NJ Homes Listing Prices Analysis

Rutwik Guntoorkar

11-08-2022

1. Introduction

Zillow, a popular real estate platform, provides monthly indices for median home prices. Using this data, we analyze the median home prices for house listings in New Jersey. This report provides insights into the time series analysis and forecasting techniques employed to understand trends and make future predictions.

Data Source The data was retrieved from Zillow Research and includes monthly median home prices for New Jersey from April 1996 onwards.

```r
library(fpp)
```

Importing Data

```
## Loading required package: forecast

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

## Loading required package: fma

## Loading required package: expsmooth

## Loading required package: lmtest

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: tseries
```

```r
library(fpp2)
```

```
## -- Attaching packages --------------------------------------------- fpp2 2.5 --

## v ggplot2 3.4.0

##

##
## Attaching package: 'fpp2'
```

```
## The following objects are masked from 'package:fpp':
##
##     ausair, ausbeer, austa, austourists, debitcards, departures,
##     elecequip, euretail, guinearice, oil, sunspotarea, usmelec
```

```r
library(TTR)
```

```
## Warning: package 'TTR' was built under R version 4.2.3
```

```r
library(ggplot2)
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(readr)

library(readr)
NJ_MedianListingPrice_AllHomes <- read_csv("/Users/rutwik/Desktop//NJ_MedianListingPrice_AllHomes.csv")
```

```
## Rows: 257 Columns: 2

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): YYYY-MM
## dbl (1): Value
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
names(NJ_MedianListingPrice_AllHomes)
```

```
## [1] "YYYY-MM" "Value"
```

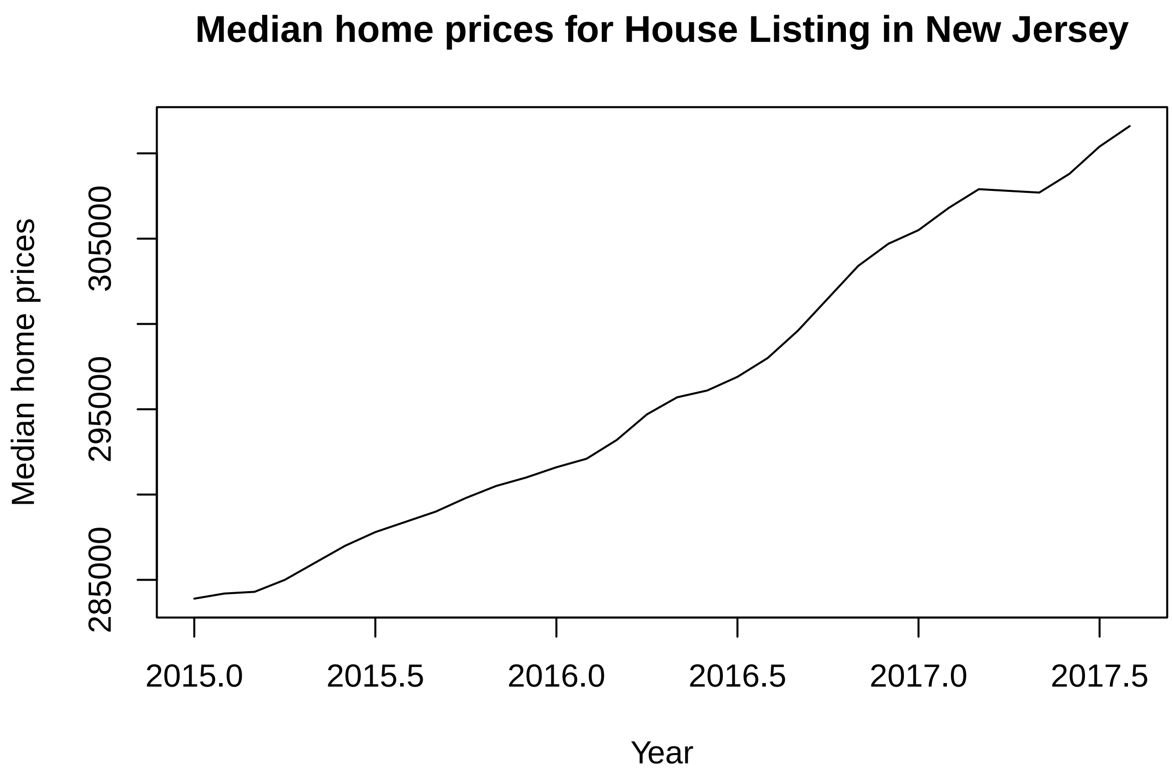
```r
NJ_Home_Raw <- NJ_MedianListingPrice_AllHomes$Value
NJ_Home_TS <- ts(NJ_Home_Raw,frequency = 12, start = c(1996,4))
```

Plot and Inference

```r
window(NJ_Home_TS,2015)
```

```
##         Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct
## 2015 283900 284200 284300 285000 286000 287000 287800 288400 289000 289800
## 2016 291600 292100 293200 294700 295700 296100 296900 298000 299600 301500
## 2017 305500 306800 307900 307800 307700 308800 310400 311600
##         Nov    Dec
## 2015 290500 291000
## 2016 303400 304700
## 2017
```

```r
data <- window(NJ_Home_TS,2015)
##Time Series Plot
plot(data,main = 'Median home prices for House Listing in New Jersey', xlab = 'Year', ylab = 'Median hom
```



Time Series Plot The time series plot shows the variation in median home prices from April 1996 to August 2017. The prices exhibited an increasing trend over the period. The data post-2015 was used for detailed analysis due to a significant market disruption during the 2008 economic crisis.

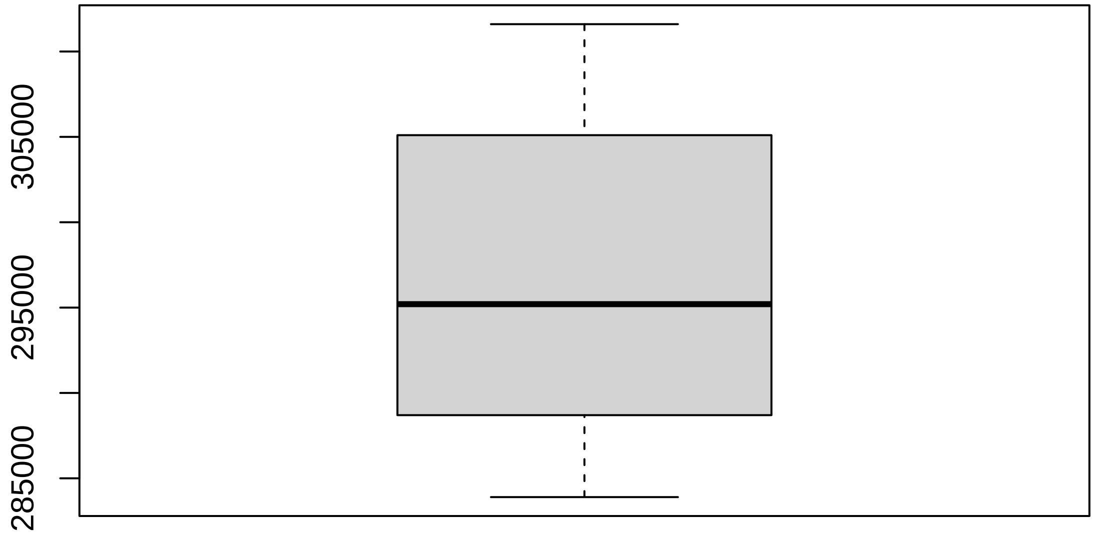
Observation: The home prices show a steady increasing trend from 2015 to 2017. There are seasonal variations indicating cyclic patterns in certain months.

Central Tendency

```r
summary(data)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  283900  288850  295200  296278  304900  311600
```

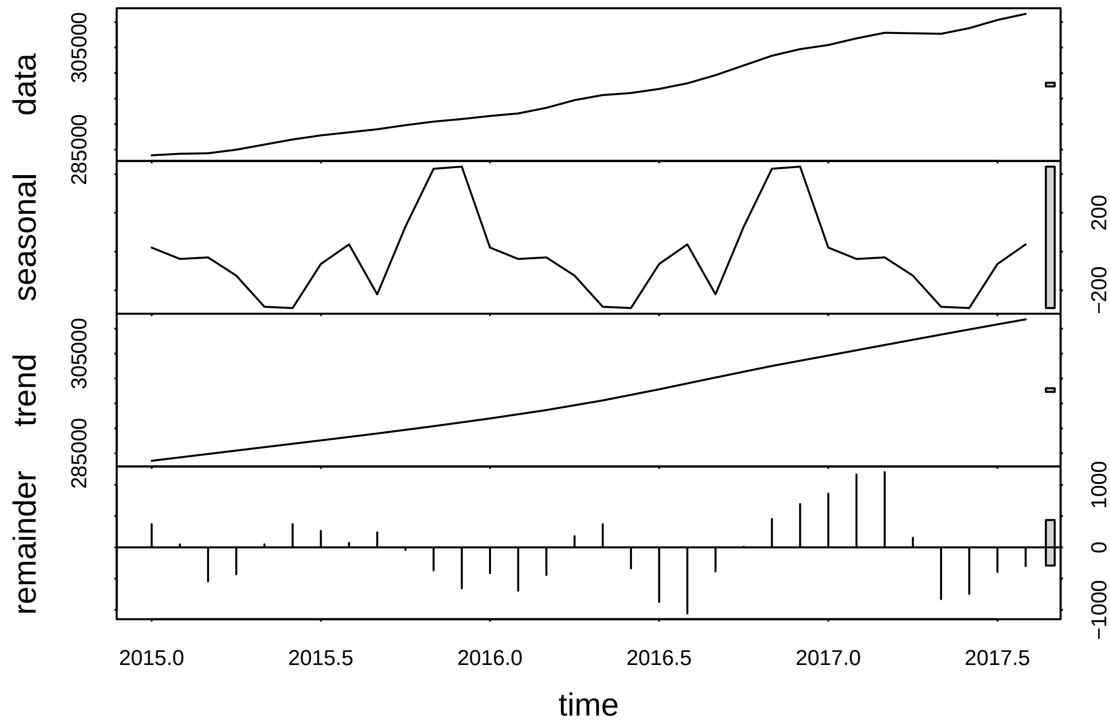
```r
boxplot(data)
```



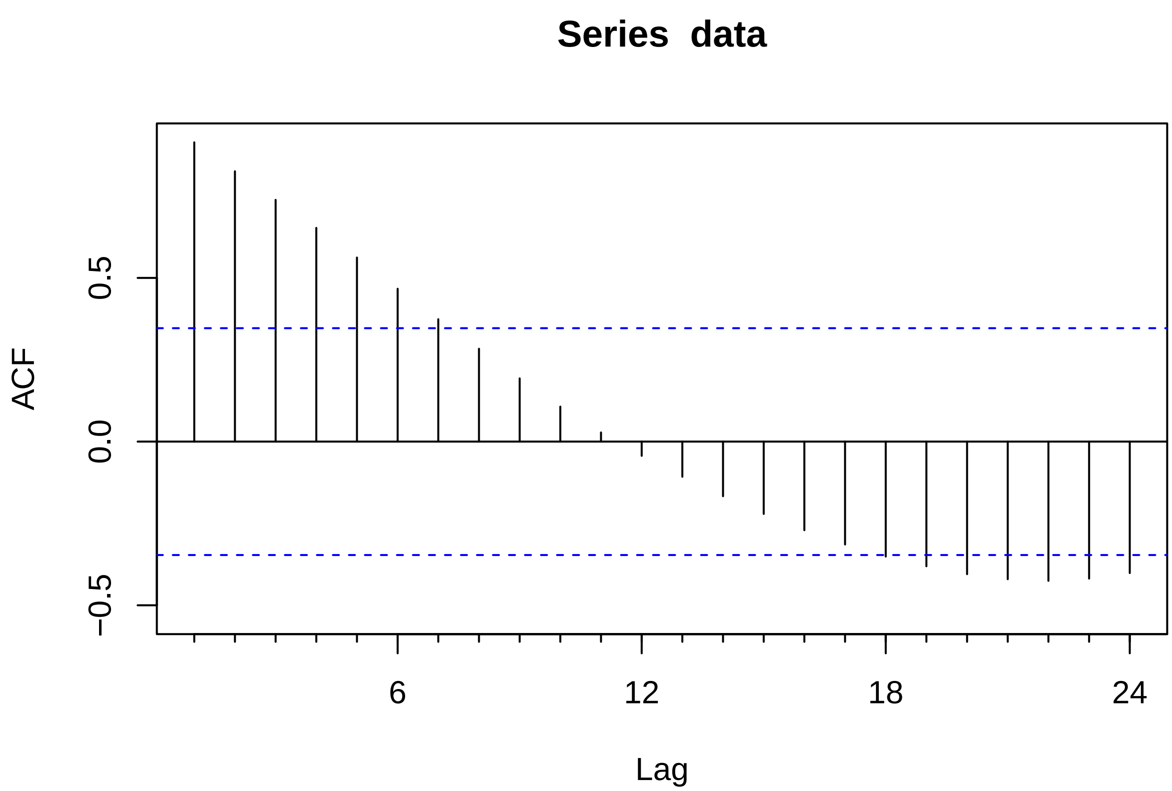
- The box plot for the given time series data shows us that the prices varies from min value (285000 to max value (305000)
- IQR ranges from 197800 to 307900
- As mean price is 263499 and median is 280200, The distribution is left skewed
- Also boxplot suggests there are no outliers

