Assessment02_pt_b

Xilin Huang Liam

20/09/2021

Loading data from Yahoo Finance

Before we starting working on building models and analysis, data need to be loaded from Yahoo Finance API. In our project, we will be collecting:

Coins (to US dollar)

- BTC-USD Bitcoin
- ETH-USD Ethereum
- ADA-USD Cardano
- DOGE-USD Dogecoin
- SHIB-USD Shiba Inu coin

Index

- ^DJI Dow Jones Industrial
- ^IXIC Nasdaq Composite
- ^GSPC S&P 500
- GC=F Gold

Stocks

- TSLA Tesla
- GOOG Google
- AAPL Apple
- NVDA Nvidia
- AMD Advanced Micro Devices
- TSM Taiwan Semiconductor Manufacturing

All the data will be saved as `xts` (Extensible Time Series) object.

```
rm(list=ls())
library(zoo)
library(xts)
library(TTR)
library(quantmod)

coin_portfolio=c("BTC-USD","ETH-USD","ADA-USD",
                 "DOGE-USD","SHIB-USD")
index_portfolio=c("^DJI","^IXIC","^GSPC","GC=F")
stock_portfolio=c("TSLA","GOOG","AAPL","NVDA","AMD","TSM")

data <- getSymbols(c(coin_portfolio,
                     index_portfolio,
                     stock_portfolio),
                   src='yahoo',
                   #from=dyear,
                   #to=d,
                   autoassign=FALSE)
```

Simple linear regression

For simple linear regression, we will use TSLA stock - Bitcoin as example. Since this project will be focusing on performing linear regression models, the data will be transformed from `xts` object to `dataframe`.

```
df_BTC = data.frame(date=index(`BTC-USD`), coredata(`BTC-USD`))
df_TSLA = data.frame(date=index(TSLA), coredata(TSLA))
```

```
library(tidyr)
# use Friday's data for weekends
df_BTCTSLA <- merge(df_BTC,df_TSLA,by='date', all.x = TRUE)
df_BTCTSLA_filled <- df_BTCTSLA %>%
fill(TSLA.Open,TSLA.High,TSLA.Low,TSLA.Close,TSLA.Adjusted,TSLA.Volume)
```

```
# subset data (2019, close price and volume)
df_BTCTSLA_sub <- subset(df_BTCTSLA_filled, date>='2019-01-01', select=c(date,TSLA.Close,TSLA.Volume,BT
row.names(df_BTCTSLA_sub) <- NULL
```

```
library(dplyr)
lag_list = c(1, 3, 5, 10, 20, 30, 100)
for (i in lag_list){
  if (i == lag_list[1]) {
    df_BTCTSLA_lag = data.frame(col1 = lag(df_BTCTSLA_sub$TSLA.Close, n = i))
    names(df_BTCTSLA_lag)[ncol(df_BTCTSLA_lag)] <- paste0("TSLA_price_lag_", i)
  } else {
    df_BTCTSLA_lag[,ncol(df_BTCTSLA_lag)+1] <- lag(df_BTCTSLA_sub$TSLA.Close, n = i)
    names(df_BTCTSLA_lag)[ncol(df_BTCTSLA_lag)] <- paste0("TSLA_price_lag_", i)
  }
}
```

```
## Warning in diff(df_BTCTSLA_lag_m$BTC.USD.Close)/df_BTCTSLA_lag_m$BTC.USD.Close:
## longer object length is not a multiple of shorter object length
```

```
## Warning in diff(df_BTCTSLA_lag_m$TSLA.Close)/df_BTCTSLA_lag_m$TSLA.Close: longer
## object length is not a multiple of shorter object length
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
library(ggfortify)
```

```
## Warning: package 'ggfortify' was built under R version 4.0.5
```

```
x_value = log(df_BTCTSLA_lag_m$BTC.USD.Close)
y_value = df_BTCTSLA_lag_m$TSLA_price_lag_1
fit=lm(data = df_BTCTSLA_lag_m, x_value~y_value)
summary(fit)
```

