In [15]:
```
from matplotlib import pyplot as plt
import pandas as pd
import seaborn as sns
import os
os. chdir('E:/Netflix/Netflix Stocks Capstone')
```

In [16]:
```
netflix_stocks=pd.read_csv('NFLX.csv')
print(netflix_stocks)
```

```
          Date        Open        High         Low       Close   Adj Close  \
0   2017-01-01  124.959999  143.460007  124.309998  140.710007  140.710007
1   2017-02-01  141.199997  145.949997  139.050003  142.130005  142.130005
2   2017-03-01  142.839996  148.289993  138.259995  147.809998  147.809998
3   2017-04-01  146.699997  153.520004  138.660004  152.199997  152.199997
4   2017-05-01  151.910004  164.750000  151.610001  163.070007  163.070007
5   2017-06-01  163.520004  166.869995  147.300003  149.410004  149.410004
6   2017-07-01  149.800003  191.500000  144.250000  181.660004  181.660004
7   2017-08-01  182.490005  184.619995  164.229996  174.710007  174.710007
8   2017-09-01  175.550003  189.949997  172.440002  181.350006  181.350006
9   2017-10-01  182.110001  204.380005  176.580002  196.429993  196.429993
10  2017-11-01  197.240005  202.479996  184.320007  195.509995  195.509995
11  2017-12-01  186.990005  194.490005  178.380005  191.960007  191.960007

       Volume
0   181772200
1    91432000
2   110692700
3   149769200
4   116795800
5   135675800
6   185144700
7   136523100
8   111427900
9   208657800
10  161719700
11  115103700
```

In [17]:
```
dowjones_stocks=pd.read_csv('DJI.csv')
print(dowjones_stocks)
```

```
          Date          Open          High           Low         Close  \
0   2017-01-01  19872.859375  20125.580078  19677.939453  19864.089844
1   2017-02-01  19923.810547  20851.330078  19831.089844  20812.240234
2   2017-03-01  20957.289063  21169.109375  20412.800781  20663.220703
3   2017-04-01  20665.169922  21070.900391  20379.550781  20940.509766
4   2017-05-01  20962.730469  21112.320313  20553.449219  21008.650391
5   2017-06-01  21030.550781  21535.029297  20994.220703  21349.630859
6   2017-07-01  21392.300781  21929.800781  21279.300781  21891.119141
7   2017-08-01  21961.419922  22179.109375  21600.339844  21948.099609
8   2017-09-01  21981.769531  22419.509766  21709.630859  22405.089844
9   2017-10-01  22423.470703  23485.250000  22416.000000  23377.240234
10  2017-11-01  23442.900391  24327.820313  23242.750000  24272.349609
11  2017-12-01  24305.400391  24876.070313  23921.900391  24719.220703

       Adj Close      Volume
0   19864.089844  6482450000
1   20812.240234  6185580000
2   20663.220703  6941970000
3   20940.509766  5392630000
4   21008.650391  6613570000
5   21349.630859  7214590000
6   21891.119141  5569720000
7   21948.099609  6150060000
8   22405.089844  6342130000
9   23377.240234  7302910000
10  24272.349609  7335640000
11  24719.220703  6589890000
```

In [18]:
```
netflix_stocks_quarterly=pd.read_csv("NFLX_daily_by_quarter.csv")
print(netflix_stocks_quarterly)
```

```
           Date        Open        High         Low       Close   Adj Close  \
0    2017-01-03  124.959999  128.190002  124.309998  127.489998  127.489998
1    2017-01-04  127.489998  130.169998  126.550003  129.410004  129.410004
2    2017-01-05  129.220001  132.750000  128.899994  131.809998  131.809998
3    2017-01-06  132.080002  133.880005  129.809998  131.070007  131.070007
4    2017-01-09  131.479996  131.990005  129.889999  130.949997  130.949997
..          ...         ...         ...         ...         ...         ...
246  2017-12-22  188.330002  190.949997  186.800003  189.940002  189.940002
247  2017-12-26  189.779999  189.940002  186.399994  187.759995  187.759995
248  2017-12-27  187.800003  188.100006  185.220001  186.240005  186.240005
249  2017-12-28  187.179993  194.490005  186.850006  192.710007  192.710007
250  2017-12-29  192.509995  193.949997  191.220001  191.960007  191.960007

       Volume Quarter
0     9437900      Q1
1     7843600      Q1
2    10185500      Q1
3    10657900      Q1
4     5766900      Q1
..        ...     ...
246   3878900      Q4
247   3045700      Q4
248   4002100      Q4
249  10107400      Q4
250   5187600      Q4

[251 rows x 8 columns]
```

