Global Plastic Production Forecasting and Time Series Analysis

Overview

In this project, we'll analyze the time series and build a prediction model to predict the Global Plastic Production for the next 10 years.

we'll be doing the following steps to implement our project:

- We'll collect the data
- we'll explore the data by checking for missing values, outliers, irrelevant data points and irrelevant columns
- We'll scale the data so that all the values are in the range of 1-100
- Then we'll convert the data into a time series and split the data into train and test set
- We'll build our prediction model using the Holt's Exponential Smoothing method

Data

- The dataset Global Plastic Production has been taken from Our world in data
- The data available is from the year 1950 to 2015

Libraries required

```
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
```

```
library(tseries)
```

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

```
library(forecast)
library(scales)
library(ggplot2)
library(expsmooth)
```

Loading the data

```
data <- read.csv("C:/Users/user/Desktop/global-plastics-production.csv")
head(data)
```

```
##   Entity     Code Year Global.plastics.production..million.tonnes...tonnes.
## 1  World OWID_WRL 1950                                              2000000
## 2  World OWID_WRL 1951                                              2000000
## 3  World OWID_WRL 1952                                              2000000
## 4  World OWID_WRL 1953                                              3000000
## 5  World OWID_WRL 1954                                              3000000
## 6  World OWID_WRL 1955                                              4000000
```

about the data

```
str(data)
```

```
## 'data.frame':    66 obs. of  4 variables:
##  $ Entity                                              : chr  "World" "World" "World" "Wor
ld" ...
##  $ Code                                                : chr  "OWID_WRL" "OWID_WRL" "OWID_
WRL" "OWID_WRL" ...
##  $ Year                                                : int  1950 1951 1952 1953 1954 195
5 1956 1957 1958 1959 ...
##  $ Global.plastics.production..million.tonnes...tonnes.: int  2000000 2000000 2000000 3000
000 3000000 4000000 5000000 5000000 6000000 7000000 ...
```

```
summary(data)
```

```
##     Entity              Code                Year     
##  Length:66          Length:66          Min.   :1950  
##  Class :character   Class :character   1st Qu.:1966  
##  Mode  :character   Mode  :character   Median :1982  
##                                        Mean   :1982  
##                                        3rd Qu.:1999  
##                                        Max.   :2015  
##  Global.plastics.production..million.tonnes...tonnes.
##  Min.   :  2000000                                   
##  1st Qu.: 20750000                                   
##  Median : 76500000                                   
##  Mean   :118530303                                   
##  3rd Qu.:198500000                                   
##  Max.   :381000000
```

Checking for missing values

```
any(is.na(data))
```

```
## [1] FALSE
```

removing the non-relevant columns

```
df <- select(data, -c(Entity, Code))
str(df)
```

```
## 'data.frame':    66 obs. of  2 variables:
##  $ Year                                            : int  1950 1951 1952 1953 1954 195
5 1956 1957 1958 1959 ...
##  $ Global.plastics.production..million.tonnes...tonnes.: int  2000000 2000000 2000000 3000
000 3000000 4000000 5000000 5000000 6000000 7000000 ...
```

renaming the columns

```
colnames(df) = c("Year", "Gpp")
```

Scaling the data so that it’s in the range of 1-100, this will simplify to analyse and understand the computation

```
df$Gpp <- rescale(df$Gpp, to = c(1,100))
head(df)
```

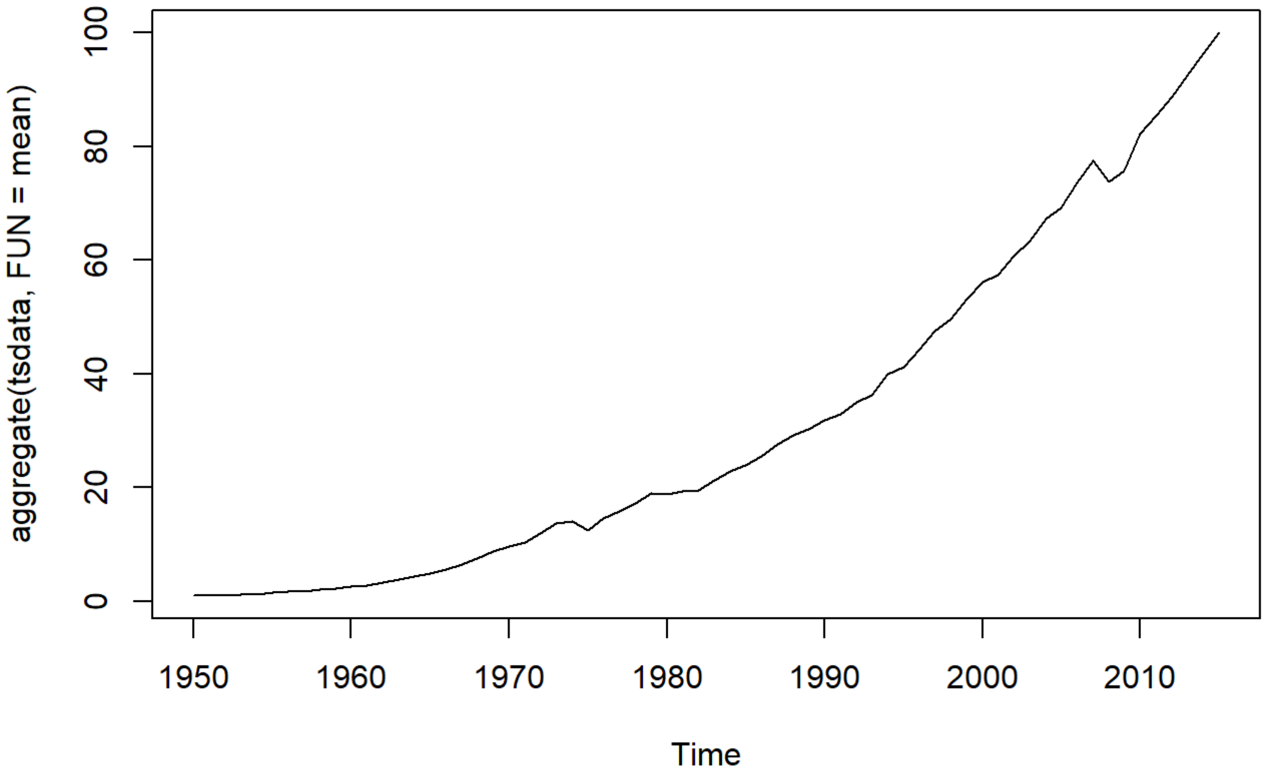
```
##   Year      Gpp
## 1 1950 1.000000
## 2 1951 1.000000
## 3 1952 1.000000
## 4 1953 1.261214
## 5 1954 1.261214
## 6 1955 1.522427
```

```
df$Gpp
```

```
##  [1]   1.000000   1.000000   1.000000   1.261214   1.261214   1.522427
##  [7]   1.783641   1.783641   2.044855   2.306069   2.567282   2.828496
## [13]   3.350923   3.873351   4.395778   4.918206   5.701847   6.485488
## [19]   7.530343   8.836412   9.620053  10.403694  11.970976  13.799472
## [25]  14.060686  12.493404  14.583113  15.889182  17.195251  19.023747
## [31]  18.762533  19.284960  19.546174  21.374670  22.941953  23.986807
## [37]  25.554090  27.643799  29.211082  30.255937  31.823219  32.868074
## [43]  34.957784  36.263852  39.920844  41.226913  44.361478  47.496042
## [49]  49.585752  53.242744  56.116095  57.422164  60.817942  63.430079
## [55]  67.348285  69.176781  73.617414  77.535620  73.878628  75.707124
## [61]  82.237467  85.372032  88.767810  92.424802  96.343008 100.000000
```

Converting the data into a time series

```
tsdata <- ts(df$Gpp, frequency = 1, start = 1950, end = 2015)
plot(aggregate(tsdata, FUN = mean))
```

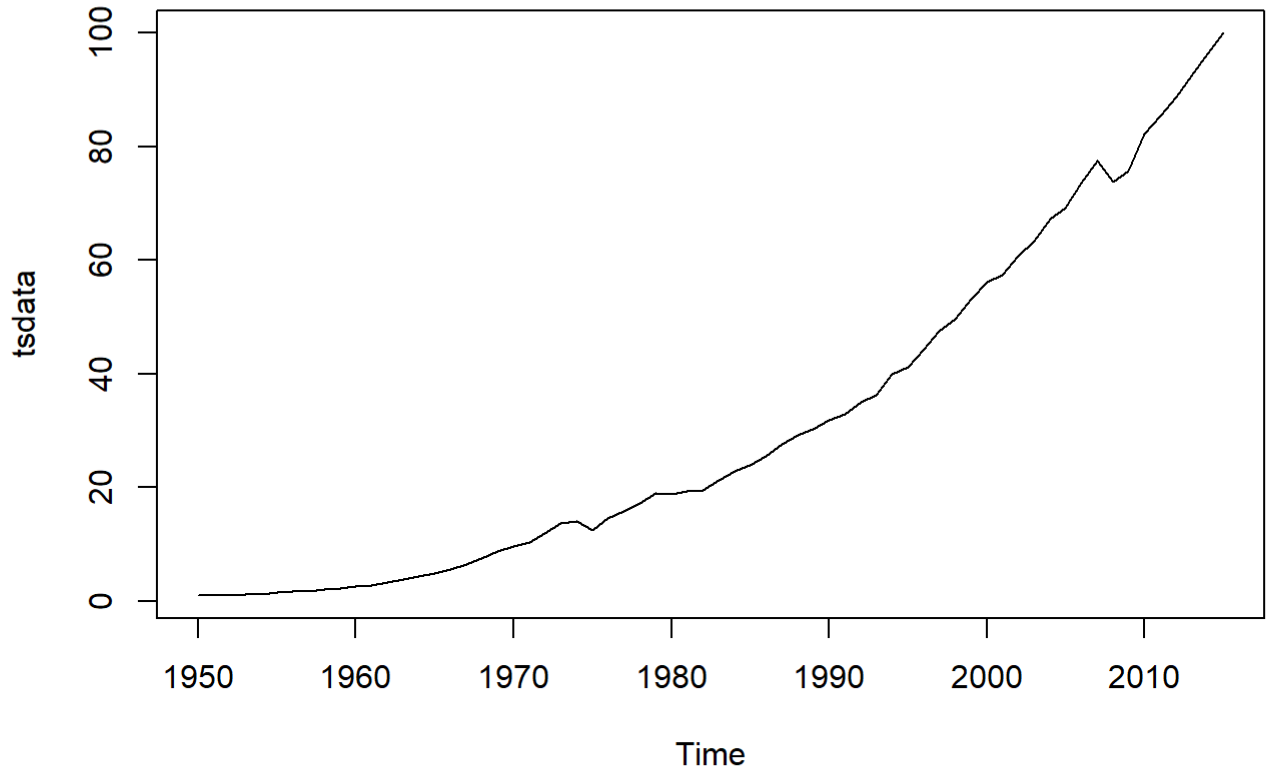


Splitting the data into training set and test set

```
trainingdata <- ts(df$Gpp, frequency = 1, start = 1950, end = 2005)
testdata <- tail(df, 10)
```

Plotting the data, from the plot we can observe that the data follows an exponential trend

```
plot(tsdata)
```



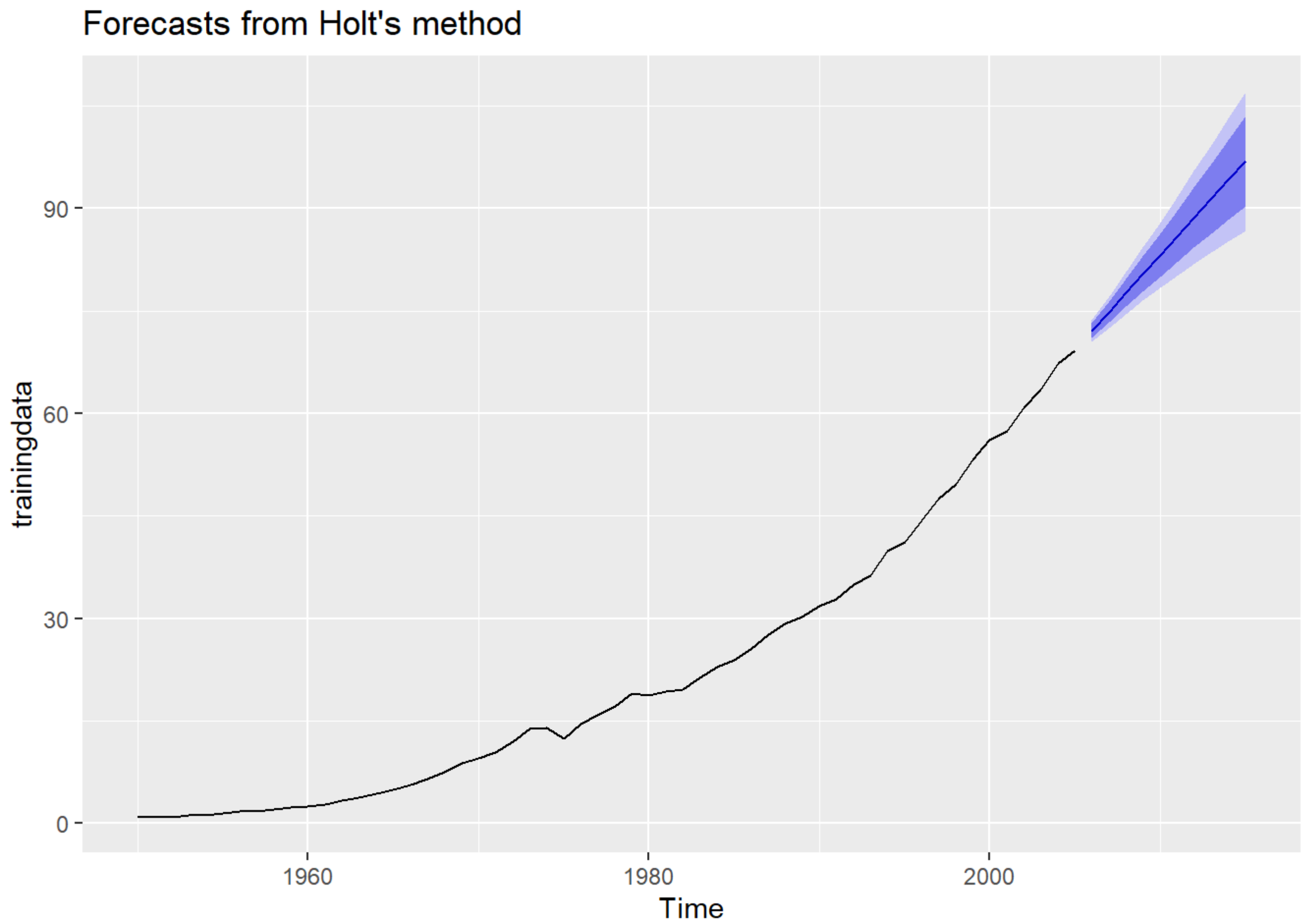
Building the prediction model using the Holt's Exponential Smoothing method and we are training our model using the train dataset

```
model <- holt(trainingdata, h = 10)
summary(model)
```

```
##
## Forecast method: Holt's method
##
## Model Information:
## Holt's method
##
## Call:
##  holt(y = trainingdata, h = 10)
##
##   Smoothing parameters:
##     alpha = 0.7886
##     beta  = 0.2397
##
##   Initial states:
##     l = 0.8611
##     b = 0.1007
##
##   sigma:  0.8175
##
##      AIC     AICc      BIC
## 208.7041 209.9041 218.8308
##
## Error measures:
##                     ME      RMSE       MAE      MPE     MAPE      MASE
## Training set 0.1979264 0.7877833 0.5684331 1.587652 4.584818 0.4352245
##                    ACF1
## Training set -0.0460277
##
## Forecasts:
##      Point Forecast    Lo 80     Hi 80    Lo 95     Hi 95
## 2006       72.11623 71.06853  73.16393 70.51392  73.71854
## 2007       74.87316 73.37040  76.37591 72.57489  77.17142
## 2008       77.63008 75.62436  79.63580 74.56260  80.69757
## 2009       80.38701 77.83403  82.93998 76.48257  84.29144
## 2010       83.14393 80.00250  86.28537 78.33952  87.94835
## 2011       85.90086 82.13232  89.66939 80.13738  91.66434
## 2012       88.65778 84.22569  93.08988 81.87948  95.43609
## 2013       91.41471 86.28446  96.54497 83.56866  99.26076
## 2014       94.17164 88.31023 100.03304 85.20739 103.13589
## 2015       96.92856 90.30442 103.55271 86.79780 107.05932
```

plotting the data with predictions on the training data

```
autoplot(model)
```



Predicted values

```
pred = forecast(model)
pred$mean
```

```
## Time Series:
## Start = 2006
## End = 2015
## Frequency = 1
##  [1] 72.11623 74.87316 77.63008 80.38701 83.14393 85.90086 88.65778 91.41471
##  [9] 94.17164 96.92856
```

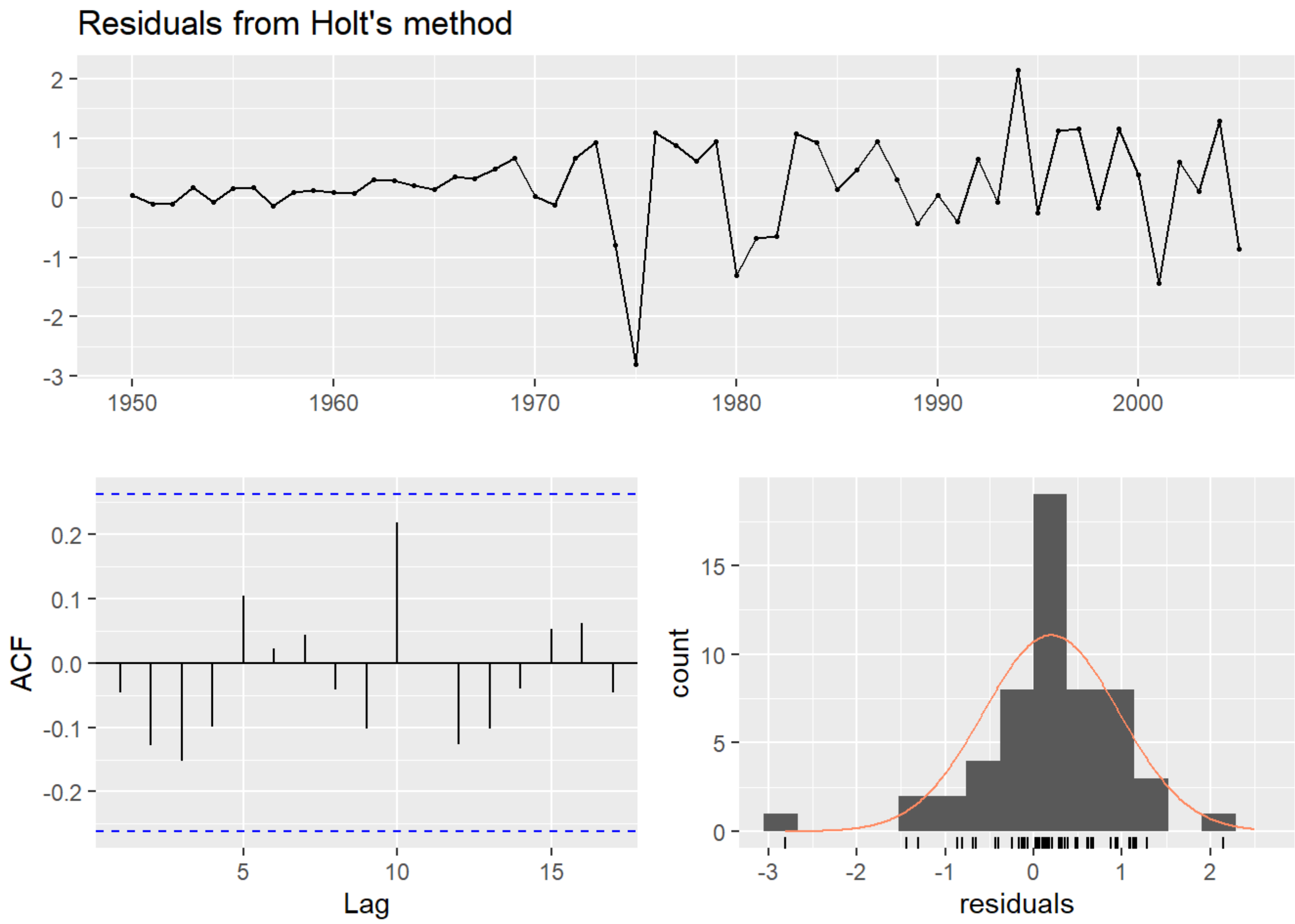
Accuracy of the model predictions

```
accuracy(pred$mean, testdata$Gpp)
```

```
##                  ME     RMSE      MAE         MPE    MAPE
## Test set 0.06599492 2.481921 2.039321 -0.09660238 2.49962
```

residuals

```
checkresiduals(model)
```



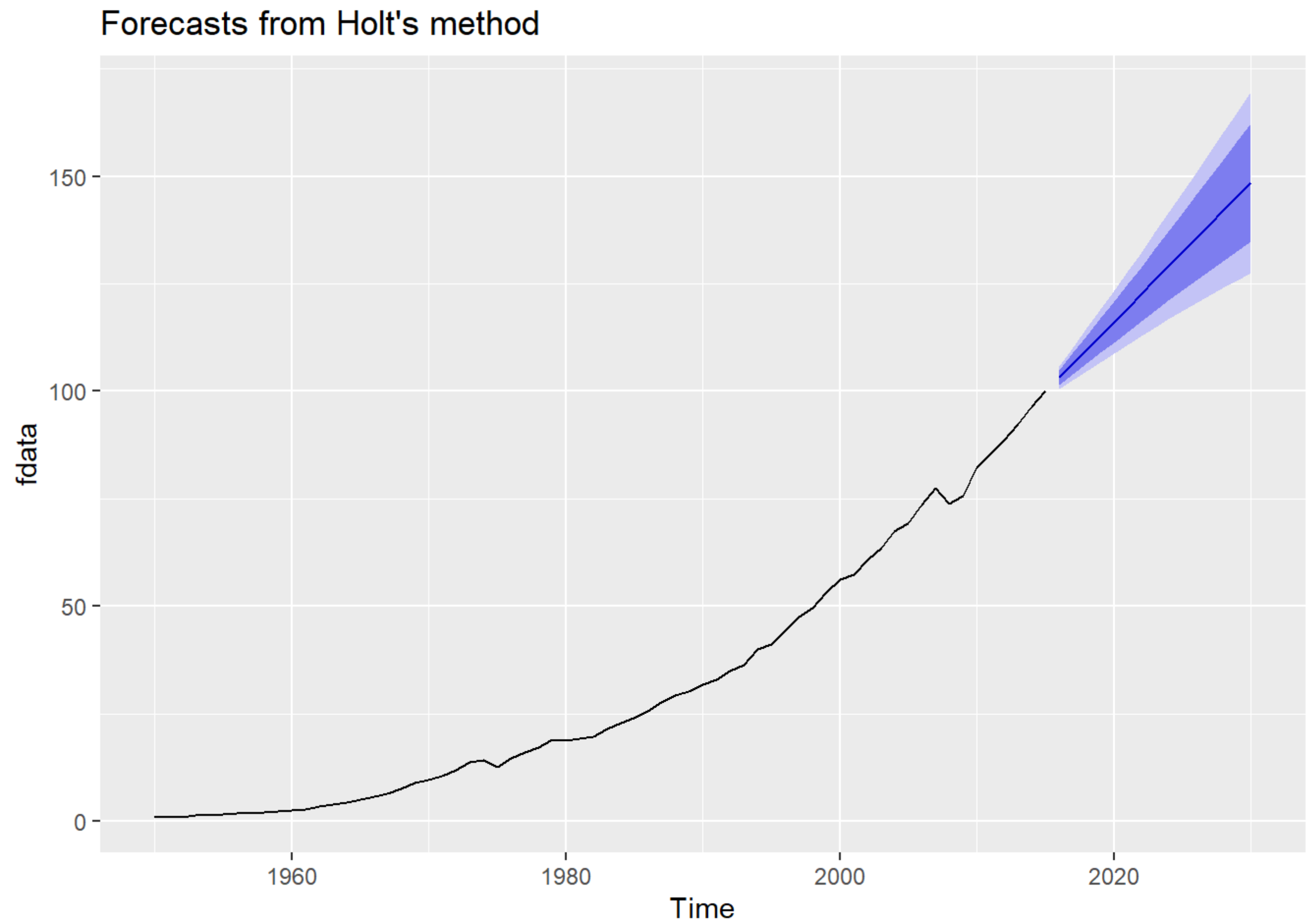

```
##
##  Ljung-Box test
##
## data:  Residuals from Holt's method
## Q* = 8.2137, df = 6, p-value = 0.2229
##
## Model df: 4.   Total lags used: 10
```

Prediction from 2015 for the next 10 years, till 2025

```
fdata <- ts(df$Gpp, frequency = 1, start = 1950, end = 2015)
ft <- holt(fdata, h = 15)
summary(ft)
```

```
##
## Forecast method: Holt's method
##
## Model Information:
## Holt's method
##
## Call:
##  holt(y = fdata, h = 15)
##
##   Smoothing parameters:
##     alpha = 0.912
##     beta  = 0.153
##
##   Initial states:
##     l = 0.7733
##     b = 0.2039
##
##   sigma:  1.3299
##
##      AIC     AICc      BIC
## 320.0253 321.0253 330.9736
##
## Error measures:
##                     ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 0.3007674 1.288977 0.793393 1.130238 4.572416 0.4689469
##                     ACF1
## Training set -0.02402403
##
## Forecasts:
##      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2016       103.1880 101.4837 104.8924 100.5814 105.7946
## 2017       106.4284 103.9386 108.9183 102.6206 110.2363
## 2018       109.6688 106.4272 112.9104 104.7112 114.6264
## 2019       112.9093 108.9134 116.9051 106.7981 119.0204
## 2020       116.1497 111.3839 120.9154 108.8611 123.4383
## 2021       119.3901 113.8331 124.9470 110.8915 127.8887
## 2022       122.6305 116.2583 129.0027 112.8850 132.3760
## 2023       125.8709 118.6581 133.0838 114.8398 136.9020
## 2024       129.1113 121.0321 137.1906 116.7552 141.4675
## 2025       132.3517 123.3801 141.3234 118.6308 146.0727
## 2026       135.5922 125.7024 145.4819 120.4671 150.7172
## 2027       138.8326 127.9992 149.6660 122.2643 155.4008
## 2028       142.0730 130.2708 153.8752 124.0231 160.1229
## 2029       145.3134 132.5177 158.1091 125.7441 164.8827
## 2030       148.5538 134.7403 162.3674 127.4278 169.6798
```

```
autoplot(ft)
```



An RMSE value of 2.4 indicates an accuracy of 97% in our predictions.

According to our predictions the global plastic production is expected to double in the next 10 years.