Project 1 - Predictor Exploration

Group: Medha Dhir, Abdul R Latif, Tamara Hahn , Lisheng Zou

Data Clean-Up

```
library(readr)
library(faraway)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.4.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
bit_data = read_csv('bitcoin.csv')
```

```
## Rows: 2920 Columns: 24
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (23): btc_market_price, btc_total_bitcoins, btc_market_cap, btc_trade_v...
## dttm  (1): Date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
price_0 = bit_data[bit_data[2]==0,]
bit_data = bit_data[bit_data[2]!=0,] # 2745 non zeros for the response variable
na_rows = which(is.na(bit_data))
clean=na.omit(bit_data)
```

We removed all 0observations that have a zero for the response variable, the bitcoin price, as those represent incomplete data. Another possibility would have been to interpolate them, and while that is a valid option, not using the data is the safest. Note that in the table, the bitcoin price is called "btc_market_price".

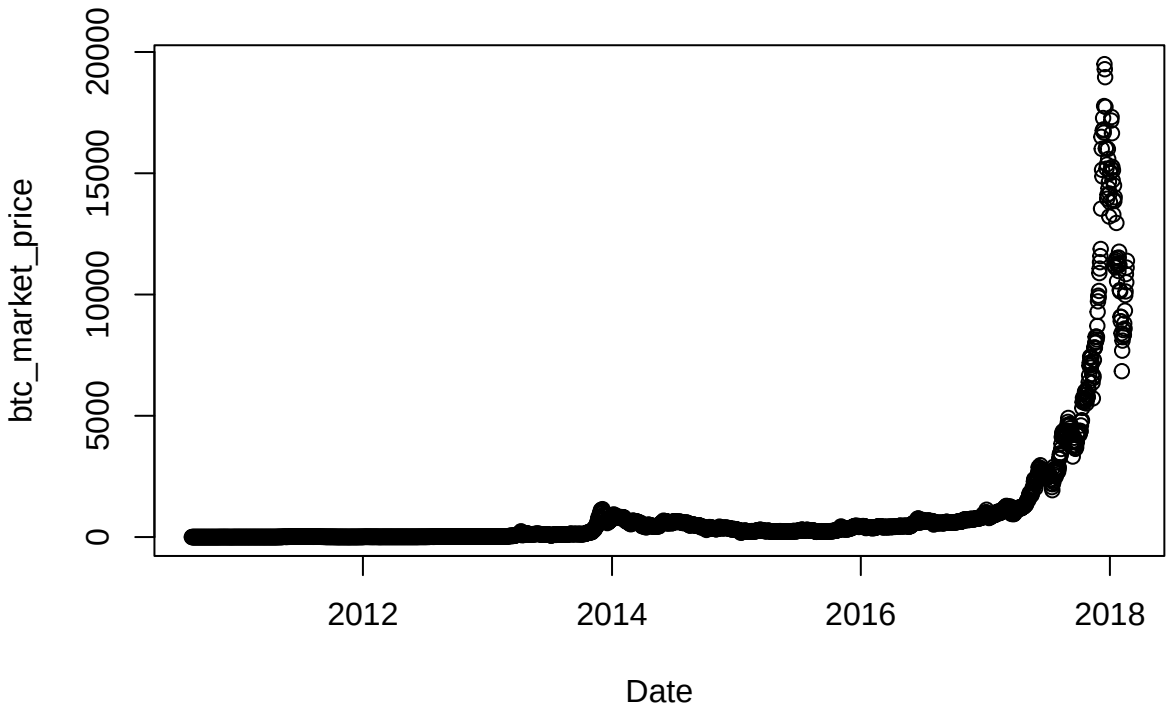
To check, uncomment the command below to see a summary of all 24 variables. The btc_market_price no longer shows a minimum of 0.

This is also a good place to do a common-sense check of whether any of the potential predictor variables show odd values.

```
# summary(clean)
```

For a quick sense of btc_market_price over time, here is a plot against the Date. (Note: That shape will appear again. Other plots of btc_market_price vs. other predictors show that similar shape that looks a bit exponential, and then drops towards the end like an upside down V).

```
plot(btc_market_price ~ Date, data=clean)
```

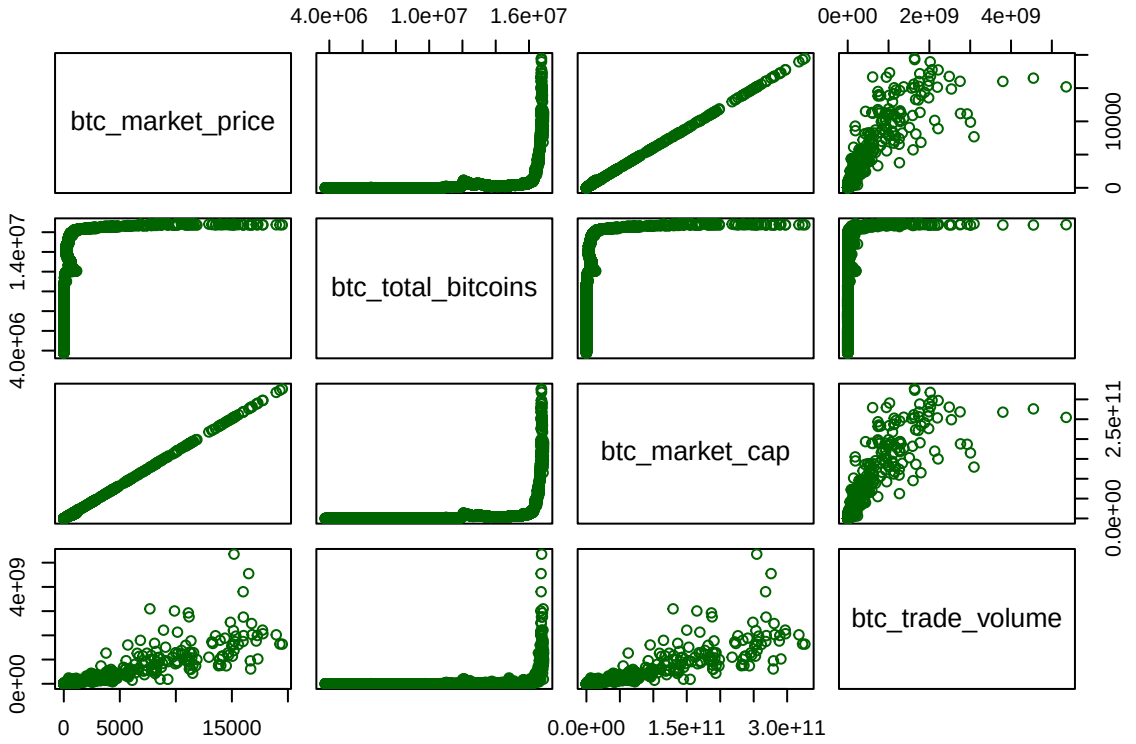


Relationships between Price and potential predictors

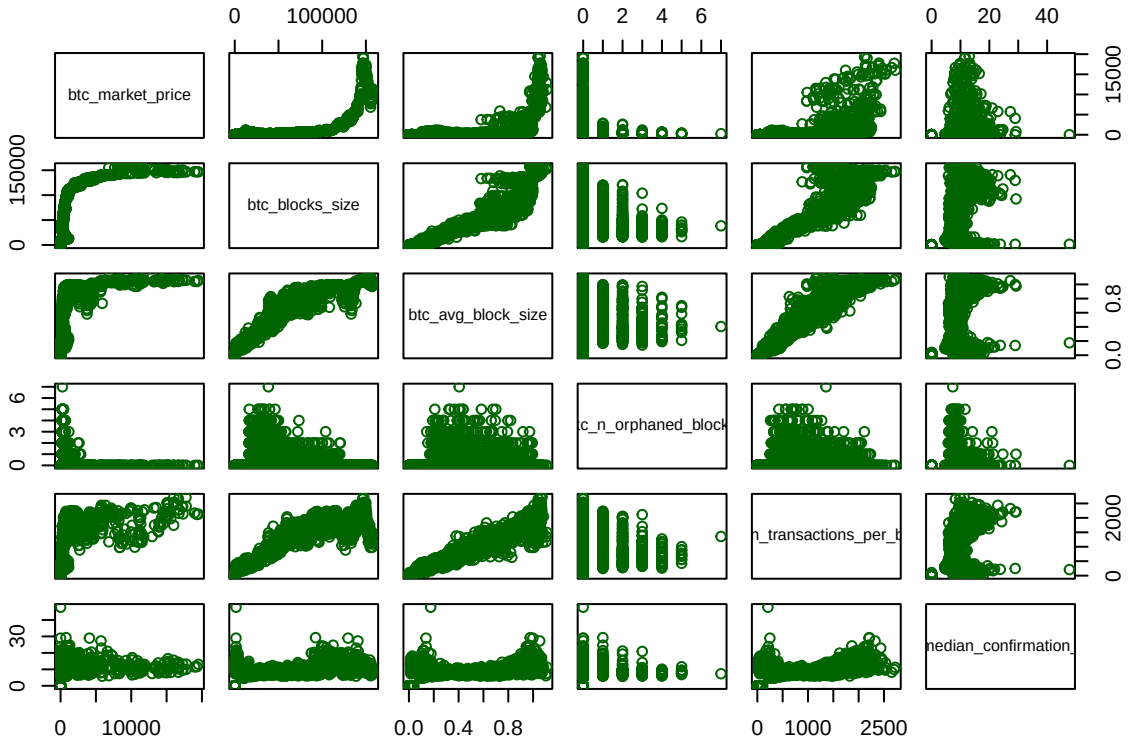
Now we will do plot pairs of btc_market_price and potential predictors in order. This will also allow us to look at some correlations between predictors. For space/readability reasons, these pair plots only plot 5-6 variables at a time, so not all predictors are plotted against each other.

The Excel Sheet has notes on which predictors to likely exclude even before we use a search algorithm or

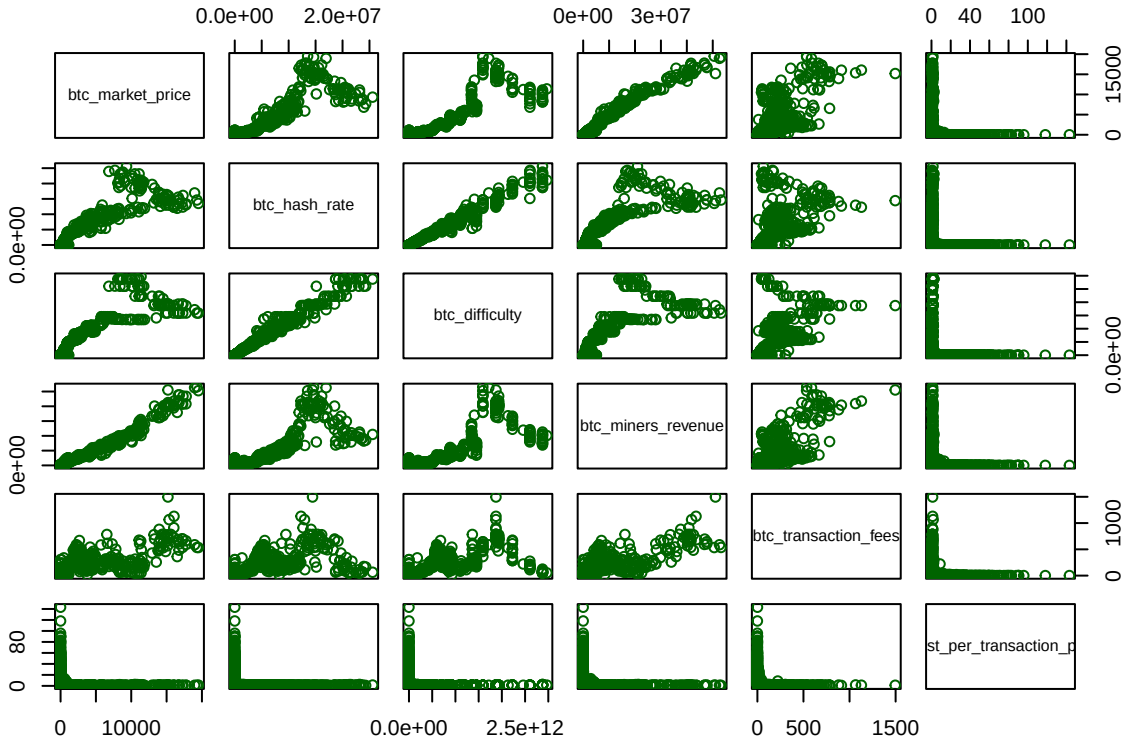
```
vars = names(clean)
pairs(clean[,vars[2:5]], col="darkgreen")
```



