```
import pandas as pd
import seaborn as sns
```

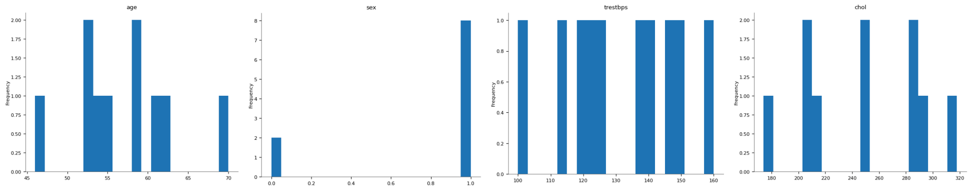
```
dt=pd.read_csv('/content/heart.csv')
```

+ Code + Text

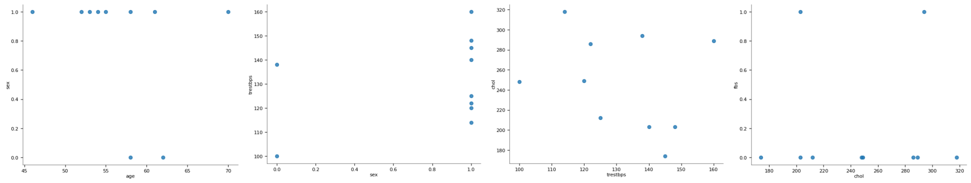
```
dt.head(10)
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
5	58	0	0	100	248	0	0	122	0	1.0	1	0	2	1
6	58	1	0	114	318	0	2	140	0	4.4	0	3	1	0
7	55	1	0	160	289	0	0	145	1	0.8	1	1	3	0
8	46	1	0	120	249	0	0	144	0	0.8	2	0	3	0
9	54	1	0	122	286	0	0	116	1	3.2	1	2	2	0

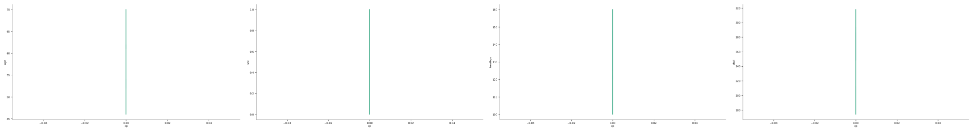
Distributions



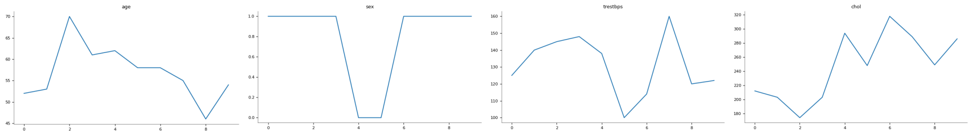
2-d distributions



Time series



Values



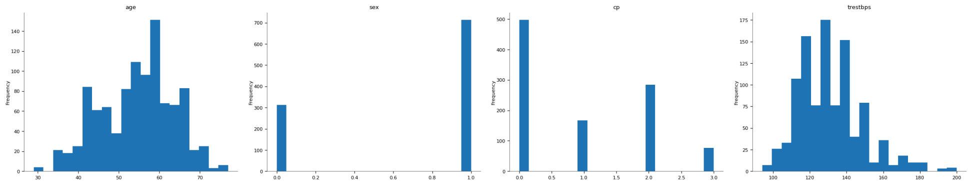
Next steps: Generate code with dt View recommended plots New interactive sheet