In [19]:
```
A=max(netflix_stocks.Date)
print(A)
B=max(netflix_stocks_quarterly.Date)
print(B)
C=min(netflix_stocks.Date)
print(C)
```

```
2017-12-01
2017-12-29
2017-01-01
```

The data is represented on monthly basis.\ The netflix and dow jones in based on months whereas netflix quarterly is daily data on quatermonth based.\ netflix_stocks_quarterly contains quarter column

In [20]:
```
print(netflix_stocks.head())
```

```
         Date        Open        High         Low       Close   Adj Close  \
0  2017-01-01  124.959999  143.460007  124.309998  140.710007  140.710007
1  2017-02-01  141.199997  145.949997  139.050003  142.130005  142.130005
2  2017-03-01  142.839996  148.289993  138.259995  147.809998  147.809998
3  2017-04-01  146.699997  153.520004  138.660004  152.199997  152.199997
4  2017-05-01  151.910004  164.750000  151.610001  163.070007  163.070007

      Volume
0  181772200
1   91432000
2  110692700
3  149769200
4  116795800
```

In [21]:
```
netflix_stocks=netflix_stocks.rename(columns={'Adj Close':'Price'})
dowjones_stocks=dowjones_stocks.rename(columns={'Adj Close':'Price'})
netflix_stocks_quarterly=netflix_stocks_quarterly.rename(columns={'Adj Close':'Price'})
```

In [22]:
```
print(netflix_stocks.head())
```

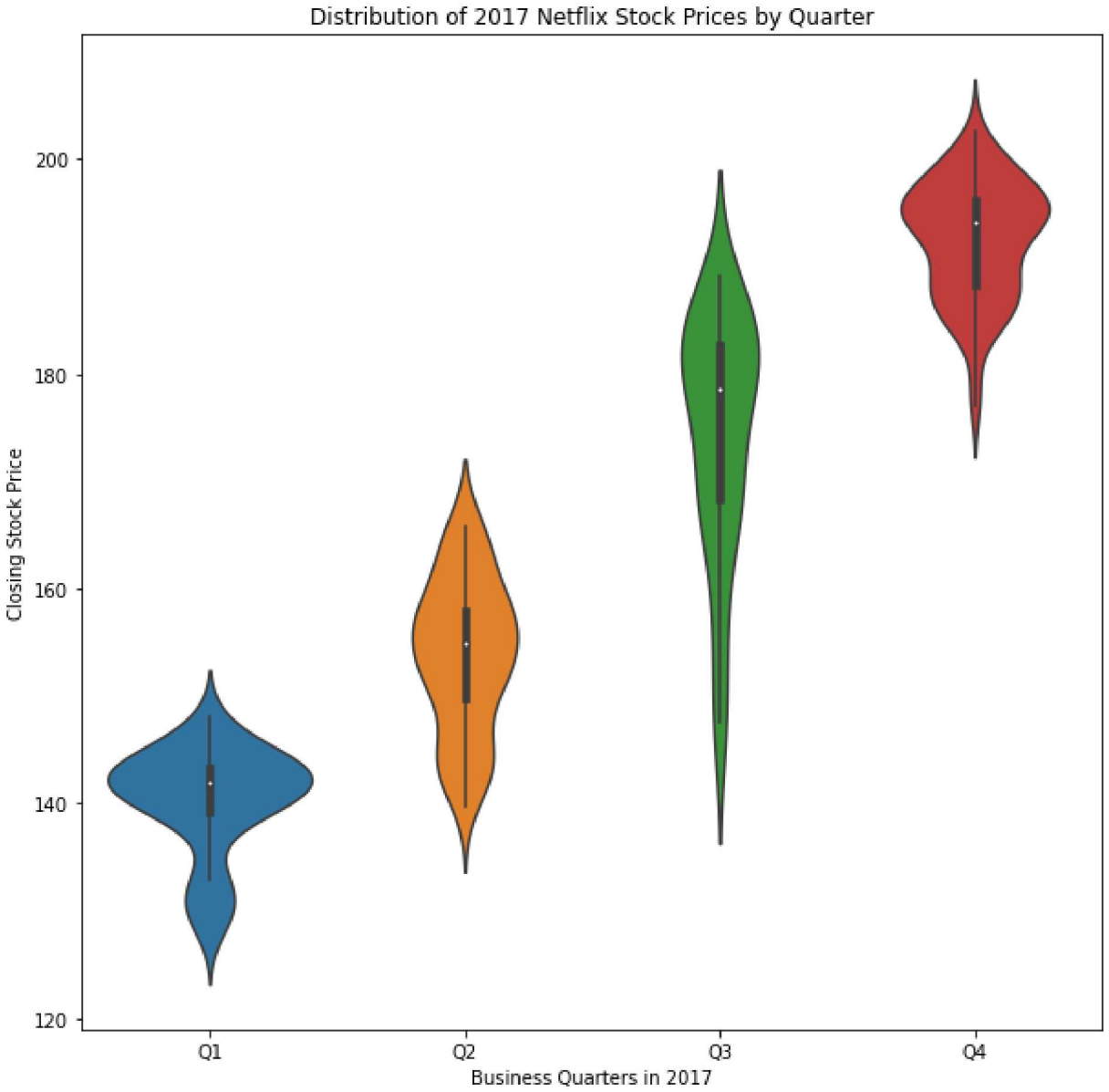
```
         Date        Open        High         Low       Close       Price  \
0  2017-01-01  124.959999  143.460007  124.309998  140.710007  140.710007
1  2017-02-01  141.199997  145.949997  139.050003  142.130005  142.130005
2  2017-03-01  142.839996  148.289993  138.259995  147.809998  147.809998
3  2017-04-01  146.699997  153.520004  138.660004  152.199997  152.199997
4  2017-05-01  151.910004  164.750000  151.610001  163.070007  163.070007

      Volume
0  181772200
1   91432000
2  110692700
3  149769200
4  116795800
```

