

1. Download the dataset: Dataset

2. Load the dataset.

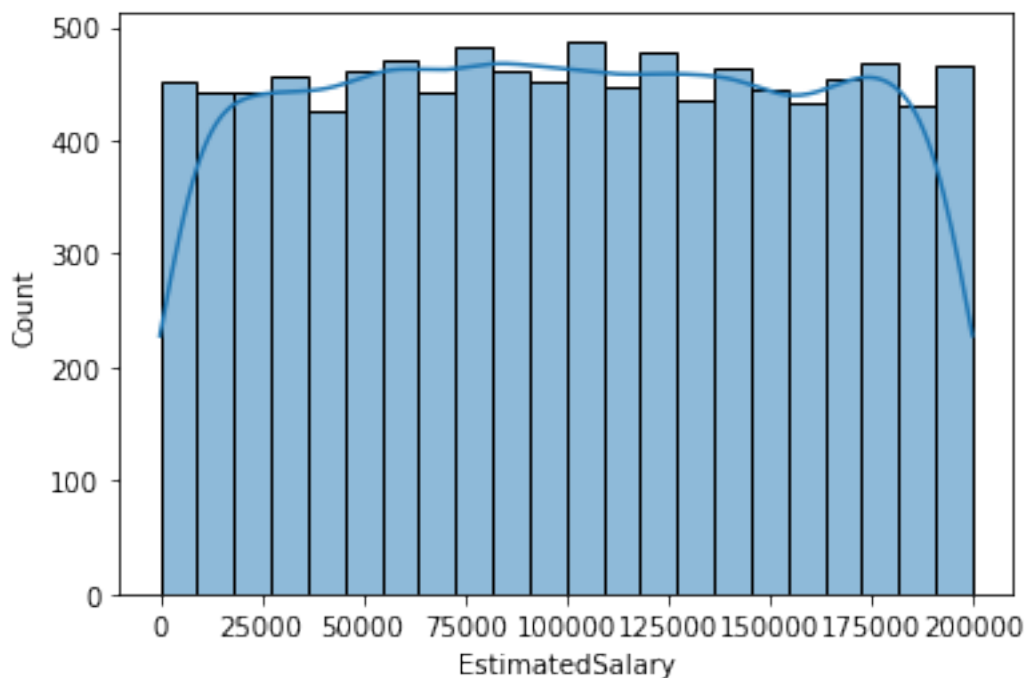
```
import numpy as np
import pandas as pd
df = pd.read_csv("Churn_Modelling.csv")
```

3. Perform Below Visualizations.

● Univariate Analysis

```
import seaborn as sns
sns.histplot(df.EstimatedSalary, kde=True)
```

