Statistical Analysis ON Ethereum

Arti Mukati

April 16, 2018

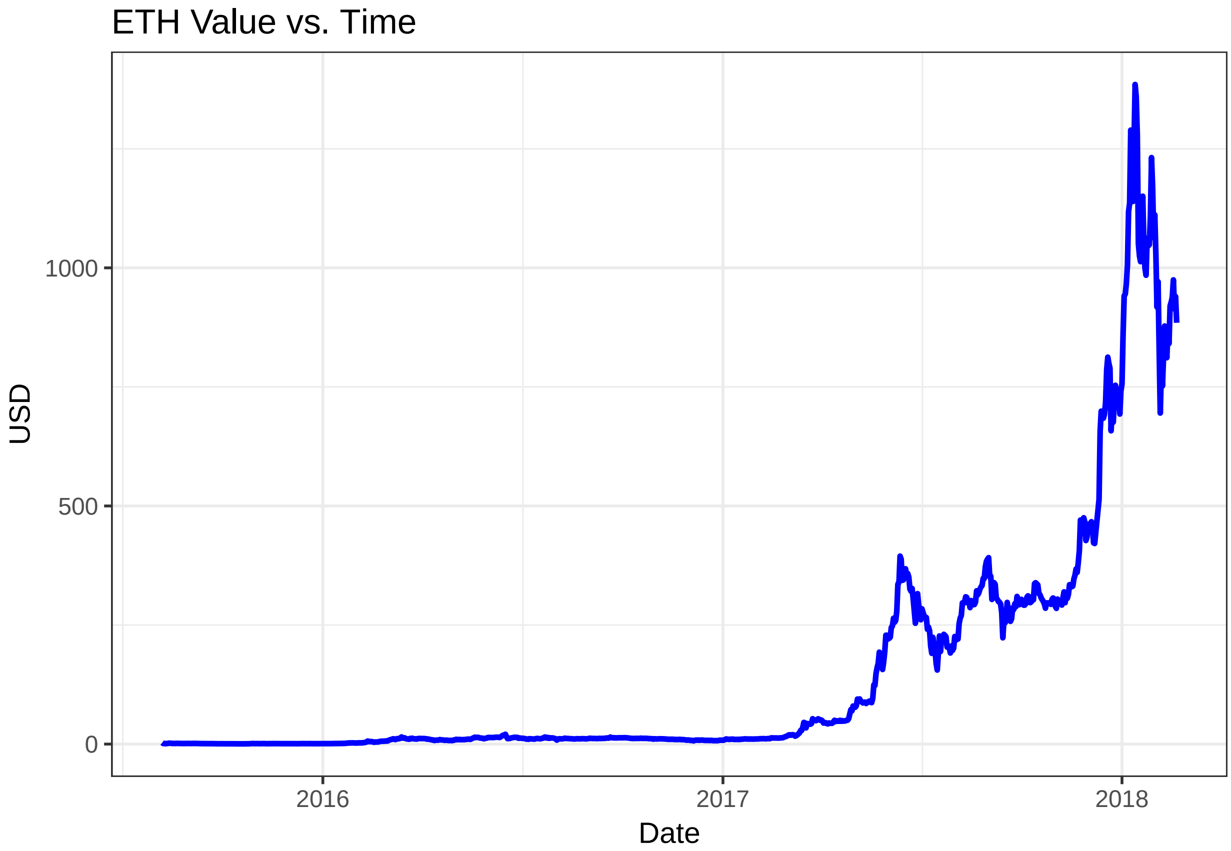
Read Data And loading library

Cryptocurrency has grown exponentially in value, resulting in it being the frequent topic of conversation and news. Many know that bitcoin is a type of cryptocurrency, but what isn't as known is that there are various types of cryptocurrency like Ethereum which offer slightly different capabilities. For the purpose of this analysis we will focus on solely Ethereum to keep control of the scope. Ethereum was created in 2013 by a 19-year-old Russian programmer and launched in 2015. For the first two years its price remained below $10. Then, in 2017, it exploded. In the space of 12 months, one unit of the cryptocurrency - called an ether - surged in value to be worth around $1,400 at its peak in January 2018.

Research Question Since Ethereum is gaining Traction after the Bitcoin collapse and seems to have a future with continuous modification in the technology, I chose to analyze the Prices of the cryptocurrency. The main objective of the paper is to look at the prices of Ethereum since the year 2015 and predict the prices of Ethereum. For this Simple linear regression, Multiple regression and Time series analysis is used.

Price ofEtherum overtime

```r
a =ggplot(eth_data, aes(eth_data$Date, eth_data$Price)) +
  geom_line(color="blue", size = 1) +
  ggtitle('ETH Value vs. Time') +
  theme(plot.title = element_text(size=20, face="bold",
    margin = margin(10, 0, 10, 0)))+
  labs(x="Date", y="USD")+
  theme(axis.text.x=element_text(angle=50, vjust=0.5)) +theme_bw()
a
```



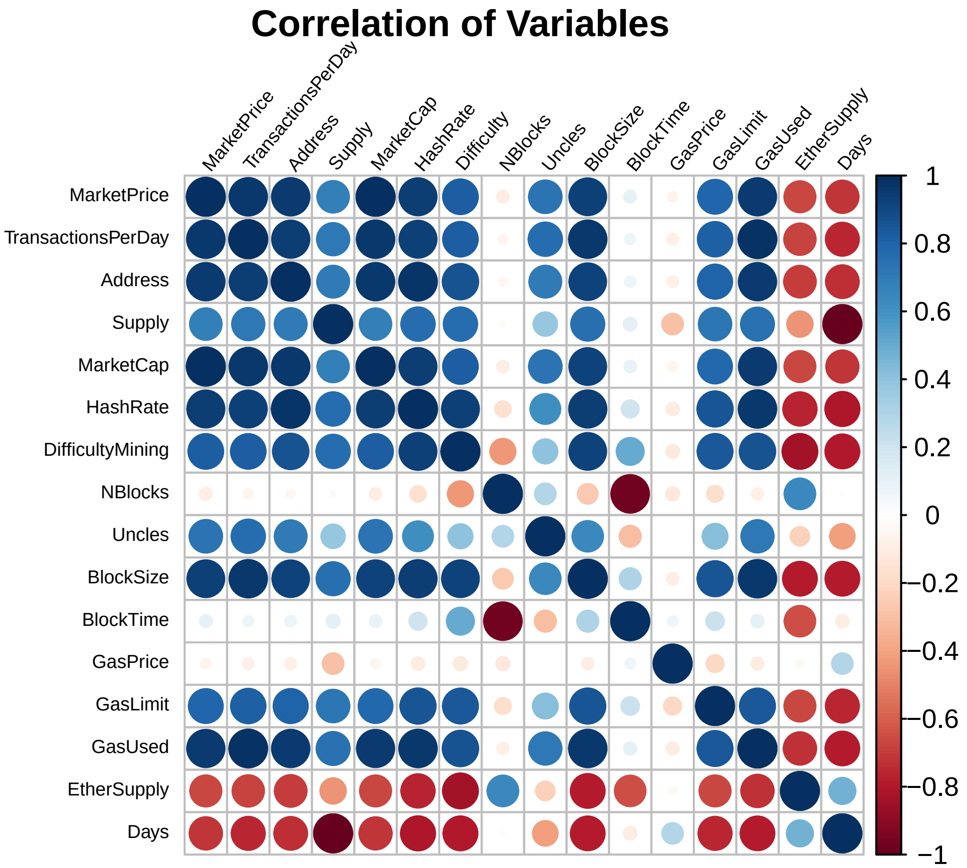
corellation analysis : The correlation plot gives the idea of the correlation between the variables. The variables which are strongly correlated to Market Price are Transactions Per day, Ethereum address, Market cap, Hash Rate, Block size and Gas used. But since these are correlated with each other, so I choose Market CAP for developing a simple linear model.

```
cor <- cor(eth_data[,c(3:17,19)])
```

```
colnames(cor) <- c("MarketPrice","TransactionsPerDay","Address", "Supply", "MarketCap", "HashRate","Dif

rownames(cor) <-c("MarketPrice","TransactionsPerDay","Address", "Supply", "MarketCap", "HashRate","Diff

corrplot(cor, method = "circle",  tl.srt = 50, tl.col = "black", tl.cex = 0.6, title = "Correlation of 
```



Simple Linear Regression:

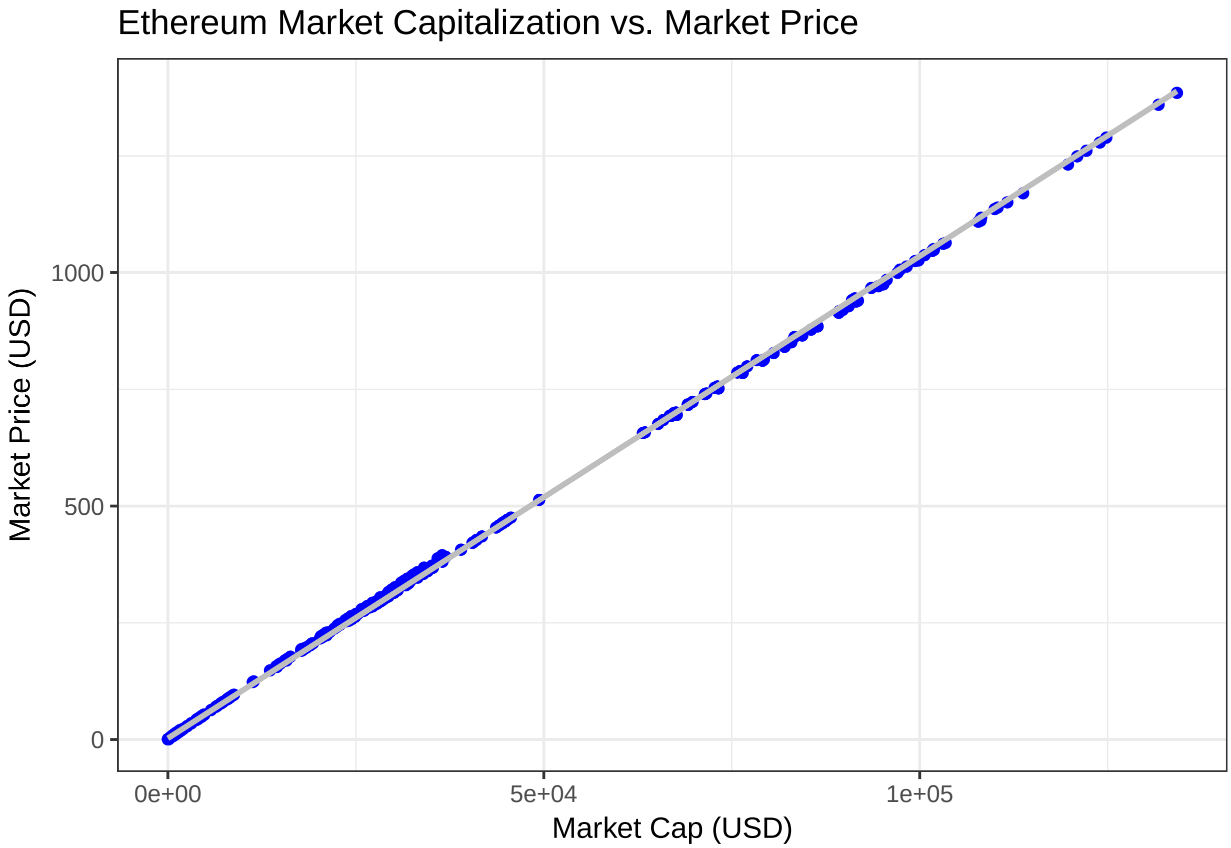
Regression analysis can be used to develop an equation showing how dependent and independent variables are related. The variable being predicted is called the dependent variable (eth_ether_price) and the variable used to predict the value of the dependent variable is called the independent variable. The regression analysis helps to determine how much the broader market capitalization impacts the prices of Ethereum.

