

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
from sklearn.preprocessing import StandardScaler
from IPython.display import display, HTML
import re
```

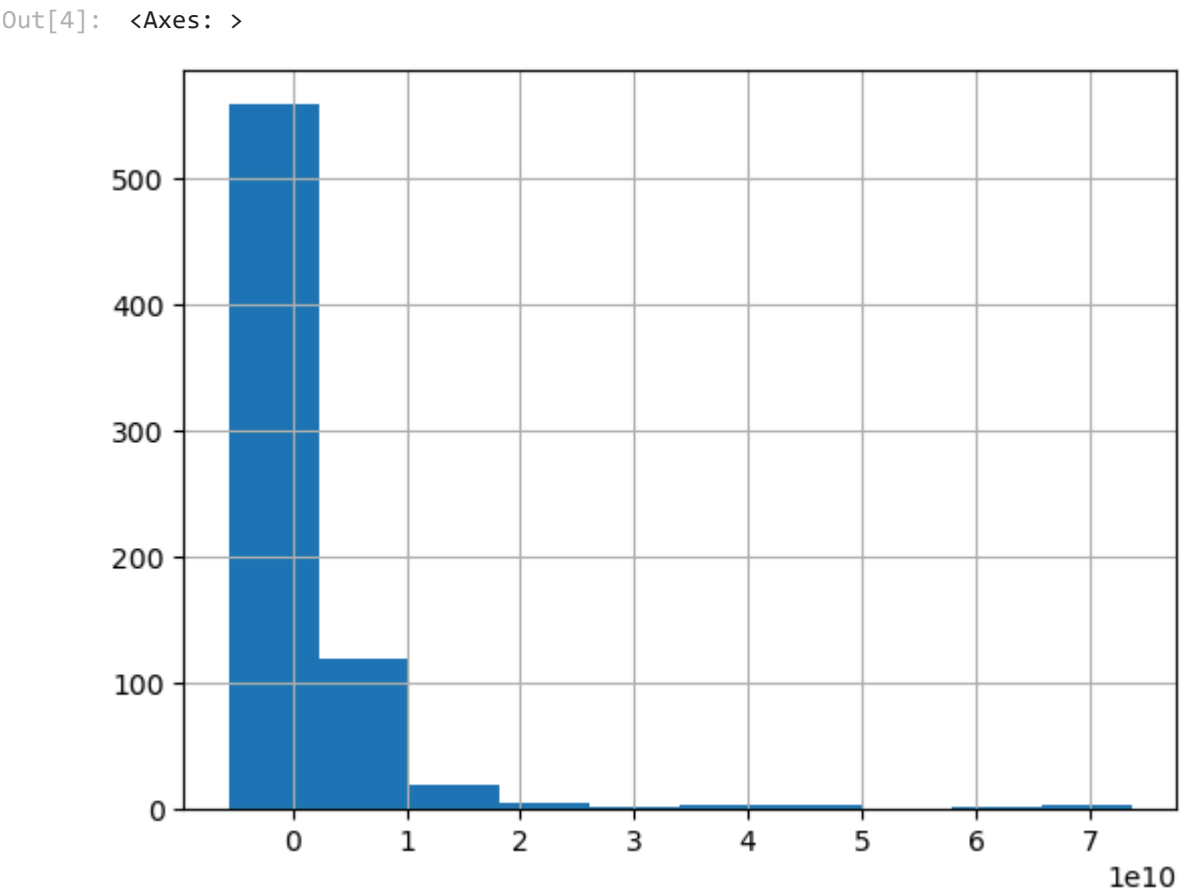
```
In [2]: data = pd.read_csv('BalanceSheetDataSet.csv')
df = data.copy()
```

```
In [3]: def getting_primary_info(df):
print("-----")
print("Veri setinin şekli", df.shape)
print("-----")
print("Veri seti değişken tipleri:\n", df.dtypes)
print("-----")
# print("Veri setinin ilk 5 satırı")
# display(HTML(df.head().to_html()))
print("-----")
print("Veri setinin istatistiki verileri")
description = df.describe()
display(HTML(description.to_html()))
print("-----")
getting_primary_info(df)

