Crude Oil Price Predictor 2020

Christopher Page

8/15/2020

Introduction/Overview/Executive Summary

Oil is an important commodity. Some of our planet's inhabitants produce it. Many more consume it. All are affected by it. Because of its wide-ranging effects, oil commands collective attention enabled by analysis. One of the key metrics to take into account in the analysis of the oil indutry is the daily closing price of crude oil futures traded on a global basis in such venues as the New York Mercantile Exchange.

Methods for predicting that price may entail studying the quantitative and qualitative factors at play inside the oil industry as well as in such adjacent industries as transportation and manufacturing. They may also entail studying the often complex diplomatic, military, economic, cultural, and environmental developments that help explain why crude oil closing prices rise and fall as they do over time.

This project proposes a simpler method, one that predicts crude oil closing prices based on time and the closing prices of complementary (e.g., gasoline) and competing (e.g., platinum) commodities. The premise is that there is much to be learned by studying the behaviors of commodity traders engaged in making daily investment decisions based on running evaluations of risks and rewards.

That premise informed the development of algorithms with the potential to predict crude oil closing prices with a viable level of accuracy, quantified as the Root Means Square Error (RMSE). Using a data set constructed with publicly available information downloaded for free from https://www.nasdaq.com/, the author developed and then evaluated fifteen such algorithms.

The evaluation process led to the selection of one algorithm that produced the lowest RMSE when applied to a randomly generated test set. That algorithm used a Ranger model [1] which took into account the *year* and *month* components of time as well as the closing prices of four other globally traded commodities: (1) *heating oil*, (2) *gasoline*, (3) *platinum*, and (4) *soybeans*.

The RMSE in question was **2.93**, a figure equating to 12% of the 23.51 produced by an initial algorithm that only took into account the average closing price of crude oil across the period of observation. Because of the relatively small RMSE, the selected algorithm could be considered analytically viable and, therefore, of use to those engaged in analzing the oil industry.

Method/Analysis

Loading Libraries The first step was to load all of the necessary *R* libraries from the publicly available repository http://cran.us.r-project.org. Those libraries included *caret*, *caTools*, *dplyr*, *forcats*, *foreach*, *gam*, *ggplot2*, *ggrepel*, *ggthemes*, *knitr*, *lubridate*, *purrr*, *randomForest*, *ranger*, and *tidyverse*. The *randomForest* and *ranger* packages proved to be of particular value because of the models they introduced.

[1] Characterized by https://www.rdocumentation.org/ as a "fast implementation of random forests"

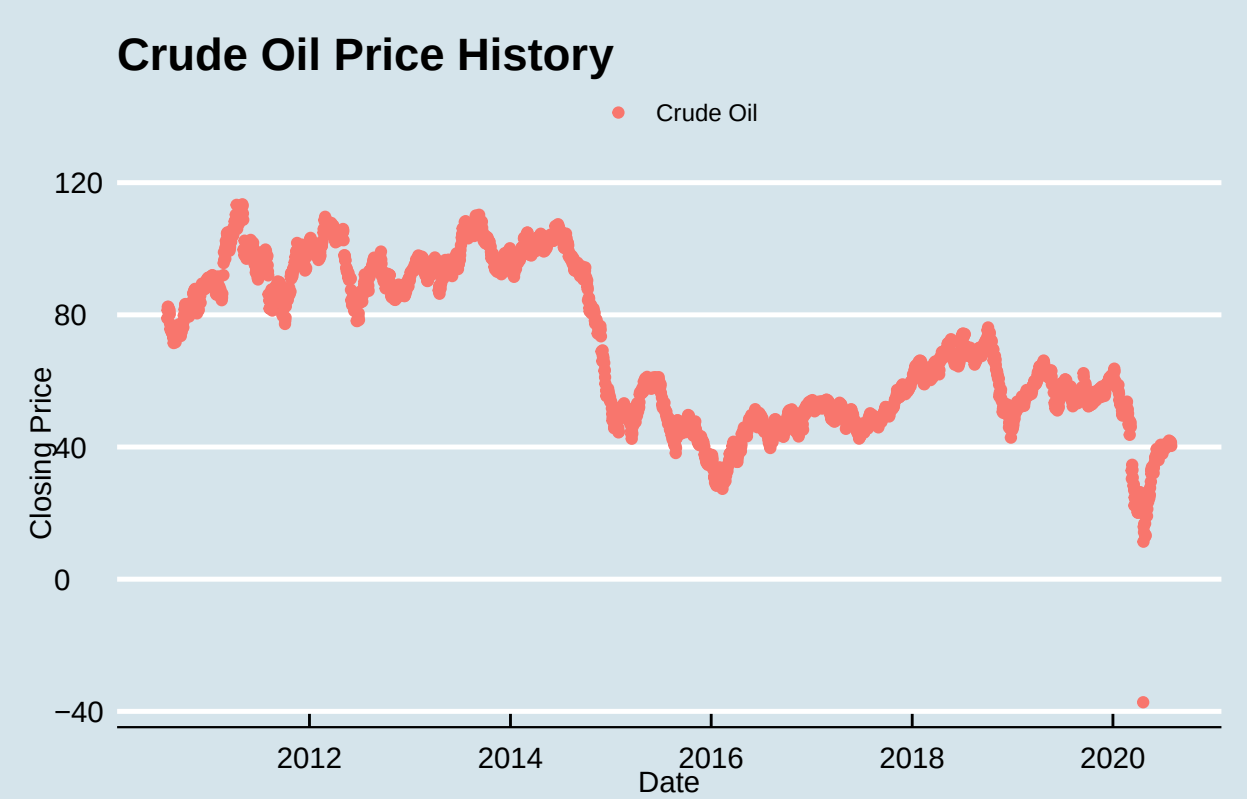
Loading and Wrangling Data The second step entailed loading the 25,192-row x 4-column *commodity_closing_prices_2010_2020.csv*, a file containing commodity closing prices for a period of observation extending from the beginning of August 2010 to the end of July 2020, and then wrangling the data in a manner that would result in ready access to information detailing the cyclical and linear components of time as well as the daily closing prices of crude oil and nine other commodities from the energy, precious metals, and agriculture sectors. The primary output of this step in the process was a 2,519-row x 19-column *crude_oil_in_commodites_market* data set with the following structure.

```
str(crude_oil_in_commodities_market)
```

```
## tibble [2,519 x 19] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ date                      : Date[1:2519], format: "2010-08-02" "2010-08-03" ...
##  $ commodity                 : chr [1:2519] "Crude Oil" "Crude Oil" "Crude Oil" "Crude Oil" ...
##  $ sector                    : chr [1:2519] "Energy" "Energy" "Energy" "Energy" ...
##  $ closing_price             : num [1:2519] 79 81.4 82.4 82.4 82.1 ...
##  $ date_year                 : num [1:2519] 2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...
##  $ date_quarter_of_the_year : int [1:2519] 3 3 3 3 3 3 3 3 3 3 ...
##  $ date_month                : Date[1:2519], format: "2010-08-01" "2010-08-01" ...
##  $ date_month_of_the_year    : num [1:2519] 8 8 8 8 8 8 8 8 8 8 ...
##  $ date_day                  : int [1:2519] 2 3 4 5 6 9 10 11 12 13 ...
##  $ date_weekday              : chr [1:2519] "Monday" "Tuesday" "Wednesday" "Thursday" ...
##  $ natural_gas_closing_price: num [1:2519] 1.8 1.83 1.85 1.8 1.73 1.81 1.79 1.68 1.68 1.64 ...
##  $ heating_oil_closing_price: num [1:2519] 2.15 2.2 2.2 2.19 2.15 2.15 2.13 2.08 2 2 ...
##  $ gasoline_closing_price   : num [1:2519] 2.17 2.19 2.18 2.16 2.11 2.12 2.09 2 1.95 1.94 ...
##  $ gold_closing_price        : num [1:2519] 1188 1196 1199 1205 1203 ...
##  $ silver_closing_price      : num [1:2519] 18.4 18.3 18.3 18.5 18.2 ...
##  $ platinum_closing_price    : num [1:2519] 1577 1587 1586 1572 1571 ...
##  $ wheat_closing_price       : num [1:2519] 688 693 724 712 722 ...
##  $ rice_closing_price        : num [1:2519] 11.1 11.2 11 11 11.2 ...
##  $ soybeans_closing_price    : num [1:2519] 1033 1012 1004 1004 1000 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Date = col_character(),
##   ..   Commodity = col_character(),
##   ..   Sector = col_character(),
##   ..   `Closing Price` = col_double()
##   .. )
```

Exploring Crude Oil in Isolation Wrangling enabled explotatory data analyses that began with an examination of crude oil in isolation across the ten-year period of observation. A plot generated through that examination made clear the history of crude oil closing prices as they rose and fell, often sharply, between the beginning of August 2010 and end of July 2020. Of particular note was the rapid descent deeply into negative territory in April 2020 [2] That descent helped account for prices having the following minimum, maximum, mean, and standard deviation.

