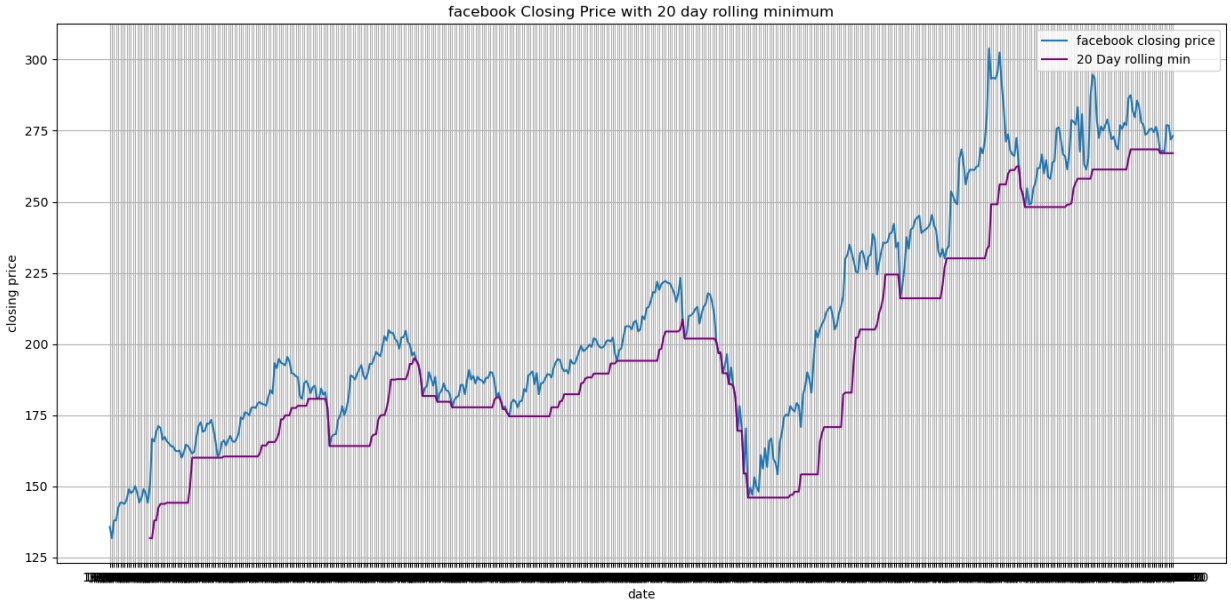
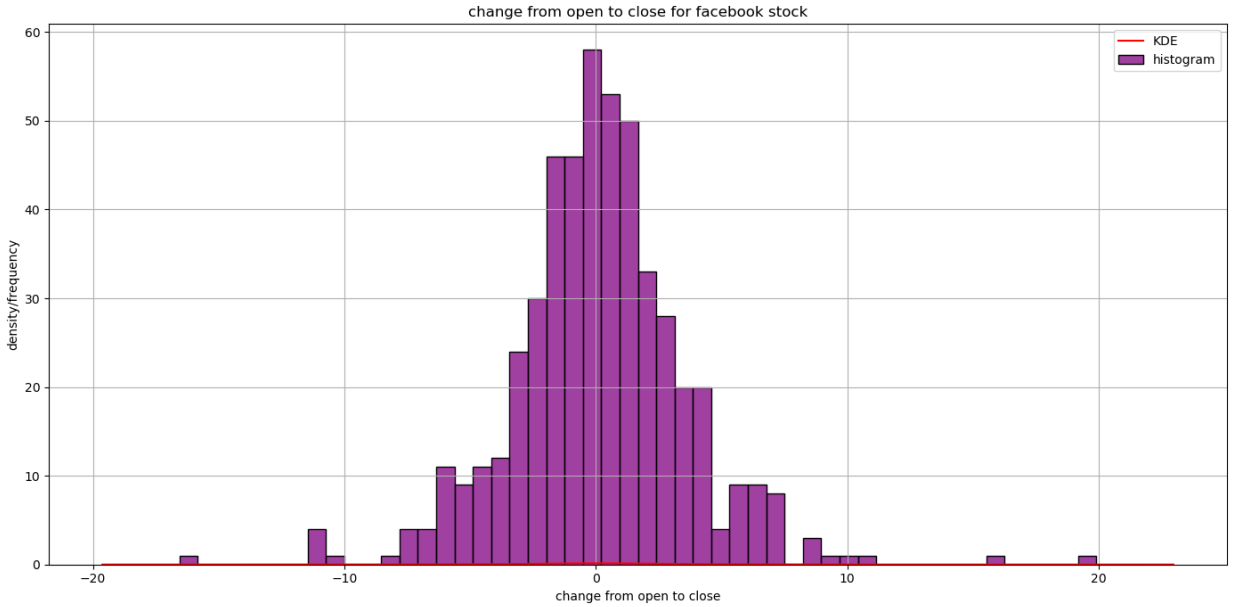
Hands-On Data Analysis with Python (2nd Edition): Page 320, Exercises 1-4, and 6

