SETUP

```
import pandas as pd
df = pd.read_csv('/content/nyc_temperatures.csv')
df.head()
```

	date	datatype	station	attributes	value
0	2018-10-01T00:00:00	TAVG	GHCND:USW00014732	H,,S,	21.2
1	2018-10-01T00:00:00	TMAX	GHCND:USW00014732	,,W,2400	25.6
2	2018-10-01T00:00:00	TMIN	GHCND:USW00014732	,,W,2400	18.3
3	2018-10-02T00:00:00	TAVG	GHCND:USW00014732	H,,S,	22.7
4	2018-10-02T00:00:00	TMAX	GHCND:USW00014732	,,W,2400	26.1

RENAMING COLUMNS

```
df.columns
```

```
Index(['date', 'datatype', 'station', 'attributes', 'value'], dtype='object')
```

```
df.rename(columns={'value':'temp_C','attributes':'flags'}, inplace=True)
```

```
df.columns
```

```
Index(['date', 'datatype', 'station', 'flags', 'temp_C'], dtype='object')
```

```
df.rename(str.upper, axis='columns').columns
```

```
Index(['DATE', 'DATATYPE', 'STATION', 'FLAGS', 'TEMP_C'], dtype='object')
```

TYPE CONVERSION

```
df.dtypes
```

```
date         object
datatype     object
station      object
flags        object
temp_C      float64
dtype: object
```

```
df.loc[:,'date'] = pd.to_datetime(df.date)
df.dtypes
```

```
<ipython-input-7-80606e5f8dec>:1: DeprecationWarning: In a future version, `df.iloc[:, i] = newvals` will attempt to set the
  df.loc[:,'date'] = pd.to_datetime(df.date)
date        datetime64[ns]
datatype            object
station             object
flags               object
temp_C             float64
dtype: object
```

```
df.date.describe()
```

```
<ipython-input-8-f7d3fa946723>:1: FutureWarning: Treating datetime data as categorical rather than numeric in `.describe` is
  df.date.describe()
count                      93
unique                     31
top       2018-10-01 00:00:00
freq                        3
first     2018-10-01 00:00:00
last      2018-10-31 00:00:00
Name: date, dtype: object
```

```
1 pd.date_range(start='2018-10-25', periods=2, freq='D').tz_localize('EST')
```

```
DatetimeIndex(['2018-10-25 00:00:00-05:00', '2018-10-26 00:00:00-05:00'], dtype='datetime64[ns, EST]', freq=None)
```

THE FREQUENCY DOES NOT CHANGE

```
1 eastern = pd.read_csv('/content/nyc_temperatures.csv', index_col='date', parse_dates=True).tz_localize('EST')
2 eastern.head()
```

date	datatype	station	attributes	value
2018-10-01 00:00:00-05:00	TAVG	GHCND:USW00014732	H,,S,	21.2
2018-10-01 00:00:00-05:00	TMAX	GHCND:USW00014732	,,W,2400	25.6
2018-10-01 00:00:00-05:00	TMIN	GHCND:USW00014732	,,W,2400	18.3
2018-10-02 00:00:00-05:00	TAVG	GHCND:USW00014732	H,,S,	22.7
2018-10-02 00:00:00-05:00	TMAX	GHCND:USW00014732	,,W,2400	26.1

```
1 eastern.tz_convert('UTC').head()
```

