Timeseries Honeywell

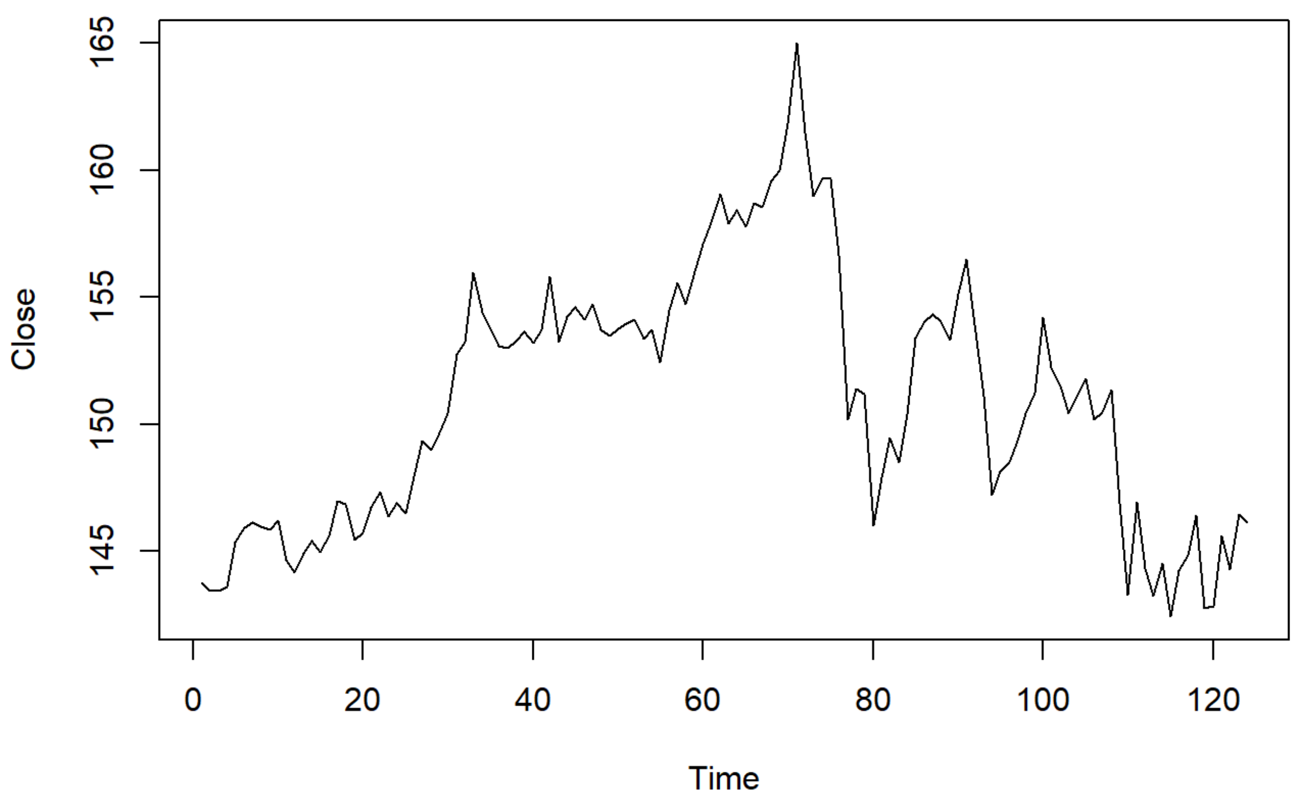
Akshay

We first load the libraries and the data file

```
library(forecast)
```

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

```
library(ggplot2)

ts_honey <-  read.csv("Honeywell.csv")
ts_honey <- ts_honey[c(-1,-3)]

ts_honey <- ts(ts_honey)
plot.ts(ts_honey)
```



Perform analysis on time seris data using Exponential smoothing and varying alpha to see which has the least SSE

```r
mod1 <- HoltWinters(ts_honey, alpha=0.15, beta=FALSE, gamma=FALSE)
mod2 <- HoltWinters(ts_honey, alpha=0.35, beta=FALSE, gamma=FALSE)

mod3 <- HoltWinters(ts_honey, alpha=0.55, beta=FALSE, gamma=FALSE)
mod4 <- HoltWinters(ts_honey, alpha=0.75, beta=FALSE, gamma=FALSE)
modauto <- HoltWinters(ts_honey,  beta=FALSE, gamma=FALSE)


mod1$SSE
```

```
## [1] 980.5197
```

```r
mod2$SSE
```

```
## [1] 558.6834
```

```r
mod3$SSE
```

```
## [1] 419.7806
```

```r
mod4$SSE
```

```
## [1] 366.2911
```

