

```
In [ ]: !pip install yfinance
```

```
In [ ]: from google.colab import files
uploaded = files.upload()
```

Choose Files No file chosen

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving fx_trades.csv to fx_trades.csv

```
In [ ]: import numpy as np
import pandas as pd
import yfinance as yfin
```

```
In [ ]: # dataframe
df = yfin.download('AAPL', start='2023-01-29', end='2023-02-03')

[*****100%*****] 1 of 1 completed
```

```
In [ ]: df.head(3) # df.tail(3)
```

```
Out[ ]:
```

	Open	High	Low	Close	Adj Close	Volume
Date						
2023-01-30	144.960007	145.550003	142.850006	143.000000	142.781998	64015300
2023-01-31	142.699997	144.339996	142.279999	144.289993	144.070023	65874500
2023-02-01	143.970001	146.610001	141.320007	145.429993	145.208282	77663600

```
In [ ]: df.index
```

```
Out[ ]: DatetimeIndex(['2023-01-30', '2023-01-31', '2023-02-01', '2023-02-02'], dtype='datetime64[ns]', name='Date', freq=None)
```

```
In [ ]: df.columns
```

```
Out[ ]: Index(['Open', 'High', 'Low', 'Close', 'Adj Close', 'Volume'], dtype='object')
```

```
In [ ]: # row selections
df.loc['2023-02-02'] # pandas series
```

```
Out[ ]: Open          1.489000e+02
High            1.511800e+02
Low             1.481700e+02
Close           1.508200e+02
Adj Close       1.505901e+02
Volume          1.183390e+08
Name: 2023-02-02 00:00:00, dtype: float64
```

```
In [ ]: df.loc['2023-02']
```

```
Out[ ]:
```

	Open	High	Low	Close	Adj Close	Volume
Date						
2023-02-01	143.970001	146.610001	141.320007	145.429993	145.208282	77663600
2023-02-02	148.899994	151.179993	148.169998	150.820007	150.590088	118339000

```
In [ ]: # row selection by position
df.iloc[0]
```

```
Out[ ]: Open      1.449600e+02
High      1.455500e+02
Low       1.428500e+02
Close     1.430000e+02
Adj Close 1.427820e+02
Volume    6.401530e+07
Name: 2023-01-30 00:00:00, dtype: float64
```

```
In [ ]: # column selection
df.Close
```

```
Out[ ]: Date
2023-01-30    143.000000
2023-01-31    144.289993
2023-02-01    145.429993
2023-02-02    150.820007
Name: Close, dtype: float64
```

```
In [ ]: df['Close']
```

```
Out[ ]: Date
2023-01-30    143.000000
2023-01-31    144.289993
2023-02-01    145.429993
2023-02-02    150.820007
Name: Close, dtype: float64
```

```
In [ ]: # cell selection
df.iloc[0, 1]
```

```
Out[ ]: 145.5500030517578
```

```
In [ ]: # add columns to dataframe
df['Change'] = df['Close'] - df['Open']
```

```
In [ ]: df.head(2)
```

```
Out[ ]:
```

	Open	High	Low	Close	Adj Close	Volume	Change
Date							
2023-01-30	144.960007	145.550003	142.850006	143.000000	142.781998	64015300	-1.960007
2023-01-31	142.699997	144.339996	142.279999	144.289993	144.070023	65874500	1.589996

```
In [ ]: df.insert(4, 'Average', (df.High + df.Low)/2)
```

```
In [ ]: df.head(2)
```

```
Out[ ]:
```

	Open	High	Low	Close	Average	Adj Close	Volume	Change
Date								
2023-01-30	144.960007	145.550003	142.850006	143.000000	144.200005	142.781998	64015300	-1.960007
2023-01-31	142.699997	144.339996	142.279999	144.289993	143.309998	144.070023	65874500	1.589996

```
In [ ]: # select rows by criteria
df.query('Change > 0.0') # 'Open > 150 and Change/Open > 0.1'
```

```
Out[ ]:
```

	Open	High	Low	Close	Average	Adj Close	Volume	Change
Date								
2023-01-31	142.699997	144.339996	142.279999	144.289993	143.309998	144.070023	65874500	1.589996
2023-02-01	143.970001	146.610001	141.320007	145.429993	143.965004	145.208282	77663600	1.459991
2023-02-02	148.899994	151.179993	148.169998	150.820007	149.674995	150.590088	118339000	1.920013

```
In [ ]: # add row
df.loc['diff'] = df.loc['2023-02-02'] - df.loc['2023-02-01']
```

```
In [ ]: df.tail(3)
```

```
Out[ ]:
```

	Open	High	Low	Close	Average	Adj Close	Volume	Change
Date								
2023-02-01 00:00:00	143.970001	146.610001	141.320007	145.429993	143.965004	145.208282	77663600.0	1.459991
2023-02-02 00:00:00	148.899994	151.179993	148.169998	150.820007	149.674995	150.590088	118339000.0	1.920013
diff	4.929993	4.569992	6.849991	5.390015	5.709991	5.381805	40675400.0	0.460022

```
In [ ]: df.drop('diff',axis=0, inplace=True)
```

```
In [ ]: df.tail(2)
```

```
Out[ ]:
```

	Open	High	Low	Close	Average	Adj Close	Volume	Change
Date								
2023-02-01 00:00:00	143.970001	146.610001	141.320007	145.429993	143.965004	145.208282	77663600.0	1.459991
2023-02-02 00:00:00	148.899994	151.179993	148.169998	150.820007	149.674995	150.590088	118339000.0	1.920013

```
In [ ]: # fill missing data
df.iloc[0,0] = np.nan
df.iloc[2,3] = np.nan
df.head(4)
```

Out[]:

	Open	High	Low	Close	Average	Adj Close	Volume	Change
Date								
2023-01-30 00:00:00	NaN	145.550003	142.850006	143.000000	144.200005	142.781998	64015300.0	-1.960007
2023-01-31 00:00:00	142.699997	144.339996	142.279999	144.289993	143.309998	144.070023	65874500.0	1.589996
2023-02-01 00:00:00	143.970001	146.610001	141.320007	NaN	143.965004	145.208282	77663600.0	1.459991
2023-02-02 00:00:00	148.899994	151.179993	148.169998	150.820007	149.674995	150.590088	118339000.0	1.920013

```
In [ ]: df.fillna(method='ffill', inplace=True)
```

```
In [ ]: df.head(4)
```

Out[]:

	Open	High	Low	Close	Average	Adj Close	Volume	Change
Date								
2023-01-30 00:00:00	NaN	145.550003	142.850006	143.000000	144.200005	142.781998	64015300.0	-1.960007
2023-01-31 00:00:00	142.699997	144.339996	142.279999	144.289993	143.309998	144.070023	65874500.0	1.589996
2023-02-01 00:00:00	143.970001	146.610001	141.320007	144.289993	143.965004	145.208282	77663600.0	1.459991
2023-02-02 00:00:00	148.899994	151.179993	148.169998	150.820007	149.674995	150.590088	118339000.0	1.920013

```
In [ ]: df.to_csv('aapl.csv')
```

```
In [ ]: df.describe()
```

Out[]:

	Open	High	Low	Close	Average	Adj Close	Volume	Change
count	3.000000	4.000000	4.000000	4.000000	4.000000	4.000000	4.000000e+00	4.000000
mean	145.189997	146.919998	143.655003	145.599998	145.287500	145.662598	8.147310e+07	0.752499
std	3.275101	2.987583	3.075497	3.532738	2.949140	3.431264	2.530941e+07	1.818674
min	142.699997	144.339996	141.320007	143.000000	143.309998	142.781998	6.401530e+07	-1.960007
25%	143.334999	145.247501	142.040001	143.967495	143.801252	143.748016	6.540970e+07	0.604992
50%	143.970001	146.080002	142.565002	144.289993	144.082504	144.639153	7.176905e+07	1.524994
75%	146.434998	147.752499	144.180004	145.922497	145.568752	146.553734	8.783245e+07	1.672501
max	148.899994	151.179993	148.169998	150.820007	149.674995	150.590088	1.183390e+08	1.920013

```
In [ ]: d = pd.read_csv('fx_trades.csv')
```

```
In [ ]: d.head(5)
```

```
Out[ ]:
```

	TradeID	Type	Date	Ccy	Amt
0	FX200171	Spot	2/16/2021	EUR	1000000
1	FX200171	Spot	2/16/2021	USD	-1100000
2	FX200172	Spot	2/16/2021	GBP	-1000000
3	FX200172	Spot	2/16/2021	EUR	1180000
4	FX200173	Forward	3/16/2021	GBP	2000000

```
In [ ]: g1 = d.groupby(['Ccy']) # keep grouping of rows
```

```
In [ ]: g1.get_group('EUR')
```

```
Out[ ]:
```

	TradeID	Type	Date	Ccy	Amt
0	FX200171	Spot	2/16/2021	EUR	1000000
3	FX200172	Spot	2/16/2021	EUR	1180000
8	FX200175	NDF	4/16/2021	EUR	-200000
15	FX200178	Forward	5/16/2021	EUR	-1170000
16	FX200179	Forward	5/16/2021	EUR	-1000000

```
In [ ]: # aggregate the transactions into positions by currency
g1.sum()
```

```
Out[ ]:
```

	Amt
Ccy	
BRL	-349000
CAD	6750000
EUR	-190000
GBP	0
USD	-4660000

```
In [ ]: g2 = d.groupby(['Ccy', 'Date'])
```

In []:

g2.sum()

Out[]:

		Amt
Ccy	Date	
BRL	4/16/2021	-349000
CAD	5/16/2021	6750000
EUR	2/16/2021	2180000
	4/16/2021	-200000
	5/16/2021	-2170000
GBP	2/16/2021	-1000000
	3/16/2021	2000000
	5/16/2021	-1000000
USD	2/16/2021	-1100000
	3/16/2021	-2600000
	4/16/2021	300000
	5/16/2021	-1260000

In []: