```python
import numpy as np
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
from sklearn.datasets import make_circles
import pandas as pd
```

```python
data = pd.read_csv("/content/heart.csv")
data.head()
```
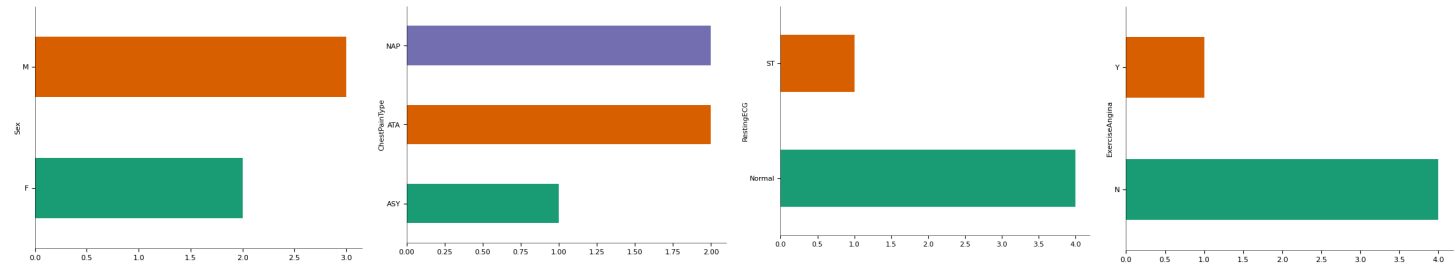
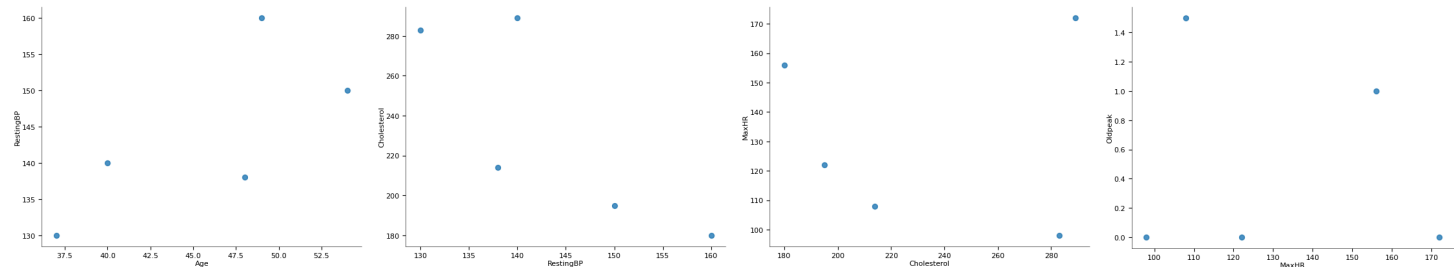|	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
0	40	M	ATA	140	289	0	Normal	172	N	0.0	Up	0
1	49	F	NAP	160	180	0	Normal	156	N	1.0	Flat	1
2	37	M	ATA	130	283	0	ST	98	N	0.0	Up	0
3	48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
4	54	M	NAP	150	195	0	Normal	122	N	0.0	Up	0