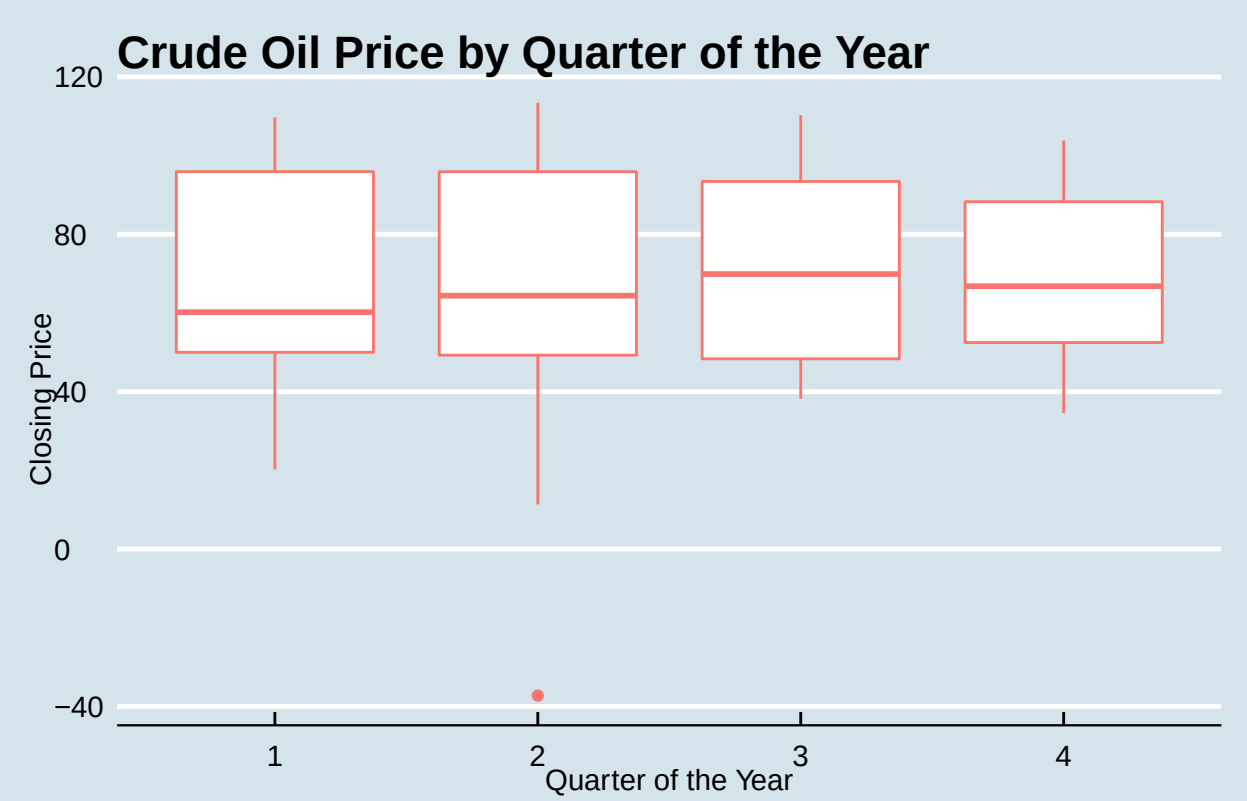
Parameters	Values
Mininum	-37.25000
Maximum	113.45000
Mean	70.13307
Standard Deviation	23.52209



Plot showing crude oil closing prices for the 2010−2020 period of observation

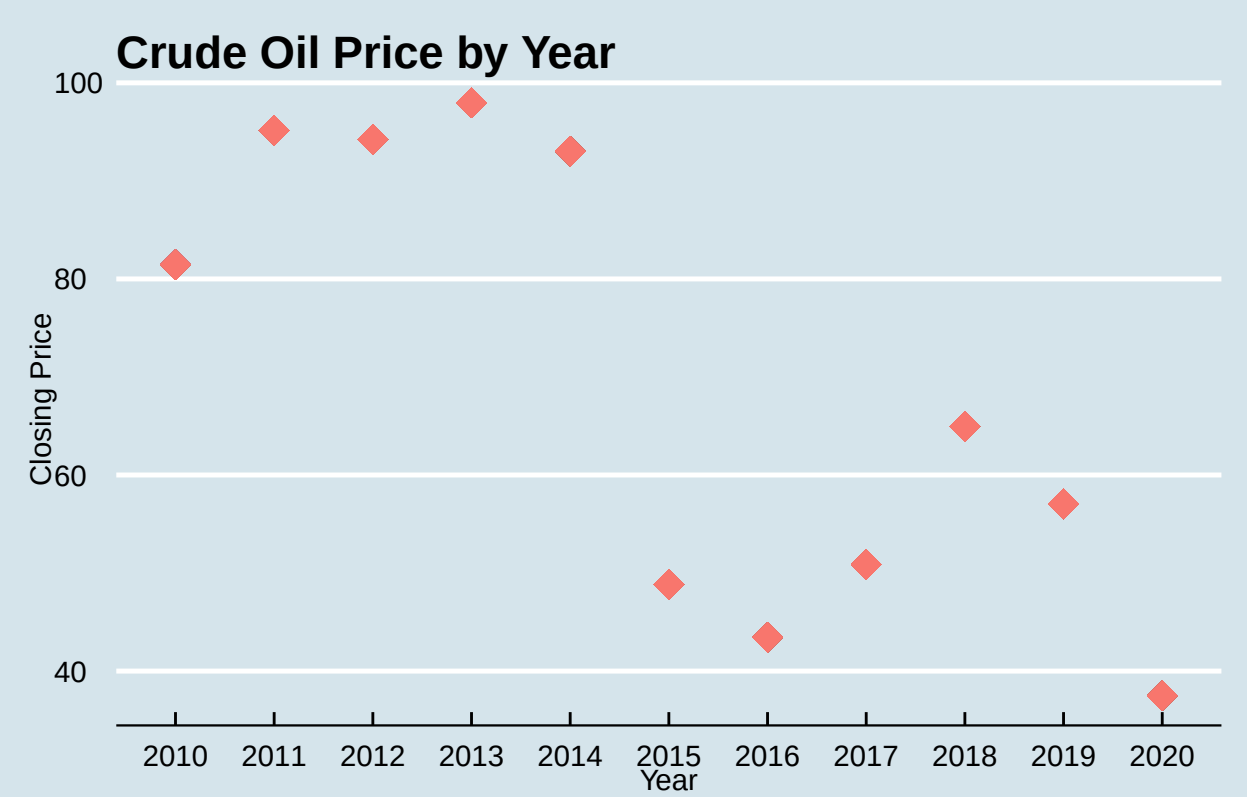
[2] See https://www.nytimes.com/2020/04/20/business/stock-market-live-trading-coronavirus.html for an example of news reports covering that event, widely characterized as "unprecedented".

The next step in the exploration process was to examine crude oil's 2010-2020 closing price history in terms of the cyclical components of time, namely the *day of the trading week* (Monday, Tuesday,. . . ,Friday) as well as the *month of the year* (1, 2,. . . , 12) and the *quarter of the year* (1, 2,.., 4). The hope was that one or more of those components would emerge as a clear differentiator and, thus, a promising predictor. The results, exemplified by the following graph of crude oil closing prices by *quarter of the year*, revealed none to be promising. Those results argued against such suggestions as "prices tend to be higher on the final day of the trading week" or "prices tend to be lower during the opening quarter of the year."



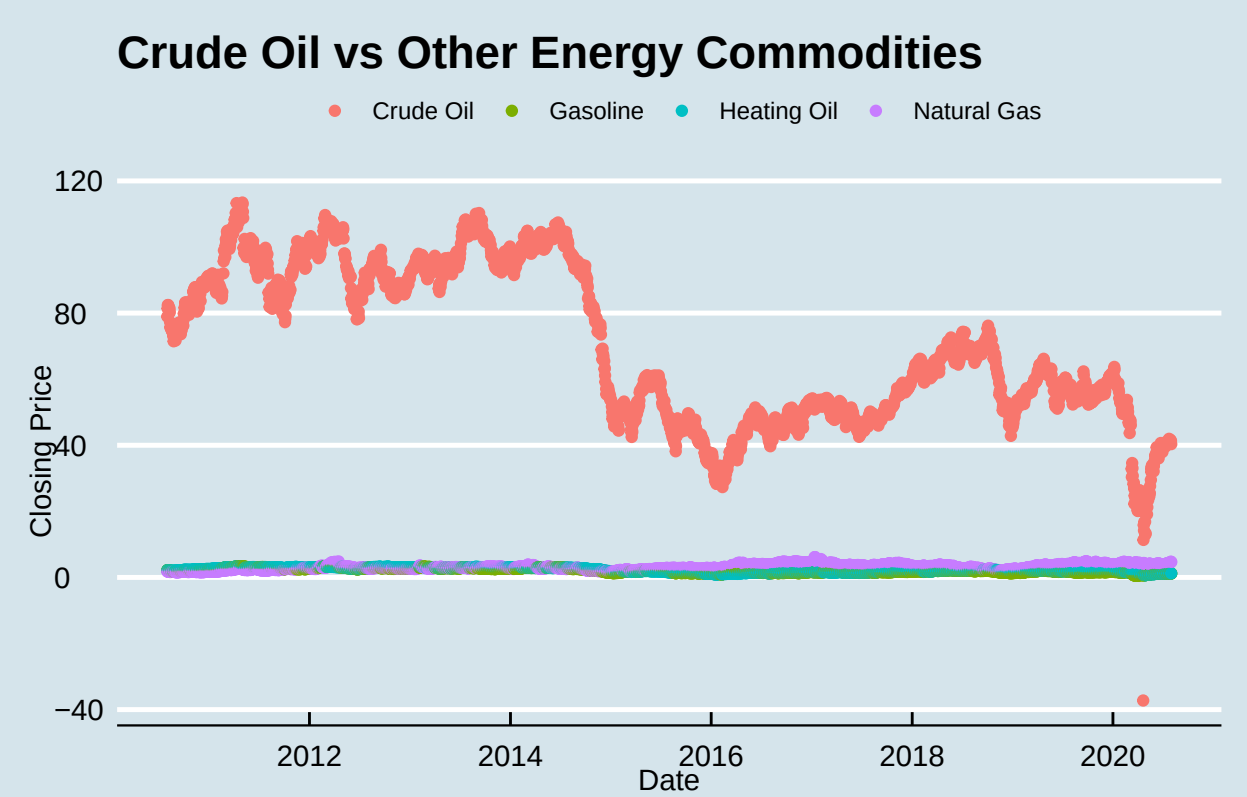
Boxplots showing breakout of crude oil closing prices by quarter of the year

