Data Visualization Project
2017 Netflix stock profile

Patrick Tsai

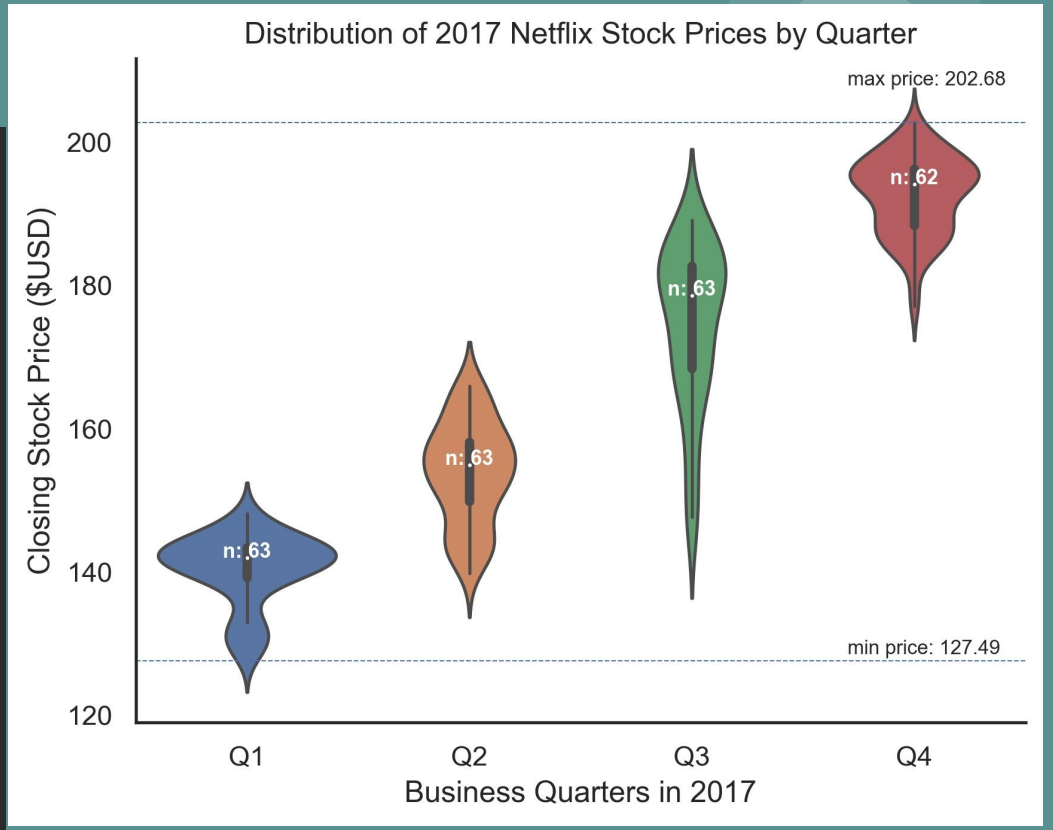
Overview:

- **Create a series of visualizations to analyze and aid in risk assessment for Netflix stocks in 2017**
- **Datasets consisted of daily recordings for Netflix quarterly earnings, as well as monthly readings for the Netflix stock and Dow Jones Industrial Average to obtain a sense of the present market**
- **Developed with the following python packages: pandas, matplotlib, seaborn**