PRICE vs MARKETCAP

```
eth_lm1 = lm(Price ~ eth_marketcap, data = eth_data)
summary(eth_lm1)
```

```
##
## Call:
## lm(formula = Price ~ eth_marketcap, data = eth_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.5995  -1.7854  -0.9017   0.9660  15.4061
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.494e+00  1.294e-01   19.28   <2e-16 ***
## eth_marketcap 1.033e-02  4.489e-06 2300.77   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.453 on 924 degrees of freedom
## Multiple R-squared:  0.9998, Adjusted R-squared:  0.9998
## F-statistic: 5.294e+06 on 1 and 924 DF,  p-value: < 2.2e-16
```

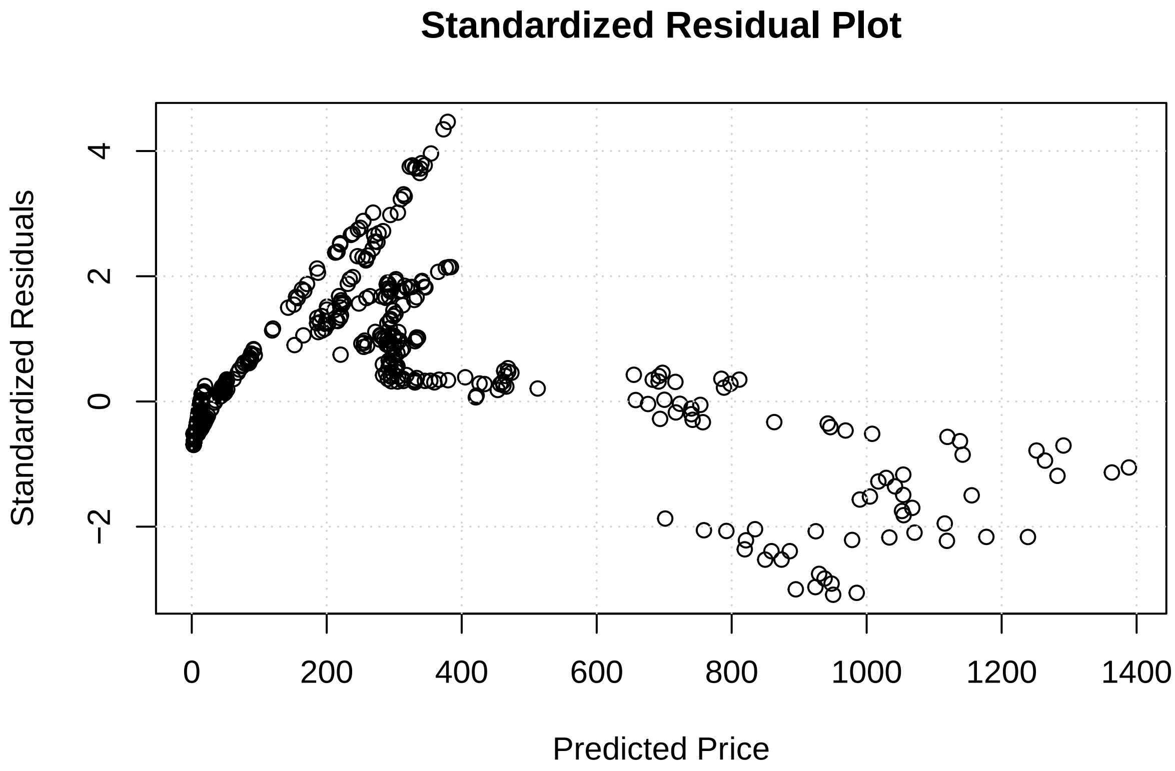
```
ggplot(eth_data, aes(eth_data$eth_marketcap, eth_data$Price)) +
  geom_point(color="blue") +
  ggtitle('Ethereum Market Capitalization vs. Market Price') +
  theme(plot.title = element_text(size=19.5, face="bold",
    margin = margin(10, 0, 10, 0)))+
  labs(x="Market Cap (USD)", y="Market Price (USD)")+
  theme(axis.text.x=element_text(angle=50, vjust=0.5)) +theme_bw() +stat_smooth(method = "lm",  formula
```



Residual plot model 1

```
std_residuals = rstandard(eth_lm1)

plot(eth_lm1$fitted.values,std_residuals, main="Standardized Residual Plot", ylab = "Standardized Residu
```



```
## integer(0)
```

The very high R2 values that you see is because the market price is determined by the market cap and vice versa. Example: The market cap is the number of ethers times the respective price. As the number of ethers at any given point is public knowledge, we can (theoretically) determine the market capitalization directly. This is what you see in the model.

Now modelling other parameters :

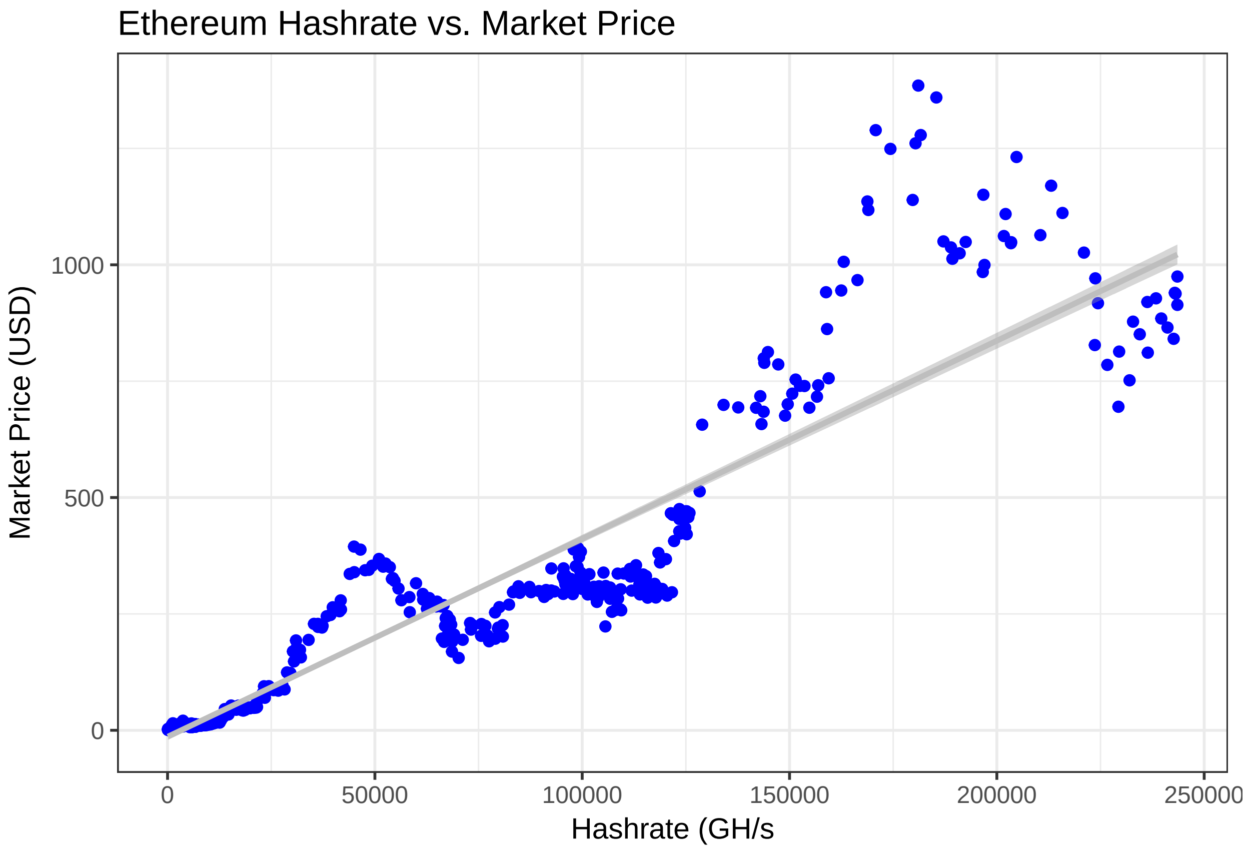
PRICE vs Hashrate

```
eth_lm2 = lm(Price ~ eth_hashrate, data = eth_data)
summary(eth_lm2)
```

```
##
## Call:
## lm(formula = Price ~ eth_hashrate, data = eth_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -266.51  -13.97    6.44   13.41  628.76
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.397e+01  3.471e+00  -4.026 6.15e-05 ***
## eth_hashrate  4.254e-03  5.043e-05  84.358  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 88.6 on 924 degrees of freedom
## Multiple R-squared:  0.8851, Adjusted R-squared:  0.885
## F-statistic:  7116 on 1 and 924 DF,  p-value: < 2.2e-16
```

