

```
In [31]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')

In [32]: data = pd.read_csv('heart.csv')
```

```
In [33]: data.head()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

```
In [34]: data.tail()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

```
In [35]: data.shape
```

Out[35]: (303, 14)

```
In [36]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
age          303 non-null int64
sex          303 non-null int64
cp           303 non-null int64
trestbps     303 non-null int64
chol         303 non-null int64
fbs          303 non-null int64
restecg      303 non-null int64
thalach      303 non-null int64
exang        303 non-null int64
oldpeak      303 non-null float64
slope        303 non-null int64
ca           303 non-null int64
thal         303 non-null int64
target       303 non-null int64
dtypes: float64(1), int64(13)
memory usage: 33.2 KB
```

```
In [37]: data.describe()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.591271
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.058560
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000

```
In [38]: data.columns
```

Out[38]: Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach', 'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'], dtype='object')

```
In [39]: data.nunique()
```

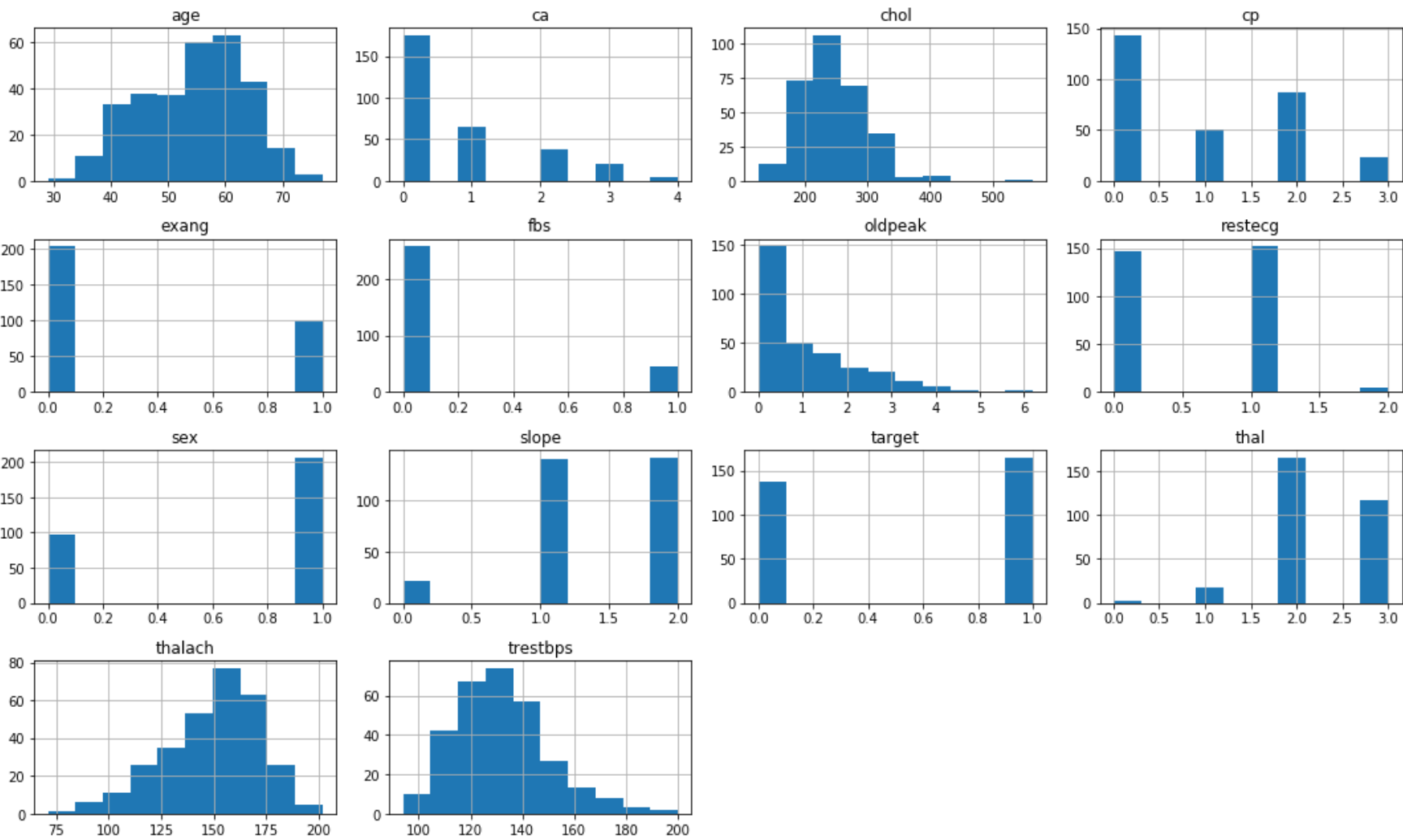
```
age          41
sex           2
cp            4
trestbps     49
chol         152
fbs           2
restecg       3
thalach       91
exang         2
oldpeak       40
slope         3
ca            5
thal          4
target        2
dtype: int64
```