In [23]:
```
print(dowjones_stocks.head())
print(netflix_stocks_quarterly.head())
```

```
         Date          Open          High           Low         Close  \
0  2017-01-01  19872.859375  20125.580078  19677.939453  19864.089844
1  2017-02-01  19923.810547  20851.330078  19831.089844  20812.240234
2  2017-03-01  20957.289063  21169.109375  20412.800781  20663.220703
3  2017-04-01  20665.169922  21070.900391  20379.550781  20940.509766
4  2017-05-01  20962.730469  21112.320313  20553.449219  21008.650391

          Price      Volume
0  19864.089844  6482450000
1  20812.240234  6185580000
2  20663.220703  6941970000
3  20940.509766  5392630000
4  21008.650391  6613570000
         Date        Open        High         Low       Close       Price  \
0  2017-01-03  124.959999  128.190002  124.309998  127.489998  127.489998
1  2017-01-04  127.489998  130.169998  126.550003  129.410004  129.410004
2  2017-01-05  129.220001  132.750000  128.899994  131.809998  131.809998
3  2017-01-06  132.080002  133.880005  129.809998  131.070007  131.070007
4  2017-01-09  131.479996  131.990005  129.889999  130.949997  130.949997

     Volume Quarter
0   9437900      Q1
1   7843600      Q1
2  10185500      Q1
3  10657900      Q1
4   5766900      Q1
```

In [24]:

```
plt.figure(figsize = (10,10))
ax=sns.violinplot(data=netflix_stocks_quarterly,x="Quarter",y="Price")
ax.set_title("Distribution of 2017 Netflix Stock Prices by Quarter")
plt.ylabel("Closing Stock Price")
plt.xlabel("Business Quarters in 2017")
plt.show
plt.savefig("Distribution of 2017 Netflix Stock Prices by Quarter.png")
```