Distributions
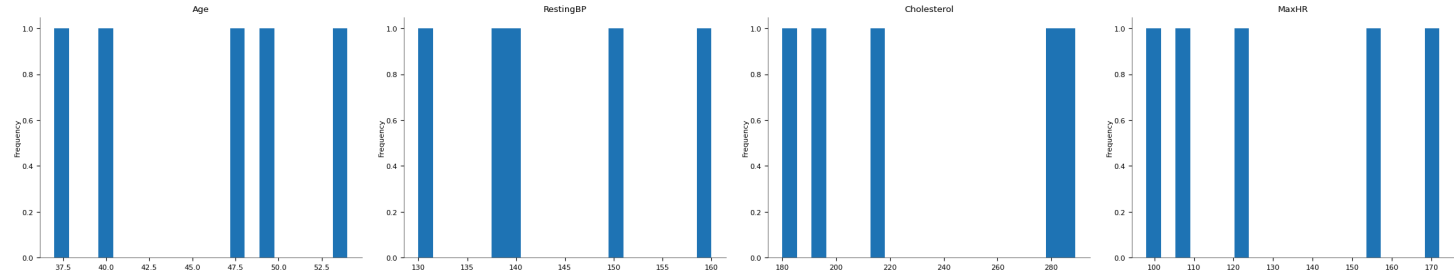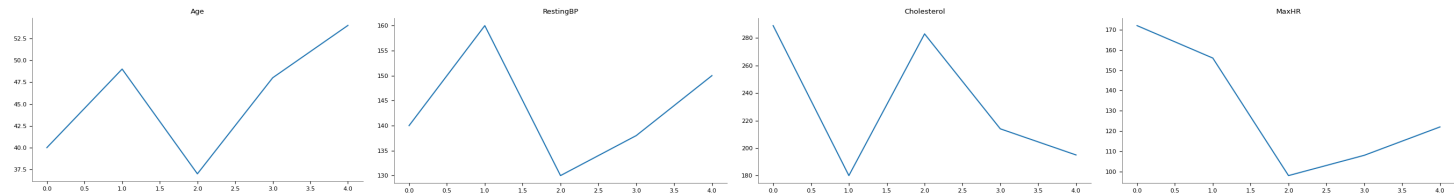

Categorical distributions

2-d distributions
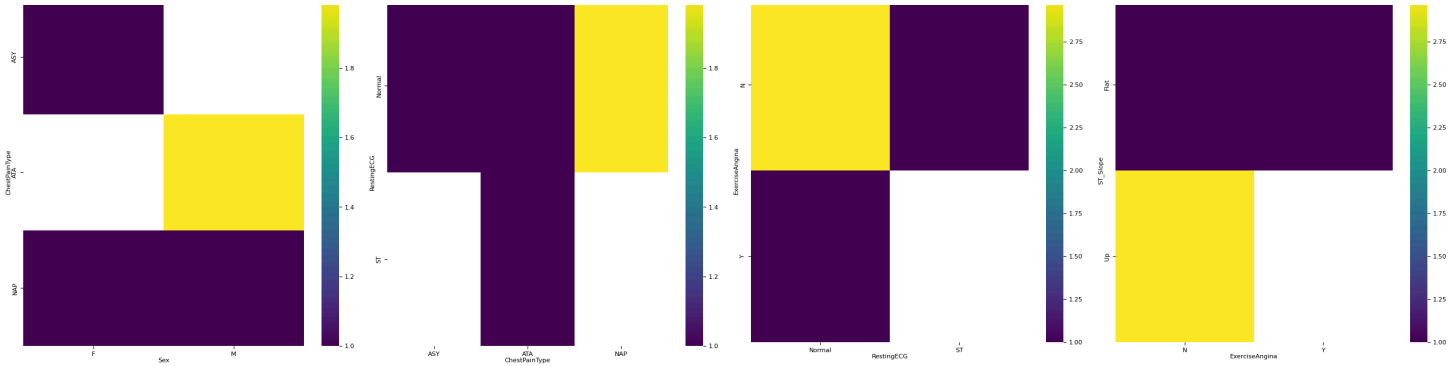

Time series
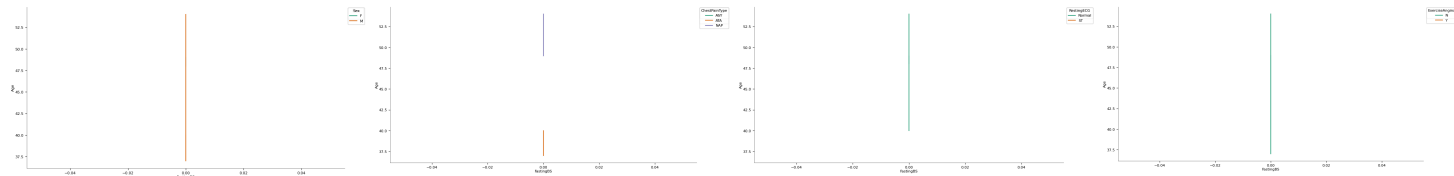
Values

2-d categorical distributions

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 set `legend=False` for the same effect.
```