Unlike the cyclical components of time, the linear components, namely the *year* and *month* of each observation, proved to be very promising. As the following plot illustrates, the results of exploratory data analysis efforts in this area confirmed that *year* was a clear differentiator and, consequently, a potentially useful predictor of crude oil closing prices. At this point in the process, there was reason to believe *year* and, even more significantly, *month* would feature prominently in the stronger of the algorithms to be developed and subsequently evaluated.



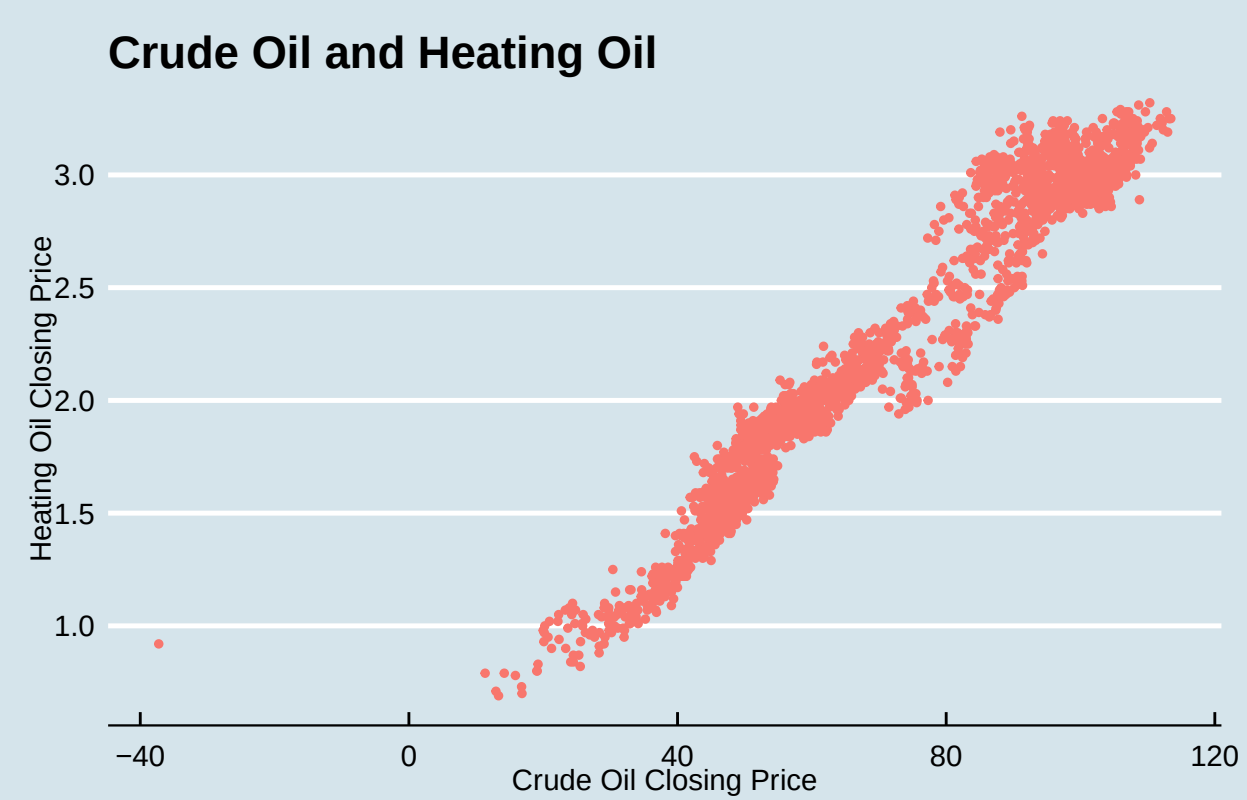
Plot showing breakout of average crude oil closing prices by year for each year in 2010–2020

Exploring Crude Oil in Comparison to other Energy Sector Commodities While the linear components of time were interesting, they weren't in and of themselves necessarily compelling. For that reason, attention turned next to examining crude oil closing price trends for the 2010-2020 period of observation in comparison to closing price trends for three complementary commodities from the energy sector. Those three complementary commodities were *heating oil*, *natural gas*, and *gasoline*. Per the following plot, their price histories were nowhere near as volatile as the price history of crude oil.



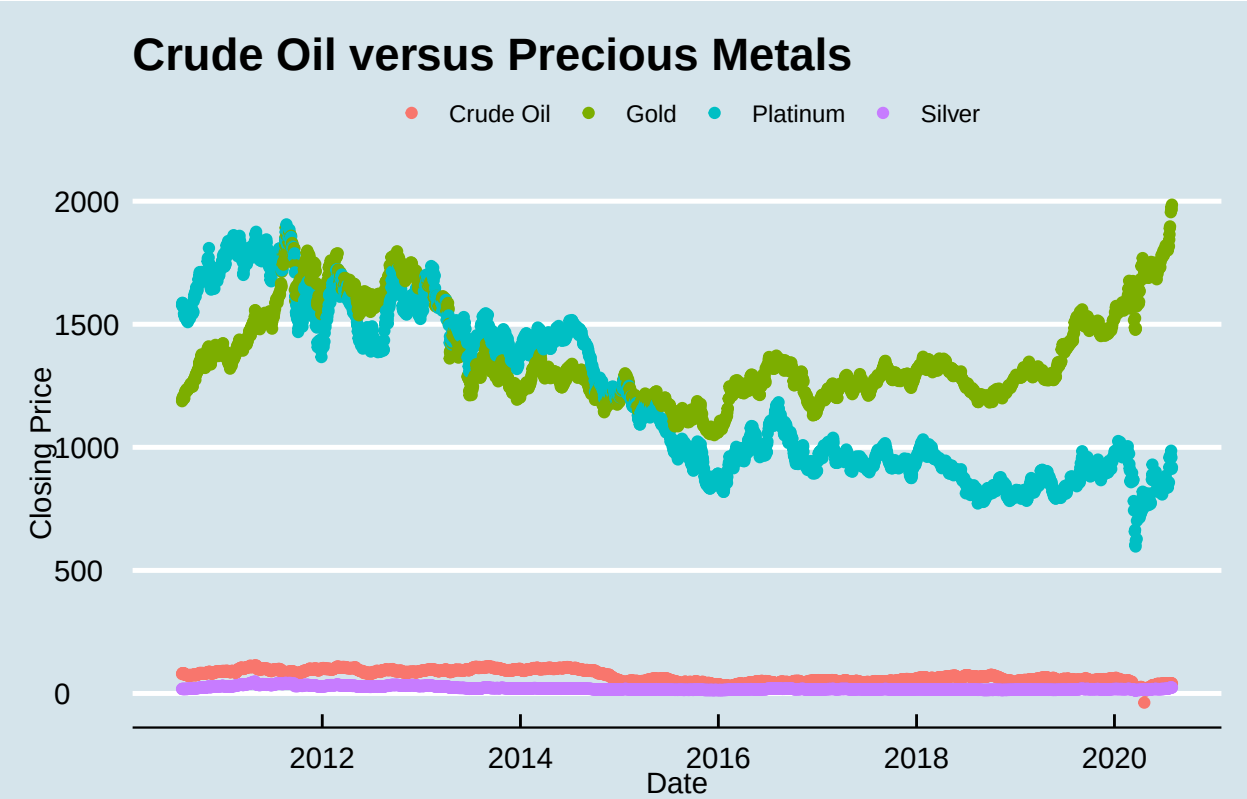
Plot comparing crude oil closing prices to those of three other commodities from the energy sector

Although they weren't as volatile as crude oil's, the daily closing prices of the other energy sector commodities were still interesting because of their correlations to the primary commodity being studied. As exemplified by the following diagram, *heating oil* and *gasoline* had the clearest correlation, presumably due to the downstream relationships those refined products have to crude oil [3]. Keeping in mind those relationships and the consequent correlations, there was reason to believe that *heating oil* and *gasoline* would, much like *year* and *month*, emerge as useful predictors of crude oil closing prices.



