

AnSeq_groupProject

- Introduction
 - [Data]
- Exploratory Data Analysis (EDA)
 - Visualization
 - Discussion
- Indicators
- [Simple Model]
 - Discussion
 - Fitting
- [Exponential Smoothing]
 - Discussion
 - Fitting
 - Analysis
 - Comparison with simple model
- ETS AND AUTO-ARIMA
 - Fitting
 - Analysis
 - Discussion
 - Comparison with other models
- Conclusion

Introduction

The objective of this report is to evaluate various existing forecasting methods and model families and to improve understanding of the specific advantages and disadvantages of each approach. The project is conducted on an existing time series dataset within the Mcomp package. The following analysis and model evaluation is based on the monthly broadcast dataset “N1483”.

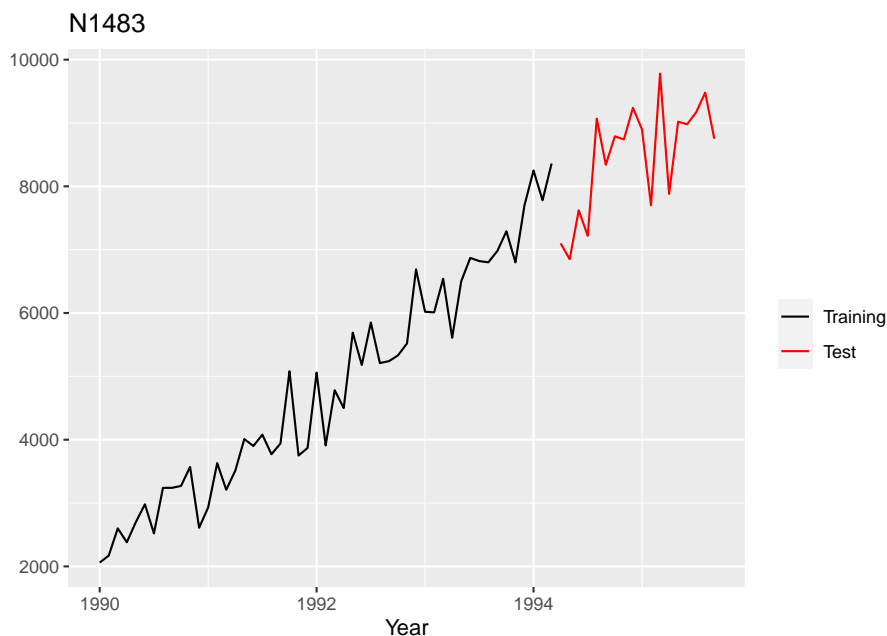
After an initial exploratory analysis of the data, three different model families of increasing complexity are evaluated and tested. The individual model performance is assessed by the specific family characteristics as well as general performance indicators. In addition, two appropriate indicators have been defined to facilitate the comparison of models from different model families.

```
subset(M3, "monthly")["N1483"]
```

```
##          Type of data  
## Period    MICRO  
##  MONTHLY      1
```

```
data <- subset(M3, "monthly")["N1483"]
```

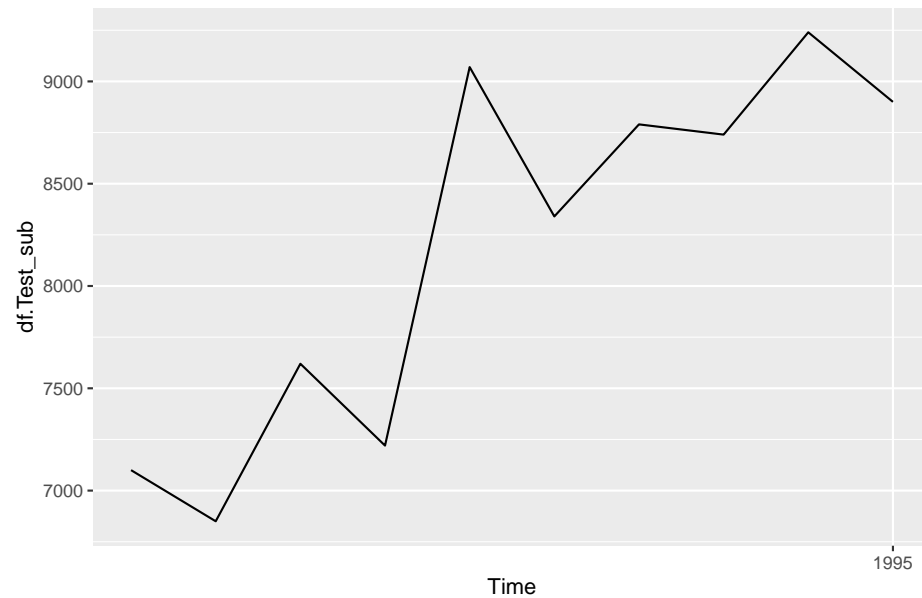
```
autoplot(data$N1483)
```



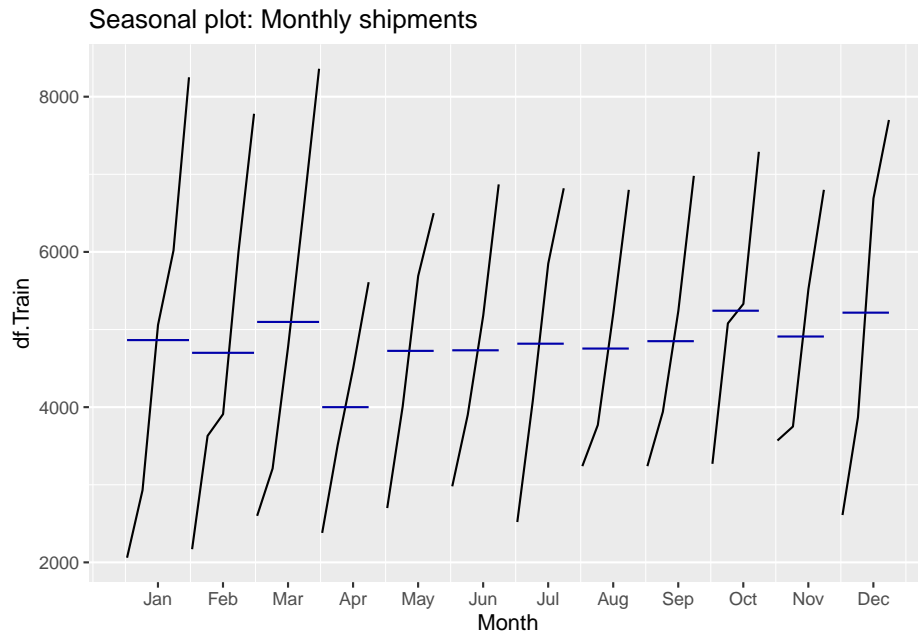
Exploratory Data Analysis (EDA)

