

Chapter 05 - Task 02 (chap 05 exercises)

Setup

```
%matplotlib inline
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

fb = pd.read_csv('data/fb_stock_prices_2018.csv', parse_dates=['date'], index_col='date')
quakes = pd.read_csv('data/earthquakes.csv')
covid = pd.read_csv('data/covid19_cases.csv', parse_dates=['dateRep'])
```

Exercise 01

Plot the rolling 20-day minimum of the Facebook closing price using pandas.

```
fb.close.rolling('20D').min().plot(title='rolling 20 day min')
```

exercise 01

Exercise 02

Create a histogram and KDE of the change from open to close in the price of Facebook stock.

```
# difference between open and close
fb['opentoclose'] = fb.open - fb.close
# create the histogram. density=true is required for the kde to show up properly
hist = fb['opentoclose'].plot(kind='hist', density=True)
fb['opentoclose'].plot(kind='kde', ax=hist)
```

exercise 02

Exercise 03

Using the earthquake data, create box plots for the magnitudes of each magType used in Indonesia.

```
# make a variable to use in data argument later
indonesia_quakes = quakes.query('parsed_place == "Indonesia"')
```

```
# seaborn to create a box plot where mag is y and each magType is on the x axis
sns.catplot(x='magType', y='mag', data=indonesia_quakes, kind='box')
```

exercise 03

Exercise 04

Make a line plot of the difference between the weekly maximum high price and the weekly minimum low price for Facebook. This should be a single line.

```
# resample data so that it is weekly instead of daily
fb_resample = fb.resample('1W').agg({'high': 'max', 'low': 'min'})
# find the difference between high and low
weekly_diff = fb_resample['high'] - fb_resample['low']
# plot it on a line
weekly_diff.plot(kind='line', title='Weekly Max High Price - Min Low Price')
```

exercise 04

Exercise 05

Plot the 14-day moving average of the daily change in new COVID-19 cases in Brazil, China, India, Italy, Spain, and the USA: a) First, use the diff() method that was introduced in the Working with time series data section of Chapter 4, Aggregating Pandas DataFrames, to calculate the day-over-day change in new cases. Then, use rolling() to calculate the 14-day moving average. b) Make three subplots: one for China; one for Spain and Italy; and one for Brazil, India, and the USA.

My results looked off and were definitely different when compared to the solution adding in the date s below helped a bit, but it still looks wrong. The data file in chap06 might be different from what they used because i also had to recreate the date column which we created in previous chapters. I don't see it being created in this chapter even though it is used.

I noticed some large entries in the data set that could throw off the rolling average a bit, but not that much. I also changed the overall date to search by to better match what the solution looked like for no luck

Section 2 of chapter 5 also doesnt have this issue

```
# Read the data and set the date column as the index
covid = pd.read_csv('data/covid19_cases.csv')
# create date column and make that the index
```

```

covid['date'] = pd.to_datetime(covid['dateRep'])
covid.set_index('date', inplace=True)
# Makes it easier down the line
covid.replace({'countriesAndTerritories': {'United_States_of_America': 'USA'}}, inplace=True)

# select dates to plot
covid = covid.loc['2020-01-01':'2020-10-01']

rolling_14day = covid.pivot_table(
    index=covid.index, columns=['countriesAndTerritories'], values='cases').apply(
        lambda x: x.diff().rolling(14).mean()
    )

# create the subplots
fig, axes = plt.subplots(1, 3, figsize=(20, 5))

# put data on each plot (axes)
rolling_14day[['China']].plot(
    ax=axes[0], title='China', style='-.c')
rolling_14day[['Italy', 'Spain']].plot(
    ax=axes[1], title='Italy & Spain', style=['-', '--'])
rolling_14day[['Brazil', 'India', 'USA']].plot(
    ax=axes[2], title='Brazil, India & USA', style=['--', ':', '-'])

```

exercise 05

Exercise 06

Using matplotlib and pandas, create two subplots side-by-side showing the effect that after-hours trading has had on Facebook's stock prices: a) The first subplot will contain a line plot of the daily difference between that day's opening price and the prior day's closing price (be sure to review the Working with time series data section of Chapter 4, Aggregating Pandas DataFrames, for an easy way to do this). b) The second subplot will be a bar plot showing the net effect this had monthly, using `resample()`. Further reading 321 c) Bonus #1: Color the bars according to whether there are gains in the stock price (green) or drops in the stock price (red). d) Bonus #2: Modify the x-axis of the bar plot to show the three-letter abbreviation for the month.

```

fb = pd.read_csv('data/fb_stock_prices_2018.csv', parse_dates=['date'], index_col='date')

# a) create the daily difference between open and previous close
daily_diff = fb['open'] - fb['close'].shift(1)

```

```

# b) monthly resample of daily_diff
monthly_resample = daily_diff.resample('1M').sum()

fig, axes = plt.subplots(1, 2, figsize=(15, 5))

daily_diff.plot(ax=axes[0])
# d) rename columns to month 3 letter
# monthly_resample.index = monthly_resample.index.strftime('%b')
monthly_resample.plot(ax=axes[1],
                      kind='bar',
                      # c) color the bars
                      color=np.where(monthly_resample <= 0, 'r', 'g'))
# d) method i like more for renaming the month column
axes[1].set_xticklabels(monthly_resample.index.strftime('%b'));

```

exercise 06

with using `axes[1].set_xticklabels(monthly_resample.index.strftime('%b'))` it would normally output the below txt code block (atleast with using `%matplotlib inline`). Appending a semicolon ; to the end of the line supresses this output. The book suggested method to change the x tick labels does not have to worry about this.

```

[Text(0, 0, 'Jan'),
 Text(1, 0, 'Feb'),
 Text(2, 0, 'Mar'),
 Text(3, 0, 'Apr'),
 Text(4, 0, 'May'),
 Text(5, 0, 'Jun'),
 Text(6, 0, 'Jul'),
 Text(7, 0, 'Aug'),
 Text(8, 0, 'Sep'),
 Text(9, 0, 'Oct'),
 Text(10, 0, 'Nov'),
 Text(11, 0, 'Dec')]

```