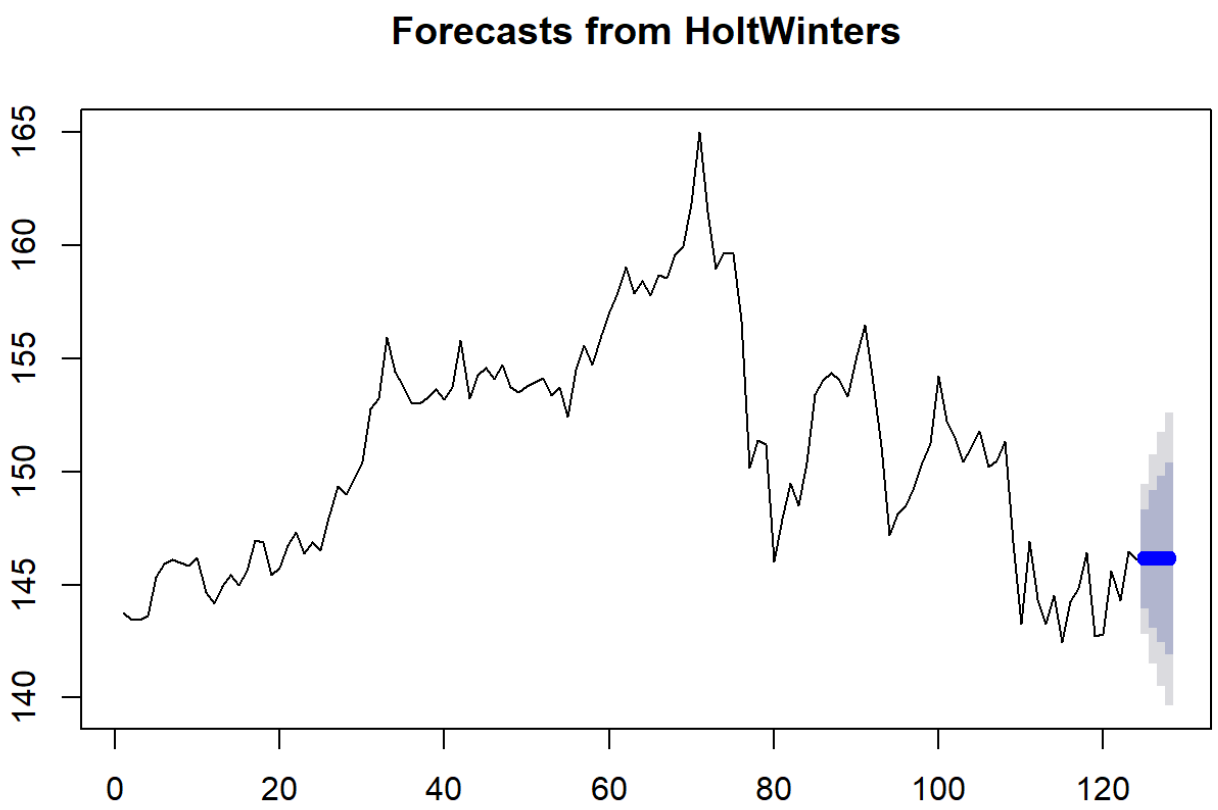
```r
modauto$SSE
```

```
## [1] 351.8547
```

```r
modforecast <- forecast(modauto, h = 4)
modforecast
```

```
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 125       146.1296 143.9533 148.3058 142.8013 149.4578
## 126       146.1296 143.1049 149.1543 141.5037 150.7554
## 127       146.1296 142.4470 149.8122 140.4975 151.7616
## 128       146.1296 141.8900 150.3692 139.6457 152.6135
```

```r
plot(modforecast)
```



Changing the beta parameter on HoltzWinters

```
ts_honey <-  read.csv("Honeywell.csv")

ts_honey <- ts_honey[c(-1,-3)]


mod1 <- HoltWinters(ts_honey, alpha=0.75, beta=0.15, gamma=FALSE)
mod2 <- HoltWinters(ts_honey, alpha=0.75, beta=0.25, gamma=FALSE)

mod3 <- HoltWinters(ts_honey, alpha=0.75, beta=0.45, gamma=FALSE)
mod4 <- HoltWinters(ts_honey, alpha=0.75, beta=0.85, gamma=FALSE)
modauto <- HoltWinters(ts_honey,   gamma=FALSE)



mod1$SSE
```

```
## [1] 396.3036
```

```
mod2$SSE
```

```
## [1] 406.4191
```

```
mod3$SSE
```

```
## [1] 427.4637
```

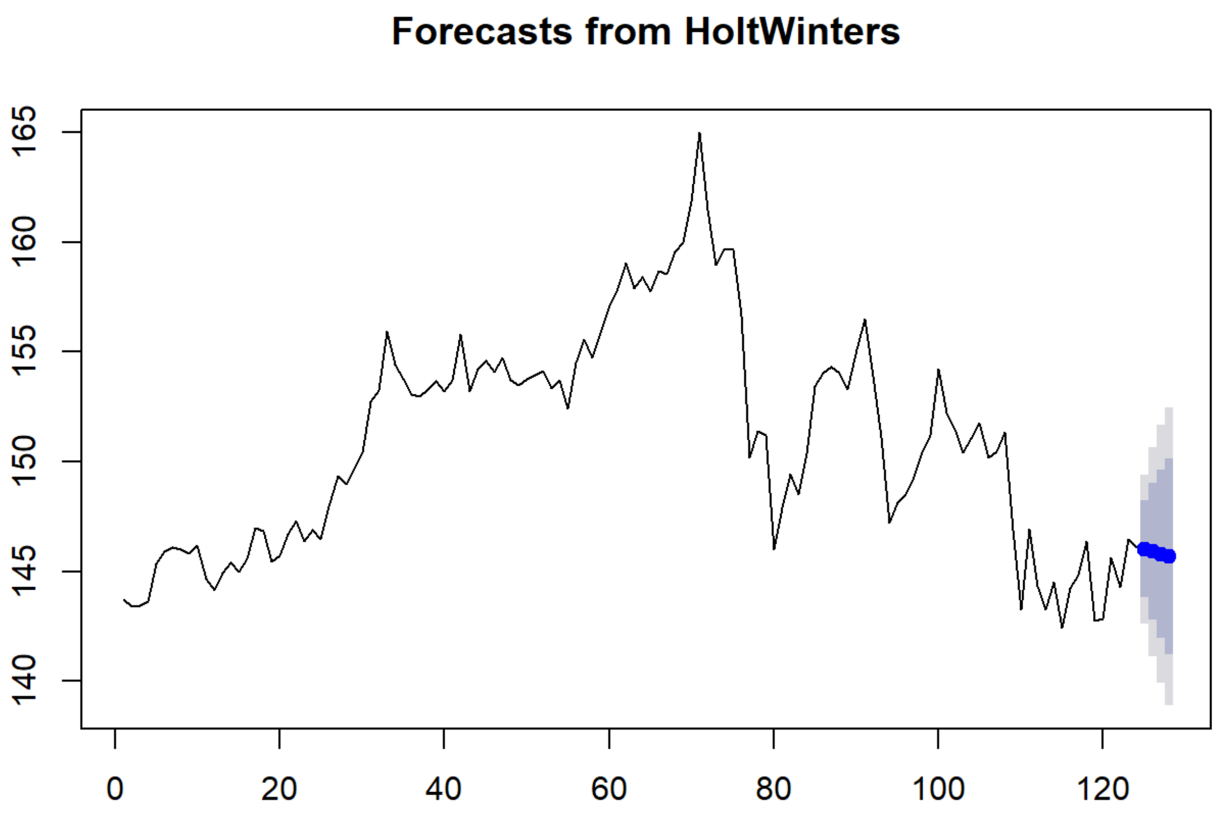
```
mod4$SSE
```

```
## [1] 513.2909
```

```
modauto$SSE
```