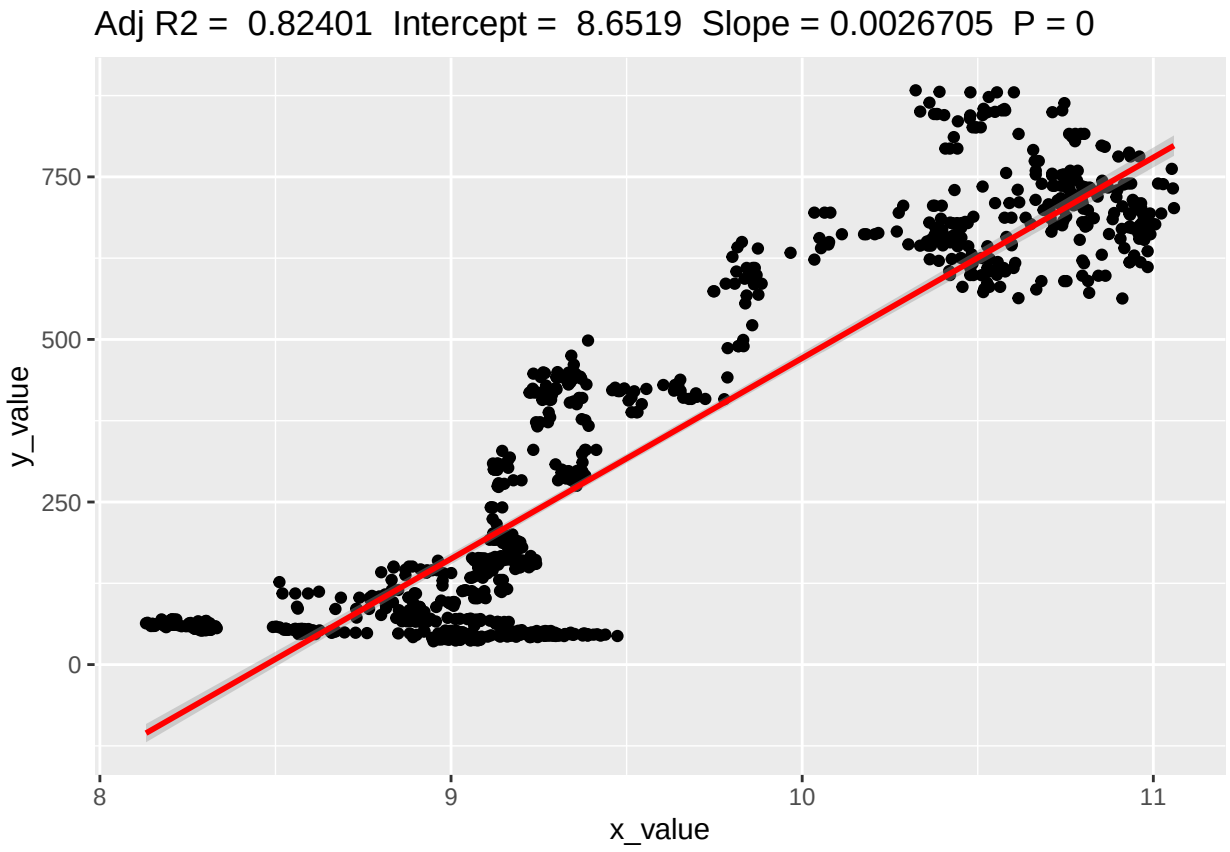
```
##
## Call:
## lm(formula = x_value ~ y_value, data = df_BTCTSLA_lag_m)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.68998 -0.25402  0.03193  0.24400  0.75645
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 8.652e+00  1.642e-02  526.84   <2e-16 ***
## y_value     2.671e-03  3.912e-05   68.26   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3439 on 994 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.8242, Adjusted R-squared:  0.824
## F-statistic:  4660 on 1 and 994 DF,  p-value: < 2.2e-16
```

```
ggplot(df_BTCTSLA_lag_m, aes(x = x_value, y = y_value)) +
  geom_point() +
  stat_smooth(method = 'lm', col = 'red') +
  labs(title = paste("Adj R2 = ", signif(summary(fit)$adj.r.squared, 5),
                     " Intercept = ", signif(fit$coef[[1]], 5),
                     " Slope =", signif(fit$coef[[2]], 5),
                     " P =", signif(summary(fit)$coef[2,4], 5)))
```

