```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
path = "/content/Data1.csv"
df = pd.read_csv(path)
df
```

	Country	Age	Salary	Purchased
0	France	44.0	72000.0	No
1	Spain	27.0	48000.0	Yes
2	Germany	30.0	54000.0	No
3	Spain	38.0	61000.0	No
4	Germany	40.0	NaN	Yes
5	France	35.0	58000.0	Yes
6	Spain	NaN	52000.0	No
7	France	48.0	79000.0	Yes
8	Germany	50.0	83000.0	No
9	France	37.0	67000.0	Yes

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Country    10 non-null     object
 1   Age        9 non-null      float64
 2   Salary     9 non-null      float64
 3   Purchased  10 non-null     object
dtypes: float64(2), object(2)
memory usage: 448.0+ bytes
```

```
df.describe()
```

	Age	Salary
count	9.000000	9.000000
mean	38.777778	63777.777778
std	7.693793	12265.579662
min	27.000000	48000.000000
25%	35.000000	54000.000000
50%	38.000000	61000.000000
75%	44.000000	72000.000000
max	50.000000	83000.000000

```
df.head(10)
```

	Country	Age	Salary	Purchased
0	France	44.0	72000.0	No
1	Spain	27.0	48000.0	Yes
2	Germany	30.0	54000.0	No
3	Spain	38.0	61000.0	No
4	Germany	40.0	NaN	Yes
5	France	35.0	58000.0	Yes
6	Spain	NaN	52000.0	No
7	France	48.0	79000.0	Yes
8	Germany	50.0	83000.0	No
9	France	37.0	67000.0	Yes

```
df.tail(10)
```

	Country	Age	Salary	Purchased
0	France	44.0	72000.0	No
1	Spain	27.0	48000.0	Yes
2	Germany	30.0	54000.0	No
3	Spain	38.0	61000.0	No
4	Germany	40.0	NaN	Yes
5	France	35.0	58000.0	Yes
6	Spain	NaN	52000.0	No
7	France	48.0	79000.0	Yes
8	Germany	50.0	83000.0	No
9	France	37.0	67000.0	Yes

```
df['Salary'].fillna(45000, inplace=True)
df['Age'].fillna(30, inplace=True)
```

```
mean_salary = np.around(df['Salary'].mean(), decimals=2)
print(mean_salary)
```

```
61900.0
```

```
mean_Age = np.around(df['Age'].mean(), decimals=2)
print(mean_Age)
```

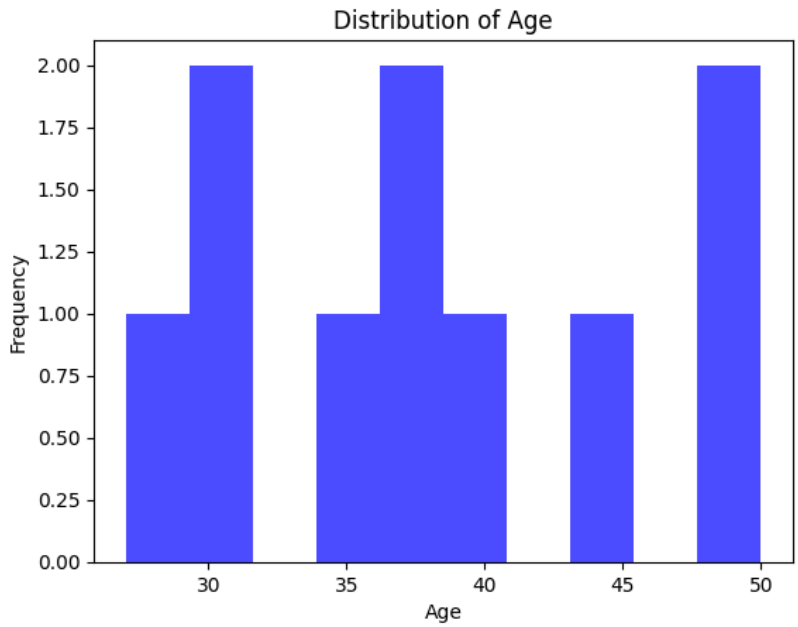
```
37.9
```

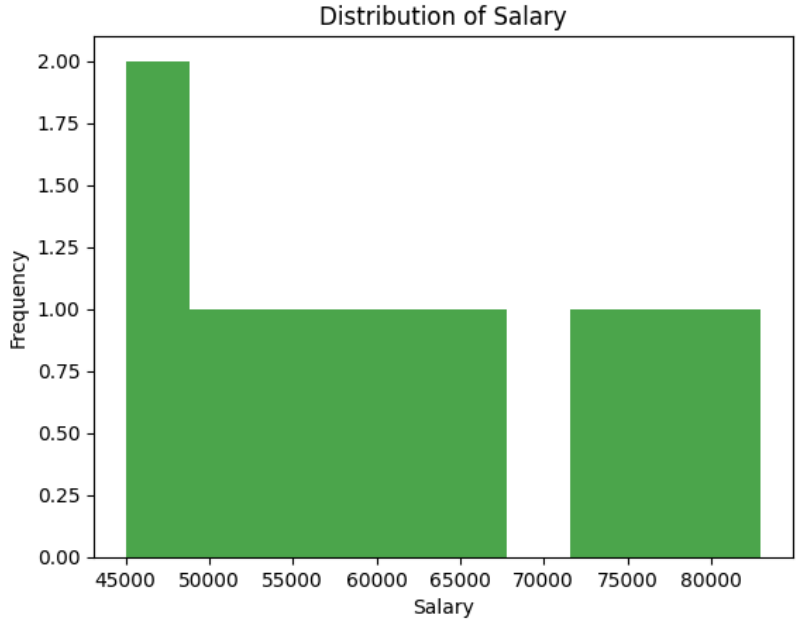
```
std_salary = np.around(df['Salary'].std(), decimals=2)
print(std_salary)
std_age = np.around(df['Age'].std(), decimals=2)
print(std_age)
```