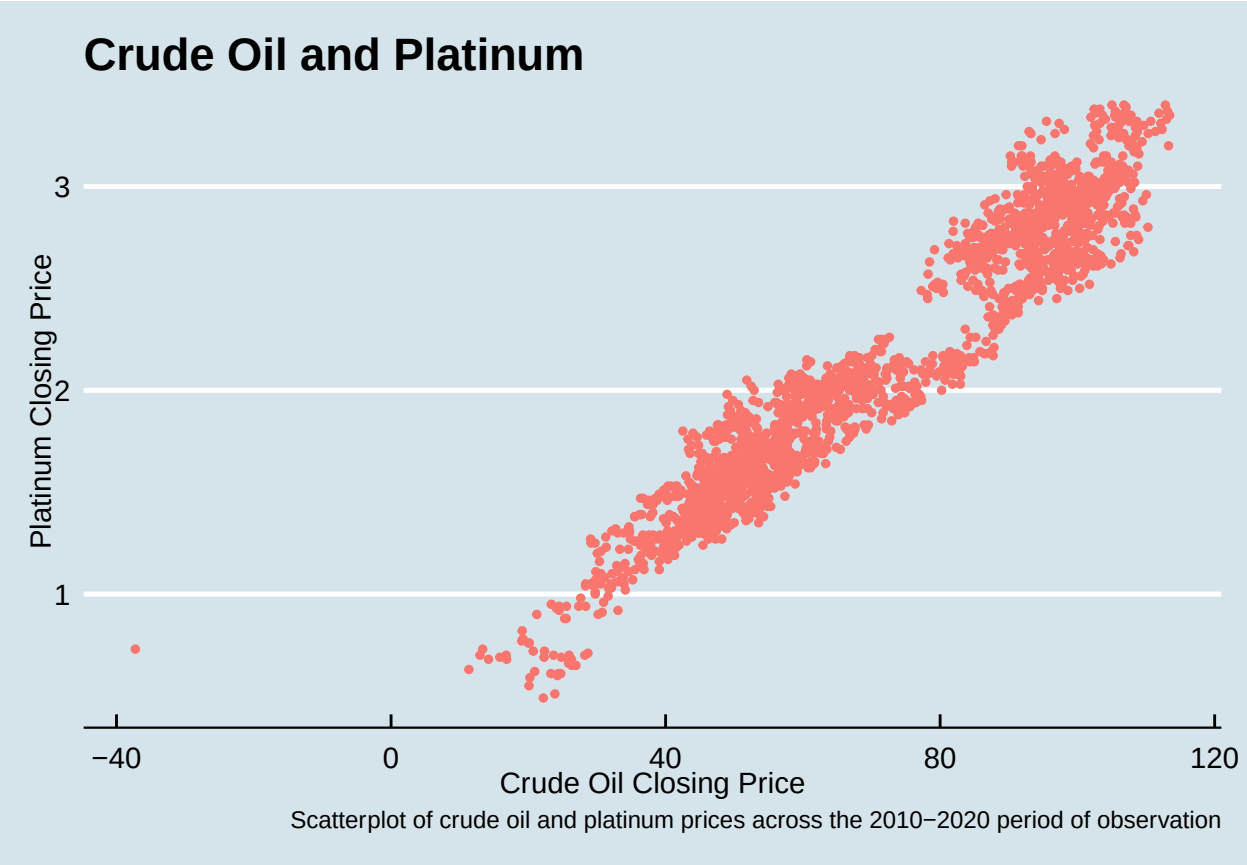
[3]See https://www.eia.gov/energyexplained/oil-and-petroleum-products/ for more information about the relationship between crude oil and the downstream products that emerge from the refining of that commodity.

Exploring Crude Oil in Comparison to Precious Metals Commodities Attention then turned to examining crude oil closing price trends for the 2010-2020 period of observation in comparison to closing price trends for three commodities in the precious metals sectors. Those three competing commodities were *gold*, *silver*, and *platinum*. Per the following plot, the price histories of the precious metals had, particularly in the cases of *gold* and *platinum*, volatilities similar to crude oil's.



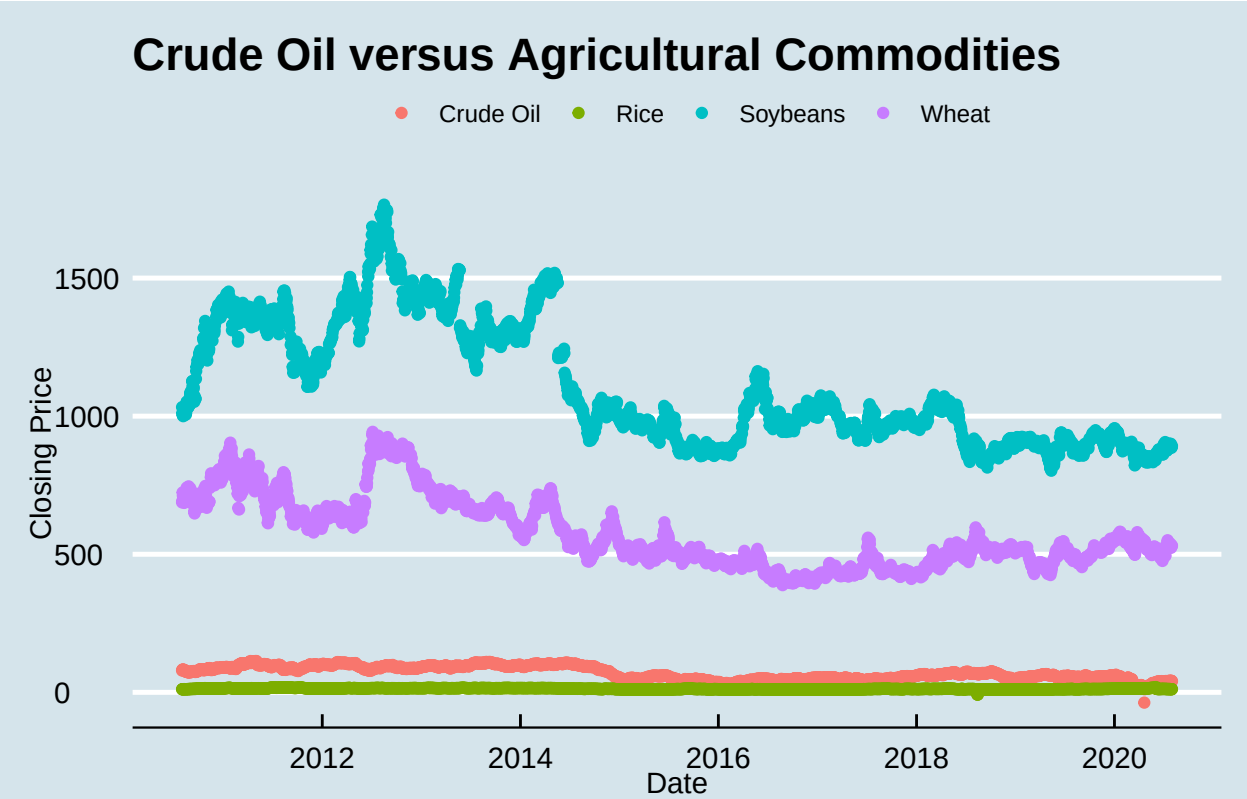
Plot comparing crude oil closing prices to those of three commodities from the precious metals sector

A deeper examination revealed the *platinum* had the clearest correlation. That revelation gave rise to the assumption that *platinum* would, much lile *heating oil* and *gasoline* as well as *year* and *month*, prove to be a useful predictor when the time came to develop and evaluate algorithms for predicting crude oil closing prices.