Decomposition

```
library(fpp)
library(fpp2)

decomp_data <- stl(data,s.window ="periodic")

#Plot of decomposition of timeseries.
plot(decomp_data)
```




```r
Acf(data)
```




```r
decomp_data
```

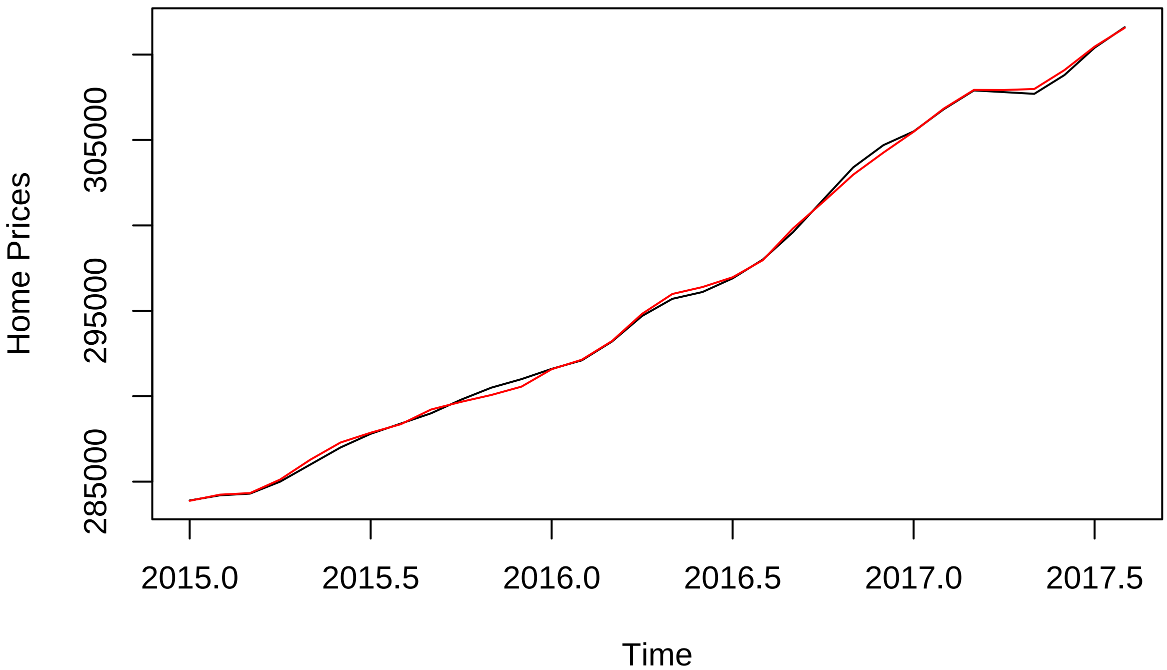
```
## Call:
## stl(x = data, s.window = "periodic")
##
```

```
## Components
##             seasonal    trend    remainder
## Jan 2015   20.74073 283506.9   372.390604
## Feb 2015  -37.92284 284190.1    47.794900
## Mar 2015  -29.91936 284873.4  -543.467855
## Apr 2015 -123.89261 285554.8  -430.942843
## May 2015 -284.53284 286236.3    48.249148
## Jun 2015 -290.87452 286918.0   372.844768
## Jul 2015  -63.88274 287599.8   264.106934
## Aug 2015   37.18421 288290.3    72.477853
## Sep 2015 -220.33493 288980.9   239.434852
## Oct 2015  128.35377 289711.7   -40.078086
## Nov 2015  427.04246 290442.5  -369.591025
## Dec 2015  438.03794 291217.7  -655.754020
## Jan 2016   20.74073 291992.9  -413.624339
## Feb 2016  -37.92284 292833.0  -695.043626
## Mar 2016  -29.91936 293673.0  -443.129964
## Apr 2016 -123.89261 294643.9   179.958943
## May 2016 -284.53284 295614.8   369.714828
## Jun 2016 -290.87452 296726.8  -335.933402
## Jul 2016  -63.88274 297838.8  -874.915086
## Aug 2016   37.18421 299022.0 -1059.202254
## Sep 2016 -220.33493 300205.2  -384.903342
## Oct 2016  128.35377 301361.8     9.871039
## Nov 2016  427.04246 302518.3   454.645420
## Dec 2016  438.03794 303568.4   693.537037
## Jan 2017   20.74073 304618.5   860.721331
## Feb 2017  -37.92284 305671.9  1166.019426
## Mar 2017  -29.91936 306725.3  1204.650471
## Apr 2017 -123.89261 307769.5   154.364218
## May 2017 -284.53284 308813.8  -829.255057
## Jun 2017 -290.87452 309835.9  -745.074794
## Jul 2017  -63.88274 310858.1  -394.227984
## Aug 2017   37.18421 311862.9  -300.048292
```

Time series decomposition helps separate the time series into trend, seasonality, and residual components. This analysis revealed that: The seasonal component is stable across the years. The trend component shows a gradual upward movement. The residual component indicates variations not captured by trend or seasonality.

Additive or Multiplicative Decomposition: The decomposition appears to be additive because the seasonal variation remains consistent regardless of changes in the trend.

```r
SATS <- seasadj(decomp_data)
plot(data, ylab="Home Prices")
lines(SATS, col="red")
```

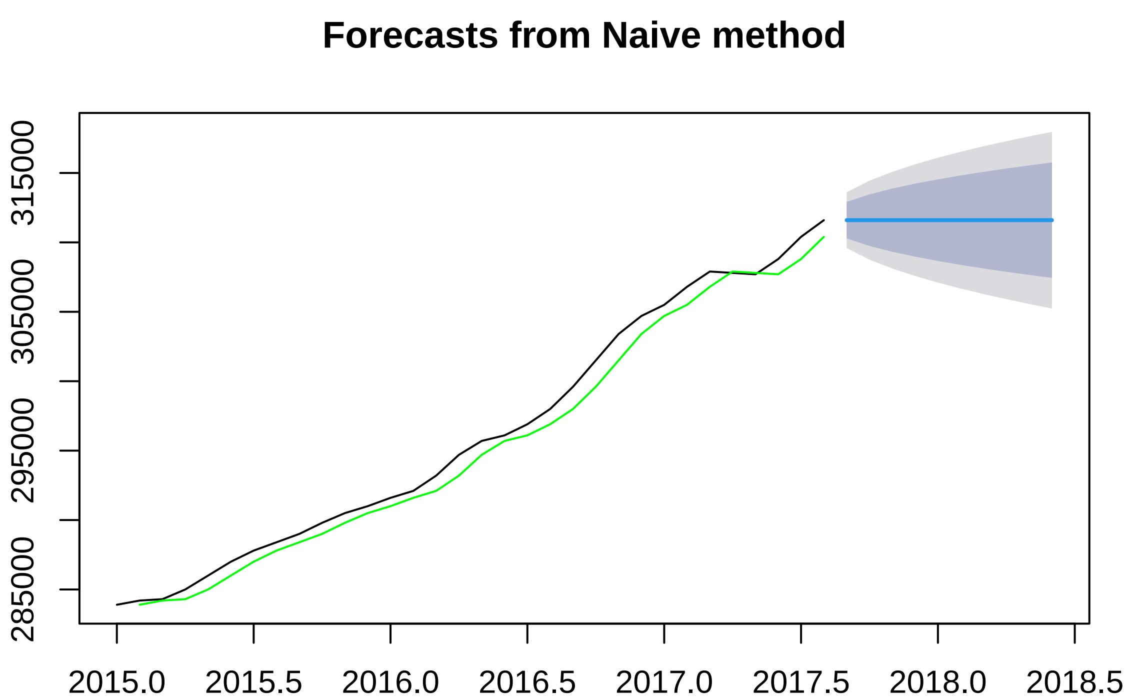


After overlaying the plot for time series adjusted for seasonality and the line for actual time series we can see that it somewhats fits the actual line, and there is negligible fluctuations at any point.

Forecasting Methods and Residual Analysis

Naïve Method

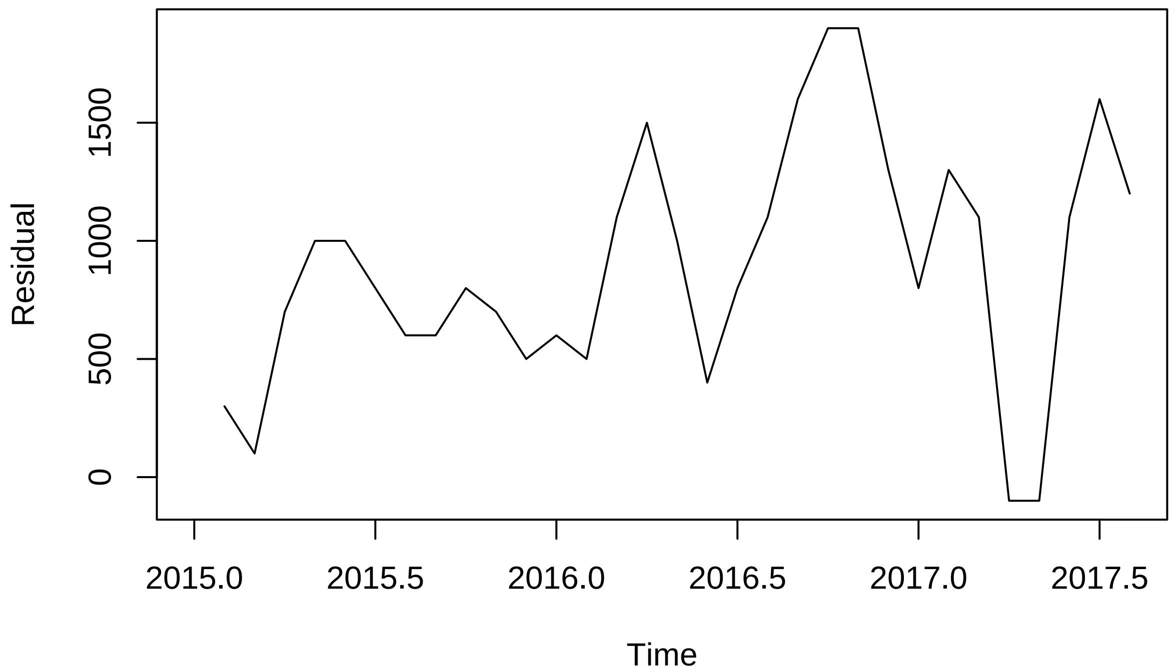
```
#naives bayes

naive_fc = naive(data)
plot(naive_fc)
lines(naive_fc$fitted, col="green")
```



