Statistical Modeling and Regression Project

Kushal

4/27/2023

Reading data from january 1st 2015 until 2023-04-05. S&P 500 is considered for the whole process. I abstracted the data at a real time from yahoofinance using quantmod library.In the below code, initially we read stock data from yahoo finance calling the yahoofinance api through quantmod package and create a dataframe df.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.0     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.1     v tibble    3.1.8
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(quantmod)
```

```
## Loading required package: xts
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
##
## Attaching package: 'xts'
##
## The following objects are masked from 'package:dplyr':
##
##     first, last
##
## Loading required package: TTR
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
begining <- as.Date("2015-01-01")
last_date <- as.Date("2023-04-05")
raw <- getSymbols("^GSPC", src = "yahoo", from = begining, to = last_date, auto.assign = FALSE)
```

```r
stock_date = index(raw)
closingprice = as.numeric(raw[, "GSPC.Close"])
df <- data.frame(stock_date, closingprice)
```

#Using Linear Regression Model

```r
linear_model <- lm(closingprice ~ stock_date, data = df)
predict_stock <- predict(linear_model, newdata = df)
mse <- mean((predict_stock - df$closingprice)^2)
cat("MSE: ",mse,"\n")
```

```
## MSE:  87223.22
```

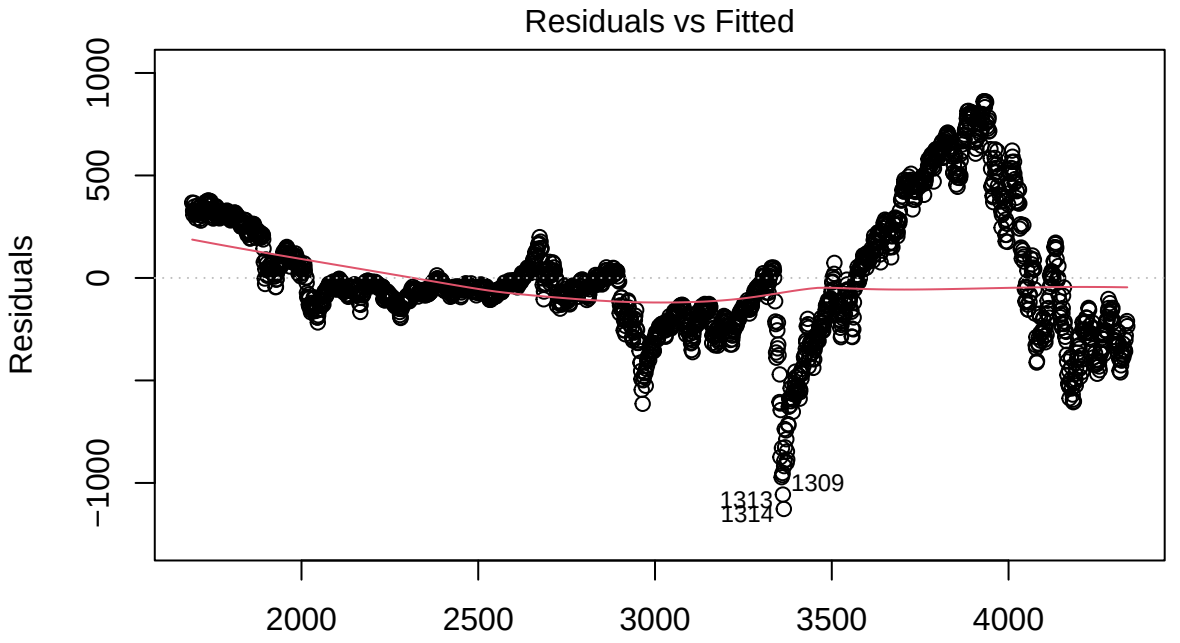
```r
rmse <- sqrt(mse)
cat("RMSE: ",rmse)
```

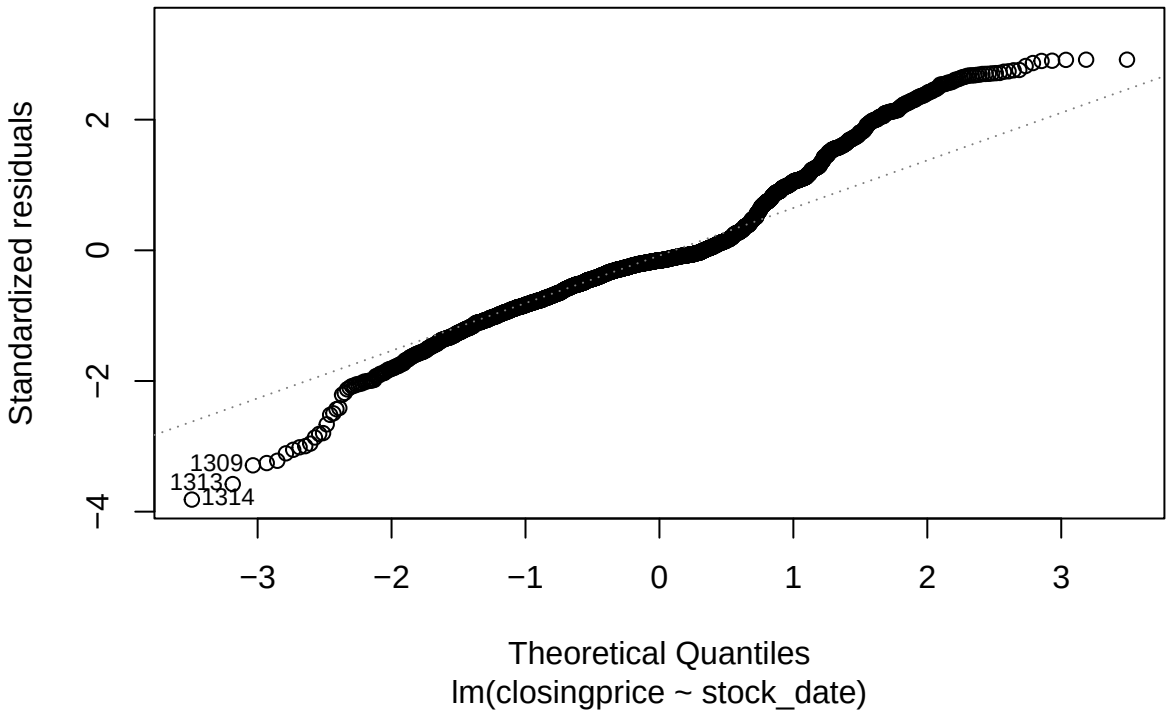
```
## RMSE:  295.3358
```

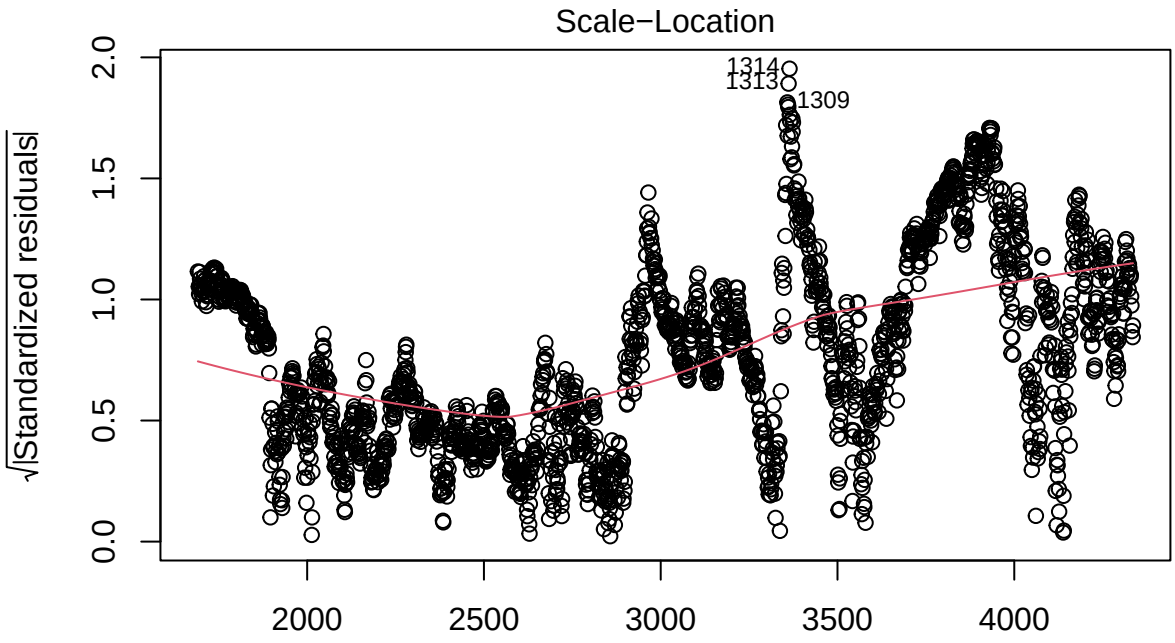
```r
summary(linear_model)
```

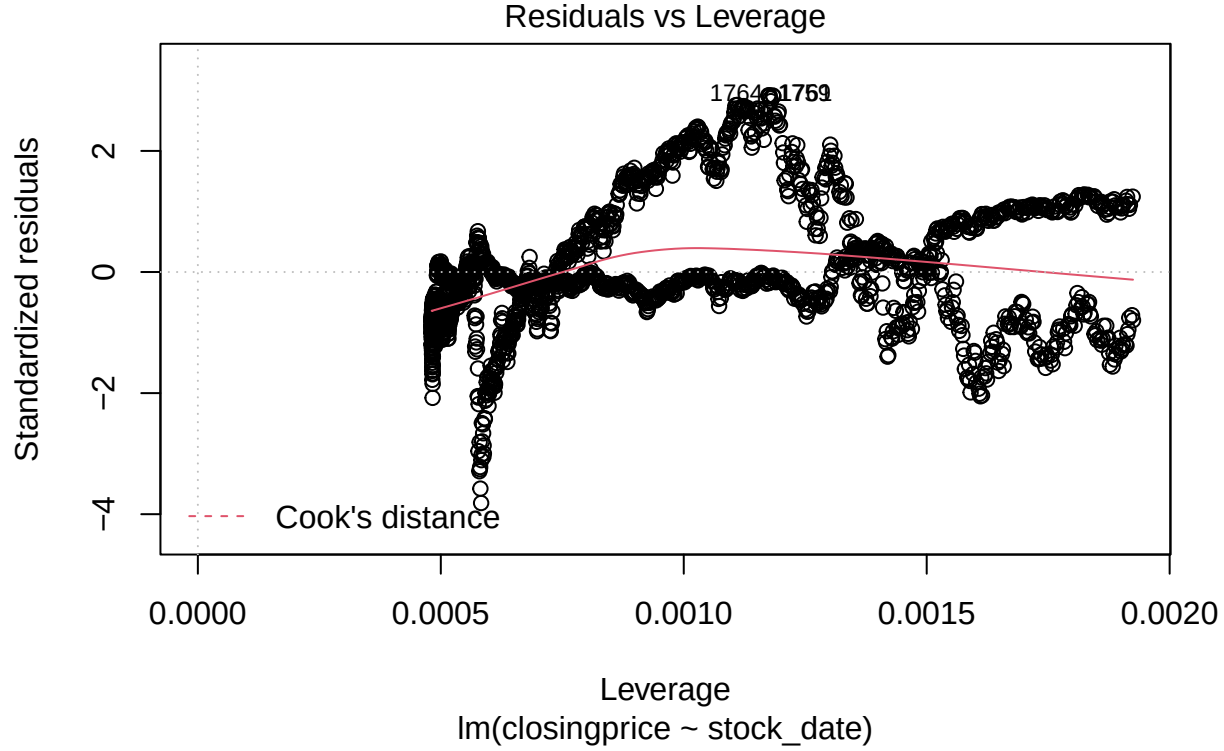
```
##
## Call:
## lm(formula = closingprice ~ stock_date, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1126.97  -168.69   -45.86   121.57   861.77
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.273e+04  1.338e+02  -95.17   <2e-16 ***
## stock_date   8.776e-01  7.448e-03  117.83   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 295.5 on 2076 degrees of freedom
## Multiple R-squared:  0.8699, Adjusted R-squared:  0.8699
## F-statistic: 1.388e+04 on 1 and 2076 DF,  p-value: < 2.2e-16
```

```r
plot(linear_model)
```

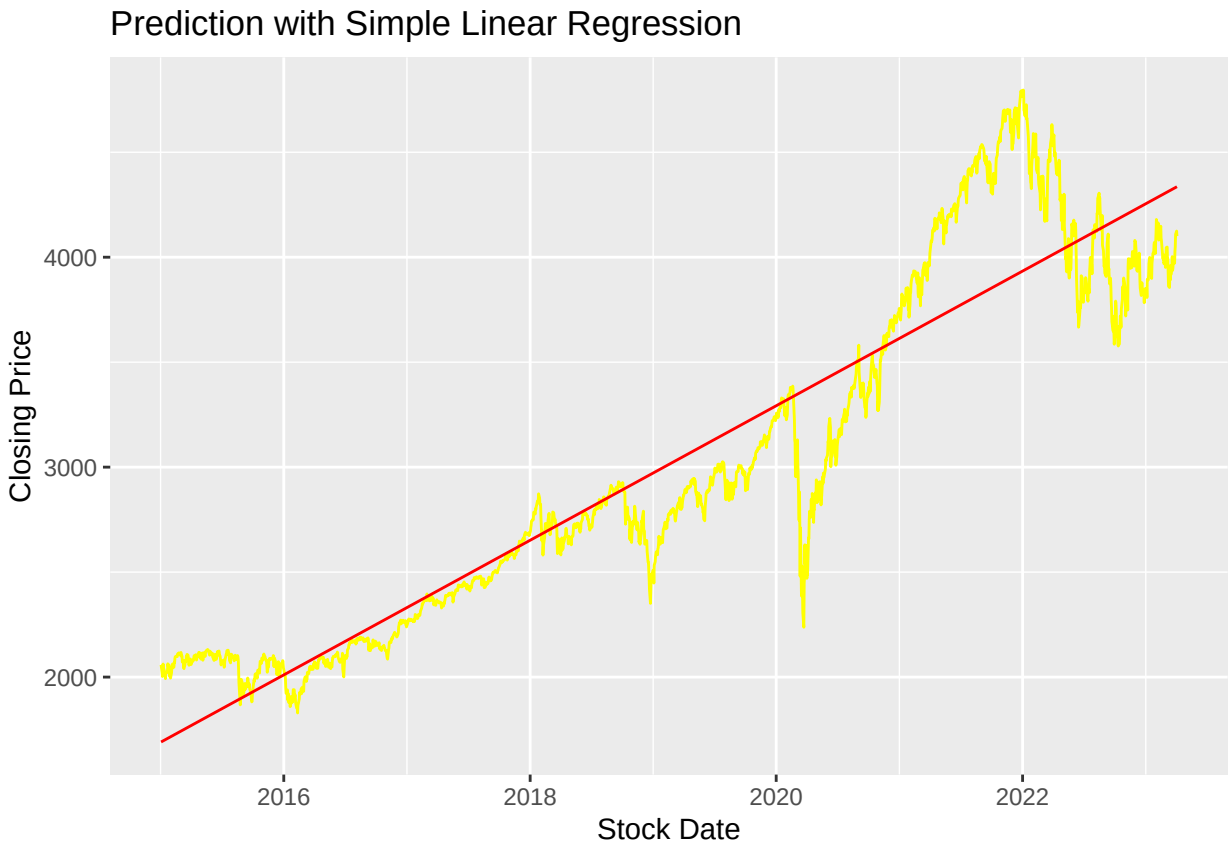

Residuals vs Fitted
Residuals
1000
500
0
−1000
1309
1313
1314
2000
2500
3000
3500
4000
Fitted values
lm(closingprice ~ stock_date)


Normal Q−Q
Standardized residuals
2
0
−2
−4
1309
1313
1314
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(closingprice ~ stock_date)