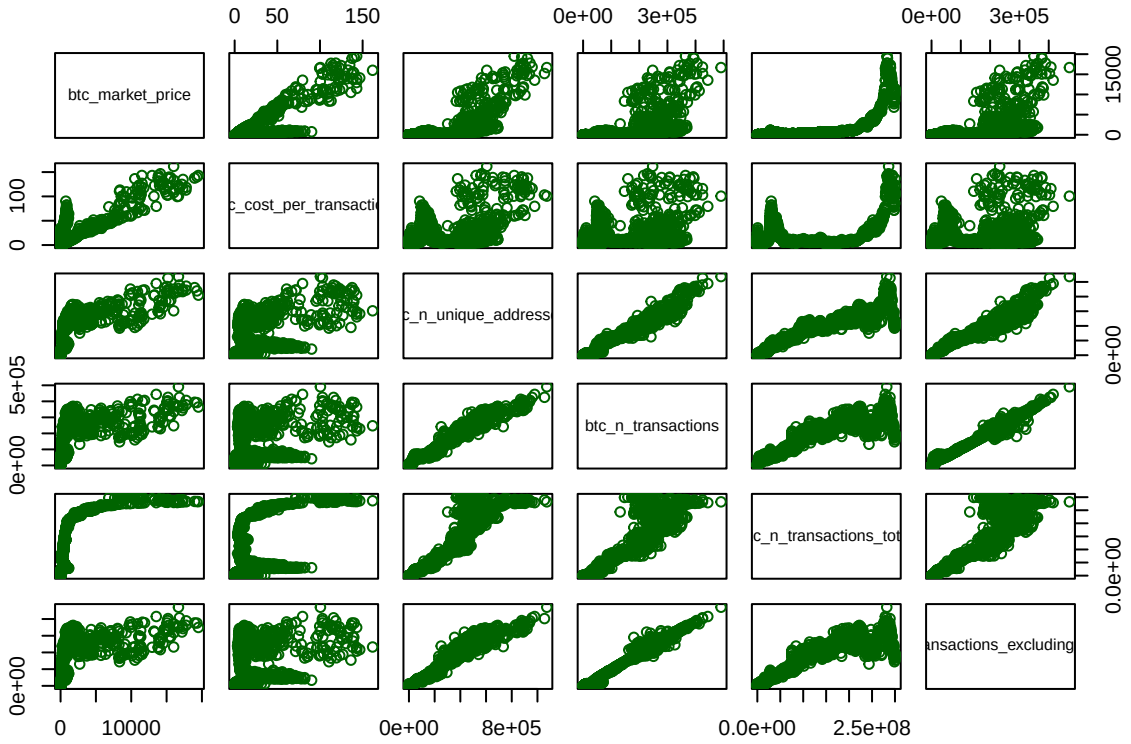

```
pairs(clean[,append(vars[2],vars[6:10])], col="darkgreen")
```