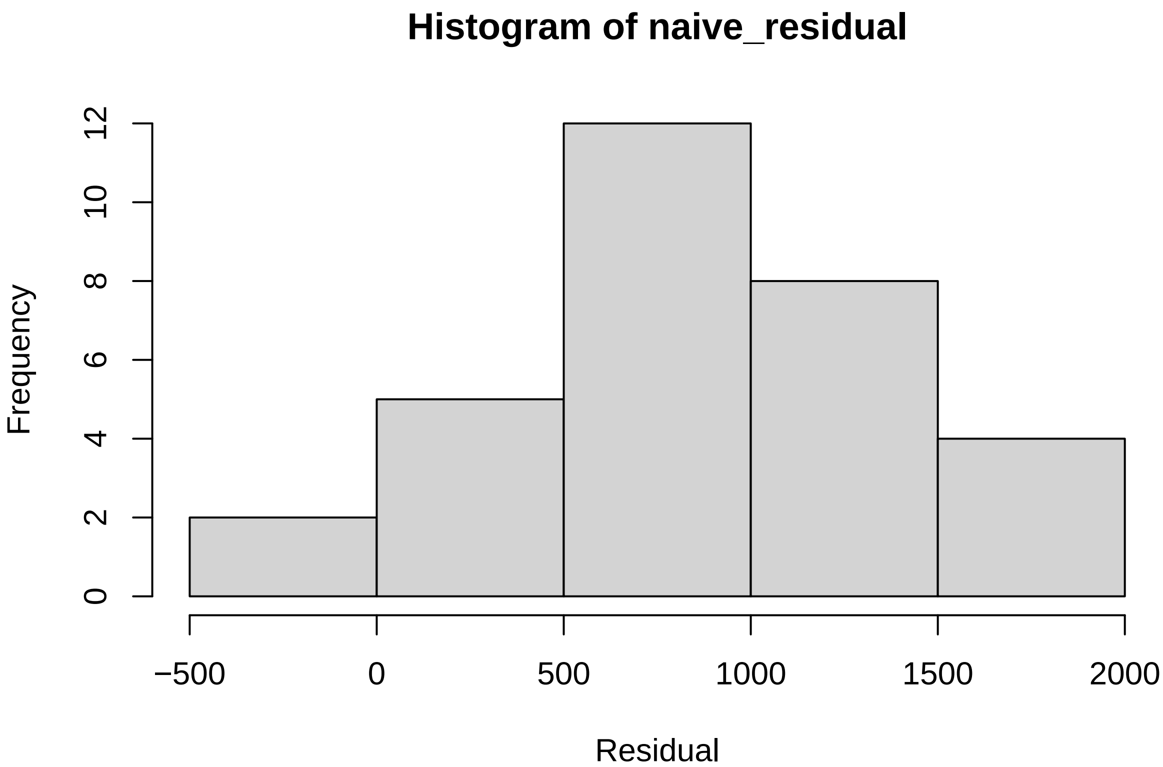

```r
accuracy(naive_fc)
```

```
##                    ME     RMSE      MAE       MPE      MAPE       MASE
## Training set 893.5484 1026.268 906.4516 0.2997294 0.3039222 0.07923528
##                   ACF1
## Training set 0.5545974
```

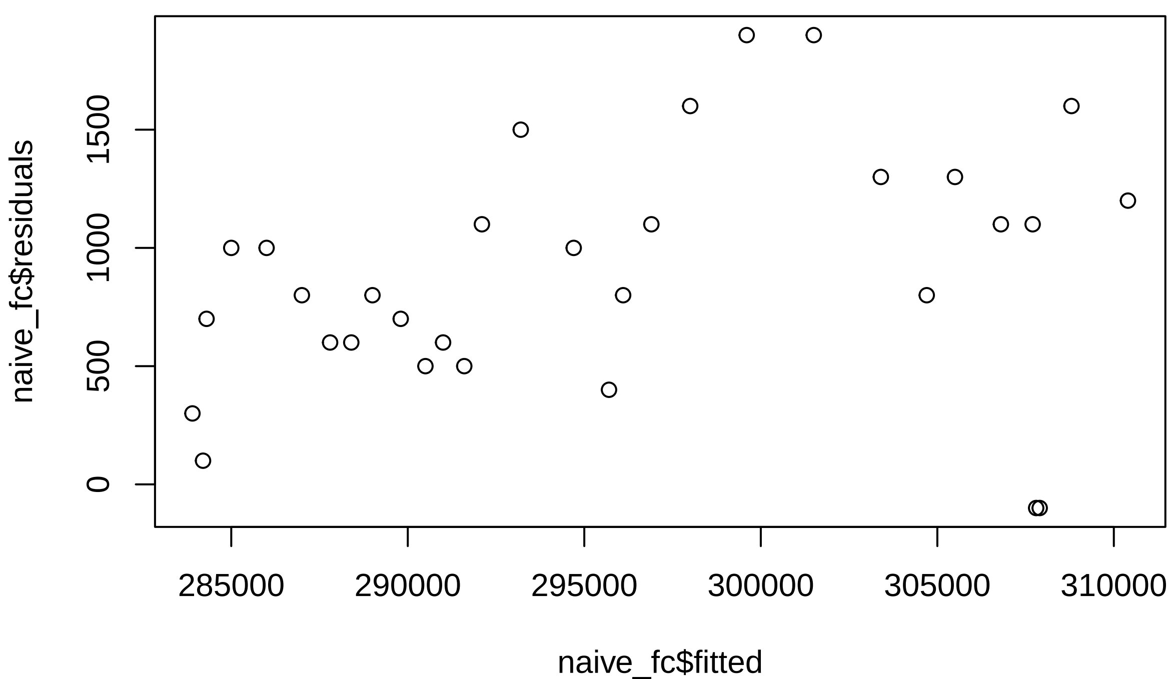
```r
naive_residual <- naive_fc$residuals
naive_fitted <- naive_fc$fitted
plot(naive_residual, ylab="Residual")
```



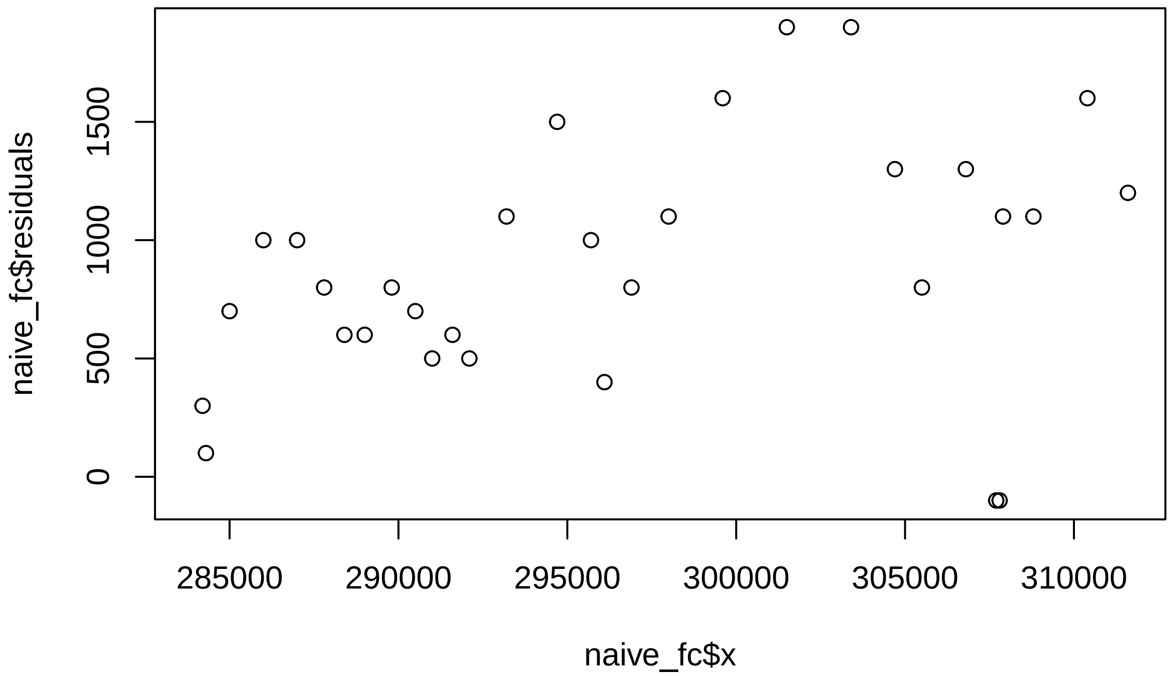
```r
hist(naive_residual, xlab="Residual")
```



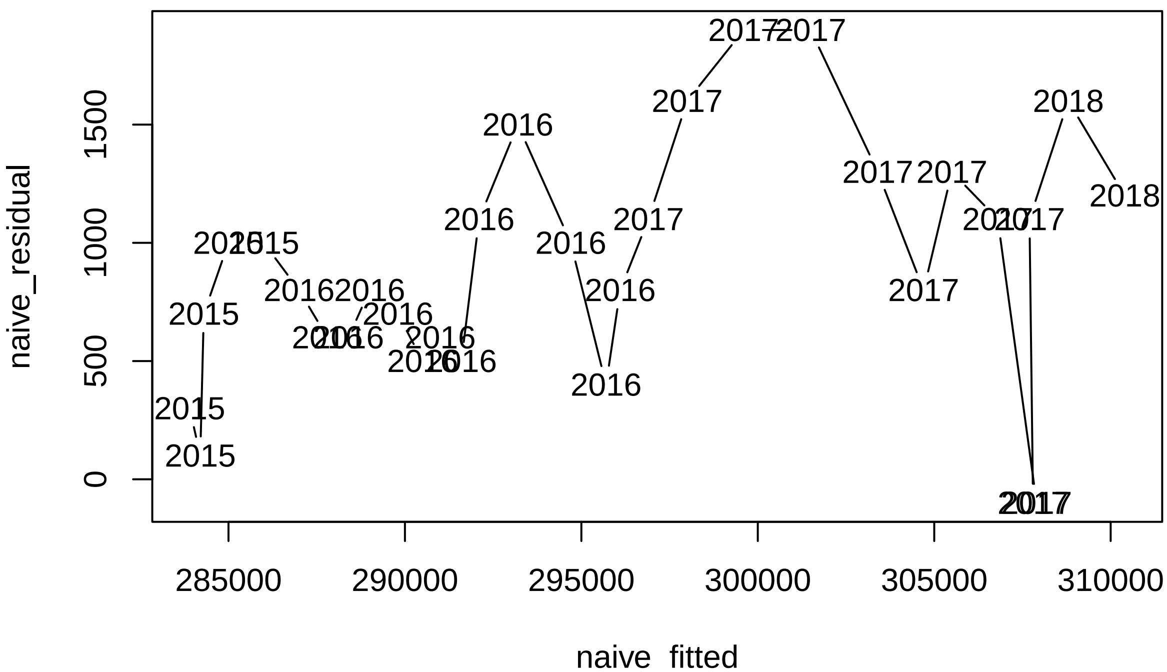

```r
plot.ts(naive_fc$fitted,naive_fc$residuals,xy.labels = FALSE,xy.lines = FALSE)
```