```
12999.57
7.77
```

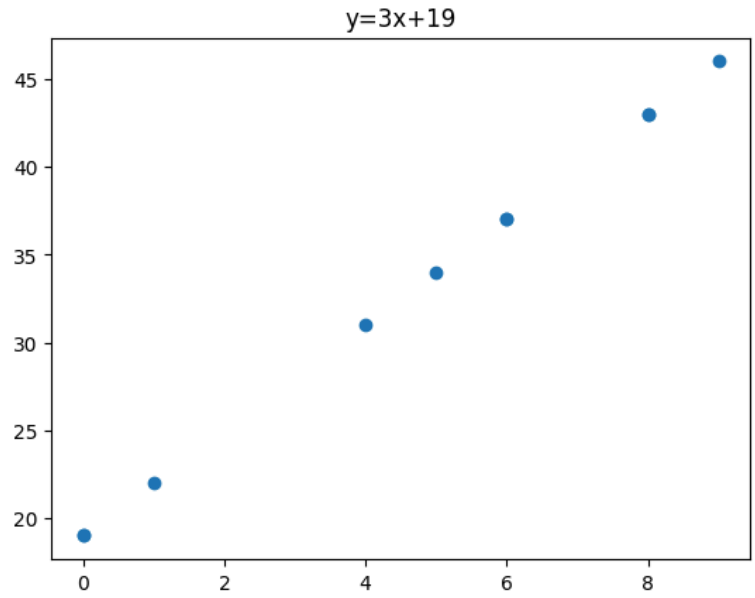
```
plt.hist(df['Age'], bins=10, color='blue', alpha=0.7)
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.title('Distribution of Age')
plt.show()

plt.hist(df['Salary'], bins=10, color='green', alpha=0.7)
plt.xlabel('Salary')
plt.ylabel('Frequency')
plt.title('Distribution of Salary')
```