```
predict_stock <- predict(linear_model, newdata = df)
ggplot() +
  geom_line(data = df, aes(x = stock_date, y = closingprice), color = "yellow") +
  geom_line(data = df, aes(x = stock_date, y = predict_stock), color = "red") +
  labs(title = "Prediction with Simple Linear Regression", x = "Stock Date", y = "Closing Price")
```



#Using Random Forest Regression

```
library(randomForest)
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
rand_model <- randomForest(closingprice ~ stock_date, data = df, ntree = 200, mtry = 1)
predict_stock <- predict(rand_model, newdata = df)
mse <- mean((predict_stock - df$closingprice)^2)
cat("MSE: ",mse,"\n")
```

```
## MSE:  252.7803
```

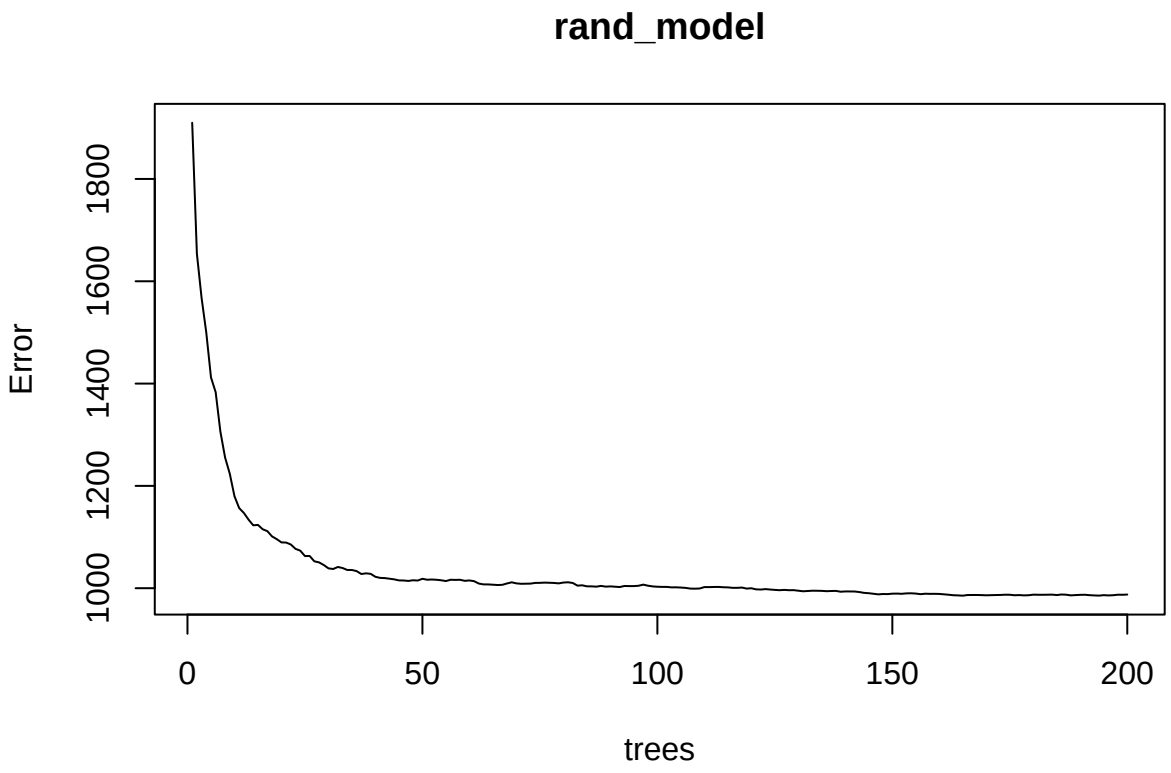
```
rmse <- sqrt(mse)
cat("RMSE: ",rmse)
```

