# Stock Market Analysis Project

### We'll be analyzing stock data related to a few car companies, from Jan 1 2012 to Jan 1 2017.

In [2]:
```
import pandas as pd
import numpy as np
from pandas_datareader import data as web
import matplotlib.pyplot as plt
```

---

## Part 1: Getting the Data

#### Use pandas_datareader to obtain the historical stock information for Tesla, General Motors, Ford from Jan 1, 2012 to Jan 1, 2017.

### Stock data will be grabbed form yahoo finance to evaluation for the given period

In [6]:
```
tesla = web.DataReader("TSLA", data_source = "yahoo", start = "2012-1-1"
, end = "2017-1-1")
```

In [4]:
```
gm = web.DataReader("GM", data_source = "yahoo", start = "2012-1-1", end
= "2017-1-1")
```

In [5]:
```
ford = web.DataReader("F", data_source = "yahoo", start = "2012-1-1", en
d = "2017-1-1")
```

### we will inspect the last 5 days of each stock for the given period to see if the results are accurate

In [7]:
```
tesla.tail()
```

Out[7]:

| Date | High | Low | Open | Close | Volume | Adj Close |
|---|---|---|---|---|---|---|
| 2016-12-23 | 213.449997 | 207.710007 | 208.000000 | 213.339996 | 4670500 | 213.339996 |
| 2016-12-27 | 222.250000 | 214.419998 | 214.880005 | 219.529999 | 5915700 | 219.529999 |
| 2016-12-28 | 223.800003 | 217.199997 | 221.529999 | 219.740005 | 3782500 | 219.740005 |
| 2016-12-29 | 219.199997 | 214.119995 | 218.559998 | 214.679993 | 4045000 | 214.679993 |
| 2016-12-30 | 217.500000 | 211.679993 | 216.300003 | 213.690002 | 4642600 | 213.690002 |

In [8]:
```
ford.tail()
```

Out[8]:

| Date | High | Low | Open | Close | Volume | Adj Close |
|---|---|---|---|---|---|---|
| 2016-12-23 | 12.46 | 12.36 | 12.43 | 12.46 | 15621200.0 | 10.322713 |
| 2016-12-27 | 12.51 | 12.36 | 12.43 | 12.39 | 19467400.0 | 10.264721 |
| 2016-12-28 | 12.45 | 12.22 | 12.37 | 12.25 | 26875400.0 | 10.148735 |
| 2016-12-29 | 12.31 | 12.22 | 12.25 | 12.23 | 19819100.0 | 10.132168 |
| 2016-12-30 | 12.28 | 12.08 | 12.24 | 12.13 | 27405700.0 | 10.049319 |

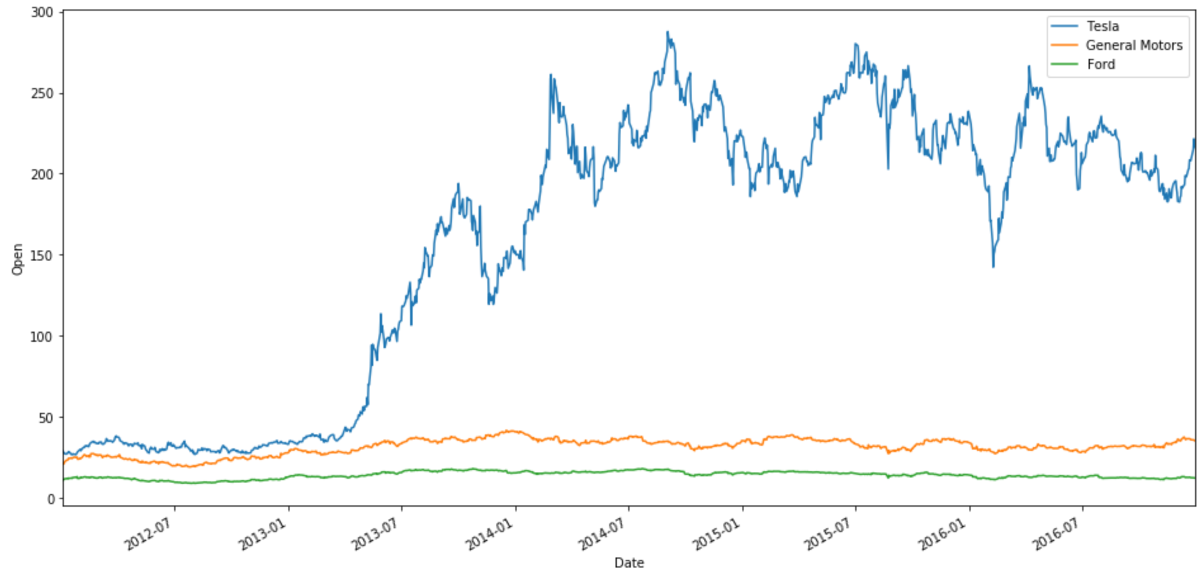
In [10]:
```
gm.tail()
```

Out[10]:

| Date | High | Low | Open | Close | Volume | Adj Close |
|---|---|---|---|---|---|---|
| 2016-12-23 | 35.869999 | 35.540001 | 35.830002 | 35.689999 | 9351200.0 | 31.276020 |
| 2016-12-27 | 35.930000 | 35.500000 | 35.799999 | 35.540001 | 6008700.0 | 31.144569 |
| 2016-12-28 | 35.799999 | 35.130001 | 35.740002 | 35.150002 | 8451900.0 | 30.802811 |
| 2016-12-29 | 35.480000 | 35.119999 | 35.250000 | 35.139999 | 4416700.0 | 30.794043 |
| 2016-12-30 | 35.310001 | 34.669998 | 35.209999 | 34.840000 | 7646100.0 | 30.531145 |

## Part 2: Visualizing the Data (Open for each Security)

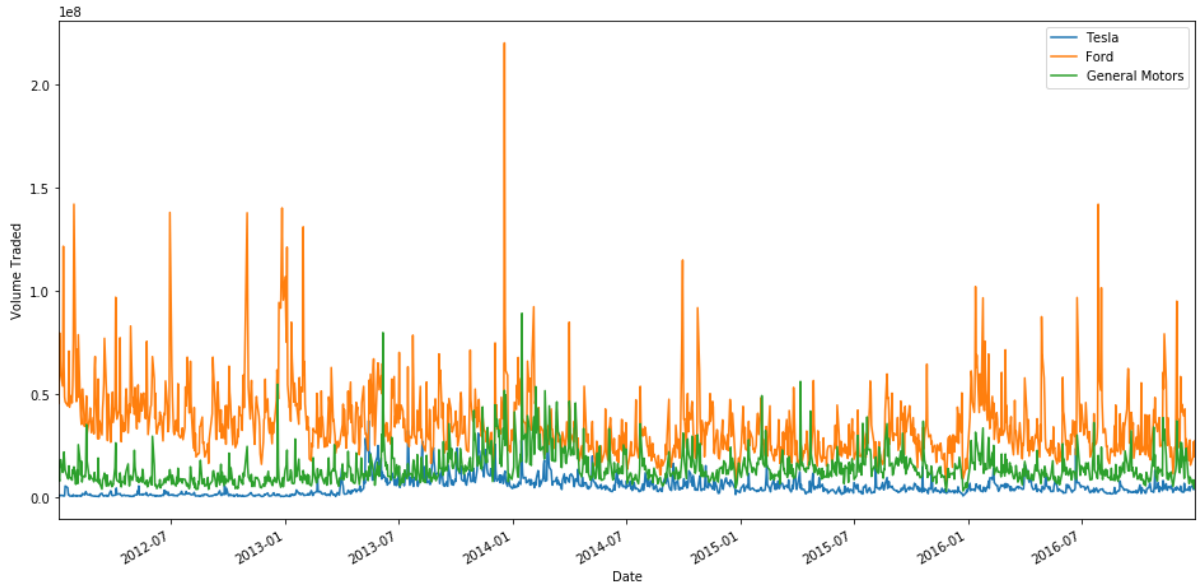
In [17]:
```
# Code Here
tesla["Open"].plot(figsize = (16,8), label = "Tesla")
gm["Open"].plot(label = "General Motors")
ford["Open"].plot(label = "Ford")
plt.ylabel("Open")
plt.legend();
```




