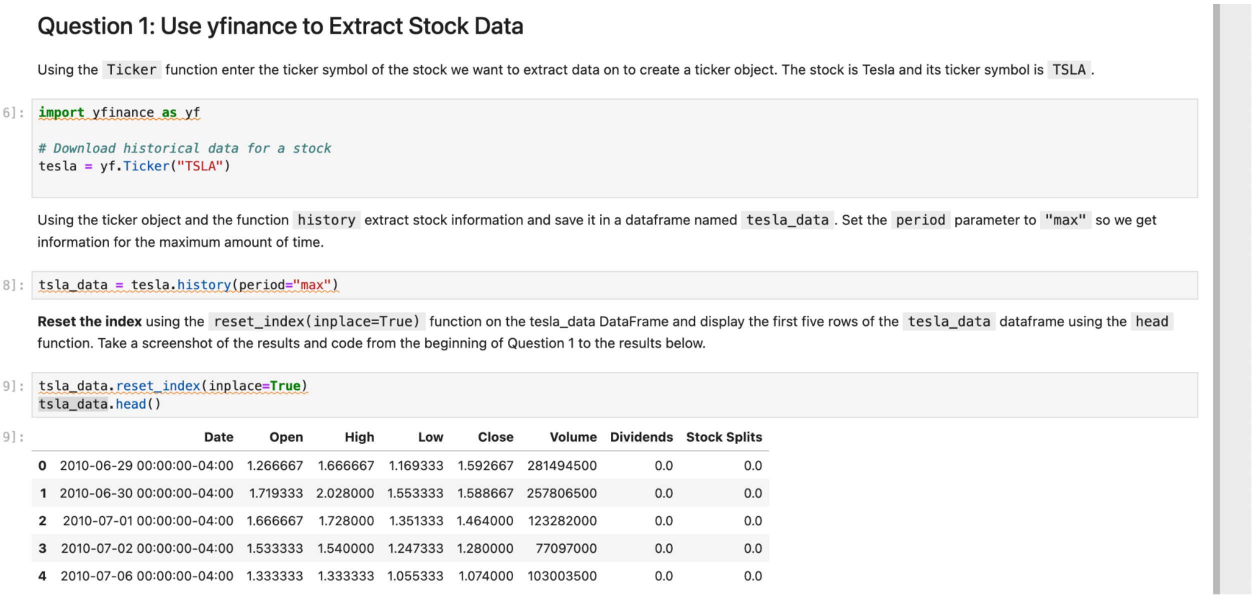


Question 1: Use yfinance to Extract Stock Data

Using the `Ticker` function enter the ticker symbol of the stock we want to extract data on to create a ticker object. The stock is Tesla and its ticker symbol is `TSLA`.

```
6]: import yfinance as yf

    # Download historical data for a stock
    tesla = yf.Ticker("TSLA")
```

Using the ticker object and the function `history` extract stock information and save it in a dataframe named `tesla_data`. Set the `period` parameter to `"max"` so we get information for the maximum amount of time.

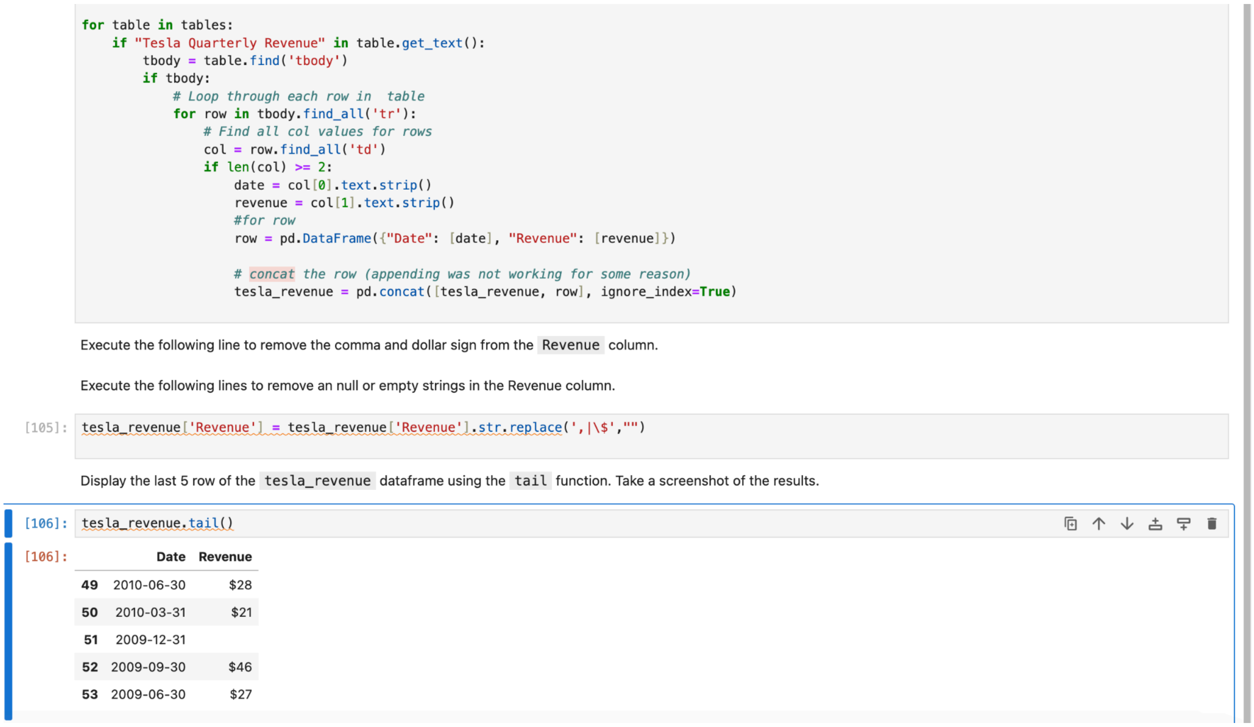
```
8]: tsla_data = tesla.history(period="max")
```

Reset the index using the `reset_index(inplace=True)` function on the tesla_data DataFrame and display the first five rows of the `tesla_data` dataframe using the `head` function. Take a screenshot of the results and code from the beginning of Question 1 to the results below.

```
9]: tsla_data.reset_index(inplace=True)
    tsla_data.head()
```

9]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2010-06-29 00:00:00-04:00	1.266667	1.666667	1.169333	1.592667	281494500	0.0	0.0
1	2010-06-30 00:00:00-04:00	1.719333	2.028000	1.553333	1.588667	257806500	0.0	0.0
2	2010-07-01 00:00:00-04:00	1.666667	1.728000	1.351333	1.464000	123282000	0.0	0.0
3	2010-07-02 00:00:00-04:00	1.533333	1.540000	1.247333	1.280000	77097000	0.0	0.0
4	2010-07-06 00:00:00-04:00	1.333333	1.333333	1.055333	1.074000	103003500	0.0	0.0



```
for table in tables:
    if "Tesla Quarterly Revenue" in table.get_text():
        tbody = table.find('tbody')
        if tbody:
            # Loop through each row in  table
            for row in tbody.find_all('tr'):
                # Find all col values for rows
                col = row.find_all('td')
                if len(col) >= 2:
                    date = col[0].text.strip()
                    revenue = col[1].text.strip()
                    #for row
                    row = pd.DataFrame({"Date": [date], "Revenue": [revenue]})

                    # concat the row (appending was not working for some reason)
                    tesla_revenue = pd.concat([tesla_revenue, row], ignore_index=True)
```

Execute the following line to remove the comma and dollar sign from the `Revenue` column.

Execute the following lines to remove an null or empty strings in the Revenue column.

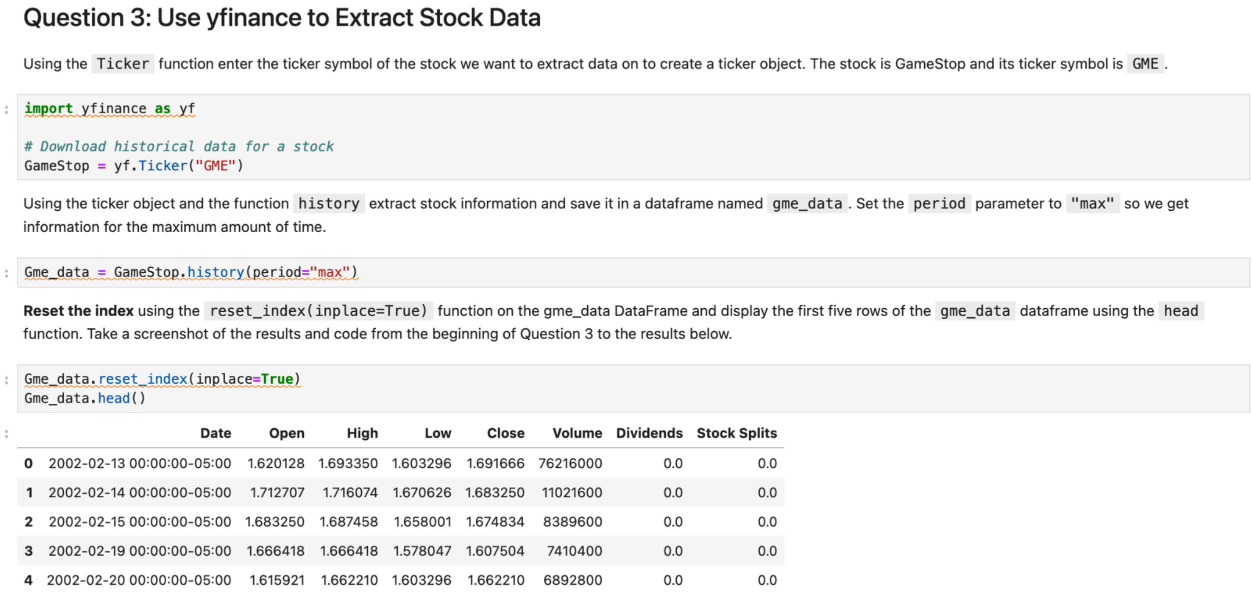
```
[105]: tesla_revenue['Revenue'] = tesla_revenue['Revenue'].str.replace(',|\$',"")
```

Display the last 5 row of the `tesla_revenue` dataframe using the `tail` function. Take a screenshot of the results.

```
[106]: tesla_revenue.tail()
```

[106]:

	Date	Revenue
49	2010-06-30	$28
50	2010-03-31	$21
51	2009-12-31	
52	2009-09-30	$46
53	2009-06-30	$27



Question 3: Use yfinance to Extract Stock Data

Using the `Ticker` function enter the ticker symbol of the stock we want to extract data on to create a ticker object. The stock is GameStop and its ticker symbol is `GME`.

```
import yfinance as yf

# Download historical data for a stock
GameStop = yf.Ticker("GME")
```

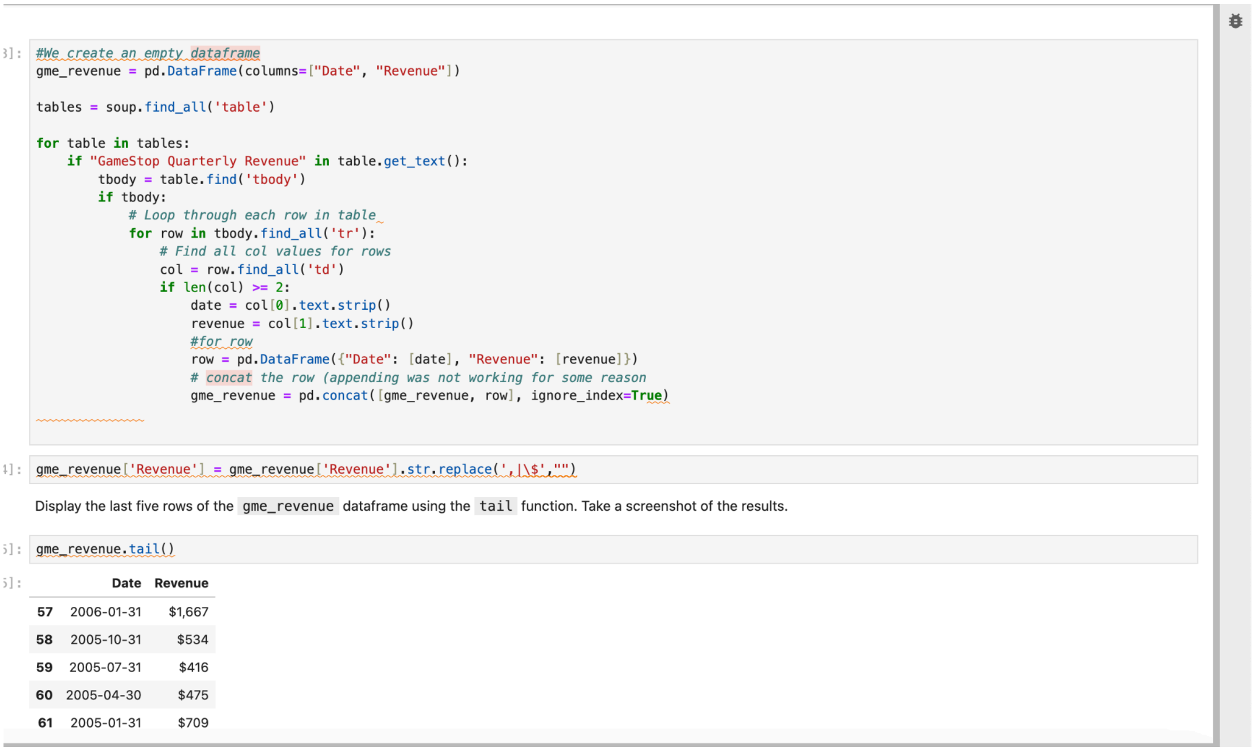
Using the ticker object and the function `history` extract stock information and save it in a dataframe named `gme_data`. Set the `period` parameter to `"max"` so we get information for the maximum amount of time.

```
Gme_data = GameStop.history(period="max")
```

Reset the index using the `reset_index(inplace=True)` function on the gme_data DataFrame and display the first five rows of the `gme_data` dataframe using the `head` function. Take a screenshot of the results and code from the beginning of Question 3 to the results below.

```
Gme_data.reset_index(inplace=True)
Gme_data.head()
```

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2002-02-13 00:00:00-05:00	1.620128	1.693350	1.603296	1.691666	76216000	0.0	0.0
1	2002-02-14 00:00:00-05:00	1.712707	1.716074	1.670626	1.683250	11021600	0.0	0.0
2	2002-02-15 00:00:00-05:00	1.683250	1.687458	1.658001	1.674834	8389600	0.0	0.0
3	2002-02-19 00:00:00-05:00	1.666418	1.666418	1.578047	1.607504	7410400	0.0	0.0
4	2002-02-20 00:00:00-05:00	1.615921	1.662210	1.603296	1.662210	6892800	0.0	0.0