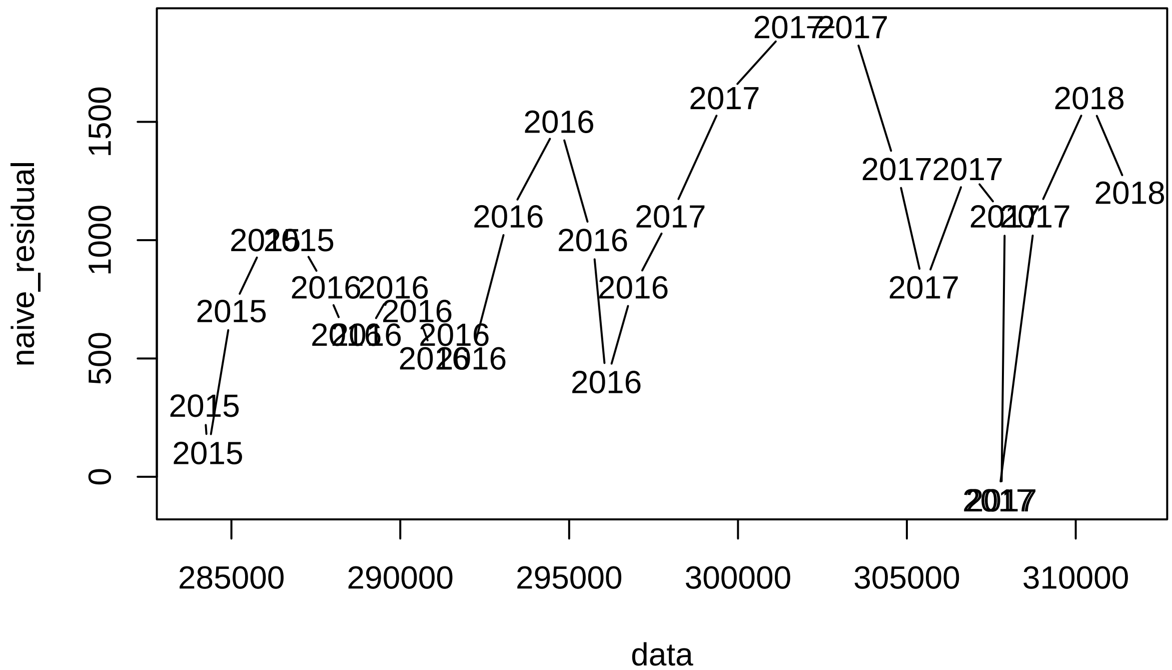

```r
plot.ts(naive_fc$x,naive_fc$residuals,xy.labels = FALSE,xy.lines = FALSE)
```



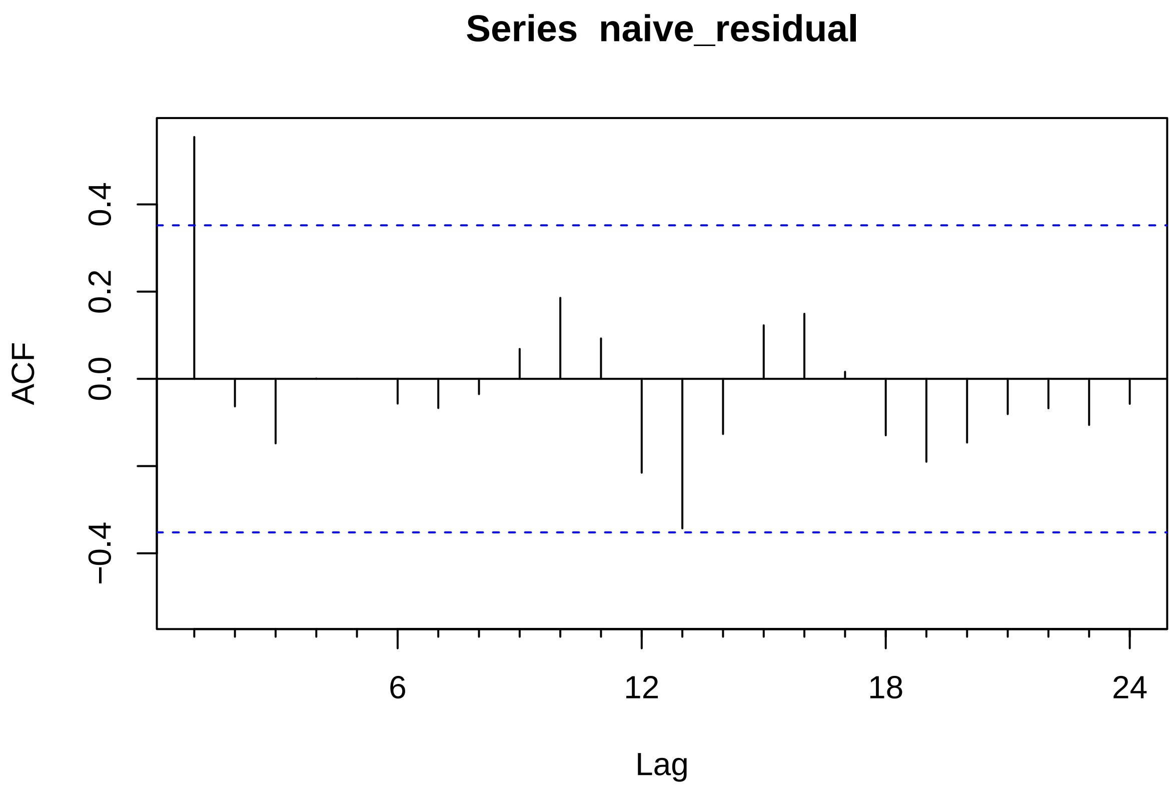

```r
plot(naive_fitted, naive_residual)
```




```r
plot(data, naive_residual)
```




```r
Acf(naive_residual)
```




```r
accuracy(naive_fc)
```

```
##                    ME     RMSE      MAE       MPE      MAPE       MASE
## Training set 893.5484 1026.268 906.4516 0.2997294 0.3039222 0.07923528
##                   ACF1
## Training set 0.5545974
```

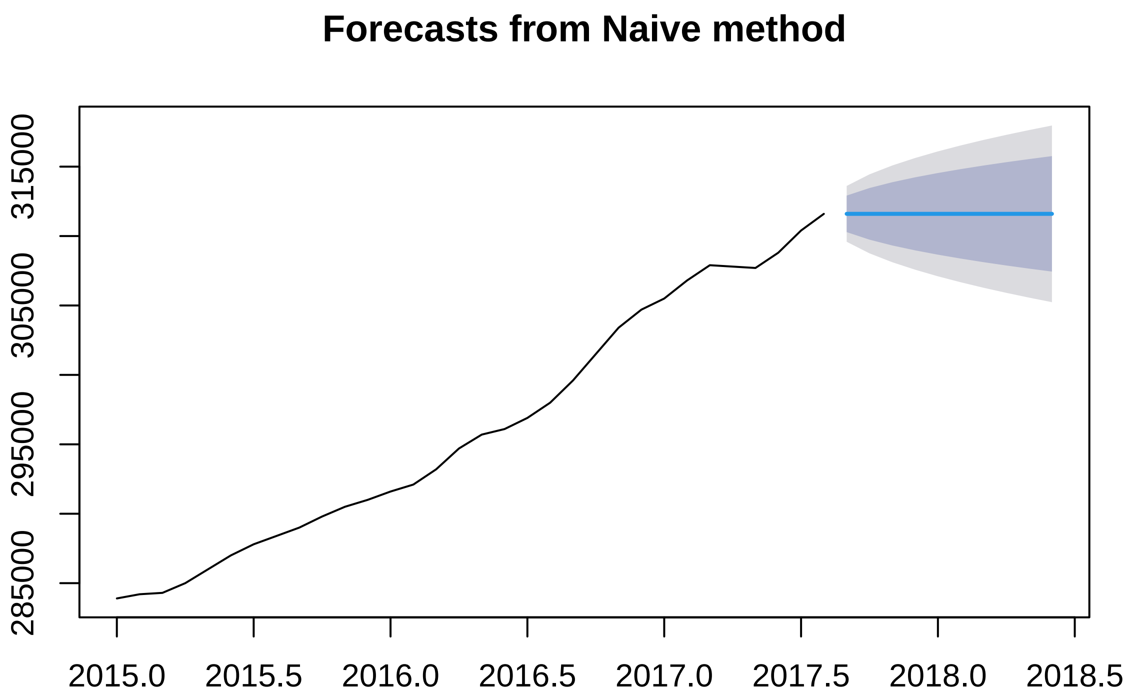
```r
naive_forecast <- forecast(naive_fc,20)
naive_forecast
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Sep 2017         311600 310284.8 312915.2 309588.6 313611.4
## Oct 2017         311600 309740.0 313460.0 308755.4 314444.6
## Nov 2017         311600 309322.0 313878.0 308116.1 315083.9
## Dec 2017         311600 308969.6 314230.4 307577.1 315622.9
## Jan 2018         311600 308659.1 314540.9 307102.3 316097.7
## Feb 2018         311600 308378.4 314821.6 306673.0 316527.0
## Mar 2018         311600 308120.3 315079.7 306278.2 316921.8
## Apr 2018         311600 307880.0 315320.0 305910.8 317289.2
## May 2018         311600 307654.4 315545.6 305565.7 317634.3
## Jun 2018         311600 307440.9 315759.1 305239.2 317960.8
```

```r
accuracy(naive_forecast)
```

```
##                    ME     RMSE      MAE       MPE      MAPE       MASE
## Training set 893.5484 1026.268 906.4516 0.2997294 0.3039222 0.07923528
##                   ACF1
## Training set 0.5545974
```

```r
plot(naive_forecast)
```



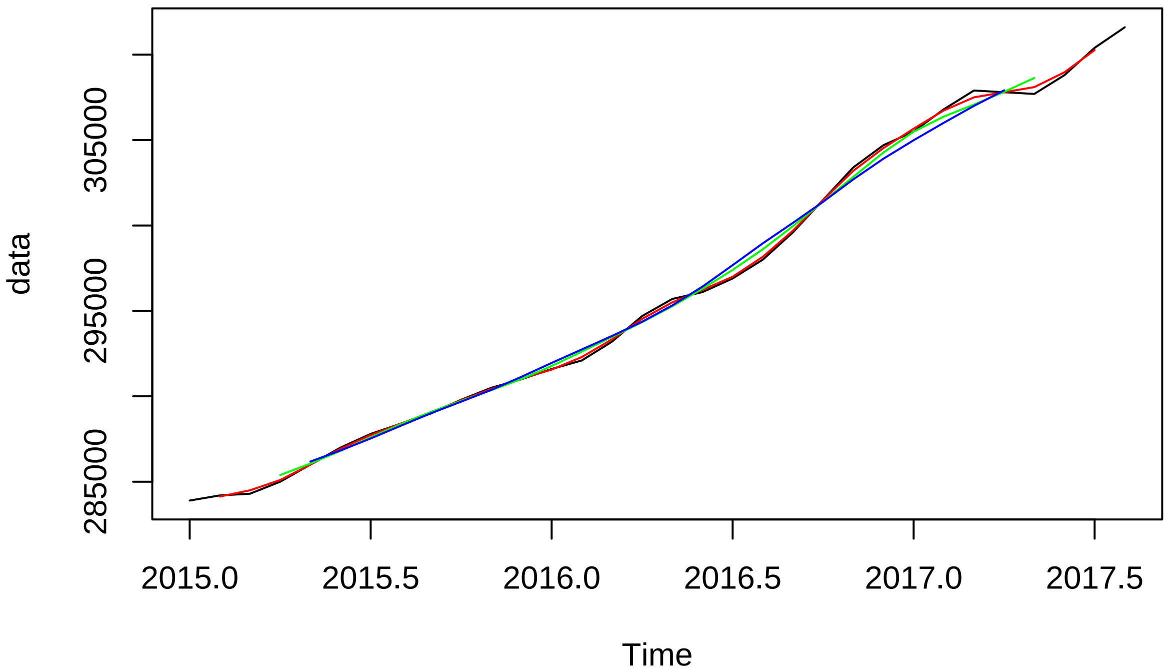
The naive method was applied, and the residual analysis showed:

Residual Plot: The residuals did not have a mean of zero, indicating a potential bias in prediction.

ACF Plot: Shows significant autocorrelation, suggesting some patterns were not captured by the naive forecast.