Visualization

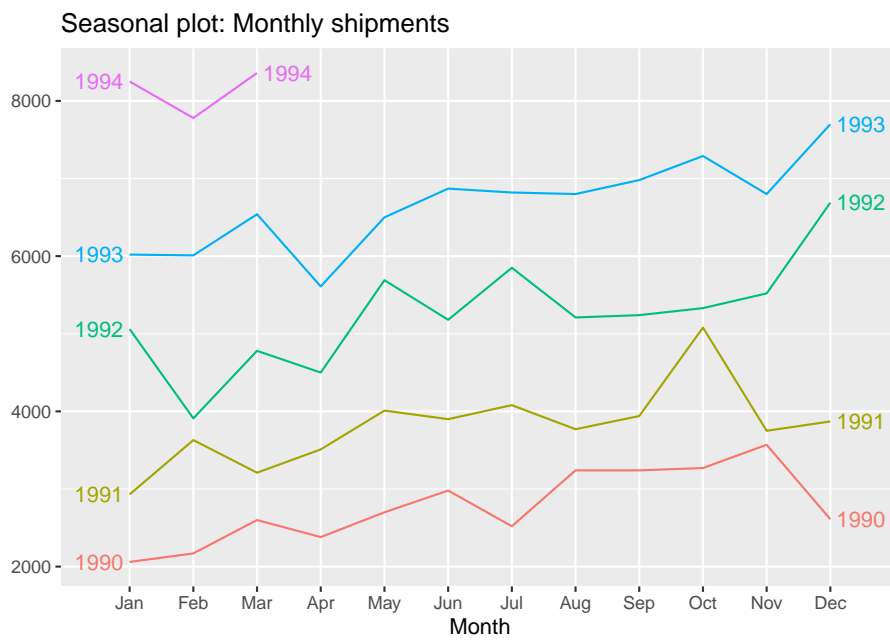
```
#use variable for future tasks  
df.Train <- data$N1483$x  
df.Test <- data$N1483$xx  
df.Test_sub <- window(data$N1483$xx , end=1995)  
autoplot(df.Test_sub)
```



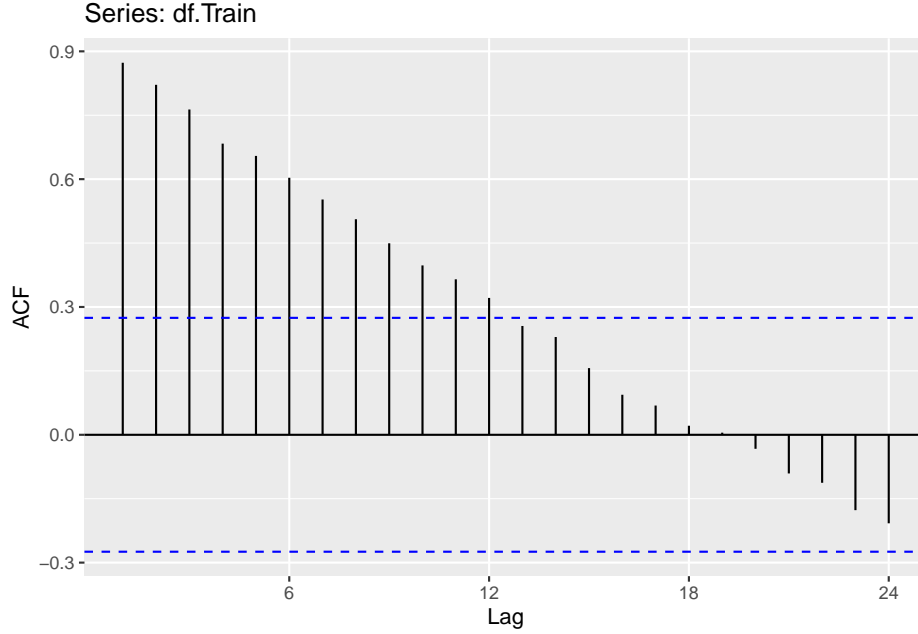
```
ggsubseriesplot(df.Train) +  
  ggtitle("Seasonal plot: Monthly shipments")
```



```
ggseasonplot(df.Train, year.labels=TRUE, year.labels.left=TRUE) +
  ggtitle("Seasonal plot: Monthly shipments")
```



```
ggAcf(df.Train)
```



Discussion

The following section outlines the selected dataset in an explanatory manner to provide a better understanding of the underlying data and to discover patterns and anomalies in the chosen time series.

The chosen dataset consists of 126 entries containing shipment data from 1985 to 1995 with a monthly frequency. As being part of the M3 competition data set, the given time series is already split up in a training and testing set containing 108 and 18 entries respectively.

As represented in the chart N1483, the chosen time series indicates a strong trend as the “amount” of shipments are growing constantly from 1985 to 1995 with a slightly stagnating trend towards the end of the time series. Furthermore, the graph implies that the time series has no seasonality. The mentioned constant trend and lack of seasonality can also be observed in appendix x and y. Appendix x clearly depicts that there is a constant growing trend in each month over the years in the given time series. In addition, this constant increasing trend over the months underlines that there is no seasonality in sequential data. The same observation can be made in appendix y, which clearly shows that there is now no seasonality as the plotted yearly shipments have no common peaks or lows. Instead the yearly lines tend to increase during the year. This growth over the year leads to the gaps between the line graphs, which represent the shipment over a year. Those described features emphasize the fact that the time series has an increasing trend and no seasonality.

The correlogram as well as the seasonal lag plot in appendix x and z, strengthen the assumption of an increasing trend and a missing seasonality. The autocorrelogram (ACF) in appendix x manifest that the autocorrelations for smaller lags are large and positive, which is typically for trendy data.

Indicators

In order to measure the performance of each model, two indicators were defined.

For the indicators the choice was the mean absolute error (MAE) and the root mean square error (RMSE). The RMSE calculates the standard deviation and the spreading of the residuals. The goal is to have an RMSE as low as possible with a rough goal of 500. The MAE is the mean absolute error which includes the effect of the absolute errors and is not strongly influenced by outliers. The goal was set to 500 as well. #

Simple Model

Discussion

As described in the exploratory analysis, the time series contains a upwards trend as well as no seasonality. Due to these characteristics of the time series, the average method model can be neglected preliminary.

Based on the characteristics of the time series, the expectation is that the random walk with drift forecasts the best compared to other simple models as this method forecasts based on the last available observation with the added average of the time series. This characteristic more likely describes the increasing trend in the time series.

