Individual Project:
Demand Forecasting for a Fast-Food Restaurant Chain

–

BS1808: Logistics and Supply Chain Analytics

Cyrill Studer
CID 01456269
CS5817@imperial.ac.uk

February 08, 2019

Table of Contents

1 Data Manipulation

1.1 PSQL Database Creation

The project was started off by loading all the received data into a PostgresSQL database. Therefore, PSQL was started in the terminal and a database called 'logistics' with the command 'CREATE DATABASE logistics;' was created. Next, tables for all 11 CSV-files were created and loaded into the newly established database with the following terminal commands:

TABLE 1 (ingredients):

```
CREATE TABLE ingredients (
IngredientName text,
IngredientShortDescription text,
IngredientId integer,
PortionUOMTypeId integer
);

\copy ingredients FROM /Users/Cyrill/Desktop/logistics/ingredients.csv WITH CSV
HEADER;
```

TABLE 2 (menuitem):

```
CREATE TABLE menuitem (
MD5KEY_MENUITEM text,
MD5KEY_ORDERSALE text,
CategoryDescription text,
DepartmentDescription text,
Description text, StoreNumber integer,
TaxInclusiveAmount integer,
TaxAmount float, AdjustedPrice float,
DiscountAmount float,
Price float,
Quantity integer,
PLU integer,
Id integer,
date text
);

\copy menuitem FROM /Users/Cyrill/Desktop/logistics/menuitem.csv WITH CSV
HEADER;
```

TABLE 3 (menu_items):

```
CREATE TABLE menu_items (
MenuItemName text,
MenuItemDescription text,
PLU integer,
MenuItemId integer,
RecipeId integer
);

\copy menu_items FROM /Users/Cyrill/Desktop/logistics/menu_items.csv WITH CSV HEADER;
```

TABLE 4 (portion_uom_types):

```
CREATE TABLE portion_uom_types (
PortionTypeDescription text,
PortionUOMTypeId integer
);

\copy portion_uom_types FROM /Users/Cyrill/Desktop/logistics/portion_uom_types.csv WITH CSV HEADER;
```

TABLE 5 (pos_ordersale)

```
CREATE TABLE pos_ordersale (
MD5KEY_ORDERSALE text,
ChangeReceived float,
OrderNumber integer,
TaxInclusiveAmount integer,
TaxAmount float,
MealLocation integer,
TransactionId integer,
StoreNumber integer,
date text
);

\copy pos_ordersale FROM /Users/Cyrill/Desktop/logistics/pos_ordersale.csv WITH CSV HEADER;
```

TABLE 6 (recipes):

```
CREATE TABLE recipes (
RecipeName text,
RecipeDescription text,
RecipeId integer
);
```

```
\copy recipes FROM /Users/Cyrill/Desktop/logistics/recipes.csv WITH CSV
HEADER;
```

TABLE 7 (recipe_ingredient_assignments):

```
CREATE TABLE recipe_ingredient_assignments (
RecipeId integer,
IngredientId integer,
Quantity integer
);

\copy recipe_ingredient_assignments FROM
/Users/Cyrill/Desktop/logistics/recipe_ingredient_assignments.csv WITH CSV
HEADER;
```

TABLE 8 (recipe_sub_recipe_assignments):

```
CREATE TABLE recipe_sub_recipe_assignments (
RecipeId integer,
SubRecipeId integer,
Factor integer
);

\copy recipe_sub_recipe_assignments FROM
/Users/Cyrill/Desktop/logistics/recipe_sub_recipe_assignments.csv WITH CSV
HEADER;
```

TABLE 9 (store_restaurant):

```
CREATE TABLE store_restaurant (
STORE_ADDRESS1 text,
STORE_ADDRESS2 text,
DISTRIBUTION_REGION text,
STORE_STATE text,
STORE_CITY text,
STORE_ZIP integer,
STORE_TYPE text,
STORE_LOYALTY_FLAG text,
STORE_NUMBER integer
);

\copy store_restaurant FROM
/Users/Cyrill/Desktop/logistics/store_restaurant.csv WITH CSV HEADER;
```

TABLE 10 (sub_recipes):

```
CREATE TABLE sub_recipes (
SubRecipeName text,
```

```
SubRecipeDescription text,
SubRecipeId integer
);

\copy sub_recipes FROM /Users/Cyrill/Desktop/logistics/sub_recipes.csv WITH CSV
HEADER;
```

TABLE 11 (sub_recipes):

```
CREATE TABLE sub_recipe_ingr_assignments (
SubRecipeId integer,
IngredientId integer,
Quantity integer
);

\copy sub_recipes FROM /Users/Cyrill/Desktop/logistics/sub_recipes.csv WITH CSV
HEADER;
```

1.2 Subresults

As a next step, five tables with sub-results were produced. The first sub-result table provides the number of sold quantities, per menu item, per restaurant, per day. This was produced by the below-stated code:

```
SELECT menuitem.id, menuitem.storenumber, pos_ordersale.date,
sum(menuitem.quantity) as qunatity
FROM menuitem
INNER JOIN pos_ordersale
ON pos_ordersale.MD5KEY_ORDERSALE = menuitem.MD5KEY_ORDERSALE
GROUP BY menuitem.storenumber, menuitem.id, pos_ordersale.date
ORDER BY pos_ordersale.date
```

Additionally, for each row the *recipeID* was added by storing the results of the above-stated code as a view *V1* and left outer joining it with the menu items table as outlined below:

```
SELECT id, storenumber, date, quantity_order, recipeid
FROM V1
LEFT OUTER JOIN menu_items
ON A1.id = menu_items.menuitemid
```

The final table with the described sub-results can be found in the 'R1' file of the 'Excel' appendix. The second sub-result table returns the required quantities of lettuce for each recipe. This table, 'R2' in the 'Excel' appendix, was produced by the following query:

```
SELECT recipeid, ingredients.ingredientid, quantity_recipe
FROM ingredients
INNER JOIN recipe_ingredient_assignments
ON ingredients.ingredientid = recipe_ingredient_assignments.ingredientid
WHERE ingredients.ingredientid = 27 or ingredients.ingredientid = 291
```

The third sub-result table returns the required quantities of lettuce required by the sub-recipes and the factor of each sub-recipe. This table, 'R3' in the 'Excel' appendix, was produced by the following query:

```
SELECT recipes.recipeid, ingredients.ingredientid, quantity_recipe, factor
FROM ingredients
INNER JOIN sub_recipe_ingr_assignments
ON ingredients.ingredientid = sub_recipe_ingr_assignments.ingredientid
INNER JOIN sub_recipes
ON sub_recipe_ingr_assignments.SubRecipeid = sub_recipes.SubRecipeid
INNER JOIN recipe_sub_recipe_assignments
ON sub_recipes.SubRecipeid = recipe_sub_recipe_assignments.SubRecipeid
INNER JOIN recipes
ON recipe_sub_recipe_assignments.recipeid = recipes.recipeid
WHERE ingredients.ingredientid = 27 or ingredients.ingredientid = 291
```

The fourth and the fifth sub-result tables 'R4' and 'R5' are connecting the work of the first three sub-result tables. By inner joining the results of 'R1' and 'R2' it gets ensured that all the relevant information regarding the required amount of lettuce from the main recipes for the orders made between the 13/03/2015 to the 15/06/2015 are in one place. This was done by the following code:

```
SELECT * from R1
INNER JOIN R2
ON R1.recipeid =R2.recipeid
```

Subsequently, within the resulting table, a new column *lettuce* was created to show the total amount of required lettuce by the main recipes per menu item, per restaurant, per day. This was done by multiplying the *quantity_order* column with the *quantity_recipe* column. Finally, a group by function was applied to get the total sum of required *lettuce* from the main recipes by store and day as outlined below:

SELECT storenumber, date, sum(lettuce) from V2
GROUP BY date, storenumber
ORDER BY storenumber, date

The final results of this sequence of queries and operations can be found in the 'R4' table of the 'Excel' appendix.

A very similar procedure was applied to get all the required information for the lettuce demanded by the sub-recipes of each recipe for the orders made between the 13/03/2015 - 15/06/2015. As a first step, the tables 'R1' and 'R3' were joined by the following query:

*SELECT * from R1*
INNER JOIN R3
ON R1.recipeid =R3.recipeid

Next, similar to the procedure for 'R4', a new column *lettuce* was created to show the total amount of required lettuce from the sub-recipes per menu item, per restaurant, per day. This was done by multiplying the *quantity_order* column with the *quantity_recipe* column and the *factor* column. Finally, a group by function was applied to get the total sum of required *lettuce* from the sub-recipes by store and day as outlined below:

SELECT storenumber, date, sum(lettuce) from V3
GROUP BY date, storenumber
ORDER BY storenumber, date

The final results of this sequence of queries and operations can be found in the 'R5' table of the 'Excel' appendix.

1.3 Relevant Time Series Data Creation

Finally, in order to get the relevant and cleaned time series data to do the lettuce forecast per store, the required amount of lettuce from the main recipes needs to be added to the required amount of lettuce from the sub-recipes. Therefore, the tables 'R4' and 'R5' need to be joined with the following query:

```
SELECT * FROM R4
LEFT JOIN R5
ON B3.storenumber_S = B4.storenumber_S AND B3.date_S = B4.date_S;
```

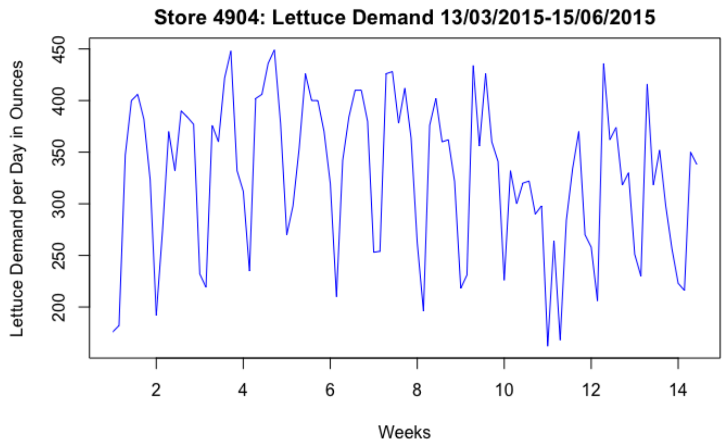
The resulting table was then exported to Excel where a new column *total lettuce* was created which adds up the required amount of lettuce from the recipes and the sub-recipes. For the time series analysis, only the columns *storenumber, date* and *total lettuce* are of interest and therefore all other columns were deleted. Finally, the table was filtered by the storenumber and the times series data for each store number was saved separately and can be found in the 'Excel' appendix under the names '4904', '12631', '20974' and '46673'.

2 Store 4904

2.1 Data Description

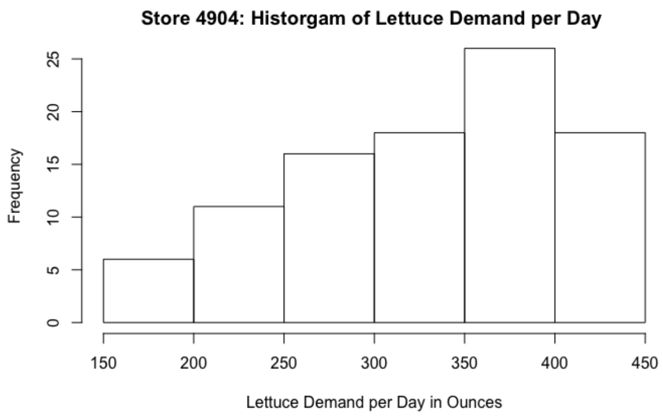
The first step of the analysis for store 4904 is to simply inspect the plot of the time series as outlined in *Illustration 1* below. The time series was created and plotted with a frequency of seven in order to convert the daily data into a weekly time frame. From a purely visual inspection point, one can assume that the times series is more or less trend-stationary and one can observe a clear seasonal pattern which occurs approximately in weekly cycles.

Illustration 1: Store 4904 Time Series Plot



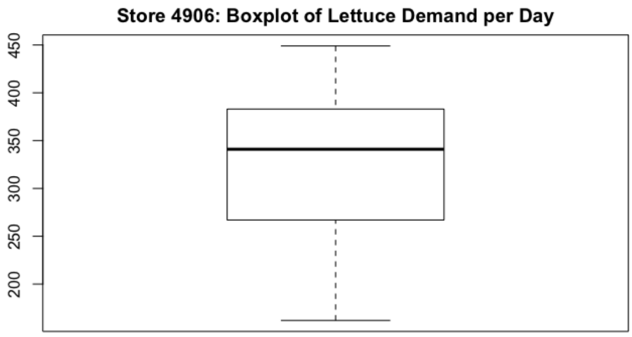
Further, with the inspection of the histogram of the time series data in *Illustration 2*, it becomes visible that the distribution of the time series data is slightly left-skewed.

Illustration 2: Store 4904 Time Series Histogram



Moreover, the inspectation of the boxplot in *Illustration 3* does not provide evidence for any significant outliers.

Illustration 3: Store 4904 Time Series Boxplot



Lastly, it is always helpful to check and to bear in mind some basic summary statistics as outlined below in *Illustration 4*.

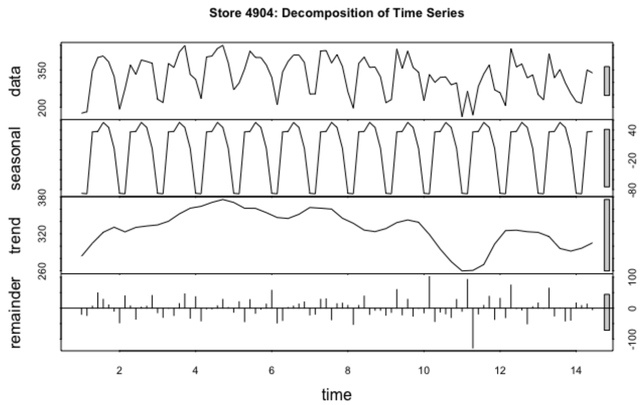
Illustration 4: Store 4904 Summary Statistics

Measurement	**Result**
Mean	327.9
Median	341
Standad Deviation	75.32

2.2 Decomposition of the Time Series

After descriptively exploring the original time series, the next step is to decomposite the time series into a seasonal, trend and remainder component as outlined in *Illustration 5* on the next page.

Illustration 5: Store 4904 Time Series Decomposition



By looking at the above illustration two key observations are apparent. First of all, the times series seems to have an additive seasonal component that repeats itself in approximately weekly cycles. Secondly, there does not appear to be any particular overall trend over the cause of the observed time period. These two observations indicate that for any good modelling practice a seasonal component needs to be included while the trend component can most likely be disregarded.

2.3 Holt-Winters

As a next step, a first modelling approach in form of the Holt-Winters model will be apply. The Holt-Winters method uses exponential smoothing to model a time series while accounting for an average component (alpha), a trend/slope component (beta) and a seasonal component (gamma). By the nature of the task, beta is set to 0. By applying the Holt-Winters function to the times series of store 4904 the following parameters result:

Alpha = 0.14

Beta = 0

Gamma = 0.22

a = 303.65

l = (48.43, 26.85, -6.69, -83.67, -79.10, 42.80, 26.72)

The above results confirm the before made assumptions. The resulting gamma = 0.22 confirms that there is considerable seasonal component observable. Overall, the Holt-Winters model results in an in-sample root mean squared error of 45.88.

Next, by applying the newer and often times better-performing ETS function of the Holt-Winters method the results are slightly different:

Alpha = 0.17

Beta = 0

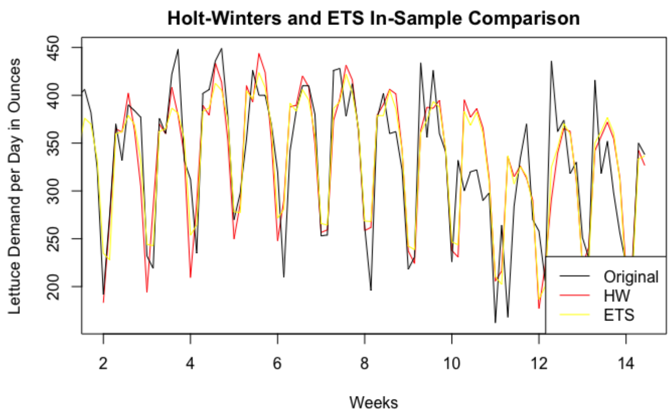
Gamma = 0.0001

a = 326.76

l = (6.77, 45.73, 57.20, 34.78, 34.50, -89.15, -89.13)

With the slightly different parameters, the resulting model proves to be an additive ETS (A,N,A) model. The ETS model reduces the root mean squared error to 42.19 compared to 45.88 of the original Holt-Winters modelling technique. *Illustration 6* shows the modelled results of the ETS(A,N,A) and the original Holt-Winters modelling technique compared to the original times series. One can observe, that the original Holt-Winters modelling technique and ETS model behave very similar and are hardly distinguishable from a purely visual point of view.

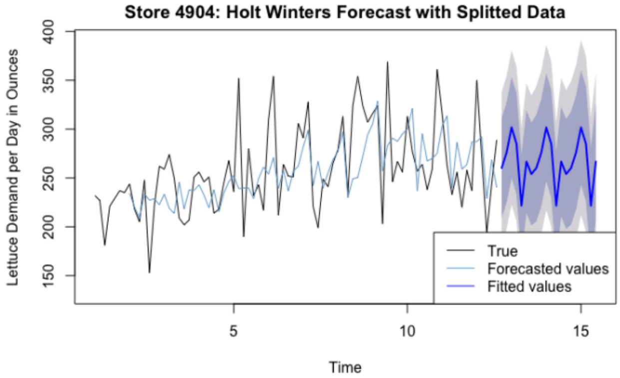
Illustration 6: Store 4904 Holt Winters and ETS In-Sample Comparison



However, in order to comprehensively and meaningfully assess the accuracy and precision of both introduced models, there is a need to test the models forecasting power. A way to do this is to split the data into a training data set and into a validation data set in order to get an out-of-sample accuracy/ error. Therefore, the 95 observations of the original time series get split into a training data set of 76 observations (80% of the original data) and into a validation data set with 19 observations (20% of the original data).

Illustration 7 below shows the forecast, the in-sample-error and the out-of-error for the splitted data set for the original Holt-Winters modelling technique.

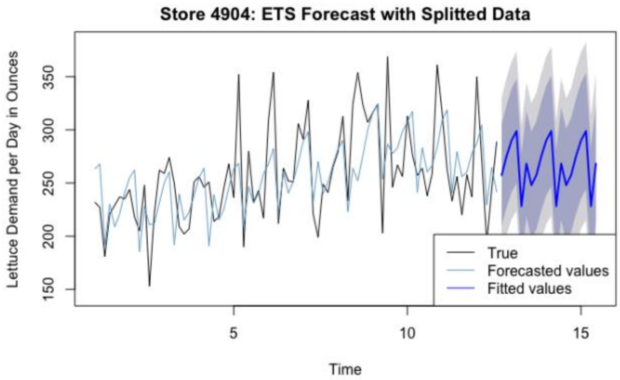
Illustration 7: Store 4904 Results Holt-Winters Forecast Splitted Data



Error Type	Result
RMSE In-Sample	45.58
RMSE Out-Of-Sample	79.38

Illustration 8 below shows the forecast, the in-sample-error and the out-of-sample-error for the splitted data set for the ETS (A,N,A) modelling technique.

Illustration 8: Store 4904 Results ETS (A,N,A) Forecast Splitted Data

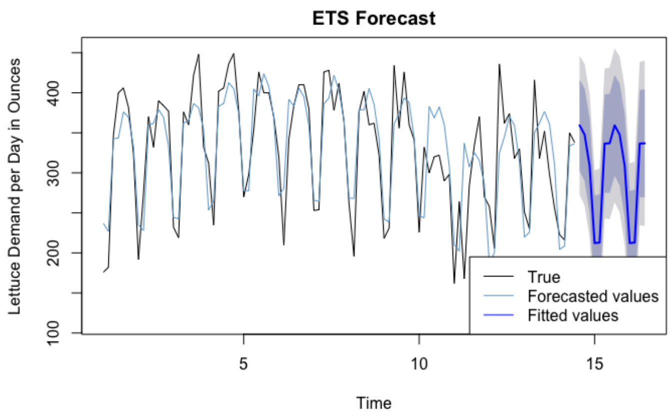


Error Type	Result
RMSE In-Sample	41.26
RMSE Out-Of-Sample	55.96

Again, from a purely visual perspective, it is hard to see any clear difference between the two forecasting modelling techniques. By consulting the out-of-sample error, one can see that the ETS model has clearly a lower RMSE. Moreover, the ETS (A,N,A) model has also a slightly lower in-sample error compared to the original Holt-Winters modelling technique. Based on that, the choice is to proceed further with ETS (A,N,A) model and to compare this model to the best performing ARIMA model.

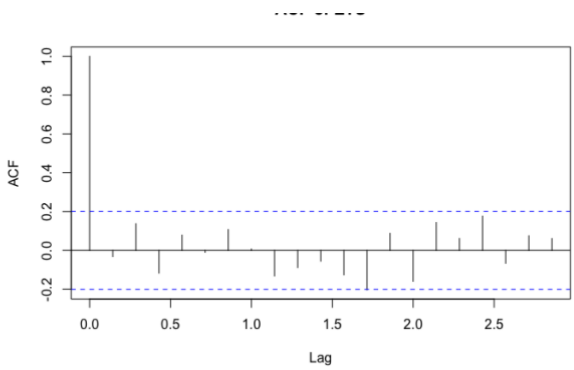
As a next step, ETS (A,N,A) is getting retrained with the entire data set (95 observations) and a 14-day forecast is made. The resulting forecast is illustrated in *Illustration 9*.

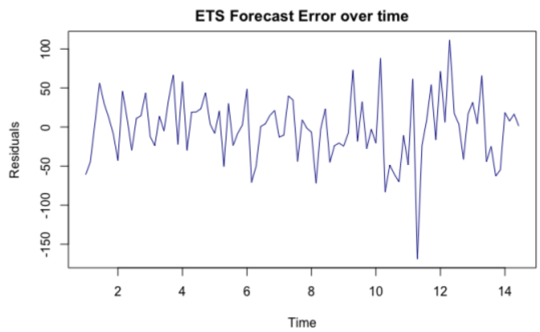
Illustration 9: Store 4904 14-Days Lettuce ETC Forecast



As a last step, there is a need to examine the resulting residuals to assess the quality of the model. More precisely, there needs to be checked whether the error term behaves heteroscedastic and whether there is a correlation between the modelled error terms and the forecasted values. For these reasons, *Illustration 10* shows the residuals over time as well as the autocorrelation of the error term.

Illustration 10: Store 4904 Residual Analysis ETS (A,N,A)





By examining the autocorrelation function of the residuals on the left-hand side of *Illustration 10,* one can see that at none of the 20 examined lags a significant correlation can be observed. Moreover, by examining the residuals over time on the right-hand side of *Illustration 10*, one can see that except an outlier around week 11, the residuals are more or less homogenously distributed (no trend in variance) and are distributed around the zero-mean. To confirm these results quantitatively the Ljung-Box, the Argumented Dickey-Fuller and the Box-Pierce test were run. As shown in *Illustration 11* there is little evidence that the errors are non-stationary or correlated/ dependent.

Illustration 11: Store 4909 Quantitative Residual Analysis ETS Model

Test	Null Hypothesis	p-value	Result (5% Significance)
Box-Ljung	Indendetly Distributed Residuals	0.150	Independent Residuals
Argumented Dickey-Fuller	Non-Trend-Stationary Residuals	0.019	Trend-Stationary Residuals
Box-Pierce	Zero Autocorrelated Residuals	0.21	Zero Autocorrelated Residuals

2.4 ARIMA

To start the construction of the ARIMA model, certain tests to check whether there is a need to preprocess the original data in order to make the data fit for the ARIMA model need to be run. More precisely, it is of interest to check whether the original data is stationary, independently distributed, non-correlated and has a constant variance. Therefore, the tests as outlined in *Illustration 12* were conducted.

Illustration 12: Quantitative Time Series Analysis

Test	Null Hypothesis	p-value	Result (5% Significance)
Box-Ljung	Indendetly Distributed Data	0.0001684	(H1) Dependent Data
Argumented Dickey-Fuller	Existence of Unit Root	0.01	(H1) Non-Existence of Unit Root
Kwiatkowski-Philips-Schmidt-Shin	Trend Stationarity	0.07518	(H0) Trend Stationary Data
Box-Pierce	zero autocorrelated data	0.0002125	(H1) Autocorrelated Data
Philips-Perrron	Integrated of order 1	0.01	(H1) Not Itegrated of Order 1

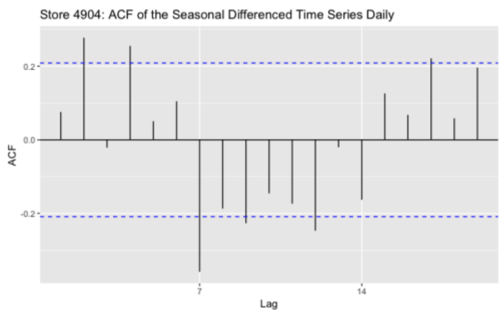
Having a look at the above test results, it can be concluded that the original time series is trend-stationary in variance and mean but also dependent/ autocorrelated. In addition to the above tests, the built-in R-Functions ‘ndiffs’ and ‘nsdiffs’ which help to determine how many order differences, respectively how many seasonal differences, one has to take in order to make the time series stationary were run as one can observe in *Illustration 13*.

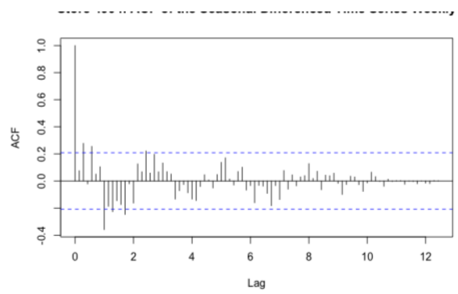
Illustration 13: Store 4904 Output R Preprocessing Functions

R-Function	**Result**
ndiff	0
nsdiff	1

The result of 'ndiff' in *Illustration 13* confirms that the times series is trend-stationary and does not need to be first-ordered differenced. However, the result of 'nsdiff' indicates that the first order difference of the seasonal component, which is equal to seven days due to the weekly time intervals, needs to be taken in order to make the time series both trend- and seasonal-stationary. Therefore, it can be derived from the so far collected information that possible ARIMA models need to be of the form ARIMA (p,0,q)(P,1,Q)(7). As a next step, the ACF and the PACF of the time series need to be inspected.

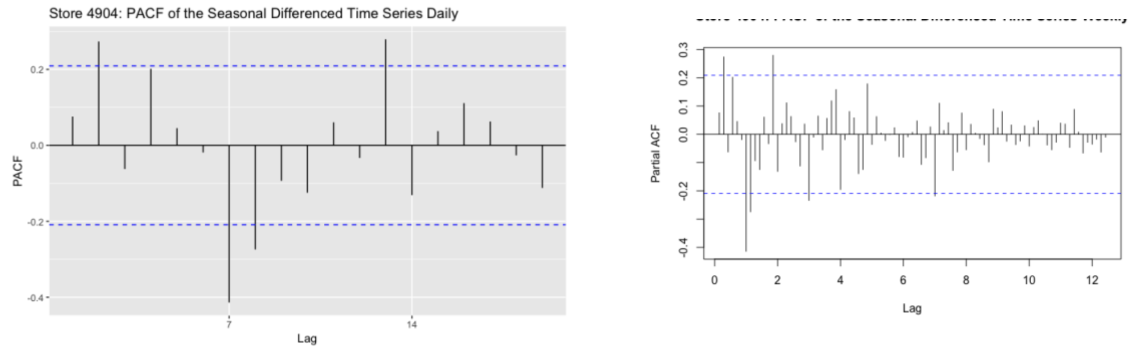
Illustration 14: Store 4904 Autocorrelation Function on Daily and Weekly Basis





From the autocorrelation function on a daily basis in *Illustration 14*, one can see that there are different statistically significant lags at a significance level of 5% for the first two weeks which can be an indication for a seasonal MA. Also, there are some significant lags for the very first days of the daily chart which indicates that maybe a p or q between 0 and 2 seems reasonable. Additional, only a slightly declining trend can be observed in the weekly based plot on the right-hand side of *Illustration 14*.

Illustration 15: Store 4904 Partial Autocorrelation Function on Daily and Weekly Basis



The fact that the PACF in *Illustration 15* has not any clearly recognizable pattern further emphasizes that there is little evidence to include a high seasonal AR and MA (>2). However, by having a closer look at the weekly PACF, one can clearly see that there are at least 1-2 statistical relevant seasonal spikes. For these reasons, it seems reasonable to test whether 1 or 2 for P or Q can boost the out-of-sample accuracy.

R has the built-in function “autoarima“ which automatically tries to fit the best parameters for the ARIMA modelling. The two best performing models according to autoarima are ARIMA (1,0,1)(0,1,1)(7) and ARIMA (2,0,0)(0,1,1)(7). Additionally, multiple other models based on the before made findings derived from the ACF and PACF plot were tried out. The best performing model based on these findings is ARIMA (1,0,3)(0,1,1)(7). As a next, step the goal is to find out which of these models performs best.

Again, as with the Holt-Winter methods, in order to critically and meaningfully assess the accuracy and precision of each model there is a need to test the models’ forecasting power. A way to do this is to split the data into a training data set and into a validation data set in order to get an out-of-sample accuracy/ error of each model. Therefore, again the splitted data is used to asses the performance of the models. *Illustration 16* sums up the in-sample and out-of-sample performance of each model.

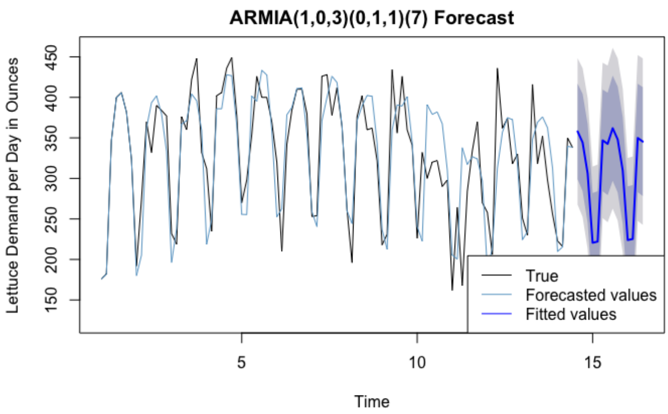
Illustration 16: Store 4904 Results ARIMA Forecast Splitted Data

	ARIMA (1,0,1)(0,1,1)(7)	**ARIMA (2,0,0)(0,1,1)(7)**	**ARIMA (1,0,3)(0,1,1)(7)**
RMSE In-Sample	44.06	45.31	42.08
RMSE Out-Of-Sample	75.64	75.29	58.82

Based on the above results, the model ARIMA (1,0,3)(0,1,1)(7) performs best. Therefore, the decision is to proceed with this model.

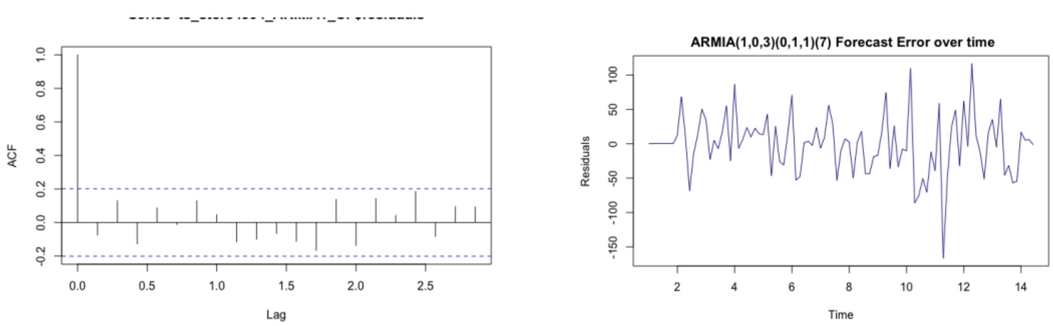
As a next step, the ARIMA (1,0,3) (0,1,1)(7) model is getting trained with the entire time series data and a forecast for the next 14 days was made. The forecast is displayed in *Illustration 17:*

Illustration 17: Store 4904 14-Days Lettuce Forecast with ARIMA (1,0,3)(0,1,1)(7)



As a last step, there is a need to again examine the resulting residuals to assess the quality of the ARIMA(1,0,3) (0,1,1)(7) model. More precisely, there is a need to check whether the error terms are heteroscedastic and whether there is correlation between the modelled error terms and the forecasted values. For these reasons, *Illustration 18* shows the residuals over time as well as the autocorrelation of the error term.

Illustration 18: Store 4904 Residual Analysis ARIMA (1,0,3)(0,1,1)(7)



By examine the autocorrelation function of the residuals on the left-hand side of *Illustration 18,* one can see that at none of the 20 examined lags a significant correlation can be observed. Moreover, by examining the residuals over time on the right-hand side of *Illustration 18,* one can see that except an outlier around week 11 the residuals are more or less homogenously distributed (no trend in variance) and are distributed around the zero mean. To confirm these results quantitatively the Ljung-Box, the Argumented Dickey-Fuller and the Box-Pierce tests were run. As shown in *Illustration 19* there is little evidence that the errors are non-stationary or correlated/ dependent. Therefore, the forecast with ARIMA(1,0,3) (0,1,1)(7) seems valid.

Illustration 19: Store 4909 Quantitative Residual Analysis ARIMA

Test	Null Hypothesis	p-value	Result (5% Significance)
Box-Ljung	Indendetly Distributed Residuals	0.3311	Independent Residuals
Argumented Dickey-Fuller	Non-Trend-Stationary Residuals	0.01	Trend-Stationary Residuals
Box-Pierce	Zero Autocorrelated Residuals	0.8911	Zero Autocorrelated Residuals

2.5 Conclusion

To conclude the analysis of the 14-days lettuce forecast for store 4904, a decision needs to be made whether to use the ETS (A,N,A) model or the ARIMA (1,0,3)(0,1,1)(7) for final forecast. As outlined in the previous analysis, both models meet the standards to do meaningful and correct forecasting. It comes now down to decide which model will likely be more precise. *Illustration 20* shows the in-sample performance on the splitted series data set and the out-of-sample performance for the splitted data of the two models.

Illustration 20: Performance Comparison ETS (A,N,A) and ARIMA (1,0,3)(0,1,1)(7)

	ETS (A,N,A)	**ARIMA (1,0,3)(0,1,1)(7)**
RMSE In-Sample (Splitted Data)	41.26	42.08
RMSE Out-Of-Sample (Splitted Data)	55.96	58.82

As one can see, the ETS model appears to perform better. Based on that, the final forecast is done with the ETS model and outlined in *Illustration 21*:

Illustration 21: Final Lettuce Forecast Lettuce for Store 4909 (in ounces)

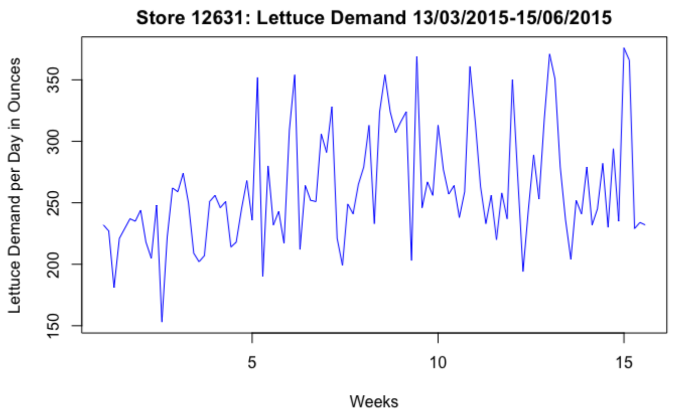
Day	Forecast	Low (80% Confidence)	High (80% Confidence)
1	359	302	416
2	348	290	406
3	309	250	367
4	212	153	272
5	213	153	273
6	337	275	398
7	337	275	399
8	359	297	422
9	348	284	411
10	309	244	373
11	212	147	277
12	213	147	279
13	337	270	403
14	337	269	404

3 Store 12631

3.1 Data Description

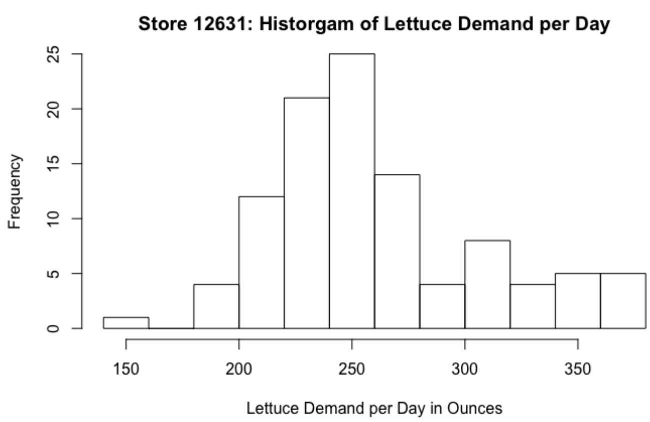
The first step of the analysis for store 12631 is to simply inspect the plot of the time series as outlined in *Illustration 22* below. The time series was created and plotted with a frequency of seven in order to convert the daily data into a weekly time frame. From a purely visual inspection point, one can assume that the times series is more or less trend-stationary and one can observe a clear seasonal pattern which occurs approximately in weekly cycles.

Illustration 22: Store 12631 Time Series Plot



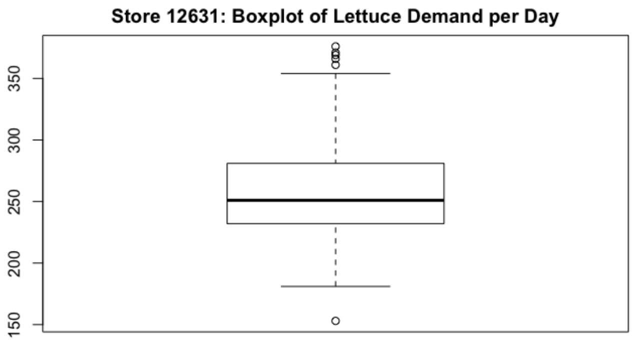
Further, with additionally inspecting the histogram of the time series in *Illustration 2*3, it becomes visible that the distribution of the time series data is approximately normally distributed with a slightly larger right-handed tail.

Illustration 23: Store 12631 Time Series Histogram



Having a view at the boxplot in *Illustration 24* confirms that the time series for store 12631 has a number of outliers, especially on the right-hand side of the distribution. Bearing this observation in mind, it is possible that for the ARIMA modelling part the times series data needs to be transformed (e.g. with the logarithm to make the data stationary).

Illustration 24: Store 12631 Time Series Boxplot



Lastly, it is always helpful to check and to bear in mind some basic summary statistics as outlined in *Illustration 25*.

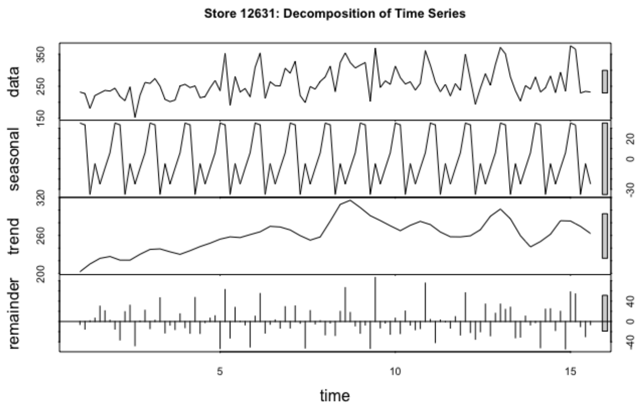
Illustration 25: Store 12631 Summary Statistics

Measurement	Result
Mean	261.3
Median	251
Standad Deviation	47.19

3.2 Decomposition of the Time Series

After descriptively exploring the original time series, the next step is to decomposite the time series into a seasonal, trend and remainder component as outlined in *Illustration 26* on the next page.

Illustration 26: Store 12631 Time Series Decomposition



By looking at the above graph two key observations are apparent. First of all, the times series seems to have a strong seasonal component that repeats itself in approximately weekly cycles. Secondly, the time series for store 12631 seems to have a slightly increasing overall trend over the cause of the observed time period which could potentially be caused by the form of seasonality. These two observations indicate that for any good modelling practice a seasonal component needs to be accounted for and there potentially needs to be controlled for a trend component. Both observations together could be an indication that the seasonality is multiplicative rather than a additive.

3.3 Holt-Winters

As a next step, a first modelling approach in form of the Holt-Winters model will be applied. The Holt-Winters Method uses exponential smoothing to model a time series while accounting for an average component (alpha), a trend/slope component (beta) and a seasonal component (gamma). By the nature of the task, beta is set to 0. However, given the observation from the time series plot and the decomposition, the model will include a multiplicative seasonality with no trend. By applying the Holt-Winters function to the times series for store 12631 the following parameter result:

Alpha = 0.09

Beta = 0

Gamma = 0.17

a = 262.14

l = (1.02, 1.04, 1.22, 1.15, 0.90, 0.99, 0.92)

The above results confirm the before made assumptions. The resulting gamma = 0.17 confirms that there is considerable seasonal component observable. Overall, the Holt-Winters model results an in-sample root mean squared error of 39.94.

Next, by applying the newer and often times better-performing ETS function of the Holt-Winters method the results are slightly different:

Alpha = 0.11

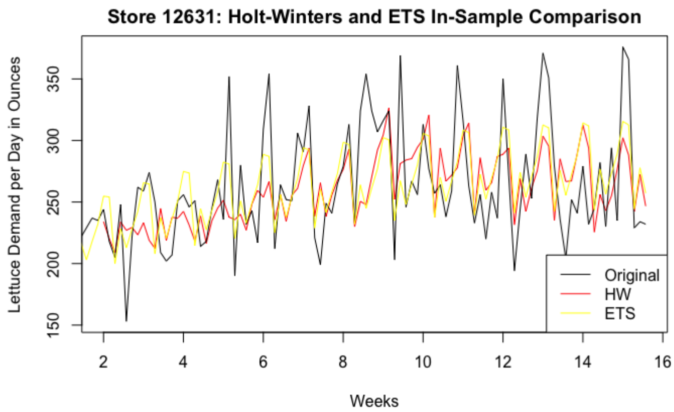
Gamma = 0.0001

l = 240.15

s = (1.02, 0.96, 0.91, 0.99, 0.86, 1.14, 1.12)

The ETS model reduces the root mean squared error to 39.94 compared to 36.76 of the original Holt-Winters modelling technique. *Illustration 27* shows the modelled results of the ETS and the original Holt-Winters modelling technique compared to the original times series. One can observe, that the original Holt-Winters modelling technique and ETS model behave very similar and are hardly distinguishable from a purely visual point of view.

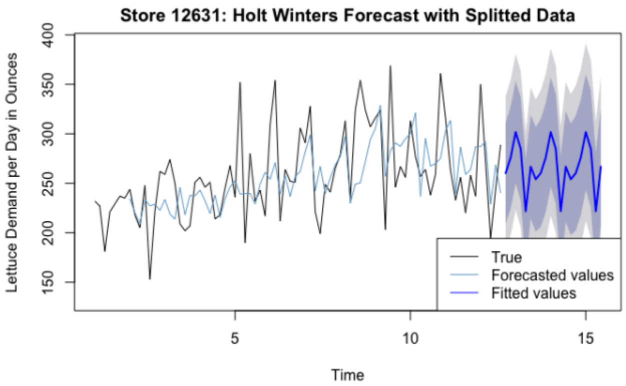
Illustration 27: Store 12631 Holt-Winters and ETS In-Sample Comparison



However, in order to comprehensively and meaningfully assess the accuracy and precision of both introduced models there is a need to test the models, forecasting power. A way to do this is to split the data into a training data set and into a validation data set in order to get an out-of-sample accuracy/ error. Therefore, the 103 observations of the original time series get split into a training data set of 83 observations (80% of the original data) and into a validation data set with 20 observations (20% of the original data).

Illustration 28 below shows the forecast, in-sample-error and out-of-sample-error for the splitted data set for the original Holt-Winters modelling technique.

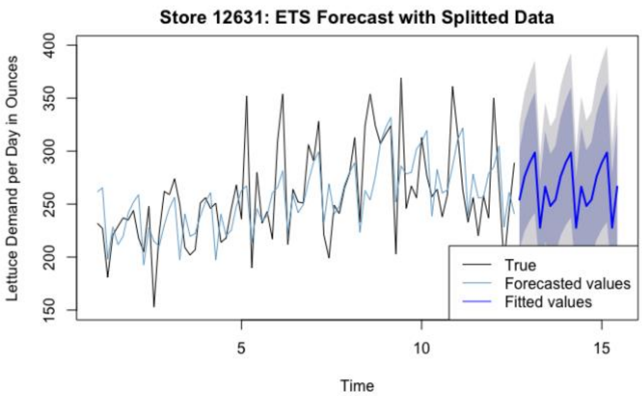
Illustration 28: Store 12631 Results Holt Winters Forecast Splitted Data



Error Type	Result
RMSE In-Sample	39.68
RMSE Out-Of-Sample	44.57

Illustration 29 below shows the forecast, in-sample-error and out-of-sample-error for the splitted data set for the ETS modelling technique.

Illustration 29: Store 12631 Results ETS Forecast Splitted Data

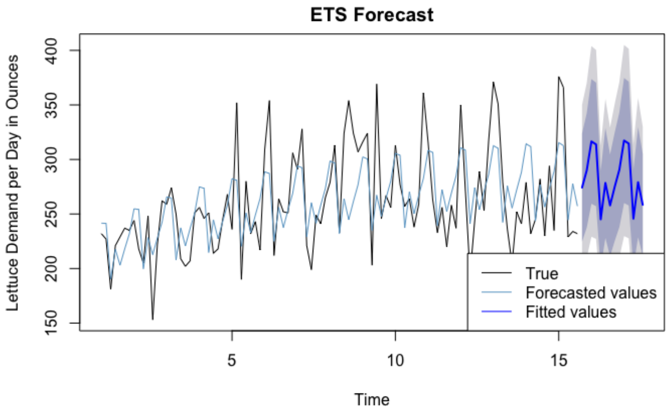


Error Type	Result
RMSE In-Sample	35.05
RMSE Out-Of-Sample	44.60

Again, from a purely visual perspective, it is hard to see any difference between the two forecasting modelling techniques. By consulting the out-of-sample error, one can see that the ETS model has clearly a lower RMSE. Moreover, the ETS model has also a lower in-sample-error compared to the original Holt-Winters modelling technique. Based on that, the choice is to proceed further with ETS model and to compare this model to the best performing ARIMA model.

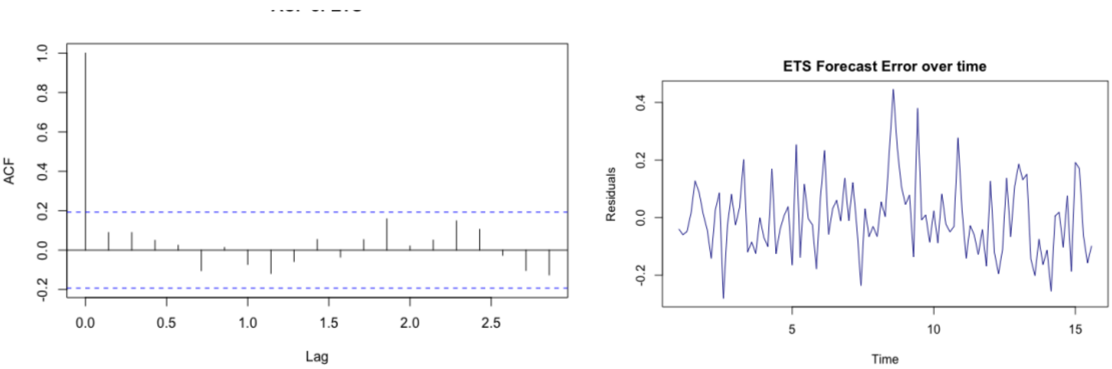
As a next step, ETS is getting retrained with the entire data set (94 observations) and a 14-day forecast is made. The resulting forecast is illustrated in *Illustration 30*.

Illustration 30: Store 4904 14 Days Lettuce ETC Forecast



As a last step, we need to examine the resulting residual to assess the quality of the model. More precisely, there is a need to check whether the error term behaves heteroscedastic over the observed period of time and whether there is a correlation between the modelled error terms and the forecasted values. For that reason, *Illustration 31* shows the residuals over time as well as the autocorrelation of the error term.

Illustration 31: Store 12631 Residual Analysis ETS Forecast



By examine the autocorrelation function of the residuals on the left-hand side of *Illustration 31*, one can see that at none of the 20 examined lags a significant correlation can be observed. Moreover, by examining the residual over time on the right-hand side of *Illustration 31*, one can see that except an outlier around week 9 the residual are more or less homogenously distributed (no trend in variance) and are distributed around the zero mean. To confirm this result quantitatively the Ljung-Box, the Argumented Dickey-Fuller and the Box-Pierce tests were run. As shown in *Illustration 32*, there is little evidence that the errors are non-stationary or correlated/ dependent.

Illustration 32: Store 12631 Quantitative Residual Analysis ETS Forecast

Test	Null Hypothesis	p-value	Result (5% Significance)
Box-Ljung	indendetly distributed residuals	0.6074	Independent Residuals
Argumented Dickey-Fuller	non-stationary residuals	0.01	Stationary Residuals
Box-Pierce	zero autocorrelated residuals	0.7438	Zero Autocorrelated Residuals

3.4 ARIMA

To start the construction of the ARIMA model, certain tests to check whether there is a need to preprocess the original data in order to make the data fit for the ARIMA model need to be run. More precisely, it is of interest to check whether the original data is stationary, independently distributed, non-correlated and has a constant variance. Therefore, the tests as outlined in *Illustration 33* were conducted.

Illustration 33: Store 12631 Quantitative Time Series Analysis

Test	Null Hypothesis	p-value	Result (5% Significance)
Box-Ljung	Indendetly Distributed Data	0.01039	(H1) Dependent Data
Argumented Dickey-Fuller	Existence of Unit Root	0.01	(H1) Non-Existence of Unit Root
Kwiatkowski-Philips-Schmidt-Shin	Trend Stationarity	0.01	(H1) No Trend Stationary Data
Box-Pierce	Zero Autocorrelated Data	0.01154	(H1) Autocorrelated Data
Philips-Perrron	Integrated of Order 1	0.01	(H1) Not Itegrated of Order 1

By having a look at the result of the Kwiatkowski-Philips-Schmidt-Shin test, it becomes obvious that the original time series is not trend-stationary (trend-stationary). Moreover, as already outlined, the seasonality is multiplicative rather than additive. Therefore, there is a need to transform the original data before continuing with the ARIMA modelling process. For these reasons, the logarithm and the first difference of the original times series was taken. Subsequently, all the above tests were run once more as outlined in *Illustration 34*.

Illustration 34: Store 12631 Quantitative Time Series Analysis (Logarithm and 1st Diff)

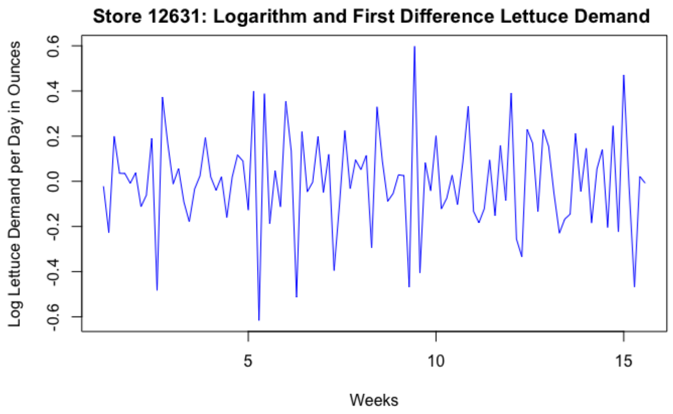
Test	Null Hypothesis	p-value	Result (5% Significance)
Box-Ljung	Indendetly Distributed Data	2.812e-06	(H1) Dependent Data
Argumented Dickey-Fuller	Existence of Unit Root	0.01	(H1) Non-Existence of Unit Root
Kwiatkowski-Philips-Schmidt-Shin	Trend Stationarity	0.1	(H0) Trend Stationary Data
Box-Pierce	Zero Autocorrelated Data	3.911e-06	(H1) Autocorrelated Data
Philips-Perrron	Integrated of Order 1	0.01	(H1) Not Itegrated of Order 1

Interpreting the above results and especially the Kwiatkowski-Philips-Schmidt-Shin test, it can be concluded that with taking the logarithm and the first order difference the times series now satisfies the requirements to fit an ARIMA model and the series was made stationary. For additional verification purposes, the built-in R-Functions 'ndiffs' and 'nsdiffs' were run which helps to determine how many additional differences, respectively how many seasonal differences one has to take in order to make the time series stationary.

R-Function	Result
ndiff	0
nsdiff	0

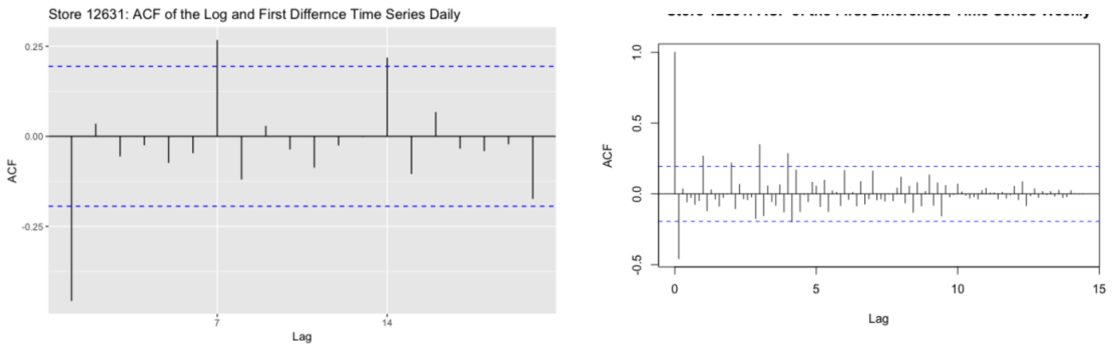
The results confirm the before made assumptions. The plot in *Illustration 35* below shows the transformed data.

Illustration 35: Store 12631 Logarithm and First Difference Time Series Plot



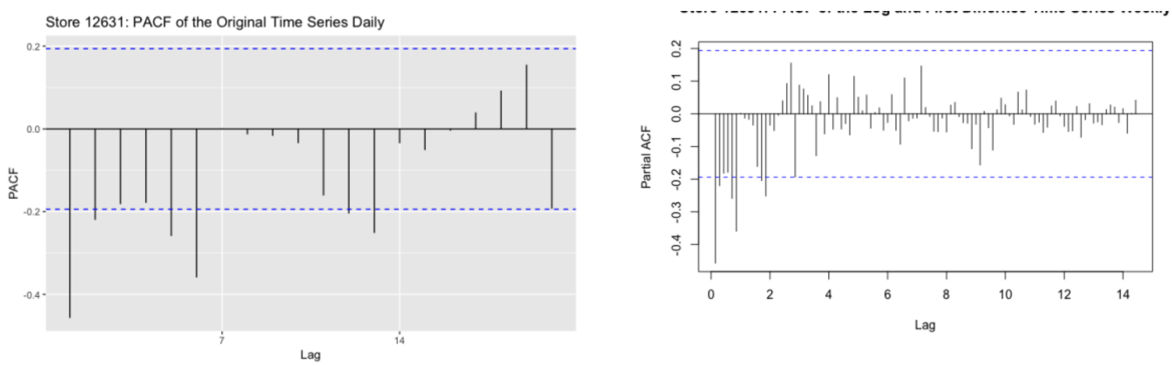
Now, the normal ARIMA modelling procedure can start. One can derive from the so far collected information that possible ARIMA models likely need to be of the form ARIMA (p,1,q)(P,0,Q)(7). As a next step the ACF and the PACF of the time series in *Illustration 36* need to be inspected.

Illustration 36: Store 12631 Autocorrelation Function on Daily and Weekly Basis



From the autocorrelation function on a daily basis one can see that there is likely only one statistically significant lag. Therefore, it is likely that p and q are 1 or 0. Moreover, a clear seasonal pattern on the basis of a 7-day interval (weekly basis) is observable. This is also confirmed by the Autocorrelation Function on a weekly basis on the right-hand side of *Illustration 36*.

Illustration 37: Store 12631 Partial Autocorrelation Function on Daily and Weekly Basis



For the Partial Autocorrelation Function on a daily basis, one in *Illustration 37* can see that there is not a systematically declining pattern observable. Moreover, one can observe significant lags only for approximately the first two weeks. Therefore, it potentially makes sense to set P and Q somewhere between 0 and 2.

R has the built-in function "auto ARIMA" which automatically tries to fit the best parameters. The two best performing models according to autoarima are ARIMA (0,1,1)(0,0,2)(7) and ARIMA (1,1,0)(0,0,2)(7). Additionally, multiple other models based on the before made findings derived from the ACF and PACF plot were tried out. The best performing model based on these findings is ARIMA (0,1,1)(1,0,2)(7). As a next, step the goal is to find out which of these models performs best.

Again, as with the Holt-Winter methods, in order to critically and meaningfully assess the accuracy and precision of each models there is a need to test the models' forecasting power. A way to do this is to split the data into a training data set and into a validation data set in order to get an out-of-sample accuracy/ error of each model. Therefore, we again use the splitted data to asses the performance of the models. *Illustration 38* sums up the in-sample and out-of-sample performance of each model.

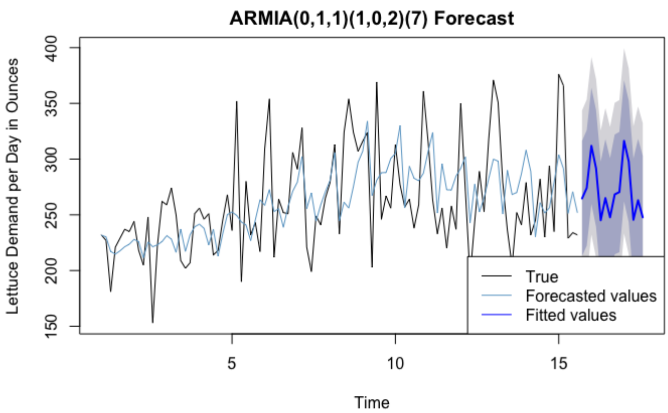
Illustration 38: Store 12631 Results ARIMA Forecast Splitted Data

	ARIMA (0,1,1)(0,0,2)(7)	**ARIMA (1,1,0)(0,0,2)(7)**	**ARIMA(0,1,1)(1,0,2)[7]**
RMSE In-Sample	40.03	45.65	37.65
RMSE Out-Of-Sample	51.48	53.43	44.73

Based on the above results, the model ARIMA (0,1,1)(1,0,2)(7) performs best. Therefore, the decision is to proceed with this model.

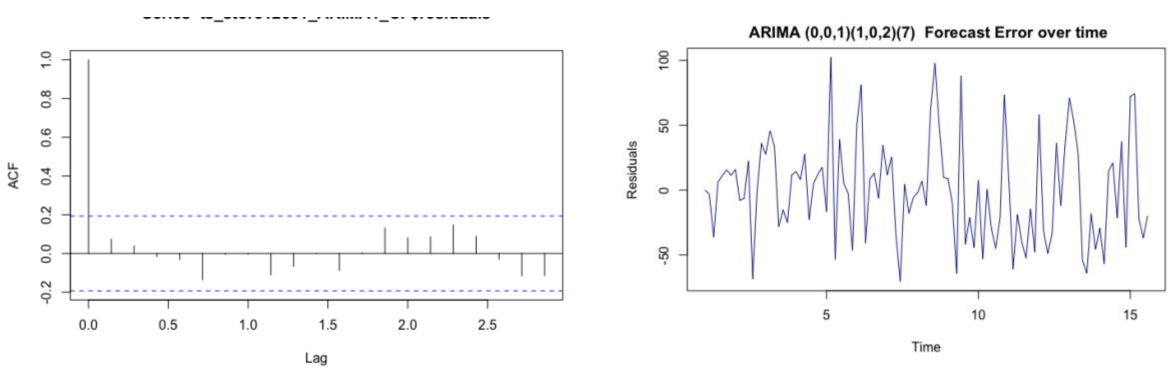
As a next step, the ARIMA (0,1,1)(1,0,2)(7) model is getting trained with the entire time series data and a forecast for the next 14 days was made. The forecast is displayed in *Illustration 39:*

Illustration 39: Store 4904 14 Days Lettuce Forecast with ARIMA (1,0,1)(0,1,1)(7)



As a last step, there is a need to again examine the resulting residuals to assess the quality of the ARIMA model. More precisely, there is a need to check whether the error term is heteroscedastic and whether there is correlation between the modelled error terms and the forecasted values. For these reasons, *Illustration 40* shows the residuals over time as well as the autocorrelation of the error term.

Illustration 40: Store 12631 ARIMA Residual Analysis



By examining the autocorrelation function of the residuals on the left-hand side of *Illustration 40*, one can see that at none of the 20 examined lags a significant correlation can be observed. Moreover, by examine the residual over time on the right-hand side of *Illustration 40*, one can see that the residuals are more or less homogenously distributed (no trend in variance) and are distributed around the zero mean. To confirm these results quantitatively the Ljung-Box, the Argumented Dickey-Fuller and the Box-Pierce tests were runned. As shown in *Illustration 41* there is little evidence that the errors are non-stationary or correlated/ dependent. Therefore, the forecast with ARIMA(0,0,1) (1,0,2)(7) seems valid.

Illustration 41: Quantitative Residual Analysis

Test	**Null Hypothesis**	**p-value**	**Result (5% Significance)**
Box-Ljung	Indendetly Distributed Residuals	0.6702	Independent Residuals
Argumented Dickey-Fuller	Non-Trend-Stationary Residuals	0.01	Trend-Stationary Residuals
Box-Pierce	Zero Autocorrelated Residuals	0.4582	Zero Autocorrelated Residuals

3.5 Conclusion

To conclude the analysis of the 14 days lettuce forecast for store 12631, a decision needs to be made whether to use the ETS model or the ARIMA (0,0,1)(1,0,2)(7) for final forecast. As outlined in the previous analysis, both models meet the standards to do meaningful and correct forecasting. It comes now down to decide which model will likely be more precise. *Illustration 42* shows the in-sample performance on the splitted series data set and the out-of-sample performance for the splitted data of the two models.

Illustration 42: Performance Comparison ETS and ARIMA (0,0,1)(1,0,2)(7)

	ETS	**ARIMA (0,0,1)(1,0,2)(7)**
RMSE In-Sample (Splitted Data)	35.05	37.65
RMSE Out-Of-Sample (Splitted Data)	44.60	44.73

As one can see, the ETS model clearly performs better. Based on that, the final forecast is done with the ETS model as outlined in *Illustration 43*.

Illustration 43: Final Lettuce Forecast Lettuce for Store 12631 (in ounces)

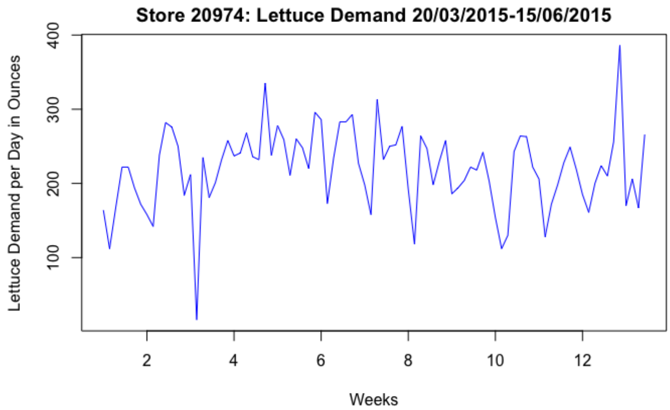
Day	**Forecast**	**Low (80% Confidence)**	**High (80% Confidence)**
1	258	210	307
2	276	224	328
3	303	245	360
4	307	249	366
5	233	188	278
6	266	215	318
7	247	199	295
8	258	208	309
9	276	222	330
10	303	243	362
11	307	246	368
12	233	187	280
13	266	213	320
14	247	197	297

4 Store 20974

4.1 Data Description

The first step of the analysis for store 20974 is to simply inspect the plot of the time series as outlined in *Illustration 44* below. The time series was created and plotted with a frequency of seven in order to convert the daily data into a weekly time frame. For store 20974, it is important to see that the data from the 06-03-2015 to the 19-03-2015 is either missing or corrupted as the values differ significantly from the other values. Therefore, the decision was made to start the underlying time series data at 20-03-2015 in order to not bias the results. From a purely visual inspection of the newly resulting time series, one can assume that the times series is more or less trend-stationary.

Illustration 44: Store 20974 Time Series Plot

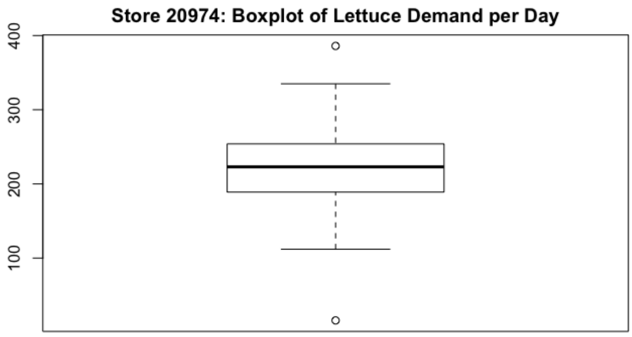


Further, with additionally inspecting the histogram in *Illustration 45*, it becomes visible that the data is very close to a normal distribution with the presence of some serious outliers.

Illustration 45: Store 20974 Time Series Histogram

The inspectation of the boxplot in *Illustration 46* confirms the before made observation of the presence of some significant outliers.

Illustration 46: Store 20974 Time Series Boxplot



Lastly, it is always helpful to check and to bear in mind some basic summary statistics as outlined below in *Illustration 47*.

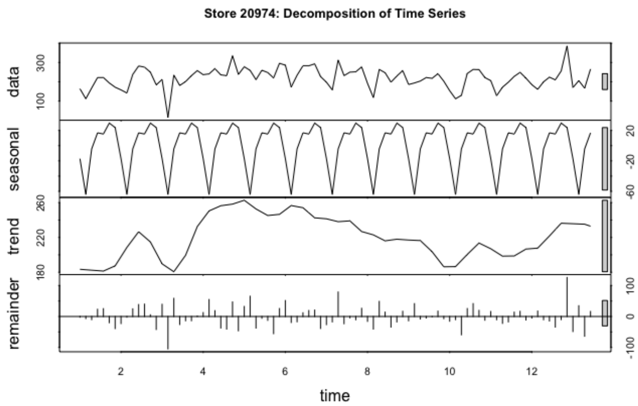
Illustration 47: Store 20974 Summary Statistics

Measurement	Result
Mean	219.7
Median	223
Standad Deviation	54.14

4.2 Decomposition of the Time Series

After descriptively exploring the original time series, the next step is to decomposite the time series into a seasonal, trend and remainder component as outlined in *Illustration 48* on the next page.

Illustration 48: Store 20974 Time Series Decomposition



By looking at the above illustration two key observations are apparent. First of all, the times series seems to have a strong additive seasonal component that repeats itself in approximately weekly cycles as already observed before. Secondly, there does not appear to be any particular overall trend over the cause of the observed time period. These two observations indicate that for any good modelling practice a seasonal component needs to be included while the trend component can most likely be disregarded.

4.3 Holt-Winters

As a next step, a first modelling approach in form of the Holt-Winters model will be applied. The Holt-Winters method uses exponential smoothing to model a time series while accounting for an average component (alpha), a trend/slope component (beta) and a seasonal component (gamma). By the nature of the task, beta is set to 0. By applying the Holt-Winters function to the times series of store 20974 the following parameter result:

Alpha = 0.15

Beta = 0

Gamma = 0.23

a = 220.11

l = (27.29, 45.89, 52.99, -22.00, -49.60, -12.64, 30.37)

The above results confirm the before made assumptions. The resulting gamma = 0.23 confirms that there is considerable seasonal component observable. However, the relatively low numbers for both parameters indicates that the model takes into account a considerable number of lags that are further away rather than values of only the very recent past days. Overall, the Holt-Winters model results in an in-sample root mean squared error of 47.29.

Next, by applying the newer and often times better-performing ETS function of the Holt-Winters method the results are slightly different.

Alpha = 0.16

Beta = 0

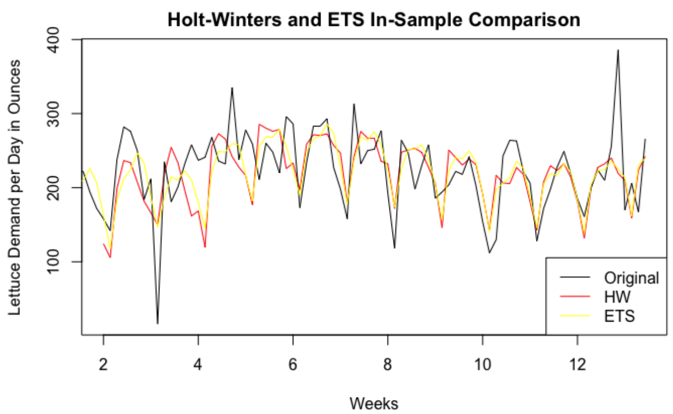
Gamma = 0.0001

a = 198.21

l = (18.32, 32.35, 18.33, 17.10, -2.61, -64.01, -19.51)

The resulting model proves to be an additive ETS model. The ETS model reduces the root mean squared error to 42.50 compared to 47.29 of the original Holt-Winters modelling technique. *Illustration 49* shows the modelled results of the ETS and the original Holt-Winters modelling technique compared to the original times series. One can observe, that the original Holt-Winters modelling technique and ETS model behave very similar and are hardly distinguishable from a purely visual point of view.

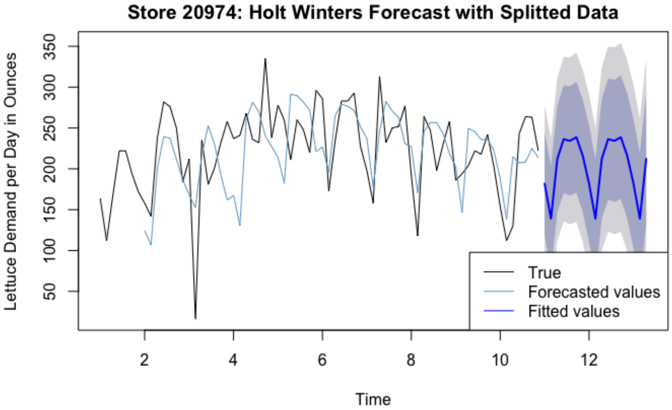
Illustration 49: Store 20974 Holt Winters and ETS In-Sample Comparison



However, in order to comprehensively and meaningfully assess the accuracy and precision of both introduced models, there is a need to test the models forecasting power. A way to do this is to split the data into a training data set and into a validation data set in order to get an out-of-sample accuracy/ error. Therefore, the 88 observations of the original time series get split into a training data set of 61 observations (80% of the original data) and into a validation data set with 17 observations (20% of the original data).

Illustration 50 below shows the forecast, in-sample-error and out-of- error for the splitted data set for the original Holt-Winters modelling technique.

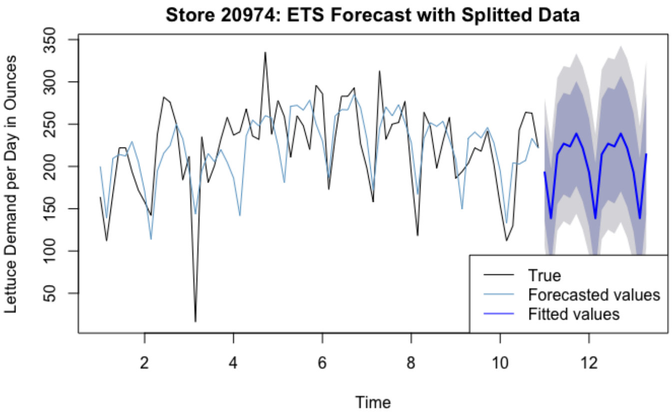
Illustration 50: Store 20974 Results Holt Winters Forecast Splitted Data



Error Type	Result
RMSE In-Sample	48.89
RMSE Out-Of-Sample	49.16

Illustration 51 below shows the forecast, in-sample-error and out-of-error for the splitted data set for the ETS modelling technique.

Illustration 51: Store 20974 Results ETS Forecast Splitted Data

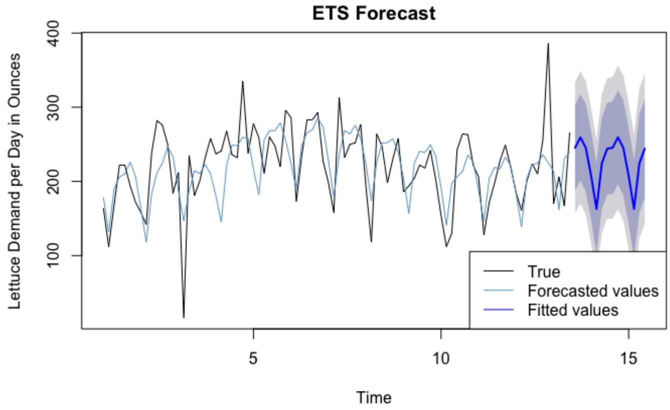


Error Type	Result
RMSE In-Sample	41.53
RMSE Out-Of-Sample	47.64

Again, from a purely visual perspective, it is hard to see any difference between the two forecasting modelling techniques. By consulting the out-of-sample error, one can see that the ETS model has clearly a lower RMSE. Moreover, the ETS model has also a slightly lower in-sample error compared to the original Holt-Winters modelling technique. Based on that, the choice is to proceed further with ETS model and to compare this model to the best performing ARIMA model.

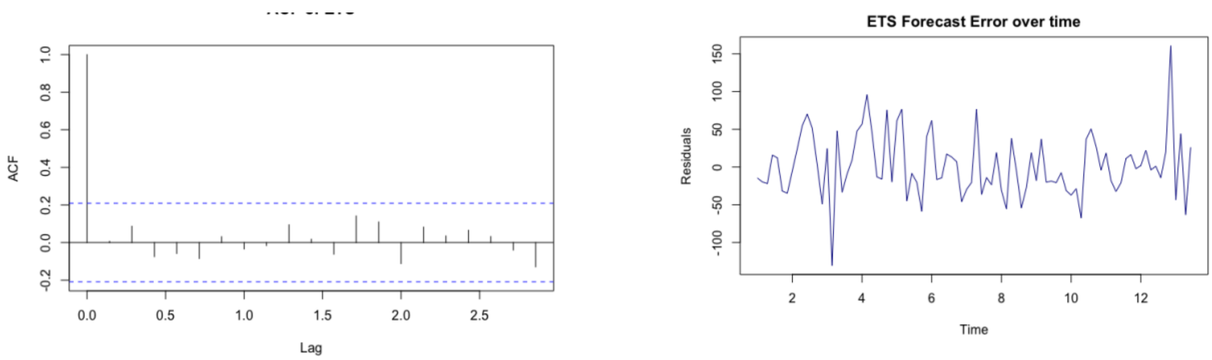
As a next step, ETS is getting retrained with the entire data set (88 observations) and a 14-day forecast is made. The resulting forecast is illustrated in *Illustration 52*.

Illustration 52: Store 20974 14-Days ETC Forecast



As a last step, there is a need to examine the resulting residual to assess the quality of the model. More precisely, there needs to be checked whether the error term behaves heteroscedastic and whether there is a correlation between the modelled error terms and the forecasted values. For these reasons, *Illustration 53* shows the residuals over time as well as the autocorrelation of the error term.

Illustration 53: Store 20974 Residual Analysis ETS



By examining the autocorrelation function of the residuals on the left-hand side of *Illustration 53*, one can see that at none of the 20 examined lags a significant correlation can be observed. Moreover, by examining the residuals over time on the right-hand side of *Illustration 53*, one can see that except an outlier around week 3 and week 13, the residuals are more or less homogenously distributed (no trend in variance) and are distributed around the zero-mean. To confirm these results quantitatively the Ljung-Box, the Argumented Dickey-Fuller and the Box-Pierce test were run. As shown in *Illustration 54* there is little evidence that the errors are non-stationary or correlated/ dependent.

Illustration 54: Store 20974 Quantitative Residual Analysis ETS

Test	Null Hypothesis	p-value	Result (5% Significance)
Box-Ljung	Indendetly Distributed Residuals	0.9126	Independent Residuals
Argumented Dickey-Fuller	Non-Trend-Stationary Residuals	0.01	Trend Stationary Residuals
Box-Pierce	zero autocorrelated residuals	0.2514	Zero Autocorrelated Residuals

4.4 ARIMA

To start the construction of the ARIMA model, certain tests to check whether there is a need to preprocess the original data in order to make the data fit for the ARIMA model need to be run. More precisely, it is of interest to check whether the original data is stationary, independently distributed, non-correlated and has a constant variance. Therefore, the tests as outlined in *Illustration 55* were conducted.

Illustration 55: Store 20974 Quantitative Time Series Analysis

Test	Null Hypothesis	p-value	Result (5% Significance)
Box-Ljung	Indendetly Distributed Data	0.01236	(H1) Dependent Data
Argumented Dickey-Fuller	Existence of Unit Root	0.01	(H1) Non-Existence of Unit Root
Kwiatkowski-Philips-Schmidt-Shin	Trend Stationarity	0.1	(H0) Trend Stationary Data
Box-Pierce	zero autocorrelated data	0.01391	(H1) Autocorrelated Data
Philips-Perrron	Integrated of order 1	0.01	(H1) Not Itegrated of Order 1

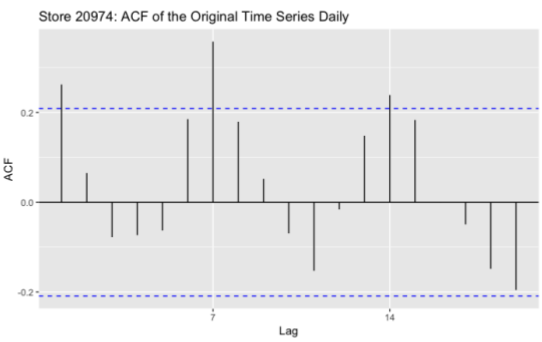
Having a look at all of the above test results, it can be concluded that the original time series is trend-stationary in variance and mean but also dependent/ autocorrelated. In addition to the above test, the built-in R-Functions ‘ndiffs’ and ‘nsdiffs’ which help to determine how many order differences, respectively how many seasonal differences one has to take in order to make the time series stationary were run as outlined in *Illustration 56.*

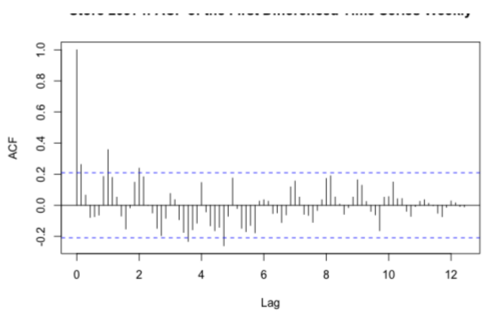
Illustration 56: Store 20974 Output R Preprocessing Functions

R-Function	**Result**
ndiff	0
nsdiff	0

The result of ‘ndiff’ confirms that the times series is trend stationery and does not need to be first-ordered differenced. Moreover, the result of ‘nsdiff’ indicates that no first-order difference of the seasonal component has to be taken and that therefore the time series is in terms of trend and seasonality stationary. Therefore, one can derive from the so far gained insights that suitable ARIMA model likely needs to be of the form ARIMA (p,0,q)(P,0,Q)(7). As a next step, the ACF and the PACF of the time series in *Illustration 57* need to be inspected.

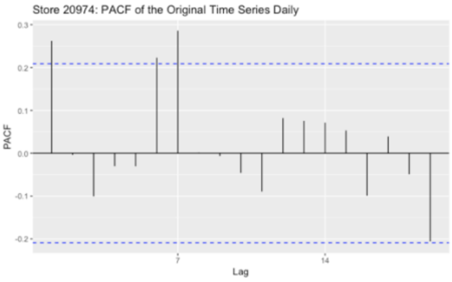
Illustration 57: Store 20974 Autocorrelation Function on Daily and Weekly Basis

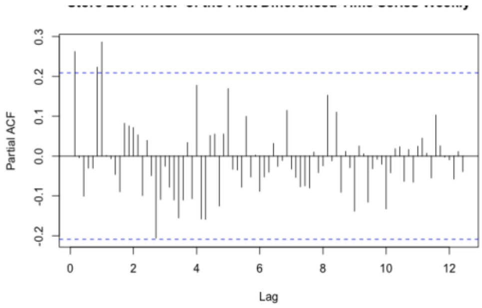




From the autocorrelation function on a daily and weekly basis one can see that there is not a highly statistically significant spike. Therefore, he non-seasonal MA and AR are likely 0 or very close to 0.

Illustration 58: Store 20974 Partial Autocorrelation Function on Daily and Weekly Basis





For the Partial Autocorrelation Function on a daily and weekly there is hardly a clear pattern observable, but one can recognize the statistically significant lags at 6 and 7. Therefore, the modelling ARIMA with Q or P = 1 is potentially suitable.

R has the built-in function “auto ARIMA” which automatically tries to fit the best parameters. The two best performing models according to autoarima are ARIMA (0,0,0)(1,0,0)(7) and ARIMA (1,0,0)(1,0,0)(7). Additionally, multiple other models based on the before made findings derived from the ACF and PACF plot were tried out. The best performing model based on these findings is ARIMA (0,0,1)(1,0,1)(7). As a next, step the goal is to find out which of these models performs best.

Again, as with the Holt-Winters methods, in order to critically and meaningfully assess the accuracy and precision of each models we need to test the models’ forecasting power. A way to do this is to split the data into a training data set and into a validation data set in order to get an out-of-sample accuracy/ error of each model. Therefore, we again use the splitted data to asses the performance of the models. *Illustration 59* sums up the in-sample and out-of-sample performance of each model.

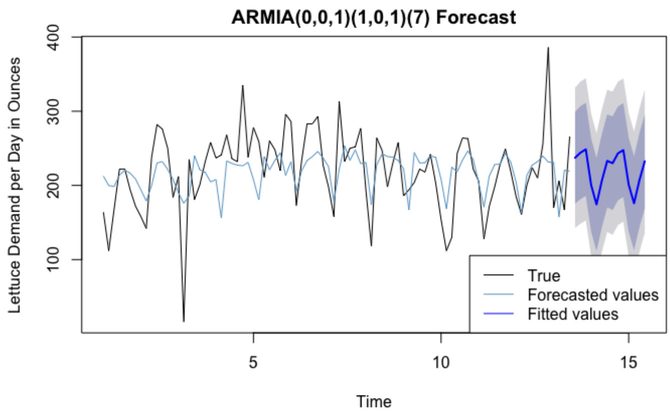
Illustration 59: Store 20974 Results ARIMA Forecast Splitted Data

	ARIMA (0,0,0)(1,0,0)(7)	**ARIMA (1,0,0)(1,0,0)(7)**	**ARIMA (0,0,1)(1,0,1)(7)**
RMSE In-Sample	51.31	48.35	44.87
RMSE Out-Of-Sample	55.49	49.39	47.90

Based on the above results, we will proceed with the best out-of-sample performing model ARIMA (0,0,1)(1,0,1)(7) as this model has the best out-of-sample and in sample performance.