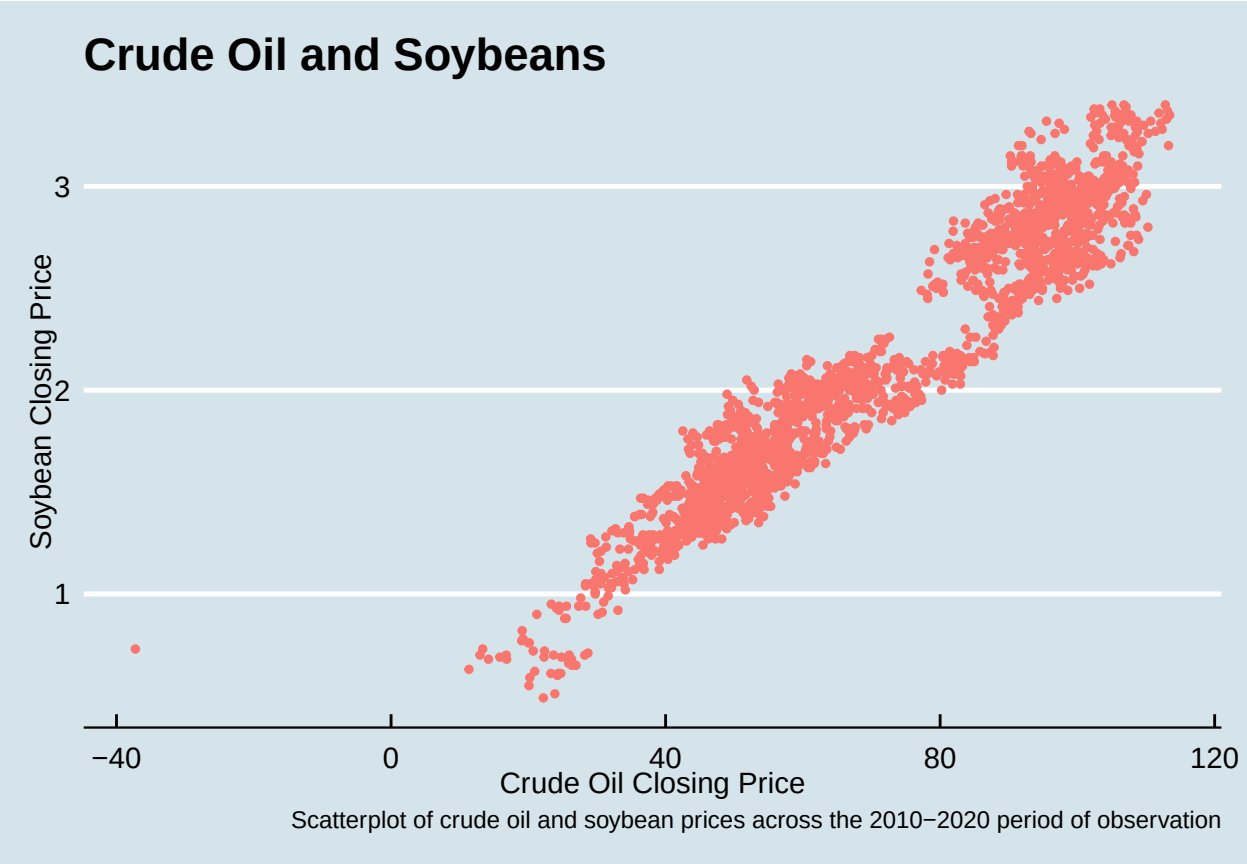
Scatterplot of crude oil and platinum prices across the 2010−2020 period of observation

Exploring Crude Oil in Comparison to Agriculture Commodities As a final step in the process of exploratory data analysis, attention shifted to examining crude oil closing price trends for the 2010-2020 period of observation in comparison to closing price trends for three commodities in the agriculture sectors. Those three competing commodities were *wheat*, *rice*, and *soybeans*. Per the following plot, the price histories of the agriculture also had, particularly in the cases of *soybeans* and *wheat*, volatilities similar to crude oil's.



Plot comparing crude oil closing prices to those of three commodities from the agriculture sector

A deeper examination revealed that *soybeans* had the closest correlation to *crude oil*. That revelation resulted in the addition of *soybeans* to a promising predictors list that already included *platinum*, *heating oil*, *gasoline*, *year*, and *month*. The assumption was those those predictors would feature prominently in the strongest of the algorithms to be developed and evaluated. Thanks to the exploratory data analysis, the path to an analytially viable prediction algorithm had become clearer.



Generating training and testing sets from the commodities data After completing the exploratory data analysis, the focus of the project turned to the generation of the training and testing sets. Eighty percent of *crude_oil_in_commodites_market* was randomly selected to be in *crude_oil_train*. The remaining twenty percent was held-back to be *crude_oil_test*.

```
set.seed(755)

test_index <- createDataPartition(y = crude_oil_in_commodities_market$commodity,
                                  times = 1, p = 0.2, list = FALSE)

crude_oil_train <- crude_oil_in_commodities_market[-test_index,]

crude_oil_test <- crude_oil_in_commodities_market[test_index,]
```

Algorithm 01 (Average) The next step involved developing and then evaluating the prediction algorithms, the first of which predicted crude oil closing prices based solely on the average crude oil price for the 2010-2020 period of observation. It would serve as a baseline from which to compare the analytical viability of other, more sophisticated algorithms. Not surprisingly, it had an RMSE that closely approximated the standard deviation of the original data set.

```
mu <- mean(crude_oil_train$closing_price)

predicted_price_algorithm_01 <- mu

RMSE01 <- RMSE(predicted_price_algorithm_01,
               crude_oil_test$closing_price)

RMSE01
```

```
## [1] 23.51292
```

Algorithm 02 (Year Effects) Incorporating *year*, the first of the two linear time components identified as promising through exploratotary data analysis, the second algorithm generated an RMSE that was signficantly smaller and, therefore, more analytically viable. That second algorithm relied not on the overall average but the yearly averages for each year in the 2010-2020 period of observation.

```
crude_oil_average_by_year <- crude_oil_train %>%
  group_by(date_year) %>%
  summarize(b_y = mean(closing_price))

predicted_price_algorithm_02 <- crude_oil_test %>%
  left_join(crude_oil_average_by_year,
            by= 'date_year') %>%
  pull(b_y)

RMSE02 <- RMSE(predicted_price_algorithm_02,
               crude_oil_test$closing_price)

RMSE02
```

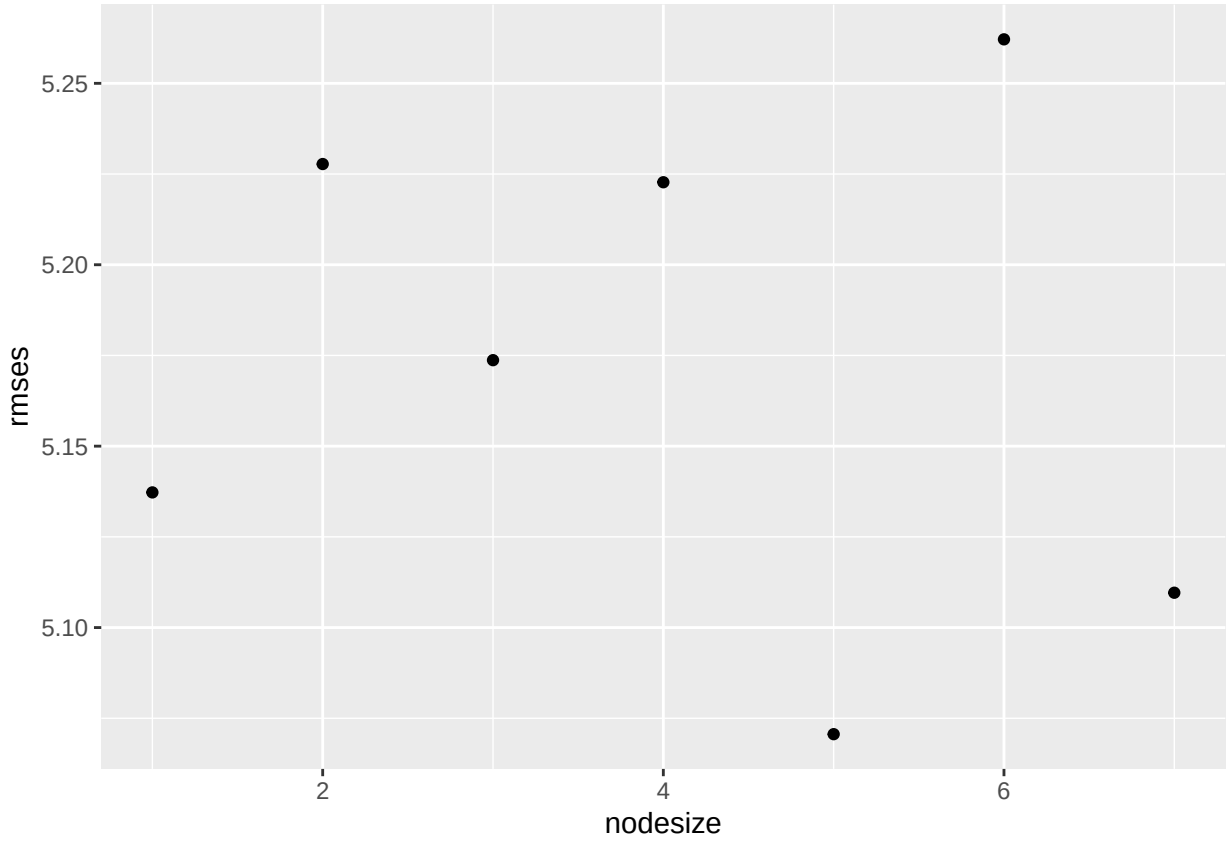
```
## [1] 8.268093
```

Algorithm 03 (Month Effects) Incorporating *month*, the second of the two linear time components identified as promising through exploratotary data analysis, the third algorithm generated an RMSE that was even smaller and, therefore, more analytically viable. That third algorithm relied not on the monthly averages for each of the months in the 2010-2020 period of observation.

```
crude_oil_average_by_month <-
  crude_oil_train %>%
  group_by(date_month) %>%
  summarize(b_m = mean(closing_price))

predicted_price_algorithm_03 <-
  crude_oil_test %>%
  left_join(crude_oil_average_by_month,
            by= 'date_month') %>%
  mutate(pred = b_m) %>%
  pull(pred)

RMSE03 <- RMSE(predicted_price_algorithm_03,
               crude_oil_test$closing_price)
```