In [1]:
```
#1. Plot the rolling 20-day minimum of the Facebook closing price using pandas.

#load pd, plt
import pandas as pd
import matplotlib.pyplot as plt

#facebook.csv
df = pd.read_csv('facebook.csv')

#column close
rolling_min = df['close'].rolling(window=20).min()

#plot 20 day close
plt.figure(figsize=(14, 7))
plt.plot(df['date'], df['close'], label='facebook closing price')
plt.plot(df['date'], rolling_min, label='20 Day rolling min', color='purple')
plt.title('facebook Closing Price with 20 day rolling minimum')
plt.xlabel('date')
plt.ylabel('closing price')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```



In [3]:
```
#2. Create a histogram and KDE of the change from open to close in the price ofFaceboo

#load pd, plt, sns
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

#facebook.csv
df = pd.read_csv('facebook.csv')

#change from open to close
df['Change'] = df['close'] - df['open']
```

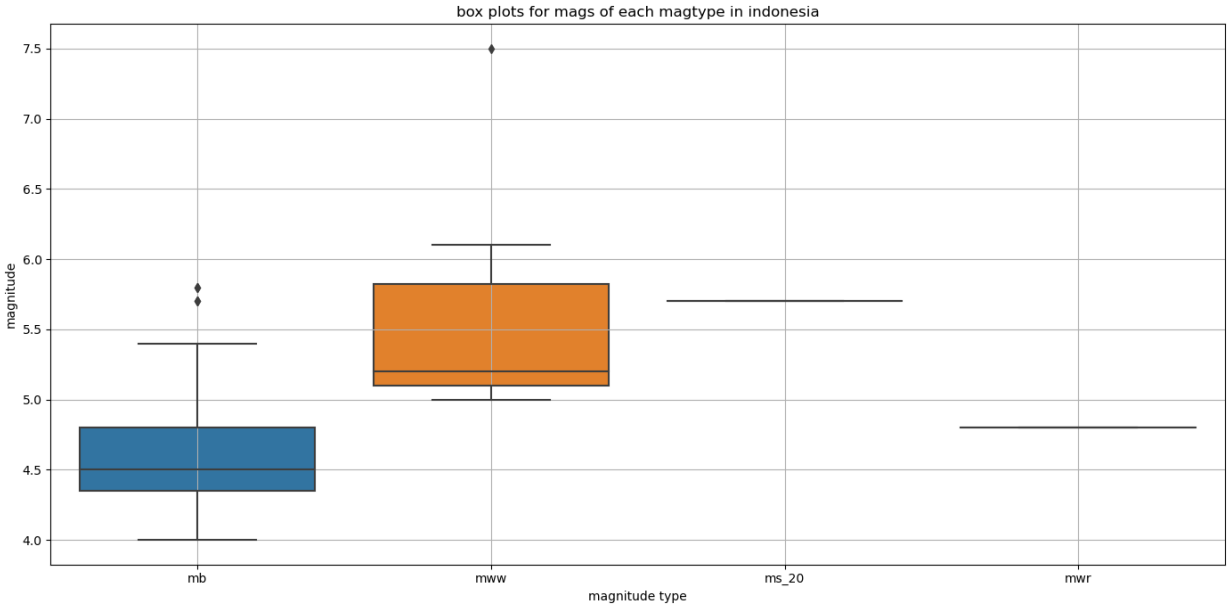
```
plt.figure(figsize=(14, 7))