## From first observation we can see that Tesla is far more volitile then Ford and GM and had never fallen below the value of Ford and GM in the given peroid. Tesla took huge spikes mad 2013, 2014, 2016, adn a big dip early 2016

#### Plot the Volume of stock traded each day.

In [20]:
```
tesla["Volume"].plot(figsize = (16,8), label = "Tesla")
ford["Volume"].plot(label = "Ford")
gm["Volume"].plot(label = "General Motors")
plt.ylabel("Volume Traded")
plt.legend();
```




## looks like Ford had a really big spike (Sell off) somewhere in late 2013. What was the date of this maximum trading volume for Ford? What happened that day?

In [24]:
```
ford["Volume"].idxmax()
```

Out[24]:
```
Timestamp('2013-12-18 00:00:00')
```

## This huge sell-off occurred on December 12th 2013 due company warning, the cost of its aggressive push to launch new products would cut into profits next year

# The Open Price Time Series Visualization makes Tesla look like its always been much more valuable as a company than GM and Ford. But to really understand this we would need to look at the total market cap of the company, not just the stock price. Unfortunately our current data doesn't have that information of total units of stock present. But what we can do as a simple calcualtion to try to represent total money traded would be to multply the Volume column by the Open price. Remember that this still isn't the actual Market Cap, its just a visual representation of the total amount of money being traded around using the time series.

#### Create a new column for each dataframe called "Total Traded" which is the Open Price multiplied by the Volume Traded.

In [25]:
```
# Code Here
tesla["Total Traded"] = tesla["Open"] * tesla["Volume"]
ford["Total Traded"] = ford["Open"] * ford["Volume"]
gm["Total Traded"] = gm["Open"] * gm["Volume"]
```

In [27]:
```
tesla.head()
```

Out[27]:

| Date | High | Low | Open | Close | Volume | Adj Close | Total Traded |
|---|---|---|---|---|---|---|---|
| 2012-01-03 | 29.500000 | 27.650000 | 28.940001 | 28.080000 | 928100 | 28.080000 | 2.685921e+07 |
| 2012-01-04 | 28.670000 | 27.500000 | 28.209999 | 27.709999 | 630100 | 27.709999 | 1.777512e+07 |
| 2012-01-05 | 27.930000 | 26.850000 | 27.760000 | 27.120001 | 1005500 | 27.120001 | 2.791268e+07 |
| 2012-01-06 | 27.790001 | 26.410000 | 27.200001 | 26.910000 | 986300 | 26.910000 | 2.682736e+07 |
| 2012-01-09 | 27.490000 | 26.120001 | 27.000000 | 27.250000 | 897000 | 27.250000 | 2.421900e+07 |

In [28]:
```
ford.head()
```

Out[28]:

| Date | High | Low | Open | Close | Volume | Adj Close | Total Traded |
|---|---|---|---|---|---|---|---|
| 2012-01-03 | 11.25 | 10.99 | 11.00 | 11.13 | 45709900.0 | 7.673051 | 5.028089e+08 |
| 2012-01-04 | 11.53 | 11.07 | 11.15 | 11.30 | 79725200.0 | 7.790251 | 8.889359e+08 |
| 2012-01-05 | 11.63 | 11.24 | 11.33 | 11.59 | 67877500.0 | 7.990177 | 7.690521e+08 |
| 2012-01-06 | 11.80 | 11.52 | 11.74 | 11.71 | 59840700.0 | 8.072903 | 7.025298e+08 |
| 2012-01-09 | 11.95 | 11.70 | 11.83 | 11.80 | 53981500.0 | 8.134951 | 6.386011e+08 |

In [29]:
```
gm.head()
```

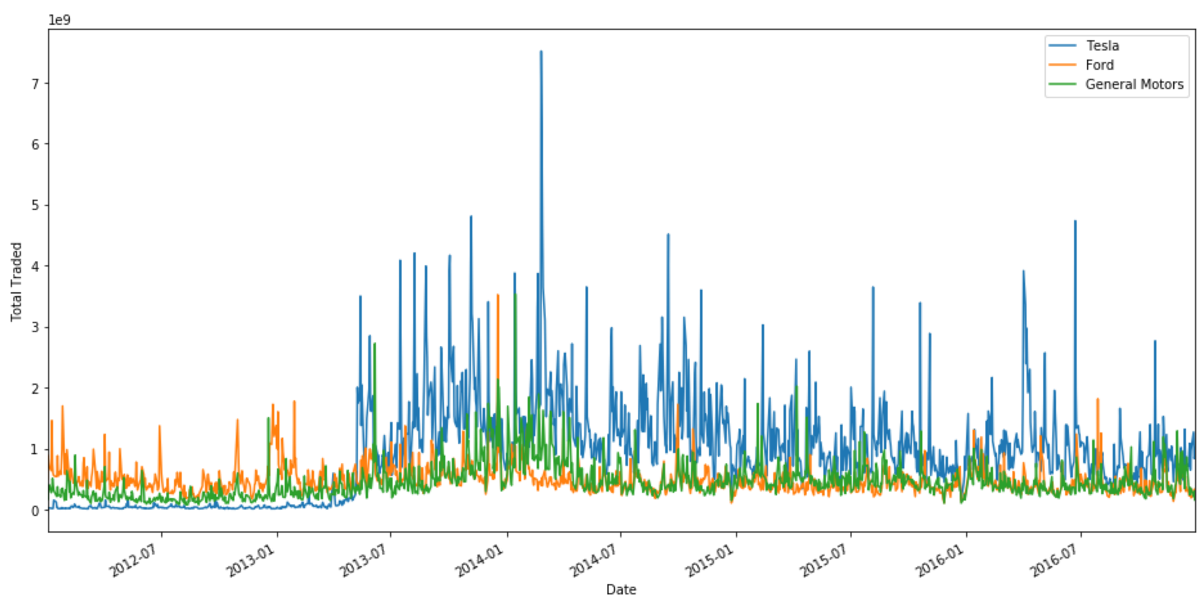
Out[29]:

| Date | High | Low | Open | Close | Volume | Adj Close | Total Traded |
|---|---|---|---|---|---|---|---|
| 2012-01-03 | 21.180000 | 20.750000 | 20.830000 | 21.049999 | 9321300.0 | 16.299799 | 1.941627e+08 |
| 2012-01-04 | 21.370001 | 20.750000 | 21.049999 | 21.150000 | 7856700.0 | 16.377232 | 1.653835e+08 |
| 2012-01-05 | 22.290001 | 20.959999 | 21.100000 | 22.170000 | 17880600.0 | 17.167059 | 3.772807e+08 |
| 2012-01-06 | 23.030001 | 22.240000 | 22.260000 | 22.920000 | 18234500.0 | 17.747812 | 4.059000e+08 |
| 2012-01-09 | 23.430000 | 22.700001 | 23.200001 | 22.840000 | 12084500.0 | 17.685862 | 2.803604e+08 |

**Plot this "Total Traded" against the time index.**

In [31]:
```
# Code here
tesla["Total Traded"].plot(figsize = (16,8), label = "Tesla")
ford["Total Traded"].plot(label = "Ford ")
gm["Total Traded"].plot(label = "General Motors")
plt.ylabel("Total Traded")
plt.legend()
```

Out[31]: <matplotlib.legend.Legend at 0x129621f90>




## Interesting, looks like there was huge amount of money traded for Tesla somewhere in early 2014. What date was that and what happened? **

In [34]:
```
tesla["Total Traded"].idxmax()
```

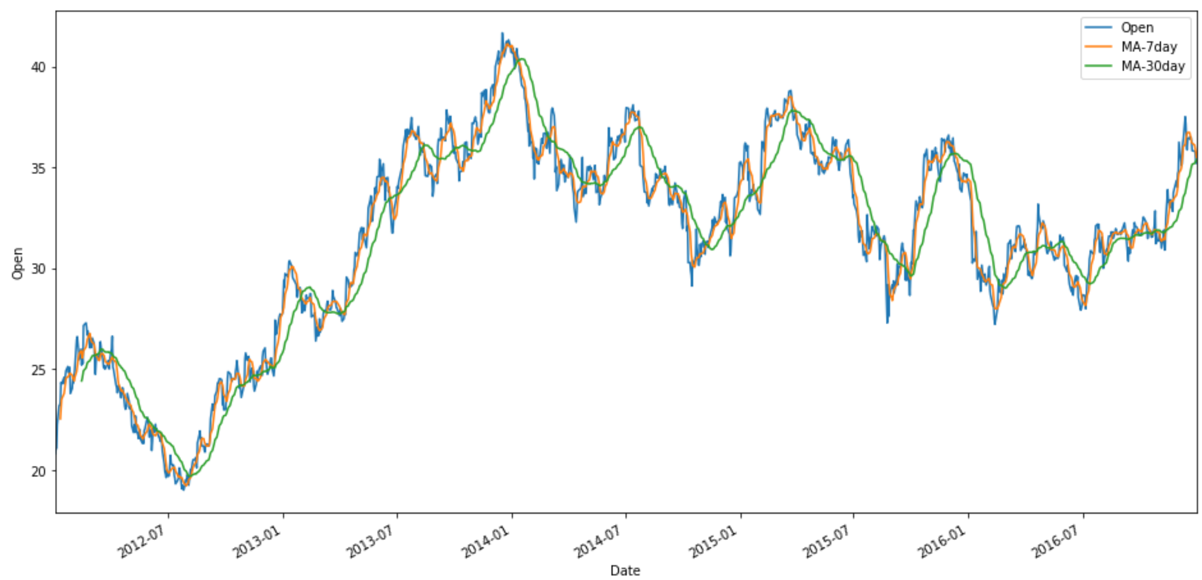
Out[34]: Timestamp('2014-02-25 00:00:00')

### The stock gained more than 30% on feb 25 2014 since Tesla (TSLA) reported much stronger-than-expected profit and raised its sales targets last week

---

### Let's plot out some MA (Moving Averages). Plot out the MA 7 (weekly) and MA 30 (Monthly) for GM. **

In [109]:
```
gm["Open"].plot(figsize = (16,8), label = "Open")
gm.rolling(7).mean()["Close"].plot(label = "MA-7day")
gm.rolling(30).mean()["Close"].plot(label = "MA-30day")
plt.ylabel("Open")
plt.legend();
```




### The 7 day moving average is not a good sample with such a large data set and timeline. The 30 days moving average allows up to make some predictions based on the previous 30 days of values. We can use the MA to focus on the trends of the stock.

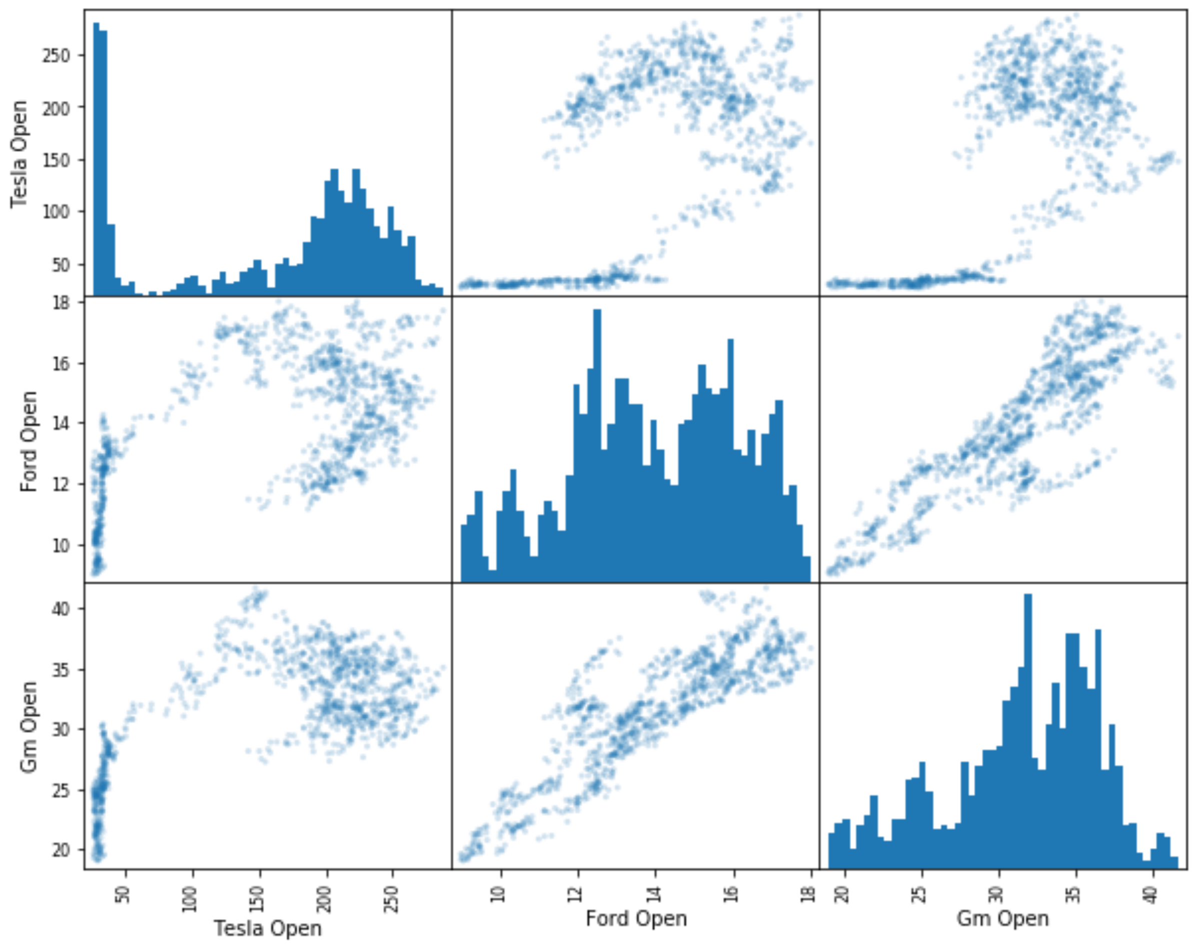
---

### Finally lets see if there is a relationship between these stocks, after all, they are all related to the car industry. We can see if this this is true through a scatter matrix plot using the Open of the stocks

In [38]:
```
from pandas.plotting import scatter_matrix
```

In [43]:
```
open_df = pd.concat([tesla["Open"], ford["Open"], gm["Open"]], axis=1)
open_df.columns = ["Tesla Open", "Ford Open", "Gm Open"]
```

In [50]:
```
scatter_matrix(open_df, figsize=(10,8),alpha = 0.2, hist_kwds= {"bins":
50});
```




#### the scatter matrix shows that there is an apparent correlation between Ford and General Motors but not much of a relation to Tesla from Ford and GM. This could be proof to the theory the Tesla is treated more like a tech company than a car company

## Daily Percentage Change

First we will begin by calculating the daily percentage change

#### Will create a new column for each dataframe called returns. This column will be calculated from the Close price column.