```
<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 set `legend=False` for the same effect.
```



```
<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 set `legend=False` for the same effect.
```
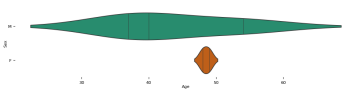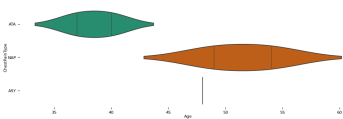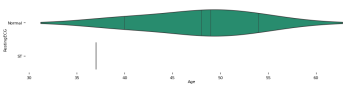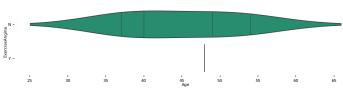


```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.
```


Next steps: [Generate code with `data`]   [⬤ View recommended plots]   [New interactive sheet]
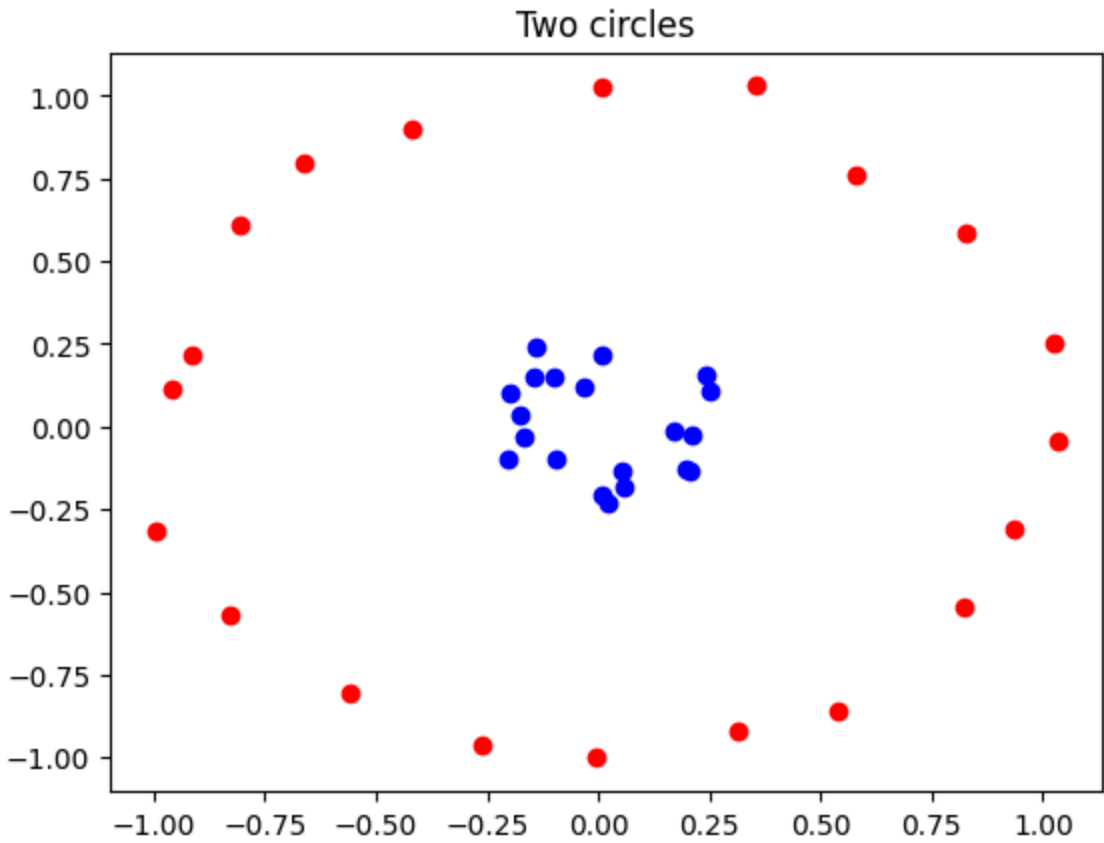
```python
from sklearn.datasets import make_circles
```

```python
x = data[['RestingBP']]
y = data['Cholesterol']
```

```python
x,y = make_circles(n_samples=40,factor=.2,noise=.05,random_state=42)
outer = y == 0
inner = y == 1
```

```python
plt.title("Two circles")
plt.plot(x[outer,0],x[outer,1],"ro")
plt.plot(x[inner,0],x[inner,1],"bo")
```

[<matplotlib.lines.Line2D at 0x7b889df316c0>]

Two circles

```
def sigmoid(x):
    return 1/(1+np.exp(-x))
n_hidden=50
n_epochs=1000
learning_rate=1
weights_hidden=np.random.normal(0.0,size=(x_train.shape[1],n_hidden))
weights_output=np.random.normal(0.0,size=(n_hidden))
hist_loss=[]
hist_accuracy=[]
print(weights_hidden)
print(weights_output)
```

```
[[ 1.90712822  0.86869885  1.21392798  0.04982007  0.58915813 -1.47448382
   1.47332968 -1.43274899  0.47807991  1.77336843 -0.4013564  -1.10825768
   0.48756745 -0.79788027 -1.0508908   1.36303802 -1.26348474  1.59858104
  -0.9825128   0.10036747  0.05412917 -0.25805945  0.51884569  2.89032571
  -1.90315622 -0.48976503 -1.25819474  1.21585369 -0.49000661 -1.40468338
   0.21310681 -0.55199225 -1.8873491   0.02783     2.15123025  0.15236713
   0.09564999  0.72492243  0.1285723   0.07417593  0.67786132 -1.77138657
  -0.71513668  1.16658386  1.27081426  0.61451917 -1.58020509 -0.35028064
  -1.14572644  0.24439568]
 [-0.55791704 -0.49139848  0.45200882 -0.57914961 -2.22766095  0.34984103
   0.39193511 -0.65332889 -0.4558023  -2.04550452  1.53197064  1.51717027
  -0.86916308 -1.82411594  1.20151206  1.52118644  1.5999309  -0.49426779
  -0.44605265 -0.3125577  -0.23470562  1.42010814 -1.82813109  0.64092938
  -0.54613372 -0.33246709  2.08990764  0.21711324  0.36213234 -0.45628796
   0.84273265  0.73192668 -0.61049679  0.62442398 -0.57703308 -0.17177245
```

```
      -0.53386659  1.17117385   0.24874547   2.21066155   0.2346783    0.79565833
       0.80438504  0.11081107   0.30326265   0.24917824   1.56575484 -0.27572649
      -0.59220492  1.16750689]
     [ 0.75200587  0.95569047   1.51640241 -1.617486     -1.14918521 -2.21633659
       0.75232901 -1.25728779   0.86339257 -0.63445907  -0.32230855 -0.01124041
      -0.39664214  1.10725318   2.28314255   0.20058233  -0.47001475 -0.6892855
       0.48216985 -1.19829677   1.12034033 -1.48105457  -0.24991913  1.10898004
      -0.79120365  0.4568618    0.29312104 -0.54065603  -1.11012512  0.05799708
       1.24634031 -0.01256391   0.02082673 -0.01843856   0.43514648 -0.34518698
       0.23433439 -0.19845976   2.60939724 -1.66597354   0.60707015  0.97740898
       1.22049937 -0.06062093   0.29283569  0.29901313   0.65807298  1.51353355
```