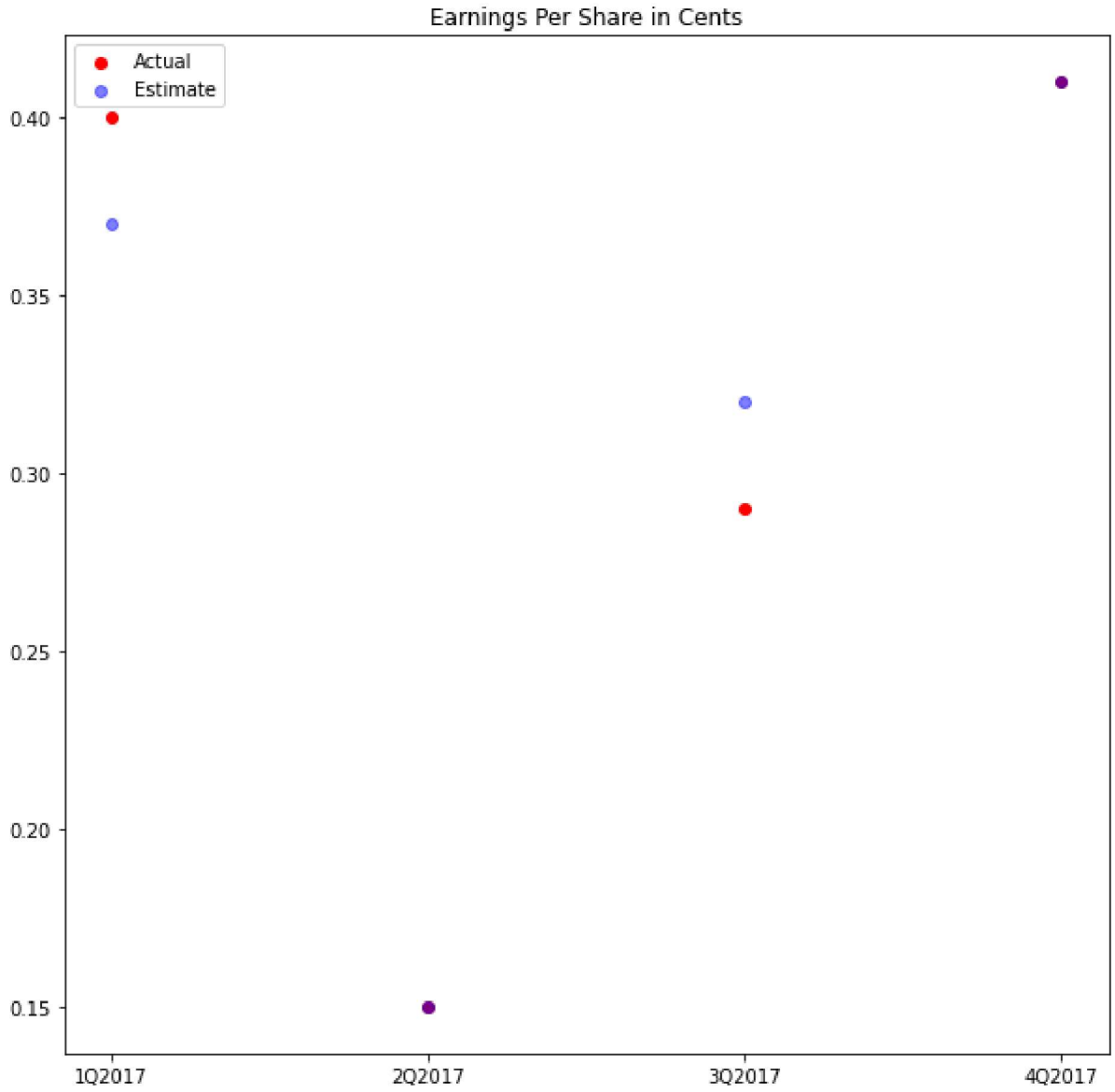

Distribution of 2017 Netflix Stock Prices by Quarter
Closing Stock Price
200
180
160
140
120
Q1
Q2
Q3
Q4
Business Quarters in 2017

Graph literacy

Netflix Stock Prices should a bull trend throughout the course of 2017 (Q1 - Q4).\ Prices fell in the 140 - 180 range throughout the year.\ The lowest price was in (Q1) at 130 and highest was in (Q4) at 210.