#histogra
sns.histplot(df['Change'], bins=50, kde=False, color='purple', label='histogram')

#kde
sns.kdeplot(df['Change'], color='red', label='KDE')

plt.title('change from open to close for facebook stock')
plt.xlabel('change from open to close')
plt.ylabel('density/frequency')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```



In [4]:

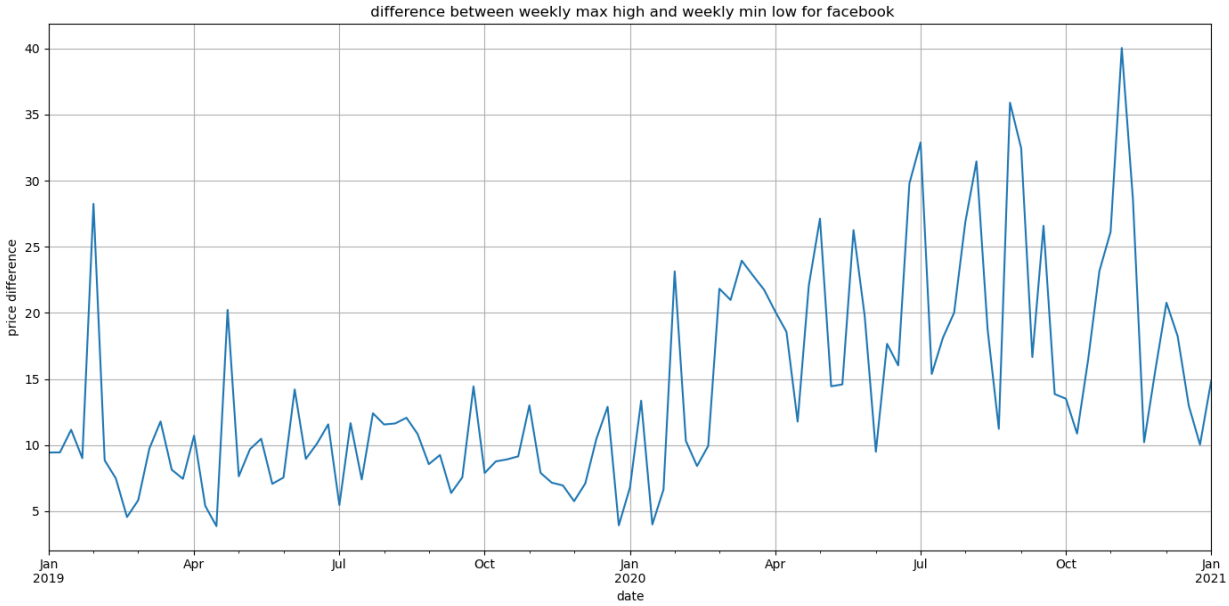
```
#3. Using the earthquake data, create box plots for the magnitudes of each magTypeused
#earthquakes . CSV.csv

#load pd, plt, sns
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

#earthquakes (1).csv
df = pd.read_csv('earthquakes (1).csv')

#filter out indonesia parsed_place
df_indonesia = df[df['parsed_place'].str.contains('Indonesia', na=False)]


plt.figure(figsize=(14, 7))

#magtype, and mag
sns.boxplot(x='magType', y='mag', data=df_indonesia)

plt.title('box plots for mags of each magtype in indonesia')
plt.xlabel('magnitude type')
```

```
plt.ylabel('magnitude')
plt.grid(True)
plt.tight_layout()
plt.show()
```