Simple Moving Averages

```
#Moving Avg
ma_forecast_1 = ma(data,order=3)
ma_forecast_2 = ma(data,order=6)
ma_forecast_3 = ma(data,order=9)

plot(data)
lines(ma_forecast_1, col="red")
lines(ma_forecast_2, col="green")
lines(ma_forecast_3, col="blue")
```

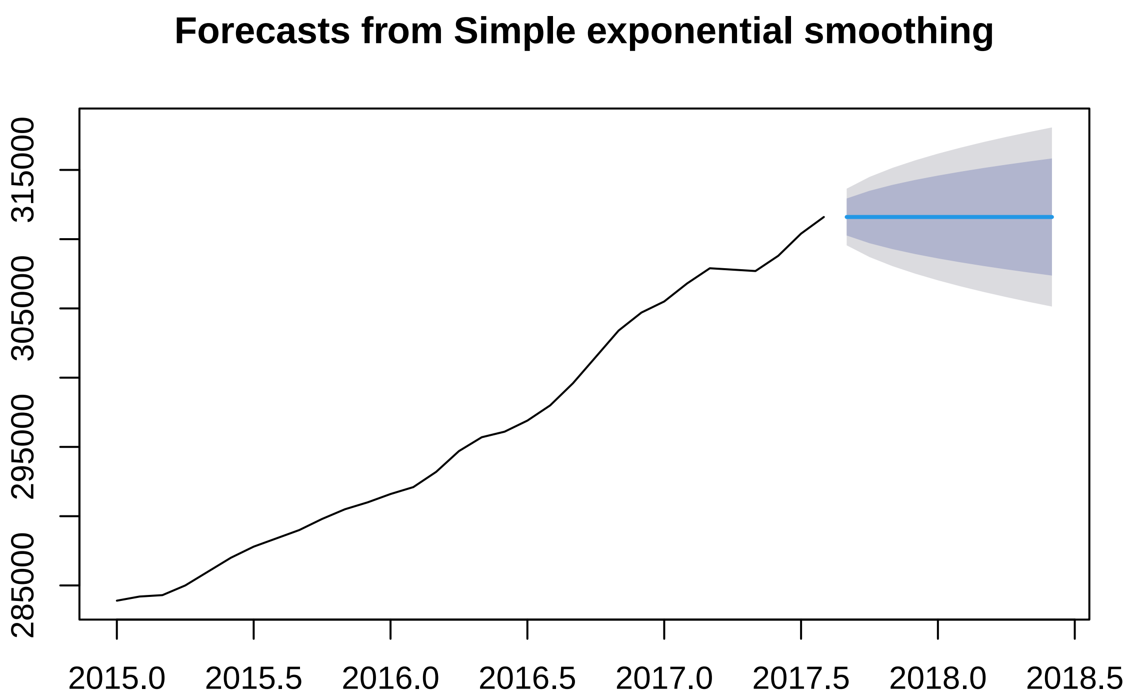


Three different moving average orders were applied: 3, 6, and 9. It was observed that: • Order 3 provided the closest approximation to the actual data.

- Higher orders smoothed the curve more, resulting in less accuracy in terms of short-term predictions.

Simple Smoothing (SES)

```
#simple smoothing
ses_fc <- ses(data)
plot(ses_fc)
```




```r
summary(ses_fc)
```

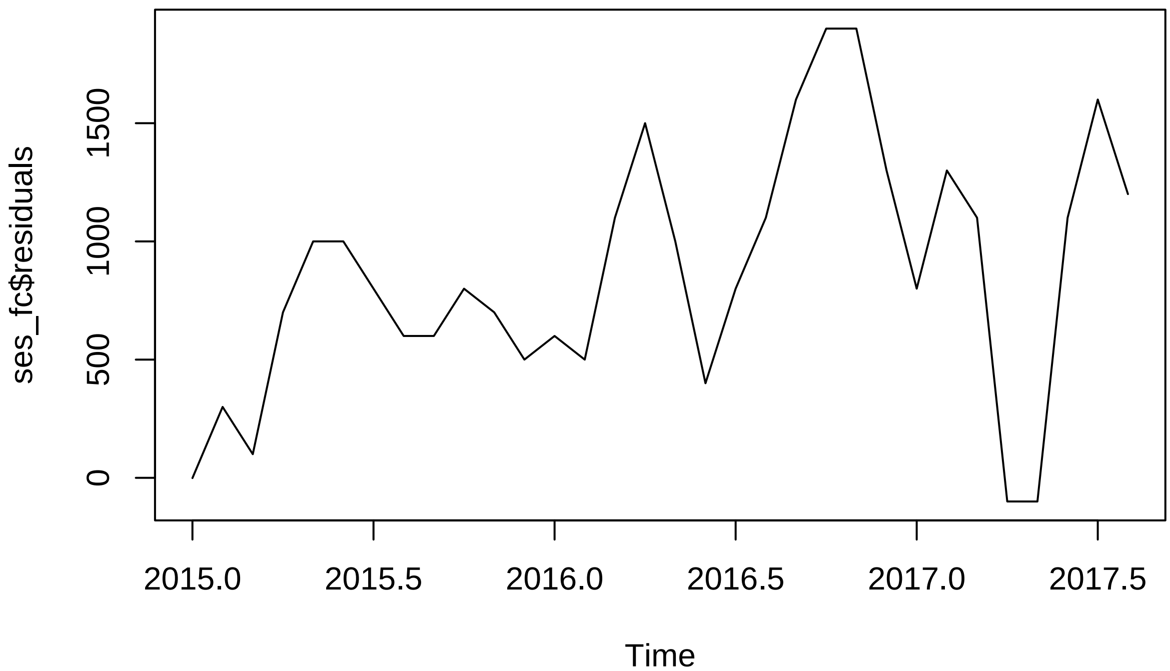
```
##
## Forecast method: Simple exponential smoothing
##
## Model Information:
## Simple exponential smoothing
##
## Call:
##  ses(y = data)
##
##   Smoothing parameters:
##     alpha = 0.9999
##
##   Initial states:
##     l = 283901.0096
##
##   sigma:  1043.324
##
##      AIC     AICc      BIC
## 559.6490 560.5061 564.0462
##
## Error measures:
##                    ME     RMSE      MAE       MPE      MAPE       MASE
## Training set 865.6766 1010.194 878.2334 0.2903796 0.2944615 0.07676865
##                   ACF1
## Training set 0.5651999
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Sep 2017       311599.9 310262.8 312937.0 309555.0 313644.8
## Oct 2017       311599.9 309709.1 313490.7 308708.1 314491.6
```

```
## Nov 2017       311599.9 309284.2 313915.6 308058.3 315141.5
## Dec 2017       311599.9 308925.9 314273.8 307510.4 315689.3
## Jan 2018       311599.9 308610.3 314589.4 307027.8 316172.0
## Feb 2018       311599.9 308325.0 314874.8 306591.4 316608.4
## Mar 2018       311599.9 308062.6 315137.1 306190.1 317009.7
## Apr 2018       311599.9 307818.4 315381.4 305816.6 317383.2
## May 2018       311599.9 307589.0 315610.7 305465.8 317734.0
## Jun 2018       311599.9 307372.1 315827.7 305134.0 318065.8
```

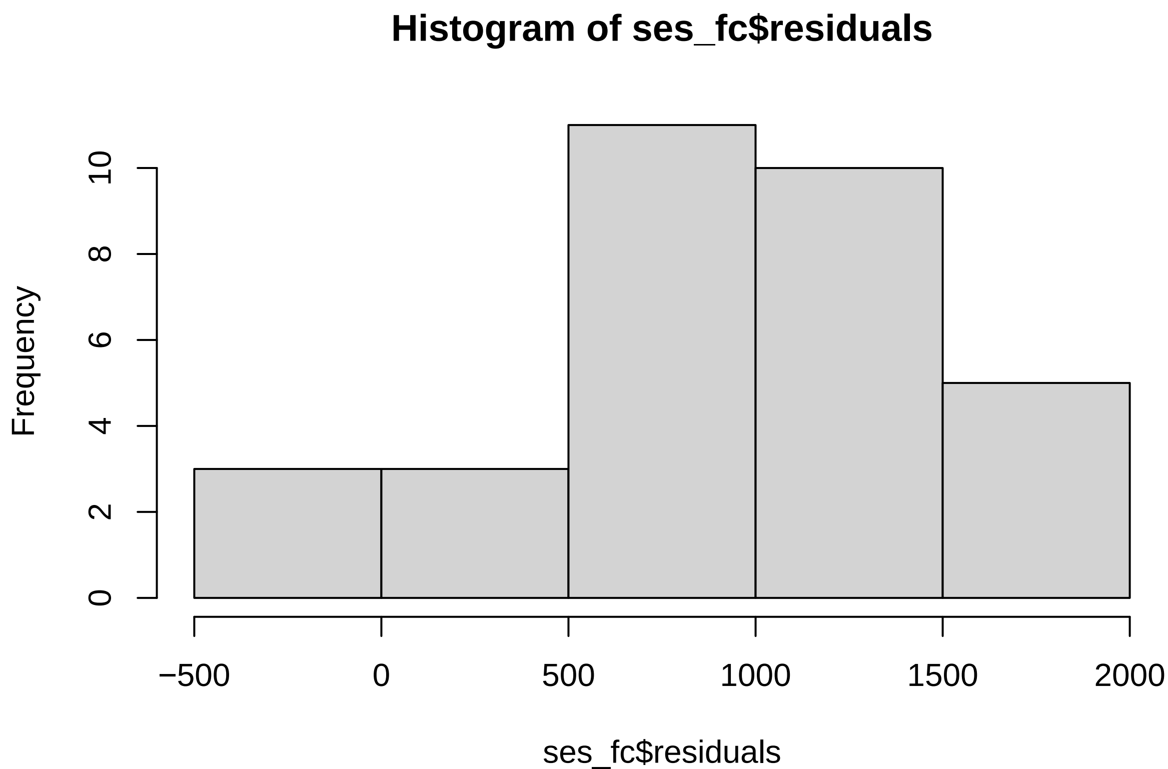
```r
attributes(ses_fc)
```

```
## $names
##  [1] "model"     "mean"      "level"     "x"         "upper"     "lower"
##  [7] "fitted"    "method"    "series"    "residuals"
##
## $class
## [1] "forecast"
```

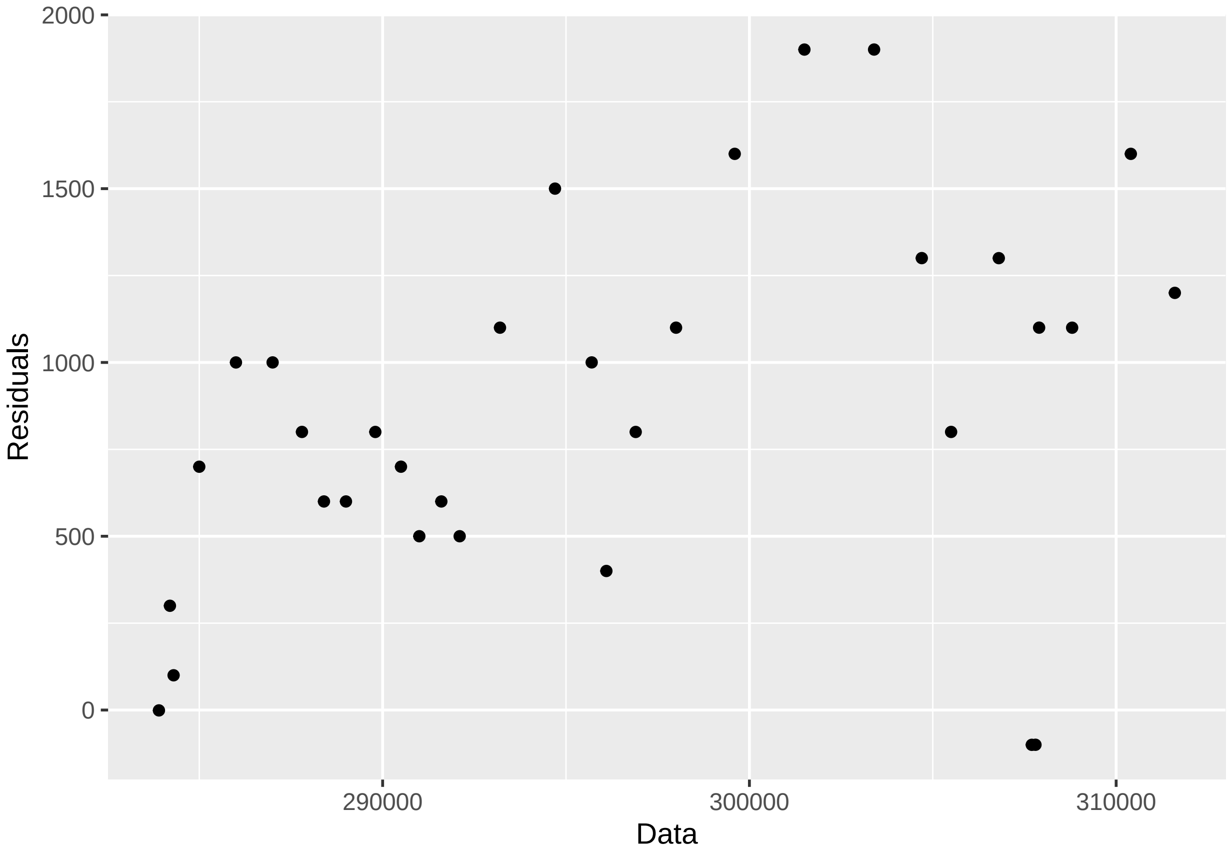
```r
plot(ses_fc$residuals)
```



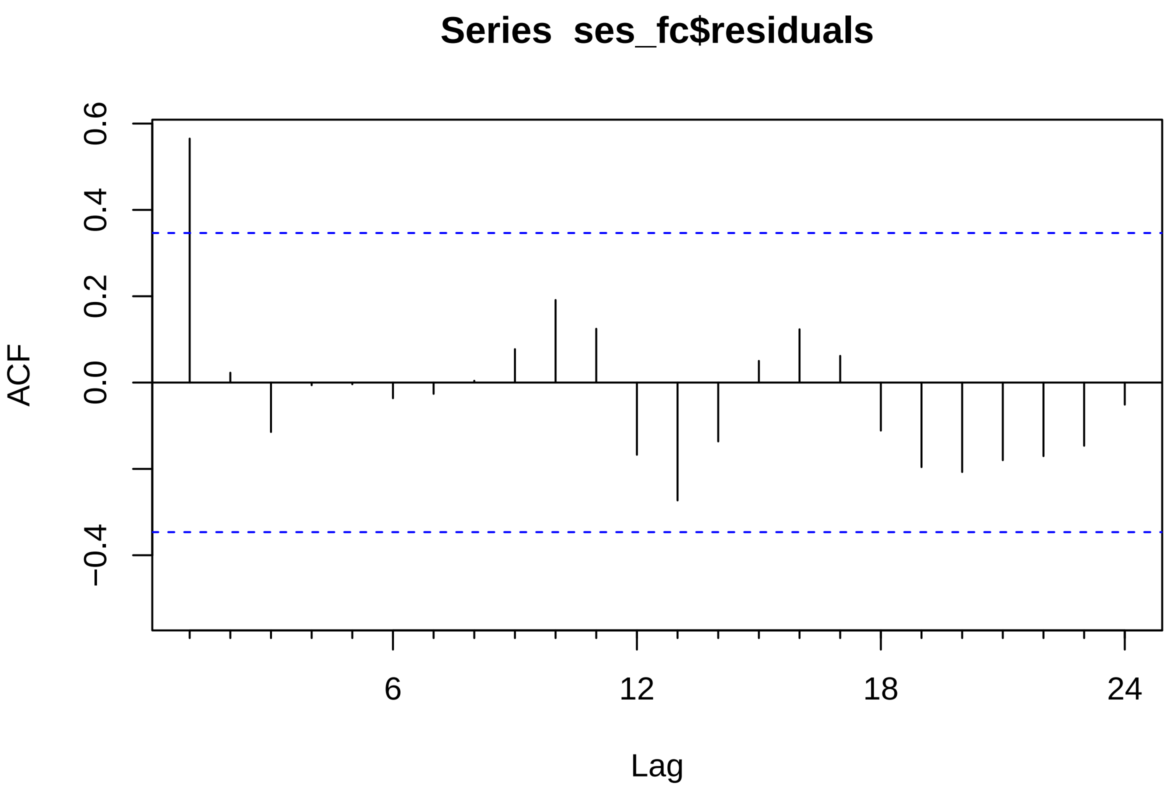

```r
hist(ses_fc$residuals)
```




```
cbind(Data = data,
      Residuals=residuals(ses_fc)) %>%
  as.data.frame() %>%
  ggplot(aes(x=Data, y=Residuals)) + geom_point()
```




```r
Acf(ses_fc$residuals)
```



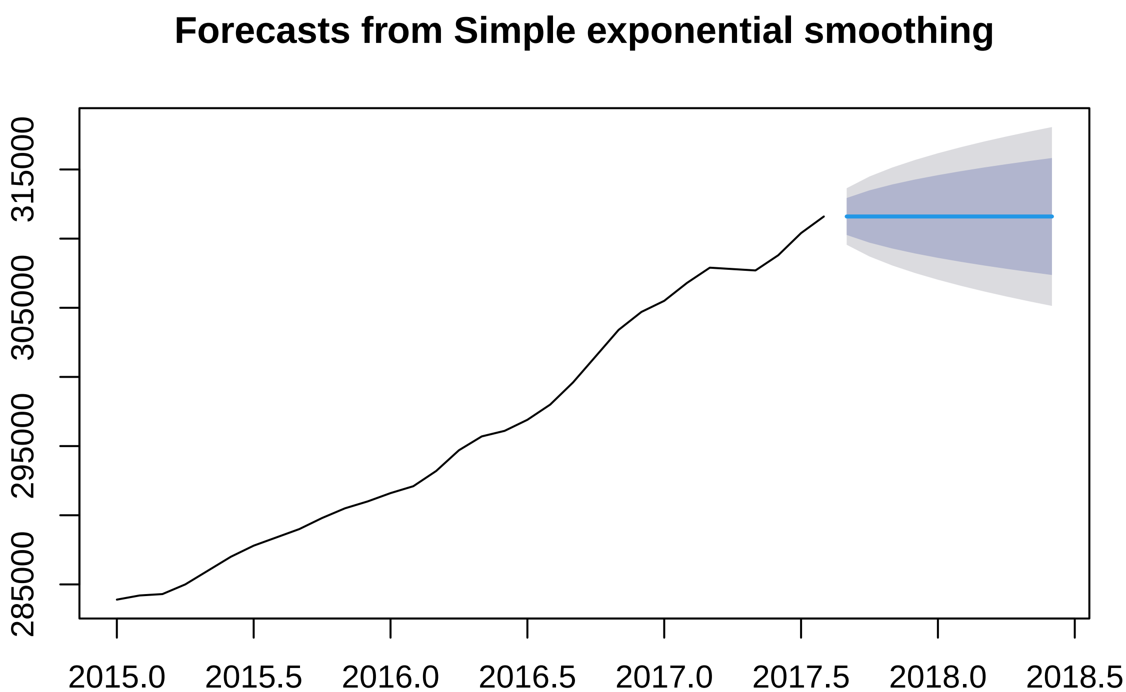
```r
accuracy(ses_fc)
```

```
##                    ME     RMSE      MAE       MPE      MAPE       MASE
## Training set 865.6766 1010.194 878.2334 0.2903796 0.2944615 0.07676865
##                   ACF1
## Training set 0.5651999
```

```r
forecast(ses_fc)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Sep 2017       311599.9 310262.8 312937.0 309555.0 313644.8
## Oct 2017       311599.9 309709.1 313490.7 308708.1 314491.6
## Nov 2017       311599.9 309284.2 313915.6 308058.3 315141.5
## Dec 2017       311599.9 308925.9 314273.8 307510.4 315689.3
## Jan 2018       311599.9 308610.3 314589.4 307027.8 316172.0
## Feb 2018       311599.9 308325.0 314874.8 306591.4 316608.4
## Mar 2018       311599.9 308062.6 315137.1 306190.1 317009.7
## Apr 2018       311599.9 307818.4 315381.4 305816.6 317383.2
## May 2018       311599.9 307589.0 315610.7 305465.8 317734.0
## Jun 2018       311599.9 307372.1 315827.7 305134.0 318065.8
```

```r
plot(forecast(ses_fc))
```



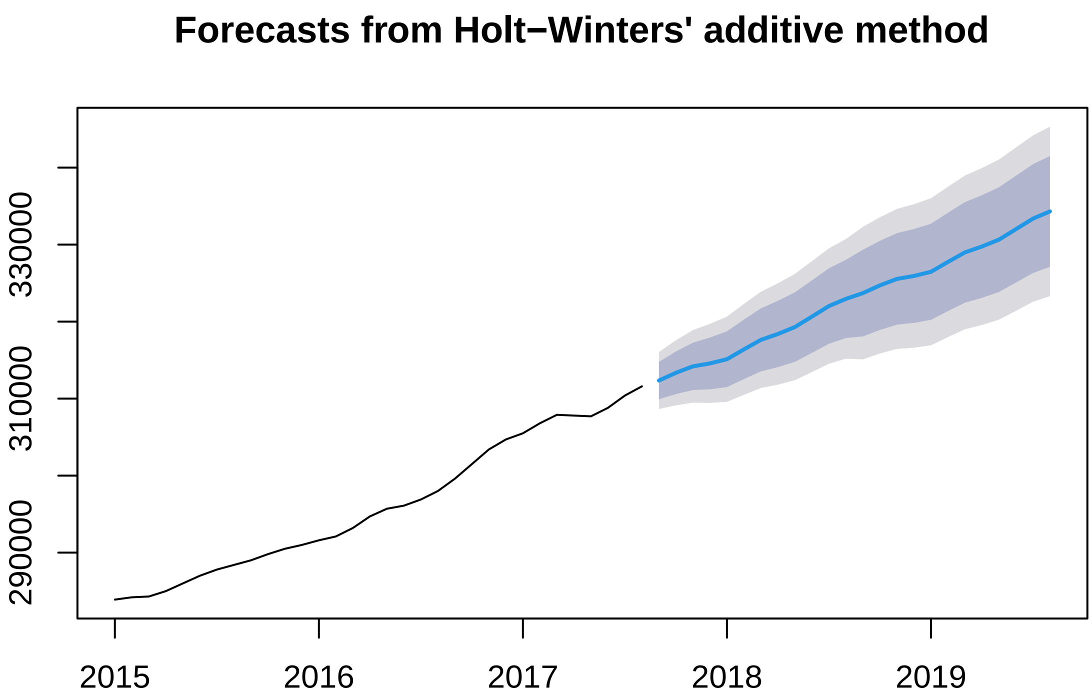
Observations :

- Alpha = 0.9999 Alpha specifies the coefficient for the level smoothing. Values near 1.0 mean that the latest value has more weight.
- Initial state: l = 283901.0096
- Sigma: 1043.324. Sigma defines the variance in the forecast predicted.
- The residuals appear to have increasing positive values and then peaked in the third quarter of the year 2016 and then dipped down.

- Most of the residual values appear to be positive and do not have a mean of zero.
- The histogram appears to be normally distributed. But the values do not have a mean zero. That is data is biased.
- The Fitted vs Residuals plot appears to have a trend. The plot slightly shows a straight diagonal line pattern.
- This means there is heteroscedasticity in the errors which means that the variance of the residuals may not be constant.
- Similar to the previous plot, the Actual vs. Residuals plot appears to have some trend in the data.
- Values of the Acf have crossed the confidence level meaning there is a trend in the residuals and we have missed some variable in our forecast. The Acf values also show seasonality in the plot and we missed this variable too.

