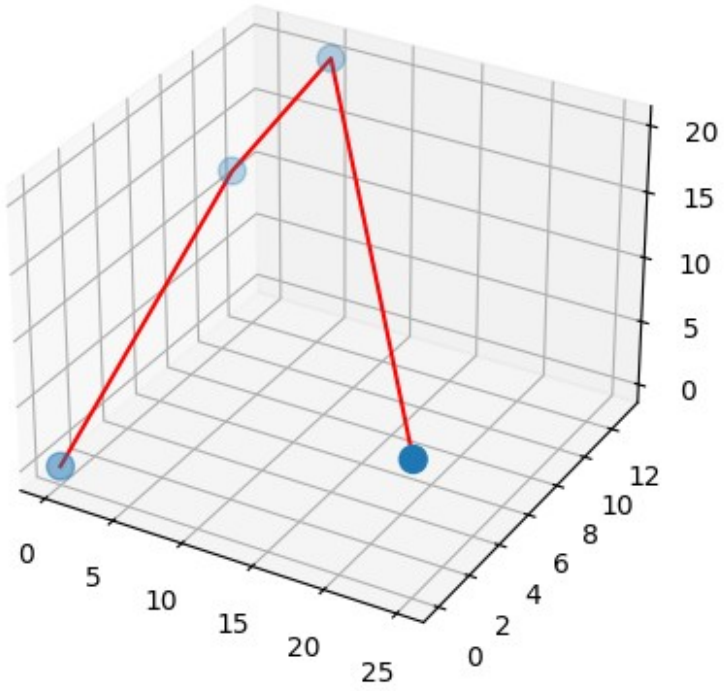
```
Text(0.5, 0, 'SR')
```



3D Line Plot

```
x = [0,1,5,25]
y = [0,10,13,0]
z = [0,13,20,9]

fig = plt.figure()

ax = plt.subplot(projection='3d')

ax.scatter3D(x,y,z,s=[100,100,100,100])
ax.plot3D(x,y,z,color='red')
```

```
[<mpl_toolkits.mplot3d.art3d.Line3D at 0x7e4c8a00c810>]
```