```
In [40]: data.duplicated().sum()
```

Out[40]: 1

## EDA

```
In [41]: fig,ax=plt.subplots(figsize=(15,9))
data.hist(ax=ax)
plt.tight_layout()
plt.show()
```



## Implement the PCA

```
In [42]: from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
```

```
In [43]: scaler = StandardScaler()
x = scaler.fit_transform(data)
```

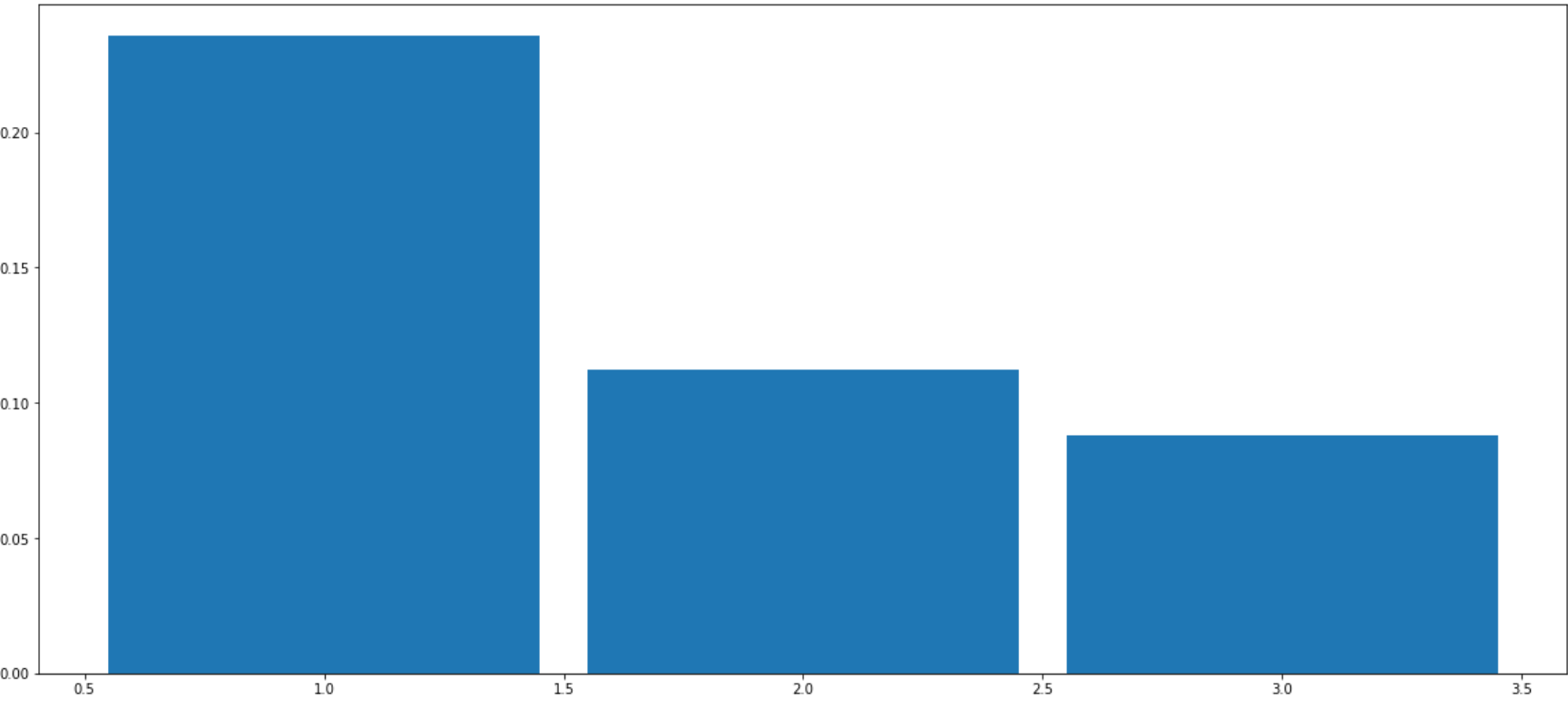
```
In [44]: pca = PCA(n_components=3)
pca.fit(x)
x = pca.transform(x)
```

```
In [45]: pca.explained_variance_ratio_
```

Out[45]: array([0.23581966, 0.11229748, 0.08800207])

```
In [46]: Variance = pca.explained_variance_ratio_
```

```
In [47]: plt.figure(figsize=(20,9))
plt.bar(range(1,len(Variance)+1),height=Variance,width=0.9)
plt.show()
```



```
In [48]: pca.explained_variance_
```

Out[48]: array([3.31240732, 1.57737062, 1.23610856])