date	datatype	station	attributes	value
2018-10-01 05:00:00+00:00	TAVG	GHCND:USW00014732	H,,S,	21.2
2018-10-01 05:00:00+00:00	TMAX	GHCND:USW00014732	,,W,2400	25.6
2018-10-01 05:00:00+00:00	TMIN	GHCND:USW00014732	,,W,2400	18.3
2018-10-02 05:00:00+00:00	TAVG	GHCND:USW00014732	H,,S,	22.7
2018-10-02 05:00:00+00:00	TMAX	GHCND:USW00014732	,,W,2400	26.1

```
1 eastern.to_period('M').index
```

```
<ipython-input-13-34a82283fe40>:1: UserWarning: Converting to PeriodArray/Index representation will drop timezone informatio
  eastern.to_period('M').index
PeriodIndex(['2018-10', '2018-10', '2018-10', '2018-10', '2018-10', '2018-10',
             '2018-10', '2018-10', '2018-10', '2018-10', '2018-10', '2018-10',
             '2018-10', '2018-10', '2018-10', '2018-10', '2018-10', '2018-10',
             '2018-10', '2018-10', '2018-10', '2018-10', '2018-10', '2018-10',
             '2018-10', '2018-10', '2018-10', '2018-10', '2018-10', '2018-10',
             '2018-10', '2018-10', '2018-10', '2018-10', '2018-10', '2018-10',
             '2018-10', '2018-10', '2018-10', '2018-10', '2018-10', '2018-10',
             '2018-10', '2018-10', '2018-10', '2018-10', '2018-10', '2018-10',
             '2018-10', '2018-10', '2018-10', '2018-10', '2018-10', '2018-10',
             '2018-10', '2018-10', '2018-10', '2018-10', '2018-10', '2018-10',
             '2018-10', '2018-10', '2018-10', '2018-10', '2018-10', '2018-10',
             '2018-10', '2018-10', '2018-10', '2018-10', '2018-10', '2018-10',
             '2018-10', '2018-10', '2018-10', '2018-10', '2018-10', '2018-10',
             '2018-10', '2018-10', '2018-10', '2018-10', '2018-10', '2018-10',
             '2018-10', '2018-10', '2018-10', '2018-10', '2018-10', '2018-10',
             '2018-10', '2018-10', '2018-10'],
            dtype='period[M]', name='date')
```

```
1 eastern.to_period('M').to_timestamp().index
```

```
<ipython-input-14-22abc5f95bfc>:1: UserWarning: Converting to PeriodArray/Index representation will drop timezone informatio
  eastern.to_period('M').to_timestamp().index
DatetimeIndex(['2018-10-01', '2018-10-01', '2018-10-01', '2018-10-01',
               '2018-10-01', '2018-10-01', '2018-10-01', '2018-10-01',
               '2018-10-01', '2018-10-01', '2018-10-01', '2018-10-01',
               '2018-10-01', '2018-10-01', '2018-10-01', '2018-10-01',
               '2018-10-01', '2018-10-01', '2018-10-01', '2018-10-01',
               '2018-10-01', '2018-10-01', '2018-10-01', '2018-10-01',
               '2018-10-01', '2018-10-01', '2018-10-01', '2018-10-01',
               '2018-10-01', '2018-10-01', '2018-10-01', '2018-10-01',
               '2018-10-01', '2018-10-01', '2018-10-01', '2018-10-01',
               '2018-10-01', '2018-10-01', '2018-10-01', '2018-10-01',
               '2018-10-01', '2018-10-01', '2018-10-01', '2018-10-01',
```

```
               '2018-10-01', '2018-10-01', '2018-10-01', '2018-10-01',
               '2018-10-01', '2018-10-01', '2018-10-01', '2018-10-01',
               '2018-10-01', '2018-10-01', '2018-10-01', '2018-10-01',
               '2018-10-01', '2018-10-01', '2018-10-01', '2018-10-01',
               '2018-10-01', '2018-10-01', '2018-10-01', '2018-10-01',
               '2018-10-01', '2018-10-01', '2018-10-01', '2018-10-01',
               '2018-10-01', '2018-10-01', '2018-10-01', '2018-10-01',
               '2018-10-01', '2018-10-01', '2018-10-01', '2018-10-01',
               '2018-10-01', '2018-10-01', '2018-10-01', '2018-10-01',
               '2018-10-01', '2018-10-01', '2018-10-01', '2018-10-01',
               '2018-10-01', '2018-10-01', '2018-10-01', '2018-10-01',
               '2018-10-01', '2018-10-01', '2018-10-01', '2018-10-01',
               '2018-10-01'],
              dtype='datetime64[ns]', name='date', freq=None)
```

```python
df = pd.read_csv('/content/nyc_temperatures.csv').rename(columns={'value' : 'temp_C', 'attributes' : 'flags'})
new_df = df.assign(date=pd.to_datetime(df.date), temp_F=(df.temp_C * 9/5) + 32)
new_df.dtypes
```

```
date        datetime64[ns]
datatype            object
station             object
flags               object
temp_C             float64
temp_F             float64
dtype: object
```

```python
new_df.head()
```

	date	datatype	station	flags	temp_C	temp_F
0	2018-10-01	TAVG	GHCND:USW00014732	H,,S,	21.2	70.16
1	2018-10-01	TMAX	GHCND:USW00014732	,,W,2400	25.6	78.08
2	2018-10-01	TMIN	GHCND:USW00014732	,,W,2400	18.3	64.94
3	2018-10-02	TAVG	GHCND:USW00014732	H,,S,	22.7	72.86
4	2018-10-02	TMAX	GHCND:USW00014732	,,W,2400	26.1	78.98

```python
df = df.assign(date=pd.to_datetime(df.date), temp_C_whole=df.temp_C.astype('int'), temp_F=(df.temp_C * 9/5) + 32, temp_F_whole=lambda x
df.head()
```

	date	datatype	station	flags	temp_C	temp_C_whole	temp_F
0	2018-10-01	TAVG	GHCND:USW00014732	H,,S,	21.2	21	70.16
1	2018-10-01	TMAX	GHCND:USW00014732	,,W,2400	25.6	25	78.08
2	2018-10-01	TMIN	GHCND:USW00014732	,,W,2400	18.3	18	64.94

```python
df_with_categories = df.assign(station=df.station.astype('category'), datatype=df.datatype.astype('category'))
df_with_categories.dtypes
```

```
date            datetime64[ns]
datatype              category
station               category
flags                   object
temp_C                 float64
temp_C_whole             int64
temp_F                 float64
temp_F_whole             int64
dtype: object
```

```python
pd.Categorical(['med', 'med', 'low', 'high'], categories=['low', 'med', 'high'], ordered=True)
```

```
['med', 'med', 'low', 'high']
Categories (3, object): ['low' < 'med' < 'high']
```

REORDERING, REINDEXING, AND SORTING

```
1 df.sort_values(by='temp_C', ascending=False).head(10)
```

	date	datatype	station	flags	temp_C	temp_C_whole	temp_F
19	2018-10-07	TMAX	GHCND:USW00014732	,,W,2400	27.8	27	82.04
28	2018-10-10	TMAX	GHCND:USW00014732	,,W,2400	27.8	27	82.04
31	2018-10-11	TMAX	GHCND:USW00014732	,,W,2400	26.7	26	80.06
4	2018-10-02	TMAX	GHCND:USW00014732	,,W,2400	26.1	26	78.98
10	2018-10-04	TMAX	GHCND:USW00014732	,,W,2400	26.1	26	78.98
25	2018-10-09	TMAX	GHCND:USW00014732	,,W,2400	25.6	25	78.08
	2018						

```
1 df.sort_values(by=['temp_C', 'date'], ascending=False).head(10)
```

	date	datatype	station	flags	temp_C	temp_C_whole	temp_F
28	2018-10-10	TMAX	GHCND:USW00014732	,,W,2400	27.8	27	82.04
19	2018-10-07	TMAX	GHCND:USW00014732	,,W,2400	27.8	27	82.04
31	2018-10-11	TMAX	GHCND:USW00014732	,,W,2400	26.7	26	80.06
10	2018-10-04	TMAX	GHCND:USW00014732	,,W,2400	26.1	26	78.98
4	2018-10-02	TMAX	GHCND:USW00014732	,,W,2400	26.1	26	78.98
25	2018-10-09	TMAX	GHCND:USW00014732	,,W,2400	25.6	25	78.08
	2018						

```
1 df.nlargest(n=5, columns='temp_C')
```

	date	datatype	station	flags	temp_C	temp_C_whole	temp_F
19	2018-10-07	TMAX	GHCND:USW00014732	,,W,2400	27.8	27	82.04
28	2018-10-10	TMAX	GHCND:USW00014732	,,W,2400	27.8	27	82.04
31	2018-10-11	TMAX	GHCND:USW00014732	,,W,2400	26.7	26	80.06

```
1 df.nsmallest(n=5, columns=['temp_C', 'date'])
```

	date	datatype	station	flags	temp_C	temp_C_whole	temp_F
65	2018-10-22	TMIN	GHCND:USW00014732	,,W,2400	5.6	5	42.08
77	2018-10-26	TMIN	GHCND:USW00014732	,,W,2400	5.6	5	42.08
62	2018-10-21	TMIN	GHCND:USW00014732	,,W,2400	6.1	6	42.98

```
1 df.sample(5, random_state=0).index
```

```
Int64Index([2, 30, 55, 16, 13], dtype='int64')
```

```
1 df.sample(5, random_state=0).sort_index().index
```

```
Int64Index([2, 13, 16, 30, 55], dtype='int64')
```

```
1 df.sort_index(axis=1).head()
```

	datatype	date	flags	station	temp_C	temp_C_whole	temp_F
0	TAVG	2018-10-01	H,,S,	GHCND:USW00014732	21.2	21	70.16
1	TMAX	2018-10-01	,,W,2400	GHCND:USW00014732	25.6	25	78.08
2	TMIN	2018-10-01	,,W,2400	GHCND:USW00014732	18.3	18	64.94

```
1 df.sort_index(axis=1).head().loc[:,'temp_C':'temp_F_whole']
```

	temp_C	temp_C_whole	temp_F	temp_F_whole
0	21.2	21	70.16	70
1	25.6	25	78.08	78
2	18.3	18	64.94	64
3	22.7	22	72.86	72
4	26.1	26	78.98	78

```
1 df.equals(df.sort_values(by='temp_C'))
```

```
False
```

```
1 df.equals(df.sort_values(by='temp_C').sort_index())
```

```
True
```

```
1 df[df.datatype == 'TAVG'].head().reset_index()
```

	index	date	datatype	station	flags	temp_C	temp_C_whole	te
0	0	2018-10-01	TAVG	GHCND:USW00014732	H,,S,	21.2	21	
1	3	2018-10-02	TAVG	GHCND:USW00014732	H,,S,	22.7	22	
2	6	2018-10-03	TAVG	GHCND:USW00014732	H,,S,	21.8	21	

```
1 df.set_index('date', inplace=True)
2 df.head()
```

date	datatype	station	flags	temp_C	temp_C_whole	temp_F	te
2018-10-01	TAVG	GHCND:USW00014732	H,,S,	21.2	21	70.16	
2018-10-01	TMAX	GHCND:USW00014732	,,W,2400	25.6	25	78.08	
2018-10-01	TMIN	GHCND:USW00014732	,,W,2400	18.3	18	64.94	

```
df['2018-10-11':'2018-10-12']
```

date	datatype	station	flags	temp_C	temp_C_whole	temp_F	te
2018-10-11	TAVG	GHCND:USW00014732	H,,S,	23.4	23	74.12	
2018-10-11	TMAX	GHCND:USW00014732	,,W,2400	26.7	26	80.06	
2018-10-11	TMIN	GHCND:USW00014732	,,W,2400	21.7	21	71.06	
2018-10-12	TAVG	GHCND:USW00014732	H,,S,	18.3	18	64.94	

```
sp = pd.read_csv('/content/sp500.csv', index_col='date', parse_dates=True).drop(columns=['adj_close'])
sp.head(10).assign(day_of_week=lambda x: x.index.day_name())
```

date	high	low	open	close	volume	day_of_week
2017-01-03	2263.879883	2245.129883	2251.570068	2257.830078	3770530000	Tuesday
2017-01-04	2272.820068	2261.600098	2261.600098	2270.750000	3764890000	Wednesday
2017-01-05	2271.500000	2260.449951	2268.179932	2269.000000	3761820000	Thursday
2017-01-06	2282.100098	2264.060059	2271.139893	2276.979980	3339890000	Friday
2017-01-09	2275.489990	2268.899902	2273.590088	2268.899902	3217610000	Monday
2017-01-10	2279.270020	2265.270020	2269.719971	2268.899902	3638790000	Tuesday

Next steps: View recommended plots

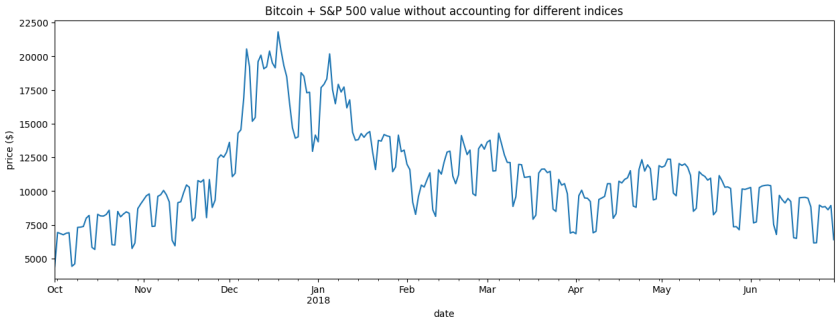
```
bitcoin = pd.read_csv('/content/bitcoin.csv', index_col='date', parse_dates=True).drop(columns=['market_cap'])
portfolio = pd.concat([sp, bitcoin], sort=False).groupby(pd.Grouper(freq='D')).sum()
portfolio.head(10).assign(day_of_week=lambda x: x.index.day_name())
```

date	high	low	open	close	volume	day_of_week
2017-01-01	1003.080000	958.700000	963.660000	998.330000	147775008	Sunday
2017-01-02	1031.390000	996.700000	998.620000	1021.750000	222184992	Monday
2017-01-03	3307.959883	3266.729883	3273.170068	3301.670078	3955698000	Tuesday
2017-01-04	3432.240068	3306.000098	3306.000098	3425.480000	4109835984	Wednesday
2017-01-05	3462.600000	3170.869951	3424.909932	3282.380000	4272019008	Thursday
2017-01-06	3328.910098	3148.000059	3285.379893	3179.179980	3691766000	Friday
2017-						

Next steps: View recommended plots

```
1 import matplotlib.pyplot as plt
```

```
1 portfolio['2017-Q4':'2018-Q2'].plot(y='close', figsize=(15, 5), legend=False, title='Bitcoin + S&P 500 value without accounting for dif
2 plt.ylabel('price ($)')
3 plt.show()
```




```
1 sp.reindex(bitcoin.index).head(10).assign(day_of_week=lambda x: x.index.day_name())
```

date	high	low	open	close	volume	day_of_week

```
1 sp.reindex(bitcoin.index, method='ffill').head(10).assign(day_of_week=lambda x: x.index.day_name())
```

date	high	low	open	close	volume	day_of_week
2017-01-01	NaN	NaN	NaN	NaN	NaN	Sunda
2017-01-02	NaN	NaN	NaN	NaN	NaN	Monda
2017-01-03	2263.879883	2245.129883	2251.570068	2257.830078	3.770530e+09	Tuesda
2017-01-04	2272.820068	2261.600098	2261.600098	2270.750000	3.764890e+09	Wednesda
2017-01-05	2271.500000	2260.449951	2268.179932	2269.000000	3.761820e+09	Thursda
2017-01-06	2282.100098	2264.060059	2271.139893	2276.979980	3.339890e+09	Frida
2017-						

```
1 import numpy as np
2 sp_reindexed = sp.reindex(bitcoin.index).assign(volume=lambda x: x.volume.fillna(0), close=lambda x: x.close.fillna(method='ffill'), op
3 sp_reindexed.head(10).assign(day_of_week=lambda x: x.index.day_name())
```

date	high	low	open	close	volume	day_of_week
2017-01-01	NaN	NaN	NaN	NaN	0.000000e+00	Sunda
2017-01-02	NaN	NaN	NaN	NaN	0.000000e+00	Monda
2017-01-03	2263.879883	2245.129883	2251.570068	2257.830078	3.770530e+09	Tuesda
2017-01-04	2272.820068	2261.600098	2261.600098	2270.750000	3.764890e+09	Wednesda
2017-01-05	2271.500000	2260.449951	2268.179932	2269.000000	3.761820e+09	Thursda
2017-01-06	2282.100098	2264.060059	2271.139893	2276.979980	3.339890e+09	Frida
2017-						

Next steps: View recommended plots

```
1 fixed_portfolio = pd.concat([sp_reindexed, bitcoin], sort=False).groupby(pd.Grouper(freq='D')).sum()
2 ax = fixed_portfolio['2017-Q4':'2018-Q2'].plot(y='close', label='reindexed portfolio of S&P 500 + Bitcoin', figsize=(15, 5), linewidth=2,
3 portfolio['2017-Q4':'2018-Q2'].plot(y='close', ax=ax, linestyle='--', label='portfolio of S&P 500 + Bitcoin w/o reindexing').set_ylabel('
4 plt.show()
```