The following table illustrates the performance of a the naïve, seasonal naïve and random walk with drift method based on the chosen indicators. The table shows that MAE and RSME of the seasonally naïve are quite high. Whereas the random walk method with drift and the naïve model have the lowest indicators.

```
naive_shipment <- naive(df.Train, h=length(df.Test))
snaive_shipment <- snaive(df.Train, h=length(df.Test))
rwf_shipment <- rwf(df.Train, drift = TRUE, h=length(df.Test))

accSimple <- rbind(
  accuracy(naive_shipment, df.Test),
  accuracy(snaive_shipment, df.Test),
  accuracy(rwf_shipment, df.Test)
)
rownames(accSimple) <- c("Naive Model - Training", "Naive Model - Test",
  "Seasonal Naive Model - Training",
  "Seasonal Naive Model- Test",
  "Random walk with drift - Training",
  "Random walk with drift -Test ")

accSimple[, c(2:3)]
```

##	RMSE	MAE
## Naive Model - Training	610.2065	487.200
## Naive Model - Test	855.6901	765.000
## Seasonal Naive Model - Training	1461.9481	1344.103
## Seasonal Naive Model- Test	1710.9338	1525.000
## Random walk with drift - Training	597.0561	466.960
## Random walk with drift -Test	1254.0269	1086.444

As visualized in appendix y the residuals of the seasonal naïve forecasting method remain rather high. The high residuals of the seasonal naïve model are expected as the chosen time series has no seasonality. Eventhough the seasonal naïve model has a p-value of 0.6792 with a Q^* of 7.4828 and therefore is not rejected by null hypothesis of the Ljung-Box test, the model is no suitable as the residuals are too high.

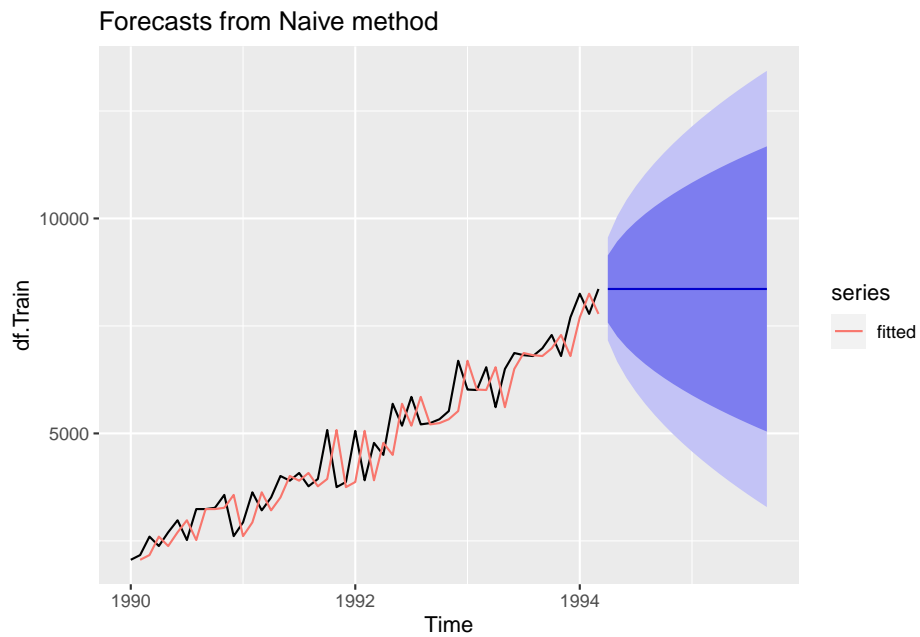
Comparing the indicators of the naïve and the random walk method, we see that the random walk method is slightly better in training accuracy. However, the naïve model performs better on test data than the random walk with drift. This can be explained by the fact that the slope of the trend decreases slightly towards the end of the time series.

The indicators of the naïve and the random walk method are clearly better than the seasonal naïve methode. Although for both models, the null hypothesis of the Ljung-Box test has to be rejected as the p-value is 0.006729 with a Q^* of 24.348 and 0.003784 with a Q^* of 24.348 respectively. Thus, both forecasts are biased. Consequently, none of the simple models is suitable for the given time series.

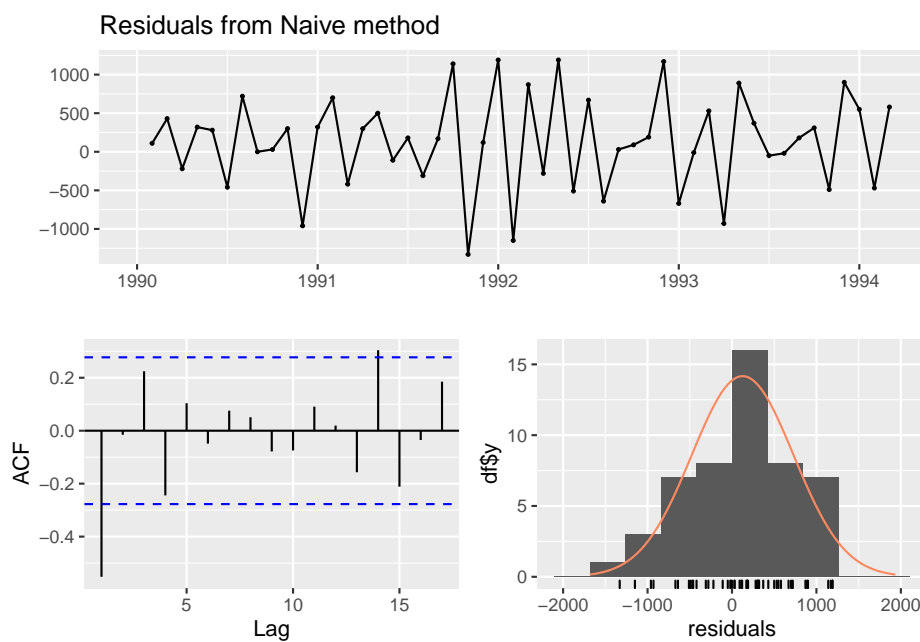
Fitting

```
autoplot(naive_shipment)+autolayer(fitted(naive_shipment), series = "fitted")
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```



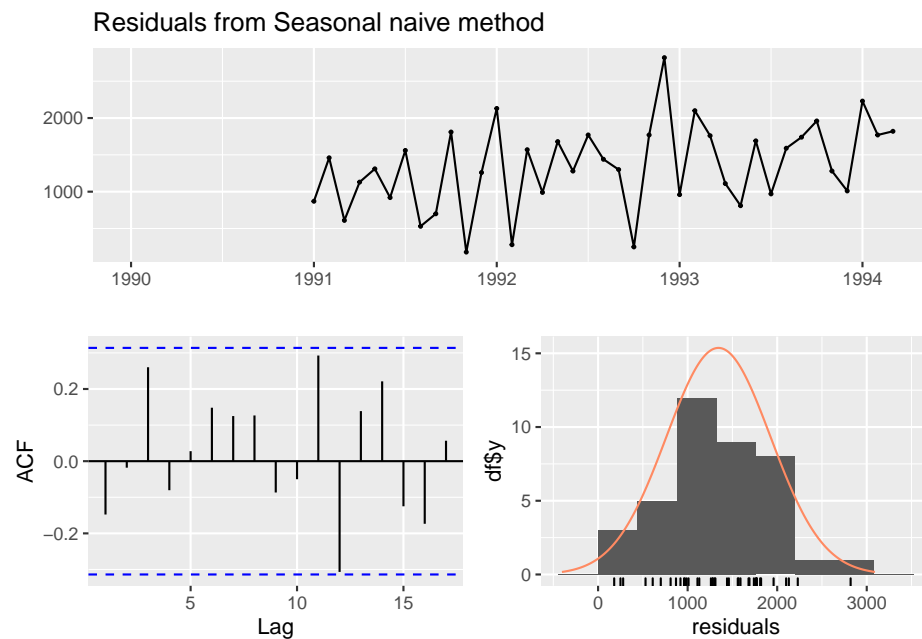
```
checkresiduals(naive_shipment)
```