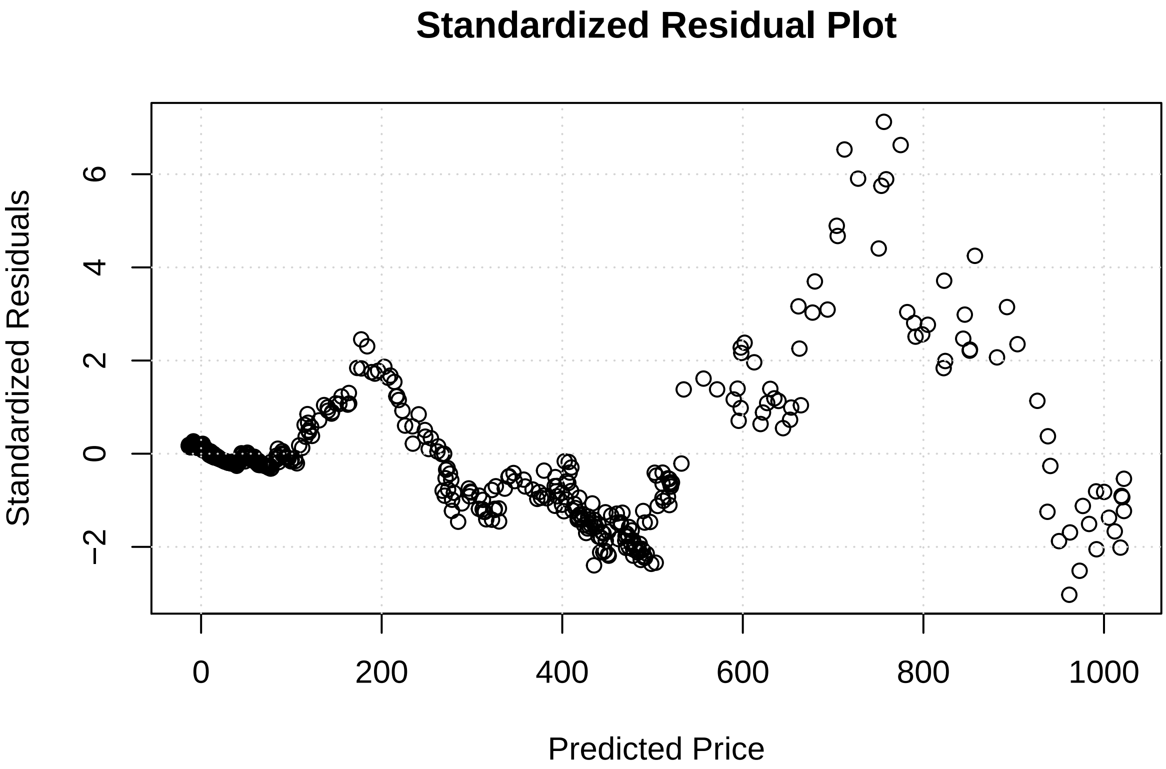
```
ggplot(eth_data, aes(eth_data$eth_hashrate, eth_data$Price)) +
  geom_point(color="blue") +
  ggtitle('Ethereum Hashrate vs. Market Price') +
  theme(plot.title = element_text(size=19.5, face="bold",
    margin = margin(10, 0, 10, 0)))+
  labs(x="Hashrate (GH/s", y="Market Price (USD)")+
  theme(axis.text.x=element_text(angle=50, vjust=0.5)) +theme_bw() +stat_smooth(method = "lm",  formula
```



Considerably good R2 values and the model appears to be significant. Residual plot model 2

```
std_residuals = rstandard(eth_lm2)

plot(eth_lm2$fitted.values,std_residuals, main="Standardized Residual Plot", ylab = "Standardized Resid
```



```
## integer(0)
```

The residual analysis graph voilates the assumptionsof the error term here The graph is having non constant variance.

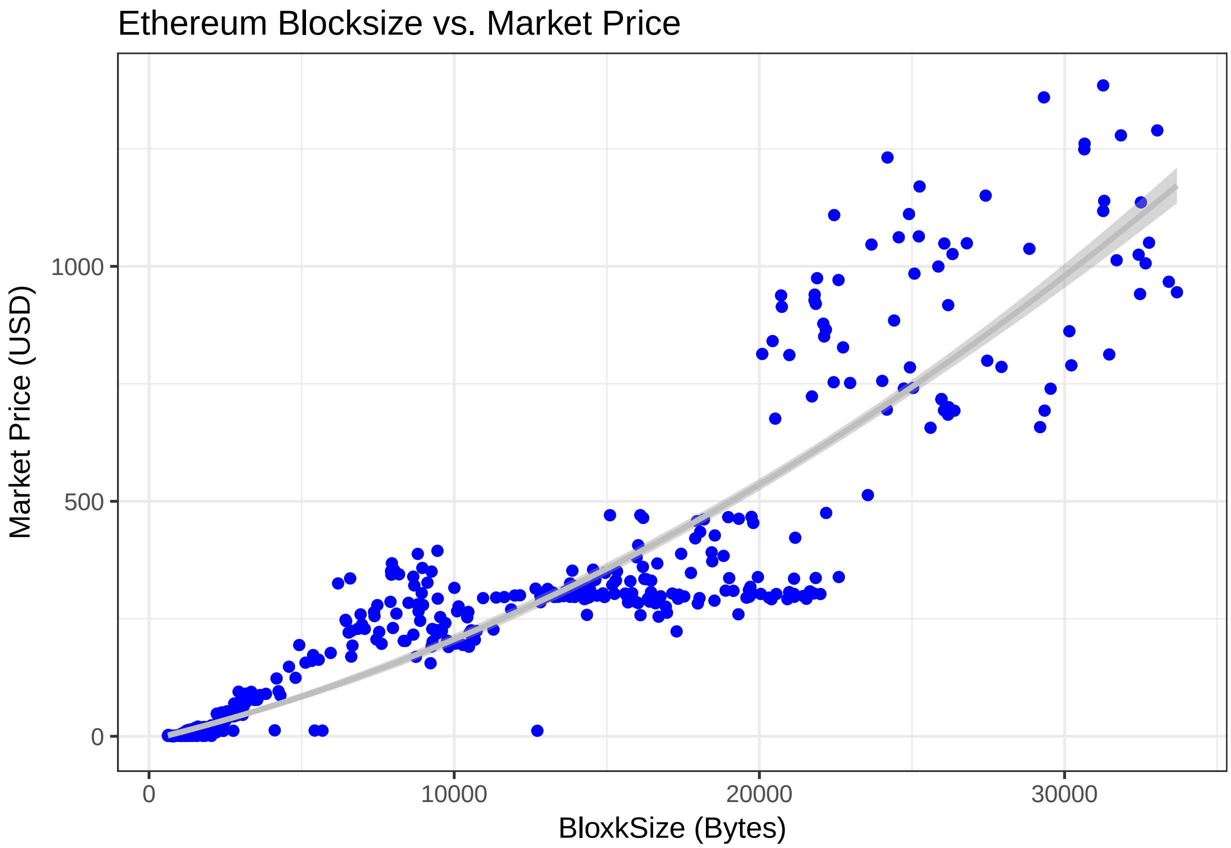
Running a simple linear regerssion with other highly corellated varibales but the residual analysis voilates the assumptions of error term.

PRICE VS hash rate

```r
eth_lm3 = lm(Price ~ eth_blocksize, data = eth_data)
eth_lm3 = lm(Price ~ poly(eth_blocksize,2), data = eth_data)
summary(eth_lm3)
```

```
##
## Call:
## lm(formula = Price ~ poly(eth_blocksize, 2), data = eth_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -312.33   -9.05   -5.27   -1.24  523.62
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              145.402      2.917   49.85   <2e-16 ***
## poly(eth_blocksize, 2)1 7397.192     88.755   83.34   <2e-16 ***
## poly(eth_blocksize, 2)2 1063.287     88.755   11.98   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 88.75 on 923 degrees of freedom
## Multiple R-squared:  0.8848, Adjusted R-squared:  0.8846
## F-statistic:  3545 on 2 and 923 DF,  p-value: < 2.2e-16
```

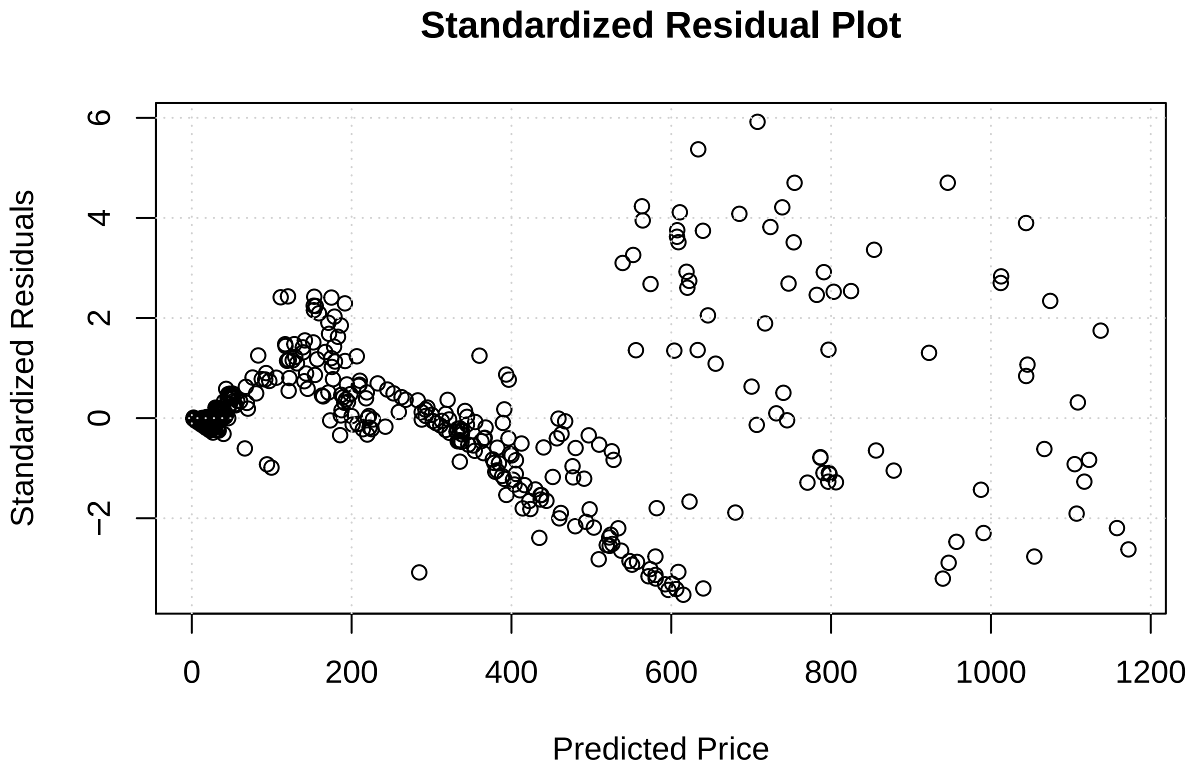
```
ggplot(eth_data, aes(eth_data$eth_blocksize, eth_data$Price)) +
  geom_point(color="blue") +
  ggtitle('Ethereum Blocksize vs. Market Price') +
  theme(plot.title = element_text(size=19.5, face="bold",
    margin = margin(10, 0, 10, 0)))+
  labs(x="BloxkSize (Bytes)", y="Market Price (USD)")+
  theme(axis.text.x=element_text(angle=50, vjust=0.5)) +theme_bw() +stat_smooth(method = "lm",  formula
```



Residual plot model 3

```
std_residuals = rstandard(eth_lm3)

plot(eth_lm3$fitted.values,std_residuals, main="Standardized Residual Plot", ylab = "Standardized Resid
```