Algorithm 04 (Random Forest - Time) While the RMSE from the third algorithm was good, it still wasn't good enough to be of great use to analysts studying the oil industry. With the goal of generating a much more robust outcome, attention shifted to the use of a Random Forest that first incoporating all of the cyclical and linear components of time before zeroing-in on the most significant. Those most significant of those components were, as anticipated, *year* and *month*. The RMSE generated through a Random Forest incorporating *year* and *month* was small but, again, not small enough to be analytically valuable

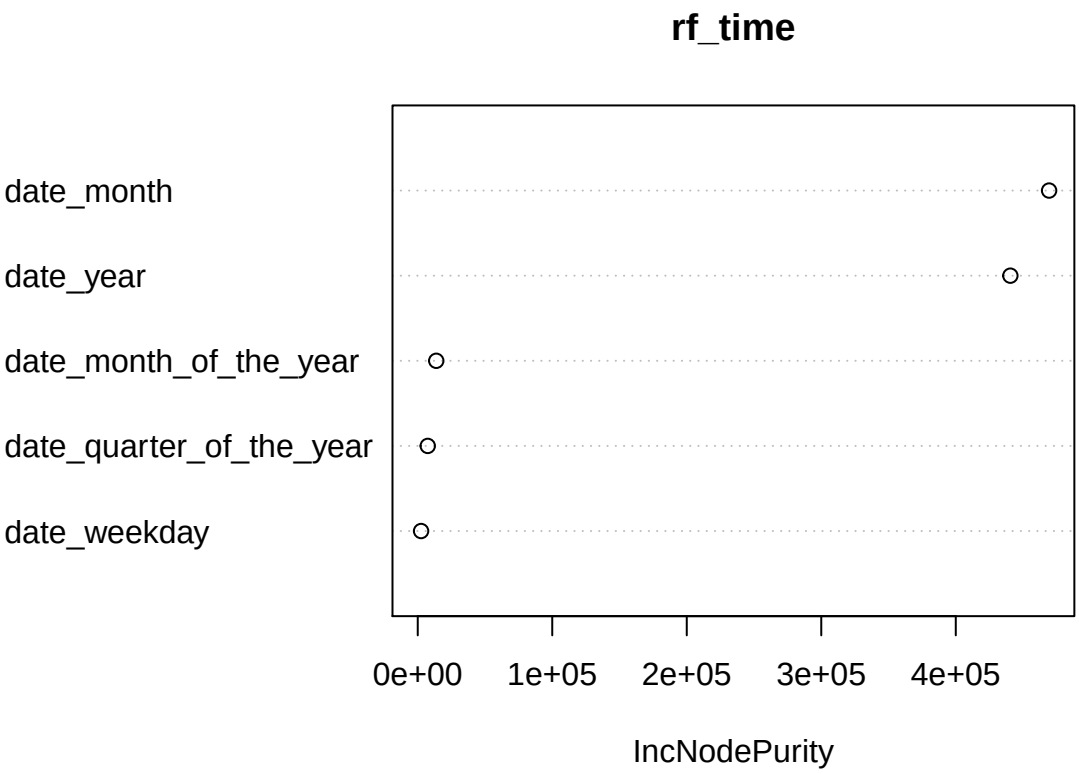
```
## Look at a Random Forest incorporating all components of time

nodesize <- seq(1, 7, 1)

rmses <- sapply(nodesize, function(n){
  rf_time = randomForest(closing_price ~
                           date_weekday +
                           date_month_of_the_year +
                           date_quarter_of_the_year +
                           date_year +
                           date_month,
                         data = crude_oil_train,
                         nodesize = n)
  pred <- predict(rf_time,
                  newdata = crude_oil_train)
  RMSE(pred,
       crude_oil_train$closing_price)
})

## Build a qplot of the RMSE results generated for each of the evaluated node sizes

qplot(nodesize, rmses)
```



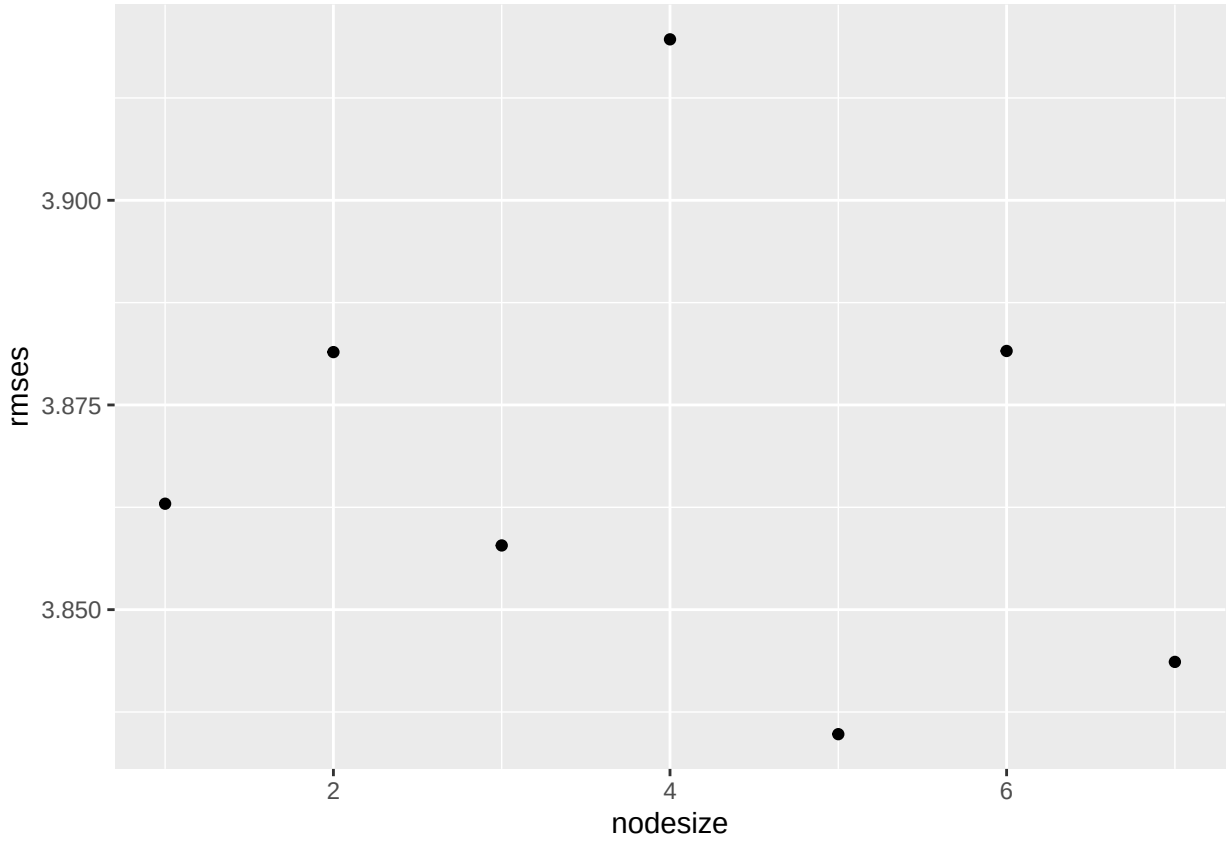

```
## Run the Random Forest model with the node size that generated the smallest RMSE

nodesize[which.min(rmses)]
```

```
## [1] 5
```

```
rf_time = randomForest(closing_price ~
                         date_weekday +
                         date_month_of_the_year +
                         date_quarter_of_the_year +
                         date_year +
                         date_month,
                       data = crude_oil_train,
                       nodesize = nodesize[which.min(rmses)])

## Build plot to help determine which components is/are the most important predictors

varImpPlot(rf_time)
```




```
## Based on that plot, revise Random Forest to focus on the month and the year

nodesize <- seq(1, 7, 1)

rmses <- sapply(nodesize, function(n){
  rf_time = randomForest(closing_price ~
                           date_month +
                           date_year,
                         data = crude_oil_train,
                         nodesize = n)
  pred <- predict(rf_time,
                  newdata = crude_oil_train)
  RMSE(pred,
       crude_oil_train$closing_price)
})

qplot(nodesize, rmses)
```




```
nodesize[which.min(rmses)]
```

```
## [1] 5
```

