Time Series Analysis by Derek Banas

Pandas Datetime Index -

setting the index of panda DF to the DateTimeIndex. Can use the date_range function like so -

```
pd.date_range('2020-10-01',periods=5,freq='D')
```

Time Resampling -

Get the average price and volume over the year

```
df_appl.resample(rule='A').mean()
```

We can also get the:

- W (weekly), M (monthly), SM (biweekly), Q (quarterly mean)
- Also D (Day), H (Hour), T (Minute), S (Second), L (Millisecond) U (Microsecond)

Moving Average -

Create a moving average by the rolling function.

```
df_appl['Price'].rolling(window=7).mean().plot()
df_appl['Price'].expanding().mean().plot()
```

Shift -

Shift DataFrame up or down.

```
df_appl.shift(1)
df_appl.shift(-1)
```

Plotting

Plotly

```
fig_2 = go.Figure()
fig_2.add_trace(go.Scatter(x=df_1.index, y=df_1['Close'],
                    mode='lines',
                    name='Closing Price'))
fig_2.update_layout(
    xaxis=dict(
        showgrid=True, showline=True, showticklabels=True
    ),
    yaxis=dict(
        showgrid=True, showline=True, showticklabels=True
    ),
    showlegend=True,
    xaxis_title='Date',
    yaxis_title='Closing Price',
    title='Plotly Amazon Stock Price'
)
```

Matplotlib Finances

```
mpf.plot(df_1, type='line')

# You can also plot multiple MAVS, volume and non-trading day data
# MAV last previous 3, 5 and 7 observations
mpf.plot(df_1, type='ohlc', mav=(3, 5, 7), volume=True,
show_nontrading=True)
```

Plotting Time Periods & Styling

```
fig_4 = plt.figure(figsize=(12,7), dpi=100)
ax_1 = fig_4.add_axes([0.0, 0.0, 0.9, 0.9])
ax_1.set_xlabel('Date')
ax_1.set_ylabel('Closing Price')
ax_1.set_title('Matplotlib Amazon Stock Price', fontsize=20)
```

```
# Matplotlib allows you to define time periods to plot
# You can style the line color, width, line style, and marker styling
df_1['Close'].plot(figsize=(12,6),
                   xlim=['2020-09-01','2020-10-16'],
                   ylim=[2750,3750],
                  color='red',
                  lw=3,
                  ls='-.',
                  marker='o', markersize=5,
                   markerfacecolor='blue',
                   markeredgecolor='red', markeredgewidth=3)
ax_1.legend(loc=0)
ax_1.grid(True, color='0.6', dashes=(5,2,1,2))

# Set tick font sizes
ax_1.tick_params(axis="x", labelsize=15)
ax_1.tick_params(axis="y", labelsize=18)

# Set axis label font sizes and turn off Date label
plt.xlabel('', fontsize=18)
plt.ylabel('Closing Price', fontsize=16)

# Set legend font size
plt.legend(loc=0, prop={'size': 16})

# Set the x axis to put ticks by week days
ax_1.xaxis.set_major_locator(dates.WeekdayLocator(byweekday=0))

# Set so that only month and day shows
# https://matplotlib.org/api/dates_api.html
# %Y - 4 digit year
# %y - 2 digit year
# %m - month as a number
# %b - month as abbreviated name
# %d - day
# %a - abbreviated day
# ax_1.xaxis.set_major_formatter(dates.DateFormatter('%m/%d'))

# We can also use major and minor locators
ax_1.xaxis.set_major_formatter(dates.DateFormatter('%d'))
ax_1.xaxis.set_minor_locator(dates.MonthLocator())
ax_1.xaxis.set_minor_formatter(dates.DateFormatter('\n\n%b'))
```

```
# Change minor label font size
ax_1.tick_params(axis='both', which='minor', labelsize=15)
```