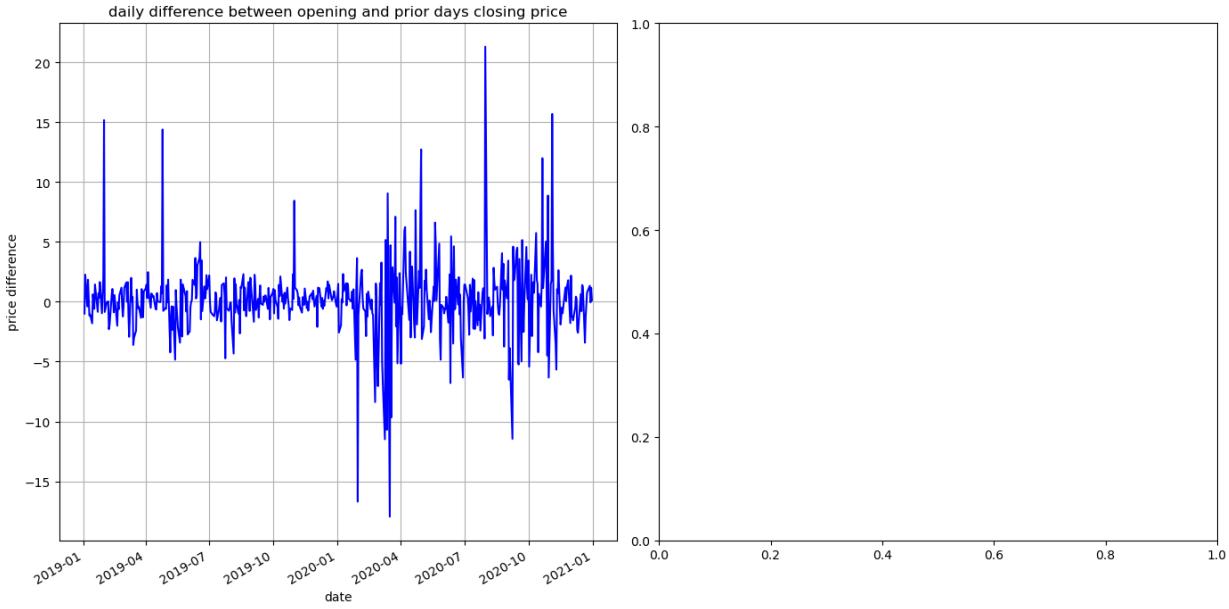
In [6]:

```
#4. Make a line plot of the difference between the weekly maximum high price and thewe


#load pd and plt
import pandas as pd
import matplotlib.pyplot as plt

#facebook.csv
df = pd.read_csv('facebook.csv', parse_dates=['date'], index_col='date')

#weekly maximum high price and minimum low price
weekly_max_high = df['high'].resample('W').max()
weekly_min_low = df['low'].resample('W').min()

#figure out the difference
weekly_difference = weekly_max_high - weekly_min_low

#plot
plt.figure(figsize=(14, 7))
weekly_difference.plot()

plt.title(' difference between weekly max high and weekly min low for facebook')
plt.xlabel('date')
plt.ylabel('price difference')
plt.grid(True)
plt.tight_layout()
plt.show()
```