Simple Smoothing - Holt-Winters Method

```
HW_forecast <- hw(data, seasonal = "additive")
plot(forecast(HW_forecast))
```




```
attributes(HW_forecast)
```

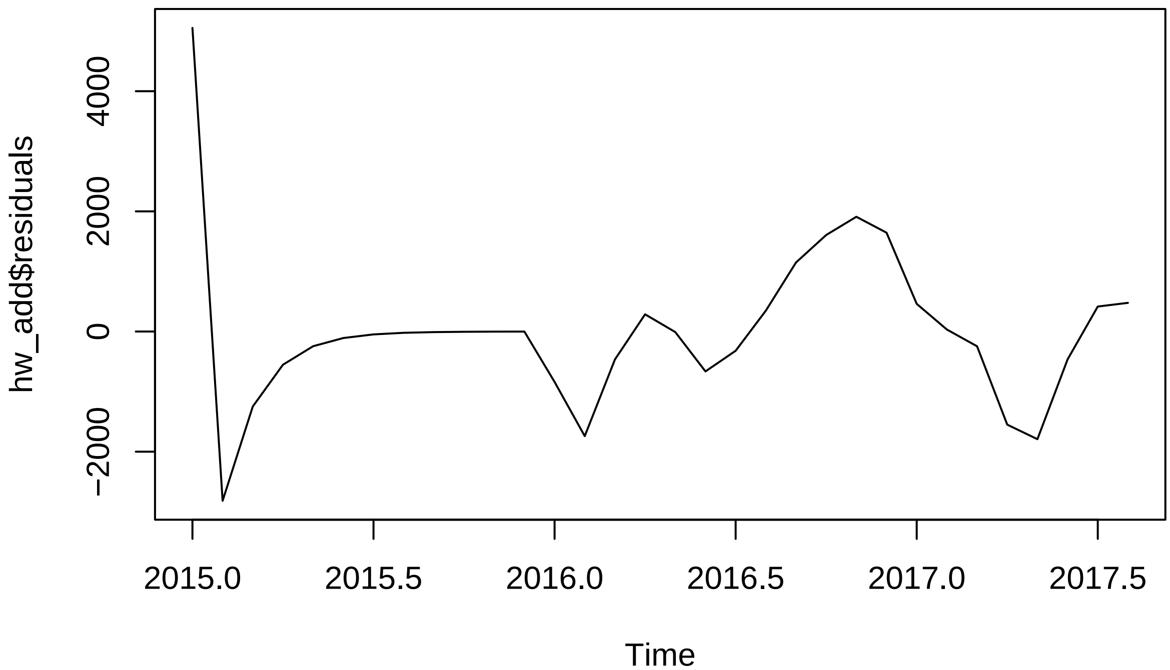
```
## $names
##  [1] "model"     "mean"      "level"     "x"         "upper"     "lower"
##  [7] "fitted"    "method"    "series"    "residuals"
##
## $class
## [1] "forecast"
```

```
hw_add <- forecast(HW_forecast)
hw_add$model
```

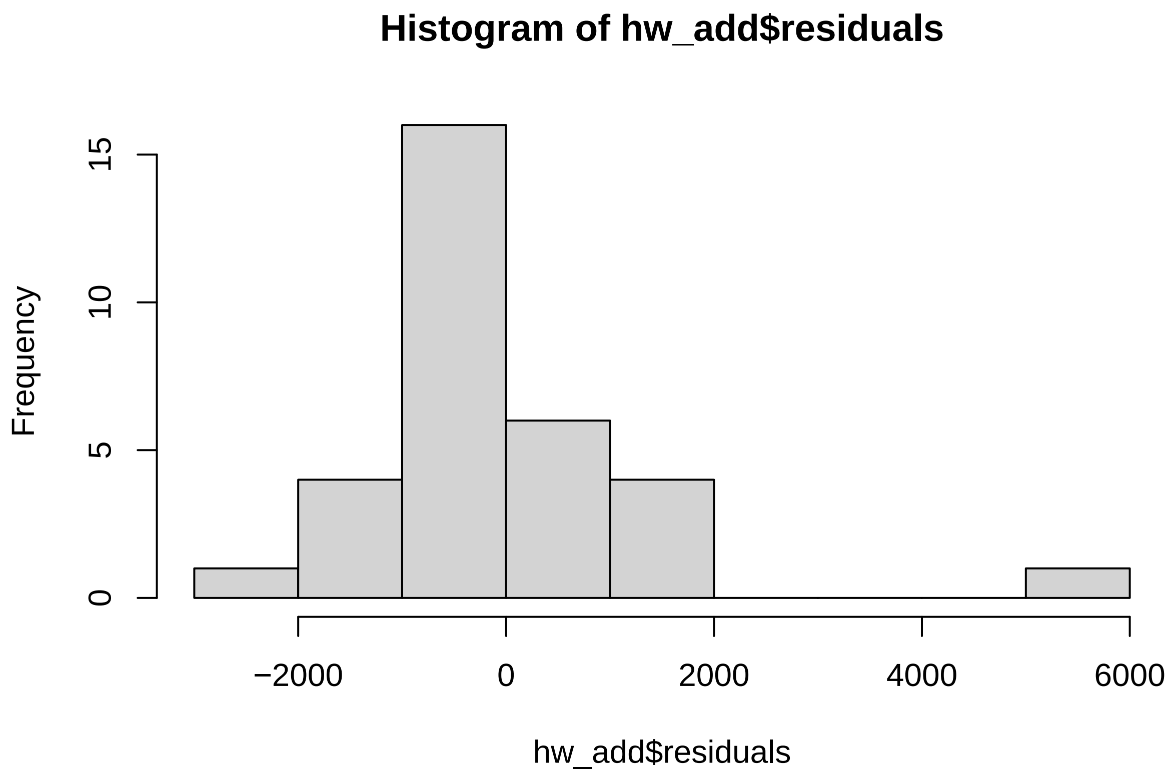
```
## Holt-Winters' additive method
##
```

```
## Call:
##  hw(y = data, seasonal = "additive")
##
##   Smoothing parameters:
##     alpha = 0.5573
##     beta  = 1e-04
##     gamma = 0.4427
##
##   Initial states:
##     l = 280670.1715
##     b = 946.3705
##     s = -1024.945 -578.7329 -332.3787 -186.2467 160.0107 506.2832
##            652.522 598.9254 545.3346 791.4534 1637.615 -2769.84
##
##   sigma:  1885.118
##
##      AIC     AICc      BIC
## 605.3945 649.1088 630.3121
```