In [51]:
```
tesla["returns"] = tesla["Close"].pct_change()
```

In [53]:
```
ford["returns"] = ford["Close"].pct_change()
```

In [54]:
```
gm["returns"] = gm["Close"].pct_change()
```

In [56]:
```
tesla.head()
```

Out[56]:

| Date | High | Low | Open | Close | Volume | Adj Close | Total Traded | returns |
|---|---|---|---|---|---|---|---|---|
| 2012-01-03 | 29.500000 | 27.650000 | 28.940001 | 28.080000 | 928100 | 28.080000 | 2.685921e+07 | NaN |
| 2012-01-04 | 28.670000 | 27.500000 | 28.209999 | 27.709999 | 630100 | 27.709999 | 1.777512e+07 | -0.013177 |
| 2012-01-05 | 27.930000 | 26.850000 | 27.760000 | 27.120001 | 1005500 | 27.120001 | 2.791268e+07 | -0.021292 |
| 2012-01-06 | 27.790001 | 26.410000 | 27.200001 | 26.910000 | 986300 | 26.910000 | 2.682736e+07 | -0.007743 |
| 2012-01-09 | 27.490000 | 26.120001 | 27.000000 | 27.250000 | 897000 | 27.250000 | 2.421900e+07 | 0.012635 |

In [57]:
```
ford.tail()
```

Out[57]:

| Date | High | Low | Open | Close | Volume | Adj Close | Total Traded | returns |
|---|---|---|---|---|---|---|---|---|
| 2016-12-23 | 12.46 | 12.36 | 12.43 | 12.46 | 15621200.0 | 10.322713 | 1.941715e+08 | 0.004839 |
| 2016-12-27 | 12.51 | 12.36 | 12.43 | 12.39 | 19467400.0 | 10.264721 | 2.419798e+08 | -0.005618 |
| 2016-12-28 | 12.45 | 12.22 | 12.37 | 12.25 | 26875400.0 | 10.148735 | 3.324487e+08 | -0.011299 |
| 2016-12-29 | 12.31 | 12.22 | 12.25 | 12.23 | 19819100.0 | 10.132168 | 2.427840e+08 | -0.001633 |
| 2016-12-30 | 12.28 | 12.08 | 12.24 | 12.13 | 27405700.0 | 10.049319 | 3.354458e+08 | -0.008177 |

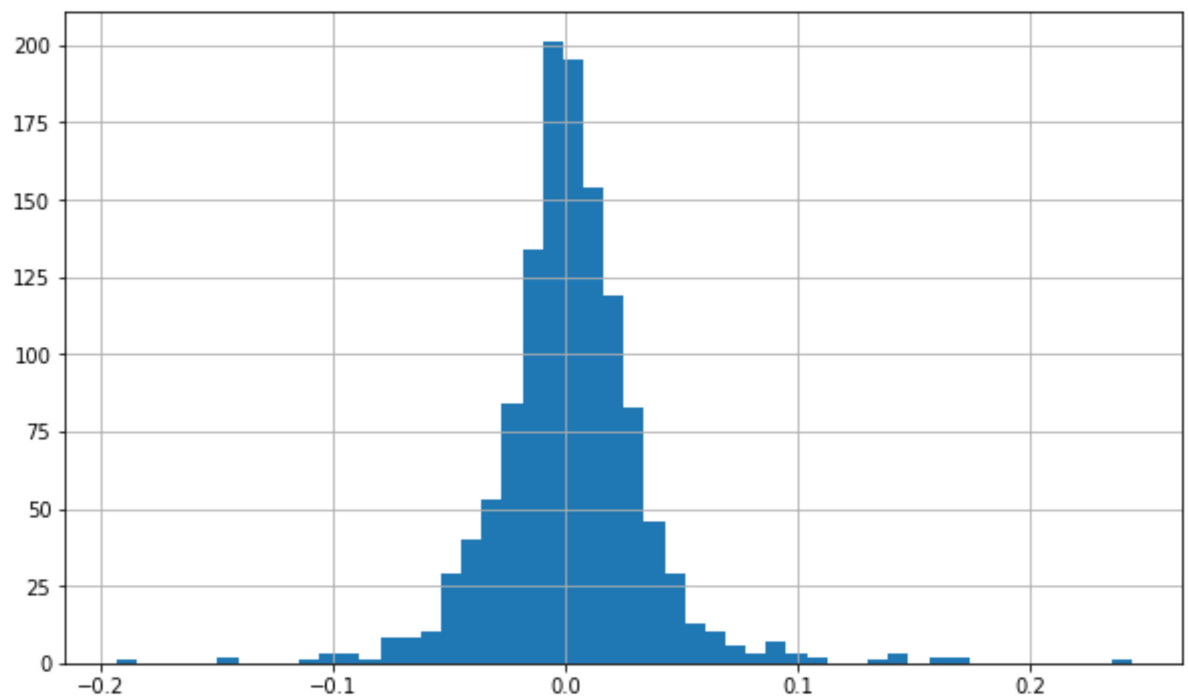
In [58]:
```
gm.tail()
```

Out[58]:

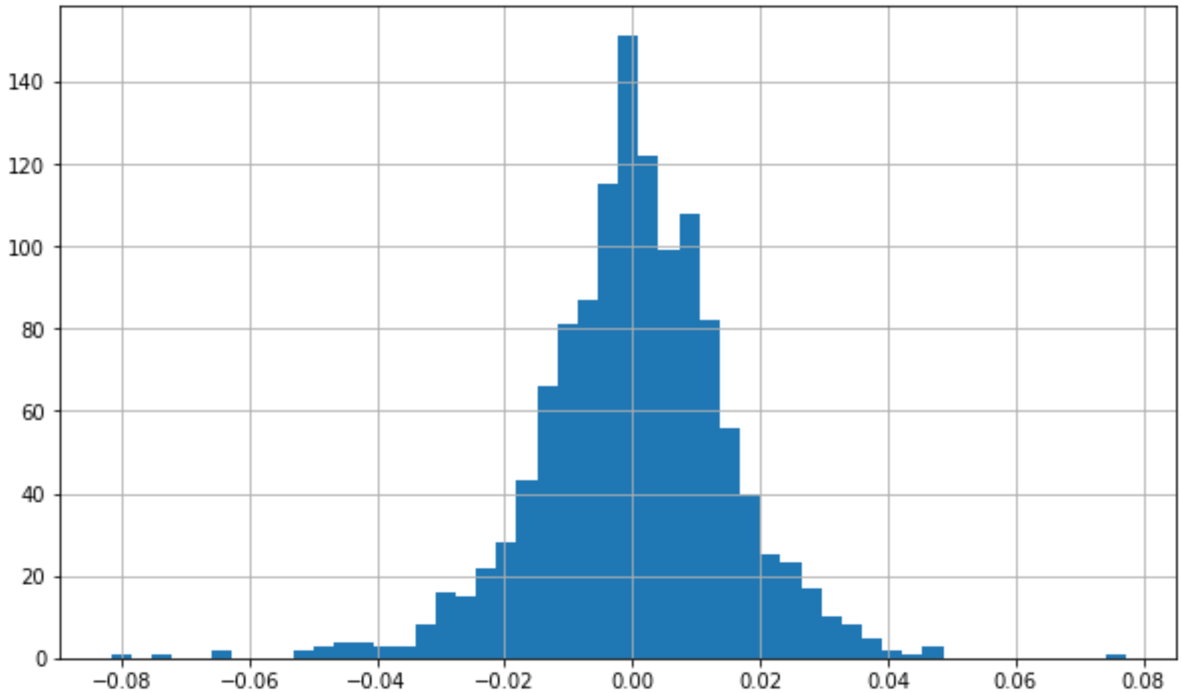
| | High | Low | Open | Close | Volume | Adj Close | Total Traded | returns |
|---|---|---|---|---|---|---|---|---|
| **Date** | | | | | | | | |
| 2016-12-23 | 35.869999 | 35.540001 | 35.830002 | 35.689999 | 9351200.0 | 31.276020 | 3.350535e+08 | 0.00000 |
| 2016-12-27 | 35.930000 | 35.500000 | 35.799999 | 35.540001 | 6008700.0 | 31.144569 | 2.151115e+08 | -0.00420 |
| 2016-12-28 | 35.799999 | 35.130001 | 35.740002 | 35.150002 | 8451900.0 | 30.802811 | 3.020709e+08 | -0.01097 |
| 2016-12-29 | 35.480000 | 35.119999 | 35.250000 | 35.139999 | 4416700.0 | 30.794043 | 1.556887e+08 | -0.00028 |
| 2016-12-30 | 35.310001 | 34.669998 | 35.209999 | 34.840000 | 7646100.0 | 30.531145 | 2.692192e+08 | -0.00853 |

## Will plot a histogram of each companies returns. Which is more volitile?

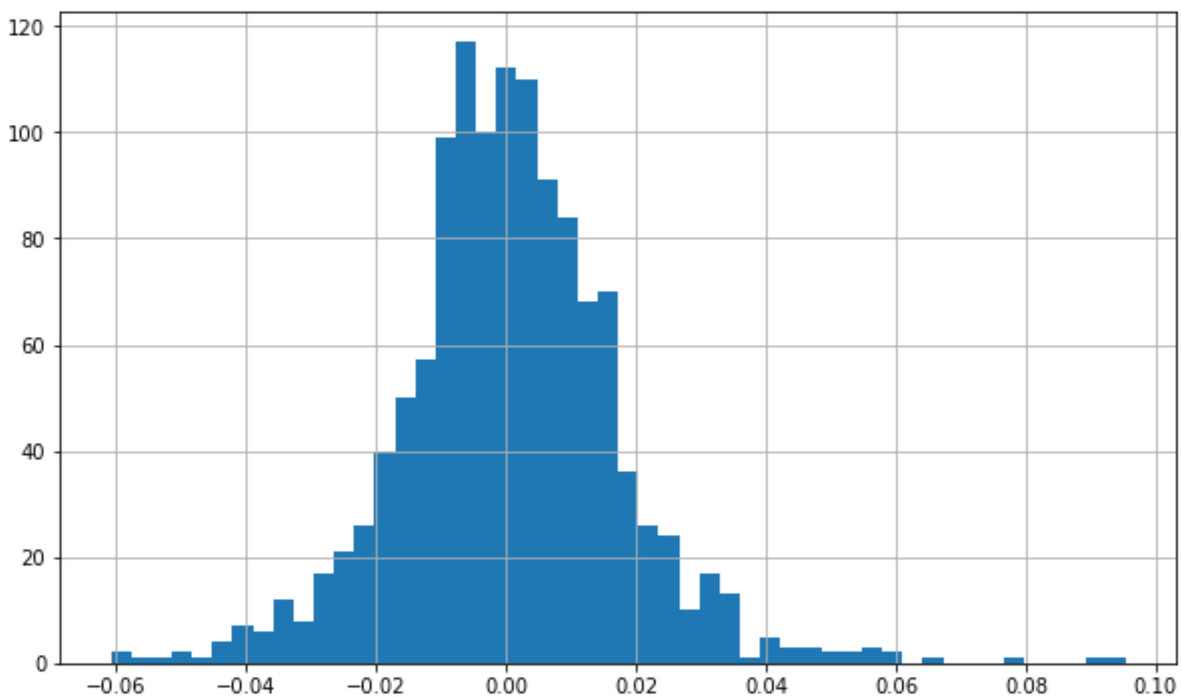
In [63]:
```
#Tesla
tesla["returns"].hist(figsize = (10,6), bins = 50);
```



In [62]:
```
#ford
ford["returns"].hist(figsize = (10,6), bins = 50);
```

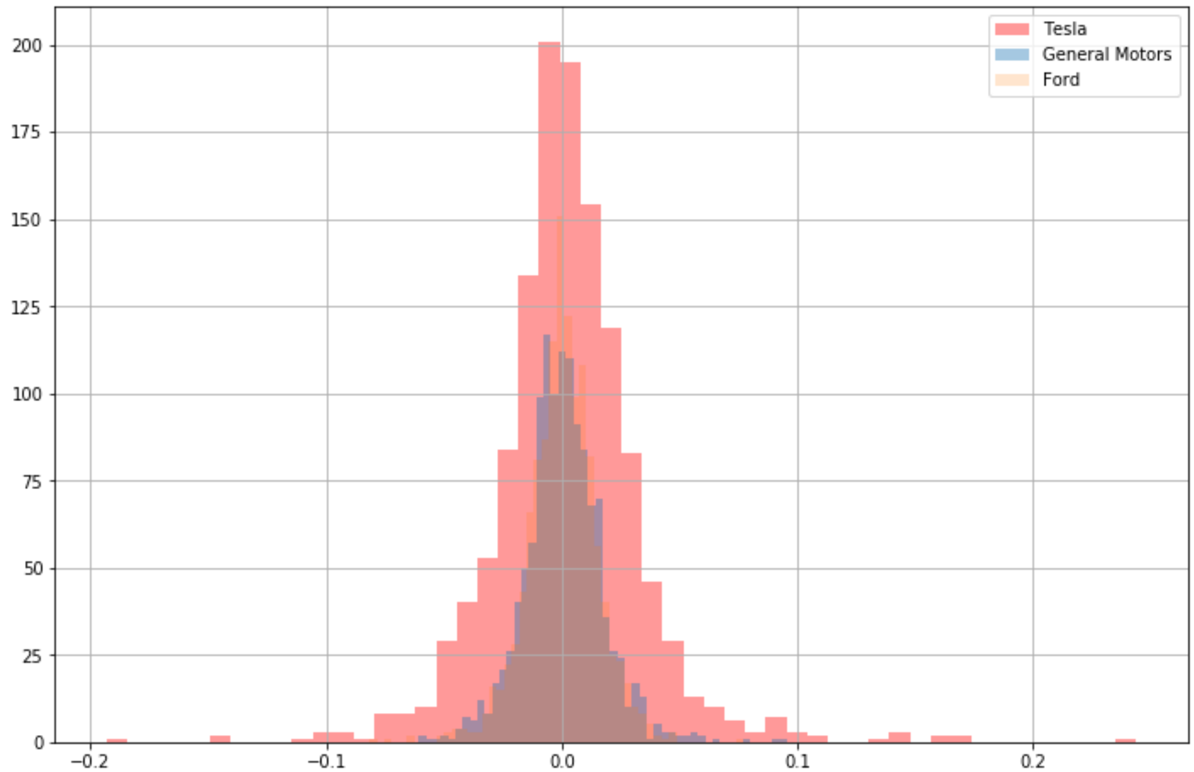



In [65]:
```
#GM
gm["returns"].hist(figsize = (10,6), bins = 50);
```




## Tesla certainly has the high volitility of the 3 companies rangining with some outlyers of a 20% loss and a 20% gain. We will plot all 3 returns on on plot to get a better sense of this

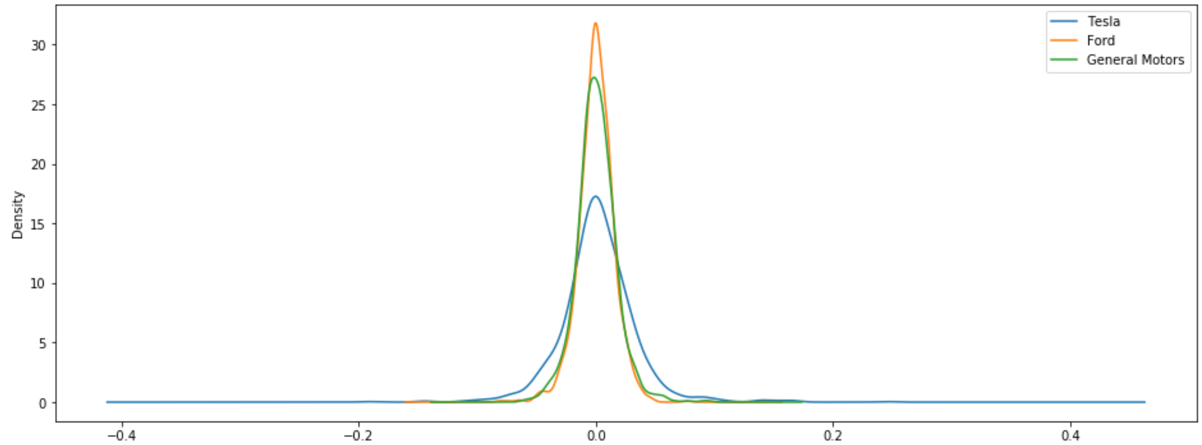
In [81]:
```
tesla["returns"].hist(label = "Tesla", figsize = (12,8), bins = 50, alph
a = 0.4, color = "r")
gm["returns"].hist(label = "General Motors", alpha = 0.4, bins = 50)
ford["returns"].hist(label = "Ford", alpha = 0.2, bins = 50)
plt.legend();
```




### Tesla is apparent to be the most volitile of the 3 stocks

## Plot a KDE instead of histograms for another view point. Which stock has the widest plot?

In [85]:
```
tesla["returns"].plot(kind = "kde", figsize = (16,6), label = 'Tesla')
ford["returns"].plot(kind = "kde", label = "Ford")
gm["returns"].plot(kind = "kde", label = "General Motors")
plt.legend();
```



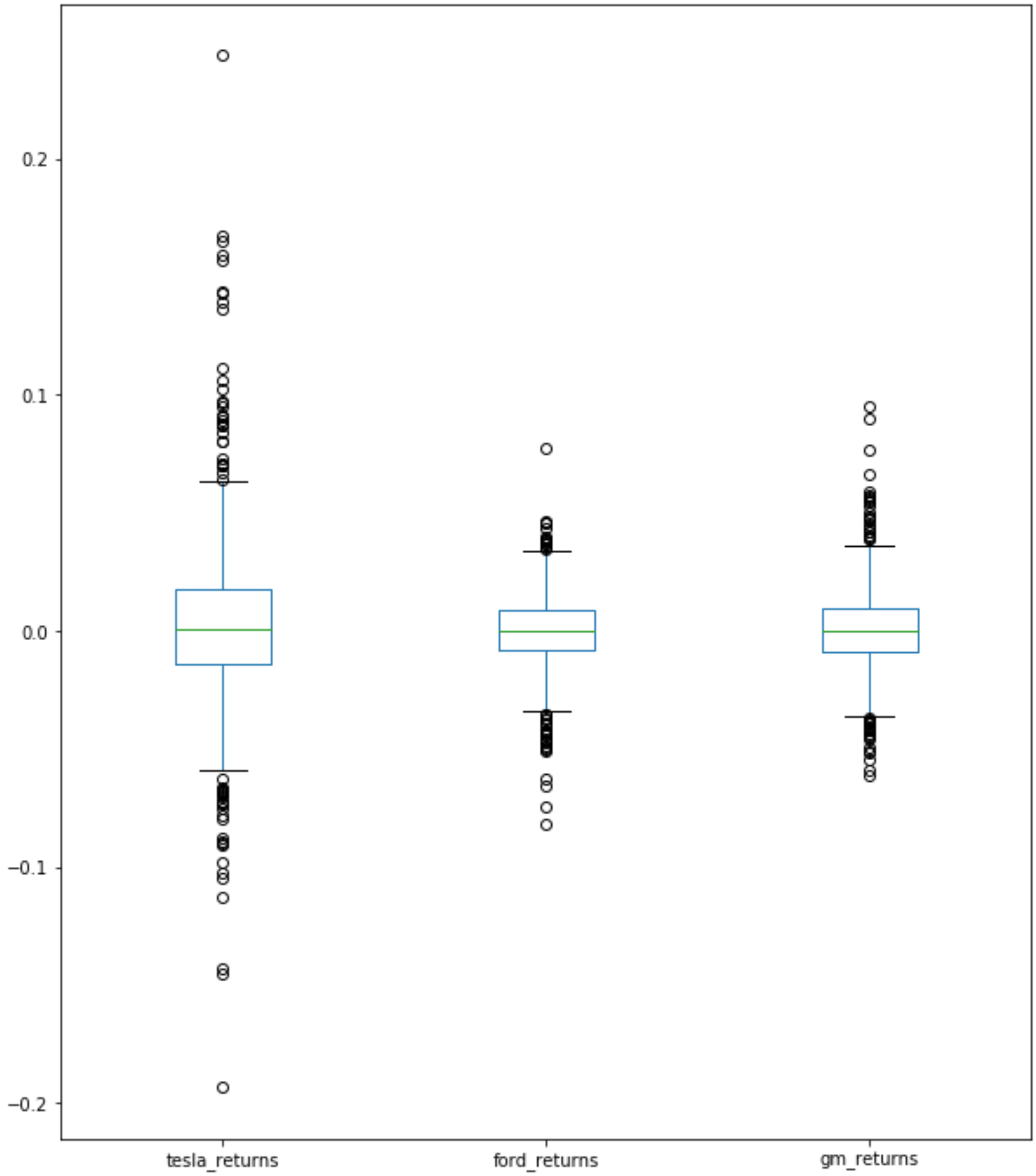

### Ford displays the lease volitility of the 3 stocks with GM falling in second. Tesla has the widest plot making it the most volitile

---

### We will test again using a box plot to test the volitility of the three socks. a new Data frame will need to be created consisting of the returns/daily change of each stocks.

In [86]:
```
df_returns = pd.concat([tesla["returns"], ford["returns"], gm["returns"
]], axis=1)
df_returns.columns = ["tesla_returns",  'ford_returns', "gm_returns"]
```

In [89]:
```
df_returns.plot(kind = "box", figsize = (10,12));
```

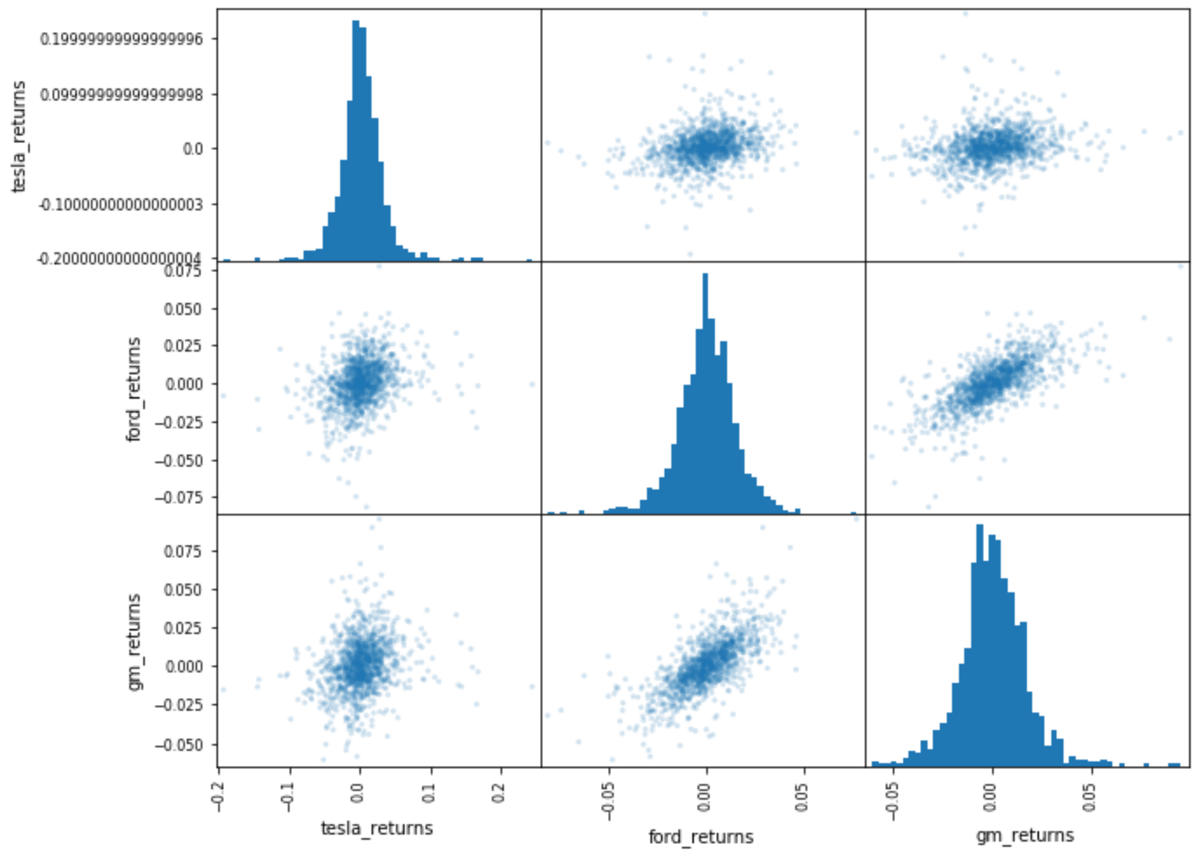



### Tesla again presents a more volitile behavior on the returns/Daily Changes

# Comparing Daily Returns between Stocks

### We will create a scatter matrix plot to see the correlation between each of the stocks daily returns. This helps answer the questions of how related the car companies are. Is Tesla begin treated more as a technology company rather than a car company by the market?

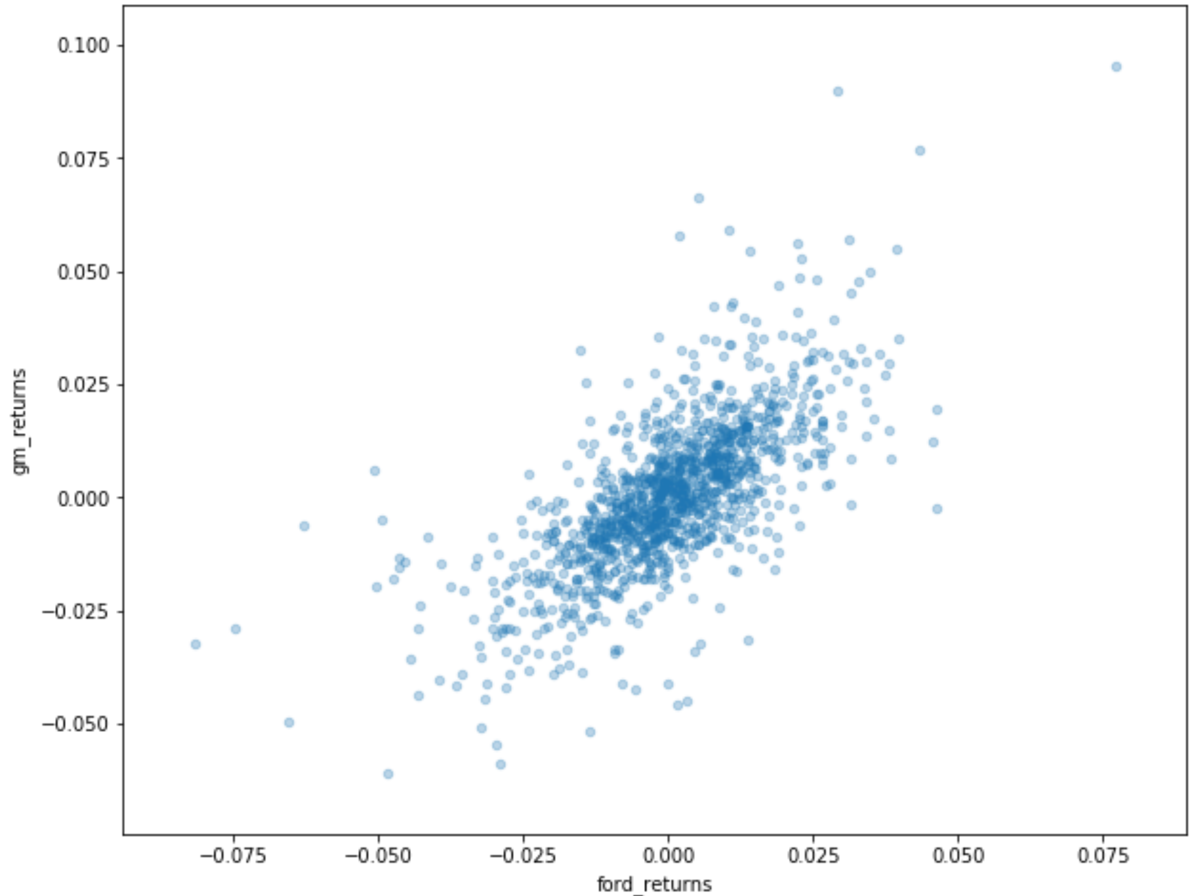
In [94]:
```
scatter_matrix(df_returns, figsize=(10,8), hist_kwds= {"bins": 50},alpha
= 0.2);
```




**WE see a clear correlation between Ford and GM, but the behavior of Tesla seems to be more independent of the other two car companies**

**It looks like Ford and GM do have some sort of possible relationship, let's plot just these two against eachother in scatter plot to view this more closely!****

In [97]:
```
df_returns.plot(kind = "scatter", x = "ford_returns", y = "gm_returns",
alpha = 0.3, figsize = (10,8));
```




## Cumulative Daily Returns

Now we can see which stock was the most wide ranging in daily returns. Tesla!

With daily cumulative returns, the question we are trying to answer is the following, if I invested $1 in the company at the beginning of the time series, how much would is be worth today? This is different than just the stock price at the current day, because it will take into account the daily returns. Our simple calculation here won't take into account stocks that give back a dividend.

In [98]:
```
tesla["Daily Return"] = (1 + tesla["returns"]).cumprod()
ford["Daily Return"] = (1 + ford["returns"]).cumprod()
gm["Daily Return"] = (1 + gm["returns"]).cumprod()
```

In [99]:
```
tesla.head()
```

Out[99]:

| Date | High | Low | Open | Close | Volume | Adj Close | Total Traded | returns |
|---|---|---|---|---|---|---|---|---|
| 2012-01-03 | 29.500000 | 27.650000 | 28.940001 | 28.080000 | 928100 | 28.080000 | 2.685921e+07 | NaN |
| 2012-01-04 | 28.670000 | 27.500000 | 28.209999 | 27.709999 | 630100 | 27.709999 | 1.777512e+07 | -0.013177 |
| 2012-01-05 | 27.930000 | 26.850000 | 27.760000 | 27.120001 | 1005500 | 27.120001 | 2.791268e+07 | -0.021292 |
| 2012-01-06 | 27.790001 | 26.410000 | 27.200001 | 26.910000 | 986300 | 26.910000 | 2.682736e+07 | -0.007743 |
| 2012-01-09 | 27.490000 | 26.120001 | 27.000000 | 27.250000 | 897000 | 27.250000 | 2.421900e+07 | 0.012635 |

In [100]:
```
gm.head()
```

Out[100]:

| Date | High | Low | Open | Close | Volume | Adj Close | Total Traded | returr |
|---|---|---|---|---|---|---|---|---|
| 2012-01-03 | 21.180000 | 20.750000 | 20.830000 | 21.049999 | 9321300.0 | 16.299799 | 1.941627e+08 | Na |
| 2012-01-04 | 21.370001 | 20.750000 | 21.049999 | 21.150000 | 7856700.0 | 16.377232 | 1.653835e+08 | 0.0047 |
| 2012-01-05 | 22.290001 | 20.959999 | 21.100000 | 22.170000 | 17880600.0 | 17.167059 | 3.772807e+08 | 0.0482 |
| 2012-01-06 | 23.030001 | 22.240000 | 22.260000 | 22.920000 | 18234500.0 | 17.747812 | 4.059000e+08 | 0.0338 |
| 2012-01-09 | 23.430000 | 22.700001 | 23.200001 | 22.840000 | 12084500.0 | 17.685862 | 2.803604e+08 | -0.0034 |

In [101]:
```
ford.head()
```

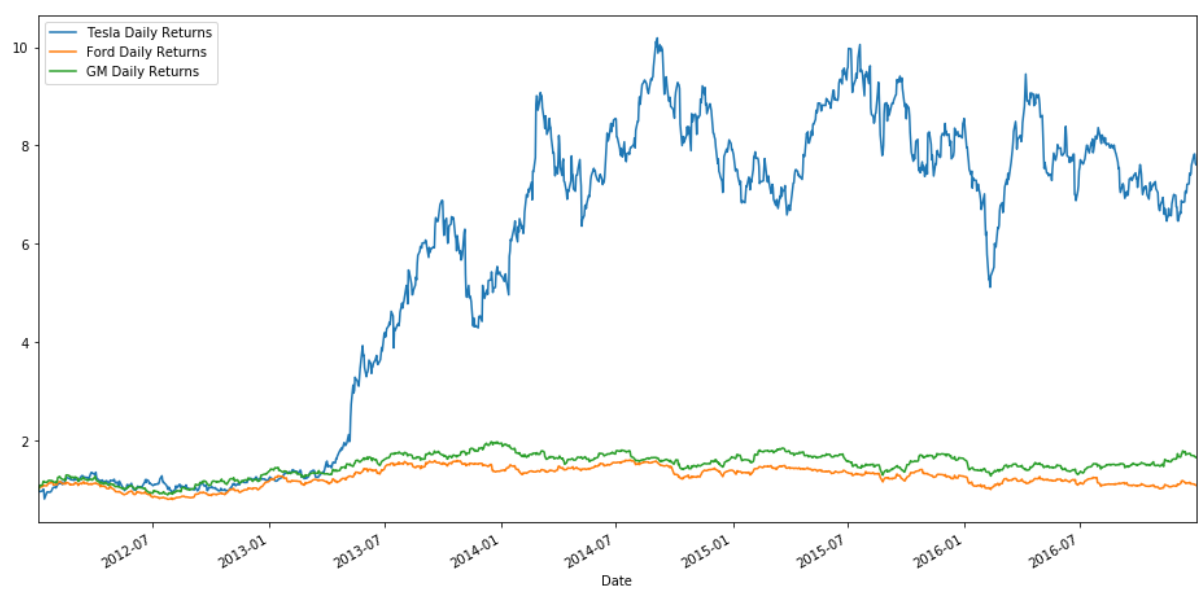
Out[101]:

| Date | High | Low | Open | Close | Volume | Adj Close | Total Traded | returns | Daily Return |
|---|---|---|---|---|---|---|---|---|---|
| 2012-01-03 | 11.25 | 10.99 | 11.00 | 11.13 | 45709900.0 | 7.673051 | 5.028089e+08 | NaN | NaN |
| 2012-01-04 | 11.53 | 11.07 | 11.15 | 11.30 | 79725200.0 | 7.790251 | 8.889359e+08 | 0.015274 | 1.015274 |
| 2012-01-05 | 11.63 | 11.24 | 11.33 | 11.59 | 67877500.0 | 7.990177 | 7.690521e+08 | 0.025664 | 1.041330 |
| 2012-01-06 | 11.80 | 11.52 | 11.74 | 11.71 | 59840700.0 | 8.072903 | 7.025298e+08 | 0.010354 | 1.052111 |
| 2012-01-09 | 11.95 | 11.70 | 11.83 | 11.80 | 53981500.0 | 8.134951 | 6.386011e+08 | 0.007686 | 1.060198 |

## Plot of the Cumulative Return columns against the time series index. Which stock showed the highest return ?

In [104]:
```
tesla["Daily Return"].plot(figsize = (16,8), label = "Tesla Daily Return
s")
ford["Daily Return"].plot(label = "Ford Daily Returns")
gm["Daily Return"].plot(label = "GM Daily Returns ")
plt.legend()
```

Out[104]: <matplotlib.legend.Legend at 0x1a3067e6d0>



#### Tesla being the most volitile of the 3 stocks also generated the highest return and ford being the least volitile provided the lowest return of the stocks.