```
##
```

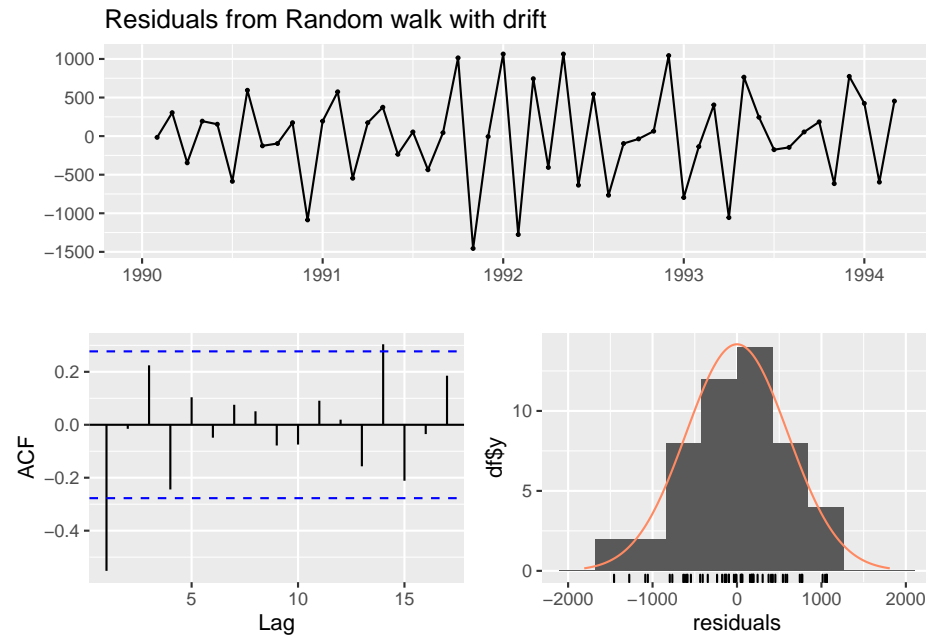
```
## Ljung-Box test
##
## data: Residuals from Naive method
## Q* = 24.348, df = 10, p-value = 0.006729
##
## Model df: 0. Total lags used: 10
```

```
checkresiduals(snaive_shipment)
```



```
##
## Ljung-Box test
##
## data: Residuals from Seasonal naive method
## Q* = 7.4828, df = 10, p-value = 0.6792
##
## Model df: 0. Total lags used: 10
```

```
checkresiduals(rwf_shipment)
```

```
##
##  Ljung-Box test
##
## data:  Residuals from Random walk with drift
## Q* = 24.348, df = 9, p-value = 0.003784
##
## Model df: 1.    Total lags used: 10
```

Exponential Smoothing (Kevin)

Discussion

For the different methods of the exponential smoothing, one may choose the one which fits best to the available data. In our case due to the positive trend and the expected stagnation (which is normal for shipments in a company due to various reasons) we choose the holt-Winters damped multiplicative model but for a better overview we fitted all holt-winters model.

Analysis

```
df.hwAdd <- hw(df.Train, seasonal = "additive",
              h = length(df.Test))
df.hwAddDamped <- hw(df.Train, seasonal = "additive",
                    damped = TRUE,
                    h = length(df.Test))
df.hwMulti <- hw(df.Train, seasonal = "multiplicative",
                h = length(df.Test))
df.hwMultiDamped <- hw(df.Train, seasonal = "multiplicative",
                      damped = TRUE, h = length(df.Test))
```

```
#plot

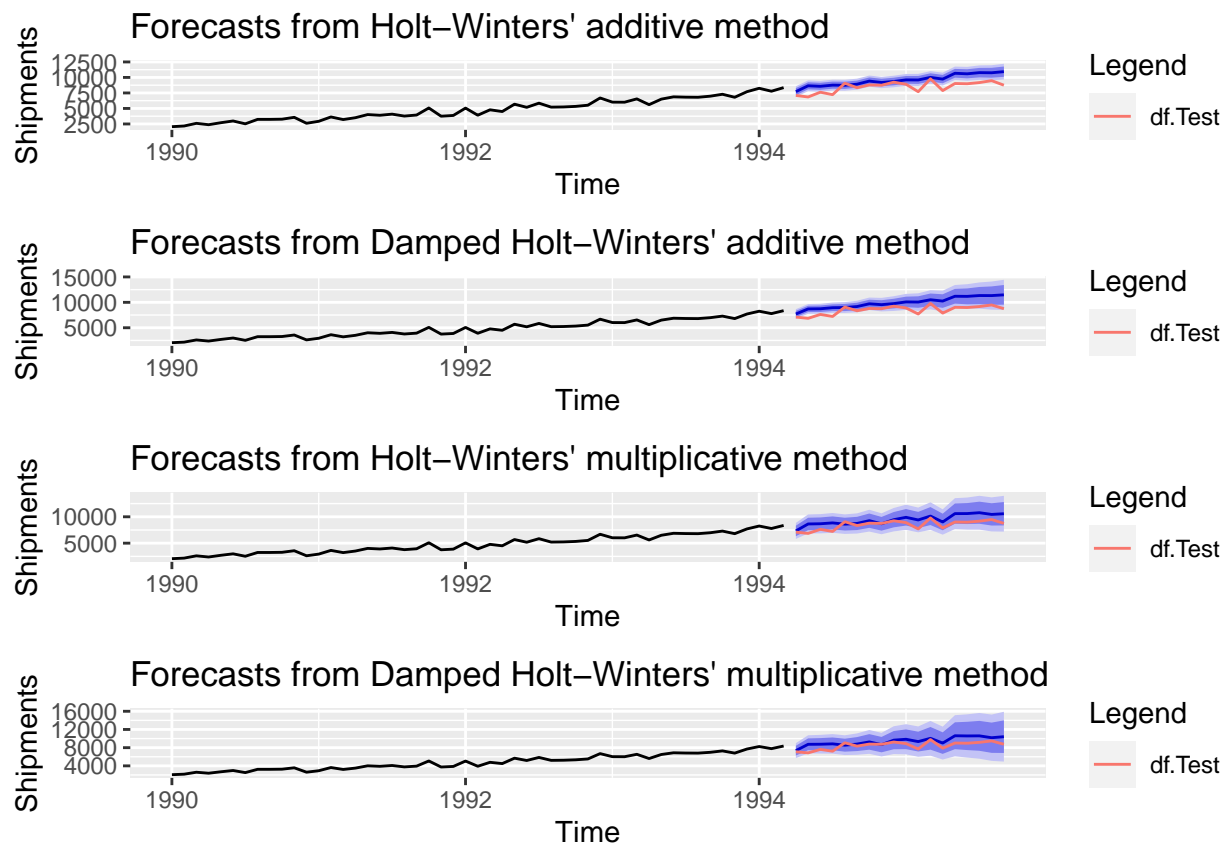
df.hwAddPlot <- autoplot(df.hwAdd, color="AdditiveMethod") +
  labs(x="Time", y="Shipments", color = "Legend") +
  autolayer(df.Test)

df.hwAddDampedPlot <- autoplot(df.hwAddDamped, color="AdditiveMethod") +
  labs(x="Time", y="Shipments", color = "Legend") +
  autolayer(df.Test)

df.hwMultiPlot <- autoplot(df.hwMulti, color="MultiMethod") +
  labs(x="Time", y="Shipments", color = "Legend") +
  autolayer(df.Test)

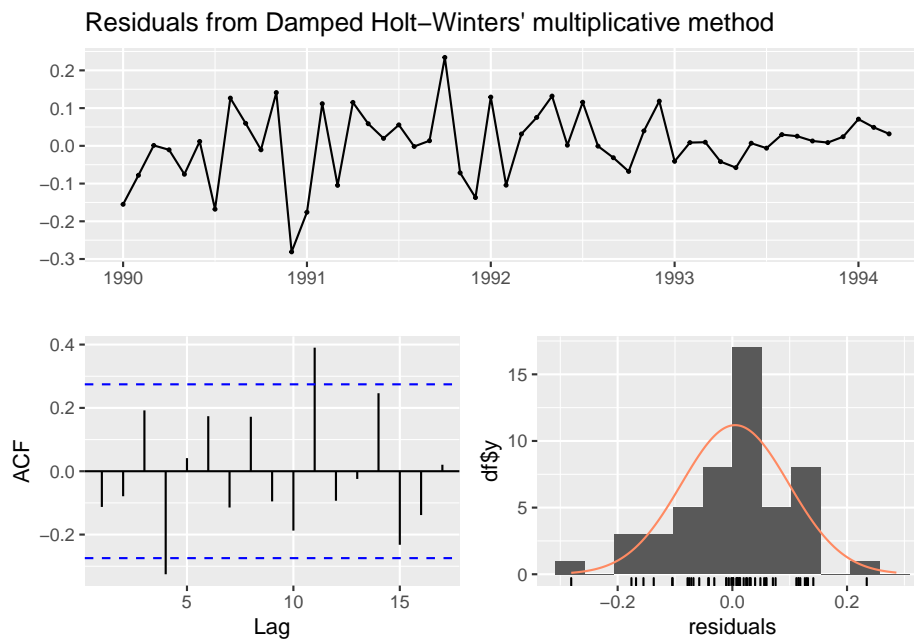
df.hwMultiDampedPlot <- autoplot(df.hwMultiDamped, color="MultiMethod") +
  labs(x="Time", y="Shipments", color = "Legend") +
  autolayer(df.Test)

grid.arrange(df.hwAddPlot, df.hwAddDampedPlot,
  df.hwMultiPlot, df.hwMultiDampedPlot, nrow=4)
```



As can be seen from the above plots, the Damped Holt-Winter's multiplicative Method seems to show a similar forecast to the test set. Before evaluating the accuracy we want to check the residuals.

```
checkresiduals(df.hwMultiDamped)
```



Residuals

```
##
##  Ljung-Box test
##
## data:  Residuals from Damped Holt-Winters' multiplicative method
## Q* = 38.801, df = 3, p-value = 1.912e-08
##
## Model df: 17.   Total lags used: 20
```

The Ljung-Box test shows a p-value $\ll 0.05$ and therefore the NULL-Hypothesis is rejected. The residuals might be auto-correlated.

From the ACF a pattern is difficult to detect but a significant-autocorrelation is shown at lag 4 and 11. R_{12} is small and indicates no seasonality.

Accuracy of the fitted values for the training data

```
accDf <- rbind(
  accuracy(df.hwAdd$fitted, df.Train),
  accuracy(df.hwAddDamped$fitted, df.Train),
  accuracy(df.hwMulti$fitted, df.Train),
  accuracy(df.hwMultiDamped$fitted, df.Train)
)
rownames(accDf) <- c("hw Additive", "hw damped Additive",
  "hw Multiplicative", "hw damped Multiplicative")

print(accDf[,2:3])
```

```
##                                RMSE                                MAE
```

```
## hw Additive          365.9992 294.5030
## hw damped Additive   393.5663 305.9258
## hw Multiplicative    357.3381 266.5627
## hw damped Multiplicative 377.8383 290.0058
```

It can be seen that the multiplicative Method shows the best results for the fitting in this scenario.

But how are the forecasting results for these models. For this approach the accuracy of the models mean prediction with the test data is calculated:

```
accDf <- rbind(
  accuracy(df.hwAdd$mean, df.Test),
  accuracy(df.hwAddDamped$mean, df.Test),
  accuracy(df.hwMulti$mean, df.Test),
  accuracy(df.hwMultiDamped$mean, df.Test)
)
rownames(accDf) <- c("hw Additive", "hw damped Additive",
  "hw Multiplicative", "hw damped Multiplicative")

print(accDf[,2:3])
```

```
##              RMSE      MAE
## hw Additive      1285.833 1109.7747
## hw damped Additive 1644.109 1456.1444
## hw Multiplicative 1178.732 1007.2867
## hw damped Multiplicative 1147.369 988.3026
```

For the forecasting the damped multiplicative model shows the best RMSE with 1147.369. Nevertheless, the prediction interval is higher than the undamped model and should therefore be considered. The reason why the damped multiplicative model outperformed for forecasting can be explained as followed: The data which are handled are shipments over the years. Although the increase in shipments over the years seems linear, a company may come to stagnation for various reasons (political, economical, logistical). Therefore with the usage of the damped model this stagnation is considered which can be seen for the ϕ parameter in the following block.