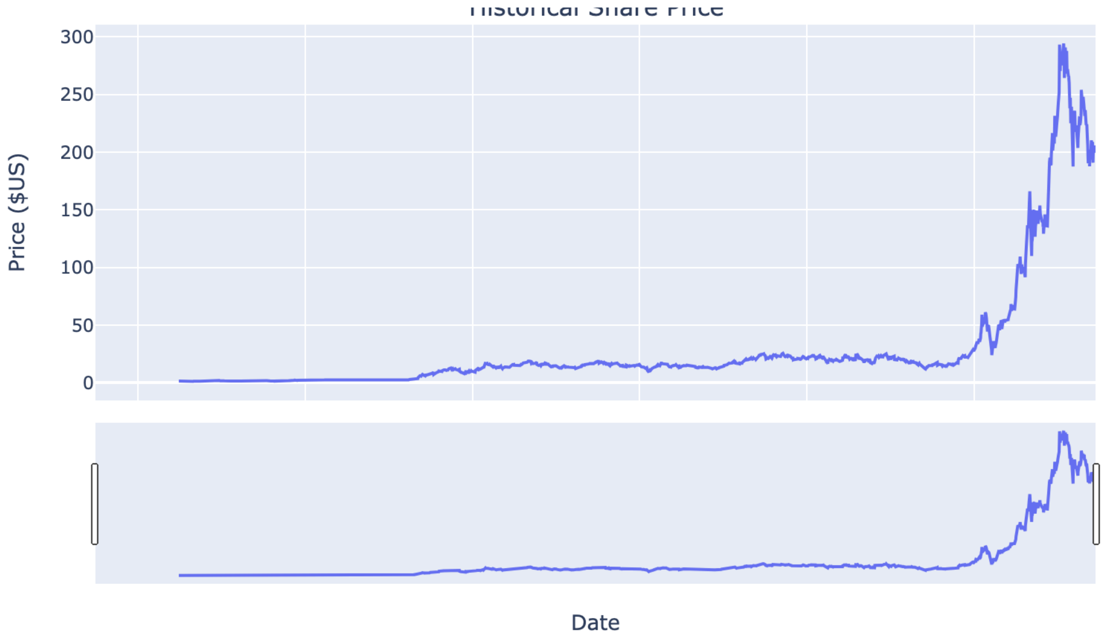
```
#We create an empty dataframe
gme_revenue = pd.DataFrame(columns=["Date", "Revenue"])

tables = soup.find_all('table')

for table in tables:
    if "GameStop Quarterly Revenue" in table.get_text():
        tbody = table.find('tbody')
        if tbody:
            # Loop through each row in table
            for row in tbody.find_all('tr'):
                # Find all col values for rows
                col = row.find_all('td')
                if len(col) >= 2:
                    date = col[0].text.strip()
                    revenue = col[1].text.strip()
                    #for row
                    row = pd.DataFrame({"Date": [date], "Revenue": [revenue]})
                    # concat the row (appending was not working for some reason
                    gme_revenue = pd.concat([gme_revenue, row], ignore_index=True)
```

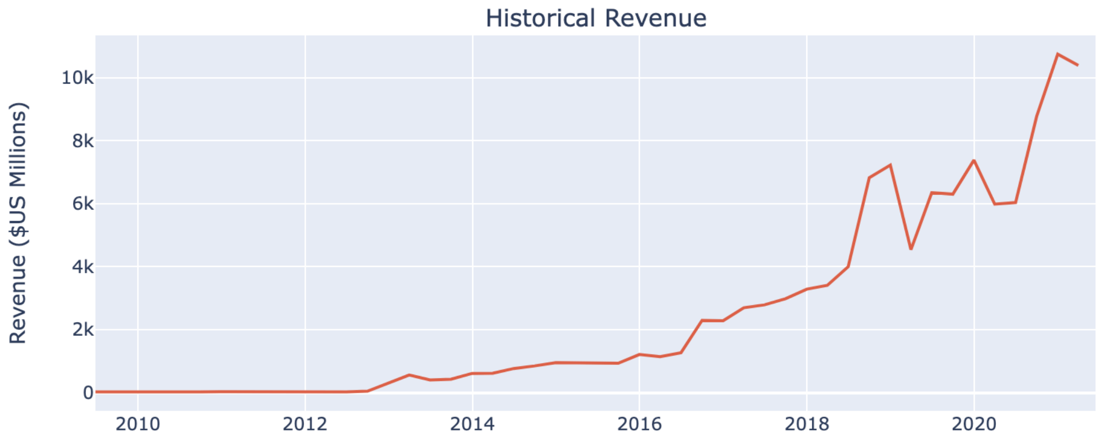
```
gme_revenue['Revenue'] = gme_revenue['Revenue'].str.replace(',|\$',"")
```

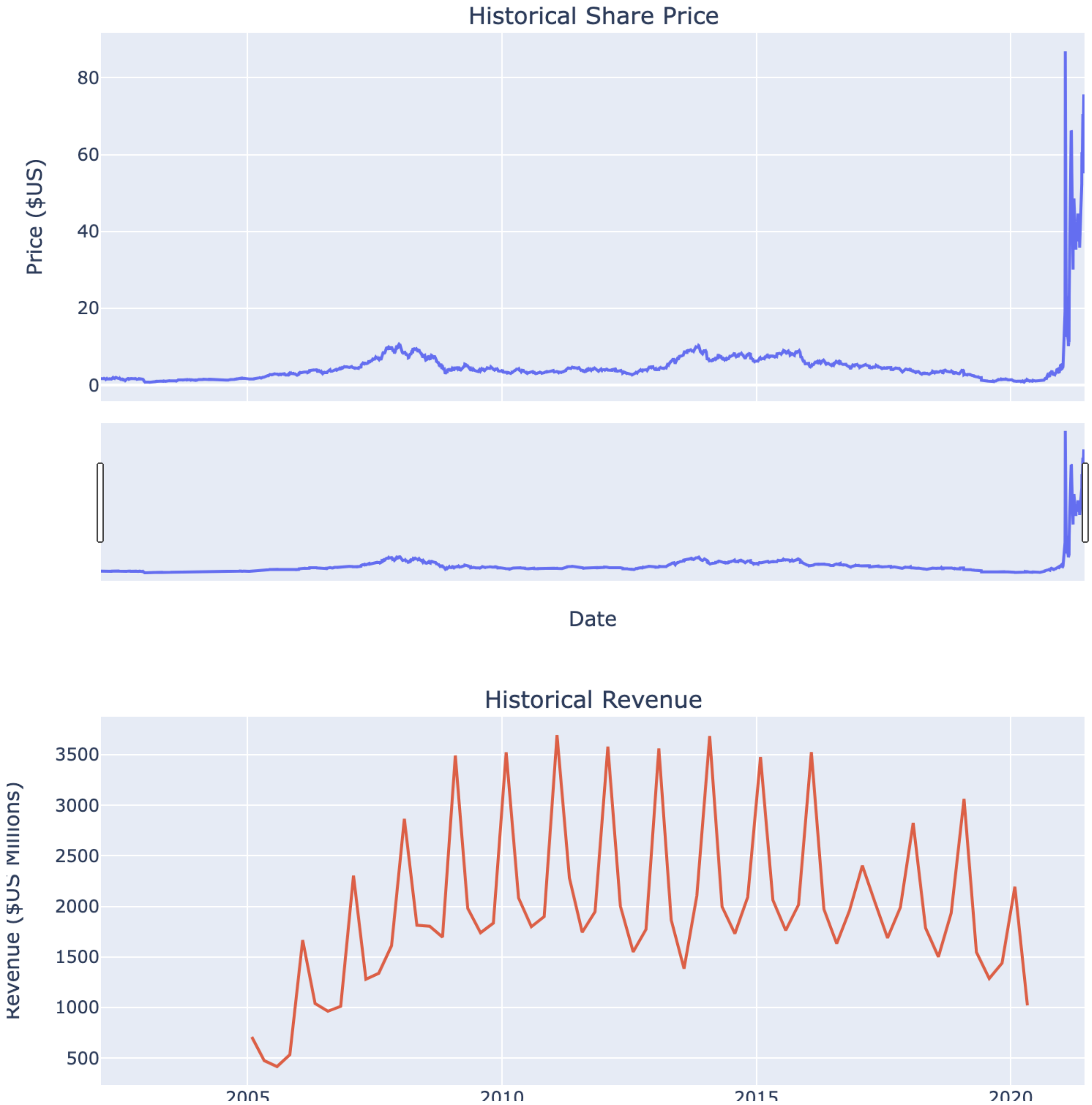
Display the last five rows of the `gme_revenue` dataframe using the `tail` function. Take a screenshot of the results.

```
gme_revenue.tail()
```

	Date	Revenue
57	2006-01-31	$1,667
58	2005-10-31	$534
59	2005-07-31	$416
60	2005-04-30	$475
61	2005-01-31	$709


Historical Share Price


Historical Revenue


Historical Share Price
Price ($US)
80
60
40
20
0
Date
Historical Revenue
Revenue ($US Millions)
3500
3000
2500
2000
1500
1000
500
2005
2010
2015
2020