

1. Represent initial sale date for every token according to data provided.

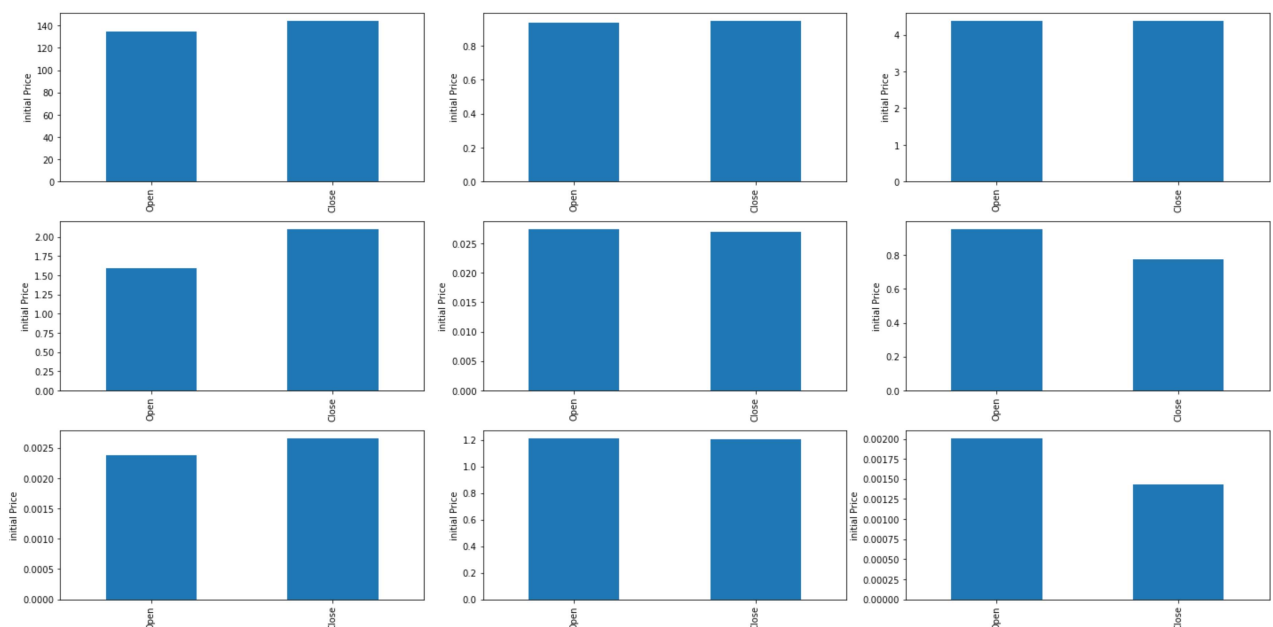
```
In [ ]: import pandas as pd
import matplotlib.pyplot as plt
```

```
In [ ]: bitcoin = pd.read_csv('D:\python\data\coin_Bitcoin.csv')
ethereum = pd.read_csv('D:\python\data\coin_Ethereum.csv')
ethereum = ethereum.sort_values('Date')
Litecoin = pd.read_csv('D:\python\data\coin_Litecoin.csv')
Monero = pd.read_csv('D:\python\data\coin_Monero.csv')
Ripple = pd.read_csv('D:\python\data\coin_Ripple.csv')
Ripple = Ripple.sort_values('Date')
Solana = pd.read_csv('D:\python\data\coin_Solana.csv')
Stellar = pd.read_csv('D:\python\data\coin_Stellar.csv')
Tether = pd.read_csv('D:\python\data\coin_Tether.csv')
Tron = pd.read_csv('D:\python\data\coin_Tron.csv')
```

```
In [ ]: fig, axes = plt.subplots(nrows=3, ncols=3, figsize=(20, 10))
bitcoin[["Open", "Close"]].iloc[0].plot(ax=axes[0][0], kind='bar')
ethereum[["Open", "Close"]].iloc[0].plot(ax=axes[0][1], kind='bar')
Litecoin[["Open", "Close"]].iloc[0].plot(ax=axes[0][2], kind='bar')
Monero[["Open", "Close"]].iloc[0].plot(ax=axes[1][0], kind='bar')
Ripple[["Open", "Close"]].iloc[0].plot(ax=axes[1][1], kind='bar')
Solana[["Open", "Close"]].iloc[0].plot(ax=axes[1][2], kind='bar')
Stellar[["Open", "Close"]].iloc[0].plot(ax=axes[2][0], kind='bar')
Tether[["Open", "Close"]].iloc[0].plot(ax=axes[2][1], kind='bar')
Tron[["Open", "Close"]].iloc[0].plot(ax=axes[2][2], kind='bar')

for ax in fig.axes:
    plt.ylabel('initial Price')
    plt.sca(ax)
    plt.xticks(rotation=90);

fig.tight_layout()
```



In []:

```

fig, ax = plt.subplots(nrows=3 , ncols=3 , figsize=(16,8))

a = bitcoin[['Open', 'Close']].iloc[0]
ax[0][0].plot(a)

b = ethereum[['Open', 'Close']].iloc[0]
ax[0][1].plot(b)

c = Litecoin[['Open', 'Close']].iloc[0]
ax[0][2].plot(c)

d = Monero[['Open', 'Close']].iloc[0]
ax[1][0].plot(d)

e = Ripple[['Open', 'Close']].iloc[0]
ax[1][1].plot(e)

f = Solana[['Open', 'Close']].iloc[0]
ax[1][2].plot(f)

g = Stellar[['Open', 'Close']].iloc[0]
ax[2][0].plot(g)

h = Tether[['Open', 'Close']].iloc[0]
ax[2][1].plot(h)

i = Tron[['Open', 'Close']].iloc[0]
ax[2][2].plot(i)

for ax in fig.axes:
    plt.ylabel('initial Price')
    plt.sca(ax)
    plt.xticks(rotation=90)
fig.tight_layout()

```