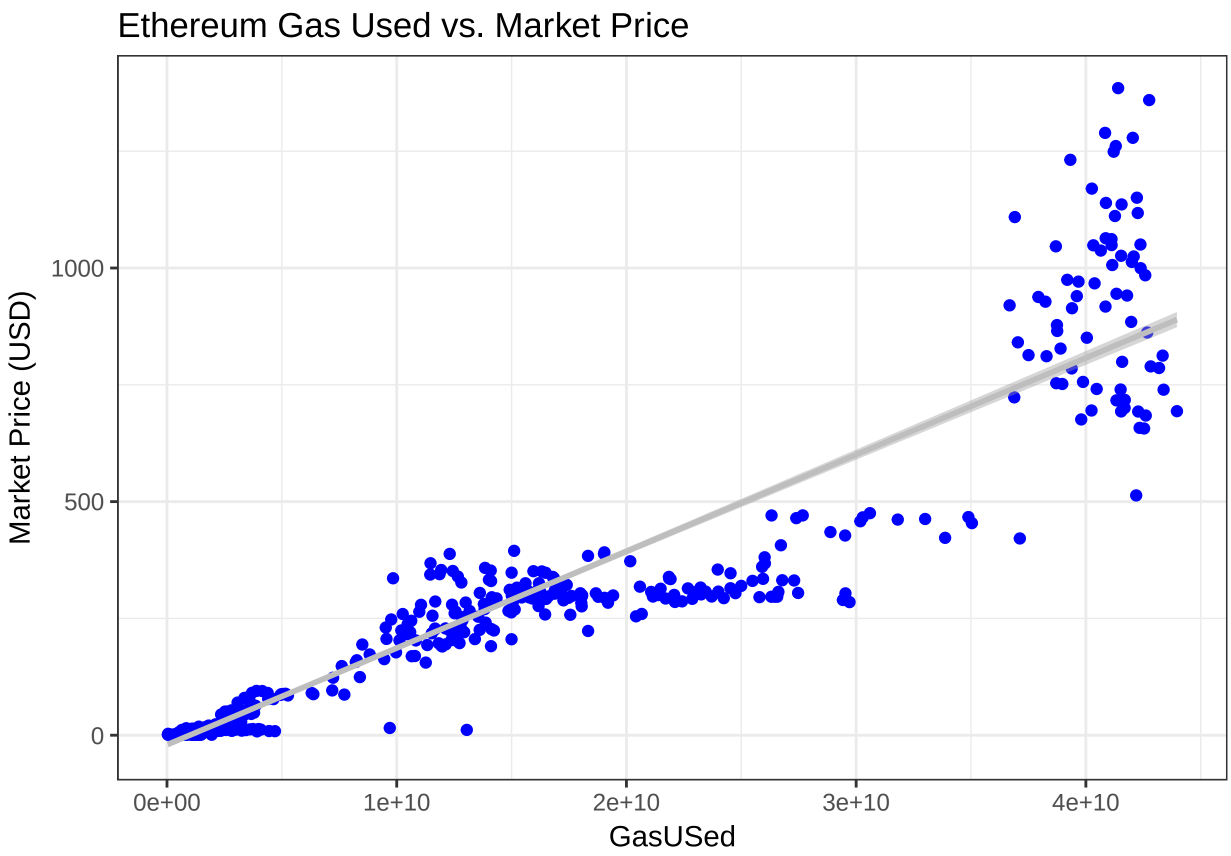
```
## integer(0)
```

PRICE VS Gas used using a quadratic model

```
eth_lm4 = lm(Price ~  eth_gasused, data = eth_data)
summary(eth_lm4)
```

```
##
## Call:
## lm(formula = Price ~ eth_gasused, data = eth_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -339.60   -6.14    4.91   15.53  548.40
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.046e+01  3.173e+00   -6.45  1.8e-10 ***
## eth_gasused  2.070e-08  2.198e-10   94.16  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 80.29 on 924 degrees of freedom
## Multiple R-squared:  0.9056, Adjusted R-squared:  0.9055
## F-statistic:  8866 on 1 and 924 DF,  p-value: < 2.2e-16
```

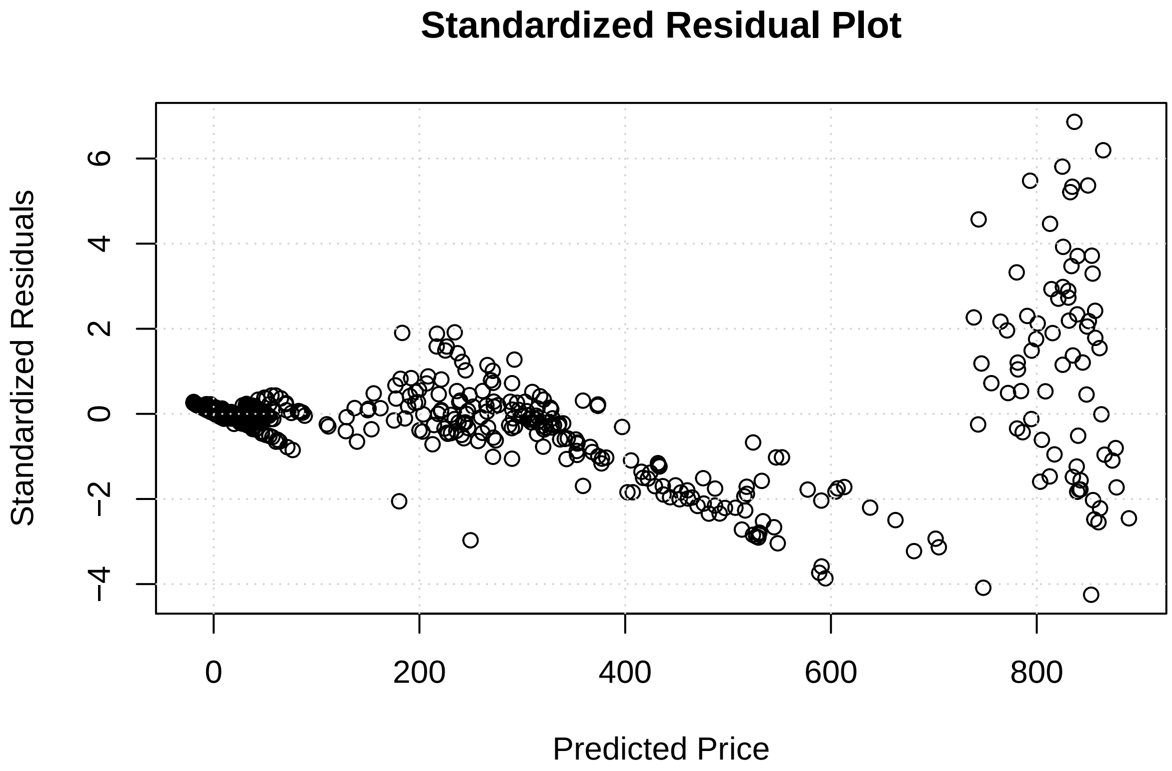
```
ggplot(eth_data, aes(eth_data$eth_gasused, eth_data$Price)) +
  geom_point(color="blue") +
  ggtitle('Ethereum Gas Used vs. Market Price') +
  theme(plot.title = element_text(size=19.5, face="bold",
    margin = margin(10, 0, 10, 0)))+
  labs(x="GasUSed", y="Market Price (USD)")+
  theme(axis.text.x=element_text(angle=50, vjust=0.5)) +theme_bw() +stat_smooth(method = "lm",  formula
```



Residual plot model 4

```
std_residuals = rstandard(eth_lm4)

plot(eth_lm4$fitted.values,std_residuals, main="Standardized Residual Plot", ylab = "Standardized Resid
```



```
## integer(0)
```

```
eth_lm5 = lm(Price ~ poly(eth_tx,2), data = eth_data)
```

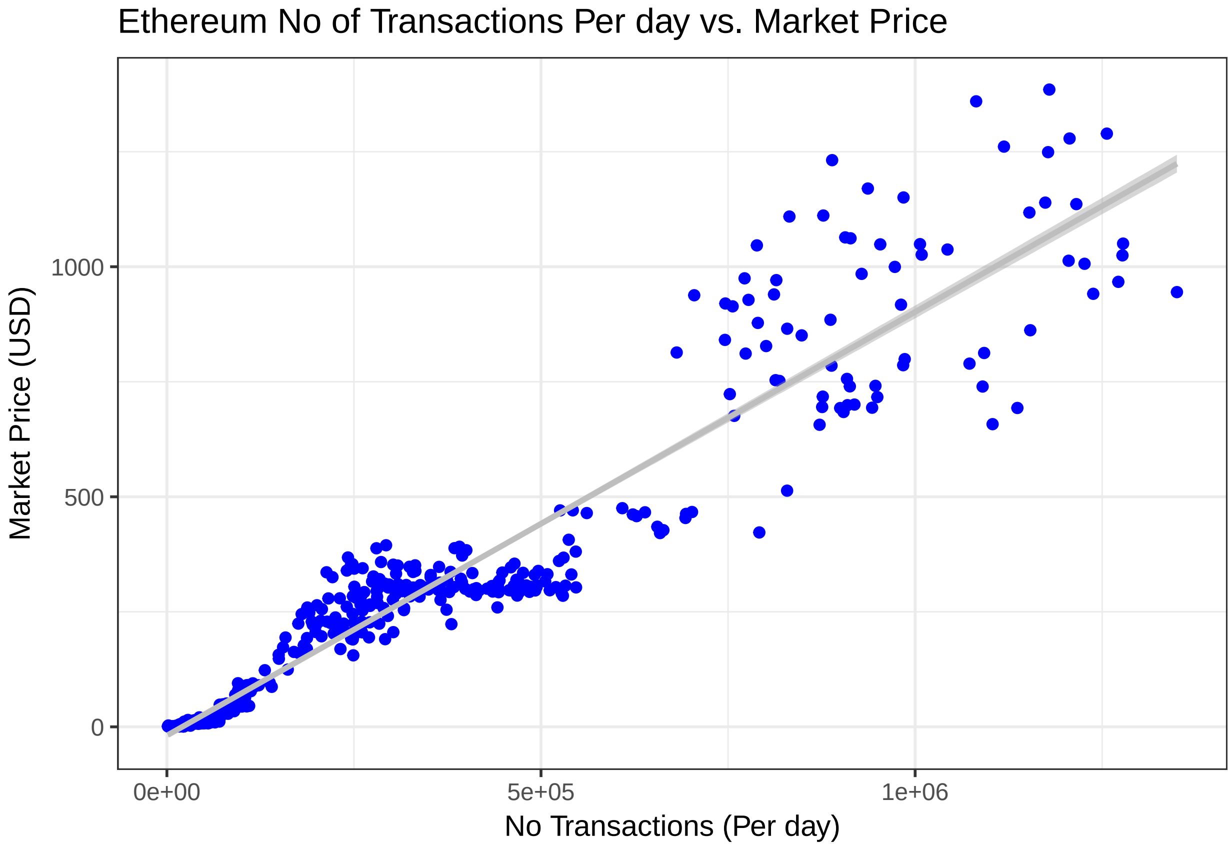
PRICE VS transaction per day

```
eth_lm5 = lm(Price ~ eth_tx, data = eth_data)

summary(eth_lm5)
```

```
##
## Call:
## lm(formula = Price ~ eth_tx, data = eth_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -339.25  -13.56   -4.33   13.28  431.91
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.892e+01  2.665e+00  -7.101 2.47e-12 ***
## eth_tx       9.207e-04  8.153e-06 112.921  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 67.94 on 924 degrees of freedom
## Multiple R-squared:  0.9324, Adjusted R-squared:  0.9324
## F-statistic: 1.275e+04 on 1 and 924 DF,  p-value: < 2.2e-16
```