Code: Distribution of Netflix stock prices

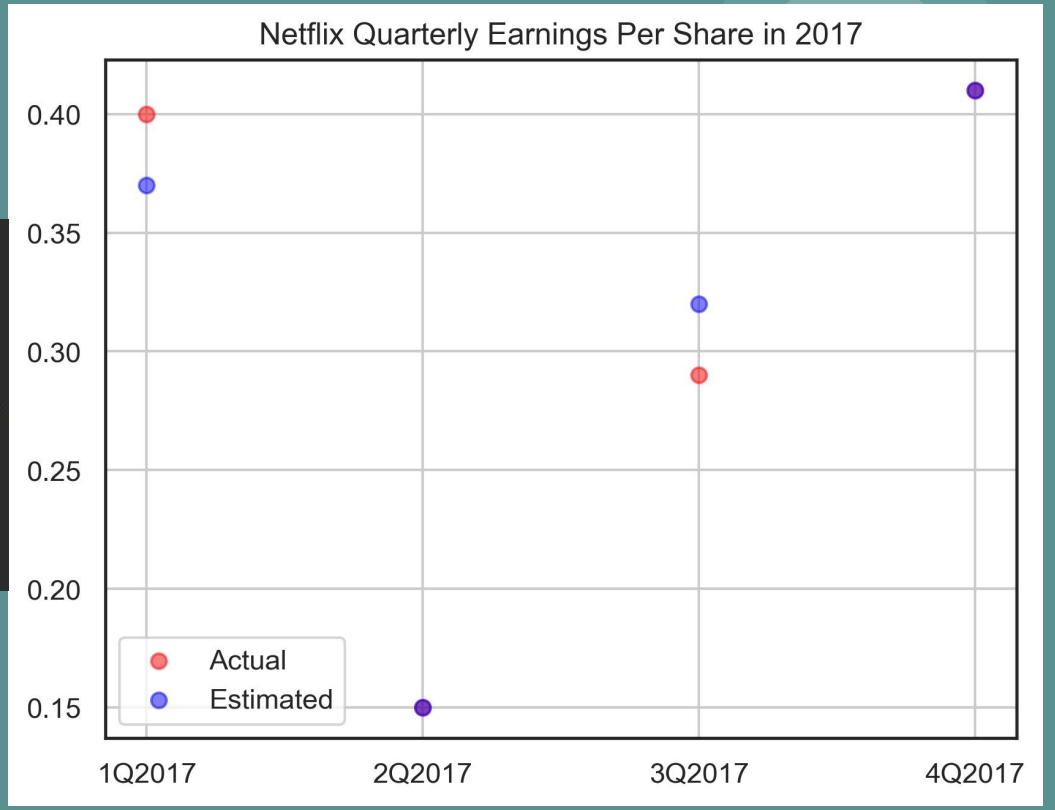
- **Since the dataset obtained from Yahoo Finance was relatively clean with no missing values, I was able to directly import the tabular files with minimal preprocessing**

```
from matplotlib import pyplot as plt
import pandas as pd
import seaborn as sns

netflix_stocks = pd.read_csv('NFLX.csv')
dowjones_stocks = pd.read_csv('DJI.csv')
netflix_stocks_quarterly = pd.read_csv('NFLX_daily_by_quarter.csv')

# renaming columns to simplify later data manipulation
netflix_stocks.rename(columns={'Adj Close':'Price'}, inplace=True)
dowjones_stocks.rename(columns={'Adj Close':'Price'}, inplace=True)
netflix_stocks_quarterly.rename(columns={'Adj Close':'Price'}, inplace=True)

# Violin plot
sns.set(style='white')
ax = sns.violinplot(data=netflix_stocks_quarterly, x='Quarter', y='Price', linewidth=1.25)
ax.axhline(min(netflix_stocks_quarterly.Price), ls='--', lw=0.5)
ax.axhline(max(netflix_stocks_quarterly.Price), ls='--', lw=0.5)
sns.despine()

# drawing min and max value lines
minNetflixQuarterly = min(netflix_stocks_quarterly.Price)
maxNetflixQuarterly = max(netflix_stocks_quarterly.Price)
ax.text(2.7,minNetflixQuarterly+1, "min price: " + str(round(minNetflixQuarterly,2)), fontsize=8)
ax.text(2.7,maxNetflixQuarterly+5.2, "max price: " + str(round(maxNetflixQuarterly,2)), fontsize=8)

# assigning sample size n label to explicitly state sample size on graph
medians = netflix_stocks_quarterly.groupby(['Quarter'])['Price'].median().values
nobs = netflix_stocks_quarterly['Quarter'].value_counts().values
nobs = [str(i) for i in nobs.tolist()]
nobs = ['n: ' + j for j in nobs]

# positioning label above median value on violin plots
pos = range(len(nobs))
for tick, label in zip(pos, ax.get_xticklabels()):
    ax.text(pos[tick], medians[tick] + 0.03, nobs[tick], horizontalalignment='center',
            size='x-small', color='w', weight='semibold')

# labelling axes and saving figure in high res
ax.set_title('Distribution of 2017 Netflix Stock Prices by Quarter')
plt.xlabel('Business Quarters in 2017')
plt.ylabel('Closing Stock Price ($USD)')
plt.savefig('NetflixQuarterlyAnalysis.jpg', dpi=300, quality=95, optimize=True, bbox_inches='tight')
plt.clf()
```



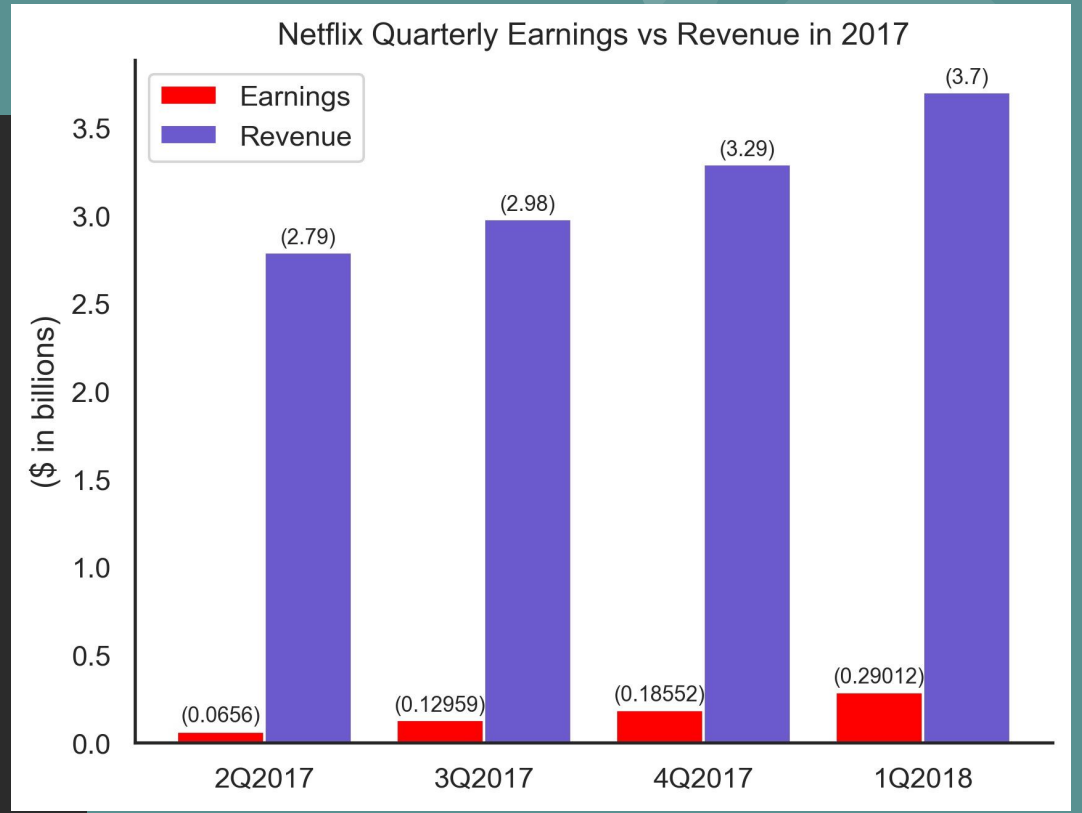
Code: Actual vs Estimated EPS by Quarter

```
# Stock EPS scatterplot
x_positions = [1, 2, 3, 4]
chart_labels = ["1Q2017","2Q2017","3Q2017","4Q2017"]
earnings_actual =[.4, .15,.29,.41]
earnings_estimate = [.37,.15,.32,.41]

plt.scatter(x_positions, earnings_actual, color='red', alpha=0.5)
plt.scatter(x_positions, earnings_estimate, color='blue', alpha=0.5)
plt.legend(['Actual','Estimated'], loc=0)
plt.xticks(x_positions, chart_labels)
plt.title("Netflix Quarterly Earnings Per Share in 2017")
plt.grid()
plt.savefig('EPS.jpg', dpi=300, quality=95, optimize=True,
            bbox_inches='tight')
plt.clf()
```