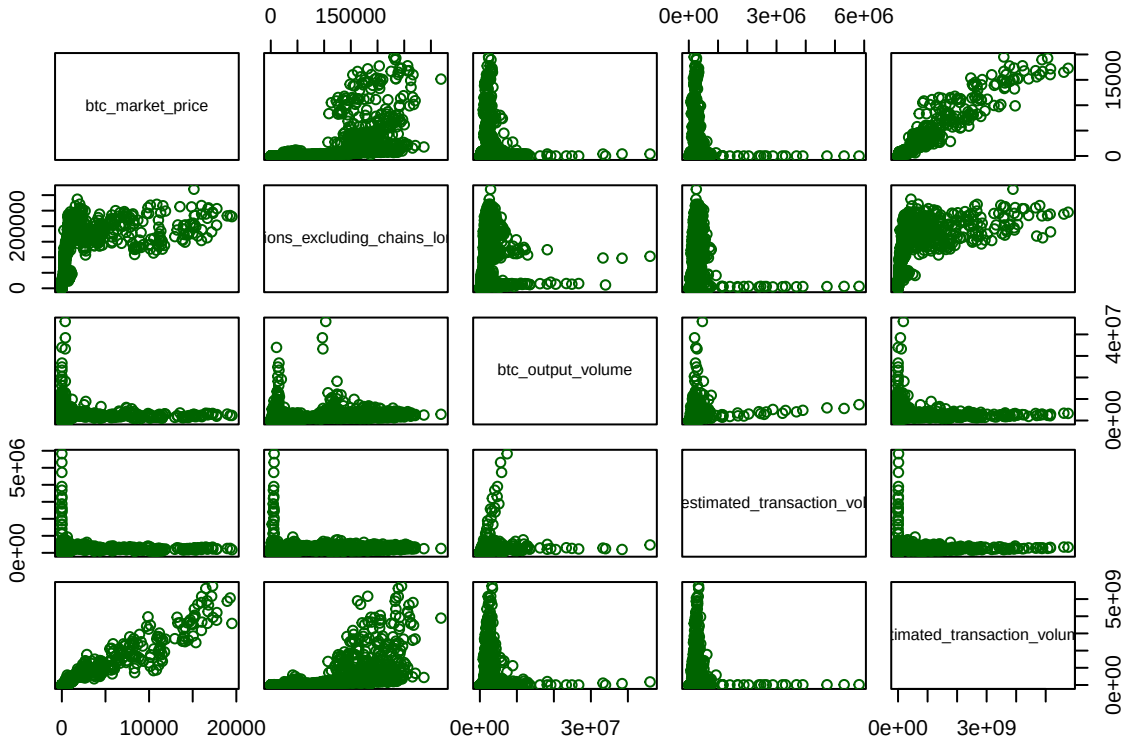

```
pairs(clean[,append(vars[2],vars[11:15])], col="darkgreen")
```




```
pairs(clean[,append(vars[2],vars[16:20])], col="darkgreen")
```




```
pairs(clean[,append(vars[2],vars[21:24])], col="darkgreen")
```



Based on the pair plots above, we identify predictors that we are likely to exclude because they do not seem to have a useful/informative relationship with the response variable btc_market_price.

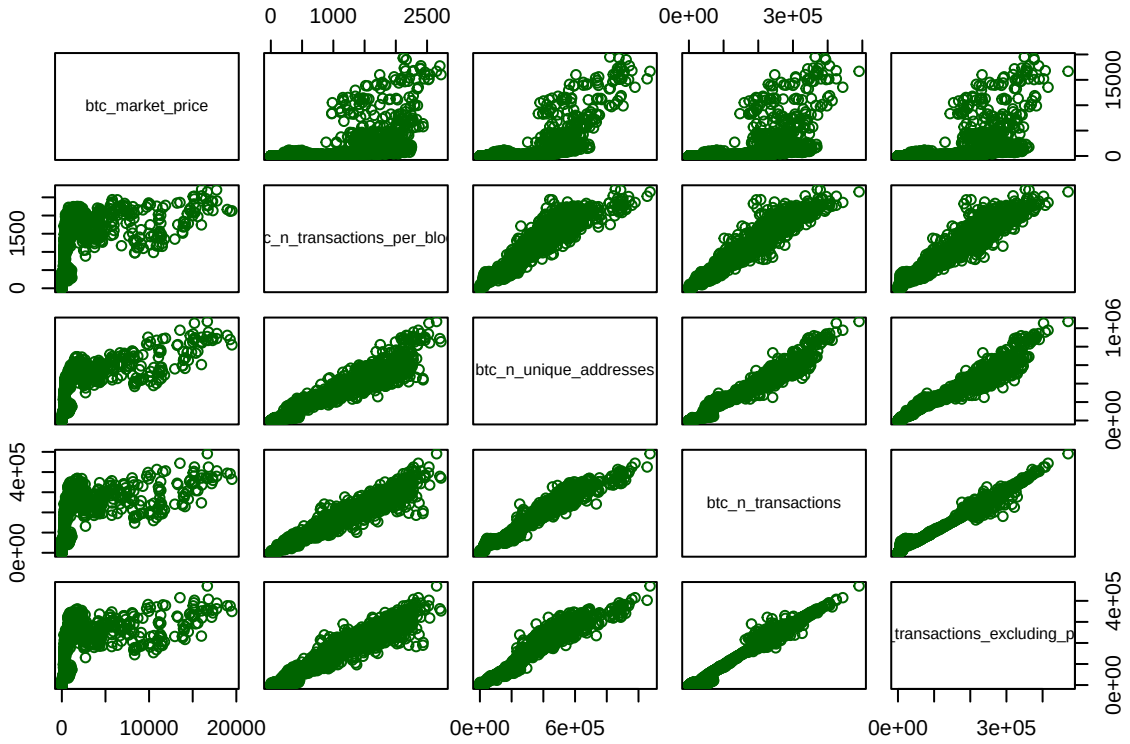
Category 0: Exclude potentially btc_n_transactions_per_block btc_median_confirmation_time btc_n_unique_addresses btc_n_transactions btc_n_transactions_excluding_popular