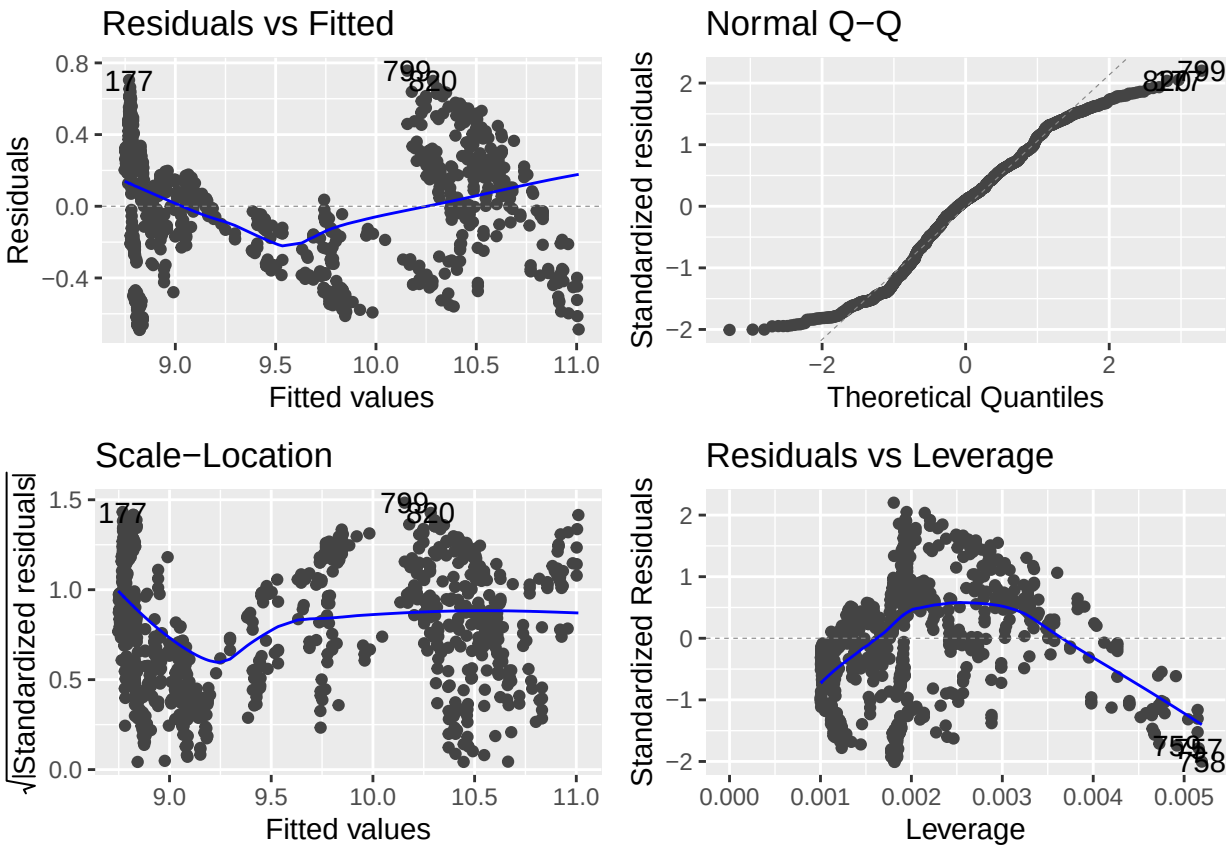
```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 6 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 6 rows containing missing values (geom_point).
```




```
autoplot(fit)
```