```
dt
```

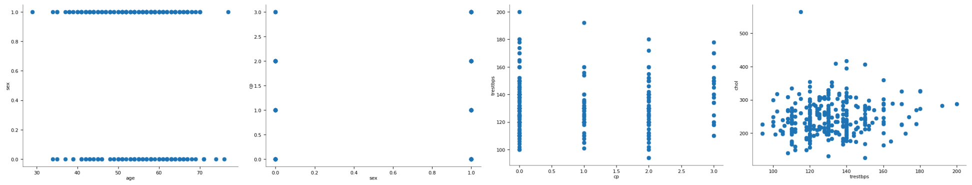
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

1025 rows × 14 columns

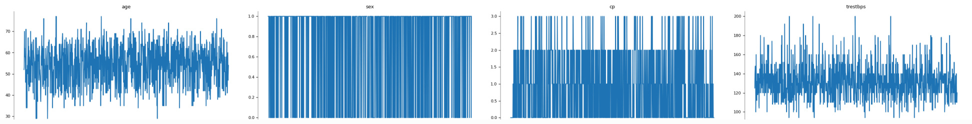
Distributions



2-d distributions



Values



Next steps: Generate code with dt | View recommended plots | New interactive sheet

```
dt.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1025 non-null   int64  
 1   sex       1025 non-null   int64  
 2   cp        1025 non-null   int64  
 3   trestbps  1025 non-null   int64  
 4   chol      1025 non-null   int64  
 5   fbs       1025 non-null   int64  
 6   restecg   1025 non-null   int64  
 7   thalach   1025 non-null   int64  
 8   exang     1025 non-null   int64  
 9   oldpeak   1025 non-null   float64
 10  slope     1025 non-null   int64  
 11  ca        1025 non-null   int64  
 12  thal      1025 non-null   int64  
 13  target    1025 non-null   int64  
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
```

```
print(dt['target'].unique())
```

```
[0 1]
```

```
from sklearn.preprocessing import LabelEncoder
```

```
l=LabelEncoder()
```

```
dt['target'] = l.fit_transform(dt['target'])
```

```
dt['target'].unique()
```

```
array([0, 1])
```

```
dt
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

1025 rows × 14 columns

Next steps: Generate code with dt View recommended plots New interactive sheet

```
from sklearn.model_selection import train_test_split
```

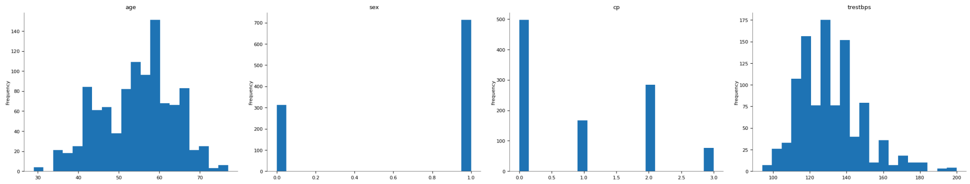
```
x = dt.drop('target', axis=1)
y = dt['target']
```

```
x
```

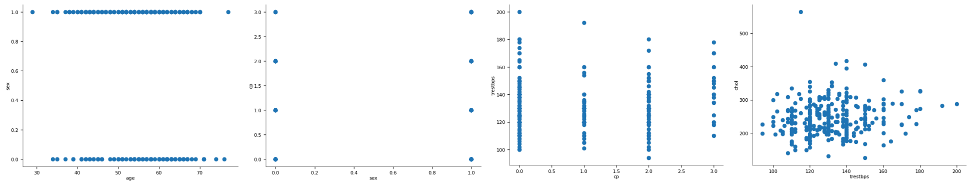
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3

1025 rows × 13 columns

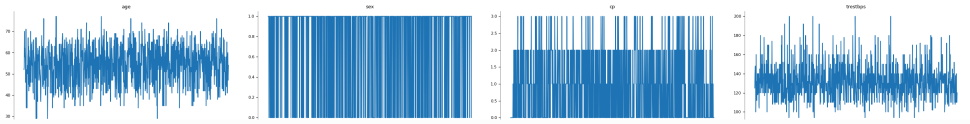
Distributions



2-d distributions



Values



Next steps: Generate code with x | View recommended plots | New interactive sheet

```
y
```