-----
Veri setinin şekli (715, 226)
-----
Veri seti değişken tipleri:
ŞirketID                int64
yıl                    int64
ay                    int64
Dönem Net Kar/Zararı    int64
Nakit ve Nakit Benzerleri int64
...
Vergi İadeleri (Ödemeleri).2 int64
Diğer Nakit Girişleri (Çıkışları).2 int64
Yabancı Para Çevrim Farklarının Etkisinden Önce Nakit ve Nakit Benzerlerindeki Net Artış (Azalış) int64
Yabancı Para Çevrim Farklarının Nakit ve Nakit Benzerleri Üzerindeki Etkisi int64
Nakit ve Nakit Benzerlerindeki Net Artış (Azalış) int64
Length: 226, dtype: object
-----
Veri setinin istatistiki verileri
```

	ŞirketID	yıl	ay	Dönem Net Kar/Zararı	Nakit ve Nakit Benzerleri	Finansal Yatırımlar	Ticari Alacaklar	Finans Sektörü Faaliyetlerinden Alacaklar	Türkiye Cumhuriyet Merkez Bankası Hesabı	Diğer Alacaklar	Müşteri Sözleşmelerinden Doğan Varlıklar	İmtiyaz Sözleşmelerine İlişkin Finansal Varlıklar	Türev Araçlar	Stoklar	Peş Ödenm Gideri
count	715.000000	715.000000	715.000000	7.150000e+02	7.150000e+02	7.150000e+02	7.150000e+02	7.150000e+02	7.150000e+02	7.150000e+02	7.150000e+02	7.150000e+02	7.150000e+02	7.150000e+02	7.150000e+02
mean	12.002797	2019.383217	7.430769	2.388901e+09	8.583014e+09	2.271270e+09	4.581171e+09	1.187092e+10	3.731317e+09	7.586390e+08	3.817790e+07	1.549263e+07	6.665678e+08	4.439744e+09	6.146441e+09
std	7.220419	2.125278	3.312256	7.132098e+09	1.832715e+10	8.647790e+09	9.249685e+09	5.633831e+10	1.770676e+10	2.364208e+09	2.028218e+08	1.086820e+08	2.706097e+09	1.017342e+10	1.600654e+10
min	0.000000	2016.000000	3.000000	-5.588000e+09	4.105000e+06	0.000000e+00	1.748839e+07	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	6.000000	2018.000000	6.000000	4.495745e+07	3.460132e+08	0.000000e+00	5.111130e+08	0.000000e+00	0.000000e+00	1.333442e+06	0.000000e+00	0.000000e+00	0.000000e+00	2.038064e+08	2.658800e+08
50%	12.000000	2019.000000	6.000000	4.965920e+08	2.289734e+09	1.002000e+06	1.633997e+09	0.000000e+00	0.000000e+00	2.817800e+07	0.000000e+00	0.000000e+00	1.655560e+06	1.049563e+09	1.446140e+09
75%	18.000000	2021.000000	9.000000	2.011972e+09	8.905397e+09	3.058807e+08	4.263423e+09	0.000000e+00	0.000000e+00	2.292735e+08	0.000000e+00	0.000000e+00	8.365850e+07	3.862659e+09	5.438590e+09
max	24.000000	2023.000000	12.000000	7.367500e+10	2.124340e+11	9.067754e+10	1.069480e+11	5.560856e+11	1.509078e+11	2.521500e+10	2.258913e+09	1.353922e+09	4.232347e+10	9.539600e+10	2.063308e+11