```
## RMSE:  15.89906
```

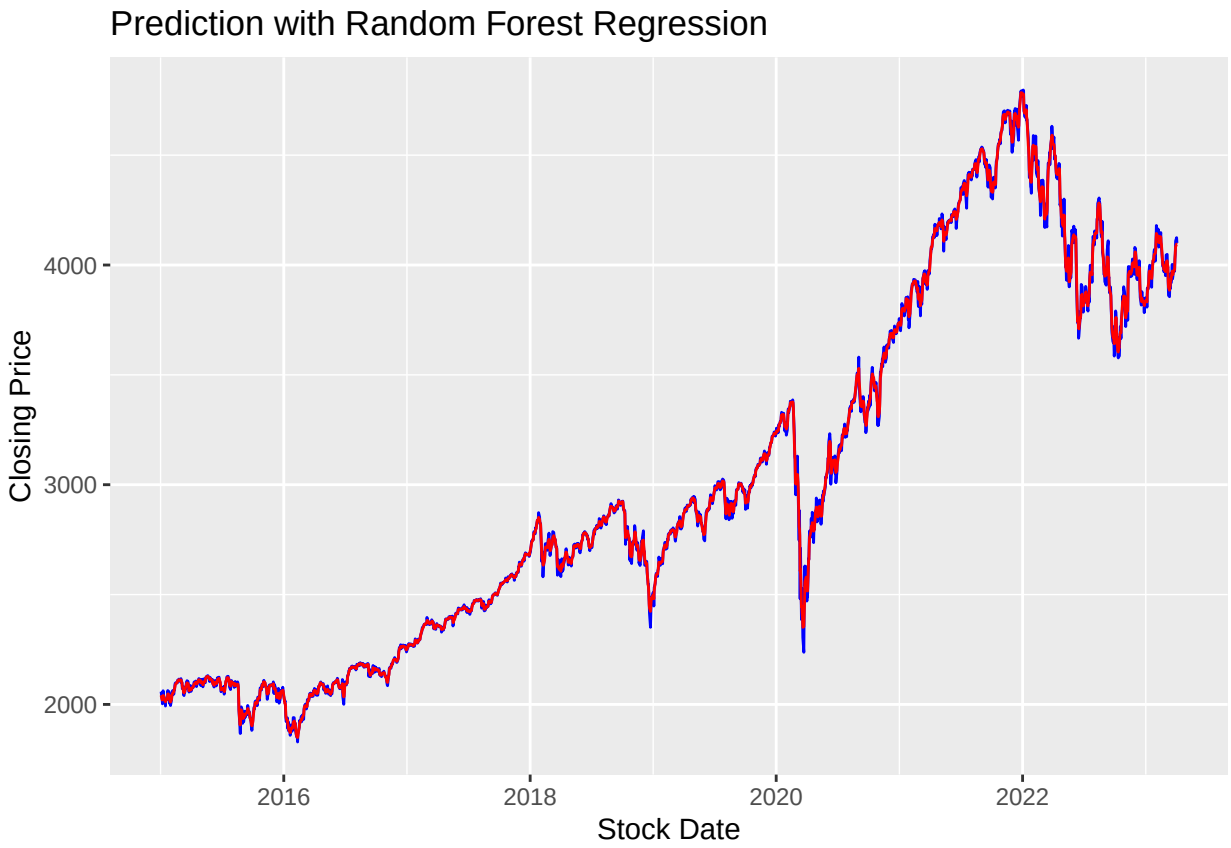
```
summary(rand_model)
```

```
##                 Length Class  Mode
## call               5   -none- call
## type               1   -none- character
## predicted       2078   -none- numeric
## mse              200   -none- numeric
## rsq              200   -none- numeric
## oob.times       2078   -none- numeric
## importance         1   -none- numeric
## importanceSD       0   -none- NULL
## localImportance    0   -none- NULL
## proximity          0   -none- NULL
## ntree              1   -none- numeric
## mtry               1   -none- numeric
## forest            11   -none- list
## coefs              0   -none- NULL
## y               2078   -none- numeric
## test               0   -none- NULL
## inbag              0   -none- NULL
## terms              3   terms  call
```

```
plot(rand_model)
```




```
ggplot() +
  geom_line(data = df, aes(x = stock_date, y = closingprice), color = "blue") +
  geom_line(data = df, aes(x = stock_date, y = predict_stock), color = "red") +
  labs(title = "Prediction with Random Forest Regression", x = "Stock Date", y = "Closing Price")
```



#Using polynomial regression

```
date_num <- as.numeric(df$stock_date)
poly_model <- lm(closingprice ~ poly(date_num, 2, raw = TRUE), data = df)
predict_stock <- predict(poly_model, newdata = df)
mse <- mean((predict_stock - df$closingprice)^2)
cat("MSE: ",mse, "\n")
```