<AxesSubplot:xlabel='EstimatedSalary', ylabel='Count'>



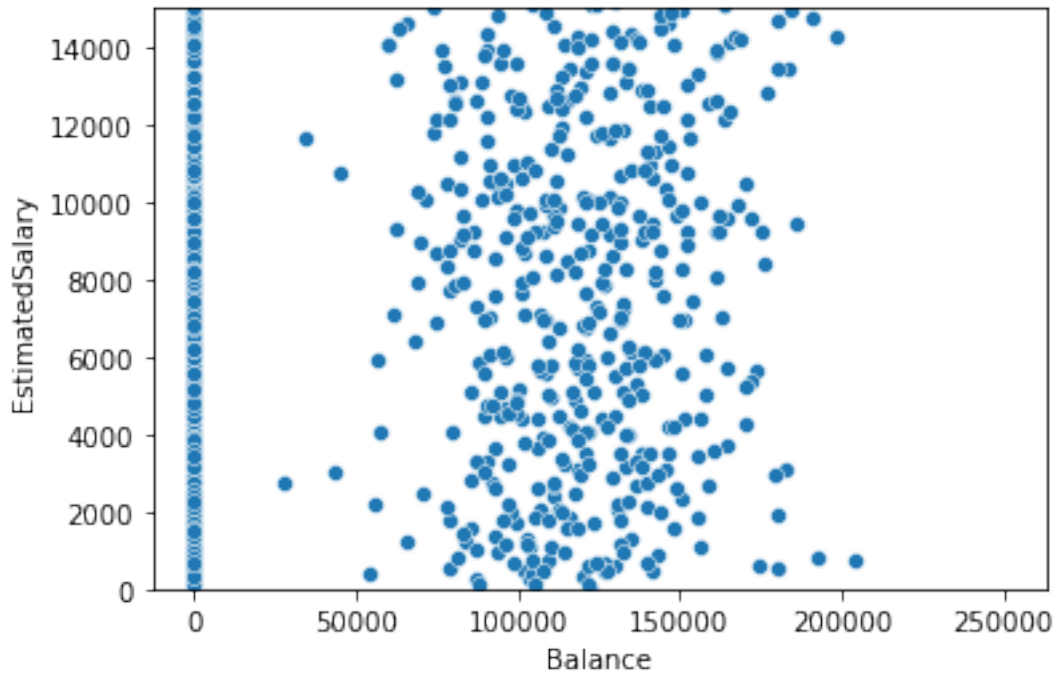
● Bi - Variate Analysis

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.scatterplot(df.Balance, df.EstimatedSalary)
plt.ylim(0, 15000)
```

C:\Users\ELC0T\anaconda3\lib\site-packages\seaborn_decorators.py:36:
FutureWarning: Pass the following variables as keyword args: x, y.

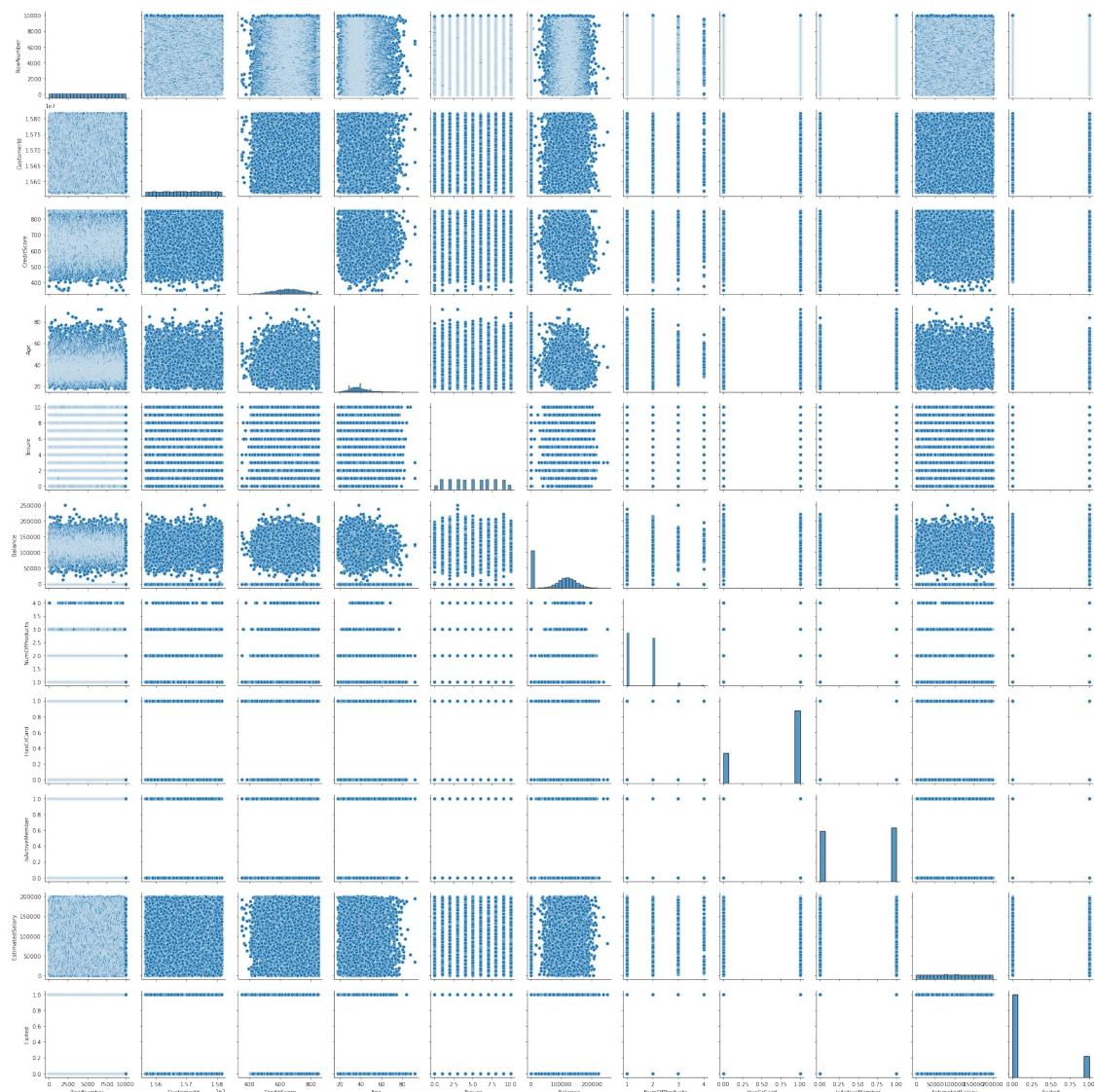
From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(  
(0.0, 15000.0)
```



● Multi - Variate Analysis

```
import seaborn as sns  
df=pd.read_csv("Churn_Modelling.csv")  
sns.pairplot(df)  
  
<seaborn.axisgrid.PairGrid at 0x1c4d49721c0>
```