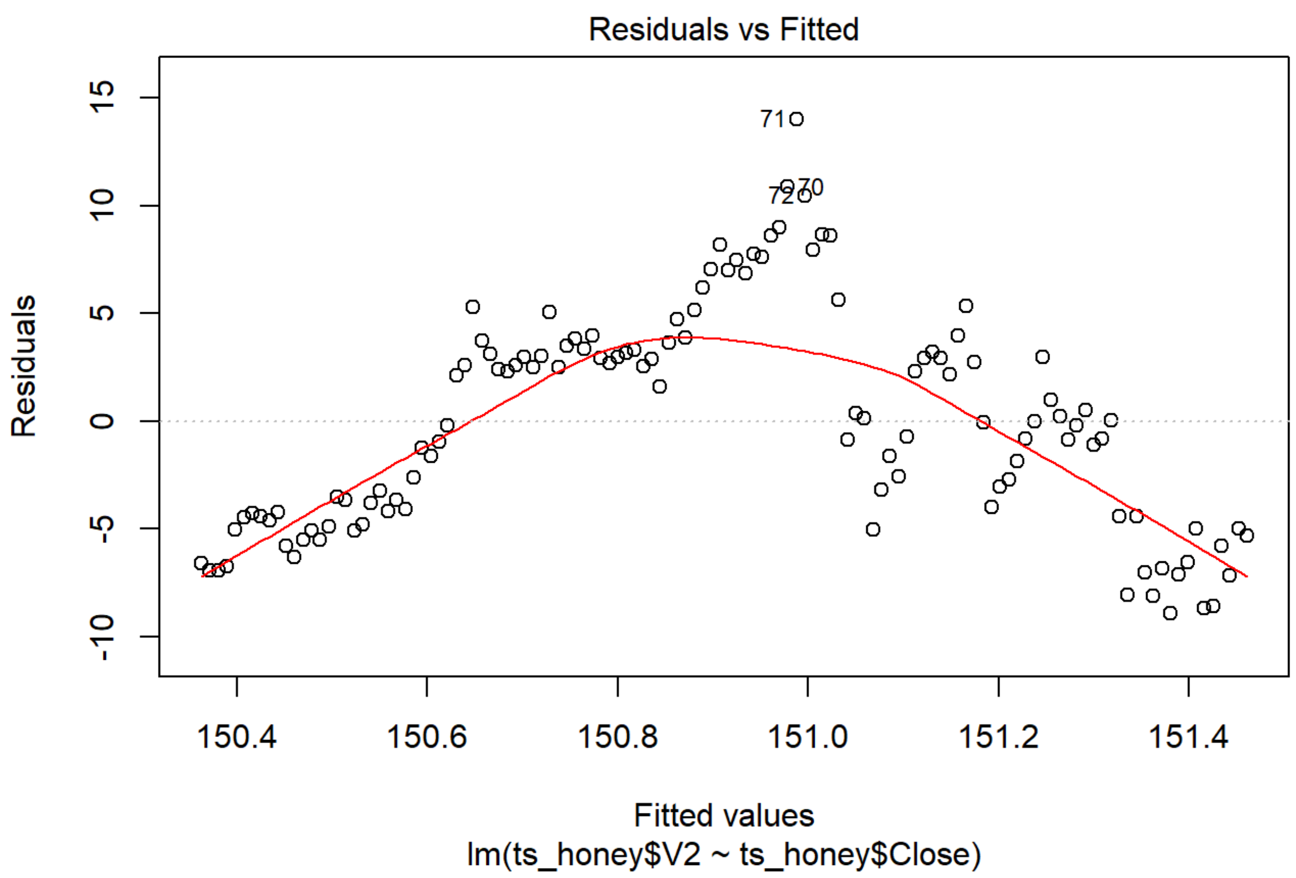
```
## [1] 361.2713
```

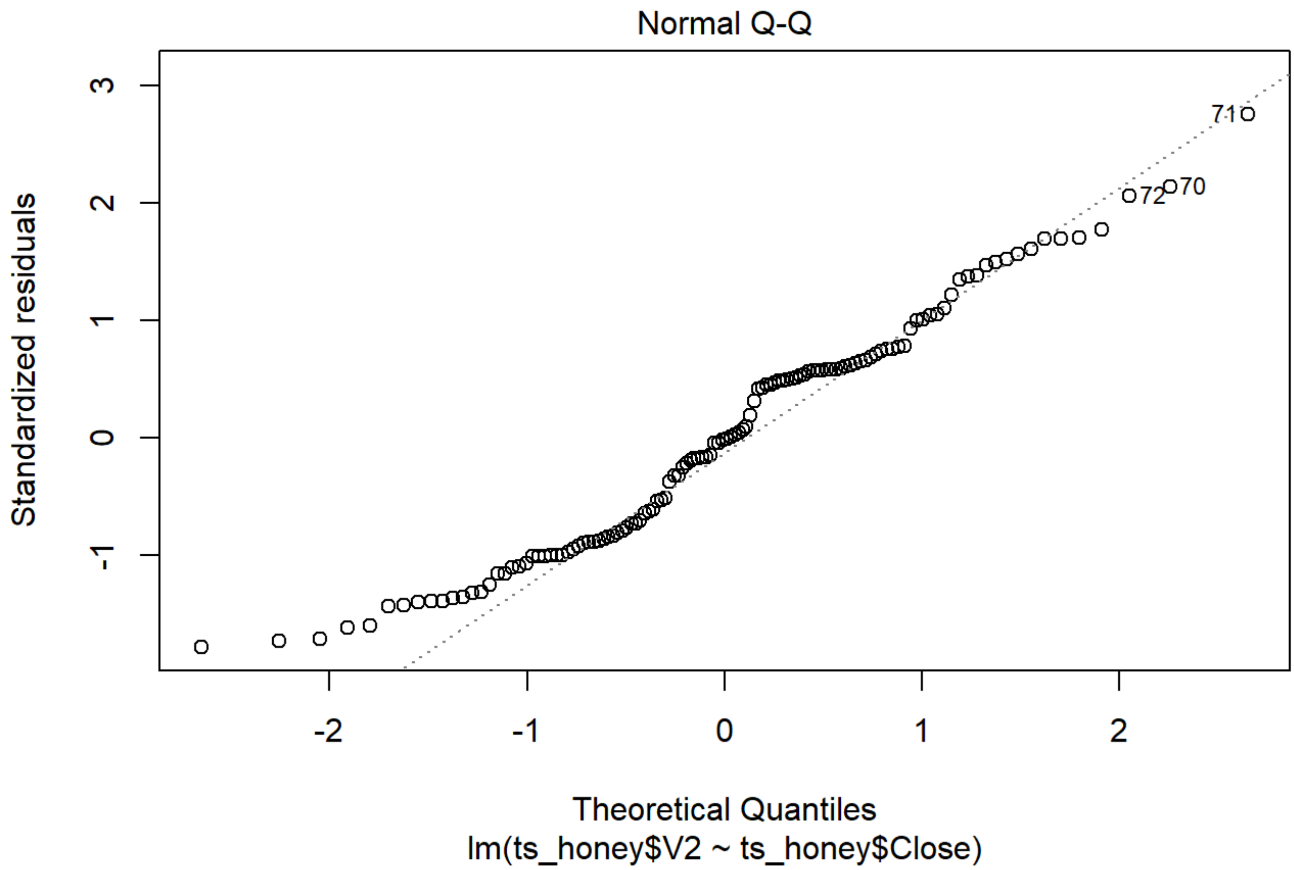
```
modforecast <- forecast(modauto, h = 4)

plot(modforecast)
```

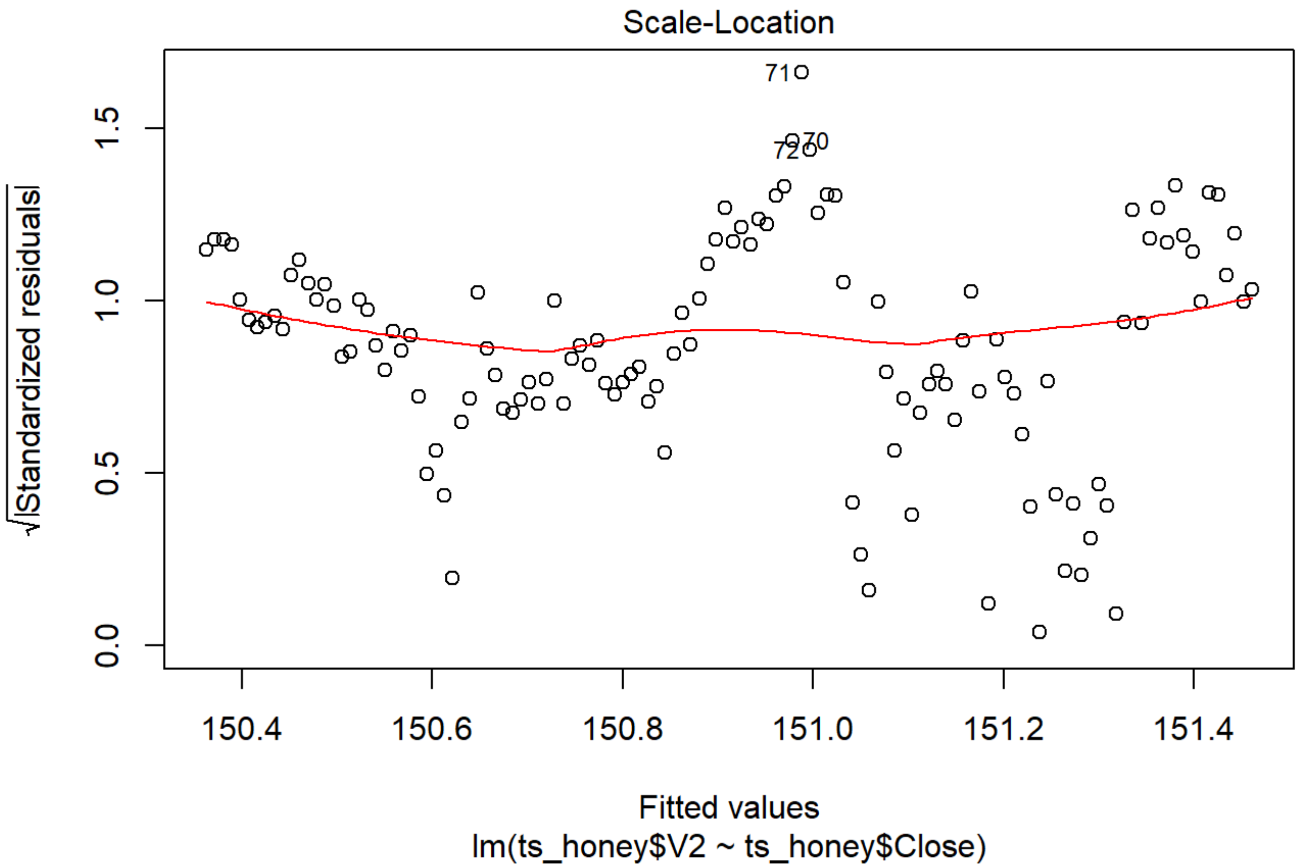


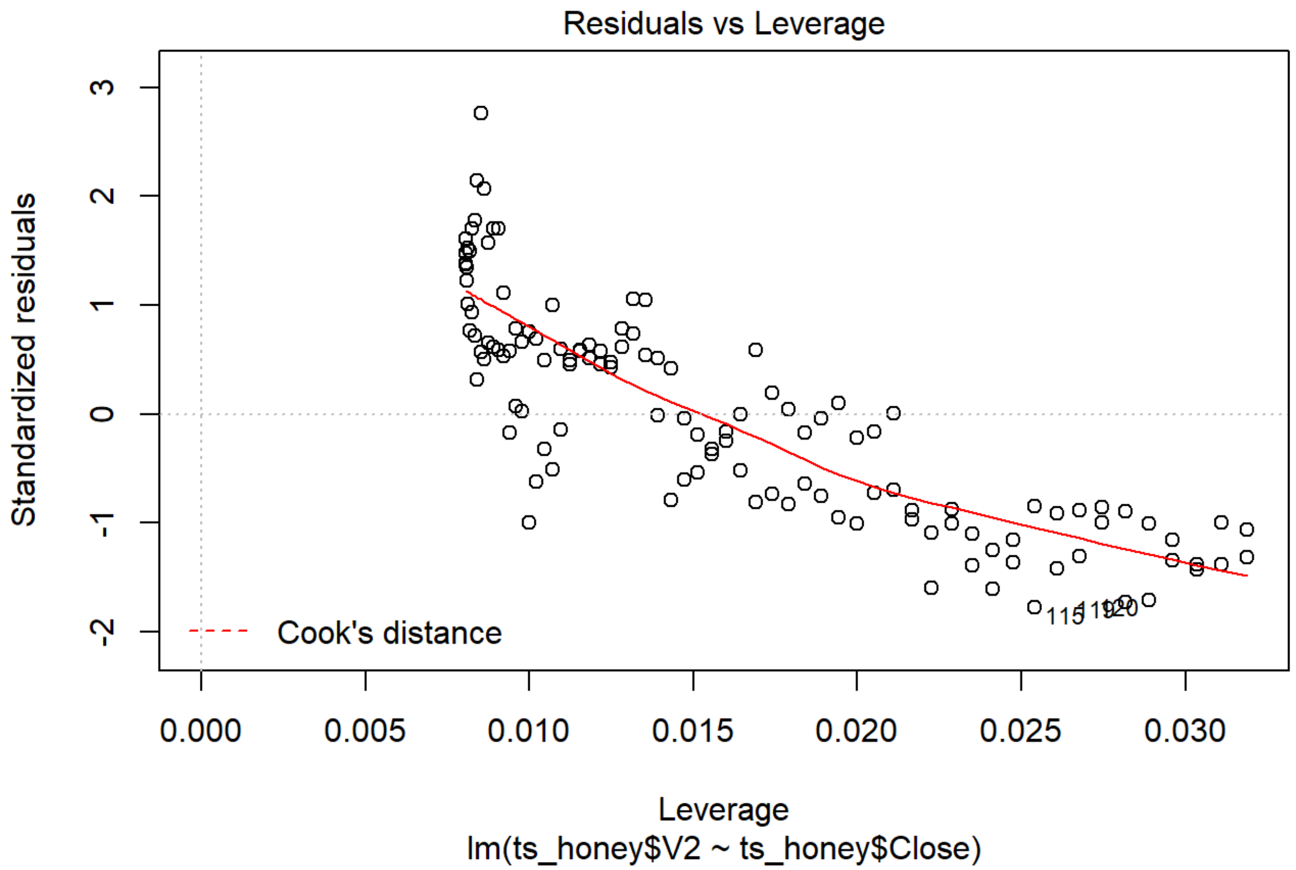
Performing the prediction using linear regression and comparing it with time series forecast.

```r
ts_honey <-  read.csv("Honeywell.csv")
ts_honey <- ts_honey[c(-1,-3)]
num <- seq(1:124)
ts_honey[,2] <- num
temp <- ts_honey[,1]
ts_honey[,1] <- ts_honey[,2]
ts_honey[,2] <- temp


df <- lm(ts_honey$V2 ~ ts_honey$Close)

plot(df)
```

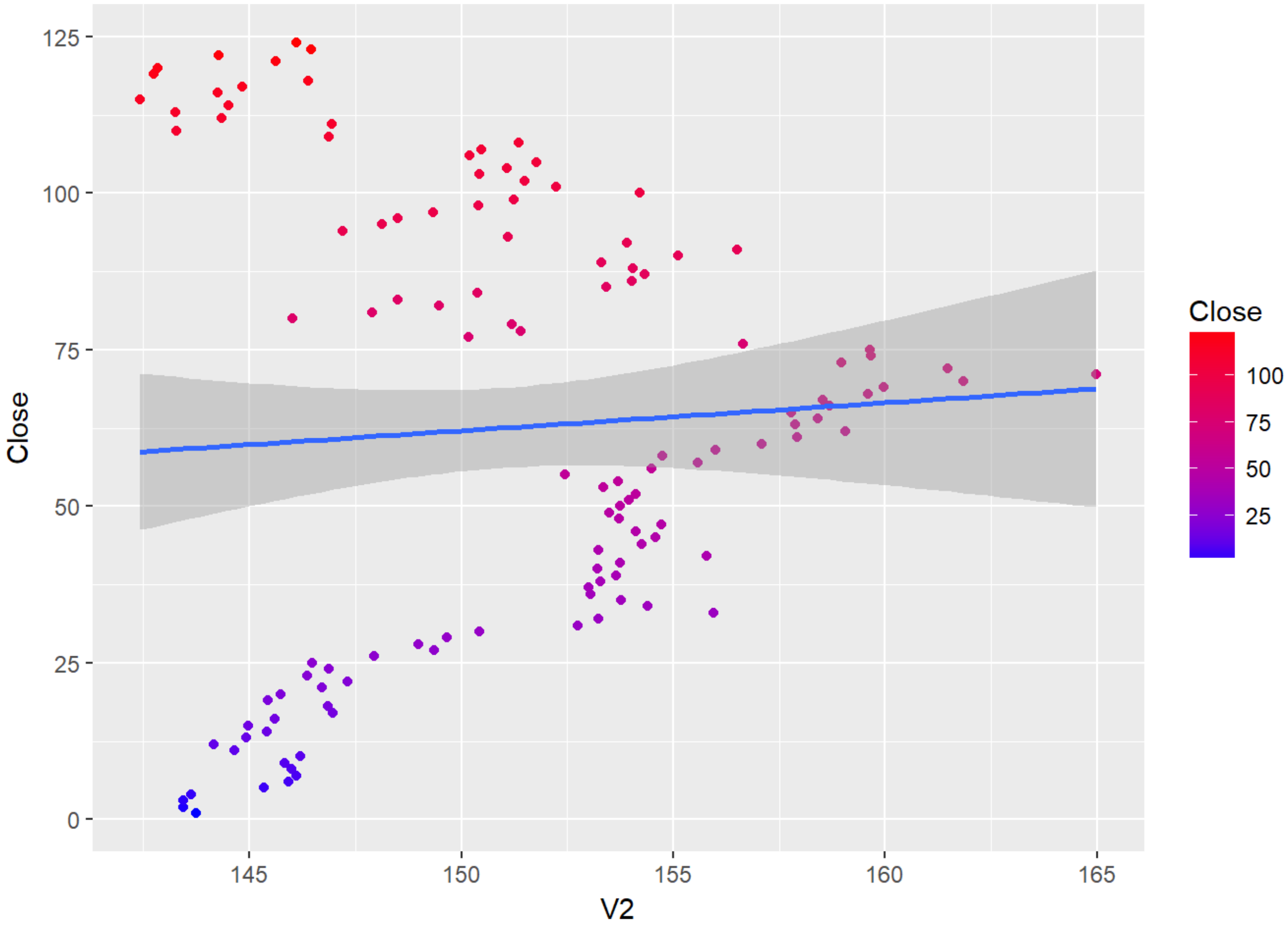



Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(ts_honey$V2 ~ ts_honey$Close)
71
72
70


Scale-Location
√|Standardized residuals|
Fitted values
lm(ts_honey$V2 ~ ts_honey$Close)
71
72
70




```
ggplot(ts_honey,aes(x=V2,y=Close)) + geom_point(aes(color = Close)) + scale_colour_
gradient(high='red',low = "blue") + geom_smooth(method='lm',formula=y~x)
```




```
summary(df)
```

```
##
## Call:
## lm(formula = ts_honey$V2 ~ ts_honey$Close)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.9502 -4.4379 -0.0406  3.2354 14.0026
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.504e+02  9.207e-01 163.307   <2e-16 ***
## ts_honey$Close 8.926e-03  1.278e-02   0.698    0.486
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.095 on 122 degrees of freedom
## Multiple R-squared:  0.003981,    Adjusted R-squared:  -0.004183
## F-statistic: 0.4876 on 1 and 122 DF,  p-value: 0.4863
```

```
new <- data.frame(x = seq(124, 247, 1))

prediction <- predict(df, newdata = new,se.fit = TRUE)

mean((ts_honey[,2] - prediction$fit)^2)
```

```
## [1] 25.54194
```

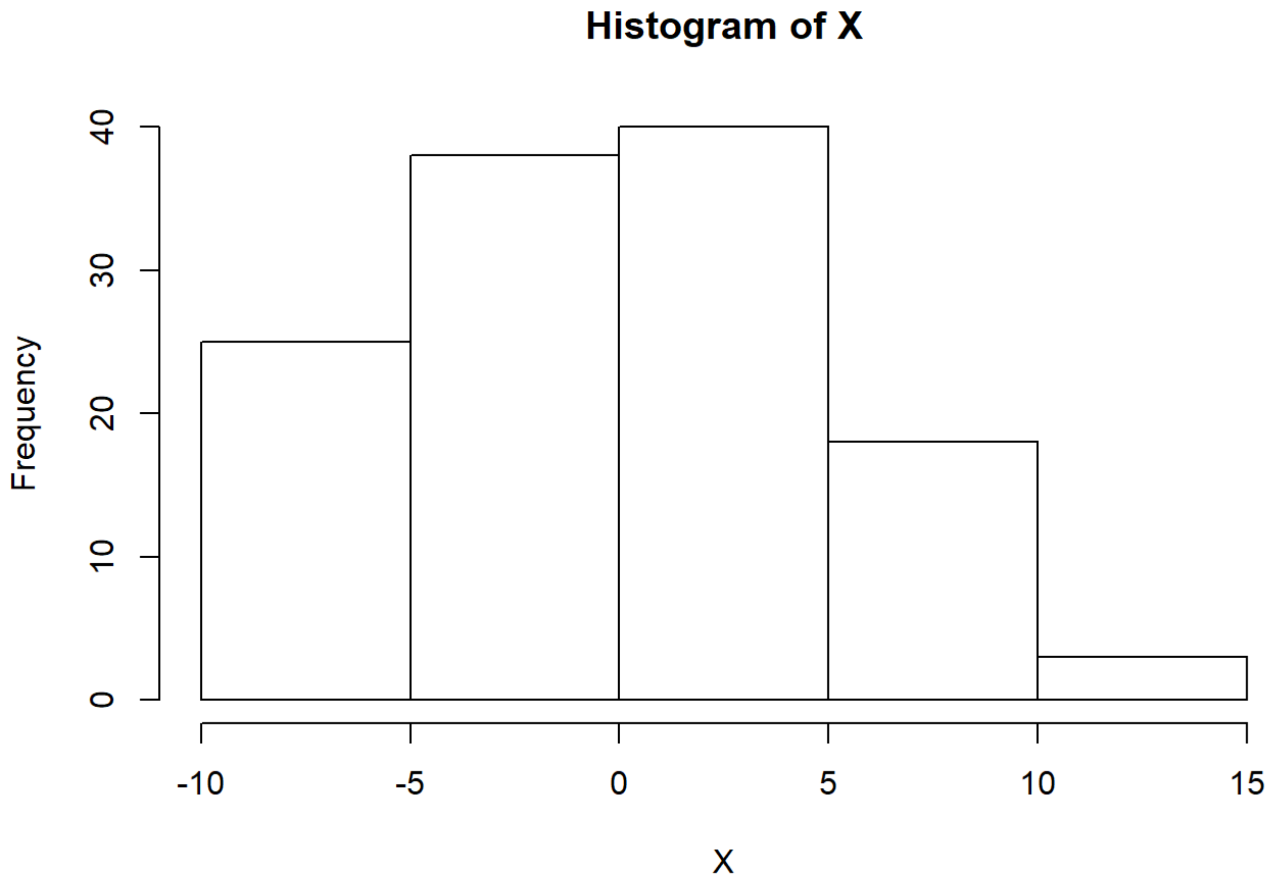
```
sse <- sum(df$residuals^2)

sse
```

```
## [1] 3167.201
```

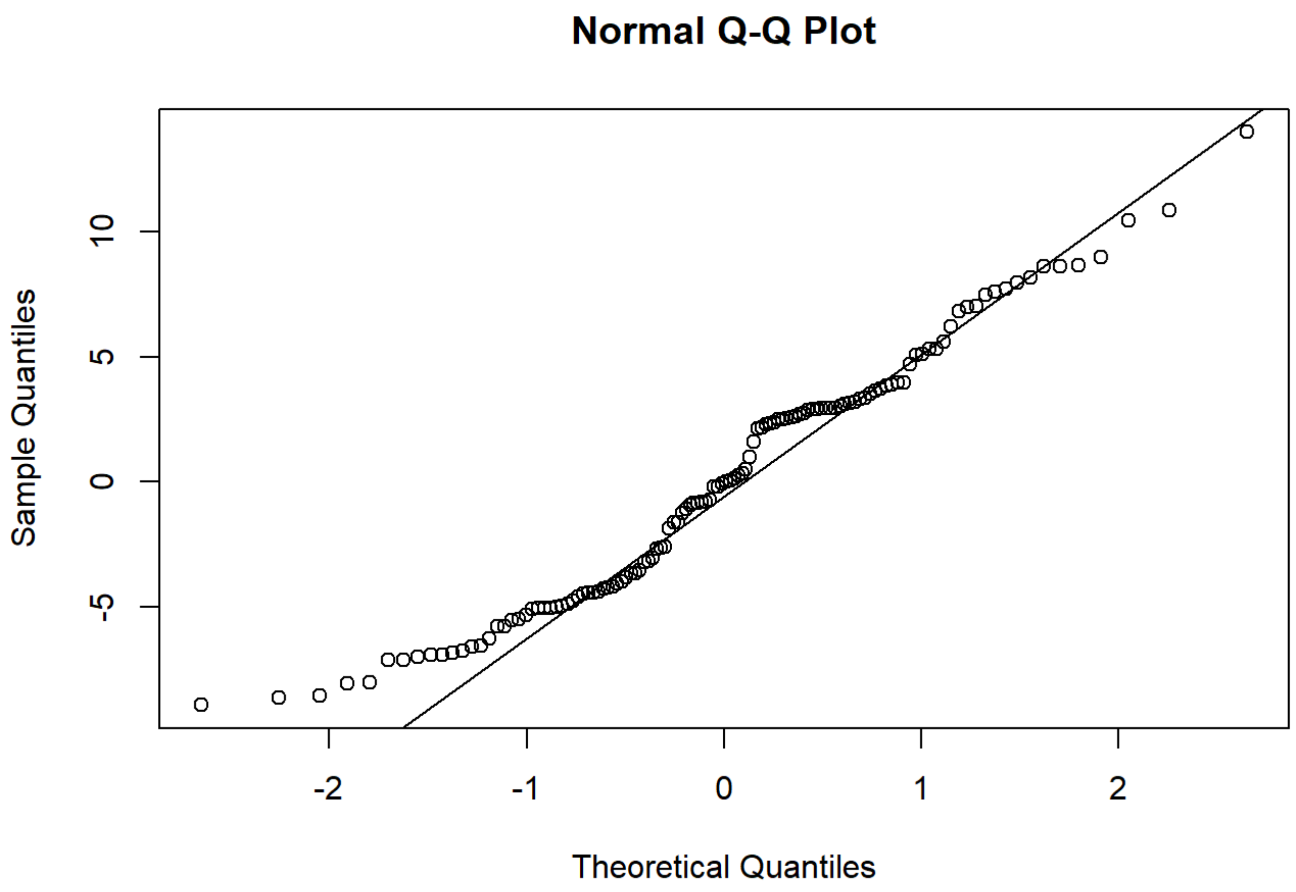
```
summary(df)
```

```
##
## Call:
## lm(formula = ts_honey$V2 ~ ts_honey$Close)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.9502 -4.4379 -0.0406  3.2354 14.0026
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.504e+02  9.207e-01 163.307   <2e-16 ***
## ts_honey$Close 8.926e-03  1.278e-02   0.698    0.486
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.095 on 122 degrees of freedom
## Multiple R-squared:  0.003981,   Adjusted R-squared:  -0.004183
## F-statistic: 0.4876 on 1 and 122 DF,  p-value: 0.4863
```

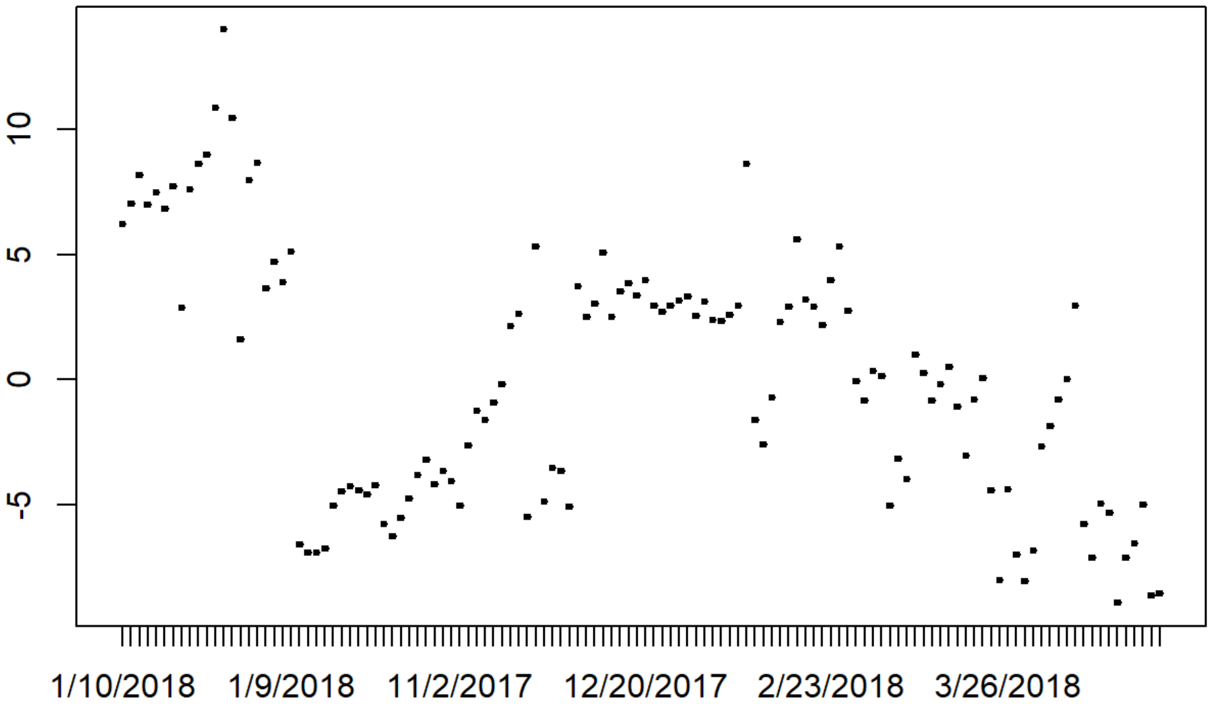
```
X <- resid(df)

hist(X)
```




```
View(X)

#Plotting probability plot
qqnorm(X)
qqline(X)
```