Category 1: Exclude definitely btc_cost_per_transaction_percent btc_cost_per_transaction btc_n_transactions_excludin btc_output_volume btc_estimated_transaction_volume_usd

As a visual Summary, here are again (isolated now) the pair plots of btc_market_price with the Category 0 and Category 1 variables to illustrate that their relationships are likely not informative for regression.

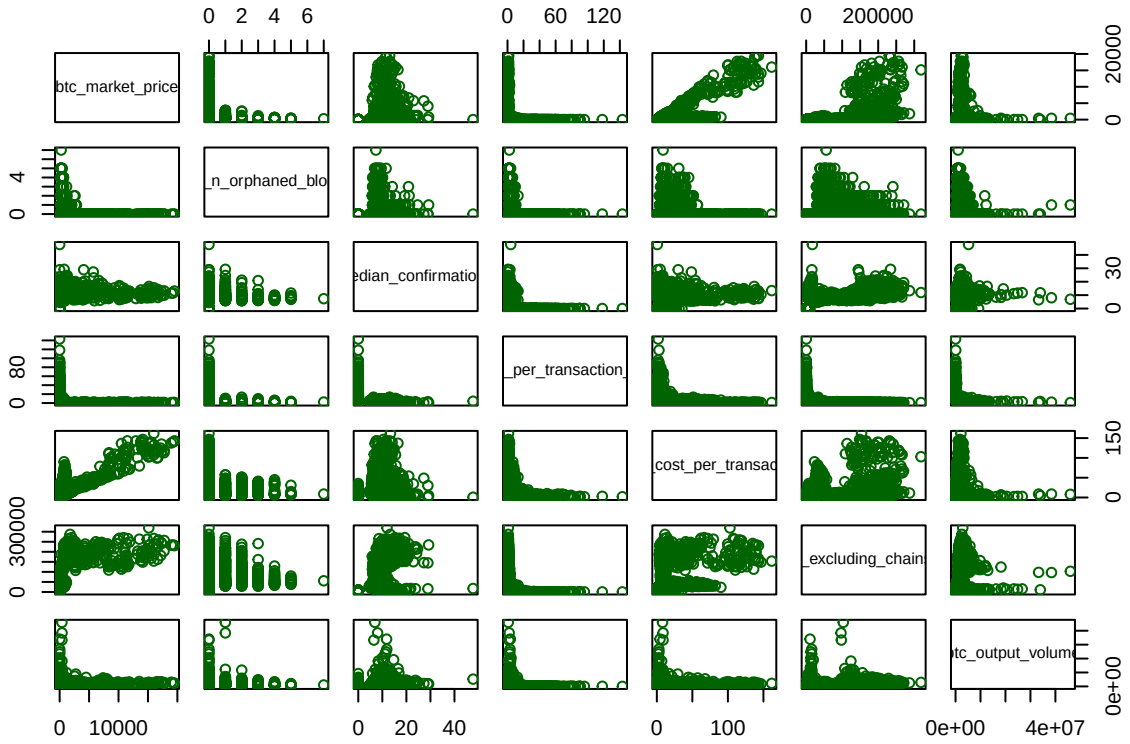
```
category_0 = c("btc_market_price","btc_n_transactions_per_block", "btc_n_unique_addresses", "btc_n_trans

pairs(clean[, (names(clean) %in% category_0)], col="darkgreen")
```




```
category_1 = c("btc_market_price","btc_cost_per_transaction_percent", "btc_cost_per_transaction", "btc_

pairs(clean[, (names(clean) %in% category_1)], col="darkgreen")
```



For convenience, the modeling will take place in the file Project_1_Models.Rmd