```
In [4]: df["Dönem Net Kar/Zararı"].hist()
```



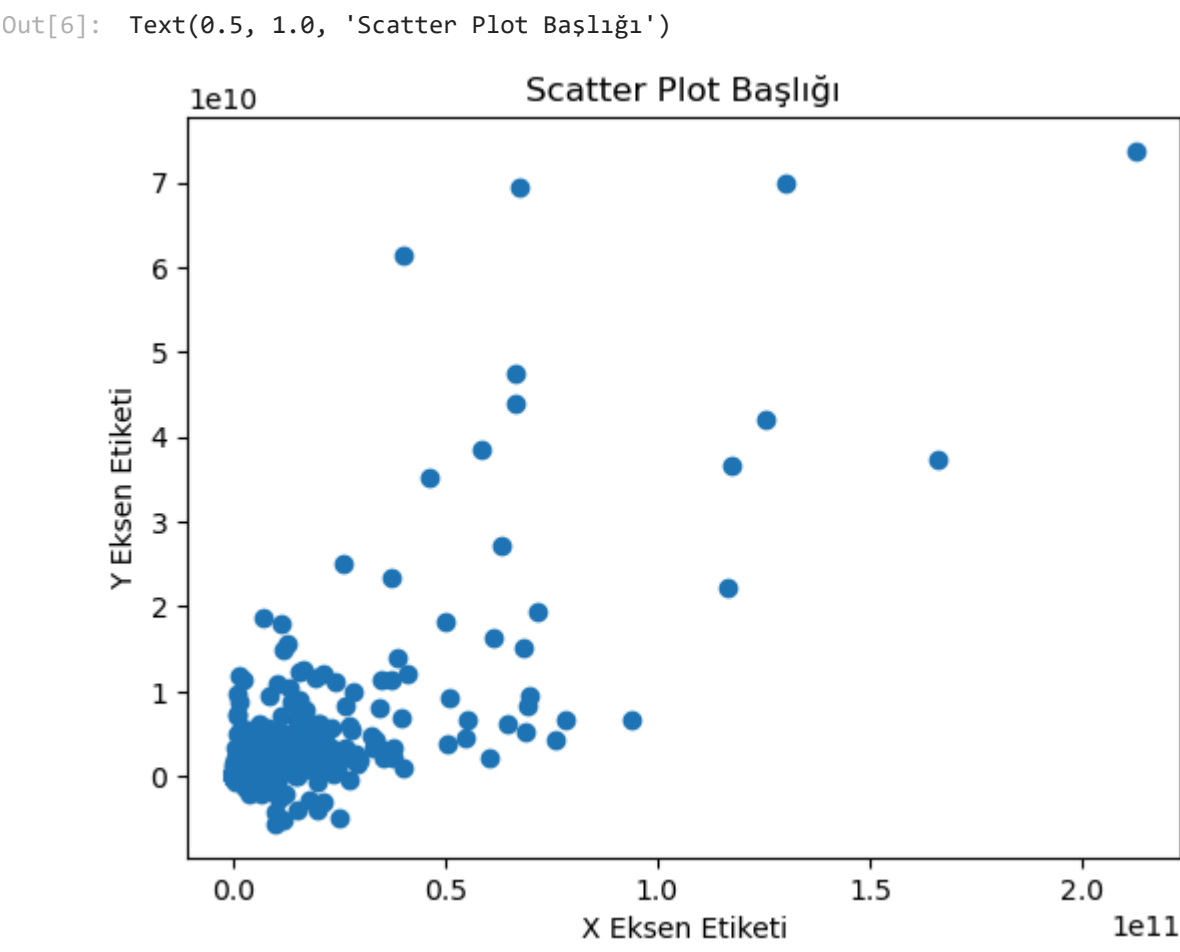
```
In [5]: df.query("Dönem Net Kar/Zararı"== 7.367500e+10)
```

Out[5]:

	ŞirketID	yıl	ay	Dönem Net Kar/Zararı	Nakit ve Nakit Benzerleri	Finansal Yatırımlar	Ticari Alacaklar	Finans Sektörü Faaliyetlerinden Alacaklar	Türkiye Cumhuriyet Merkez Bankası Hesabı	Diğer Alacaklar	...	Türev Araçlardan Nakit Girişleri.1	Türev Araçlardan Nakit Çıkışları	Ödenen Temettüler	Ödenen Faiz.1	Alınan Faiz.2	Vergi İadeleri (Ödemeleri).2	Diğer Nakit Girişleri (Çıkışları).2
687	24	2023	6	73675000000	212434000000	24647000000	106948000000	510922000000	144349000000	13525000000	...	0	5711000000	-17128000000	-8306000000	4902000000	0	-629000000

1 rows x 226 columns

```
In [6]: plt.scatter(df['Nakit ve Nakit Benzerleri'], df['Dönem Net Kar/Zararı'])
plt.xlabel('X Eksen Etiket')
plt.ylabel('Y Eksen Etiket')
plt.title('Scatter Plot Başlığı')
```



```
In [7]: df["ŞirketID"].value_counts()
```

Out[7]:

ŞirketID	count
0	29
11	29
23	29
22	29
21	29
19	29
18	29
15	29
14	29
1	29
12	29
10	29
9	29
4	29
3	29
13	28
8	28
7	28
16	28
17	28
6	28
5	28
20	28
2	28
24	28

Name: count, dtype: int64

```
In [8]: from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor

df = df.dropna()

X, y = df.drop(columns=["Nakit ve Nakit Benzerleri"], df["Dönem Net Kar/Zararı"])

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=47)

model = LinearRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

r2 = r2_score(y_test, y_pred)
print("R-squared:", r2)

model_random_forest = RandomForestRegressor(n_estimators=100, min_samples_split=2)
model_random_forest.fit(X_train, y_train)

y_pred_rf = model_random_forest.predict(X_test)

r2_rf = r2_score(y_test, y_pred_rf)
print("Random Forest Test R^2 Score:", r2_rf)

y_pred_train = model_random_forest.predict(X_train)

r2_rf_train = r2_score(y_train, y_pred_train)
print("Random Forest Train R^2 Score:", r2_rf_train)

R-squared: 1.0
Random Forest Test R^2 Score: 0.9201522273737457
Random Forest Train R^2 Score: 0.9928604497186072
```

```
In [ ]:
```