```
plt.show()
```



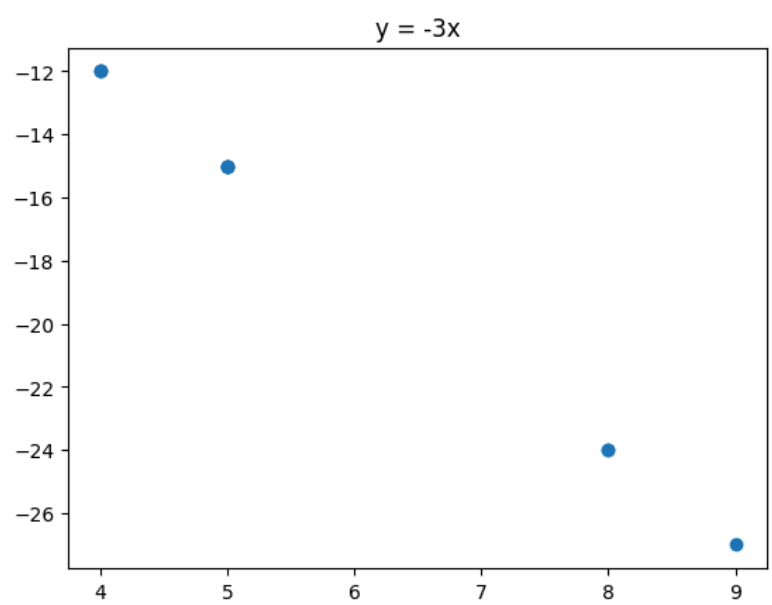



```
# random plot
import numpy as np
x_point = np.random.randint(0,10,10)
y_line = 3*x_point+19
plt.scatter(x_point,y_line)
plt.title("y=3x+19")
plt.show()
data = np.corrcoef(x_point,y_line)
print(data)
```



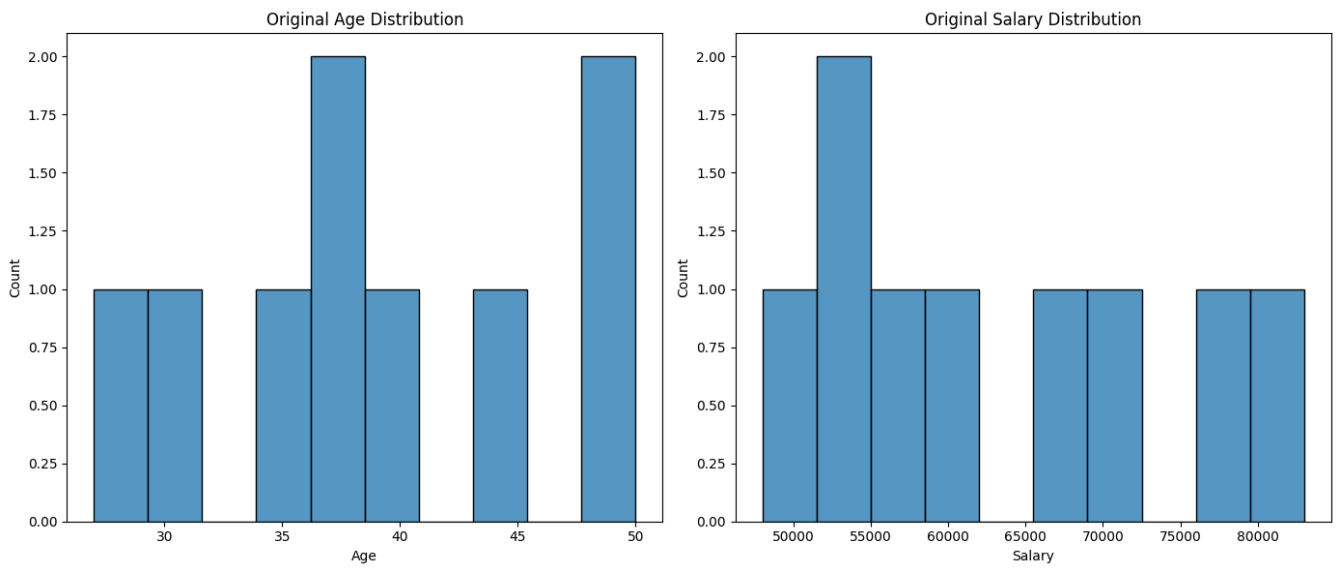

```
[[1. 1.]
 [1. 1.]]
```

```
import numpy as np
x_point = np.random.randint(0,10,10)
y_line = -3*x_point
plt.scatter(x_point,y_line)
plt.title("y = -3x")
plt.show()
data = np.corrcoef(x_point,y_line)
print(data)
```