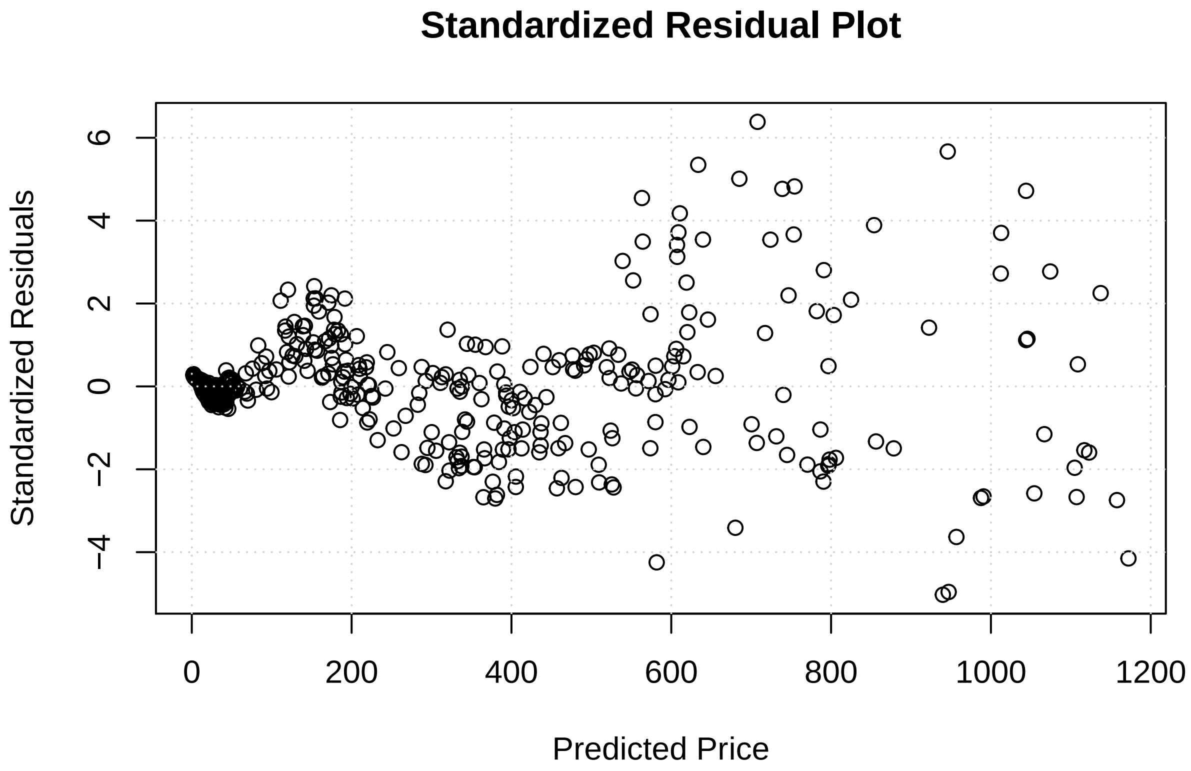
```
ggplot(eth_data, aes(eth_data$eth_tx, eth_data$Price)) +
  geom_point(color="blue") +
  ggtitle('Ethereum No of Transactions Per day vs. Market Price') +
  theme(plot.title = element_text(size=19.5, face="bold",
    margin = margin(10, 0, 10, 0)))+
  labs(x="No Transactions (Per day)", y="Market Price (USD)")+
  theme(axis.text.x=element_text(angle=50, vjust=0.5)) +theme_bw() +stat_smooth(method = "lm",  formula
```



Residual plot model 5

```
std_residuals = rstandard(eth_lm5)

plot(eth_lm3$fitted.values,std_residuals, main="Standardized Residual Plot", ylab = "Standardized Resid
```



```
## integer(0)
```

Multiple regression including all variables

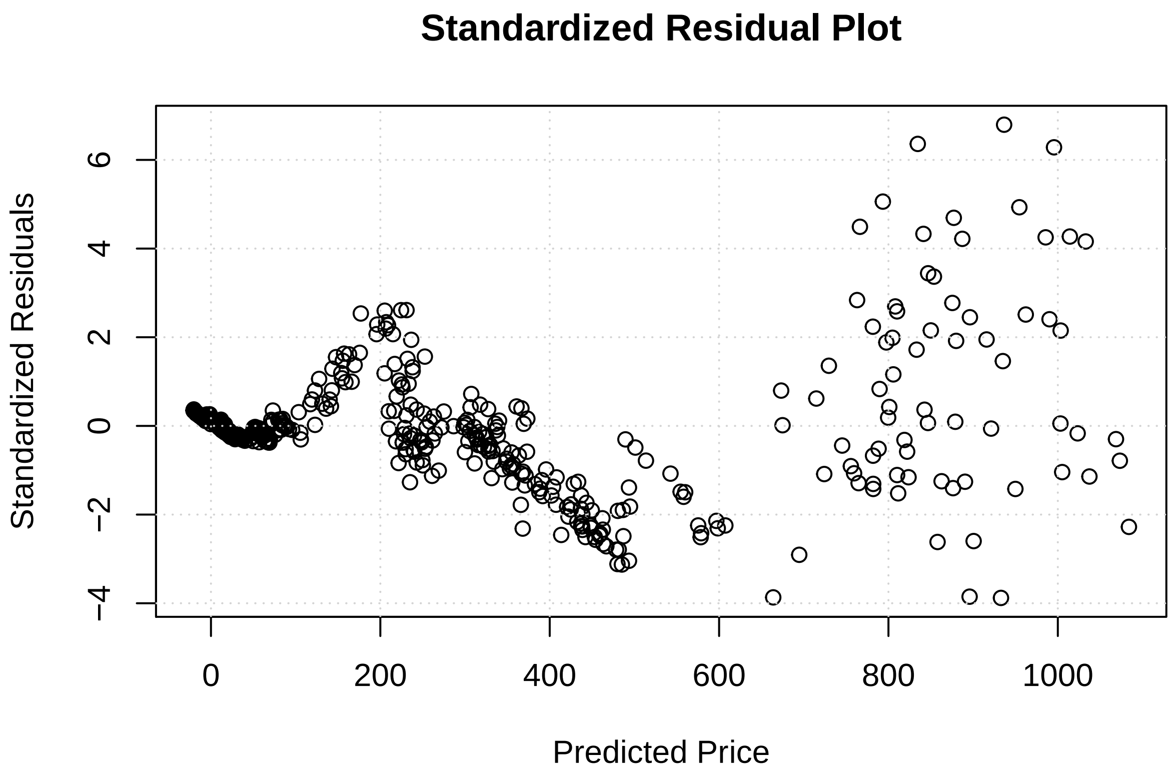
```
eth_lm_all = lm(eth_data$Price ~ eth_hashrate + eth_tx , data = eth_data)
summary(eth_lm_all)
```

```
##
## Call:
## lm(formula = eth_data$Price ~ eth_hashrate + eth_tx, data = eth_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -241.43   -9.86    0.16   15.61  422.91
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.173e+01  2.471e+00  -8.796   <2e-16 ***
## eth_hashrate  1.311e-03  1.034e-04  12.674   <2e-16 ***
## eth_tx        6.614e-04  2.180e-05  30.332   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 62.74 on 923 degrees of freedom
## Multiple R-squared:  0.9424, Adjusted R-squared:  0.9423
## F-statistic:  7557 on 2 and 923 DF,  p-value: < 2.2e-16
```

Residual Plot Analysis for multiple regression model

```
std_residuals = rstandard(eth_lm_all)

plot(eth_lm_all$fitted.values,std_residuals, main="Standardized Residual Plot", ylab = "Standardized Res
```



```
## integer(0)
```

Variable Selection for Multiple Regression: Single-linear regression looks to determine the value of a dependent variable using one independent variable. In multiple regression, multiple independent variables are used to predict the dependent variable. For multiple regression, the F test is used to determine whether a significant relationship exists between the dependent variable and the set of all the independent variables. The F test is referred to as the test of overall significance. A separate t test is conducted for each of the independent variables in the model. Each of these t tests is referred to as a test of individual significance. Including all the variables increases the complexity of the equation. Therefore, we need to choose the equation which can explain most of the variance without making it more complex. To decide which variables to include I will be using the Stepwise regression. Although Best subsets regression is best for this process, but because I was not able to run that in my R (It got my system Hanged) so decided to go with the stepwise regression.

Forward Lm Method

```
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 3.4.4
```

