Residuals

Pawan Kumar

30/05/2021

Contents

```r
# attaching course package
library(fpp2)
```

Residuals given by: $e_t = y_t$ - $y_{forecasted}$

Properties of a good model:

1. The residuals are uncorrelated. If there are correlations between residuals, then there is information left in the residuals which should be used in computing forecasts.
2. The residuals have zero mean. If the residuals have a mean other than zero, then the forecasts are biased. *Note: Adjusting for bias is easy: if the residuals have mean not equal to zero, then simply add the mean of the r*
3. Residuals should have constant variance
4. The residuals should be normally distributed

Data Downloading: “Quandl”

Steps:

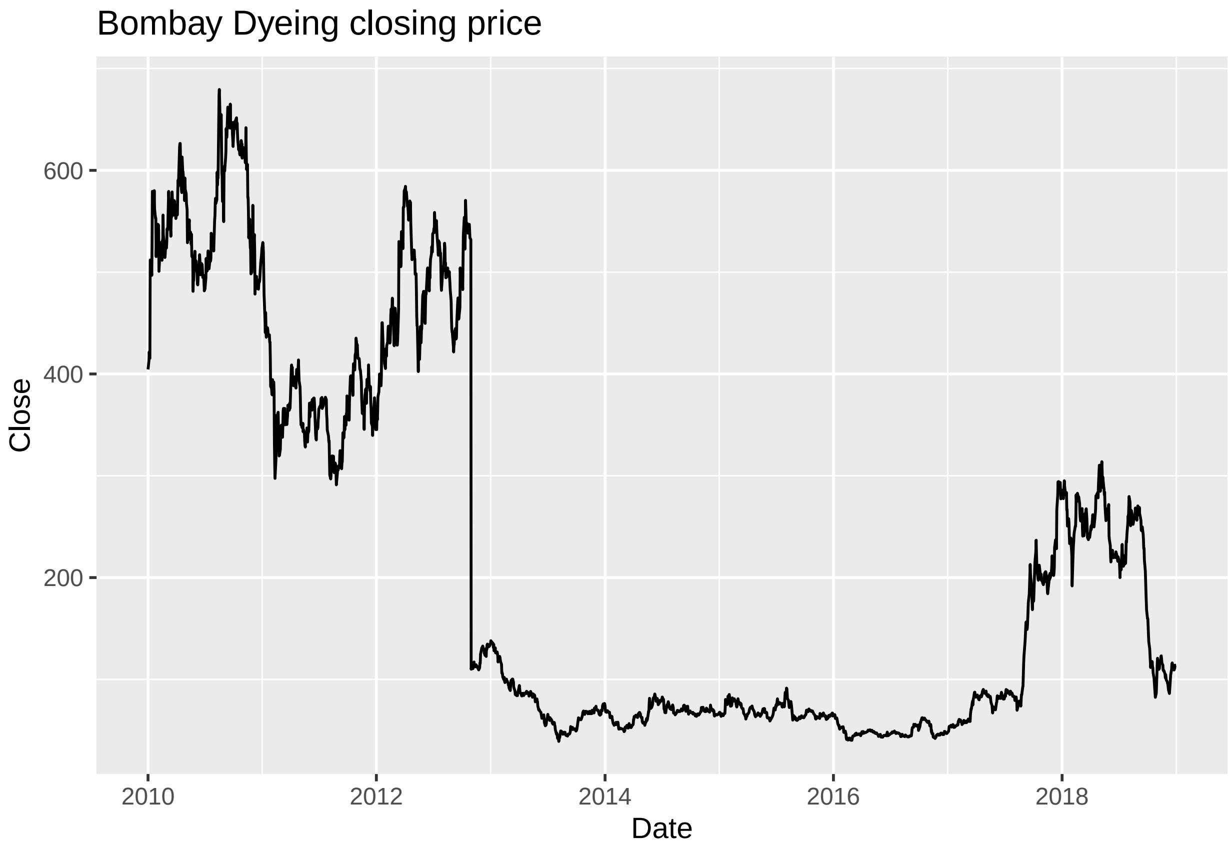
1. Create account at Quandl
2. The first time you will make account, a api key will be created
3. Choose the data type or the region for data fetching
4. Link for R related codes to fetch data - R_Quandl

```r
# Data downloading using "Quandl"

# Installation - install.packages("Quandl")

# Loading the package
library(Quandl)
# downloading data for Apple
bombay_dying <- Quandl("XBOM/500020_UADJ", api_key="Dyqv-xPx3cgEbv31yVh9")
```

```
## Extract the closing price i.e.

bombay_dying_close <- bombay_dying[,c(1,5)]
```

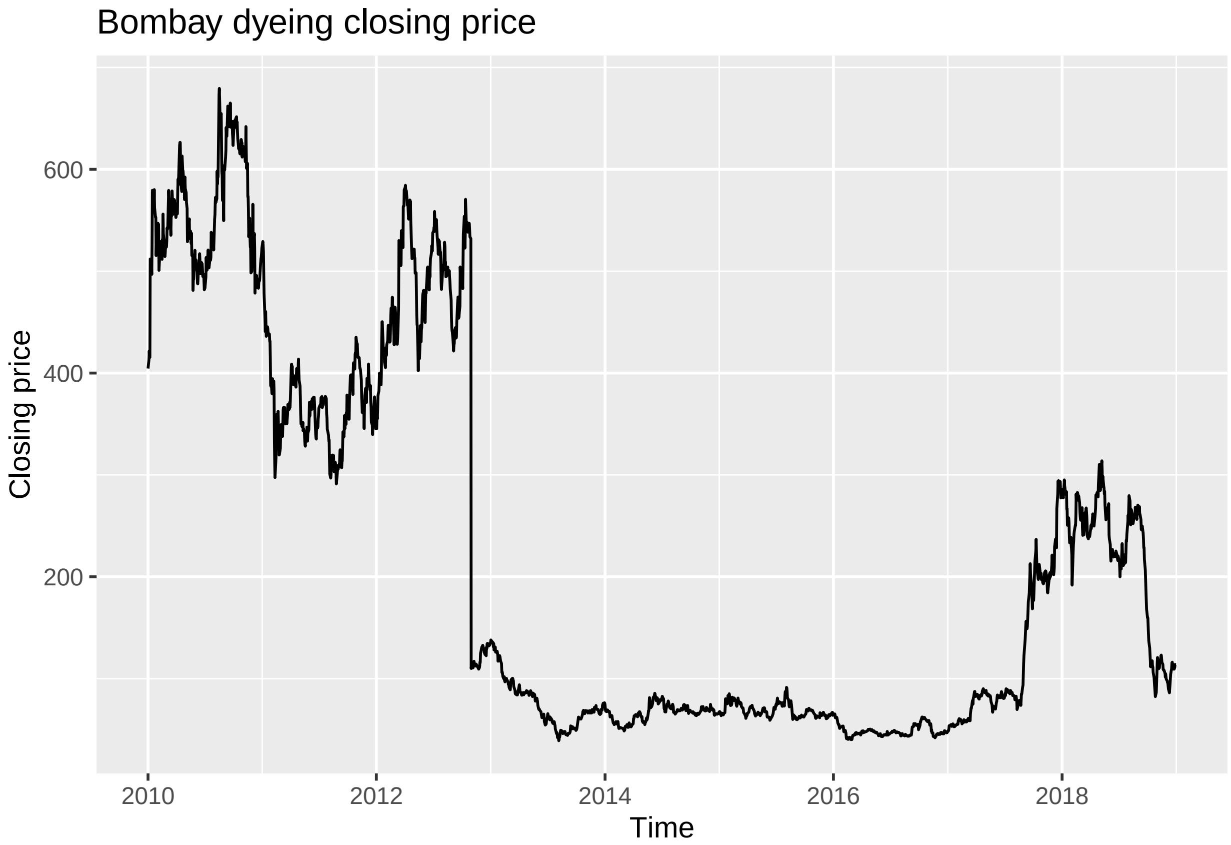
Watch out for the error, to plot a time series data we need to either convert it into type ***“xts” or ’zoo"*** - a datatype for time series data or use ***ggplot*** which uses data as a ***dataframe***

```
# use ggplot to plot the time series
library(ggplot2)
ggplot(data = bombay_dying_close, aes(x = Date, y = Close)) + geom_line() + ggtitle("Bombay Dyeing clos
```



Convert to type “xts” and then plotting

```
# install package "tidyquant"
# import the package thereafter
library(tidyquant)
library(timetk)
library(magrittr)

close_price <- bombay_dying_close %>% tk_xts()

View(close_price)

autoplot(close_price) + xlab("Time") + ylab("Closing price") + ggtitle("Bombay dyeing closing price")
```

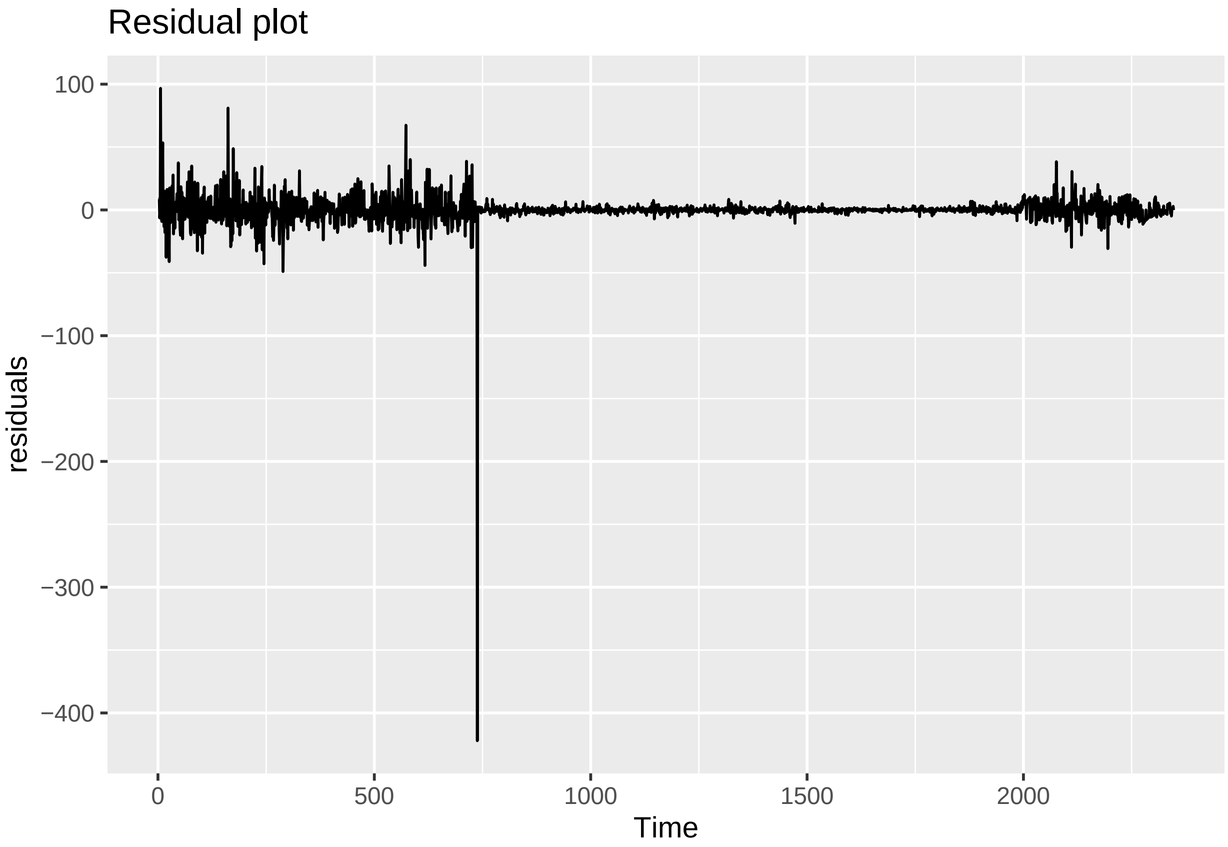


Note: For most of the time series data, the object that goes as an argument is "xts" so better to convert a dataframe to xts.

Plotting the residuals

Most common method deployed for daily time series of equity data is "naive" method.

```
# using naive method

forecast <- naive(close_price)

residual_value <- residuals(forecast)

autoplot(residual_value) + xlab("Time") + ylab("residuals") + ggtitle("Residual plot")
```