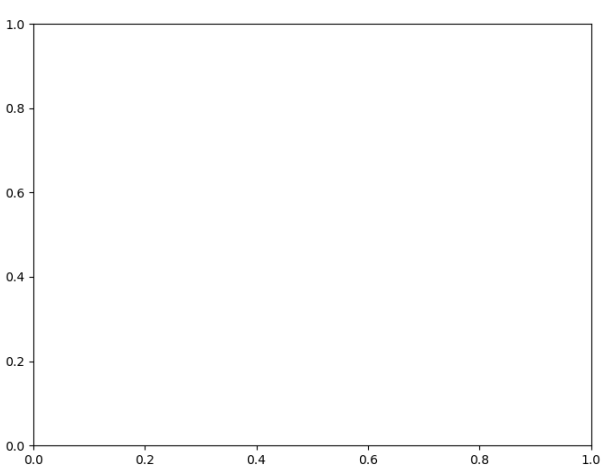


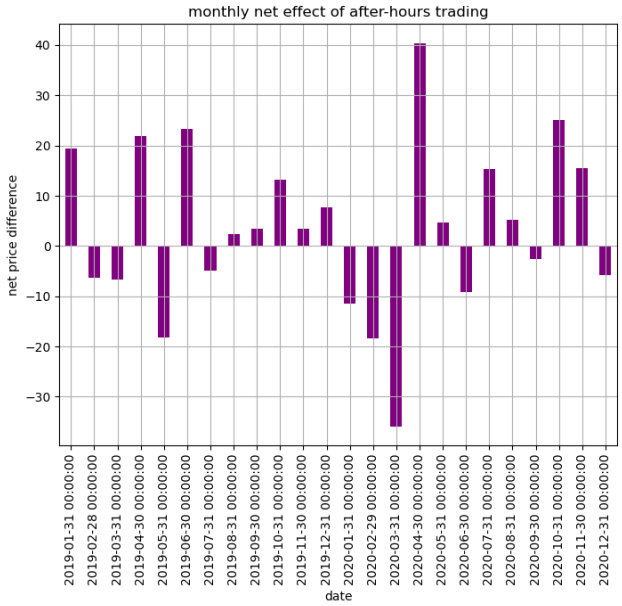
1. Using matplotlib and pandas, create two subplots side-by-side showing theeffect that after-hours trading has had on Facebook's stock prices:

In [7]:
```
#a) The first subplot will contain a line plot of the daily difference between that da


#load pd, plt
import pandas as pd
import matplotlib.pyplot as plt

#facebook.csv
df = pd.read_csv('facebook.csv', parse_dates=['date'], index_col='date')

#daily difference between that day's opening price and the prior day's closing price
df['after_hours_effect'] = df['open'] - df['close'].shift(1)

#subplots
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(14, 7))

# First subplot: Line plot of after-hours effect on stock prices
df['after_hours_effect'].plot(ax=axes[0], color='blue')
axes[0].set_title('daily difference between opening and prior days closing price')
axes[0].set_xlabel('date')
axes[0].set_ylabel('price difference')
axes[0].grid(True)

plt.tight_layout()
plt.show()
```



In [15]:
```
#b) The second subplot will be a bar plot showing the net effect this had monthly,usir

#load pd, and plt
import pandas as pd
import matplotlib.pyplot as plt

#facebook.csv
df = pd.read_csv('facebook.csv', parse_dates=['date'], index_col='date')


#daily difference between that day's opening price and the prior day's closing price
df['after_hours_effect'] = df['open'] - df['close'].shift(1)

#monthly resample of the after-hours effect
monthly_effect = df['after_hours_effect'].resample('M').sum()

#subplots
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(14, 7))

#bar plot showing the monthly net effect of after-hours trading
monthly_effect.plot(kind='bar', ax=axes[1], color='purple')
axes[1].set_title('monthly net effect of after-hours trading')
axes[1].set_xlabel('date')
axes[1].set_ylabel('net price difference')
axes[1].grid(True)

plt.tight_layout()
plt.show()
```


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


monthly net effect of after-hours trading
net price difference
40
30
20
10
0
−10
−20
−30
2019-01-31 00:00:00
2019-02-28 00:00:00
2019-03-31 00:00:00
2019-04-30 00:00:00
2019-05-31 00:00:00
2019-06-30 00:00:00
2019-07-31 00:00:00
2019-08-31 00:00:00
2019-09-30 00:00:00
2019-10-31 00:00:00
2019-11-30 00:00:00
2019-12-31 00:00:00
2020-01-31 00:00:00
2020-02-29 00:00:00
2020-03-31 00:00:00
2020-04-30 00:00:00
2020-05-31 00:00:00
2020-06-30 00:00:00
2020-07-31 00:00:00
2020-08-31 00:00:00
2020-09-30 00:00:00
2020-10-31 00:00:00
2020-11-30 00:00:00
2020-12-31 00:00:00
date

In [ ]: