Danny Vilela

Attempting the Impossible[1]**:**
Interpreting and Predicting Tesla's Stock Valuation

DANNY VILELA
New York University
April 13, 2016

Introduction

It's no surprise that Tesla Motors has been taking the technology and automotive worlds by storm. The promise of clean-fuel cars that offer top-notch engineering, safety ratings above any car ever tested, and modern aesthetic at an affordable price is, seemingly, pie-in-the-sky – but it couldn't be closer to reality. Tesla's 2012 IPO signaled the dawn of a tech-automotive empire, with CEO Elon Musk leading the charge. Elon, in fact, doesn't settle for just CEO of Tesla Motors: he's also CEO and CTO of SpaceX and Chairman of SolarCity, two high-profile tech innovations that, despite seeming too pie-in-the-sky, are led by someone who has finished his first and is reaching for his second slice.

With the recent record-setting preorder numbers for the Tesla Model 3, speculating over Tesla Motors' stock and attempting to predict its performance is at the tip of every formally-trained financial analysts tongue. Tesla has grown and increased sales at every quarter, and the opportunity to invest in Tesla could pay for itself many times over.

Problem

With weekly financial data recording Tesla Motors (TSLA) stock price, we can easily view the company's growth as a growing financial power as well as its consistent stock market uptick. With time series data, however, we note a particular classification of problems we may have to deal with – including autocorrelation – that prove to be more involved than previous regressions.

With regards to problem domain, this problem is one of futility: it's difficult to believe that a novice attempt at modeling and predicting stock prices will prove insightful, but we do so regardless for the sake of data analysis. Using this data, an analyst would most likely be interested in the stock's high so that, once reached, they can optimize their profits. That said, we define our

[1]By "impossible" I mean "meaningless", of course. Stock prediction, like we've discussed in class, hopefully isn't this easy – but it's worth a shot.

problem: **how do we best represent and predict Tesla's future stock price?**

Naturally, we admit that there are numerous external factors (e.g. press coverage) and complex dimensions (e.g. investor relations) that we won't be able to quantify and, therefore, include in our model. That said, we look to our provided predictors to begin building our model:

1. our stock's price at opening: `Open`
2. our stock's price at closing: `Close`
3. the number of millions of transactions in which our stock was involved: `Volume`
4. our stock's relative search volume[2] as reported by Google: `Search.Interest`

Of our predictors, we note that `Search.Interest` was not provided by Yahoo Finance. I decided to include the historical Google Trend data for our stock, which reflects the relative search volume for our search terms "tesla". Given Google's role as gatekeeper to online information, it at least seems plausible that an increase in "tesla" searches might have a direct impact on tomorrow's stock price high – after all, even traders use Google. We choose to naïvely proceed with our model and make any adjustments as we go along.

Model Definition

We can represent our earlier assertion through the regression model:

$$\texttt{High} = \beta_0 + \beta_1 \times \texttt{Open} + \beta_2 \times \texttt{Close} + \beta_3 \times \texttt{Volume} + \beta_4 \times \texttt{Search.Interest}$$

Data Sourcing and Cleaning

I sourced my data by using Yahoo Finance's publicly-available dataset of daily, weekly, and monthly records for publicly-traded companies. Furthermore, I utilized Google Trends data for the search query "tesla" in order to obtain the `Search.Interest` data. Both datasets contained weekly data, so naturally we join both datasets together by pairing their respective dates.

Aside from some basic sorting of dates and removing some of the columns within our data that were not used within the model, our data was fairly clean.

[2] see final section 'Data' for an explanation of our Google Trend predictor, `Search.Interest`

Exploratory Data Analysis

First, let's take a quick look at a summary of our provided data:

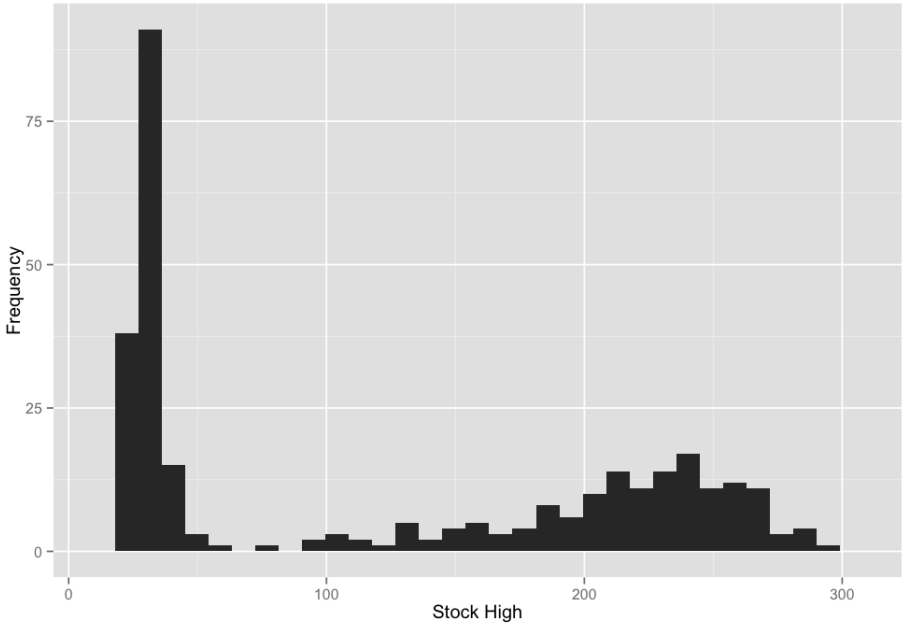
```
> summary(tesla_data)
```

```
     Date                 Open              High
 Min.   :2010-07-05 | Min.   : 17.95 | Min.   : 19.59 |
 1st Qu.:2011-12-13 | 1st Qu.: 29.27 | 1st Qu.: 30.45 |
 Median :2013-05-24 | Median : 93.81 | Median :100.23 |
 Mean   :2013-05-23 | Mean   :118.25 | Mean   :124.06 |
 3rd Qu.:2014-11-01 | 3rd Qu.:210.20 | 3rd Qu.:224.12 |
 Max.   :2016-04-11 | Max.   :278.88 | Max.   :291.42 |

     Low              Close            Volume          Search.Interest
 Min.   : 14.98 | Min.   : 17.40 | Min.   :  0.2597 | Min.   :  6.00 |
 1st Qu.: 27.51 | 1st Qu.: 29.34 | 1st Qu.:  1.1502 | 1st Qu.:  7.00 |
 Median : 85.65 | Median : 97.42 | Median :  2.9906 | Median : 13.00 |
 Mean   :112.81 | Mean   :118.79 | Mean   :  4.2178 | Mean   : 13.74 |
 3rd Qu.:203.76 | 3rd Qu.:210.98 | 3rd Qu.:  6.0334 | 3rd Qu.: 18.00 |
 Max.   :273.66 | Max.   :280.02 | Max.   : 23.4151 | Max.   :100.00 |
```

The summary data reveals a few interesting insights, if only surface-level:

1. Our `Open`, `High`, and `Low` features show an interesting progression in terms of investor interest and stock valuation. To have a stock value range from ∼$20 to ∼$280 tells us a lot about the stock's value and potentially volatility over time.
2. Our `Volume` feature is also very volatile, with its minimum approximately 16 times below the average.

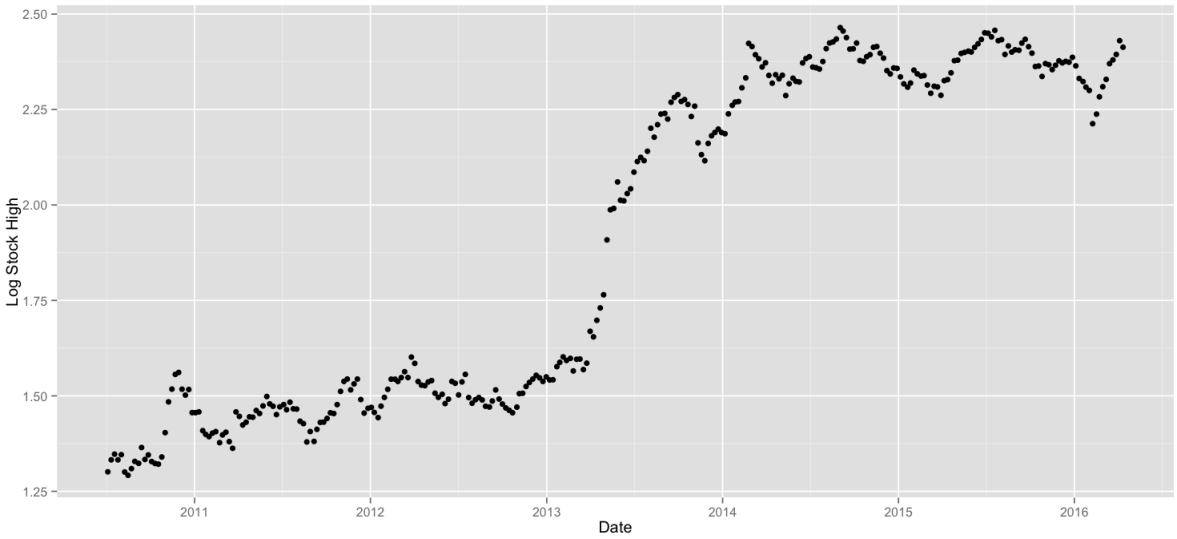
Let's look at a histogram of Tesla's stock price `High`:



We note that since our `High` variable is very right-tailed and represents money, it's natural to take the target as a $\log_{10}$ high. We add this variable into our dataset as `log.High` and note that we are now fitting a semilog model:

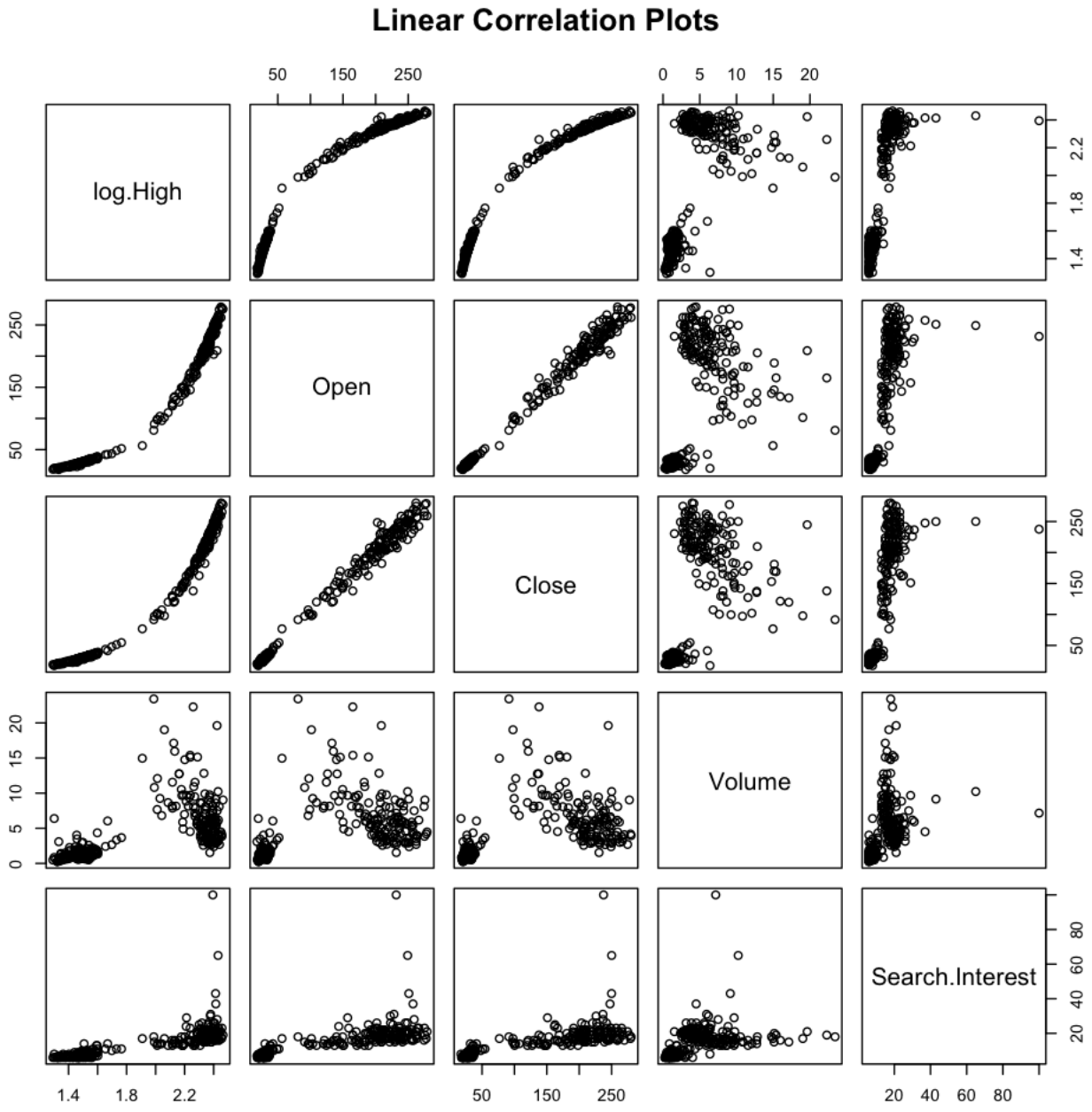
$$\begin{aligned}\texttt{log.High} = \beta_0 + \beta_1 \times \texttt{Open} + \beta_2 \times \texttt{Close} \\ + \beta_3 \times \texttt{Volume} + \beta_4 \times \texttt{Search.Interest}\end{aligned}$$

Now let's look at a plot of Tesla's logged stock price `log.High` over time:



We note the dramatic interval towards the end of 2013 Q1 where Tesla's stock skyrockets. This is explained mostly by Tesla's defying expectations for a Q1 loss and instead reporting a $15 million profit along with investors warming up to Elon Musk's capabilities as "the next Steve Jobs".

Before looking at the results of the linear regression, let's plot each predictor against one another in a scatterplot matrix:



We can already see a few ominous signs. In particular:

- `log.High` against `Open` is almost exactly a logarithmic model, implying that there is likely collinearity between the two. Likewise for `log.High` and `Close`.
- Plotting `Open` against `Close` tells us that they are much too similar and screams for us to notice that the two predictors are collinear and will influence our one another's variance inflation factor.
- `Volume` doesn't seem to be a particularly effective predictor, however a more formal analysis is required before deeming it ineffective.

Finally, let's look at the result of a linear regression:

```
> summary(tesla_regression)

Residuals:
               +---------+---------+------------+----------+----------+
               |   Min   |    1Q   |   Median   |    3Q    |   Max    |
               +---------+---------+------------+----------+----------+
               |-0.225893|-0.043901|  0.001092  | 0.044974 | 0.155031 |
               +---------+---------+------------+----------+----------+

Coefficients:
                       +-----------+------------+---------+----------+
                       |  Estimate | Std. Error | t value | Pr(>|t|) |
      +-----------------+-----------+------------+---------+----------+
      | (Intercept)     | 1.3436700 | 0.0070169  | 191.491 |  < 2e-16 | ***
      | Open            | 0.0014843 | 0.0004130  |   3.594 | 0.000381 | ***
      | Close           | 0.0026501 | 0.0004133  |   6.412 | 5.65e-10 | ***
      | Volume          | 0.0165199 | 0.0010832  |  15.251 |  < 2e-16 | ***
      | Search.Interest | 0.0002417 | 0.0006144  |   0.393 | 0.694347 |
      +-----------------+-----------+------------+---------+----------+
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
---

Residual standard error: 0.06389 on 297 degrees of freedom
Multiple R-squared: 97.83%,    Adjusted R-squared: 97.80%
F-statistic: 3346 on 4 and 297 DF, p-value: < 2.2e-16
-------------------------------------------------------------------------------
> vif(tesla_regression)
```

```
VIF:
           Open         Close      Volume   Search.Interest
     110.309823   110.725425   1.380269          2.119782
--------------------------------------------------------------------------------
Regression Equation

log.High = 1.3436700 + 0.0014843 Open + 0.0026501 Close
                     + 0.0165199 Volume + 0.0002417 Search.Interest
```

For our naïve model, we've done well for ourselves – or so it seems! The coefficients can be interpreted as follows:

1. A \$1 increase in Tesla's weekly stock `Open` value is associated with increasing the weekly `High` (not `log.High`) by .3% ($10^{.0014843} = 1.003$), holding all else fixed.
2. A \$1 increase in Tesla's weekly stock `Close` value seems to be associated with increasing the weekly `High` by .6% ($10^{.0026501} = 1.006$), holding all else fixed.
3. A 1 million frequency increase in Tesla's weekly `Volume` value is associated with increasing the weekly `High` by 3.8% ($10^{0.0165199} = 1.038771199$), holding all other variables fixed.
4. Lastly, a 1% increase in the number of Google searches for "tesla" relative to the all-time max number of Google searches for "tesla" is associated with increasing the weekly `High` by essentially .1% ($10^{0.0002417} = 1.001$), holding all else fixed.

With the exception of `Search.Interest`, we see that our predictors' p-values imply high statistical significance at the .01 level.

The approximate 95% prediction interval of $\pm 2s$ for `log.High` corresponds to a multiplicative interval in the original (`High`) scale, since adding $2s$ to `log.High` is the same as multiplying `High` by 10^{2s}, while subtracting $2s$ from `log.High` is equivalent to multiplying `High` by 10^{-2s}.

Our R^2 is a very strong 97.83%, however we note that our `Open` and `Close` predictors imply very, *very* strong correlation between our two predictors and our response variable, `log.High`. Calculating our pass-able VIF cutoff:

$$VIF < max(10, \frac{1}{1 - R^2_{model}}) \rightarrow max(10, \frac{1}{1 - 0.9783}) \rightarrow VIG < max(10, 46.083)$$

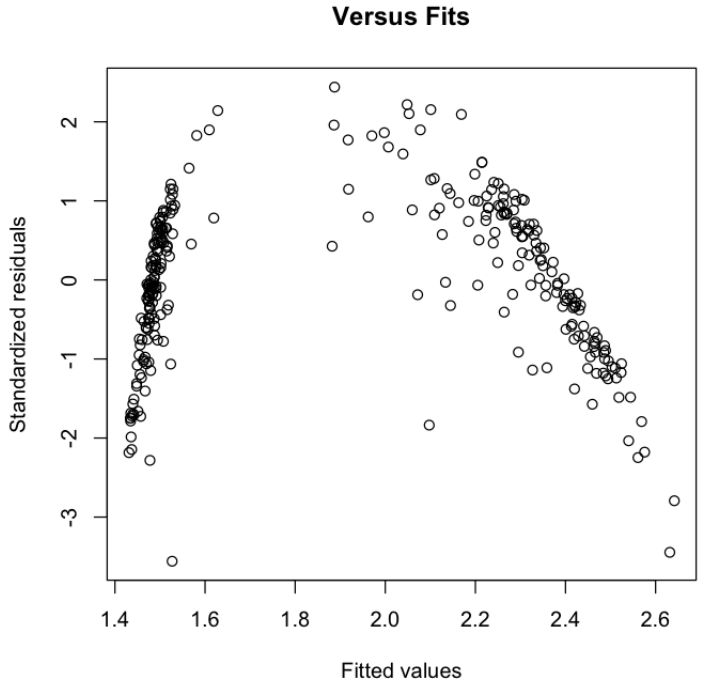
we see that, indeed, `Open` and `Close`'s VIF scores are much too high and we need to consider a best subset model. We also could have seen this correlation through the following output:

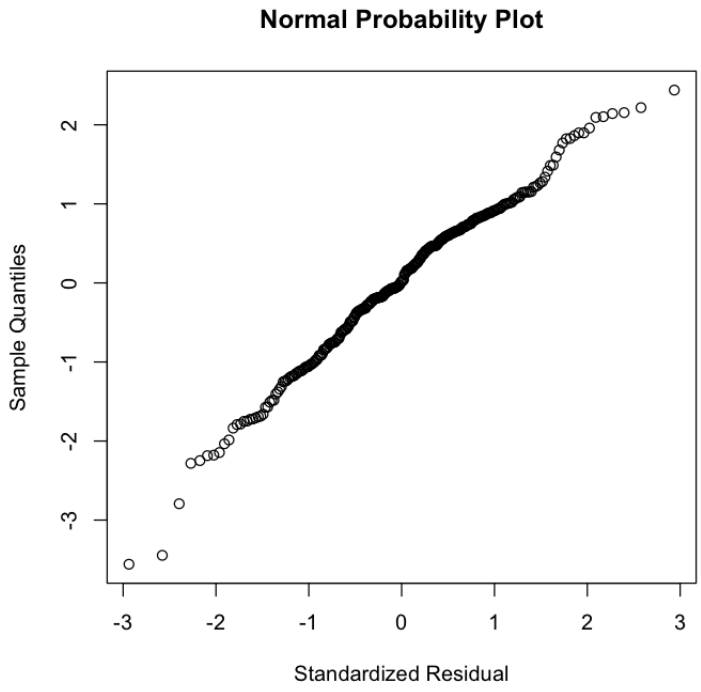
```
> cor(cbind(log.High, Open, Close, Volume, Search.Interest))
```

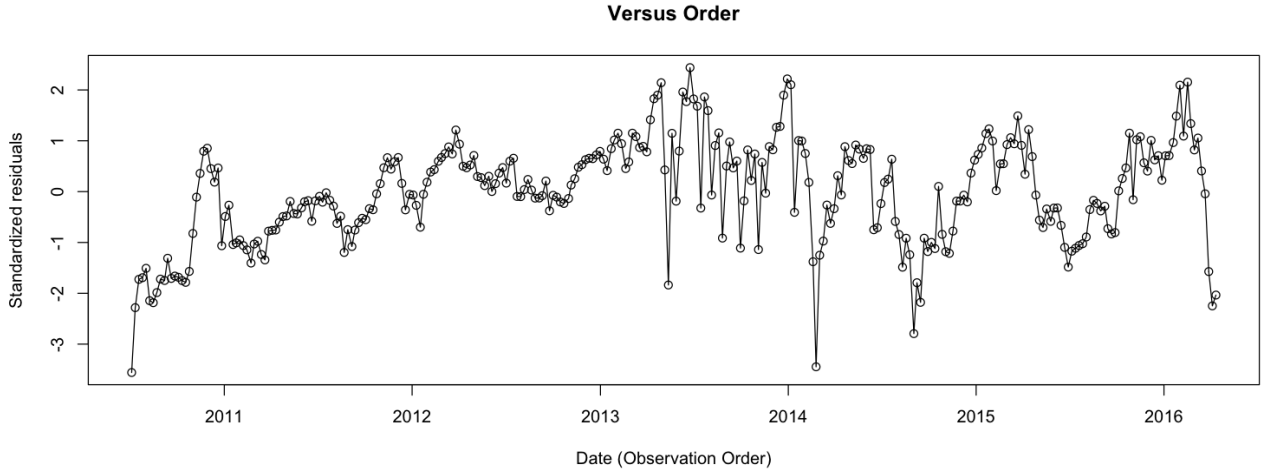
```
Correlations
                log.High  Open      Close     Volume    Search.Interest
log.High        1.0000000
Open            0.9781396 1.0000000
Close           0.9797301 0.9954470 1.0000000
Volume          0.6131227 0.5069942 0.5098143 1.0000000
Search.Interest 0.7211174 0.7188305 0.7191252 0.4530636 1.0000000
```

Note the correlations that are not 1.00 (hence, variables that are truly being compared to *other* variables) come very close: 0.978 and 0.995 – and for our two suspects, `Open` and `Close`.

Before choosing best subsets, let's consider the following residual plots for `log.High`:







Looking at the residual plots, we see:

- An odd relationship between the fitted values and standardized residuals. We see a surprising dichotomy where two natural subgroups seem to present themselves. This is most likely being driven by our earlier observation of Tesla's stock remaining low from mid-2010 to 2013 then suddenly skyrocketing to multiple times its value.
- Our normal plot trails off on both tails, supporting a fat-tailed distribution. Again, this is expected due to the distribution of our stock prices and the April 2013 stock price skyrocket.

- The time-ordered residuals don't seem informative as to whether or not our dataset contains autocorrelation.

We note that we could have also looked at the results of the following command in order to segue into our z-statistic:

```
> cbind(stdres, hatvalues(tesla_regression), cooks.distance(tesla_regression))
```

```
          stdres
1   -3.558521039 0.012865780 3.300874e-02
2   -2.282352578 0.007547758 7.923258e-03
3   -1.725839517 0.007029019 4.216858e-03
4   -1.693163428 0.007703989 4.451457e-03
5   -1.508806459 0.007518644 3.449167e-03
6   -2.145340970 0.007676977 7.121308e-03
7   -2.184891999 0.007820304 7.525290e-03
8   -1.986179653 0.007679355 6.105761e-03
9   -1.721042354 0.007795964 4.654595e-03
10  -1.748257089 0.007903111 4.869503e-03
11  -1.310860201 0.007336141 2.539851e-03
12  -1.707444690 0.007530167 4.423954e-03
13  -1.661121211 0.007224351 4.015877e-03
14  -1.684821884 0.007868225 4.502414e-03
15  -1.750356370 0.007938387 4.903166e-03
16  -1.784998015 0.007934486 5.096640e-03
17  -1.569586169 0.007801285 3.874073e-03
18  -0.823101529 0.007461891 1.018682e-03
19  -0.108459143 0.007558553 1.791827e-05
20   0.358410191 0.006532511 1.689340e-04
..       ...         ...         ...
295  1.337905707 0.021646782 7.920976e-03
296  0.819977411 0.009297749 1.262026e-03
297  1.053972123 0.011448098 2.572895e-03
298  0.405899770 0.025876550 8.753062e-04
299 -0.042882222 0.012705187 4.732807e-06
300 -1.572722607 0.577535717 6.762746e-01
301 -2.247564033 0.173109173 2.115076e-01
302 -2.034841424 0.047812487 4.158244e-02
```

Finally, we consider the results of the Durbin-Watson statistic:

```
> dwtest(log.High ~ Open + Close + Volume + Search.Interest)

Durbin-Watson test:

DW = 0.4098, p-value < 2.2e-16
alternative hypothesis: true autocorrelation is greater than 0
```

What does this mean for us and our model? Well, given our sample size of 302 we can appeal to the approximate formula for the z-statistic:

$$z = (\frac{DW}{2} - 1)\sqrt{n} = (\frac{0.4098}{2} - 1)\sqrt{302} = -0.7951 \times 17.3781472 = -13.817364839$$

A z-statistic of -13.82 implies that the Durbin-Watson statistic is more than 13.5 standard deviations away from its mean and is normally distributed – not likely at all. We would strongly reject the null hypothesis of no autocorrelation. However, we must reconsider our model before determining the presence of autocorrelation.

Given that our two predictors `Open` and `Close` had VIF scores that gave off strong impressions of multicollinearity, we opt to choose the best subset of our model that potentially omits either or both predictors in favor of less collinearity. Turning to best subsets, we see:

```
> leaps(cbind(Open, Close, Volume, Search.Interest), log.High, nbest = 2)
> leaps(cbind(Open, Close, Volume, Search.Interest), log.High, nbest = 2, method = "adjr2")
> leaps(cbind(Open, Close, Volume, Search.Interest), log.High, nbest = 2, method = "r2")

Note:
Open            -> [1]
Close           -> [2]
Volume          -> [3]
Search.Interest -> [4]

Vars    R^2      R^2 (adj)    Mallows_Cp     S        [1]   [2]   [3]   [4]
1    0.9598710   0.9597372    250.933281  0.08643           X
1    0.9567572   0.9566130    293.528198  0.08972     X
2    0.9773210   0.9771693     14.230388  0.06508           X     X
2    0.9752489   0.9750833     42.575776  0.06799     X           X
3    0.9782769   0.9780582      3.154721   0.0638     X     X     X        *
3    0.9773440   0.9771159     15.916191  0.06516           X     X     X
4    0.9782882   0.9779958      5.000000  0.06389     X     X     X     X  *
```

We note a few things of importance here. First, we opt to skip our potential subset of three variables marked with an asterisk ("*") because we would be making the same mistake – we need a model that includes either `Open` or `Closed`, and not both because otherwise we'll see problems with collinearity. Therefore, our pool of subset models is filtered down to the following:

Vars	R^2	R^2 (adj)	Mallows_Cp	S	[1]	[2]	[3]	[4]
1	0.9598710	0.9597372	250.933281	0.08643		X		
1	0.9567572	0.9566130	293.528198	0.08972	X			
2	0.9773210	0.9771693	14.230388	0.06508		X	X	
2	0.9752489	0.9750833	42.575776	0.06799	X		X	
3	0.9773440	0.9771159	15.916191	0.06516		X	X	X

We proceed with the subset model tests:

- We look to see at around what p does our R^2 begins to level off. We note nominal increases in R^2 from p = 1 to p = 2 and p = 2 to p = 3. We tend towards a smaller p, and so our first test nominates a model with **p = 2**.
- We look to see at around what p does our R^2_adj maximize. This one isn't too difficult – our highest adjusted R^2 is at p = 2 (only nominally better than our p = 3 model), and so we again nominate a model with **p = 2**.
- We look for a model whose C_p criterion best satisfies $C_p \leq p + 1$ and/or minimize(C_p). We note that none of the models satisfy our prior ideal C_p, however we do minimize C_p at $p = 2$, and so we yet again confirm a subset model where **p = 2**.
- We look to Akaike's information criterion and corrected criterion to determine which model maximizes AIC and which minimizes the corrected AIC. We note the AIC and corrected AIC values below:

Vars	AIC	AIC_c
1	-617.8418	-617.7613
1	-595.2726	-595.1921
2	-788.1816	-788.0469
2	-761.7769	-761.6423
3	-786.4876	-786.2849

  We see that our minimized AIC and AIC corrected is very clearly at the $p = 1$ level, electing **p = 1** as our AIC-favored subset model.

All that said, we note that a majority of our tests elect our $p = 2$ model where `log.High` is predicted with predictors `Close` and `Volume`. While AIC and AICc do nominate a $p = 1$ model, we

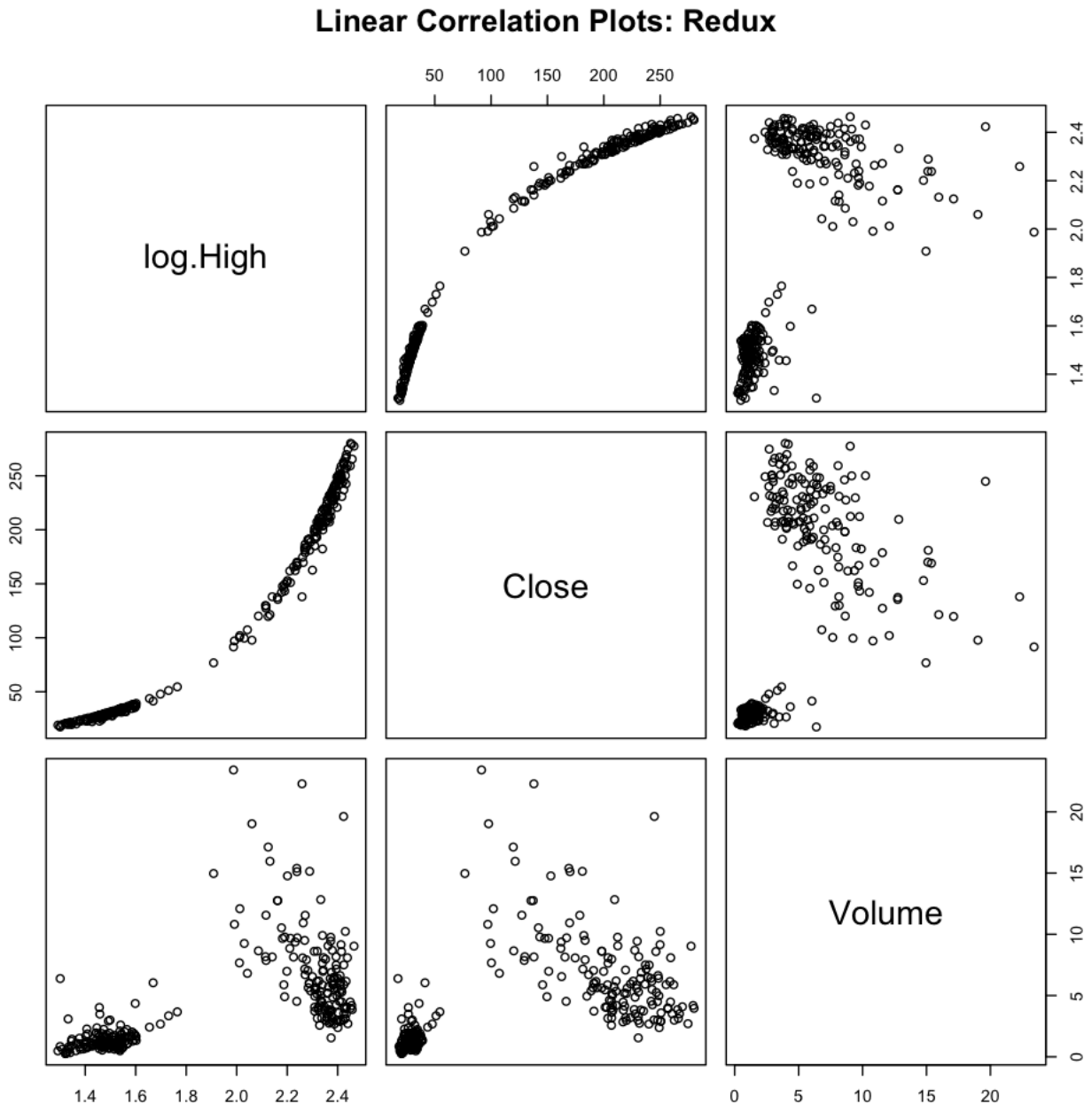
choose to move forward with $p = 2$ because it places no additional taxations on our assumptions (we will always have weekly closing stock value and weekly stock volume) and will more likely than not produce a more fitting model. Our model becomes:

$$\texttt{log.High} = \beta_0 + \beta_1 \times \texttt{Close} + \beta_2 \times \texttt{Volume}$$

We treat our model like new and begin from square one.

Exploratory Data Analysis: Redux

There's no need to look at summary data of our raw values since our earlier, more complex model did so. We move to plot our target variable and predictors against one another in a scatterplot matrix:



While we see that `Close` and our target `log.High` still share a strong logarithmic relationship, we also see much less immediate evidence for collinearity among our predictors.

Let's look at the results of our linear regression:

```
> summary(redux_regression)

Residuals:
   +-----------+-----------+----------+----------+----------+
```

```
   |    Min    |     1Q    |  Median  |    3Q    |    Max   |
   +-----------+-----------+----------+----------+----------+
   | -0.260668 | -0.042893 | 0.003875 | 0.045702 | 0.173021 |
   +-----------+-----------+----------+----------+----------+

Coefficients:
              +-----------+------------+---------+----------+
              | Estimate  | Std. Error | t value | Pr(>|t|) |
+-------------+-----------+------------+---------+----------+
| (Intercept) | 1.345e+00 | 6.310e-03  | 213.20  | < 2e-16  | ***
| Close       | 4.141e-03 | 4.651e-05  |  89.04  | < 2e-16  | ***
| Volume      | 1.656e-02 | 1.092e-03  |  15.17  | < 2e-16  | ***
+-------------+-----------+------------+--------+-----------+
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
---

Residual standard error: 0.06508 on 299 degrees of freedom
Multiple R-squared: 97.73%,    Adjusted R-squared: 97.72%
F-statistic:  6443 on 2 and 299 DF,  p-value: < 2.2e-16
-------------------------------------------------------------------------------
> vif(redux_regression)

VIF:
         Close     Volume
      1.351188   1.351188
-------------------------------------------------------------------------------
Regression Equation

log.High = 1.345 + .004141 Close + .01656 Volume
```

Our model is looking much better! Interpreting the predictors, we note:

- A \$1 increase in Tesla's weekly stock `Close` is associated with increasing the weekly `High` by .9% ($10^{.0041}$ = 1.009485302), holding all else fixed.
- A 1 million frequency increase in Tesla's weekly `Volume` predictor is associated with increasing the weekly `High` by 46% ($10^{0.1656}$ = 1.464198643), holding all other variables fixed.

We note that both predictors' p-values imply high statistical significance by any measure.

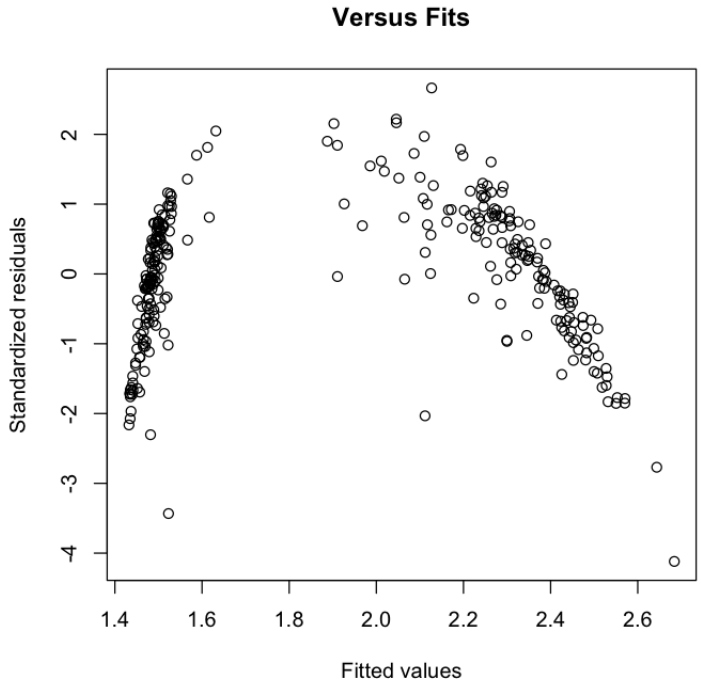
The approximate 95% prediction interval of $\pm 2s$ for `log.High` corresponds to a multiplicative interval in the original (`High`) scale, since adding $2s$ to `log.High` is the same as multiplying `High` by 10^{2s}, while subtracting $2s$ from `log.High` is equivalent to multiplying `High` by 10^{-2s}. We note that our simplified model's residual standard error is nominally higher than our previous model's standard error, but we aren't concerned about their comparative values.

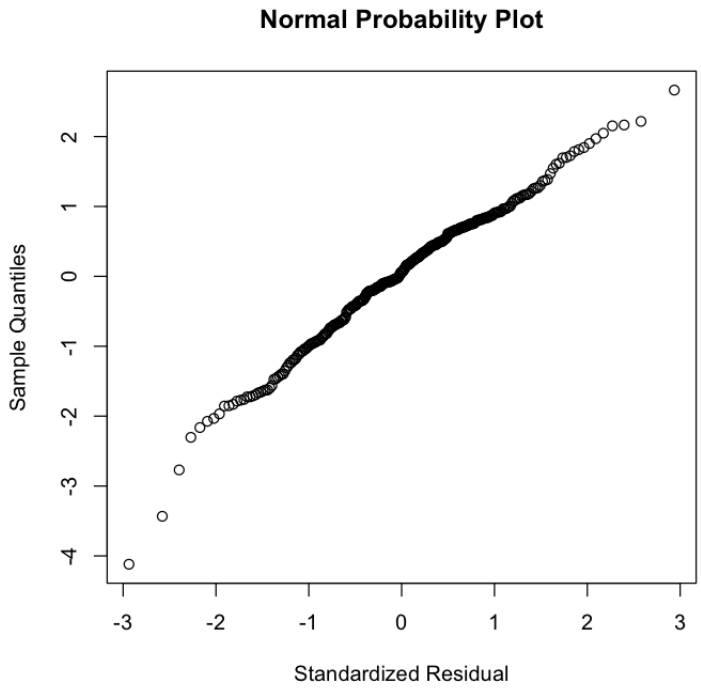
With an R^2 of 97.73% and low VIF scores among our predictors, we can safely say that this model does not contain collinear predictors and that it accounts for much of the data within our dataset. Just to be sure, we calculate our pass-able VIF cutoff:

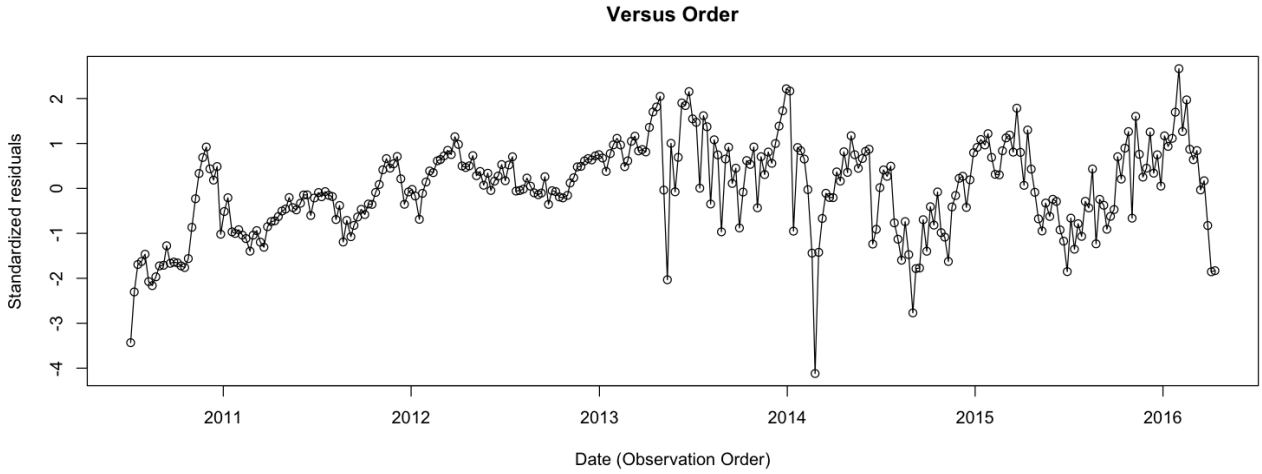
$$VIF < max(10, \frac{1}{1 - R^2_{model}}) \rightarrow max(10, \frac{1}{1 - 0.9773}) \rightarrow VIG < max(10, 44.05)$$

and note that our predictors' VIFs are clear.

Before going further, let's look at the residual plots for our simplified model's `log.High`:







Looking at the residual plots:

- We still see an odd relationship between the fitted values and standardized residuals. The two subgroups within our data are still very pronounced, and we continue with the understanding that it is due to Tesla Motors' stock price skyrocketing towards the end of 2013 Q1.
- Our normal probability plot is much more stable, however we do note it still contains a pronounced right tail.

- Our time-ordered residual plot has somewhat shifted most plotted points towards a standardized residual of 0. We note the persistent, dramatic downturn in Q1 2014 when Tesla underwent another, smaller increase in stock value resulting in our model's less-than-ideal fit on that particular point.

We note that we also could have looked at the results of the following binding of our simplified model's standard residuals, hat values, and Cook's distance for each observation within our dataset:

```
    redux_stdres
1    -3.43284382 0.012574856 5.002483e-02
2    -2.30352878 0.007234692 1.288960e-02
3    -1.69322516 0.007028715 6.764682e-03
4    -1.62654805 0.007604546 6.757733e-03
5    -1.46430364 0.007424627 5.346286e-03
6    -2.07362376 0.007526497 1.086958e-02
7    -2.16347371 0.007761520 1.220430e-02
8    -1.96761659 0.007614056 9.901367e-03
9    -1.72427692 0.007664786 7.654810e-03
10   -1.70989779 0.007851135 7.712135e-03
11   -1.27591493 0.007242402 3.958782e-03
12   -1.66710634 0.007456154 6.959380e-03
13   -1.63958818 0.007138114 6.442330e-03
14   -1.65773952 0.007821698 7.221421e-03
15   -1.72433279 0.007893117 7.885169e-03
16   -1.76064599 0.007886650 8.213989e-03
17   -1.56163799 0.007759303 6.356897e-03
18   -0.86589311 0.007197374 1.811834e-03
19   -0.23049066 0.006357654 1.133058e-04
20    0.33249252 0.006476404 2.402140e-04
..       ....        ....        ....
295   0.87100880 0.005251453 1.335027e-03
296   0.63947745 0.005951206 8.160682e-04
297   0.84131604 0.007326949 1.741462e-03
298  -0.03412873 0.010540422 4.135984e-06
299   0.17030912 0.008629663 8.416130e-05
300  -0.82540912 0.008683129 1.989212e-03
301  -1.85486951 0.012639984 1.468170e-02
302  -1.83086395 0.011036940 1.246980e-02
```

Finally, we consider the results of the Durbin-Watson statistic:

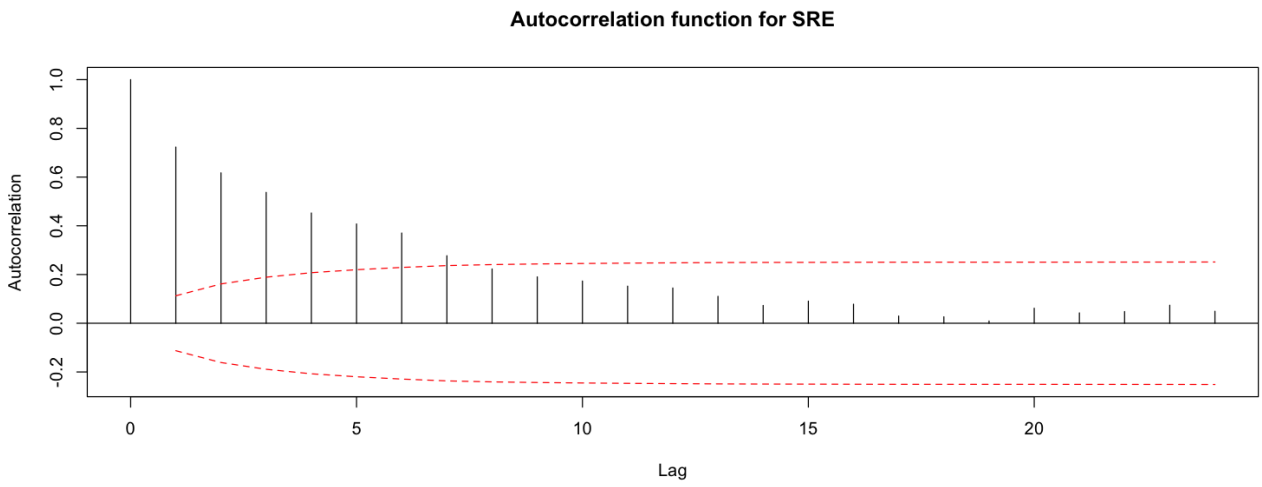
```
> dwtest(log.High ~ Close + Volume)

Durbin-Watson test

DW = 0.4977, p-value < 2.2e-16
alternative hypothesis: true autocorrelation is greater than 0
```

Again, what does this mean for us and our model? Our sample size of n = 302 means we can appeal to the approximate formula for the z-statistic:

$$z = (\frac{DW}{2} - 1)\sqrt{n} = (\frac{0.4977}{2} - 1)\sqrt{302} = -0.75115 \times 17.3781472 \approx -13.05$$

A z-statistic of -13.05 implies that the Durbin-Watson statistic is more than 13 standard deviations away from its mean and is normally distributed – not likely at all. We would strongly reject the null hypothesis of no autocorrelation.

Let's look at the ACF plot for our data's standardized residuals:



The vertical bars are giving us the sample autocorrelations, and given that they are above the line we know that there is evidence for autocorrelation. The red lines correspond to the values that would be statistically significant at a .05 level against the null hypothesis of a particular autocorrelation equal to 0 or not equal to 0. The plot also tells us that the autocorrelations are going down faster-than-geometrically at first, then slow down to a geometric decay.

Let's see how a non-parametric Runs test evaluates the autocorrelation of our dataset:

```
> runs.test(redux_stdres)
```

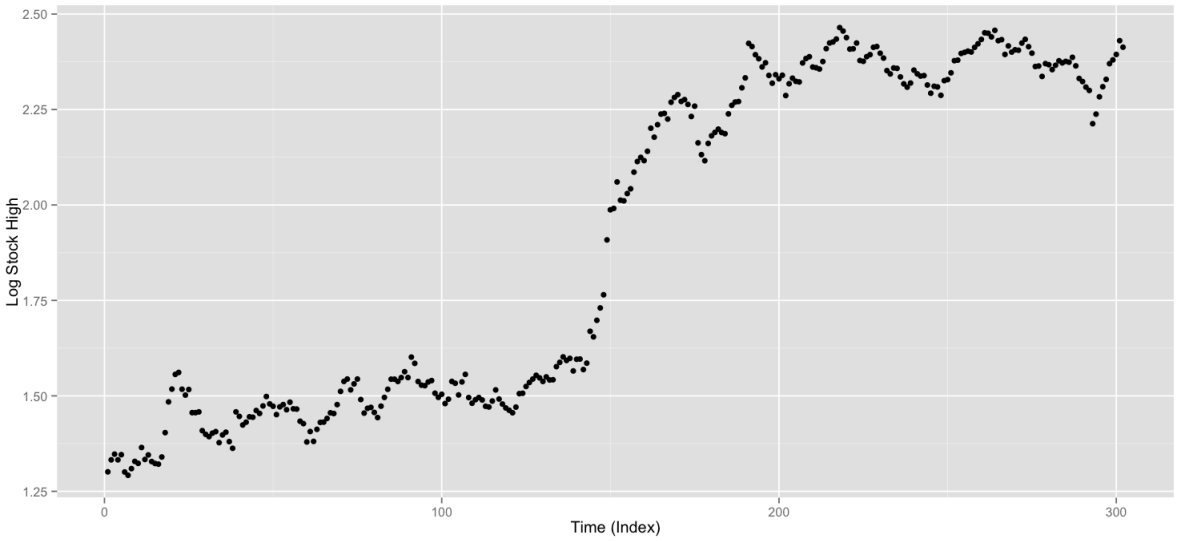
```
Runs Test - Two sided

Standardized Runs Statistic = -11.8737
p-value < 2.2e-16
```

Given the tiny reported p-value and previous understanding of our dataset, we determine that the Runs test is also supporting our overwhelming evidence of autocorrelation.

We next explore the four big ideas that we should be considering for autocorrelative data: detrending, deseasonalizing, lagging, and differencing.

Detrending

In order to account for trends, we look at a time series plot of our `log.High`:

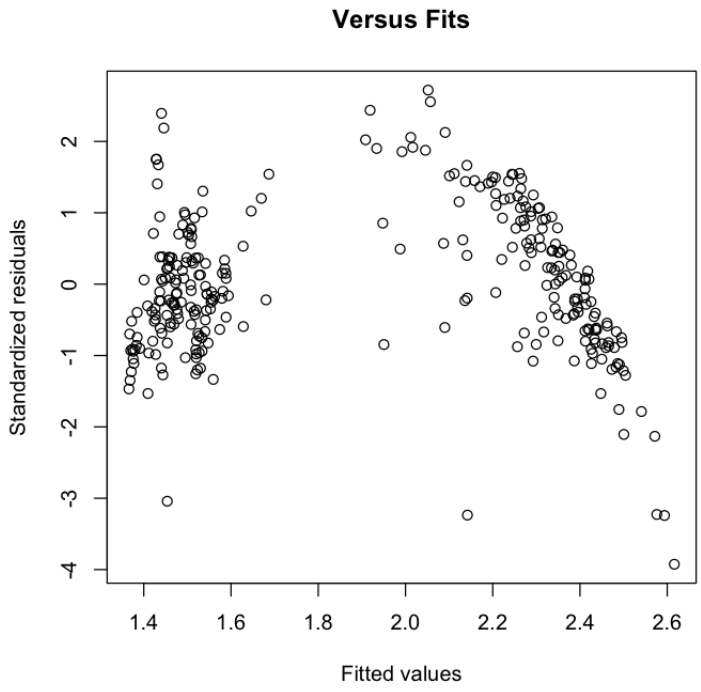


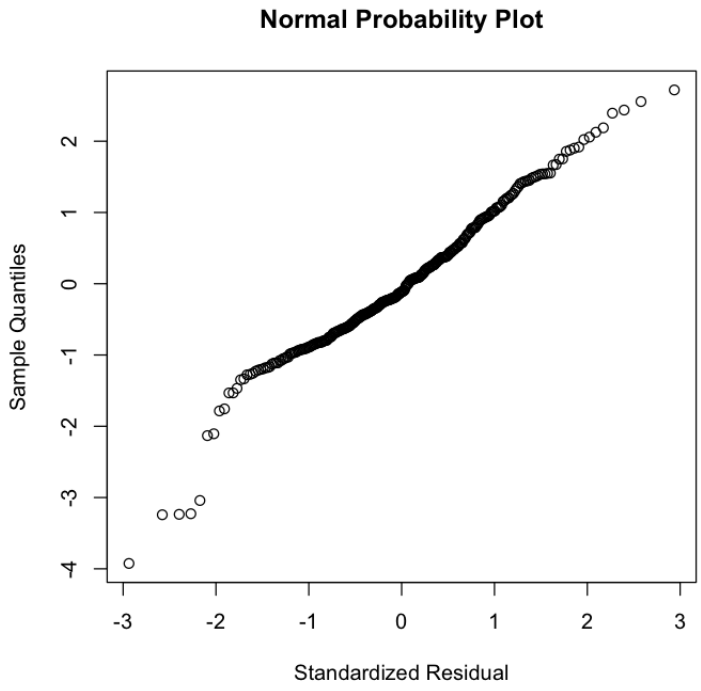
and ultimately decide to include `Time` as a predictor within our model. Further support for including `Time` as a predictor for Tesla's stock price is their critical growth as a tech company over the period from their initial public offering to the present. We redo our regression:

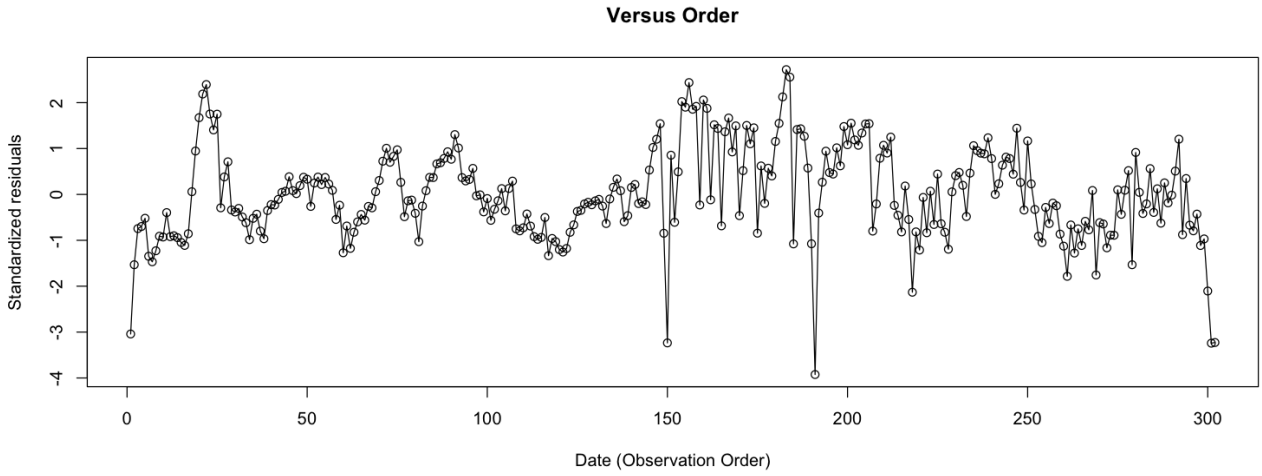
```
> summary(with_time)

Residuals:
  +-----------+-----------+-----------+----------+----------+
  |    Min    |    1Q     |  Median   |    3Q    |   Max    |
  +-----------+-----------+-----------+----------+----------+
  | -0.193012 | -0.033432 | -0.005684 | 0.031421 | 0.137932 |
```

```
  +-----------+-----------+-----------+----------+----------+

Coefficients:
              +-----------+------------+---------+----------+
              | Estimate  | Std. Error | t value | Pr(>|t|) |
+-------------+-----------+------------+---------+----------+
| (Intercept) | 1.289e+00 | 6.392e-03  | 201.63  |  <2e-16  | ***
| Close       | 3.222e-03 | 7.556e-05  |  42.65  |  <2e-16  | ***
| Volume      | 1.691e-02 | 8.529e-04  |  19.82  |  <2e-16  | ***
| Time        | 1.084e-03 | 7.813e-05  |  13.87  |  <2e-16  | ***
+-------------+-----------+------------+---------+----------+
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
---

Residual standard error: 0.05082 on 298 degrees of freedom
Multiple R-squared: 98.62%,    Adjusted R-squared: 98.61%
F-statistic:  7107 on 3 and 298 DF,  p-value: < 2.2e-16
-------------------------------------------------------------------------------
> vif(with_time)

VIF:
       Close     Volume       Time
    5.848716   1.352352   5.424926
-------------------------------------------------------------------------------
Regression Equation

log.High = 1.289 + .003222 Close + .01691 Volume + .001084 Time
```

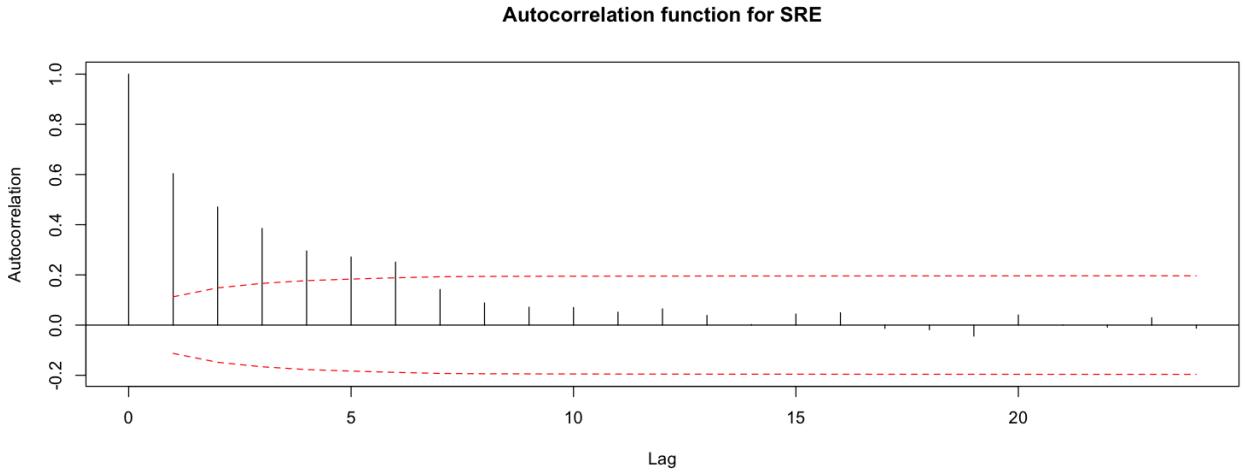
We see that each of our predictors are still very statistically significant, including our new `Time` variable. Although we see a slight increase in the VIF values for `Close` and `Time`, but nothing to worry about given the high VIF cap for our R^2. We look to the residual plots:


Versus Fits
Standardized residuals
Fitted values
1.4
1.6
1.8
2.0
2.2
2.4
2.6
-4
-3
-2
-1
0
1
2


Normal Probability Plot
Sample Quantiles
Standardized Residual
-3
-2
-1
0
1
2
3
-4
-3
-2
-1
0
1
2



Looking to the ACF plot with `Time` included:



And the Durbin-Watson statistic reports:

```
> dwtest(log.High ~ Close + Volume + Time)

Durbin-Watson test

DW = 0.7192, p-value < 2.2e-16
alternative hypothesis: true autocorrelation is greater than 0
```

we note that while including `Time` in our model was meaningful and brought our Durbin-Watson statistic closer to a safe range, as a predictor it does not "fix" or address the autocorrelation. That said, we are right to include a `Time` variable since we have years-worth of data over time.

We move onto considering deseasonalization in our model to address the autocorrelation.

Deseasonalizing

We note that, from the numerous plots we have constructed, seasonality is insignificant – there are no discernible seasons for the stock market (especially given its global presence). Furthermore, we do not have enough data to warrant $k - 1$ indicator variables for every week in our dataset. We note that if there *were* seasonal effects in the stock market, we would have much larger economic and financial problems than not having enough data.

We move onto considering lagging in our model to address the autocorrelation.

Lagging

We assert that utilizing a lagged response variable would prove immensely useful in predicting future response variable values. Given the nature of stock prices, we can confidently say that We include `lag.log.High` in our model, and look to the regression output to see the effects therein:

```
> summary(with_lag)

Residuals:
    +-----------+-----------+----------+----------+----------+
    |    Min    |     1Q    |  Median  |    3Q    |   Max    |
    +-----------+-----------+----------+----------+----------+
    | -0.083907 | -0.013289 | 0.000255 | 0.014780 | 0.080197 |
    +-----------+-----------+----------+----------+----------+

Coefficients:
               +-----------+------------+---------+----------+
               | Estimate  | Std. Error | t value | Pr(>|t|) |
+--------------+-----------+------------+---------+----------+
| (Intercept)  | 3.073e-01 | 2.937e-02  |  10.463 | < 2e-16  | ***
| Close        | 7.489e-04 | 8.125e-05  |   9.217 | < 2e-16  | ***
| Volume       | 5.484e-03 | 5.164e-04  |  10.619 | < 2e-16  | ***
| Time         | 2.319e-04 | 4.303e-05  |   5.389 | 1.45e-07 | ***
| lag.log.High | 7.632e-01 | 2.268e-02  |  33.653 | < 2e-16  | ***
+--------------+-----------+------------+---------+----------+
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
---

Residual standard error: 0.02285 on 296 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared: 99.72%,Adjusted R-squared: 99.72%
F-statistic: 2.649e+04 on 4 and 296 DF,  p-value: < 2.2e-16
-------------------------------------------------------------------------------
> vif(with_lag)

VIF:
       Close     Volume       Time   lag.log.High
   33.325254   2.450811   8.059575      54.759126

-------------------------------------------------------------------------------
Regression Equation

log.High = .3073 + 0.0007 Close + 0.0055 Volume + 0.0002 Time + 0.7632 lag.log.High
```

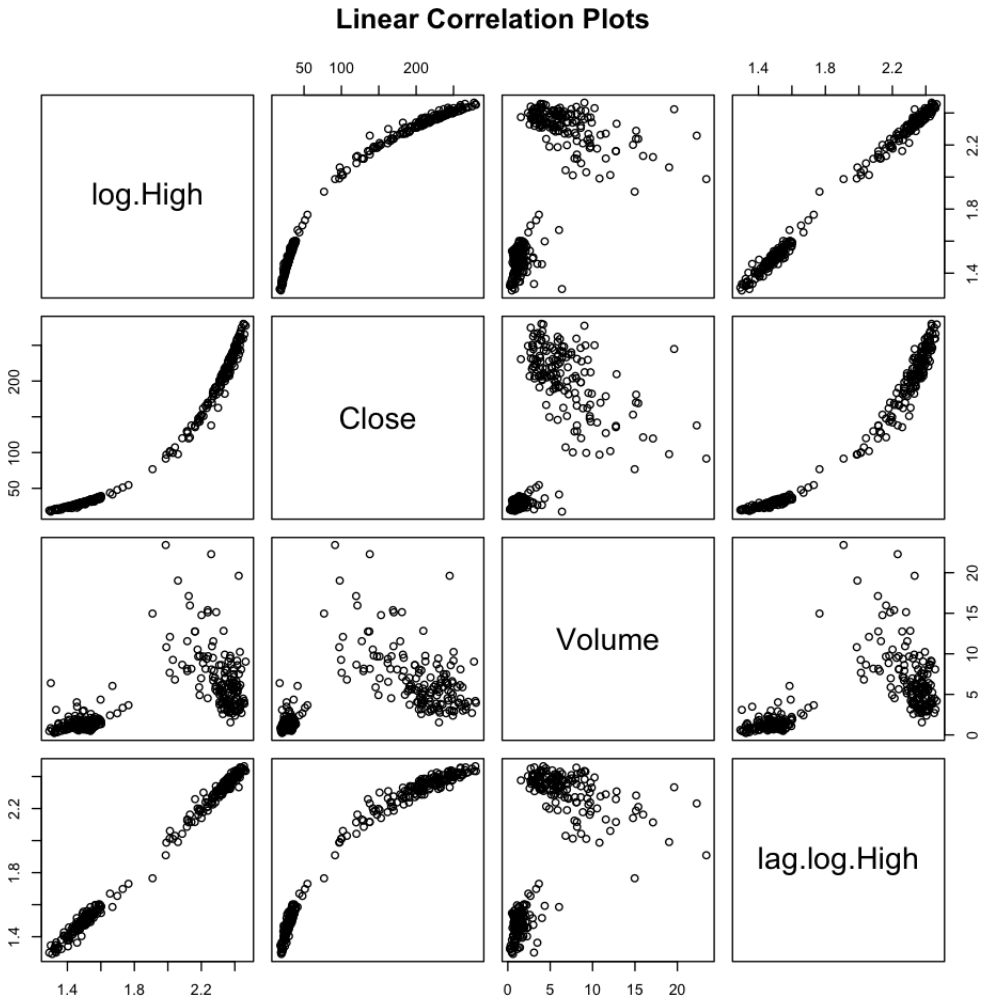
Interpreting our regression equation, we say:

- A \$1 increase in Tesla's weekly stock `Close` value is associated with increasing the weekly `High` by .1% ($10^{0.0007} = 1.001613109$), holding all other variables fixed.
- A 1 million frequency increase in Tesla's weekly `Volume` predictor is associated with increasing the weekly `High` by 1.2% ($10^{0.0055} = 1.012744749$), holding all else fixed.
- A one week increase in team is associated with increasing Tesla's stock price `High` by essentially 0% ($10^{0.0002} = 1.000460623$), all else fixed. This makes sense because if nobody touched the stock for a week, it would be strange to see any changes.
- Lastly, we know that an additive +1 increase in last week's `log.High` results in a 79.7%($10^{0.7632} = 5.796955947$) increase in this week's stock price.

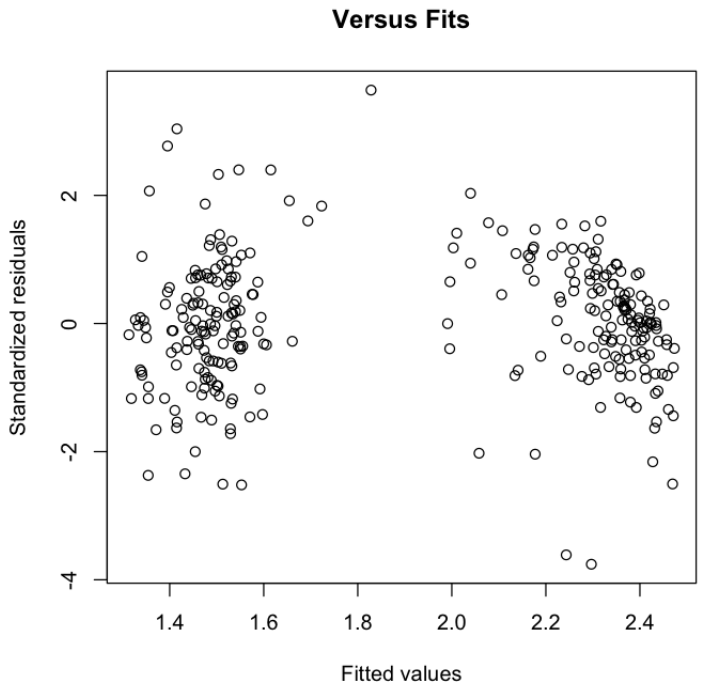
We note a few things about our regression output:

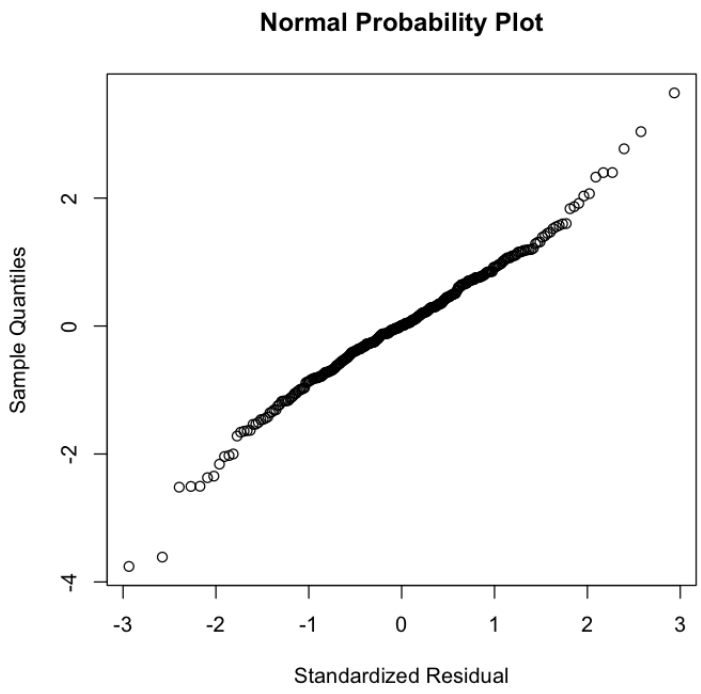
- All of our predictors are incredibly statistically significant.
- Our R^2 of 99.72% gives us a very tight fit on the sample data, most likely due to the dominating effect that `lag.log.High` has on our prediction task.
- The approximate 95% prediction interval of $\pm 2s$ for `log.High` corresponds to a multiplicative interval in the original (`High`) scale, since adding $2s$ to `log.High` is the same as multiplying `High` by 10^{2s}, while subtracting $2s$ from `log.High` is equivalent to multiplying `High` by 10^{-2s}.

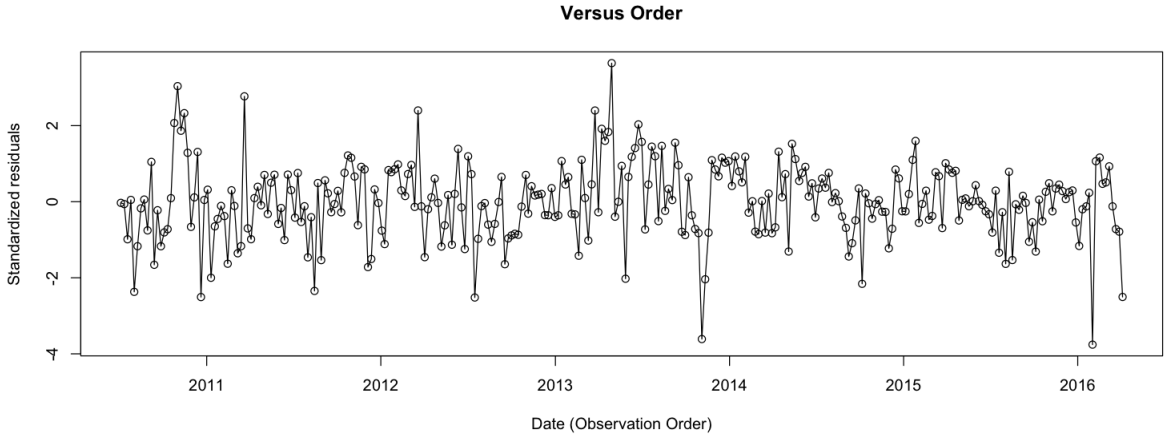
- Lastly, we see that the VIF values have increased and that `lag.log.High` seems to exhibit collinearity. We expect it to be very collinear with `Close`, given the range of a stock's price per week. Looking at our collinearity scatterplot matrix confirms this suspicion:



Looking to our residual plots:


Versus Fits
Standardized residuals
Fitted values
-4
-2
0
2
1.4
1.6
1.8
2.0
2.2
2.4

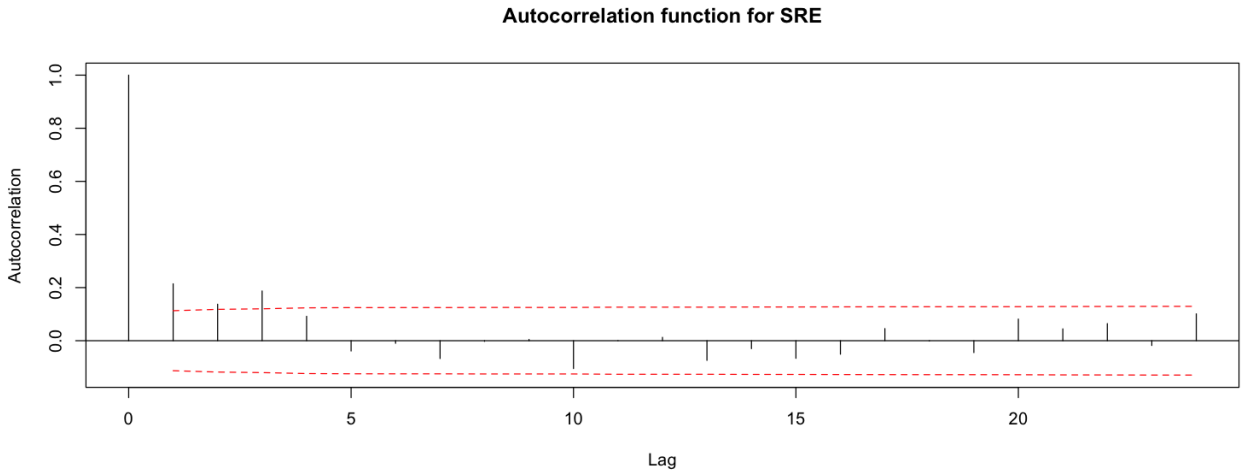

Normal Probability Plot
Sample Quantiles
Standardized Residual
-4
-2
0
2
-3
-2
-1
0
1
2
3



Our residual plots look better:

- Our normal probability plot is now slightly left tailed but more normally distributed than before
- Our versus fits – while still effected by the Q1 2013 stock price increase – are tending closer towards a standardized residual of 0.

Looking to the ACF plot with `lag.log.High` included:



As we can see, aside from the initial autocorrelation, our ACF plot is much better-behaved and suggests that including the lagged, logged `High` as a predictor was an excellent choice given our goal was prediction. We cannot utilize the Durbin-Watson statistic due to including a lagged response variable, however through our Runs test we can see that our p-value has slightly increase, although still claiming evidence for autocorrelation.

```
Runs Test - Two sided

Standardized Runs Statistic = -2.5979,
p-value = 0.009379
```

By including last week's response variable within our predictor set, we note its inherent collinearity, domination of our model's structure, and

Differencing

While differencing would give us the ability to answer questions related to the expected returns on Tesla Motors' stock price, we are more concerned with prediction of next week's stock price.

We move on from differencing and arrive at our final model:

$$\begin{aligned}\texttt{log.High} = .3073 + 0.0007\ \texttt{Close} + 0.0055\ \texttt{Volume} \\ +0.0002\ \texttt{Time} + 0.7632\ \texttt{lag.log.High}\end{aligned}$$

Summary

We started off assuming a naïve model before noting its collinearity (seen in the scatterplot matrix and regression/VIF output) and utilizing a best subset model in its place. After establishing a best subset model, we re-did the regression task and were satisfied with the model's coverage, predictive potential, and lack of collinearity. We noted that the Durbin-Watson statistic was strongly rejecting the null hypothesis of no autocorrelation, and we employed the four big concepts associated with time series data in order to make sense of what was going on.

We considered the four big ideas behind time series data – detrending, deseasonalizing, lagging, and differencing. Of them, we found that detrending – that is, accounting for time as a predictor variable – and lagging – utilizing last week's stock price `log.High` as a predictor for this week's stock price `log.High` – proved most pertinent and impactful when discussing stock data with no seasonalization.

We saw that for a prediction task, utilizing last week's result was more important than anything else in predicting this week's result, and so on.

This report gave me a much better handle on using R as a programming language. Utilizing the `dplyr` module made the tricky – often messy – task of working with the data frames much easier, and actually made sense to implement. Despite having to context switch between Python and Java for other courses, working with R has become much easier to do over time (most likely

due to the steep learning curve), however I have yet to embrace functional programming. That said, this project gave me a much stronger feel for working with R for data analysis, visualization, and reporting.

Resources

All files pertaining to this report, including the R data analysis script, dataset, plots, and this PDF are open to the public and hosted on GitHub (`github.com/dannyfig/multivariate/time_series_regression`).

I sourced Tesla's weekly stock data through Yahoo Finance, which provides free financial stock data for as long as your search term has been around. I then sourced Tesla's weekly trend data through Google Trends, which allows you to download a CSV of your search term's history from a start date to an end date.

Data

I feel the need to explain the added column provided by Google Trends, `Search.Interest`. According to Google, the numbers represent search interest *relative to the highest point on the chart*. So, consider the largest value (100) in the `Search.Interest` column – the value on 28 March, 2016. The 28th – just 3 days before the announcement of the Tesla Model S, which went on to receive 180,000 preorders in 24 hours and 325,000 preorders in one week ($7.5 billion and $14 billion, respectively) – represents the relative highest number of search volume for the "tesla" search term. Therefore, looking at, say, 4 April 2016's `Search.Interest` value of 65, we mean to say that the 4th of April had 65% the search traffic relative to the 28th of March.

Aside from that, here's a snapshot of our data:

	Date	Open	High	Low	Close	Volume	S.I	log.High	Time	lag.log.High
1	2010-07-05	20.00	20.00	14.98	17.40	6.3876	8	1.301030	1	NA
2	2010-07-12	17.95	21.50	16.90	20.64	3.0877	7	1.332438	2	1.301030
3	2010-07-19	21.37	22.25	19.50	21.29	1.4351	7	1.347330	3	1.332438
4	2010-07-26	21.50	21.50	19.55	19.94	0.6104	7	1.332438	4	1.347330
5	2010-08-02	20.50	22.18	19.52	19.59	0.8799	6	1.345962	5	1.332438
6	2010-08-09	19.90	19.98	17.39	18.32	0.8433	6	1.300595	6	1.345962
7	2010-08-16	18.45	19.59	18.26	19.10	0.4820	6	1.292034	7	1.300595
8	2010-08-23	19.09	20.39	18.56	19.70	0.6155	6	1.309417	8	1.292034
9	2010-08-30	19.70	21.30	19.33	21.05	0.4701	6	1.328380	9	1.309417
10	2010-09-07	20.61	21.05	19.76	20.17	0.3236	6	1.323252	10	1.328380
..	...	..	..	..	..	...	..	...	..	...

292	2016-02-01	188.76	199.52	157.74	162.60	6.5454	25	2.299986	292	2.308714
293	2016-02-08	157.10	163.26	141.05	151.04	9.6749	29	2.212880	293	2.299986
294	2016-02-16	158.70	172.95	154.11	166.58	4.5254	23	2.237921	294	2.212880
295	2016-02-22	170.12	192.00	167.84	190.34	5.6346	21	2.283301	295	2.237921
296	2016-02-29	192.40	204.03	181.50	201.04	5.4609	22	2.309694	296	2.283301
297	2016-03-07	197.68	213.29	197.40	207.50	4.2207	22	2.328970	297	2.309694
298	2016-03-14	212.65	234.48	210.64	232.74	3.8203	22	2.370106	298	2.328970
299	2016-03-21	235.34	239.88	215.00	227.75	4.8656	24	2.379994	299	2.370106
300	2016-03-28	231.61	247.90	225.00	237.59	7.1637	100	2.394277	300	2.379994
301	2016-04-04	249.12	269.34	240.00	250.07	10.2321	65	2.430301	301	2.394277
302	2016-04-11	251.00	258.99	245.30	249.92	9.1539	43	2.413283	302	2.430301