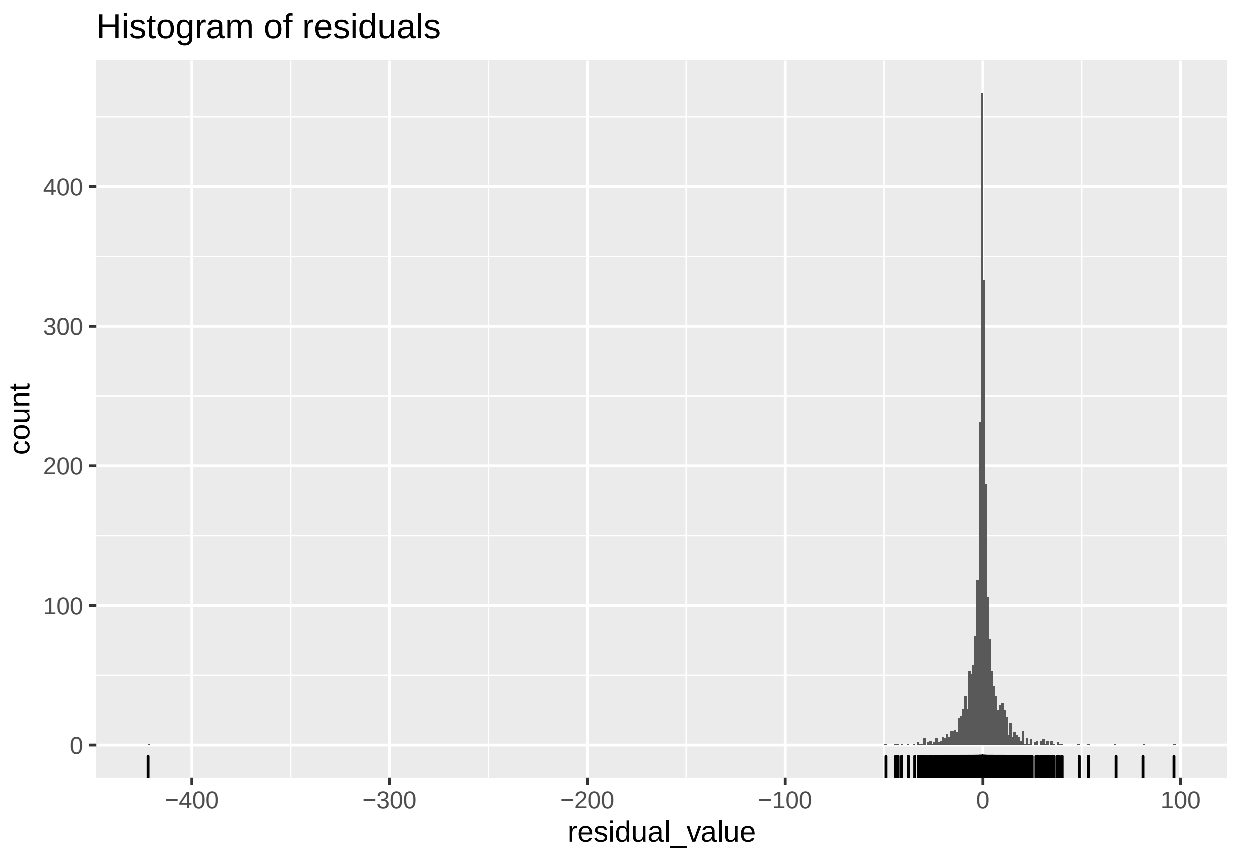


As we can see, that mean is close “zero” barring few outliers.

Residual diagnostic

1. Checking the normality of residuals using the histogram

```
gghistogram(residual_value) + ggtitle("Histogram of residuals")
```



As we can observe that the histogram somewhat resembles bell shape curve. Further statistical test can be carried out to check for normality ***"Shapiro Wilk"*** test

H_o : The data is normally distributed

H_a : The data is not normally distributed

If, p-value is less than the tolerance level α, then reject the null hypothesis.

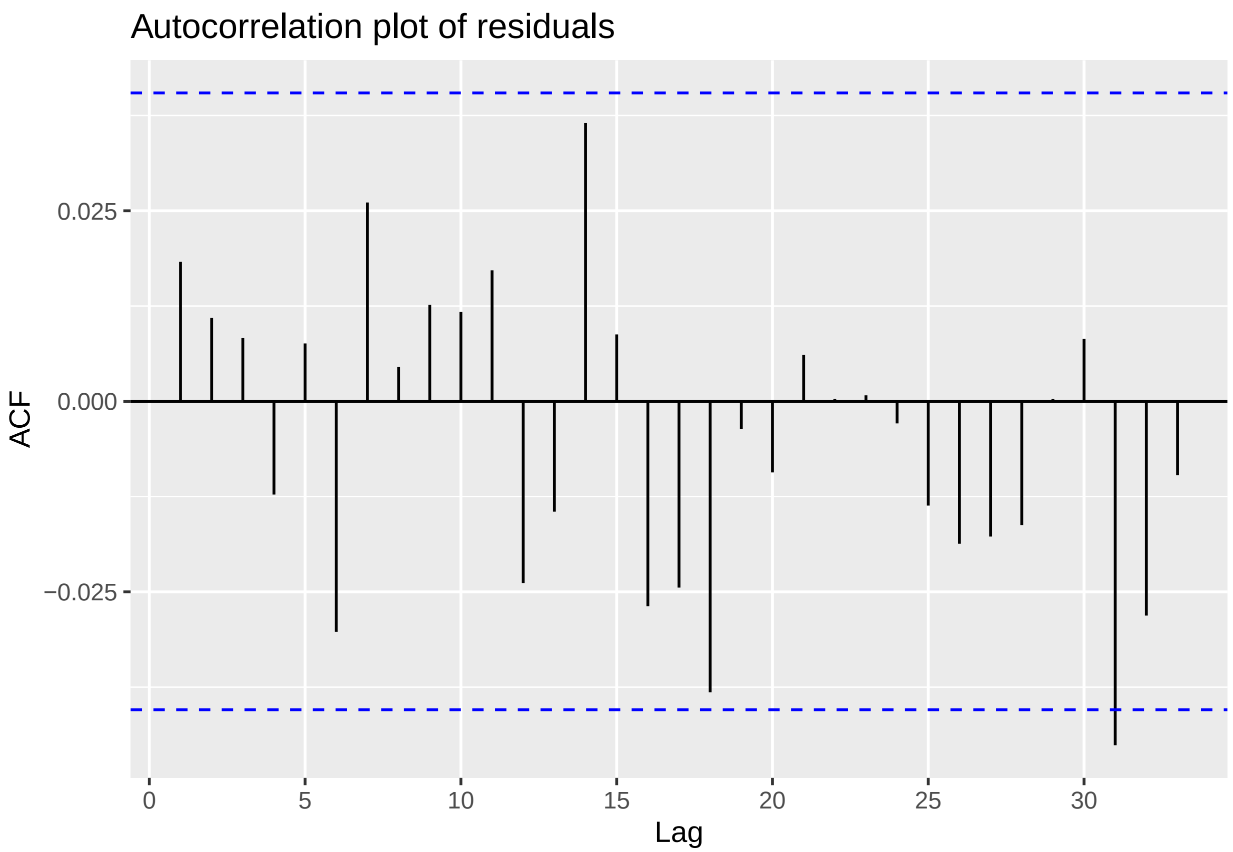
```
shapiro.test(residual_value)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residual_value
## W = 0.4395, p-value < 2.2e-16
```

Since we can reject the null hypothesis, signifying the data deviates from normal distribution.

2. Correlation analysis amongst the residuals - using autocorrelation

```
ggAcf(residual_value) + ggtitle("Autocorrelation plot of residuals")
```



Portmanteau test for autocorrelation: A test for a group of autocorrelations is called a portmanteau test. It tests whether, the first "h" autocorrelations are significantly different from what would be expected from a white noise process.

One such Portmanteau test is **Box-Pierce** test

Box-Pierce Test

Q = T * $\sum_{k=1}^{h} r_k^2$

where, "h" is the maximum lag taken and T is number of observation. The value of "Q" depends on each autocorrelation values, in case each r_k is close to zero, the aggregate would be close to zero. However, in case of outliers the value will be much higher.

Advisory:

a. For non-seasonal data - h = 10

b. For seasonal data - h = 2m, where m is the period of seasonality.

Important: In case the "h" is very large use h = T/5 to perform the test. (In case of lag detection it may be more than 10)

$\mathrm{H_0}$: is that our model *does not* show lack of fit (or in simple terms—the model is just fine).

$\mathrm{H_a}$: is just that the model *does* show a lack of fit.

acceptance or rejection depends on comparing the p-value with tolerance α.

```
# As there is no seasonality, we opt for lag length h = 10
Box.test(residual_value, lag = 10, type = "Box-Pierce", fitdf = 0)
```

```
##
##  Box-Pierce test
##
## data:  residual_value
## X-squared = 6.2041, df = 10, p-value = 0.7978
```

So we cannot reject the null hypothesis as p-values is not less than $\alpha = 5\%$

Ljung-Box Test

A more accurate test to check autocorrelation.

Q' = T(T+2) $\sum_{k=1}^{h} (T-k)^{-1} r_k^2$

A larger value indicate that the autocorrelation is not coming from a white noise i.e. its not just a random observation.