4. Perform descriptive statistics on the dataset.

```
df=pd.read_csv("Churn_Modelling.csv")
df.describe(include='all')
```

	RowNumber	CustomerId	Surname	CreditScore	Geography
Gender	\				
count	10000.00000	1.000000e+04	10000	10000.000000	10000
unique	NaN	NaN	2932	NaN	3
2					
top	NaN	NaN	Smith	NaN	France
Male					
freq	NaN	NaN	32	NaN	5014
5457					
mean	5000.50000	1.569094e+07	NaN	650.528800	NaN

NaN					
std	2886.89568	7.193619e+04	NaN	96.653299	NaN
NaN					
min	1.000000	1.556570e+07	NaN	350.000000	NaN
NaN					
25%	2500.75000	1.562853e+07	NaN	584.000000	NaN
NaN					
50%	5000.50000	1.569074e+07	NaN	652.000000	NaN
NaN					
75%	7500.25000	1.575323e+07	NaN	718.000000	NaN
NaN					
max	10000.00000	1.581569e+07	NaN	850.000000	NaN
NaN					

	Age	Tenure	Balance	NumOfProducts
HasCrCard \				
count	10000.000000	10000.000000	10000.000000	10000.000000
10000.000000				
unique	NaN	NaN	NaN	NaN
NaN				
top	NaN	NaN	NaN	NaN
NaN				
freq	NaN	NaN	NaN	NaN
NaN				
mean	38.921800	5.012800	76485.889288	1.530200
0.70550				
std	10.487806	2.892174	62397.405202	0.581654
0.45584				
min	18.000000	0.000000	0.000000	1.000000
0.00000				
25%	32.000000	3.000000	0.000000	1.000000
0.00000				
50%	37.000000	5.000000	97198.540000	1.000000
1.00000				
75%	44.000000	7.000000	127644.240000	2.000000
1.00000				
max	92.000000	10.000000	250898.090000	4.000000
1.00000				

	IsActiveMember	EstimatedSalary	Exited
count	10000.000000	10000.000000	10000.000000
unique	NaN	NaN	NaN
top	NaN	NaN	NaN
freq	NaN	NaN	NaN
mean	0.515100	100090.239881	0.203700
std	0.499797	57510.492818	0.402769
min	0.000000	11.580000	0.000000
25%	0.000000	51002.110000	0.000000
50%	1.000000	100193.915000	0.000000

75%	1.000000	149388.247500	0.000000
max	1.000000	199992.480000	1.000000

5. Handle the Missing values.

```
from ast import increment_lineno
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(color_codes=True)
df=pd.read_csv("Churn_Modelling.csv")
df.head()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age
\	0	1	15634602	Hargrave	619	France	Female
1	2	15647311	Hill	608	Spain	Female	41
2	3	15619304	Onio	502	France	Female	42
3	4	15701354	Boni	699	France	Female	39
4	5	15737888	Mitchell	850	Spain	Female	43

	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	\
0	2	0.00	1	1	1	
1	1	83807.86	1	0	1	
2	8	159660.80	3	1	0	
3	1	0.00	2	0	0	
4	2	125510.82	1	1	1	

	EstimatedSalary	Exited
0	101348.88	1
1	112542.58	0
2	113931.57	1
3	93826.63	0
4	79084.10	0

6. Find the outliers and replace the outliers

```
import pandas as pd
import matplotlib
from matplotlib import pyplot as pyplot
%matplotlib inline
matplotlib.rcParams['figure.figsize']=(10,6)
```

```
df=pd.read_csv("Churn_Modelling.csv")
df.sample(5)
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	
Age \	2289	2290	15789097	Keeley	644	France	Male
48	8327	8328	15766787	Piazza	707	France	Female
35	6626	6627	15619932	Lombardi	847	France	Male
66	3501	3502	15802060	Ch'ang	646	Germany	Female
30	9467	9468	15734850	Milanesi	676	Spain	Male
36							

	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	\
2289	8	0.00	2	0		1
8327	9	0.00	2	1		1
6626	7	123760.68	1	0		1
3501	10	100548.67	2	0		0
9467	1	82729.49	1	1		0

	EstimatedSalary	Exited
2289	44965.54	1
8327	70403.65	0
6626	53157.16	0
3501	136983.77	0
9467	113810.12	0