	target
0	0
1	0
2	0
3	0
4	0
...	...
1020	1
1021	0
1022	0
1023	1
1024	0

1025 rows × 1 columns

```
dt.corr()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca
age	1.000000	-0.103240	-0.071966	0.271121	0.219823	0.121243	-0.132696	-0.390227	0.088163	0.208137	-0.169105	0.271551
sex	-0.103240	1.000000	-0.041119	-0.078974	-0.198258	0.027200	-0.055117	-0.049365	0.139157	0.084687	-0.026666	0.111729
cp	-0.071966	-0.041119	1.000000	0.038177	-0.081641	0.079294	0.043581	0.306839	-0.401513	-0.174733	0.131633	-0.176206
trestbps	0.271121	-0.078974	0.038177	1.000000	0.127977	0.181767	-0.123794	-0.039264	0.061197	0.187434	-0.120445	0.104554
chol	0.219823	-0.198258	-0.081641	0.127977	1.000000	0.026917	-0.147410	-0.021772	0.067382	0.064880	-0.014248	0.074259
fbs	0.121243	0.027200	0.079294	0.181767	0.026917	1.000000	-0.104051	-0.008866	0.049261	0.010859	-0.061902	0.137156
restecg	-0.132696	-0.055117	0.043581	-0.123794	-0.147410	-0.104051	1.000000	0.048411	-0.065606	-0.050114	0.086086	-0.078072
thalach	-0.390227	-0.049365	0.306839	-0.039264	-0.021772	-0.008866	0.048411	1.000000	-0.380281	-0.349796	0.395308	-0.207888
exang	0.088163	0.139157	-0.401513	0.061197	0.067382	0.049261	-0.065606	-0.380281	1.000000	0.310844	-0.267335	0.107849
oldpeak	0.208137	0.084687	-0.174733	0.187434	0.064880	0.010859	-0.050114	-0.349796	0.310844	1.000000	-0.575189	0.221816
slope	-0.169105	-0.026666	0.131633	-0.120445	-0.014248	-0.061902	0.086086	0.395308	-0.267335	-0.575189	1.000000	-0.073440
ca	0.271551	0.111729	-0.176206	0.104554	0.074259	0.137156	-0.078072	-0.207888	0.107849	0.221816	-0.073440	1.000000
thal	0.072297	0.198424	-0.163341	0.059276	0.100244	-0.042177	-0.020504	-0.098068	0.197201	0.202672	-0.094090	0.149014
target	-0.229324	-0.279501	0.434854	-0.138772	-0.099966	-0.041164	0.134468	0.422895	-0.438029	-0.438441	0.345512	-0.382085

```
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.25)
```

```
xtest
```

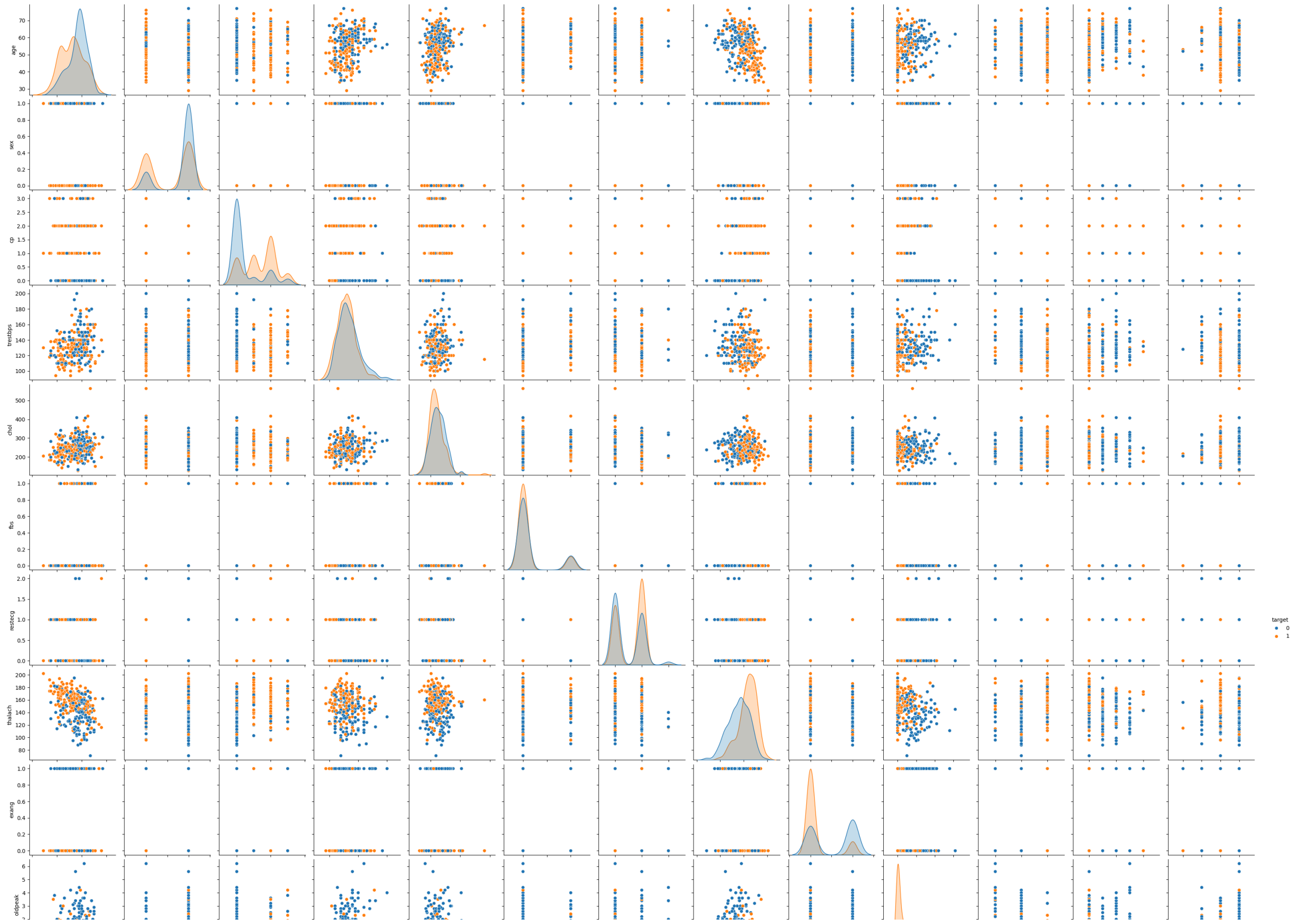
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
675	58	1	0	100	234	0	1	156	0	0.1	2	1	3
302	55	0	1	132	342	0	1	166	0	1.2	2	0	2
97	53	1	0	123	282	0	1	95	1	2.0	1	2	3
322	45	1	0	142	309	0	0	147	1	0.0	1	3	3
8	46	1	0	120	249	0	0	144	0	0.8	2	0	3
...	...	...	...	...	...	...	...	...	...	...	...	...	...
749	58	1	1	125	220	0	1	144	0	0.4	1	4	3
95	45	0	0	138	236	0	0	152	1	0.2	1	0	2
445	52	1	1	128	205	1	1	184	0	0.0	2	0	2
278	55	1	0	160	289	0	0	145	1	0.8	1	1	3
894	51	1	0	140	299	0	1	173	1	1.6	2	0	3

257 rows × 13 columns

Next steps: Generate code with xtest View recommended plots New interactive sheet

```
sns.pairplot(dt,hue='target')
```

```
<seaborn.axisgrid.PairGrid at 0x7bf89fbce810>
```



```
1 from sklearn.linear_model import LogisticRegression
```

```
1 lr=LogisticRegression()
```

```
1 lr.fit(xtrain,ytrain)
```

```
/usr/local/lib/python3.11/dist-packages/sklearn/linear_model/_logistic.py:465: ConvergenceWarning: lbfgs failed to converge (status=
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
```