Both Q and Q' follow a ??2 distribution with (h-k) degrees of freedom. “k” is number of parameters in the model. In case if it is calculated fro raw data, rather than an output of a model, then put k = 0.

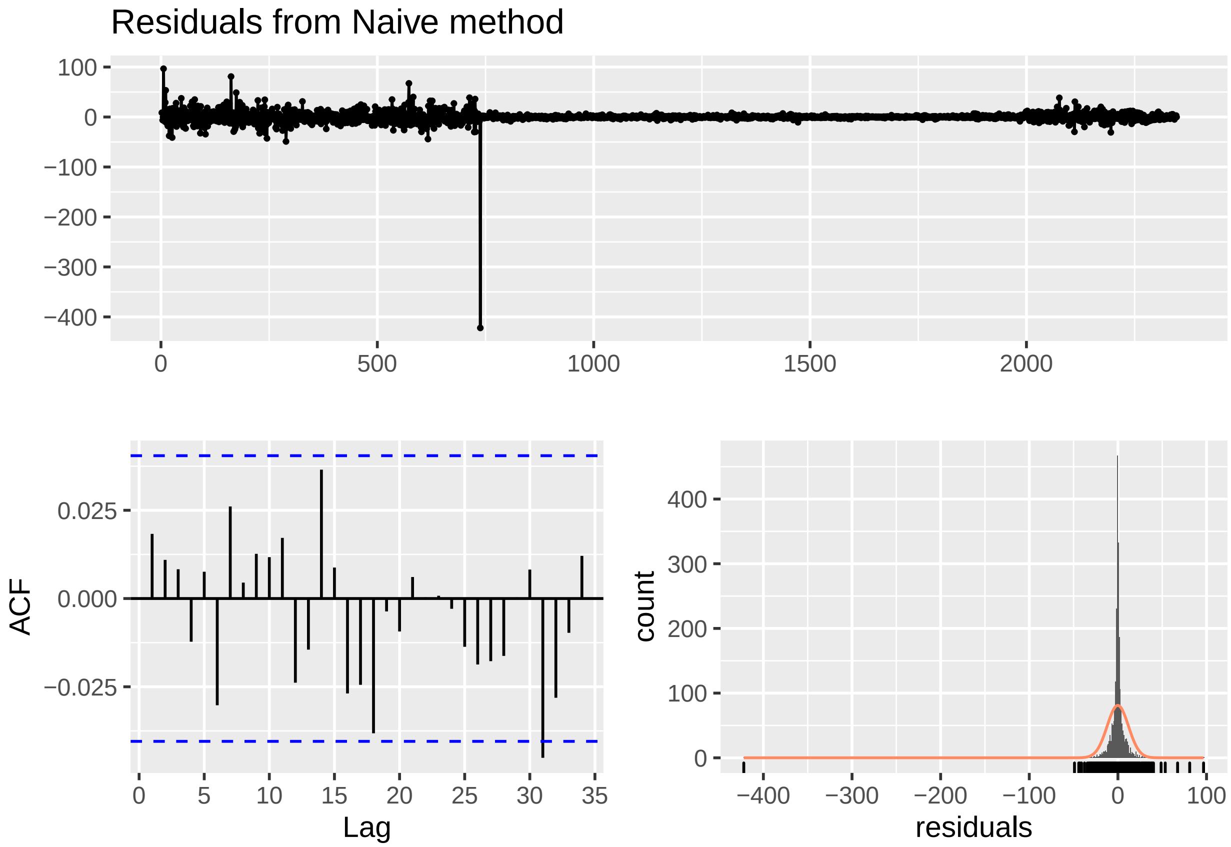
```
Box.test(residual_value, lag = 10, type = "Ljung-Box", fitdf = 0)
```

```
##
##  Box-Ljung test
##
## data:  residual_value
## X-squared = 6.2243, df = 10, p-value = 0.7961
```

Thus, the residuals show no significant deviation from model fit. It signifies that relevant information has been extracted. However, normality still remains the concern.

Tip -> For autocorrelation diagnostics we had to separately plot the correlogram and then perform portmanteau test (Box-Pierce & Ljung-Box) . However, a R package exists that combines the plot and test together.

```
checkresiduals(forecast)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from Naive method
## Q* = 6.2243, df = 10, p-value = 0.7961
##
## Model df: 0.   Total lags used: 10
```