```
## MSE:  81045.81
```

```
rmse <- sqrt(mse)
cat("RMSE: ",rmse)
```

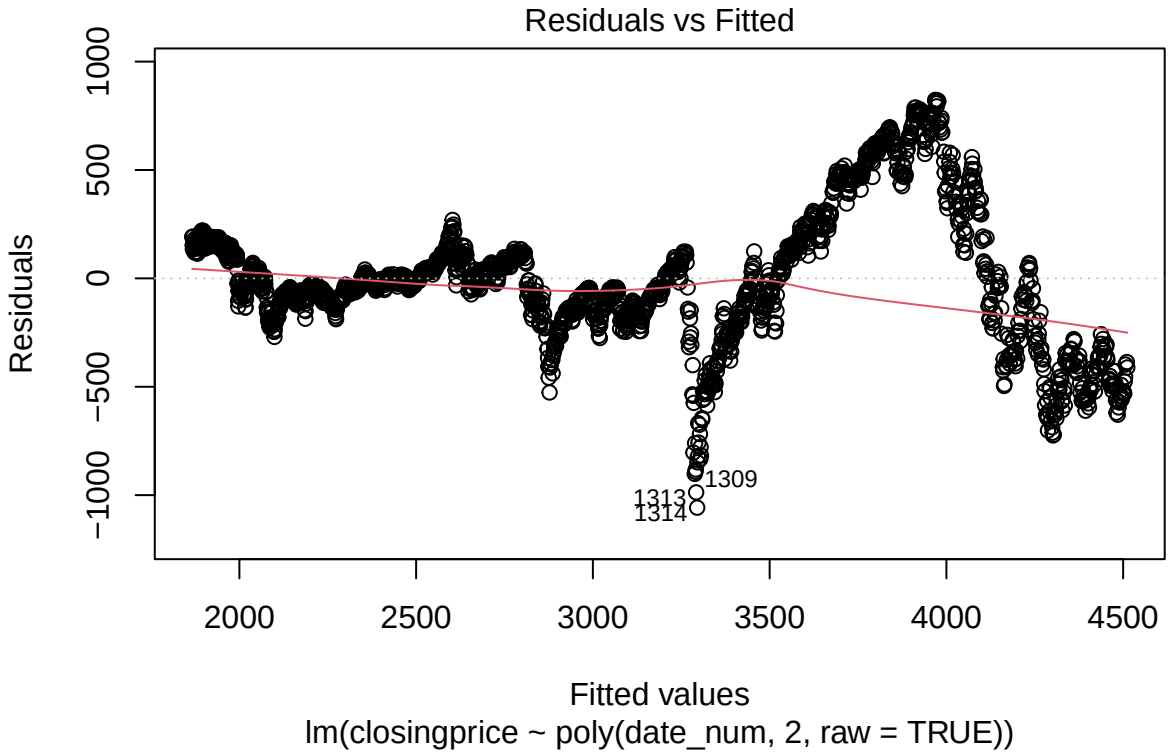
```
## RMSE:  284.6855
```

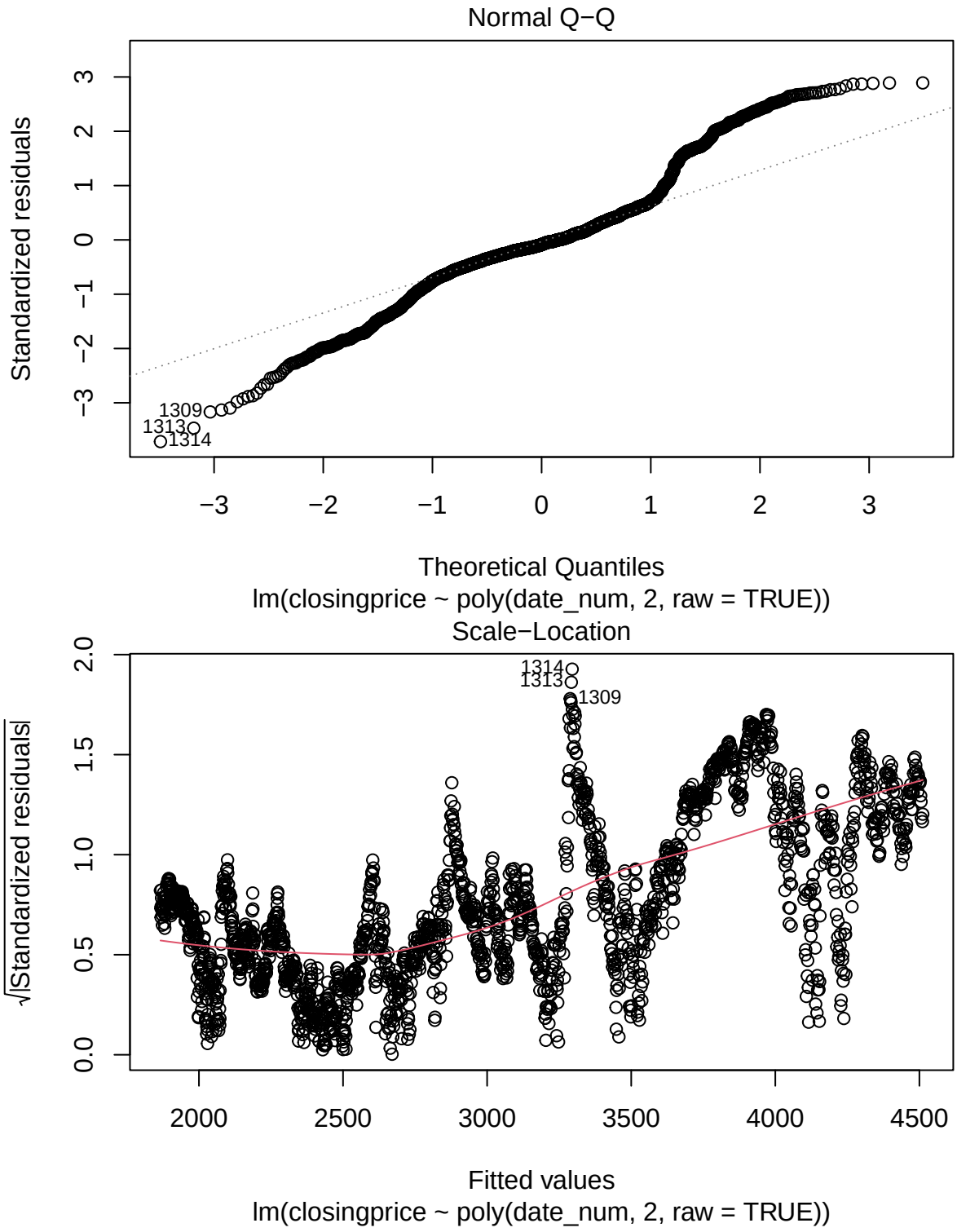
```
summary(poly_model)
```

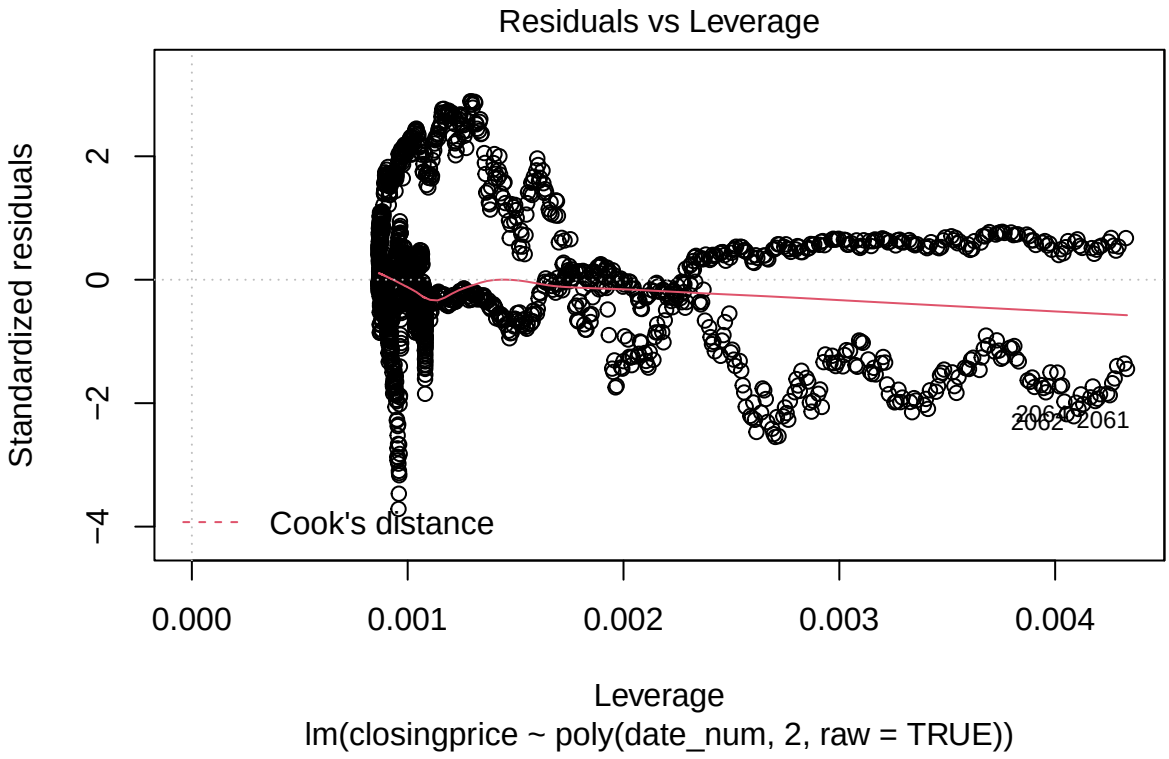
```
##
## Call:
## lm(formula = closingprice ~ poly(date_num, 2, raw = TRUE), data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1057.64  -134.95   -24.54   117.58   822.98
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     2.456e+04  2.968e+03   8.274 2.28e-16 ***
## poly(date_num, 2, raw = TRUE)1 -3.289e+00  3.314e-01  -9.925  < 2e-16 ***
```