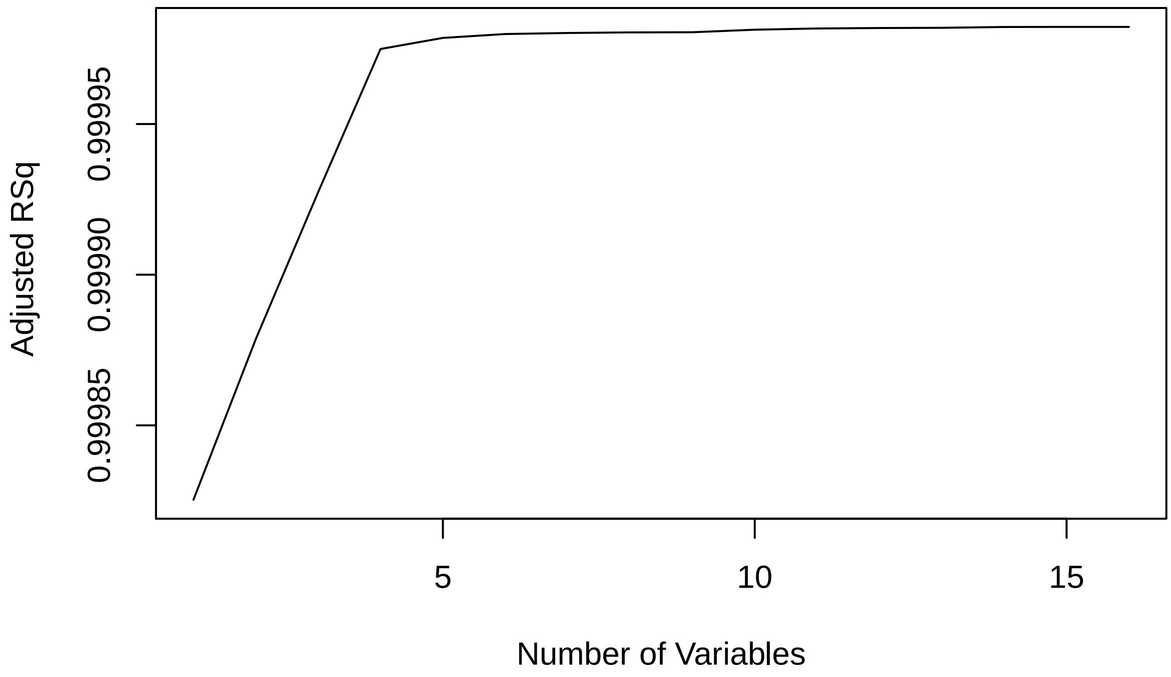
```
forward_lm = regsubsets(Price ~ ., data = eth_data[,c(3:17, 19:20)], method = "forward", nvmax = length
(forward_lm_summary = summary(forward_lm))
```

```
## Subset selection object
## Call: regsubsets.formula(Price ~ ., data = eth_data[, c(3:17, 19:20)],
##     method = "forward", nvmax = length(eth_data[, c(3:20)]) -
##         1)
## 16 Variables  (and intercept)
##                 Forced in Forced out
## eth_tx              FALSE      FALSE
## eth_address         FALSE      FALSE
## eth_supply          FALSE      FALSE
## eth_marketcap       FALSE      FALSE
## eth_hashrate        FALSE      FALSE
## eth_difficulty      FALSE      FALSE
## eth_blocks          FALSE      FALSE
## eth_uncles          FALSE      FALSE
## eth_blocksize       FALSE      FALSE
## eth_blocktime       FALSE      FALSE
## eth_gasprice        FALSE      FALSE
## eth_gaslimit        FALSE      FALSE
## eth_gasused         FALSE      FALSE
## eth_ethersupply     FALSE      FALSE
## Days                FALSE      FALSE
## Date                FALSE      FALSE
## 1 subsets of each size up to 16
## Selection Algorithm: forward
##           eth_tx eth_address eth_supply eth_marketcap eth_hashrate
## 1  ( 1 )  " "    " "         " "        "*"           " "
## 2  ( 1 )  " "    " "         "*"        "*"           " "
## 3  ( 1 )  " "    "*"         "*"        "*"           " "
## 4  ( 1 )  " "    "*"         "*"        "*"           "*"
## 5  ( 1 )  " "    "*"         "*"        "*"           "*"
## 6  ( 1 )  "*"    "*"         "*"        "*"           "*"
## 7  ( 1 )  "*"    "*"         "*"        "*"           "*"
## 8  ( 1 )  "*"    "*"         "*"        "*"           "*"
## 9  ( 1 )  "*"    "*"         "*"        "*"           "*"
## 10  ( 1 ) "*"    "*"         "*"        "*"           "*"
## 11  ( 1 ) "*"    "*"         "*"        "*"           "*"
## 12  ( 1 ) "*"    "*"         "*"        "*"           "*"
## 13  ( 1 ) "*"    "*"         "*"        "*"           "*"
## 14  ( 1 ) "*"    "*"         "*"        "*"           "*"
## 15  ( 1 ) "*"    "*"         "*"        "*"           "*"
## 16  ( 1 ) "*"    "*"         "*"        "*"           "*"
##           eth_difficulty eth_blocks eth_uncles eth_blocksize eth_blocktime
## 1  ( 1 )  " "            " "        " "        " "           " "
## 2  ( 1 )  " "            " "        " "        " "           " "
## 3  ( 1 )  " "            " "        " "        " "           " "
## 4  ( 1 )  " "            " "        " "        " "           " "
## 5  ( 1 )  " "            " "        " "        " "           " "
## 6  ( 1 )  " "            " "        " "        " "           " "
## 7  ( 1 )  " "            " "        "*"        " "           " "
## 8  ( 1 )  " "            "*"        "*"        " "           " "
## 9  ( 1 )  " "            "*"        "*"        " "           " "
## 10  ( 1 ) " "            "*"        "*"        " "           " "
## 11  ( 1 ) " "            "*"        "*"        " "           " "
## 12  ( 1 ) " "            "*"        "*"        "*"           " "
## 13  ( 1 ) " "            "*"        "*"        "*"           "*"
## 14  ( 1 ) "*"            "*"        "*"        "*"           "*"
## 15  ( 1 ) "*"            "*"        "*"        "*"           "*"
## 16  ( 1 ) "*"            "*"        "*"        "*"           "*"
##           eth_gasprice eth_gaslimit eth_gasused eth_ethersupply Days Date
## 1  ( 1 )  " "          " "          " "         " "             " "  " "
## 2  ( 1 )  " "          " "          " "         " "             " "  " "
## 3  ( 1 )  " "          " "          " "         " "             " "  " "
## 4  ( 1 )  " "          " "          " "         " "             " "  " "
## 5  ( 1 )  " "          " "          "*"         " "             " "  " "
## 6  ( 1 )  " "          " "          "*"         " "             " "  " "
## 7  ( 1 )  " "          " "          "*"         " "             " "  " "
## 8  ( 1 )  " "          " "          "*"         " "             " "  " "
## 9  ( 1 )  " "          " "          "*"         " "             "*"  " "
## 10  ( 1 ) " "          " "          "*"         " "             "*"  "*"
## 11  ( 1 ) " "          " "          "*"         "*"             "*"  "*"
## 12  ( 1 ) " "          " "          "*"         "*"             "*"  "*"
## 13  ( 1 ) " "          " "          "*"         "*"             "*"  "*"
## 14  ( 1 ) " "          " "          "*"         "*"             "*"  "*"
## 15  ( 1 ) " "          "*"          "*"         "*"             "*"  "*"
## 16  ( 1 ) "*"          "*"          "*"         "*"             "*"  "*"
```

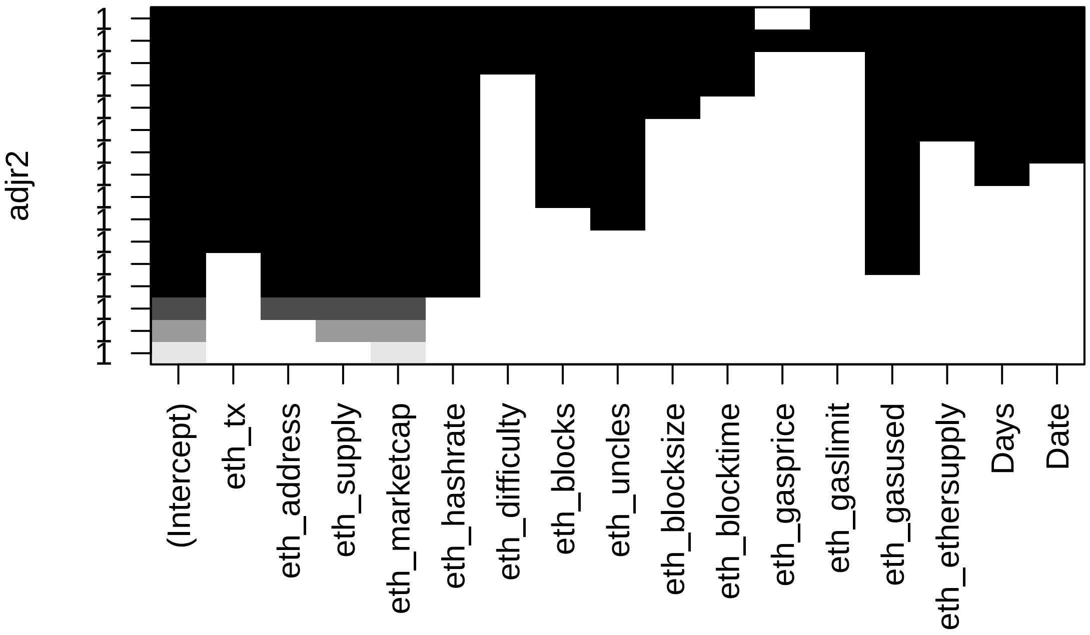
```
forward_lm_summary$adjr2
```

```
##  [1] 0.9998253 0.9998786 0.9999274 0.9999749 0.9999786 0.9999799 0.9999802
##  [8] 0.9999804 0.9999805 0.9999813 0.9999817 0.9999819 0.9999819 0.9999822
## [15] 0.9999822 0.9999822
```

```
plot(forward_lm_summary$adjr2,xlab =" Number of Variables ",ylab=" Adjusted RSq",type="l")
```



```
plot(forward_lm, scale='adjr2')
```



Back ward lm

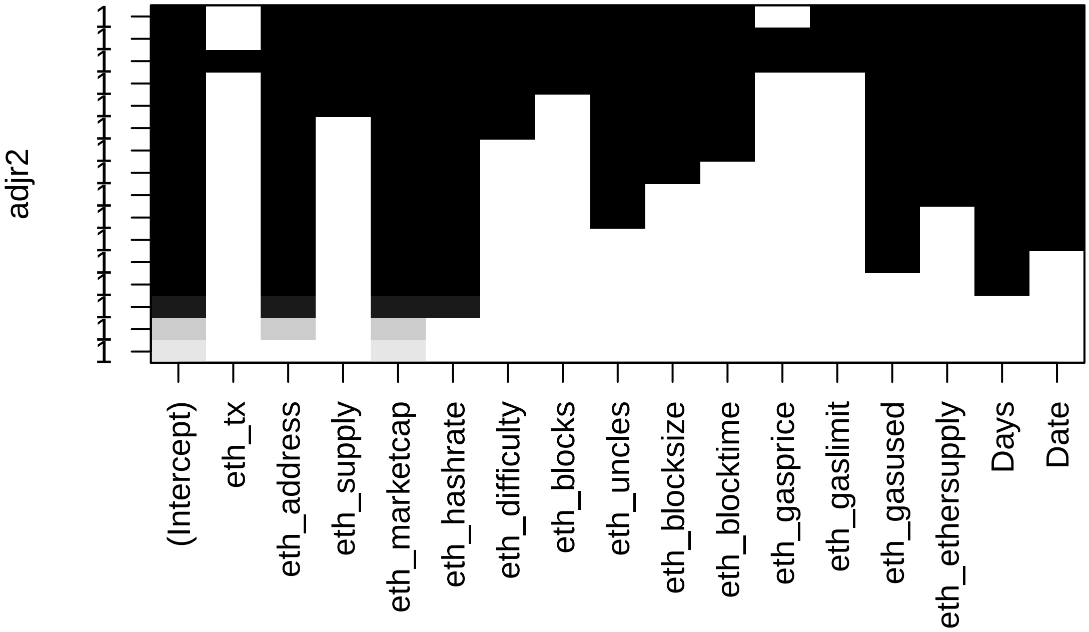
```
backward_lm = regsubsets(Price ~ ., data = eth_data[,c(3:17, 19:20)], method = "backward", nvmax = leng
(backward_lm_summary = summary(backward_lm))
```

```
## Subset selection object
## Call: regsubsets.formula(Price ~ ., data = eth_data[, c(3:17, 19:20)],
##     method = "backward", nvmax = length(eth_data[, c(3:20)]) -
##         1)
## 16 Variables  (and intercept)
##                 Forced in Forced out
## eth_tx              FALSE      FALSE
## eth_address         FALSE      FALSE
## eth_supply          FALSE      FALSE
## eth_marketcap       FALSE      FALSE
## eth_hashrate        FALSE      FALSE
## eth_difficulty      FALSE      FALSE
## eth_blocks          FALSE      FALSE
## eth_uncles          FALSE      FALSE
## eth_blocksize       FALSE      FALSE
## eth_blocktime       FALSE      FALSE
## eth_gasprice        FALSE      FALSE
## eth_gaslimit        FALSE      FALSE
## eth_gasused         FALSE      FALSE
## eth_ethersupply     FALSE      FALSE
## Days                FALSE      FALSE
## Date                FALSE      FALSE
## 1 subsets of each size up to 16
## Selection Algorithm: backward
##           eth_tx eth_address eth_supply eth_marketcap eth_hashrate
## 1  ( 1 )  " "    " "         " "        "*"           " "
## 2  ( 1 )  " "    "*"         " "        "*"           " "
## 3  ( 1 )  " "    "*"         " "        "*"           "*"
## 4  ( 1 )  " "    "*"         " "        "*"           "*"
## 5  ( 1 )  " "    "*"         " "        "*"           "*"
## 6  ( 1 )  " "    "*"         " "        "*"           "*"
## 7  ( 1 )  " "    "*"         " "        "*"           "*"
## 8  ( 1 )  " "    "*"         " "        "*"           "*"
## 9  ( 1 )  " "    "*"         " "        "*"           "*"
## 10  ( 1 ) " "    "*"         " "        "*"           "*"
## 11  ( 1 ) " "    "*"         " "        "*"           "*"
## 12  ( 1 ) " "    "*"         "*"        "*"           "*"
## 13  ( 1 ) " "    "*"         "*"        "*"           "*"
## 14  ( 1 ) " "    "*"         "*"        "*"           "*"
## 15  ( 1 ) " "    "*"         "*"        "*"           "*"
## 16  ( 1 ) "*"    "*"         "*"        "*"           "*"
##           eth_difficulty eth_blocks eth_uncles eth_blocksize eth_blocktime
## 1  ( 1 )  " "            " "        " "        " "           " "
## 2  ( 1 )  " "            " "        " "        " "           " "
## 3  ( 1 )  " "            " "        " "        " "           " "
## 4  ( 1 )  " "            " "        " "        " "           " "
## 5  ( 1 )  " "            " "        " "        " "           " "
## 6  ( 1 )  " "            " "        " "        " "           " "
## 7  ( 1 )  " "            " "        "*"        " "           " "
## 8  ( 1 )  " "            " "        "*"        " "           " "
## 9  ( 1 )  " "            " "        "*"        "*"           " "
## 10  ( 1 ) " "            " "        "*"        "*"           "*"
## 11  ( 1 ) "*"            " "        "*"        "*"           "*"
## 12  ( 1 ) "*"            " "        "*"        "*"           "*"
## 13  ( 1 ) "*"            "*"        "*"        "*"           "*"
## 14  ( 1 ) "*"            "*"        "*"        "*"           "*"
## 15  ( 1 ) "*"            "*"        "*"        "*"           "*"
## 16  ( 1 ) "*"            "*"        "*"        "*"           "*"
##           eth_gasprice eth_gaslimit eth_gasused eth_ethersupply Days Date
## 1  ( 1 )  " "          " "          " "         " "             " "  " "
## 2  ( 1 )  " "          " "          " "         " "             " "  " "
## 3  ( 1 )  " "          " "          " "         " "             " "  " "
## 4  ( 1 )  " "          " "          " "         " "             "*"  " "
## 5  ( 1 )  " "          " "          "*"         " "             "*"  " "
## 6  ( 1 )  " "          " "          "*"         " "             "*"  "*"
## 7  ( 1 )  " "          " "          "*"         " "             "*"  "*"
## 8  ( 1 )  " "          " "          "*"         "*"             "*"  "*"
## 9  ( 1 )  " "          " "          "*"         "*"             "*"  "*"
## 10  ( 1 ) " "          " "          "*"         "*"             "*"  "*"
## 11  ( 1 ) " "          " "          "*"         "*"             "*"  "*"
## 12  ( 1 ) " "          " "          "*"         "*"             "*"  "*"
## 13  ( 1 ) " "          " "          "*"         "*"             "*"  "*"
## 14  ( 1 ) " "          "*"          "*"         "*"             "*"  "*"
## 15  ( 1 ) "*"          "*"          "*"         "*"             "*"  "*"
## 16  ( 1 ) "*"          "*"          "*"         "*"             "*"  "*"
```

```
# get the adjusted r squared values for each size of the model
backward_lm_summary$adjr2 # seems like the model with 5 variables is best.
```

```
##  [1] 0.9998253 0.9998511 0.9999640 0.9999752 0.9999786 0.9999796 0.9999804
##  [8] 0.9999813 0.9999814 0.9999820 0.9999820 0.9999822 0.9999822 0.9999822
## [15] 0.9999822 0.9999822
```

```
plot(backward_lm, scale='adjr2')
```



Step wise regression

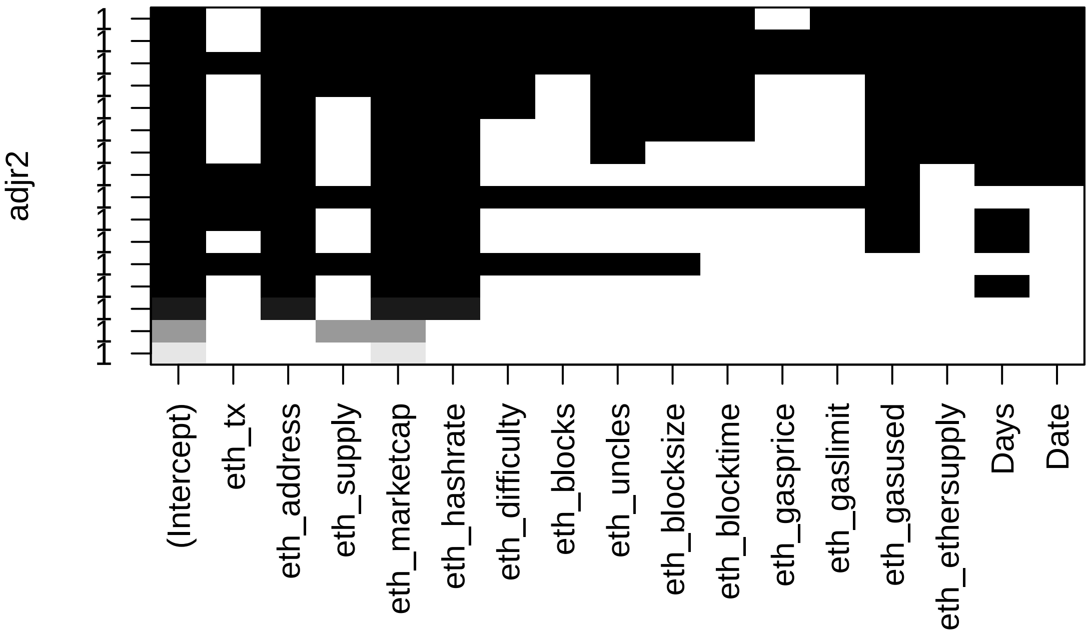
```
stepwise_lm = regsubsets(Price ~ ., data = eth_data[,c(3:17, 19:20)], method = "seqrep", nvmax = length
(stepwise_lm_summary = summary(stepwise_lm))
```

```
## Subset selection object
## Call: regsubsets.formula(Price ~ ., data = eth_data[, c(3:17, 19:20)],
##     method = "seqrep", nvmax = length(eth_data[, c(3:20)]) -
##         1)
## 16 Variables  (and intercept)
##                 Forced in Forced out
## eth_tx              FALSE      FALSE
## eth_address         FALSE      FALSE
## eth_supply          FALSE      FALSE
## eth_marketcap       FALSE      FALSE
## eth_hashrate        FALSE      FALSE
## eth_difficulty      FALSE      FALSE
## eth_blocks          FALSE      FALSE
## eth_uncles          FALSE      FALSE
## eth_blocksize       FALSE      FALSE
## eth_blocktime       FALSE      FALSE
## eth_gasprice        FALSE      FALSE
## eth_gaslimit        FALSE      FALSE
## eth_gasused         FALSE      FALSE
## eth_ethersupply     FALSE      FALSE
## Days                FALSE      FALSE
## Date                FALSE      FALSE
## 1 subsets of each size up to 16
## Selection Algorithm: 'sequential replacement'
##           eth_tx eth_address eth_supply eth_marketcap eth_hashrate
## 1  ( 1 )  " "    " "         " "        "*"           " "
## 2  ( 1 )  " "    " "         "*"        "*"           " "
## 3  ( 1 )  " "    "*"         " "        "*"           "*"
## 4  ( 1 )  " "    "*"         " "        "*"           "*"
## 5  ( 1 )  " "    "*"         " "        "*"           "*"
## 6  ( 1 )  "*"    "*"         " "        "*"           "*"
## 7  ( 1 )  "*"    "*"         " "        "*"           "*"
## 8  ( 1 )  " "    "*"         " "        "*"           "*"
## 9  ( 1 )  "*"    "*"         "*"        "*"           "*"
## 10  ( 1 ) " "    "*"         " "        "*"           "*"
## 11  ( 1 ) " "    "*"         " "        "*"           "*"
## 12  ( 1 ) " "    "*"         "*"        "*"           "*"
## 13  ( 1 ) "*"    "*"         "*"        "*"           "*"
## 14  ( 1 ) " "    "*"         "*"        "*"           "*"
## 15  ( 1 ) " "    "*"         "*"        "*"           "*"
## 16  ( 1 ) "*"    "*"         "*"        "*"           "*"
##           eth_difficulty eth_blocks eth_uncles eth_blocksize eth_blocktime
## 1  ( 1 )  " "            " "        " "        " "           " "
## 2  ( 1 )  " "            " "        " "        " "           " "
## 3  ( 1 )  " "            " "        " "        " "           " "
## 4  ( 1 )  " "            " "        " "        " "           " "
## 5  ( 1 )  " "            " "        " "        " "           " "
## 6  ( 1 )  " "            " "        " "        " "           " "
## 7  ( 1 )  " "            " "        " "        " "           " "
## 8  ( 1 )  " "            " "        "*"        " "           " "
## 9  ( 1 )  "*"            "*"        "*"        "*"           " "
## 10  ( 1 ) " "            " "        "*"        "*"           "*"
## 11  ( 1 ) "*"            " "        "*"        "*"           "*"
## 12  ( 1 ) "*"            " "        "*"        "*"           "*"
## 13  ( 1 ) "*"            "*"        "*"        "*"           "*"
## 14  ( 1 ) "*"            "*"        "*"        "*"           "*"
## 15  ( 1 ) "*"            "*"        "*"        "*"           "*"
## 16  ( 1 ) "*"            "*"        "*"        "*"           "*"
##           eth_gasprice eth_gaslimit eth_gasused eth_ethersupply Days Date
## 1  ( 1 )  " "          " "          " "         " "             " "  " "
## 2  ( 1 )  " "          " "          " "         " "             " "  " "
## 3  ( 1 )  " "          " "          " "         " "             " "  " "
## 4  ( 1 )  " "          " "          " "         " "             "*"  " "
## 5  ( 1 )  " "          " "          "*"         " "             "*"  " "
## 6  ( 1 )  " "          " "          "*"         " "             "*"  " "
## 7  ( 1 )  " "          " "          "*"         " "             "*"  "*"
## 8  ( 1 )  " "          " "          "*"         "*"             "*"  "*"
## 9  ( 1 )  " "          " "          " "         " "             " "  " "
## 10  ( 1 ) " "          " "          "*"         "*"             "*"  "*"
## 11  ( 1 ) " "          " "          "*"         "*"             "*"  "*"
## 12  ( 1 ) " "          " "          "*"         "*"             "*"  "*"
## 13  ( 1 ) "*"          "*"          "*"         " "             " "  " "
## 14  ( 1 ) " "          "*"          "*"         "*"             "*"  "*"
## 15  ( 1 ) "*"          "*"          "*"         "*"             "*"  "*"
## 16  ( 1 ) "*"          "*"          "*"         "*"             "*"  "*"
```

```r
stepwise_lm_summary$adjr2
```

```
##  [1] 0.9998253 0.9998786 0.9999640 0.9999752 0.9999786 0.9999799 0.9999807
##  [8] 0.9999813 0.9999769 0.9999820 0.9999820 0.9999822 0.9999804 0.9999822
## [15] 0.9999822 0.9999822
```

```r
plot(stepwise_lm, scale= "adjr2") #
```



