```
# Importing the necessary libraries

import numpy as np, pandas as pd
import seaborn as sns

import warnings
warnings.filterwarnings("ignore")

# Read the CSV file

df = pd.read_csv("Footwear_v2.csv")
df.head()
```

	Supplier	Delhi	Mumbai	Jaipur	Hyderabad
0	Supplier 1	4.57%	4.18%	7.90%	8.54%
1	supplier 2	2.60%	1.88%	8.99%	9.23%
2	supplier 3	2.26%	7.48%	6.71%	7.84%
3	supplier 4	6.47%	6.70%	7.94%	9.89%
4	supplier 5	6.82%	1.17%	8.54%	5.67%

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

```
# to check for null rows in each columns
df.isnull().sum()
```

	0
Supplier	0
Delhi	0
Mumbai	0
Jaipur	0
Hyderabad	0

```
# To check the Statistics
df.describe()
```

	Supplier	Delhi	Mumbai	Jaipur	Hyderabad
count	30	30	30	30	30
unique	30	30	29	29	30
top	Supplier 1	4.57%	2.42%	8.99%	8.54%
freq	1	1	2	2	1

```
df.shape
```

```
(30, 5)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 5 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Supplier   30 non-null     object
 1   Delhi      30 non-null     object
 2   Mumbai     30 non-null     object
 3   Jaipur     30 non-null     object
 4   Hyderabad  30 non-null     object
dtypes: object(5)
memory usage: 1.3+ KB
```

```
# clean the % sign
def clean(string):
    clean = "".join(filter(lambda x:x!= '%', string))
    return float(clean)
```

```
df.Supplier.value_counts()
```

Supplier	count
Supplier 1	1
supplier 2	1
supplier 29	1
supplier 28	1
supplier 27	1
supplier 26	1
supplier 25	1
supplier 24	1
supplier 23	1
supplier 22	1
supplier 21	1
supplier 20	1
supplier 19	1
supplier 18	1
supplier 17	1
supplier 16	1
supplier 15	1
supplier 14	1
supplier 13	1
supplier 12	1
supplier 11	1
supplier 10	1
supplier 9	1
supplier 8	1
supplier 7	1
supplier 6	1
supplier 5	1
supplier 4	1
supplier 3	1
supplier 30	1

```
# to take care of the supplier column
def supply_cleaner(string):
    return string.lower()
```

```
# apply the function that we have created
df["Supplier"] = df["Supplier"].apply(supply_cleaner)
df["Delhi"] = df["Delhi"].apply(clean)
df["Mumbai"] = df["Mumbai"].apply(clean)
df["Jaipur"] = df["Jaipur"].apply(clean)
df["Hyderabad"] = df["Hyderabad"].apply(clean)
```

```
df.head()
```

	Supplier	Delhi	Mumbai	Jaipur	Hyderabad
0	supplier 1	4.57	4.18	7.90	8.54
1	supplier 2	2.60	1.88	8.99	9.23
2	supplier 3	2.26	7.48	6.71	7.84
3	supplier 4	6.47	6.70	7.94	9.89
4	supplier 5	6.82	1.17	8.54	5.67

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

```
df.info()
```

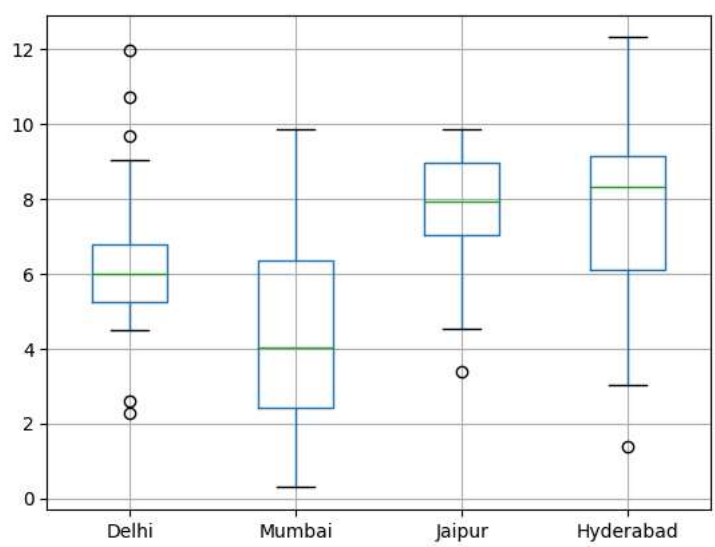
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 5 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Supplier   30 non-null     object
 1   Delhi      30 non-null     float64
 2   Mumbai     30 non-null     float64
 3   Jaipur     30 non-null     float64
 4   Hyderabad  30 non-null     float64
dtypes: float64(4), object(1)
memory usage: 1.3+ KB
```