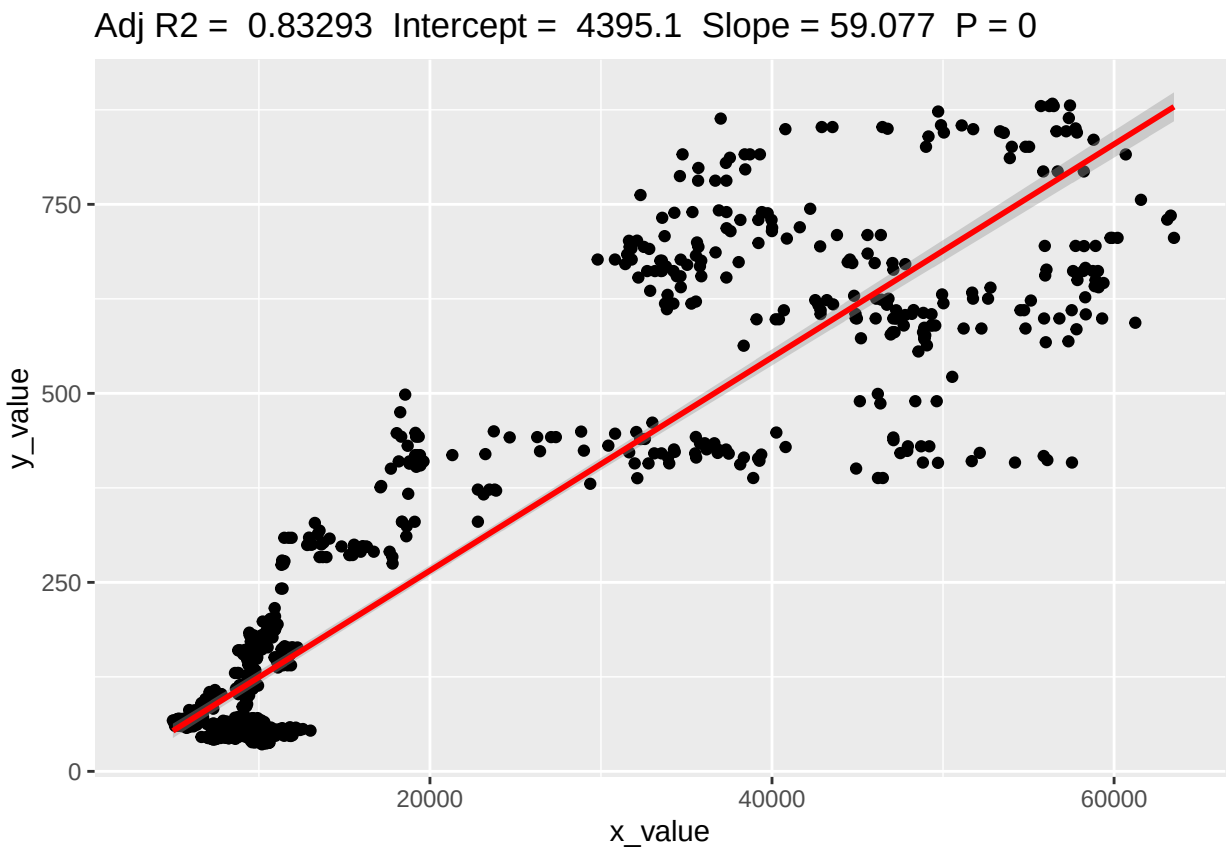


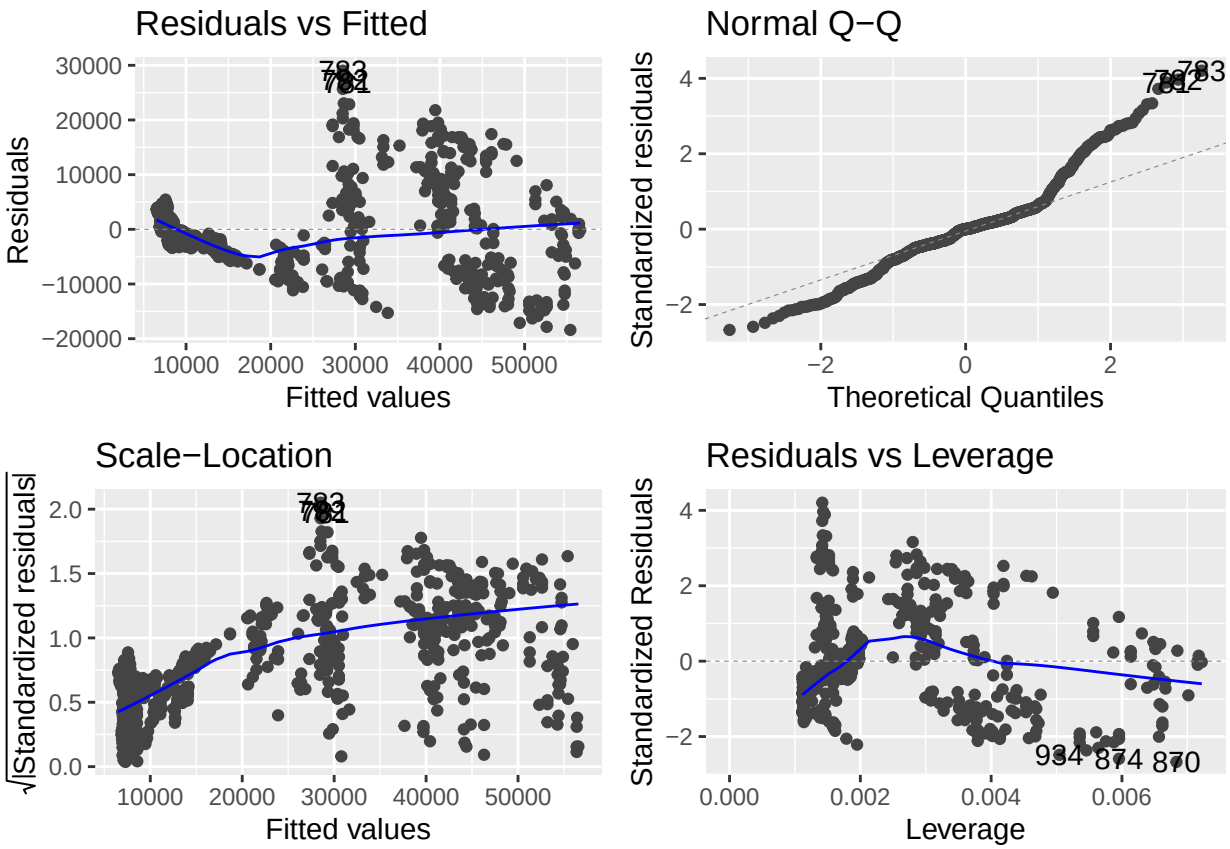

```
##
## Call:
## lm(formula = x_value ~ y_value, data = df_BTCTSLA_lag_m)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -18401  -3352     17   2688  29012
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4395.1183   333.4078   13.18   <2e-16 ***
## y_value       59.0768     0.8838   66.84   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6907 on 895 degrees of freedom
##   (105 observations deleted due to missingness)
## Multiple R-squared:  0.8331, Adjusted R-squared:  0.8329
## F-statistic:  4468 on 1 and 895 DF,  p-value: < 2.2e-16

## `geom_smooth()` using formula 'y ~ x'

## Warning: Removed 105 rows containing non-finite values (stat_smooth).

## Warning: Removed 105 rows containing missing values (geom_point).
```


Adj R2 = 0.83293 Intercept = 4395.1 Slope = 59.077 P = 0
y_value
750
500
250
0
20000
40000
60000
x_value