```
## poly(date_num, 2, raw = TRUE)2  1.161e-04  9.232e-06  12.576  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 284.9 on 2075 degrees of freedom
## Multiple R-squared:  0.8791, Adjusted R-squared:  0.879
## F-statistic:  7546 on 2 and 2075 DF,  p-value: < 2.2e-16
```

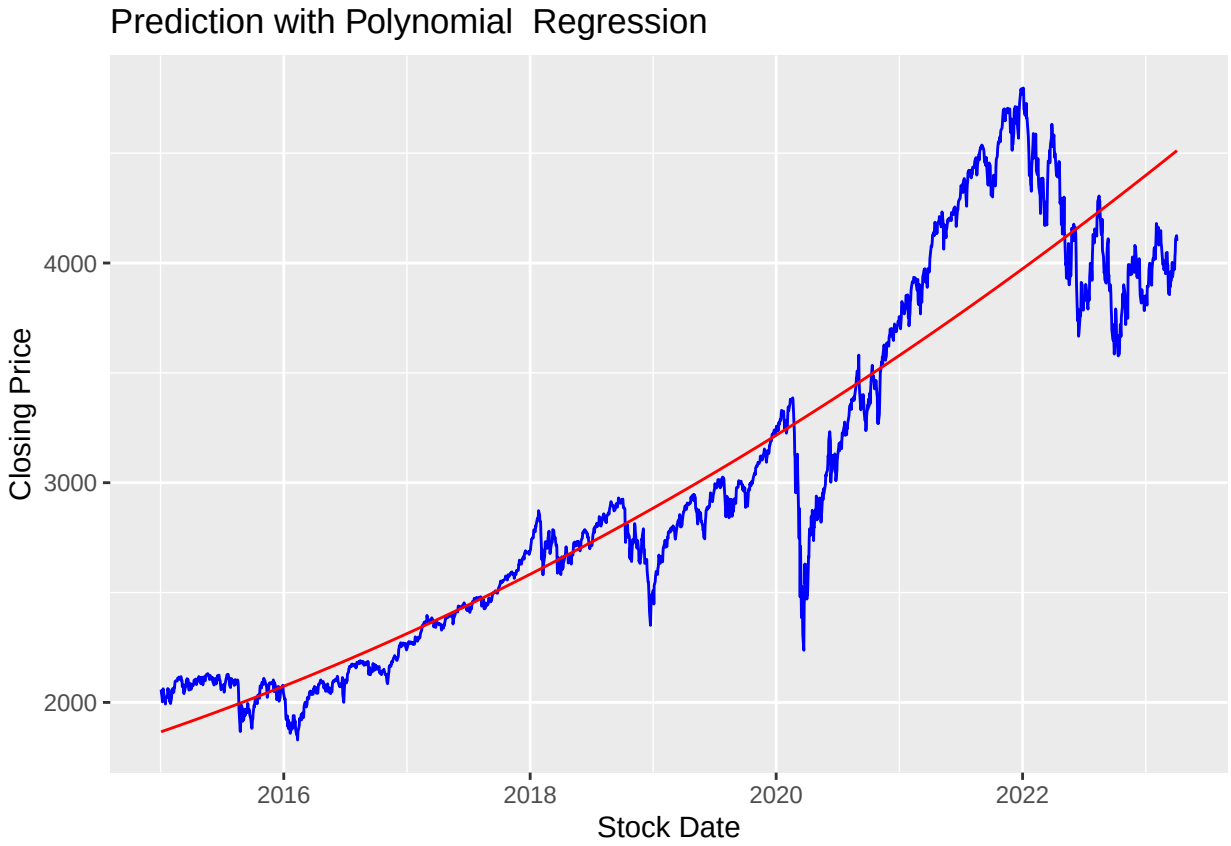
```
plot(poly_model)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(closingprice ~ poly(date_num, 2, raw = TRUE))
Scale−Location
√|Standardized residuals|
Fitted values
lm(closingprice ~ poly(date_num, 2, raw = TRUE))
1309
1313
1314




```
ggplot() +
  geom_line(data = df, aes(x = stock_date, y = closingprice), color = "blue") +
  geom_line(data = df, aes(x = stock_date, y = predict_stock), color = "red") +
  labs(title = "Prediction with Polynomial  Regression", x = "Stock Date", y = "Closing Price")
```




```
AIC(linear_model,  poly_model)
```

```
##              df      AIC
## linear_model  3 29542.91
## poly_model    4 29392.26
```

```
BIC(linear_model,  poly_model)
```

```
##              df      BIC
## linear_model  3 29559.82
## poly_model    4 29414.82
```

Project Analysis:

For the three models, I have calculated MSE and RMSE to analyse how each model is doing. Model based on random forest has the lowest RMSE of 15.92 which is significantly lower than the linear(RMSE=295.3358) and polynomial(RMSE=284.6855) regression. It looks like for this dataset, with two variables, closing price depending only on the date, random forest is doing good compared to linear regression and polynomial regression. But when we consider only linear and polynomial, we can see that polynomial regression has lower rmse. Also comapring AIC and BIC between the polynomial and linear regression models, polynomial has lower AIC and BIC values indicating it as a best model. If we had more predictors, the model performance based on other attributes can be different. These evaluation metrics can play a significant role. below are some of the r programming implemented for my project.