```
df.describe()
```

	Delhi	Mumbai	Jaipur	Hyderabad
count	30.000000	30.000000	30.000000	30.000000
mean	6.324000	4.555000	7.695667	7.727667
std	2.095982	2.519117	1.586503	2.477642
min	2.260000	0.290000	3.400000	1.370000
25%	5.237500	2.432500	7.017500	6.110000
50%	6.000000	4.020000	7.920000	8.305000
75%	6.782500	6.335000	8.980000	9.135000
max	11.960000	9.850000	9.870000	12.310000

```
sub_df = df[["Delhi","Mumbai","Jaipur","Hyderabad"]]
sub_df.boxplot()
```

```
<Axes: >
```




```
data = pd.read_csv("crypto.csv")
```

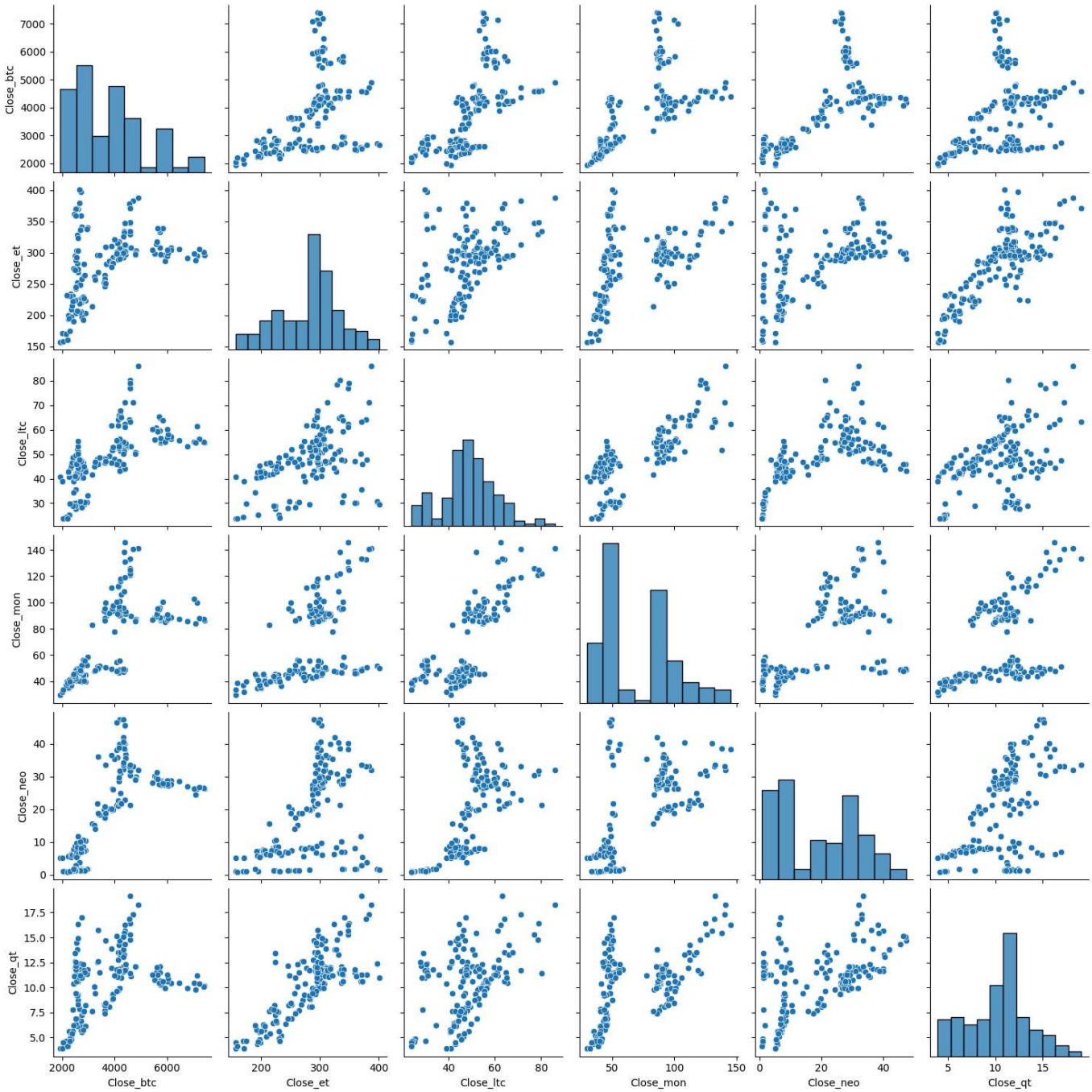
```
data.head()
```

	Close_btc	Close_et	Close_ltc	Close_mon	Close_neo	Close_qt
0	7144.38	294.66	61.30	99.76	26.23	11.21
1	7022.76	298.89	55.17	102.92	26.32	10.44
2	7407.41	296.26	54.75	86.35	26.38	10.13
3	7379.95	300.47	55.04	87.30	26.49	10.05
4	7207.76	305.71	56.18	87.99	26.82	10.38

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

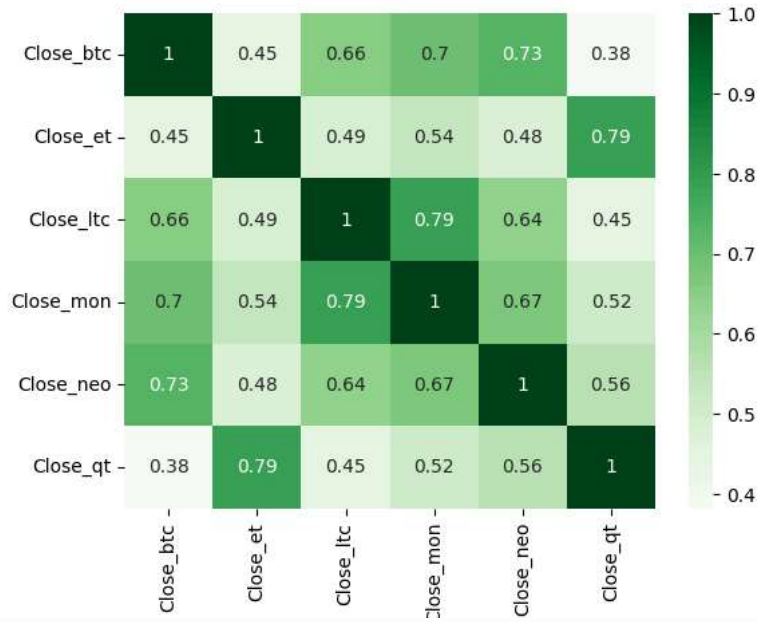
```
sns.pairplot(data)
```