In [29]:

```
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.figure(figsize = (10,10))
plt.scatter(x_positions,earnings_actual,color="red")
plt.scatter(x_positions,earnings_estimate,color="blue",alpha=0.5)
plt.legend(["Actual", "Estimate"])
plt.xticks(x_positions, chart_labels)
plt.title("Earnings Per Share in Cents")
plt.savefig("Earnings Per Share in Cents.png")
plt.show()
```


Earnings Per Share in Cents
Actual
Estimate
0.40
0.35
0.30
0.25
0.20
0.15
1Q2017
2Q2017
3Q2017
4Q2017

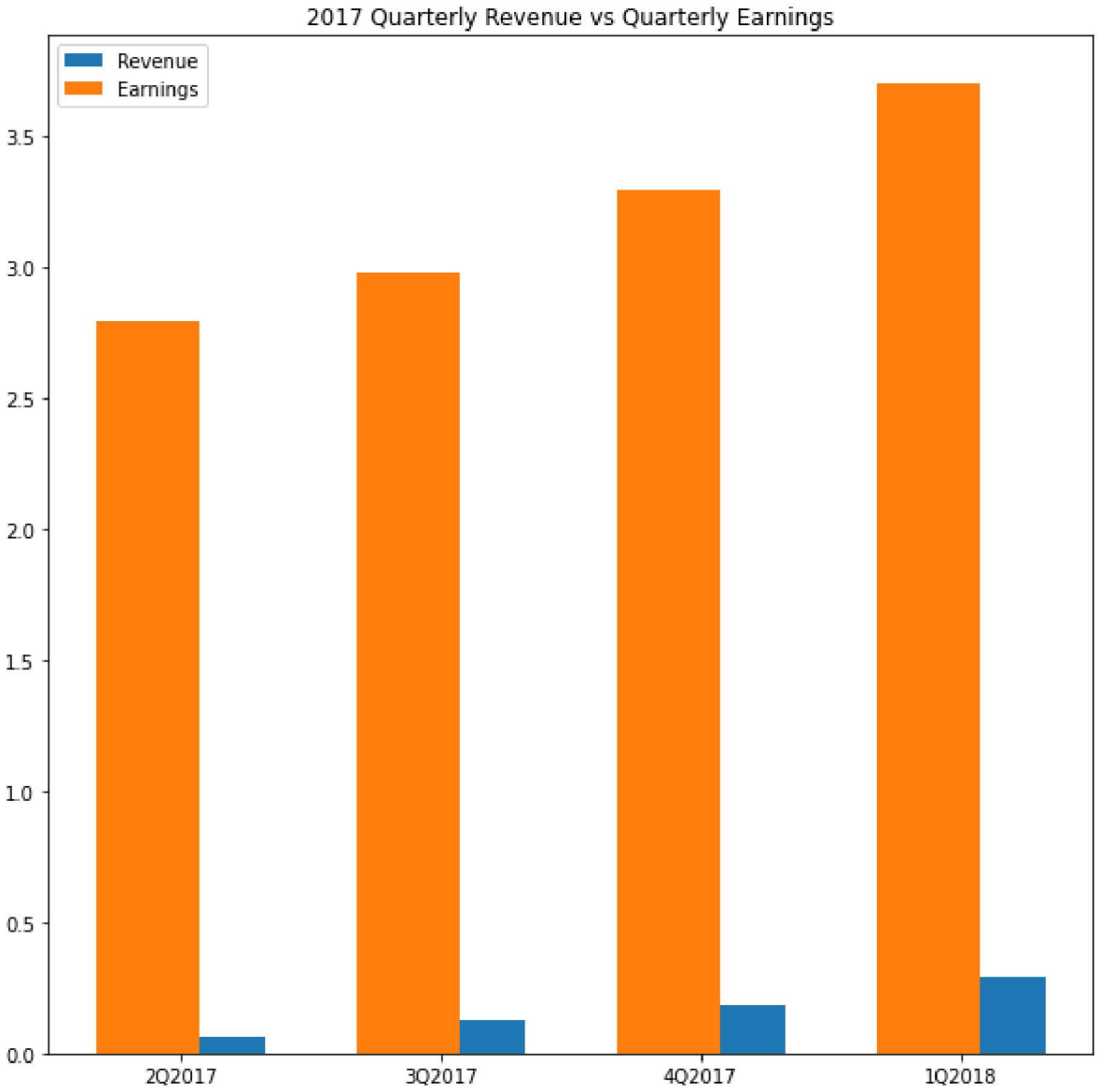
Graph Literacy

- What do the purple dots tell us about the actual and estimate earnings per share in this graph? Hint: In color theory red and blue mix to make purple.

> Purple dot shows where the estimated value and actual value overlapped.

