time_series

October 22, 2020

[1]:
```
import pandas as pd
%pylab inline
pylab.rcParams['figure.figsize'] = (14,6)
```

```
/Users/illarion.khliestov/Projects/data_mining/.venv/lib/python3.6/site-
packages/pandas/compat/__init__.py:120: UserWarning: Could not import the lzma
module. Your installed Python is incomplete. Attempting to use lzma compression
will result in a RuntimeError.
  warnings.warn(msg)

Populating the interactive namespace from numpy and matplotlib
```

[2]:
```
# data dumped from https://www.investing.com/currencies/usd-eur-historical-data
base_file = 'us_500_futures.csv'
# base_file = 'usd_eur_data.csv'
```

[3]:
```
# load data and get some information about it
df = pd.read_csv(base_file, index_col='Date')
print("Shape:", df.shape)
print("Columns:", df.columns)
```

```
Shape: (2600, 6)
Columns: Index(['Price', 'Open', 'High', 'Low', 'Vol.', 'Change %'],
dtype='object')
```

[4]:
```
# Drop unecessary columnts
columns_to_drop = [c for c in df.columns if c not in ['Date', 'Price']]
df = df.drop(columns_to_drop, axis=1)
print("Columns:", df.columns)
df.head(5)
```

```
Columns: Index(['Price'], dtype='object')
```

[4]:

Date	Price
Feb 21, 2020	3,339.25
Feb 20, 2020	3,369.25
Feb 19, 2020	3,387.25

```
Feb 18, 2020  3,369.25
Feb 17, 2020  3,376.00
```

[5]:
```
# change Data order, so the oldest entries will be the first one
df = df.reindex(index=df.index[::-1])
# make sure prices are numeric
df['Price'] = pd.to_numeric(df['Price'].apply(lambda x: x.replace(',', '')))
# and convert date to datetime format
df.index = pd.to_datetime(df.index)
df.head(5)
```

[5]:
```
              Price
Date
2010-02-22  1107.50
2010-02-23  1097.25
2010-02-24  1103.50
2010-02-25  1102.25
2010-02-26  1103.50
```

[6]:
```
# add missed entries(holidays) to dataframe
full_time_range = pd.date_range(start=df.index[0], end=df.index[-1])
df = df.reindex(index=full_time_range)
print(df.head(10))
df = df.interpolate()
print("\nAfter interpolation:")
print(df.head(10))
```

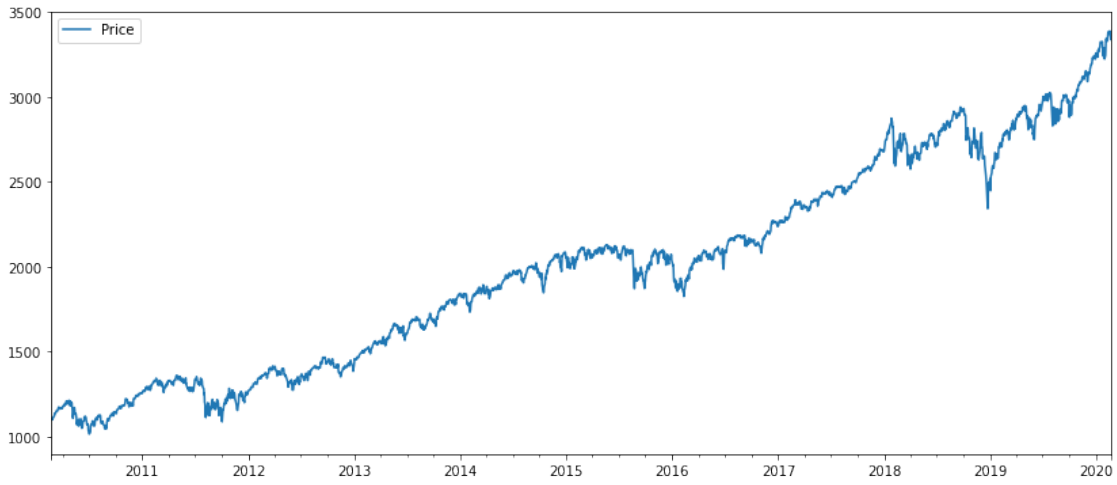
```
              Price
2010-02-22  1107.50
2010-02-23  1097.25
2010-02-24  1103.50
2010-02-25  1102.25
2010-02-26  1103.50
2010-02-27      NaN
2010-02-28      NaN
2010-03-01  1114.50
2010-03-02  1117.50
2010-03-03  1118.50

After interpolation:
                 Price
2010-02-22  1107.500000
2010-02-23  1097.250000
2010-02-24  1103.500000
2010-02-25  1102.250000
2010-02-26  1103.500000
2010-02-27  1107.166667
```

```
2010-02-28  1110.833333
2010-03-01  1114.500000
2010-03-02  1117.500000
2010-03-03  1118.500000
```

[7]:
```
df.plot()
```

[7]: <AxesSubplot:>



[12]:
```
def list_available_years(df):
    return sorted(list(set([idx.year for idx in df.index])))
```

[13]:
```
df[df[inde]]
```

[13]: [2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020]