```
[[ 1. -1.]
 [-1.  1.]]
```

```
import seaborn as sns
plt.figure(figsize=(14,6))
plt.subplot(1,2,1)
sns.histplot(df['Age'].dropna(),bins=10)
plt.title("Original Age Distribution")
plt.subplot(1,2,2)
sns.histplot(df['Salary'].dropna(),bins=10)
plt.title("Original Salary Distribution")
plt.tight_layout()
```



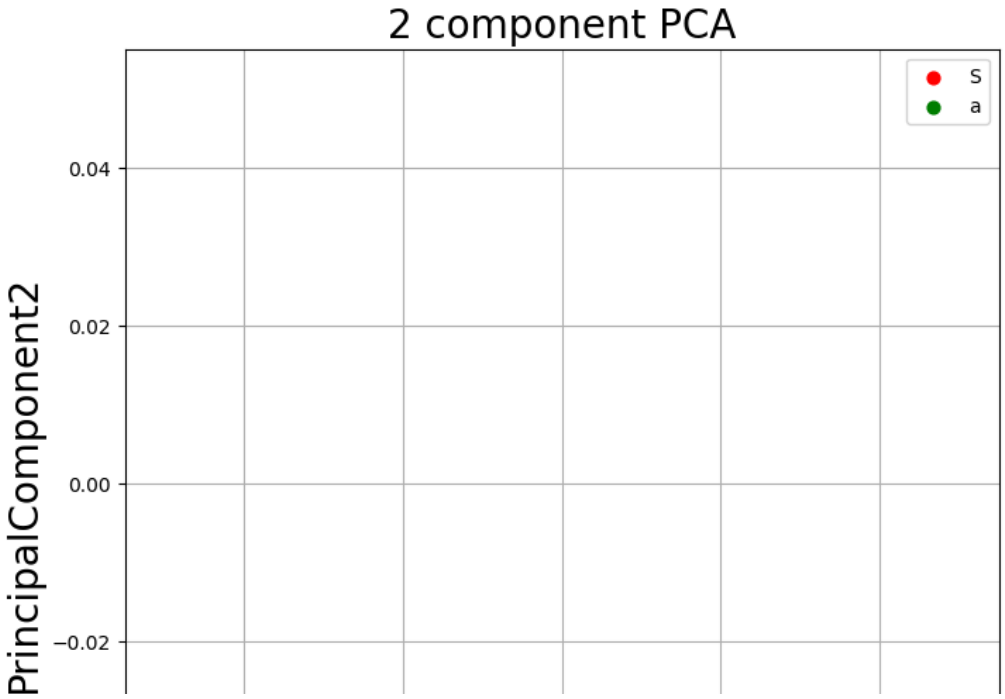

```
#PCA implementation
from sklearn.preprocessing import StandardScaler
features = ['Age','Salary']
x = df.loc[:, features].values
y = df.loc[:, ['Purchased']].values
x = StandardScaler().fit_transform(x)
```

```
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
principalComponents = pca.fit_transform(x)
principalDf = pd.DataFrame(data = principalComponents, columns = ['pt1', 'pt2'])
```

```
final = pd.concat([principalDf, df[['Purchased']]], axis = 1)
final
```

	pt1	pt2	Purchased
0	1.164506	0.006300	No
1	-1.843032	-0.249064	Yes
2	-1.211107	-0.305183	No
3	-0.042006	0.061200	No
4	-0.767462	1.170527	Yes
5	-0.501920	-0.054692	Yes
6	-1.325781	-0.190509	No
7	1.949736	-0.011188	Yes
8	2.371019	-0.048601	No
9	0.206048	-0.378789	Yes

```
fig = plt.figure(figsize = (8,8))
a= fig.add_subplot(1,1,1)
a.set_xlabel('PrincipalComponent1', fontsize = 20)
a.set_ylabel('PrincipalComponent2', fontsize = 20)
a.set_title('2 component PCA', fontsize = 20)
purchased =  ['Age','Salary']
colors = ['r', 'g']
for purchased, color in zip(purchased,colors):
  indicesToKeep = final['Purchased'] == purchased
  a.scatter(final.loc[indicesToKeep, 'pt1']
            , final.loc[indicesToKeep, 'pt2']
            , c = color
            , s = 40)
a.legend(purchased)
a.grid()
```


2 component PCA
S
a
0.04
0.02
0.00
−0.02
PrincipalComponent2