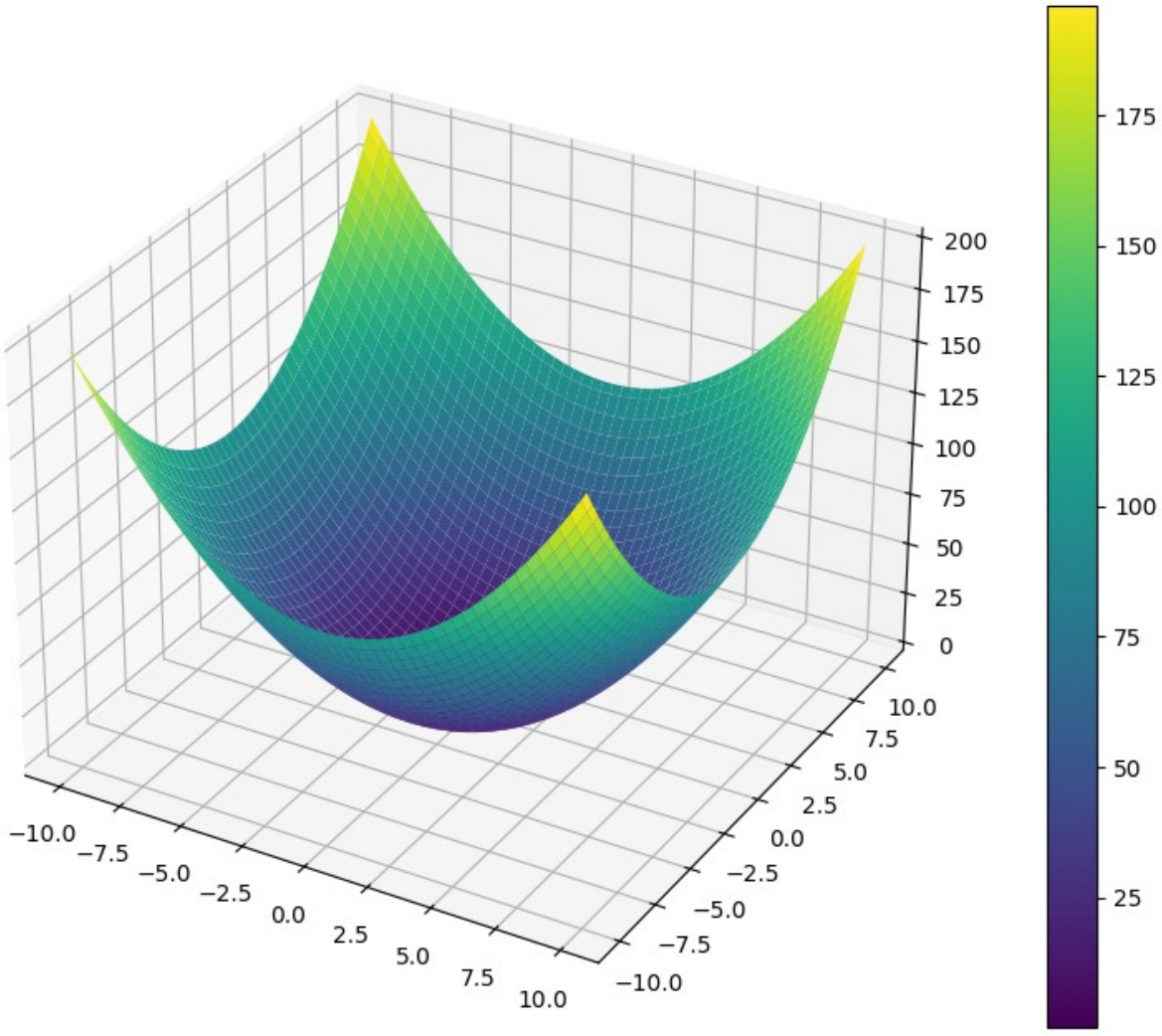
3D Surface Plot

```
x = np.linspace(-10,10,100)
y = np.linspace(-10,10,100)

xx, yy = np.meshgrid(x, y)
z = xx**2 + yy**2
z.shape
```

```
(100, 100)
```

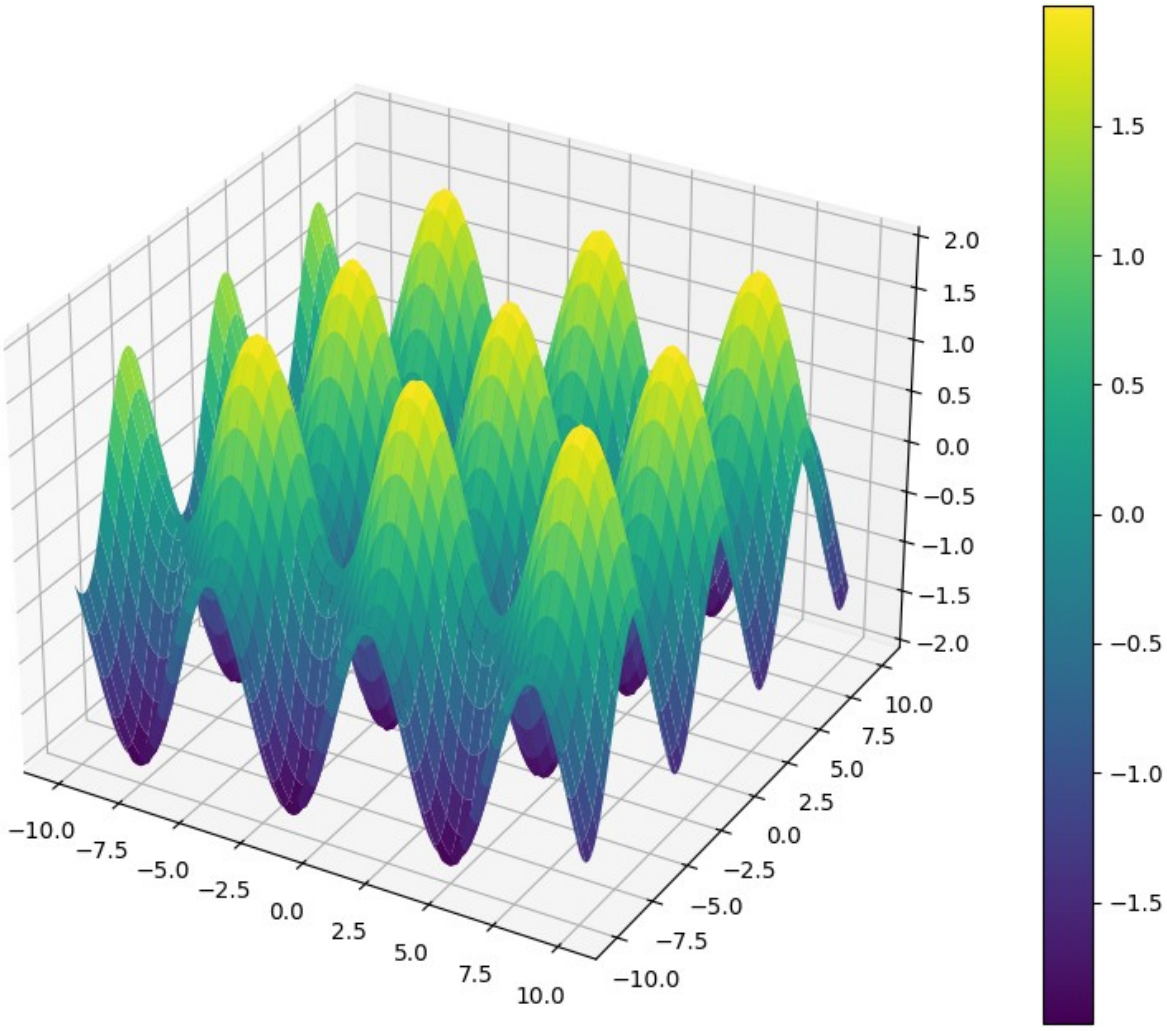
```
fig = plt.figure(figsize=(12,8))

ax = plt.subplot(projection='3d')

p = ax.plot_surface(xx,yy,z,cmap='viridis')