In [26]:

```
# The metrics below are in billions of dollars
revenue_by_quarter = [2.79, 2.98,3.29,3.7]
earnings_by_quarter = [.0656,.12959,.18552,.29012]
quarter_labels = ["2Q2017","3Q2017","4Q2017", "1Q2018"]

# Revenue
n = 1
t = 2
d = 4
w = 0.5
bars1_x = [t*element + w*n for element
             in range(d)]



# Earnings
n = 2
t = 2
d = 4 # Number of sets of bars
w = 0.5 # Width of each bar
bars2_x = [t*element + w*n for element
             in range(d)]




middle_x = [ (a + b) / 2.0 for a, b in zip(bars1_x, bars2_x)]
labels = ["Revenue", "Earnings"]
plt.figure(figsize = (10,10))
plt.bar(bars2_x, earnings_by_quarter)
plt.bar(bars1_x, revenue_by_quarter)
plt.legend(labels)
plt.title('2017 Quarterly Revenue vs Quarterly Earnings')
plt.xticks(middle_x, quarter_labels)
plt.savefig("2017 Quarterly Revenue vs Quarterly Earnings.png")
```


2017 Quarterly Revenue vs Quarterly Earnings
Revenue
Earnings
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
2Q2017
3Q2017
4Q2017
1Q2018

Graph Literacy

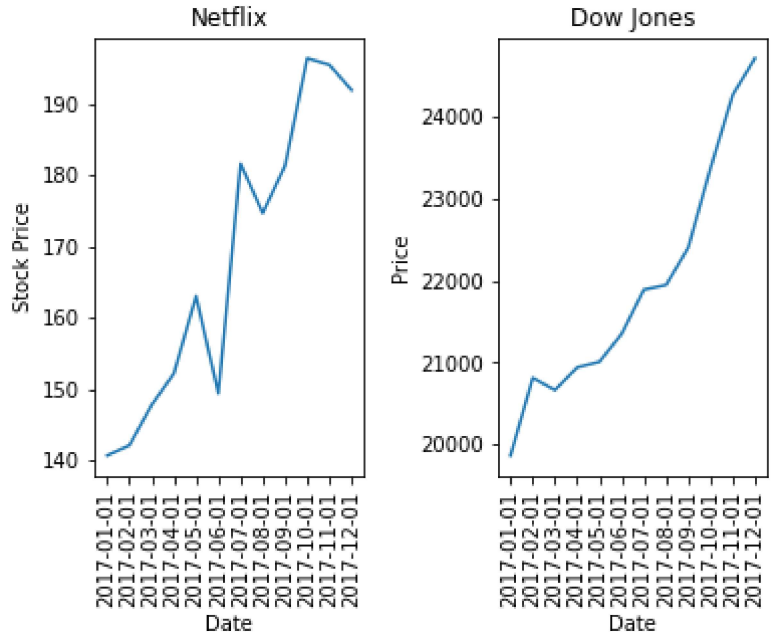
What are your first impressions looking at the visualized data?

- Does Revenue follow a trend?
- Do Earnings follow a trend?
- Roughly, what percentage of the revenue constitutes earnings?

Graph Literacy answers

> The revenue follows an upward trend throughout.\ The earnings also tend to increase overtime.\ Roughly 14% of revenue constitutes earnings

In [33]:
```
# Left plot Netflix
ax1 = plt.subplot(1,2,1)
plt.plot(netflix_stocks['Date'], netflix_stocks['Price'])
plt.title("Netflix")
ax1.set_xlabel("Date")
ax1.set_ylabel("Stock Price")
plt.subplots_adjust(wspace = 0.5)
plt.xticks(rotation = 'vertical')



# Right plot Dow Jones
ax2 = plt.subplot(1,2,2)
plt.plot(dowjones_stocks['Date'], dowjones_stocks['Price'])
plt.title('Dow Jones')
ax2.set_xlabel('Date')
ax2.set_ylabel('Price')
plt.subplots_adjust(wspace = 0.5)
plt.xticks(rotation = 'vertical')

plt.savefig("Netflix vs Dow Jones stocks.png", bbox_inches='tight')
```



- How did Netflix perform relative to Dow Jones Industrial Average in 2017?
- Which was more volatile?
- How do the prices of the stocks compare?

> Netflix stock performed well relative to Dow Jones industrial average in 2017.\ The Netflix share was more volatile as seen from graph of different peak and valleys.\ The Dow Jones stock price is more.

In [ ]: