```python
import pandas as pd
df = pd.read_csv('data/faang.csv')
print(df.dtypes)
```

✓ 0.0s

```
ticker     object
date       object
open      float64
high      float64
low       float64
close     float64
volume      int64
dtype: object
```

With 'faang' dataset, use type conversion to change the date column into a datetime and the volume column into integers. Then, sort by date and ticker.

```python
df=df.assign(
    date=pd.to_datetime(df.date),
    volume=df.volume.astype('int')
)
print(df.dtypes)
```

✓ 0.0s Python

```
ticker            object
date      datetime64[ns]
open             float64
high             float64
low              float64
close            float64
volume             int32
dtype: object
```

```python
df=df.sort_values(by=['date', 'ticker'])
print(df.head())
```

✓ 0.0s Python

```
     ticker       date       open       high        low      close    volume
251    AAPL 2018-01-02   166.9271   169.0264   166.0442   168.9872  25555934
502    AMZN 2018-01-02  1172.0000  1190.0000  1170.5100  1189.0100   2694494
0        FB 2018-01-02   177.6800   181.5800   177.5500   181.4200  18151903
1004   GOOG 2018-01-02  1048.3400  1066.9400  1045.2300  1065.0000   1237564
753    NFLX 2018-01-02   196.1000   201.6500   195.4200   201.0700  10966889
```

Find the seven rows with the highest value for volume.

```
print(df.nlargest(n=7, columns='volume'))
```

✓ 0.0s

	ticker	date	open	high	low	close	volume
142	FB	2018-07-26	174.8900	180.1300	173.7500	176.2600	169803668
53	FB	2018-03-20	167.4700	170.2000	161.9500	168.1500	129851768
57	FB	2018-03-26	160.8200	161.1000	149.0200	160.0600	126116634
54	FB	2018-03-21	164.8000	173.4000	163.3000	169.3900	106598834
433	AAPL	2018-09-21	219.0727	219.6482	215.6097	215.9768	96246748
496	AAPL	2018-12-21	156.1901	157.4845	148.9909	150.0862	95744384
463	AAPL	2018-11-02	207.9295	211.9978	203.8414	205.8755	91328654

Right now the data is somewhere between long and wide format. Use melt() to make it a completely long format. Hint: date and ticker are our ID variables (they uniquely identify each row). We need to melt the rest so that we don't have separate columns for open, high, low, close, and volume.

+ Code + Markdown

```
df = pd.melt(df, id_vars=['date', 'ticker'],
             value_vars=['open', 'high', 'low', 'close', 'volume'],
             )
print(df.head())
```

✓ 0.0s Python

	date	ticker	variable	value
0	2018-01-02	AAPL	open	166.9271
1	2018-01-02	AMZN	open	1172.0000
2	2018-01-02	FB	open	177.6800
3	2018-01-02	GOOG	open	1048.3400
4	2018-01-02	NFLX	open	196.1000

Suppose we found out there was a glitch in how the data was recorded on July 26, 2018. How should we handle this? Note that there is no coding required for this exercise.

Firstly, I would identify which among the variables and data points had an error. Secondly, if I can contact the data providers, I would consult with them how the glitch happened and correct the data points accordingly. Else, I would isolate the data in that date and figure out a way to correct the data myself.