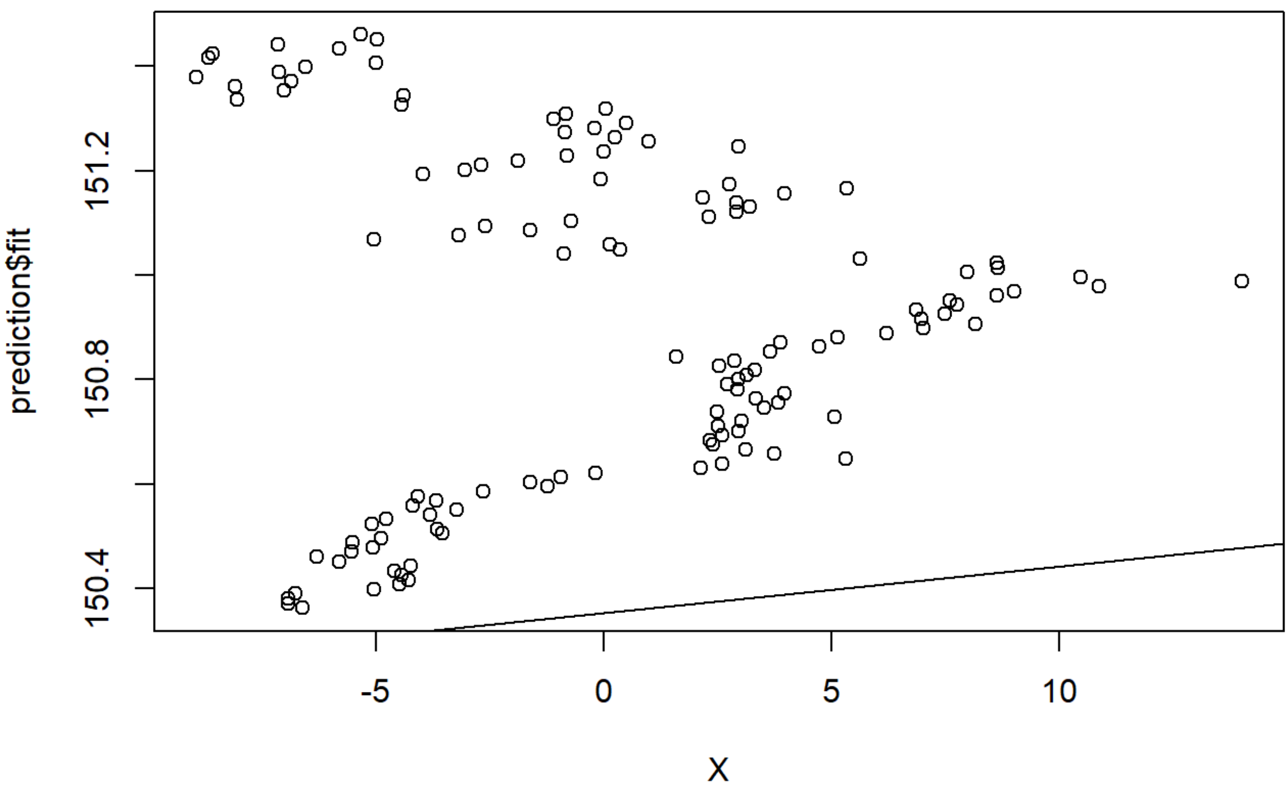

```
ts_honey <-  read.csv("Honeywell.csv")

plot(ts_honey$Date,X, type = "p")
```




```
prediction$fit
```

```
##        1        2        3        4        5        6        7        8 
## 150.3626 150.3715 150.3804 150.3893 150.3983 150.4072 150.4161 150.4251 
##        9       10       11       12       13       14       15       16 
## 150.4340 150.4429 150.4518 150.4608 150.4697 150.4786 150.4875 150.4965 
##       17       18       19       20       21       22       23       24 
## 150.5054 150.5143 150.5232 150.5322 150.5411 150.5500 150.5589 150.5679 
##       25       26       27       28       29       30       31       32 
## 150.5768 150.5857 150.5946 150.6036 150.6125 150.6214 150.6304 150.6393 
##       33       34       35       36       37       38       39       40 
## 150.6482 150.6571 150.6661 150.6750 150.6839 150.6928 150.7018 150.7107 
##       41       42       43       44       45       46       47       48 
## 150.7196 150.7285 150.7375 150.7464 150.7553 150.7642 150.7732 150.7821 
##       49       50       51       52       53       54       55       56 
## 150.7910 150.8000 150.8089 150.8178 150.8267 150.8357 150.8446 150.8535 
##       57       58       59       60       61       62       63       64 
## 150.8624 150.8714 150.8803 150.8892 150.8981 150.9071 150.9160 150.9249 
##       65       66       67       68       69       70       71       72 
## 150.9338 150.9428 150.9517 150.9606 150.9696 150.9785 150.9874 150.9963 
##       73       74       75       76       77       78       79       80 
## 151.0053 151.0142 151.0231 151.0320 151.0410 151.0499 151.0588 151.0677 
##       81       82       83       84       85       86       87       88 
## 151.0767 151.0856 151.0945 151.1034 151.1124 151.1213 151.1302 151.1392 
##       89       90       91       92       93       94       95       96 
## 151.1481 151.1570 151.1659 151.1749 151.1838 151.1927 151.2016 151.2106 
##       97       98       99      100      101      102      103      104 
## 151.2195 151.2284 151.2373 151.2463 151.2552 151.2641 151.2730 151.2820 
##      105      106      107      108      109      110      111      112 
## 151.2909 151.2998 151.3088 151.3177 151.3266 151.3355 151.3445 151.3534 
##      113      114      115      116      117      118      119      120 
## 151.3623 151.3712 151.3802 151.3891 151.3980 151.4069 151.4159 151.4248 
##      121      122      123      124 
## 151.4337 151.4426 151.4516 151.4605
```

```r
plot(X,prediction$fit, type = "p")
abline(df)
```



prediction$fit
151.2
150.8
150.4
-5
0
5
10
X