7. Check for Categorical columns and perform encoding.

```
df=pd.read_csv("Churn_Modelling.csv")
df.columns
import pandas as pd
import numpy as np
headers=['RowNumber','CustomerId','Surname','CreditScore','Geography',
'Gender','Age','Tenure','Balance','NumofProducts','HasCard',
'IsActiveMember','EstimatedSalary','Exited']
import seaborn as sns
df.head()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age
\							
0	1	15634602	Hargrave	619	France	Female	42
1	2	15647311	Hill	608	Spain	Female	41
2	3	15619304	Onio	502	France	Female	42

3	4	15701354	Boni	699	France	Female	39
4	5	15737888	Mitchell	850	Spain	Female	43

	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	\
0	2	0.00	1	1	1	
1	1	83807.86	1	0	1	
2	8	159660.80	3	1	0	
3	1	0.00	2	0	0	
4	2	125510.82	1	1	1	

	EstimatedSalary	Exited
0	101348.88	1
1	112542.58	0
2	113931.57	1
3	93826.63	0
4	79084.10	0

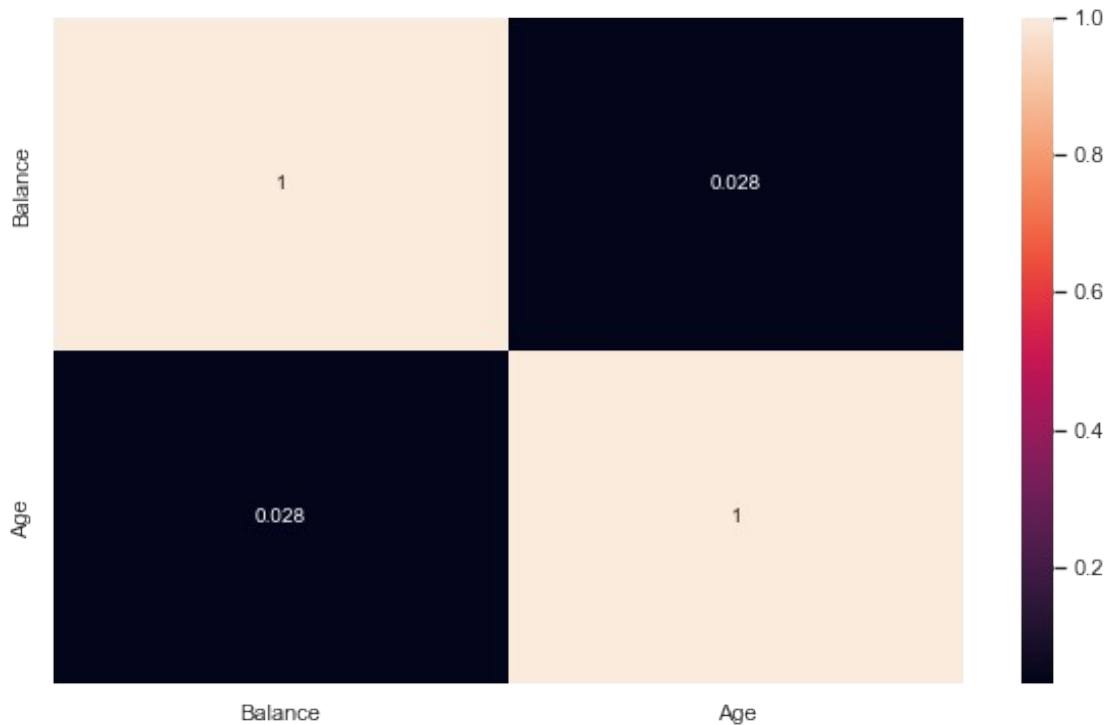
8. Split the data into dependent and independent variables.

```
x=df.iloc[:, :-1].values
print(x)
y=df.iloc[:, :-1]._values
print(y)

[[1 15634602 'Hargrave' ... 1 1 101348.88]
 [2 15647311 'Hill' ... 0 1 112542.58]
 [3 15619304 'Onio' ... 1 0 113931.57]
 ...
 [9998 15584532 'Liu' ... 0 1 42085.58]
 [9999 15682355 'Sabbatini' ... 1 0 92888.52]
 [10000 15628319 'Walker' ... 1 0 38190.78]]
[1 0 1 ... 1 1 0]
```

9. Scale the independent variables

```
import seaborn as sns
df=pd.read_csv("Churn_Modelling.csv")
dff=df[['Balance', 'Age']]
sns.heatmap(dff.corr(), annot=True)
sns.set(rc={'figure.figsize':(40,40)})
```



10. Split the data into training and testing

```
from scipy.sparse.construct import random
x=df.iloc[:, 1:2].values
y=df.iloc[:,2].values
from sklearn.model_selection import train_test_split
x_train, x_test, y_train,
y_test=train_test_split(x,y,test_size=0.2,random_state=0)
print('Row count of x_train table'+'-'+str(f"{len(x_train):,}"))
print('Row count of y_train table'+'-'+str(f"{len(y_train):,}"))
print('Row count of x_test table'+'-'+str(f"{len(x_test):,}"))
print('Row count of y_test table'+'-'+str(f"{len(y_test):,}"))
```

```
Row count of x_train table-8,000
Row count of y_train table-8,000
Row count of x_test table-2,000
Row count of y_test table-2,000
```