```
summary(df.hwMultiDamped$model)$par
```

##	alpha	beta	gamma	phi	l	b
##	1.572216e-01	5.089893e-02	1.012398e-04	9.800000e-01	2.198914e+03	1.614285e+02
##	s0	s1	s2	s3	s4	s5
##	1.029713e+00	9.391970e-01	1.020829e+00	9.846672e-01	9.745092e-01	1.026325e+00
##	s6	s7	s8	s9	s10	
##	1.036962e+00	1.054998e+00	9.063895e-01	1.026080e+00	9.660842e-01	

The ϕ Parameter has the value 0.98 which shows the change of the slope over time.

Nevertheless, this is just a hypothesis and more time has to be invested to justify this.

Here the original and fitted values from the damped multiplicative model.

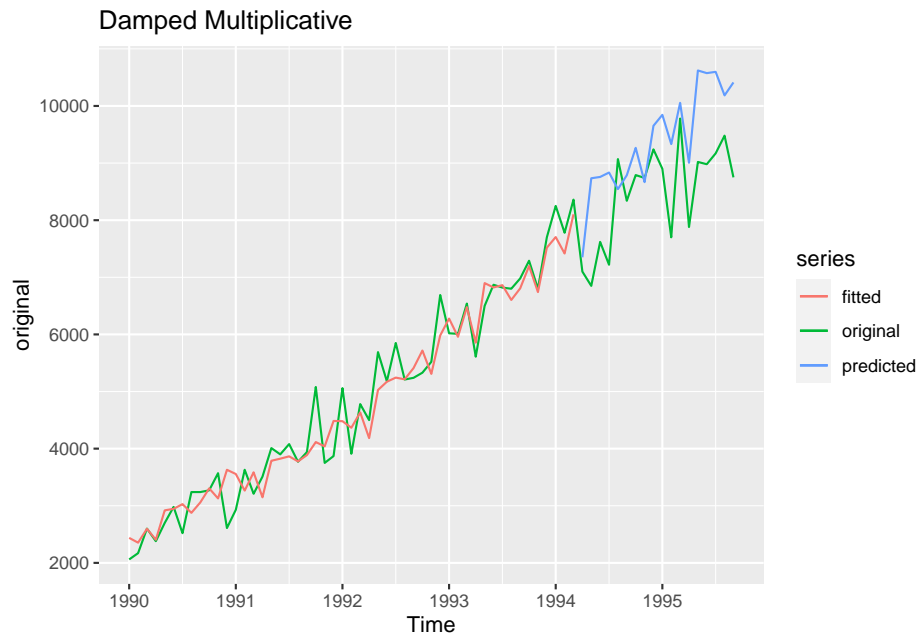
```
original <- ts(c(df.Train,df.Test), start=start(df.Train), frequency=frequency(df.Train))

autoplot(original, series = "original") +
```

```

autolayer(df.hwMultiDamped$fitted, series = "fitted") +
autolayer(df.hwMultiDamped$mean, series = "predicted") +
ggtitle("Damped Multiplicative")

```



Comparison with simple model

The performance indicators in the table above illustrate that the damped holts-Winter method outperforms the simple mode on the training set. Surprisingly, the Simple Model performs better than Exponential Smoothing Models on the test data. This is surprising, since Simple Models are normally only applicable for very simple models and short prediction intervals.

```

accDf <- rbind(
  accuracy(naive_shipment, df.Test),
  accuracy(df.hwMultiDamped, df.Test) #Anpassung des Models
)
rownames(accDf) <- c("Naive Model - Training",
  "Naive Model - Test", "hw damped Multiplicative - Training",
  "hw damped Multiplicative - Test")

accDf[, c(2:3)]

```

##	RMSE	MAE
## Naive Model - Training	610.2065	487.2000
## Naive Model - Test	855.6901	765.0000
## hw damped Multiplicative - Training	377.8383	290.0058
## hw damped Multiplicative - Test	1147.3688	988.3026

In the previous chapters it was stated that the chosen time series has high fluctuation and shows a general stagnation. This behavior is hard to model with simple models. Additionally we can conclude that exponential smoothing models are not suitable either since the Exponential Smoothing Models performed even worse in our test-set accuracy.

ETS AND AUTO-ARIMA

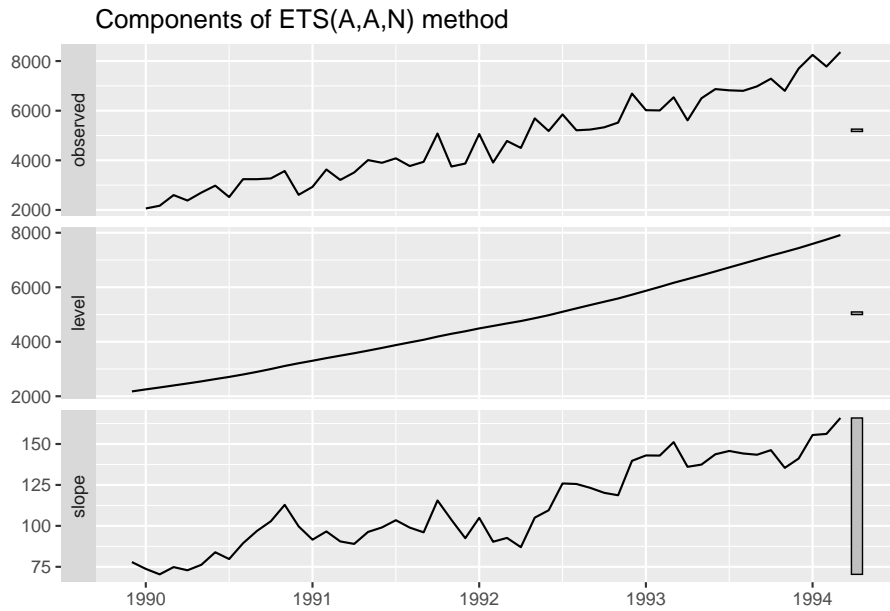
Analysis

ETS

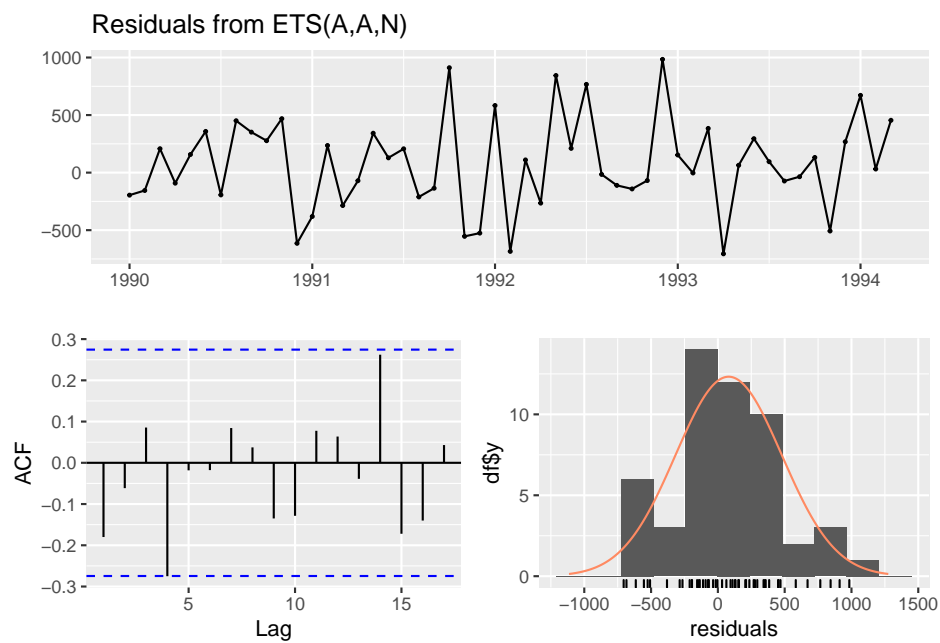
```
df.ets <- ets(df.Train)
summary(df.ets)
```

```
## ETS(A,A,N)
##
## Call:
## ets(y = df.Train)
##
## Smoothing parameters:
##   alpha = 0.0214
##   beta  = 0.0214
##
## Initial states:
##   l = 2177.7851
##   b = 77.8987
##
## sigma: 419.3386
##
##      AIC      AICc      BIC
## 822.3027 823.6361 831.9619
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set 80.73092 402.5582 316.9789 0.8628456 7.31002 0.2358294 -0.1799548
```

```
autoplot(df.ets)
```



```
checkresiduals(df.ets)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,A,N)
## Q* = 9.5401, df = 6, p-value = 0.1454
##
## Model df: 4.    Total lags used: 10
```

The ets approach defined holts damped linear trend method to be the most suitable candidate.

The output of the ETS is explained as follows:

The first A stands for the additive Noise, the second A defines an additive trend and the third N shows that there is no seasonality. Additionally the model does not add the damped factor ϕ .

The p-value is > 0.5 and the AFC doesn't show any correlation which is good. The Residual spread looks normal with no clear bias visible.

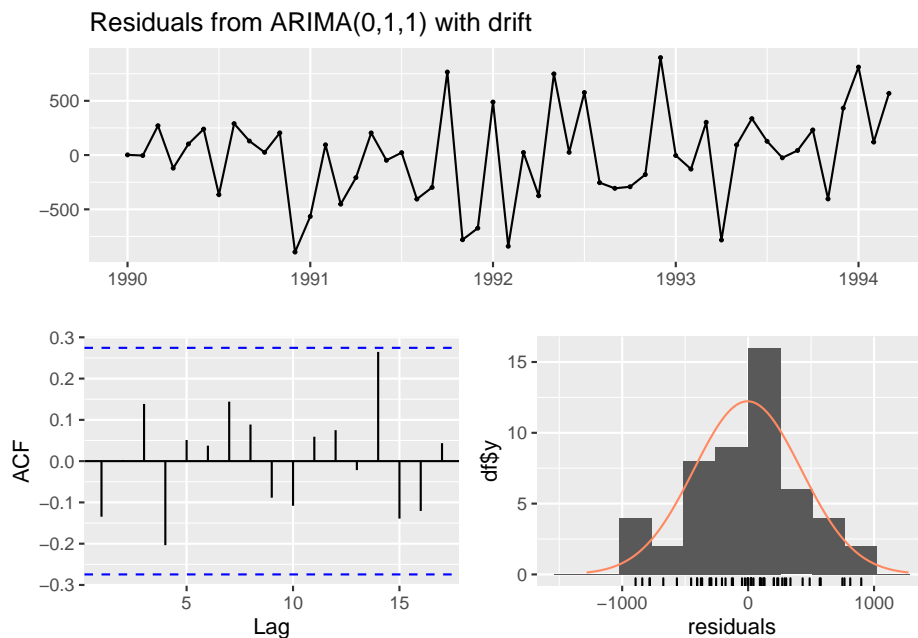
auto.arima

The data doesn't seem to have changing variance and therefore the log10 transformation is not used.

```
## Series: df.Train
## ARIMA(0,1,1) with drift
##
## Coefficients:
##          ma1      drift
##        -0.8696  115.5011
## s.e.    0.0849   9.3001
##
## sigma^2 = 189095: log likelihood = -374.38
## AIC=754.76  AICc=755.29  BIC=760.5
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -4.482698 421.8673 325.1873 -1.376498 7.360849 0.2419364
##              ACF1
## Training set -0.1346102
```

The following plot shows the checkresidual function for the autoarima model.

```
checkresiduals(df.autoarima)
```




```
##
## Ljung-Box test
##
## data: Residuals from ARIMA(0,1,1) with drift
## Q* = 7.7275, df = 9, p-value = 0.5618
##
## Model df: 1. Total lags used: 10
```

The auto.arima approach defined ARIMA(0,1,1) with drift to be the best candidate. This means its a model with no autoregression (first 0), using the first derivative (second 1) and considering seasonality (first order moving average part). Additionally a constant is used which is shows as drift.

The p-value is » 0.5 and the AFC doesnt show any correlation which is good. The Residual spread looks normal with no clear bias visible.

Perfomance analysis

Test set

```
df.etsFor <- df.ets %>% forecast(h=18)
df.autoarimaFor <- df.autoarima %>% forecast(h=18)

accDf <- rbind(
  accuracy(df.etsFor, df.Test),
  accuracy(df.autoarimaFor, df.Test)
)
rownames(accDf) <- c("ets - train", "ets - test", "autoarima - train",
  "autoarima - test")

print(accDf[,2:3])
```

```
##
## RMSE MAE
## ets - train 402.5582 316.9789
## ets - test 1231.0338 1047.5889
## autoarima - train 421.8673 325.1873
## autoarima - test 798.6903 661.7872
```

As we can see from the forecasting accuracy (MAE) on the test set, the autoarima model shows a better result than the ets approach by almost 400 error units.

Nevertheless, a better approach to choose the forecasting model is to run a crossvalidation with the original data containing train and test set.

CV on train & test set

```
#the result of ets gave an additive error, damped additive trend and no season holt's approach (holt)
holtError <- tsCV(original, holt, drift = TRUE, h=1, damped = TRUE)

#results of auto.arima gave ARIMA(0,1,1) with constant
armaFun <- function(x, h){forecast(Arima(x, order=c(0,1,1)),
  h=h, include.constant = TRUE)} #helper function
armaError <- tsCV(original, armaFun, h=1)
```

Holt CV RMSE

```
print(sqrt(mean(holtError^2, na.rm = TRUE)))
```

```
## [1] 569.8849
```

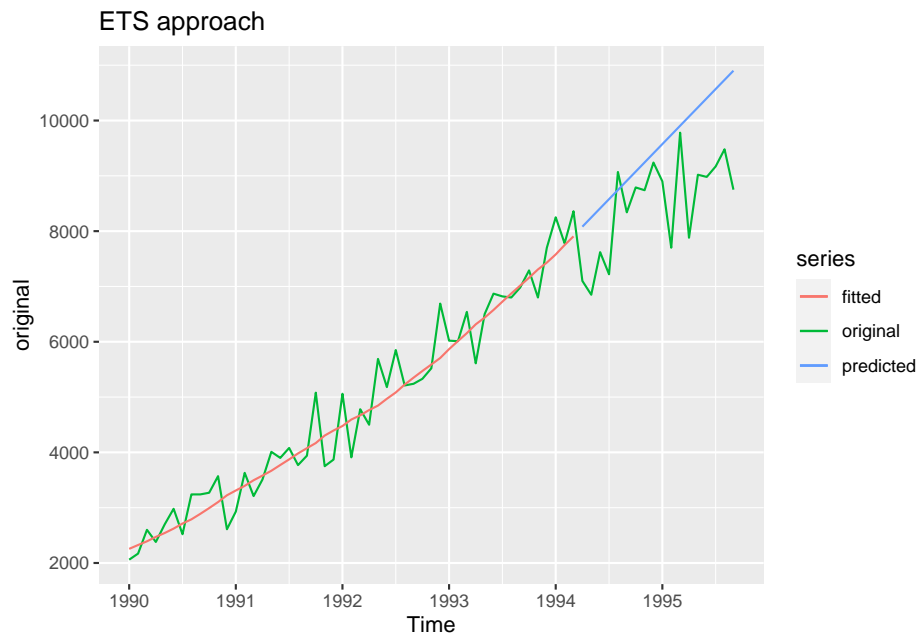
Arima CV RMSE

```
print(sqrt(mean(arimaError^2, na.rm = TRUE)))
```

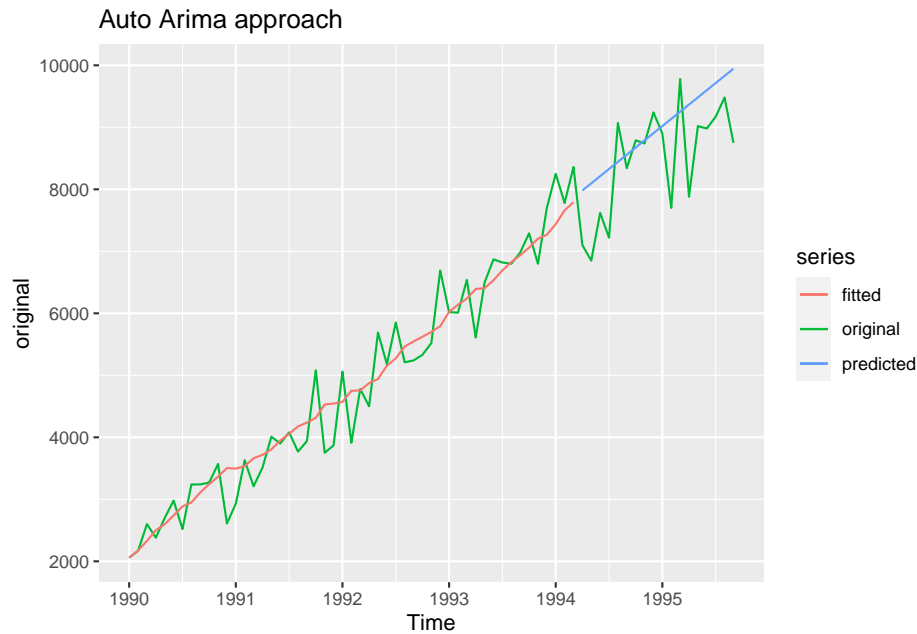
```
## [1] 623.9015
```

For the Cross-Validation with Train and Test set we see that the RMSE of the holt's linear trend is smaller than the Arima(0,1,1) approach. Nevertheless, with a difference of only 55.02 the decision for the better model is still difficult.

```
autoplot(original, series = "original") +  
  autolayer(df.ets$fitted, series = "fitted") +  
  autolayer(df.etsFor$mean, series = "predicted") + ggtitle("ETS approach")
```



```
autoplot(original, series = "original") +  
  autolayer(df.autoarima$fitted, series = "fitted") +  
  autolayer(df.autoarimaFor$mean, series = "predicted") + ggtitle("Auto Arima approach")
```



Comparison with other models

For the final comparison we take the models for each chapter which show the best results.

The analysis showed that the simple and the holt-Winter damped multiplicative model are the most suitable candidates from their families in the case of our chosen time-series.

The results of the ets / arima comparison showed that the holt damped linear trend was the best candidate for ets and ARIMA(0, 1, 1) the best for the auto.arima function.

As a final thought, the fact that the Naive model outperformed the hw damped multiplicative as well as the ets (holt damped linear trend) model is based on the time series structure. The Training doesn't show a strong damped effect compared with the test series. This leads the models which include trend to overestimate the slope and therefore have a higher error than the models which consider a damped effect. This is also the reason why the naive model which only models a vertical line seems to outperform the more complex models. Nevertheless, the naive model would not be suitable for a more distant forecast and its explainability is not sufficient as well.

```
accDf <- rbind(
  accuracy(naive_shipment, df.Test),
  accuracy(df.hwMultiDamped, df.Test),
  accuracy(df.etsFor, df.Test),
  accuracy(df.autoarimaFor, df.Test)
)

rownames(accDf) <- c("Naive - Train", "Naive - Test",
  "hw damped Multiplicative - Train", "hw damped Multiplicative - Test",
  "ets(A,A,N) - Train", "ets(A,A,N) - Test",
  "ARIMA(0,1,1) - Train", "ARIMA(0,1,1) - Test" )

print(accDf[,2:3])
```

##

RMSE

MAE

## Naive - Train	610.2065	487.2000
## Naive - Test	855.6901	765.0000
## hw damped Multiplicative - Train	377.8383	290.0058
## hw damped Multiplicative - Test	1147.3688	988.3026
## ets(A,A,N) - Train	402.5582	316.9789
## ets(A,A,N) - Test	1231.0338	1047.5889
## ARIMA(0,1,1) - Train	421.8673	325.1873
## ARIMA(0,1,1) - Test	798.6903	661.7872

Conclusion

After having fitted and compared three different types of model families in the section before, this section summarizes and concludes the most important insights gained from this experiment on analysis of time series data.

As the time series consists of characteristics which are rather hard to be modeled. The damped effect in the test-set which is not clearly visible in the training makes it harder for the model to forecast. There is no specific model which fits the underlying complexity of the chosen time series specifically well.

In general, simple models are unable to provide reliable far forecasts for a time series with such characteristics. This can be seen in the high values of the indicators (MAE and RMSE). Exponential smoothing methods such as Holt-Winter's multiplicative method, on the other hand, have better performance indicators in the training data because they account for trend and seasonality simultaneously. However, even these methods did not perform sufficiently in the test data. This can be justified by the stagnating trend in the training set in the given time series. The fact that the naive models performance is better than the Exponential smoothing methods can be seen as a result of a probabilistic effect.

The arima approach ended up showing the most promising indicators for the test set. The autoregressive behavior lead to the stabilizing trend in the training data which decreased the error and linear-like forecast was sufficient enough to outperform all the other models in this case.