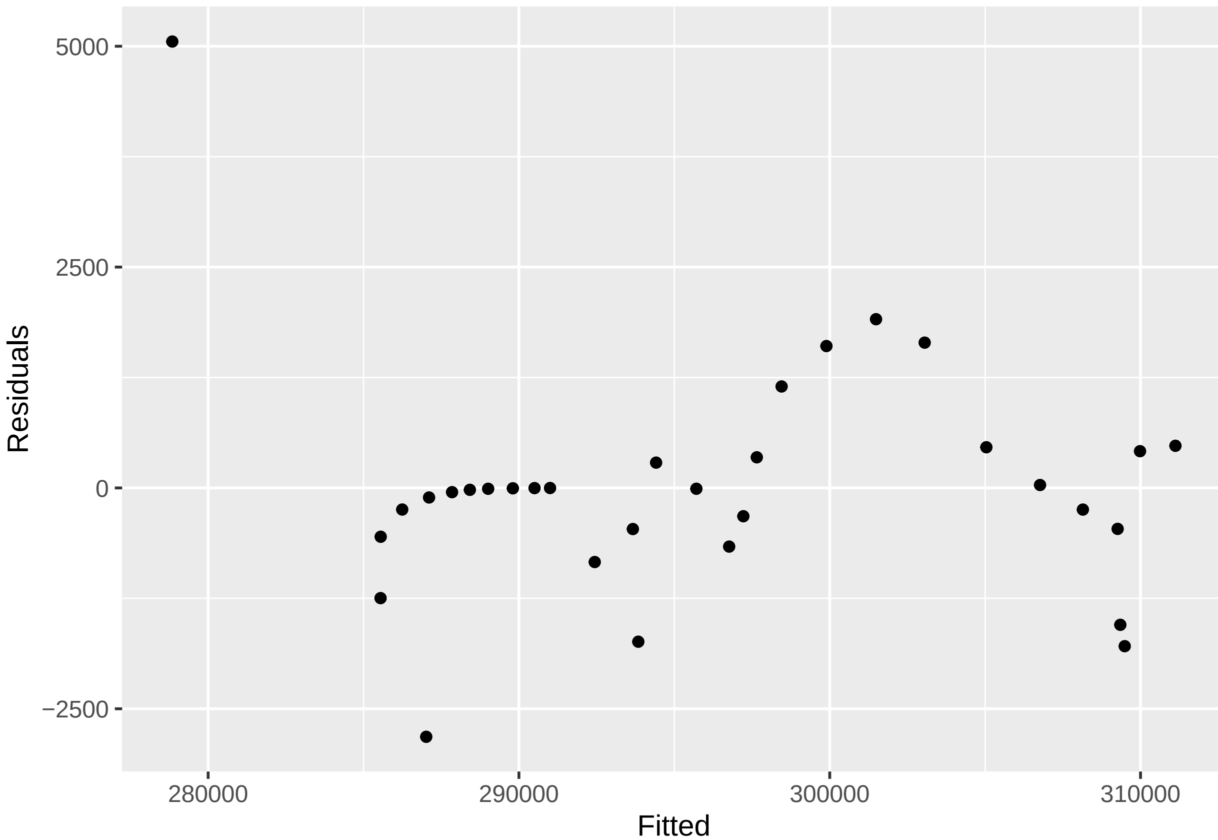
```r
plot(hw_add$residuals)
```



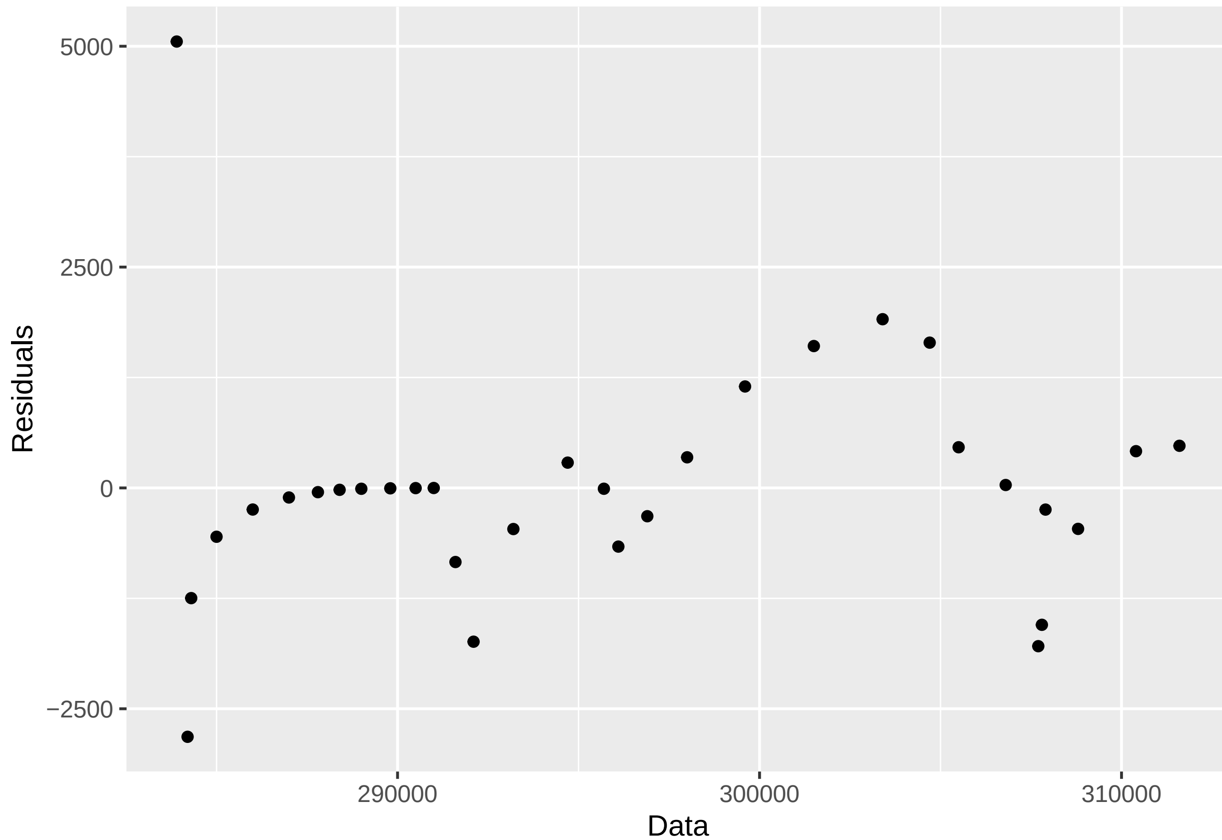

```r
hist(hw_add$residuals)
```



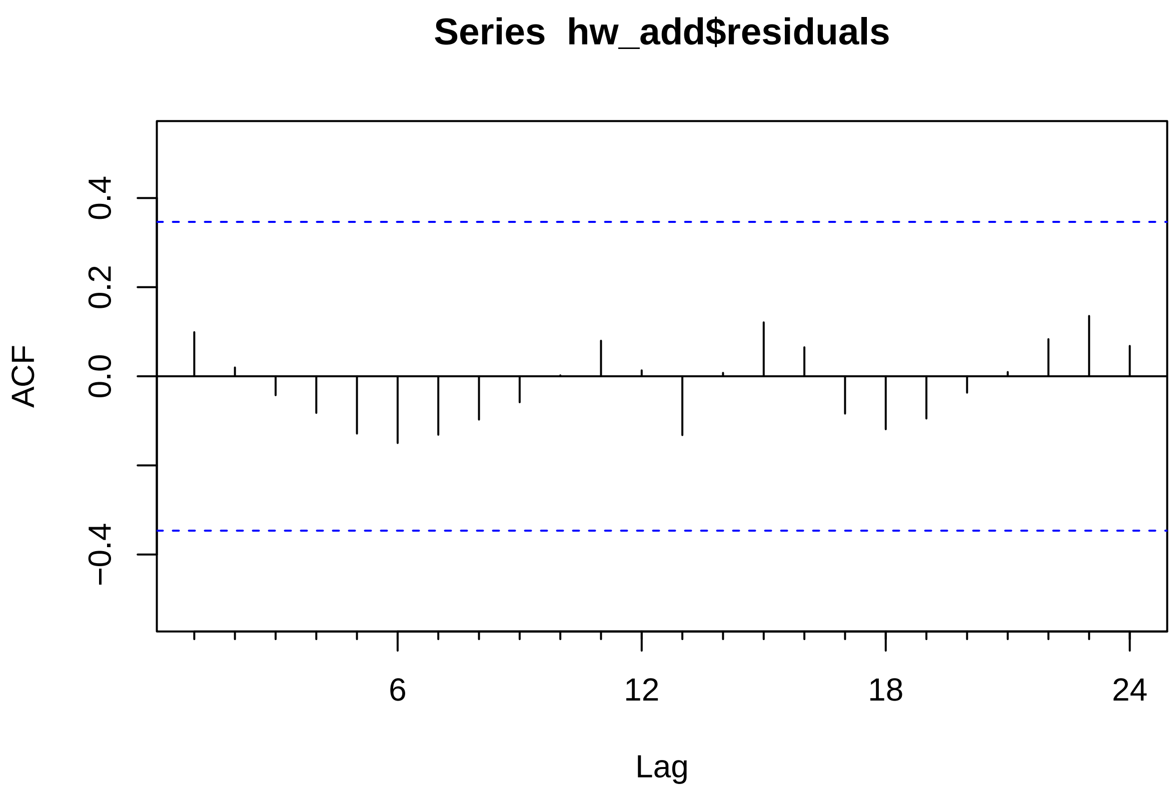

```
cbind(Fitted = fitted(hw_add),
      Residuals=residuals(hw_add)) %>%
  as.data.frame() %>%
  ggplot(aes(x=Fitted, y=Residuals)) + geom_point()
```




```r
cbind(Data = data,
      Residuals=residuals(hw_add)) %>%
  as.data.frame() %>%
  ggplot(aes(x=Data, y=Residuals)) + geom_point()
```




```
Acf(hw_add$residuals)
```



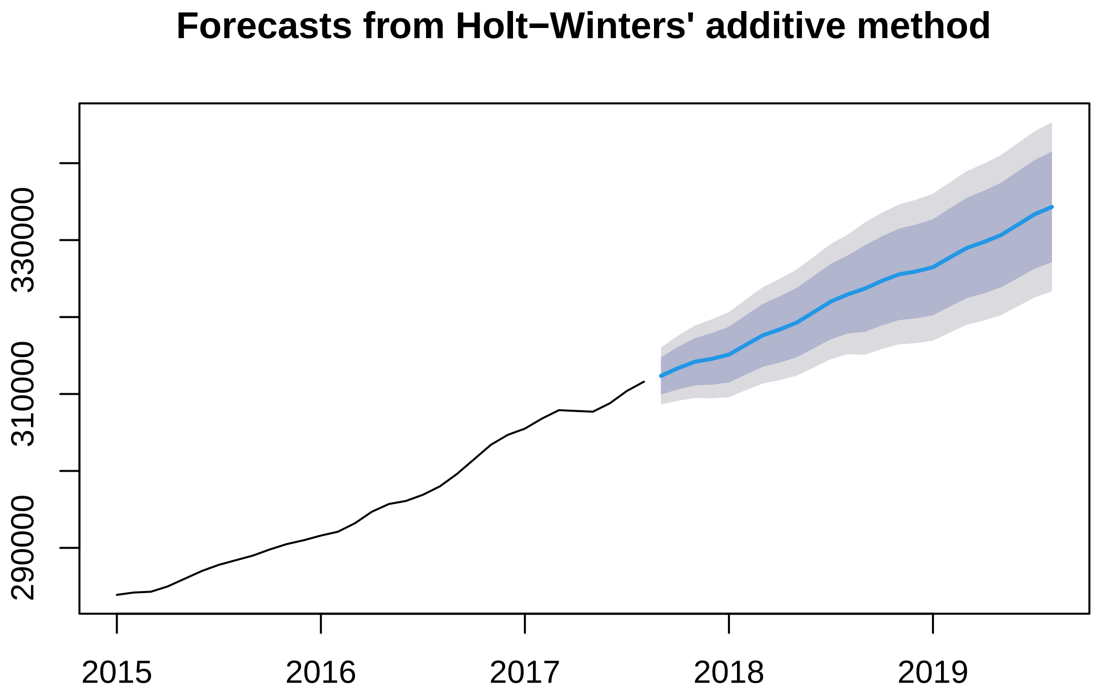
```r
accuracy(hw_add)
```

```
##                   ME     RMSE      MAE        MPE      MAPE       MASE
## Training set 7.34857 1332.979 829.0124 0.00176648 0.2811894 0.07246612
##                    ACF1
## Training set 0.09897217
```

```r
forecast(HW_forecast)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Sep 2017       312349.2 309933.3 314765.1 308654.4 316044.0
## Oct 2017       313354.3 310588.5 316120.2 309124.3 317584.3
## Nov 2017       314190.0 311113.6 317266.3 309485.1 318894.9
## Dec 2017       314573.2 311214.8 317931.6 309437.0 319709.4
## Jan 2018       315116.4 311497.8 318735.0 309582.2 320650.6
## Feb 2018       316397.6 312536.3 320259.0 310492.2 322303.1
## Mar 2018       317634.1 313544.2 321724.0 311379.2 323889.0
## Apr 2018       318397.5 314091.1 322703.8 311811.5 324983.4
## May 2018       319295.8 314783.4 323808.3 312394.6 326197.0
## Jun 2018       320654.2 315944.5 325363.9 313451.4 327857.0
## Jul 2018       322022.5 317123.5 326921.5 314530.1 329514.9
## Aug 2018       322956.7 317875.4 328038.1 315185.5 330728.0
## Sep 2018       323705.9 318078.2 329333.6 315099.1 332312.7
## Oct 2018       324711.1 318923.8 330498.3 315860.2 333561.9
## Nov 2018       325546.7 319604.1 331489.2 316458.3 334635.0
## Dec 2018       325929.9 319835.9 332023.9 316610.0 335249.8
## Jan 2019       326473.1 320231.4 332714.9 316927.2 336019.1
## Feb 2019       327754.4 321368.2 334140.6 317987.5 337521.2
## Mar 2019       328990.8 322463.3 335518.3 319007.9 338973.8
## Apr 2019       329754.2 323088.4 336420.0 319559.7 339948.7
## May 2019       330652.6 323851.1 337454.0 320250.7 341054.4
## Jun 2019       332010.9 325076.5 338945.3 321405.7 342616.2
## Jul 2019       333379.2 326314.3 340444.1 322574.4 344184.0
## Aug 2019       334313.5 327120.4 341506.6 323312.6 345314.3
```

```r
plot(forecast(HW_forecast))
```



####Holt-Winters' additive model was used with the following parameters:

- Alpha: 0.5573
- Beta: 1e-04
- Gamma: 0.4427

The model provided the best fit among the methods, as indicated by the lower error values.

Conclusion:

The analysis of New Jersey median home prices from April 1996 to August 2017 reveals significant trends and seasonal patterns. Our objective was to identify key insights into the historical data and utilize various forecasting techniques to predict future movements in home prices. The study included decomposition analysis, central tendency measures, and application of multiple forecasting methods—Naive, Simple Moving Averages, Simple Exponential Smoothing (SES), and Holt-Winters' models—to evaluate their efficacy and accuracy.

Key Findings:

1. **Trend and Seasonality**:
    - The time series analysis indicates a steady upward trend in home prices since 2015, suggesting a robust growth in the New Jersey housing market post the 2008 economic crisis.
    - The decomposition analysis shows a strong seasonal component, which recurs consistently over the years. This seasonality highlights predictable variations, with home prices typically peaking in the winter months (November–December) and declining during summer (June–July). Such trends could be attributed to seasonal buying patterns, where the demand for housing generally increases towards the end of the year.
2. **Central Tendency and Spread**:
    - The data's central tendency metrics, including mean, median, and quartiles, further confirm a steady growth in prices. The IQR (Interquartile Range) analysis showed minimal dispersion, indicating consistent market behavior without extreme fluctuations or outliers.
3. **Model Performance and Forecasting**:

- The Naive method, which uses the previous period's value as a forecast, was quick to implement but lacked precision, as indicated by the high Mean Absolute Error (MAE) and autocorrelation in residuals.
- The Simple Moving Averages (orders 3, 6, and 9) provided a smoother approximation of trends but tended to overlook short-term fluctuations, making them more suitable for capturing long-term movements.
- The Simple Exponential Smoothing (SES) method resulted in better performance than the Naive method by adjusting for recent changes. However, its residuals analysis showed some bias, indicating an incomplete capture of seasonality.
- The Holt-Winters method emerged as the most reliable forecasting technique. By incorporating both trend and seasonality components, it produced the most accurate forecasts with the lowest error values. The additive Holt-Winters model fit well with the data structure, affirming that seasonal variations remain constant in size over time while the trend component changes.

4. **Future Forecast and Market Implications**:
   - All three forecasting models—Naive, SES, and Holt-Winters—predict continued growth in home prices over the next year. However, the Holt-Winters forecast is particularly optimistic, projecting a steady increase in prices for the next 12 months, with the possibility of surpassing $330,000 by mid-2019.
   - The economic and market conditions influencing these price movements should be further investigated. Factors such as interest rates, housing supply, and demographic shifts could either accelerate or dampen the predicted growth.
5. **Model Comparison and Best Choice**:
   - **Holt-Winters** was the superior model for forecasting home prices due to its comprehensive approach to handling trend and seasonality. Its lower error rates (e.g., MAE and RMSE) and better residual distribution suggest that it captures the underlying patterns more effectively than other models.

Final Recommendations: Given the analysis and the strength of the Holt-Winters model, it is recommended to use this model for future forecasting of New Jersey median home prices. The model's consideration of both seasonal and trend factors allows for a more nuanced and accurate forecast, making it particularly suitable for real estate market analyses. Future updates to the model should include new data points and incorporate external economic factors to enhance prediction accuracy.

In conclusion, the New Jersey housing market is projected to continue its upward trajectory in the near future, with median home prices expected to rise steadily. This trend aligns with broader economic conditions, suggesting sustained demand for housing in the region. Such insights are invaluable for real estate investors, policy makers, and prospective home buyers aiming to make informed decisions based on data-driven projections.