fig.colorbar(p)
```

```
<matplotlib.colorbar.Colorbar at 0x7e4c88ab1fd0>
```



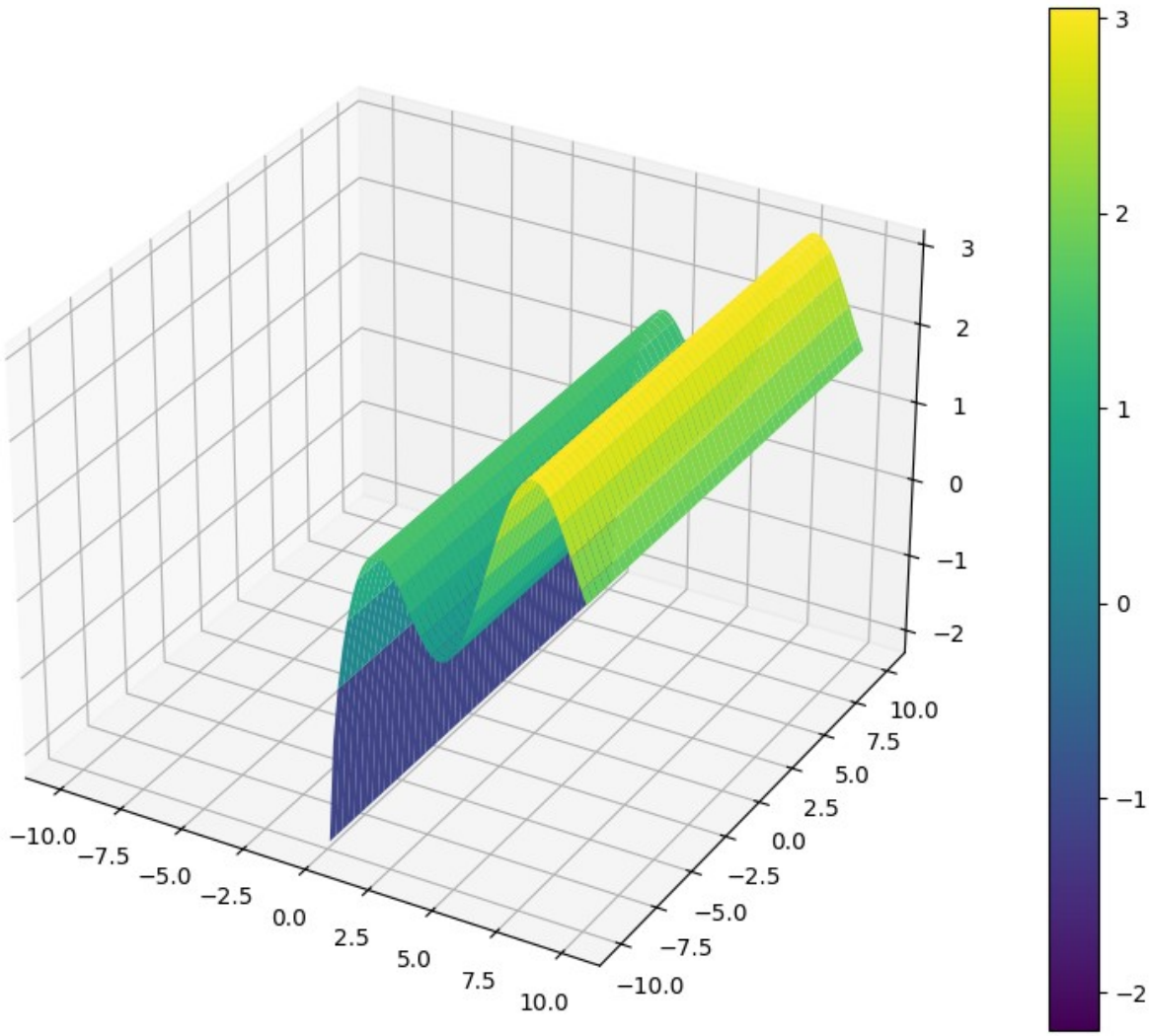

```
z = np.sin(xx) + np.cos(yy)

fig = plt.figure(figsize=(12,8))

ax = plt.subplot(projection='3d')

p = ax.plot_surface(xx,yy,z,cmap='viridis')
fig.colorbar(p)
```

```
<matplotlib.colorbar.Colorbar at 0x7e4c88898bd0>
```



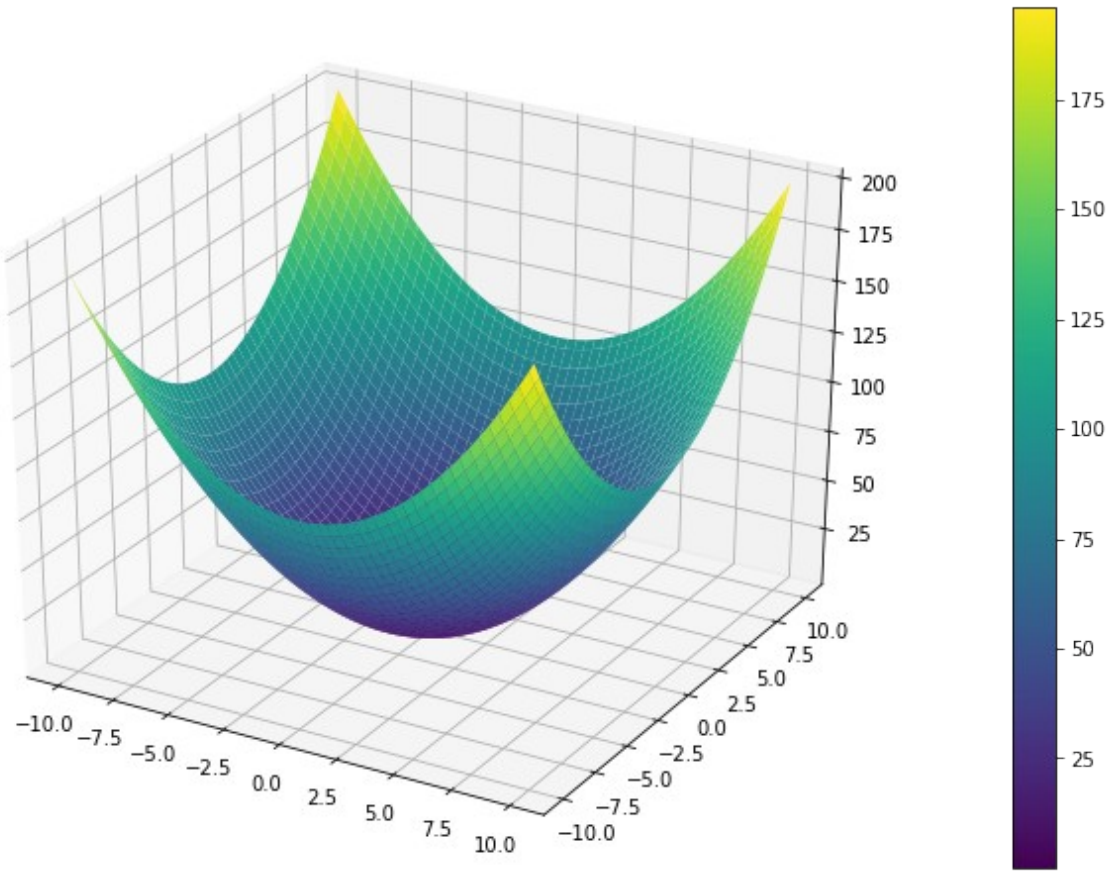

```
z = np.sin(xx) + np.log(xx)

fig = plt.figure(figsize=(12,8))

ax = plt.subplot(projection='3d')

p = ax.plot_surface(xx,yy,z,cmap='viridis')
fig.colorbar(p)
```

```
<ipython-input-37-bbcd37ea4152>:1: RuntimeWarning: invalid value
encountered in log
  z = np.sin(xx) + np.log(xx)

<matplotlib.colorbar.Colorbar at 0x7e4c88426750>
```




```
fig = plt.figure(figsize=(12,8))

ax = plt.subplot(projection='3d')

p = ax.plot_surface(xx,yy,z,cmap='viridis')
fig.colorbar(p)
```

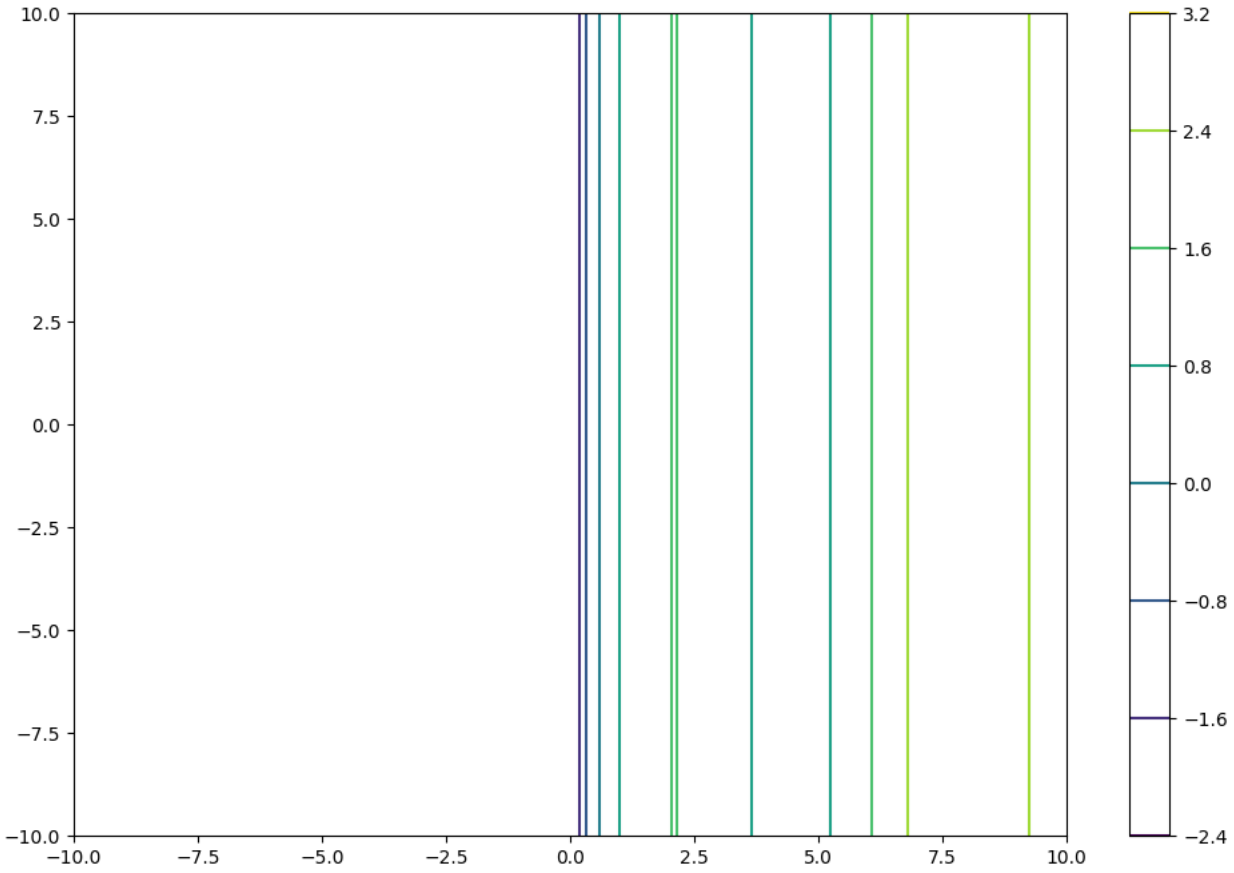
```
<matplotlib.colorbar.Colorbar at 0x7f5e136f8970>
```



Contour Plot

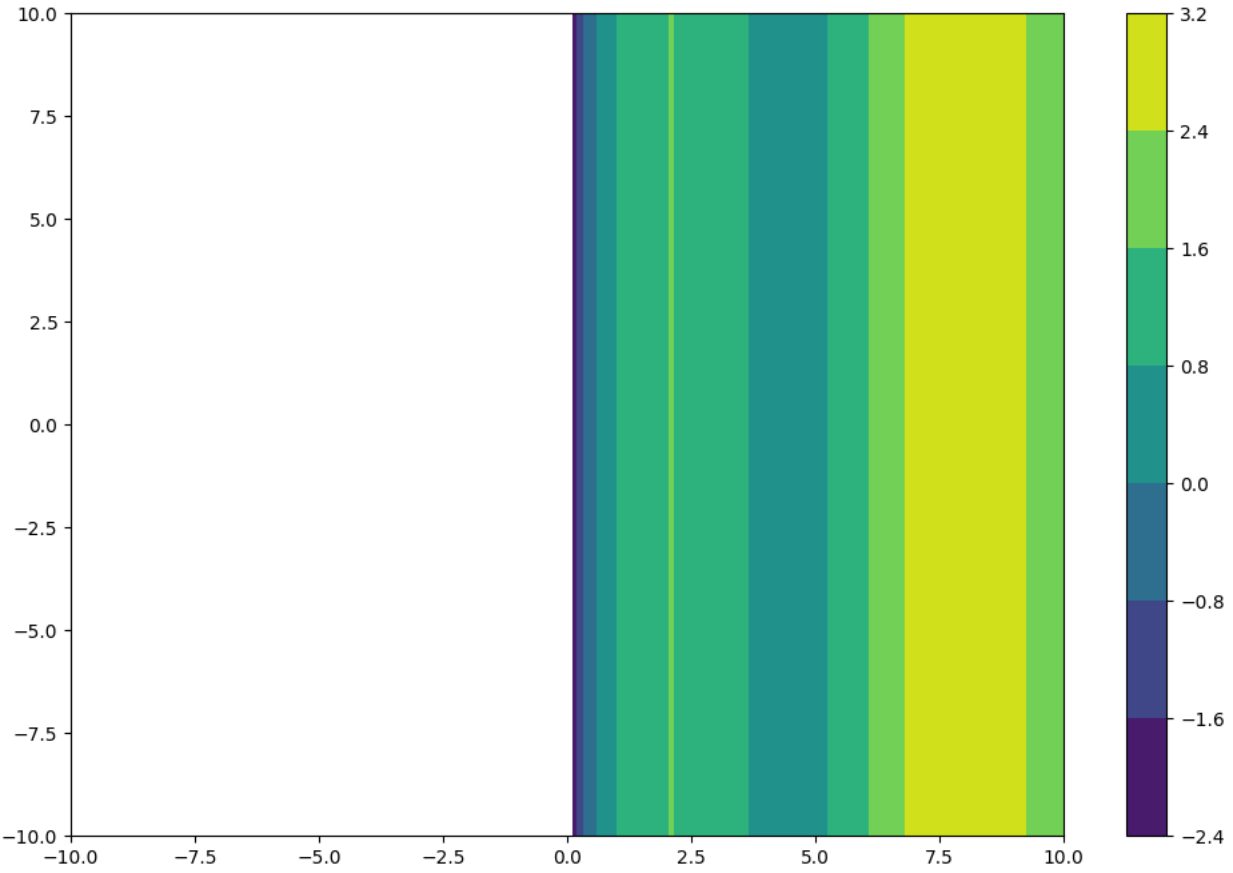
```
fig = plt.figure(figsize=(12,8))

ax = plt.subplot()

p = ax.contour(xx,yy,z,cmap='viridis')
fig.colorbar(p)
```

```
<matplotlib.colorbar.Colorbar at 0x7e4c83d11610>
```



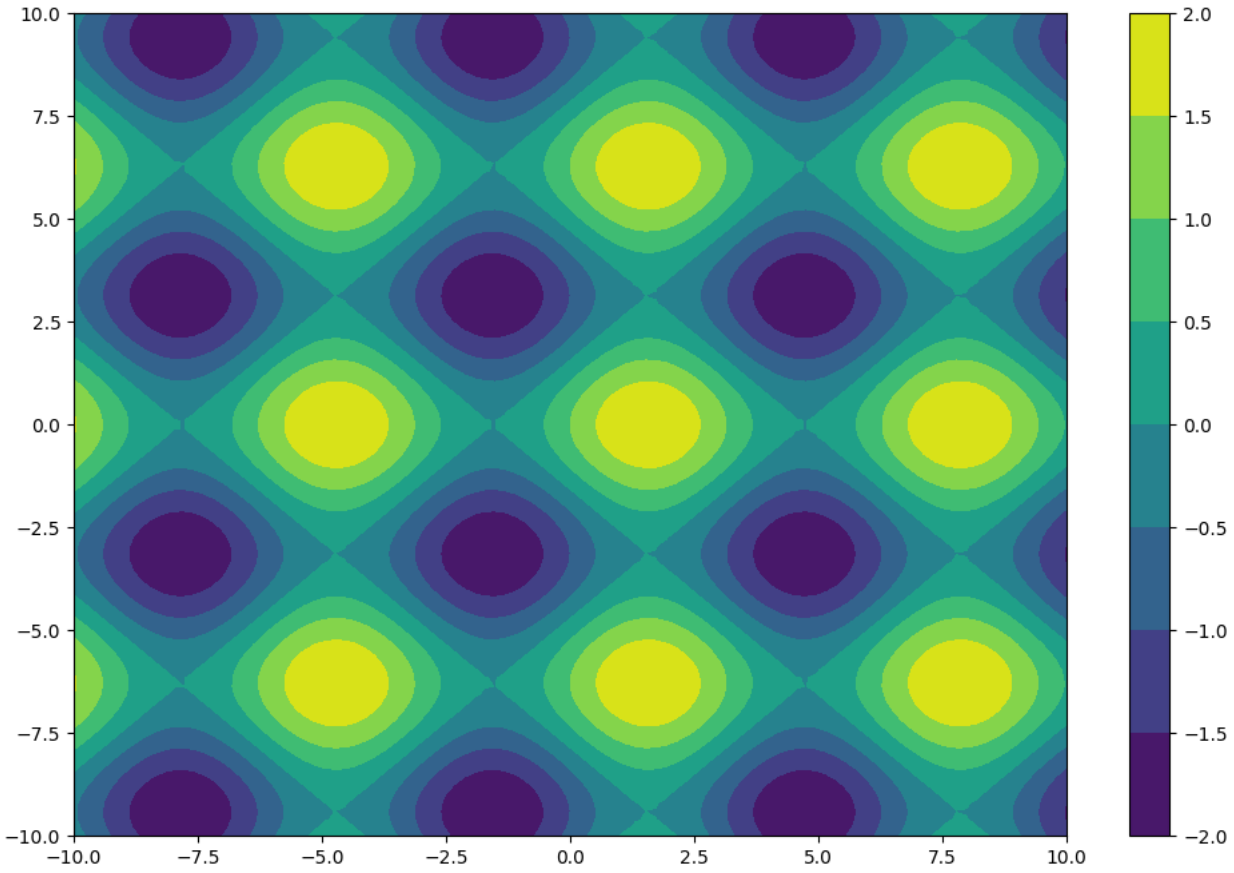

```
fig = plt.figure(figsize=(12,8))

ax = plt.subplot()

p = ax.contourf(xx,yy,z,cmap='viridis')
fig.colorbar(p)
```

```
<matplotlib.colorbar.Colorbar at 0x7e4c83d75010>
```




```
z = np.sin(xx) + np.cos(yy)

fig = plt.figure(figsize=(12,8))

ax = plt.subplot()

p = ax.contourf(xx,yy,z,cmap='viridis')
fig.colorbar(p)
```

```
<matplotlib.colorbar.Colorbar at 0x7e4c83bd1610>
```


10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0