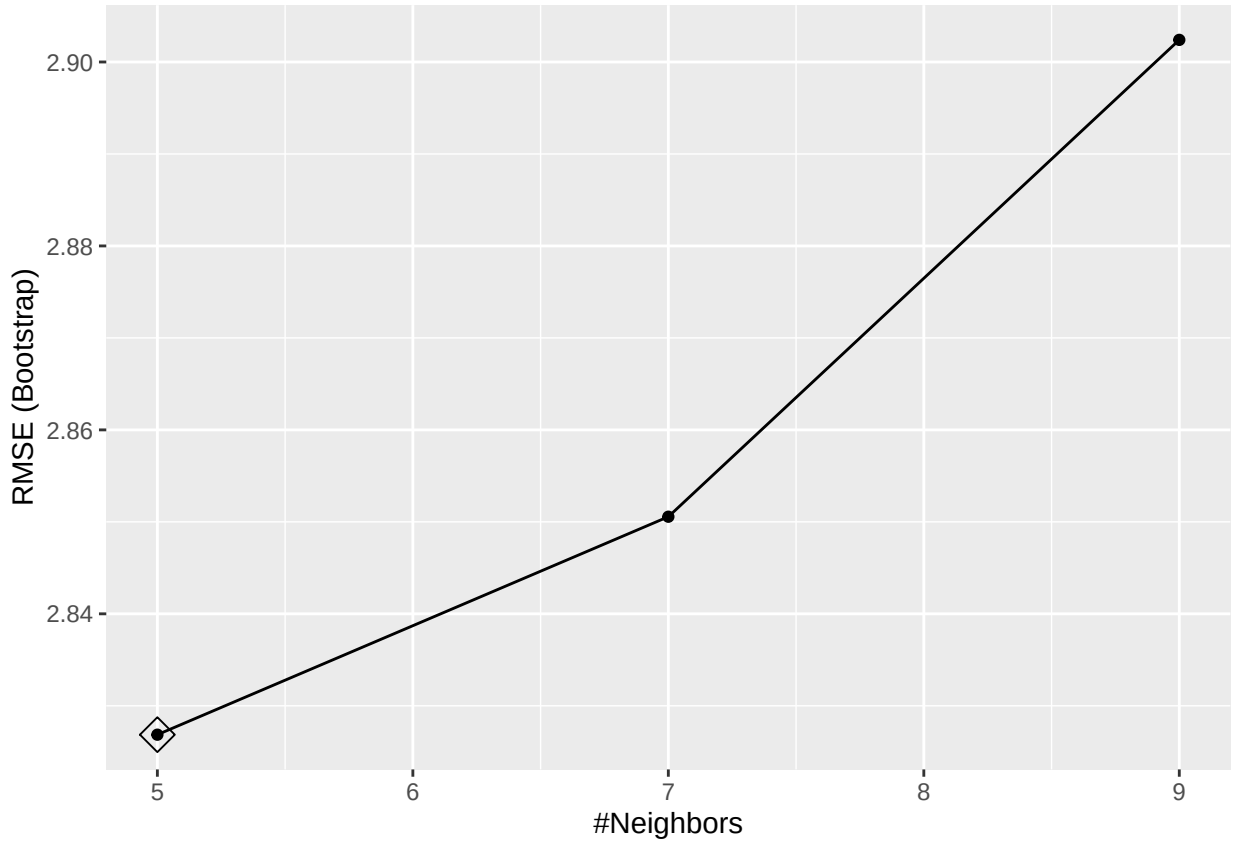
```
rf_time = randomForest(closing_price ~
                         date_month +
                         date_year,
                       data = crude_oil_train,
                       nodesize = nodesize[which.min(rmses)])

pred <- predict(rf_time, newdata = crude_oil_train)

predicted_price_algorithm_04 <-
  predict(rf_time,
          newdata = crude_oil_test)

RMSE04 <- RMSE(predicted_price_algorithm_04,
               crude_oil_test$closing_price)

RMSE04
```

```
## [1] 4.751267
```

Algorithm 05 (KNN - Time) With the intention of finding an algorithm that would be of significant analytical value, attention shifted again, this time to an algorithm that predicted crude oil closing prices

based on a K-Nearest Neighbors (KNN) model of the two key time components: *year* and *month*. The RMSE, much like the RMSE generated via the fourth item, stil not significantly better than the outcome obtained from a simple model of the effects of the month-by-month average closing price.

```
knn_time <-
  train(closing_price ~
          date_year +
          date_month,
        method = "knn",
        data = crude_oil_train)

ggplot(knn_time, highlight = TRUE)
```




```
pred <- predict(knn_time,
                newdata = crude_oil_train)

predicted_price_algorithm_05 <-
  predict(knn_time,
          newdata = crude_oil_test)

RMSE05 <- RMSE(predicted_price_algorithm_05,
               crude_oil_test$closing_price)

RMSE05
```

```
## [1] 3.768831
```

Algorithm 06 (Ranger - Time)

The continued quest for a lower RMSE and, with it, a more valuable algorithm led to the use of a Ranger model incorporating *year* and *month*. That effort's outcome suggested the Ranger model would, alongside KNN, be worthy of further exploration when the time came to consider not only time but the prices of complementary and competing commodities from the energy, precious metals, and agriculture sectors.

```
ranger_time <-
  train(closing_price ~
          date_year +
          date_month,
        method = "ranger",
        data = crude_oil_train)
```

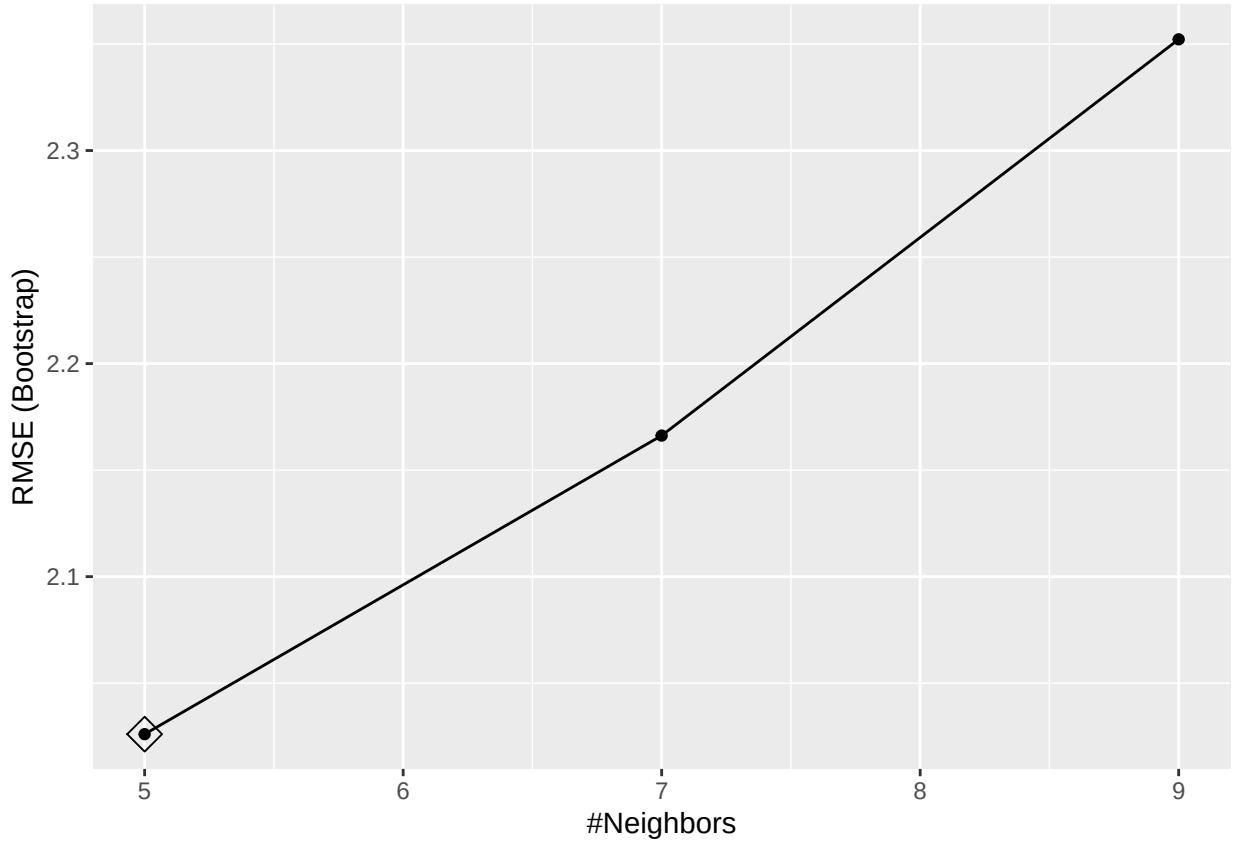
```
## note: only 1 unique complexity parameters in default grid. Truncating the grid to 1 .
```

```
pred <- predict(ranger_time,
                newdata = crude_oil_train)

predicted_price_algorithm_06 <-
  predict(ranger_time,
          newdata = crude_oil_test)

RMSE06 <- RMSE(predicted_price_algorithm_06,
               crude_oil_test$closing_price)

RMSE06
```

```
## [1] 3.768988
```

Algorithm 07 (KNN - Time and Energy) Continuing the question for a lower RMSE, the focus of the algorithm effort fell on the use of a KNN model that took into account not only *year* and *month* but the closing prices of two energy sector commodities: *heating oil* and *gasoline*. The generated RMSE, the first to cross below the 3.00 threshold, spoke to the potential that the seventh algorithm would be of significant analytical value.

```
knn_time_energy <-
  train(closing_price ~
          date_year +
          date_month +
          heating_oil_closing_price +
          gasoline_closing_price,
        method = "knn",
        data = crude_oil_train)

ggplot(knn_time_energy, highlight = TRUE)
```