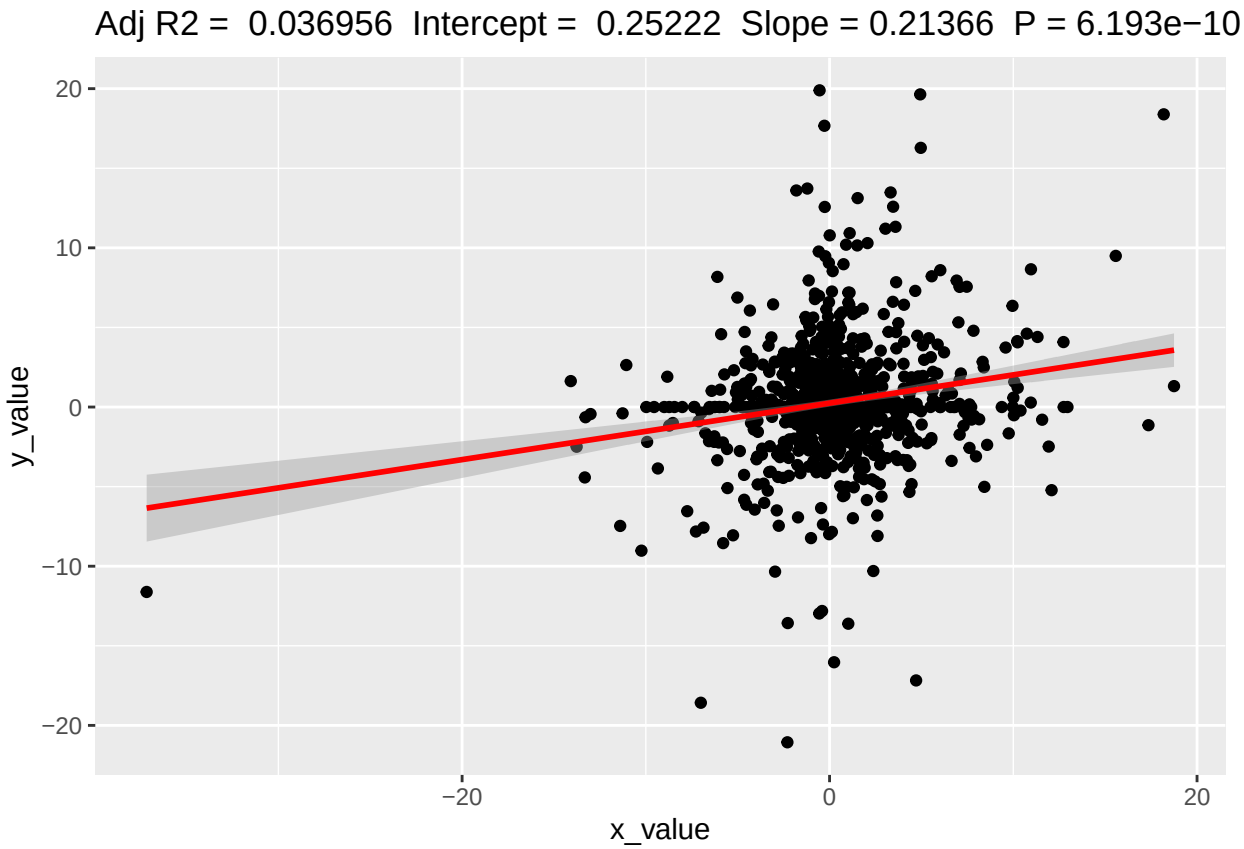


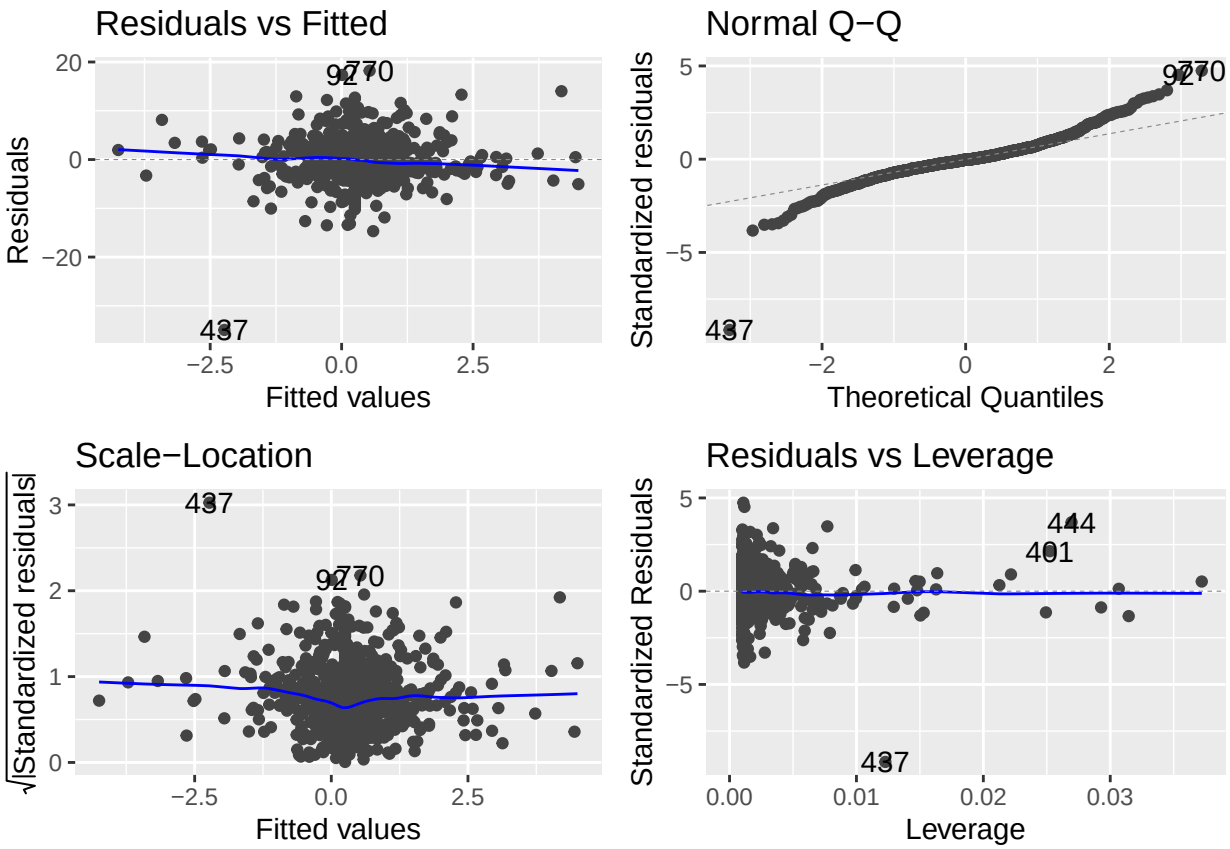

```
##
## Call:
## lm(formula = x_value ~ y_value, data = df_BTCTSLA_lag_m)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.940  -1.814  -0.123   1.742  18.214
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.2522     0.1224   2.061   0.0395 *
## y_value       0.2137     0.0342   6.247 6.19e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.84 on 990 degrees of freedom
##   (10 observations deleted due to missingness)
## Multiple R-squared:  0.03793,    Adjusted R-squared:  0.03696
## F-statistic: 39.03 on 1 and 990 DF,  p-value: 6.193e-10
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 10 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 10 rows containing missing values (geom_point).
```


Adj R2 = 0.036956 Intercept = 0.25222 Slope = 0.21366 P = 6.193e−10
20
10
0
−10
−20
y_value
−20
0
20
x_value


Residuals vs Fitted
Residuals
Fitted values
Normal Q−Q
Standardized residuals
Theoretical Quantiles
Scale−Location
√|Standardized residuals|
Fitted values
Residuals vs Leverage
Standardized Residuals
Leverage