Code: Reported Earnings versus Revenue

```
# Side side bar graphs- units in billions of USD
revenue_by_quarter = [2.79, 2.98,3.29,3.7]
earnings_by_quarter = [.0656,.12959,.18552,.29012]
quarterlyLabels = ["2Q2017","3Q2017","4Q2017", "1Q2018"]

# Revenue
n = 1 # iterator(out of 2)
t = 2 # Number of dataset
d = 4 # Number of sets of bars
w = 0.8 # Width of each bar
bars1_x = [t*element + w*n for element in range(d)]

# Earnings
n, t, d, w = 2, 2, 4, 0.8
bars2_x = [t*element + w*n for element in range(d)]
fig, ax3 = plt.subplots()
bar1 = ax3.bar(bars1_x, earnings_by_quarter, color='red')
bar2 = ax3.bar(bars2_x, revenue_by_quarter, color='slateblue')

middle_x = [(a + b) / 2 for a, b in zip(bars1_x, bars2_x)]
legendLabels = ["Earnings", "Revenue"]
ax3.set_title('Netflix Quarterly Earnings vs Revenue in 2017')
ax3.set_ylabel('($ in billions)')
ax3.set_xticks(middle_x)
ax3.set_xticklabels(quarterlyLabels)
ax3.legend(legendLabels, loc=0)
ax3.spines['right'].set_visible(False)
ax3.spines['top'].set_visible(False)

bar1Labels, bar2Labels = list(), list()
temp = zip(earnings_by_quarter,revenue_by_quarter)
for x,y in temp:
    bar1Labels.append(x)
    bar2Labels.append(y)

for x,y in zip(bars1_x,bar1Labels):
    ax3.annotate('(%s)' % y, xy=(x,round(y+0.05,2)), fontsize='x-small', ha='center')
for x,y in zip(bars2_x,bar2Labels):
    ax3.annotate('(%s)' % y, xy=(x,round(y+0.05,2)), fontsize='x-small', ha='center')

plt.savefig('EarningsVsRevenue.jpg', dpi=300, quality=95, optimize=True, bbox_inches='tight')
plt.clf()
```



Code: Comparison of Netflix vs DJIA

```
# Netflix stock vs DJI average 2017 comparison
fig = plt.figure(figsize=(12,7))
fig.suptitle("Netflix stock prices vs " \
             "Dow Jones Industrial Avg in 2017")
month_labels = ['Jan','Feb','Mar','Apr','May','Jun',
                'Jul','Aug','Sep','Oct','Nov','Dec']

# Left plot Netflix
ax1 = plt.subplot(1,2,1)
ax1.plot(netflix_stocks.Date, netflix_stocks.Price,
         c='b', marker='x')
ax1.set_title('Netflix')
ax1.set_xlabel("Date")
ax1.set_ylabel("Stock Price")
ax1.set_xticklabels(month_labels)

# Right plot Dow Jones
ax2 = plt.subplot(1,2,2)
ax2.plot(dowjones_stocks.Date, dowjones_stocks.Price,
         c='r', marker='o')
ax2.set_title('Dow Jones')
ax2.set_xlabel("Date")
ax2.set_ylabel("Price-weighted index")
ax2.set_xticklabels(month_labels)

plt.subplots_adjust(wspace=0.4)
plt.savefig('NFLXvDJI.jpg', dpi=300, quality=95,
            optimize=True, bbox_inches='tight')
plt.clf()
```