Multiple Regression model for highly correlated variables Multiple Regression:

Summary of the multiple regression model using the variables selected using the above Stepwise regression method. The dependent variable being the Market price and the independent variables being hash Rate and the Ethereum addresses. The test for individual significance shows that the model is highly significant as the p-value is very low. The Adjusted R sq. is 1 which shows that the model can explain almost 100% of the variance in the

```r
eth_Mlm = lm(Price ~ eth_hashrate + eth_address, data = eth_data)
summary(eth_Mlm)
```

```
##
## Call:
## lm(formula = Price ~ eth_hashrate + eth_address, data = eth_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -351.29  -21.45   -0.51    2.69  505.45
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.107e-01  2.968e+00  -0.071    0.943
## eth_hashrate  1.899e-04  2.050e-04   0.926    0.354
## eth_address   3.951e-05  1.950e-06  20.258   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 73.76 on 923 degrees of freedom
## Multiple R-squared:  0.9204, Adjusted R-squared:  0.9203
## F-statistic:  5340 on 2 and 923 DF,  p-value: < 2.2e-16
```

F test for Overall Significance: Figure 11 shows the F test results, which shows that the model is overall significant. The p-value calculated by the F statistic is very less which makes the model significant.

```
SSTotal <- var( eth_data$Price ) * (nrow(eth_data)-1)
SSE     <- sum( eth_Mlm$resid^2 )
SSreg   <- SSTotal - SSE
dfE   <- eth_Mlm$df.residual
dfReg <- nrow(eth_data) - 1 - dfE
MSreg <- SSreg / dfReg
MSE   <- SSE / dfE
(Fstat <- MSreg / MSE) # F statistics of the test
```

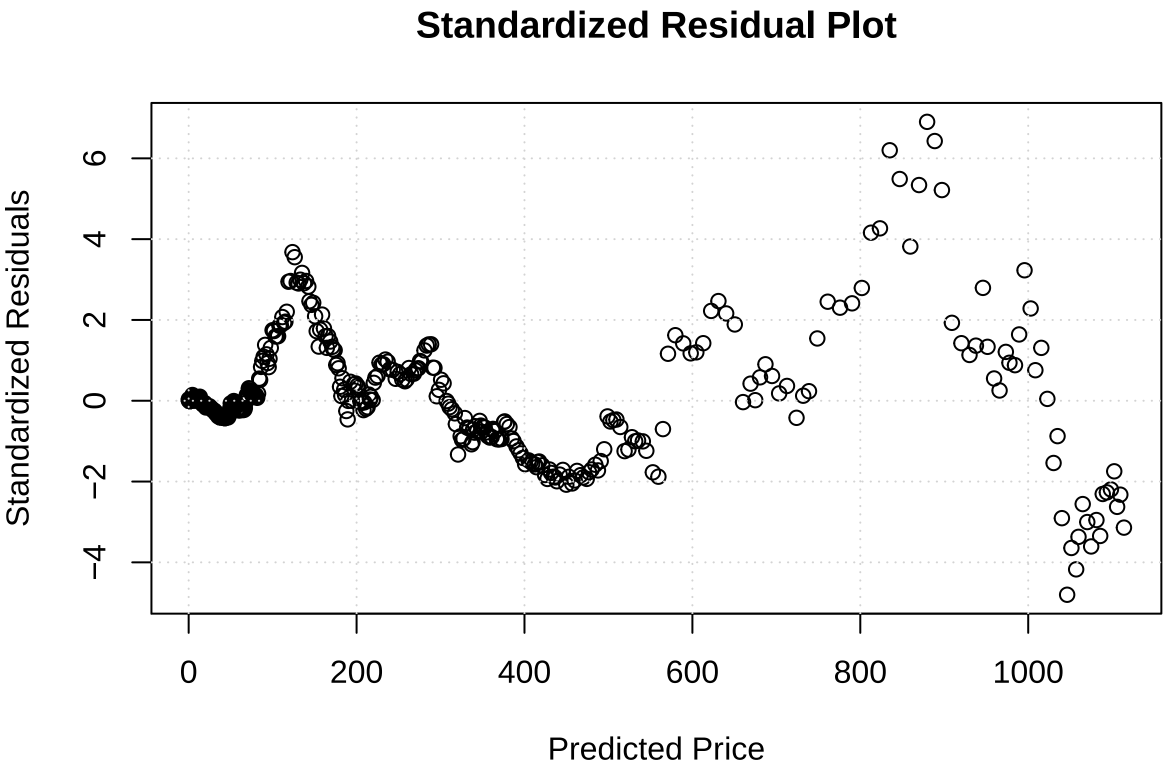
```
## [1] 5339.853
```

```
(pval  <- pf( Fstat , dfReg, dfE , lower.tail=FALSE )) # Pvalue for the test
```

```
## [1] 0
```

Residual Analysis : Again we see the error term assumptions being voilated , non constant variance

```
std_residuals = rstandard(eth_Mlm)
resd = eth_Mlm$residuals
plot(eth_Mlm$fitted.values,std_residuals, main="Standardized Residual Plot", ylab = "Standardized Residu
```



```
## integer(0)
```

Autocorrelation

When autocorrelation is present, one of the regression assumptions is violated: the error terms are not independent. In this case, serious errors can be made in performing tests of significance based upon the assumed regression model. The Durbin-Watson statistic can be used to detect first-order autocorrelation. First order means that the error term in each period depends on the error term in the previous period. Auto correlation often can be visualized by plotting the residuals of regression equation over time. Figure 14 shows the result of the Durbin Watson test. The p-value < 2.2e-16 suggests that the alternative hypothesis is correct. There is strong evidence that the data is autocorrelated.

```
library(lmtest)
```

```
## Warning: package 'lmtest' was built under R version 3.4.2
```

```
## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 3.4.2
```

```
##
## Attaching package: 'zoo'
```

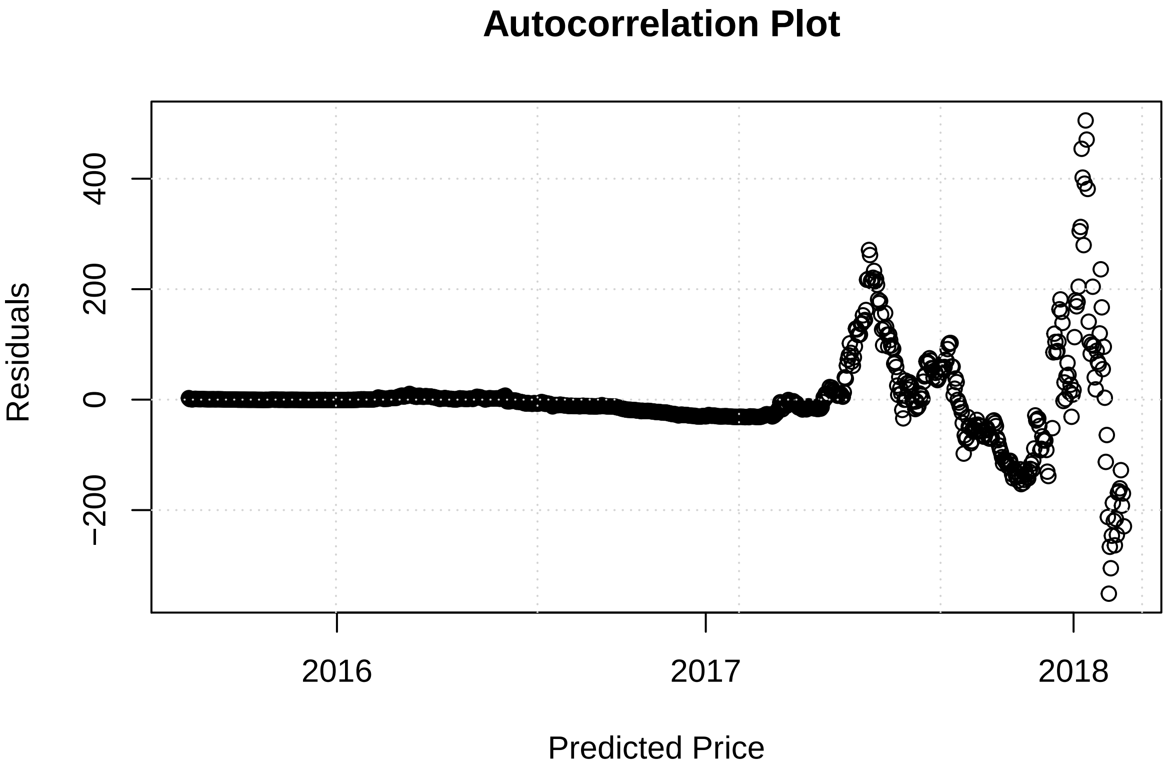
```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
(autocorrelation = dwtest(Price ~ eth_hashrate + eth_address , alternative = "two.sided", data = eth_da
```

```
##
##  Durbin-Watson test
```

```
##
## data:  Price ~ eth_hashrate + eth_address
## DW = 0.090995, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is not 0
```

```
plot(eth_data$Date,resd, main=" Autocorrelation Plot", ylab = " Residuals", xlab = "Predicted Price") +
```



```
## integer(0)
```