```
<seaborn.axisgrid.PairGrid at 0x7d244735e2c0>
```



```
data2 = data.corr()
sns.heatmap(data2, cmap = "Greens", annot = True)
```

```
<Axes: >
```




```
df = pd.read_csv("Footwear_v2.csv")
df.head()
```

	Supplier	Delhi	Mumbai	Jaipur	Hyderabad
0	Supplier 1	4.57%	4.18%	7.90%	8.54%
1	supplier 2	2.60%	1.88%	8.99%	9.23%
2	supplier 3	2.26%	7.48%	6.71%	7.84%
3	supplier 4	6.47%	6.70%	7.94%	9.89%
4	supplier 5	6.82%	1.17%	8.54%	5.67%

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

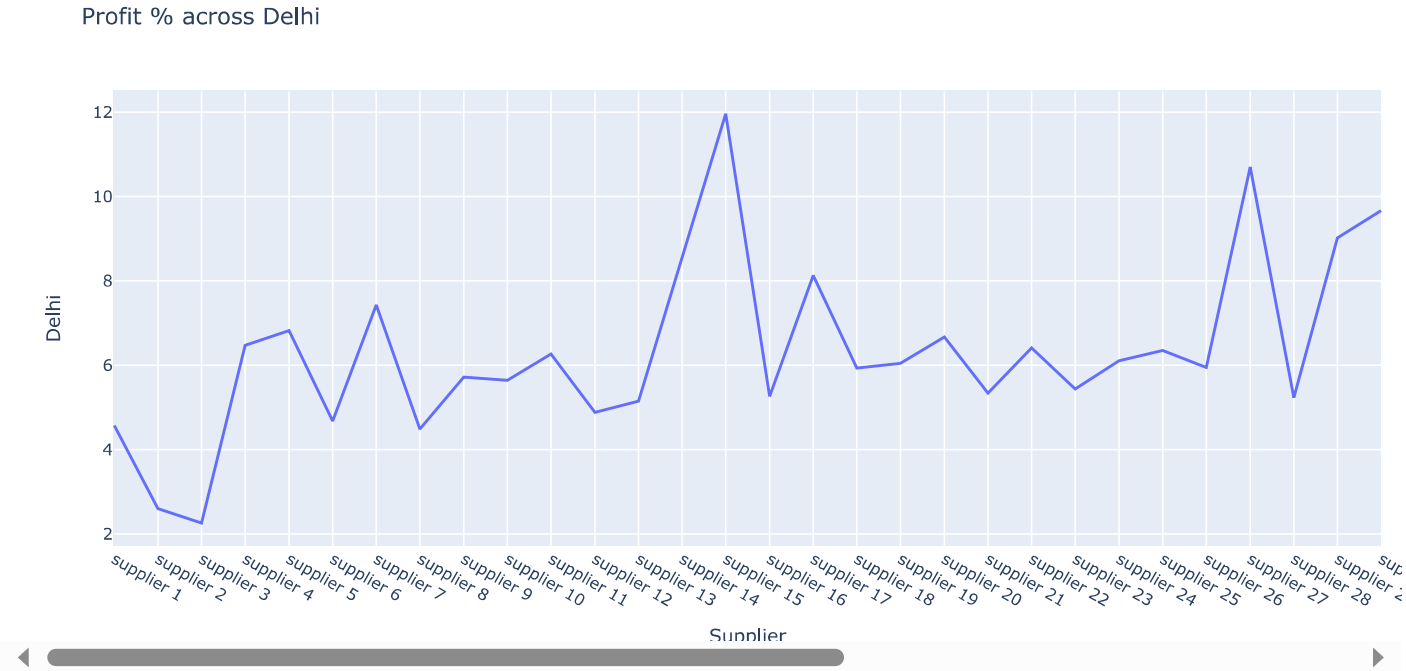
```
# clean the % sign
def clean(string):
    clean = "".join(filter(lambda x:x!= '%', string))
    return float(clean)


# to take care of the supplier column
def supply_cleaner(string):
    return string.lower()


# apply the function that we have created
df["Supplier"] = df["Supplier"].apply(supply_cleaner)
df["Delhi"] = df["Delhi"].apply(clean)
df["Mumbai"] = df["Mumbai"].apply(clean)
df["Jaipur"] = df["Jaipur"].apply(clean)
df["Hyderabad"] = df["Hyderabad"].apply(clean)


sub_df = df[["Delhi","Mumbai","Jaipur","Hyderabad"]]


import plotly.express as px
fig = px.line(df, x = "Supplier", y = "Delhi", title= "Profit % across Delhi" )
fig.show()
```



Start coding or generate with AI.