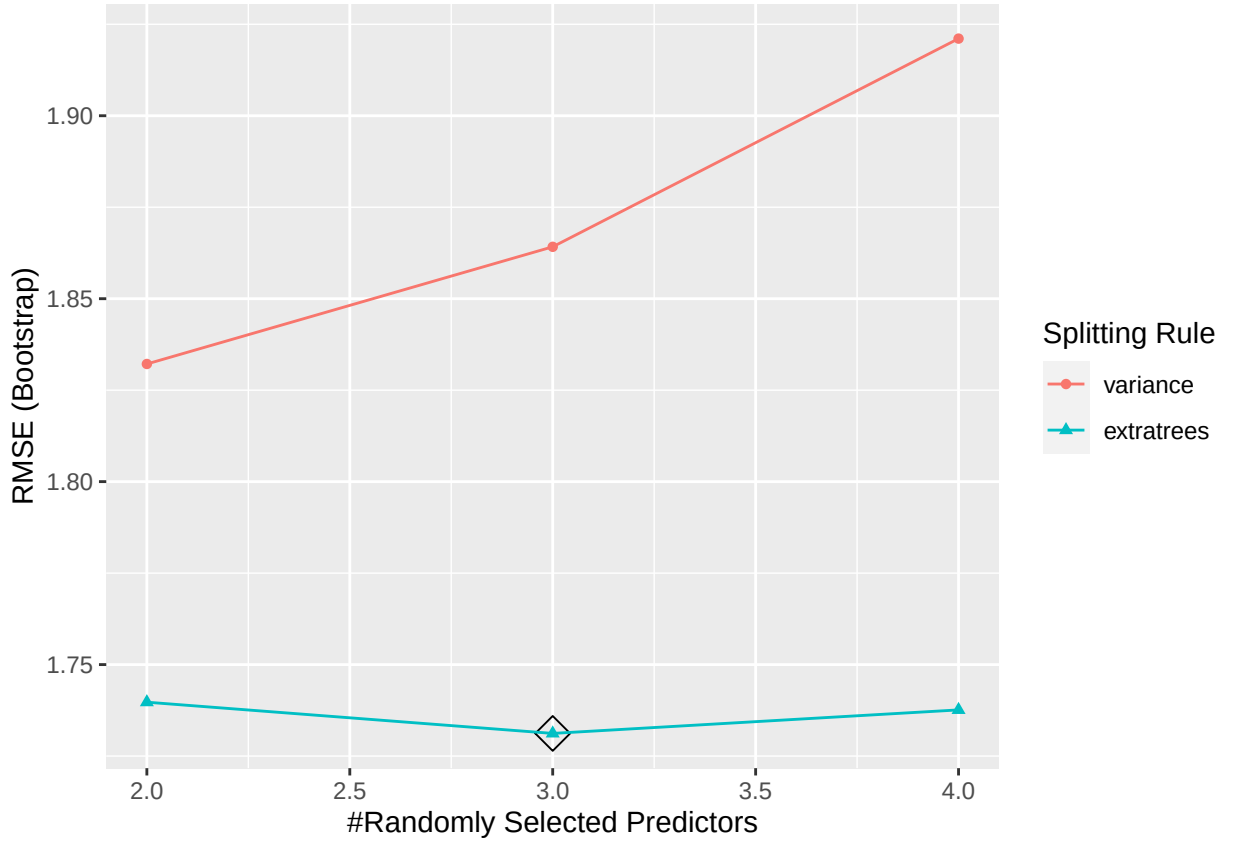

```
pred <- predict(knn_time_energy,
                newdata = crude_oil_train)

predicted_price_algorithm_07 <-
  predict(knn_time_energy,
          newdata = crude_oil_test)

RMSE07 <- RMSE(predicted_price_algorithm_07,
               crude_oil_test$closing_price)

RMSE07
```

```
## [1] 2.956255
```

Algorithm 08 (Ranger - Time and Energy) The next attempt centered on the use of a Ranger model that also took into account not only *year* and *month* but the closing prices of two energy sector commodities: *heating oil* and *gasoline*. The generated RMSE, the second to cross below the 3.00 threshold, spoke to the potential that the eighth algorithm would, like the seventh, be of significant analytical value.

```
ranger_time_energy <-
  train(closing_price ~
          date_year +
          date_month +
          heating_oil_closing_price +
          gasoline_closing_price,
```

```
        method = "ranger",
        data = crude_oil_train)

ggplot(ranger_time_energy, highlight = TRUE)
```

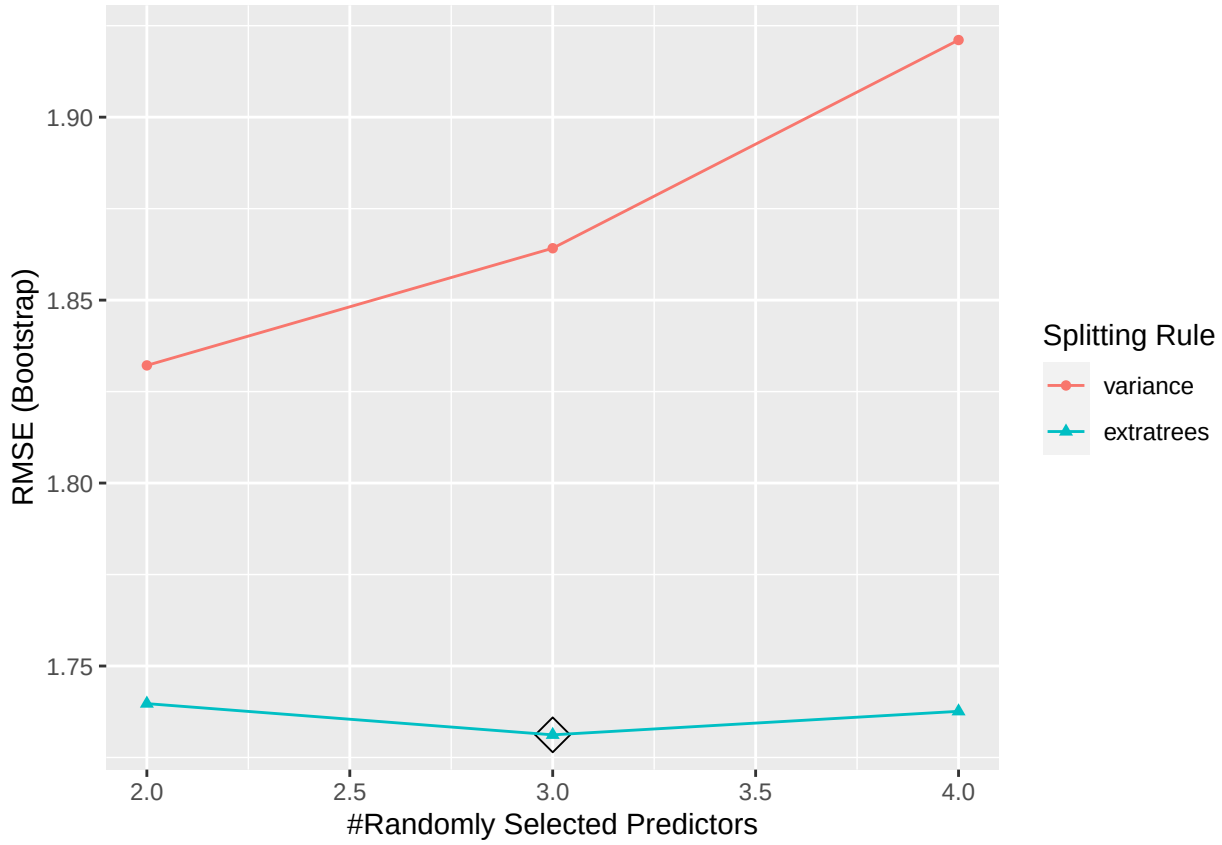



```
pred <- predict(ranger_time_energy,
                newdata = crude_oil_train)

predicted_price_algorithm_08 <-
  predict(ranger_time_energy,
          newdata = crude_oil_test)

RMSE08 <- RMSE(predicted_price_algorithm_08,
               crude_oil_test$closing_price)

RMSE08
```

```
## [1] 2.963471
```

Algorithm 09 (Ranger - Time, Energy, Precious Metals, Agriculture) Incorporating the rest of the promising predictors and leveraging the identified strength of Ranger, the next step entailed evaluating an algorithm that made use of a Ranger model accounting for *year*, *month*, and the closing prices of *heating oil* and *gasoline* but the closing prices of *platinum* and *soybeans*. The generated RMSE, the first to cross below the 2.95 threshold, made clear that the ninth item would of significant value to those engaged in analyzing the oil industry through crude oil price trends.

```
ranger_time_energy_precious_metals_agriculture <-
  train(closing_price ~
          date_year +
          date_month +
          heating_oil_closing_price +
          gasoline_closing_price +
          platinum_closing_price +
          soybeans_closing_price,
          method = "ranger",
        data = crude_oil_train)

ggplot(ranger_time_energy, highlight = TRUE)
```




```
pred <- predict(ranger_time_energy_precious_metals_agriculture,
                newdata = crude_oil_train)

predicted_price_algorithm_09 <-
  predict(ranger_time_energy_precious_metals_agriculture,
          newdata = crude_oil_test)

RMSE09 <- RMSE(predicted_price_algorithm_09,
               crude_oil_test$closing_price)

RMSE09
```

```
## [1] 2.933687
```