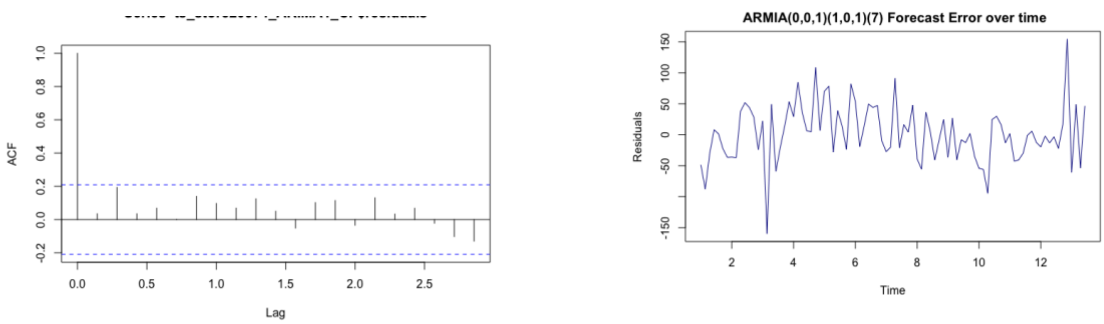
As a next step, the ARIMA (0,0,1) (1,0,1)(7) model is getting trained with the entire time series data and a forecast for the next 14 days was made. The forecast is displayed in *Illustration 60*:

Illustration 60: Store 20974 14-Days Lettuce Forecast with ARIMA (1,0,1)(0,1,1)(7)



As a last step, there is a need to again examine the resulting residuals to assess the quality of the ARIMA(0,0,1) (1,0,1)(7) model. More precisely, there is a need to check whether the error terms are heteroscedastic and whether there is correlation between the modelled error terms and the forecasted values. For these reasons, *Illustration 61* shows the residuals over time as well as the autocorrelation of the error term.

Illustration 61: Store 20974 Residual Analysis ARIMA



By examining the autocorrelation function of the residuals on the left-hand side of *Illustration 61*, one can see that at none of the 20 examined lags a significant correlation can be observed. Moreover, by examining the residual over time on the right-hand side of *Illustration 61*, one can see that except an outlier around week 3 and 13 the residuals are more or less homogenously distributed (no trend in variance) and are distributed around the zero mean. To confirm these results quantitatively the Ljung-Box, the Argumented Dickey-Fuller and the Box-Pierce tests were run. As shown in *Illustration 62* there is little evidence that the errors are non-stationary or correlated/ dependent. Therefore, the forecast with ARIMA(0,0,1) (1,0,1)(7) seems valid.

Illustration 62: Store 20974 Quantitative Residual Analysis ARIMA

Test	Null Hypothesis	p-value	Result (5% Significance)
Box-Ljung	Indendetly Distributed Residuals	0.5999	Independent Residuals
Argumented Dickey-Fuller	Non-Trend-Stationary Residuals	0.0356	Trend-Stationary Residuals
Box-Pierce	Zero Autocorrelated Residuals	0.7425	Zero Autocorrelated Residuals

4.5 Conclusion

To conclude the analysis of the 14-days lettuce forecast for store 20974, a decision needs to be made whether to use the ETS model or the ARIMA (0,0,1)(0,1,1)(7) for the final forecast. As outlined in the previous analysis, both models meet the standards to do meaningful and correct forecasting. It comes now down to decide which model will likely be more precise. I*llustration 63* shows the in-sample performance on the splitted series data set and the out-of-sample performance for the splitted data of the two models.

Illustration 63: Performance Comparison ETS (A,N,A) and ARIMA (1,0,3)(0,1,1)(7)

	ETS (A,N,A)	ARIMA (1,0,1)(0,1,1)(7)
RMSE In-Sample (Splitted Data)	41.53	44.87
RMSE Out-Of-Sample (Splitted Data)	47.64	47.90

As one can see, the ETS model performs better. Based on that, the final forecast is done with the ETS model as outlined in *Illustration 64*.

Illustration 64: Final Lettuce Forecast Lettuce for Store 20974 (in ounces)

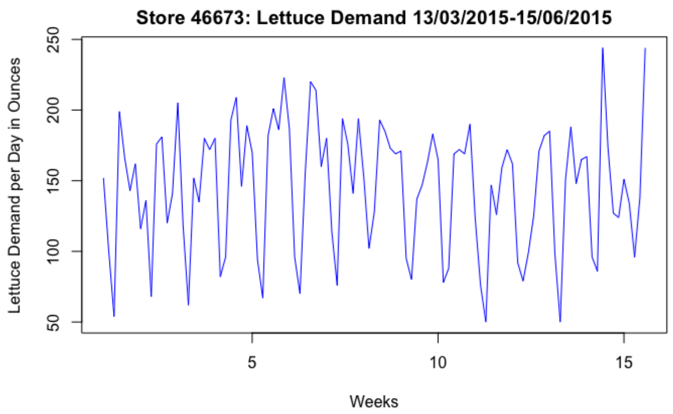
Day	Forecast	Low (80% Confidence)	High (80% Confidence)
1	245	188	303
2	259	201	318
3	245	186	304
4	208	148	267
5	163	103	223
6	224	163	286
7	244	182	306
8	245	183	308
9	259	196	323
10	245	181	309
11	208	143	272
12	163	98	228
13	224	158	290
14	244	178	311

5 Store 46673

5.1 Data Description

The first step of the analysis part for store 46673 is to simply inspect the plot of the time series as outlined in *Illustration 65* below. The time series was created and plotted with a frequency of seven in order to convert the daily data into a weekly time frame. From a purely visual inspection point, one can assume that the times series is more or less trend-stationary and one can observe a clear seasonal pattern which occurs approximately in weekly cycles.

Illustration 65: Store 46673 Time Series Plot

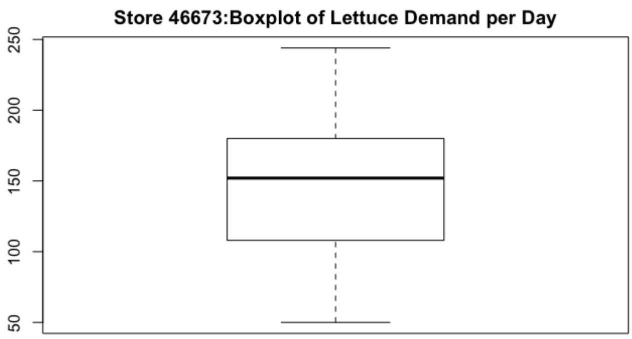


Further, with additionally inspecting the histogram in Illustration 66, it becomes visible that the distribution of the data is slightly left-skewed with an additional spike on the left-hand side.

Illustration 66: Store 46673 Time Series Histogram

Moreover, he inspectation of the boxplot in *Illustration 67* does not provide evidence for any significant outliers.

Illustration 67: Store 46673 Time Series Boxplot



Lastly, it is always helpful to check and to bear in mind some basic summary statistics as outlined in *Illustration 68*.

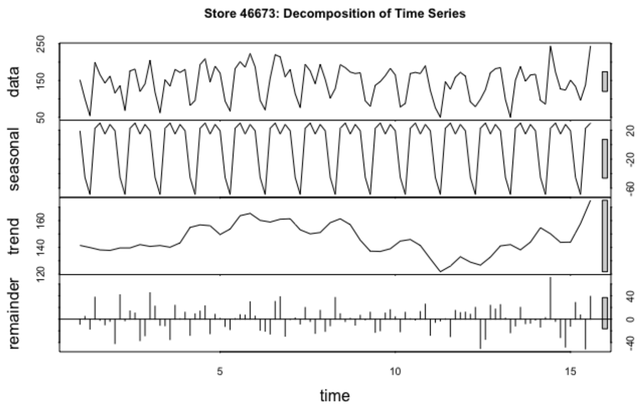
Illustration 68: Store 46673 Summary Statistics

Measurement	**Result**
Mean	154.4
Median	152
Standad Deviation	45.31

5.2 Decomposition of the Time Series

After descriptively exploring the original time series, the next step is to decomposite the time series into a seasonal, trend and remainder component as outlined in *Illustration 69* on the next page.

Illustration 69: Store 46673 Time Series Decomposition



By looking at the above illustration two key observations are apparent. First of all, the times series seems to have a strong additive seasonal component that repeats itself in approximately weekly cycles as already observed before. Secondly, there does not appear to be any particular overall trend over the cause of the observed time period. These two observations indicate that for any good modelling practice a seasonal component needs to be included while the trend component can most likely be disregarded.

5.3 Holt-Winters

As a next step, a first modelling approach in form of the Holt-Winters model will be applied. The Holt-Winters method uses exponential smoothing to model a time series while accounting for an average component (alpha), a trend/slope component (beta) and a seasonal component (gamma). By the nature of the task, beta is set to 0. By applying the Holt-Winters function to the times series of store 46673 the following parameter result:

Alpha = 0.04

Beta = 0

Gamma = 0.28

a = 143.30

l = (11.10, 21.38, 21.20, -35.24, -60.32, 23.38, 45.49)

The above results confirm the before made assumptions. The resulting gamma = 0.22 confirms that there is considerable seasonal component observable. Overall, the Holt-Winters model results in an in-sample root mean squared error of 28.29.

Next, by applying the newer and often times better-performing ETS function of the Holt-Winters method the results are slightly different.

Alpha = 0.0001

Beta = 0

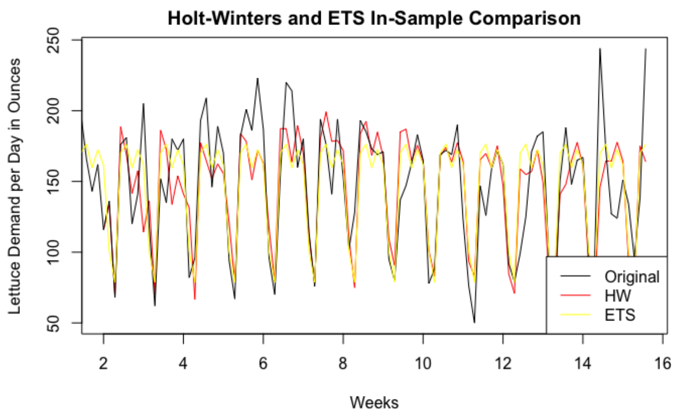
Gamma = 0.0001

a = 145.63

l = (26.82, 13.19, 30.70, 24.32, -66.75, -45.13, -16.13)

The resulting model proves to be a ETS model. The ETS model also reduces the root mean squared error to 25.48 compared to 28.97 of the orginal Holt-Winters modelling technique. *Illustration 70* shows the modelled result ot ETS and the orginal Holt-Winters modelling technique compared to the original times series. One can observe, that the original Holts Winter modelling technique and ETS model behave very similar and are hardly distinguishable from a purely visual point of view.

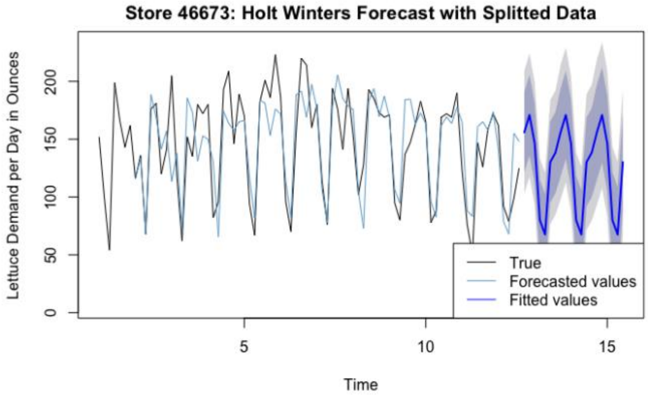
Illustration 70: Store 46673 Holt Winters and ETS In-Sample Comparison



However, in order to comprehensively and meaningfully assess the accuracy and precision of both introduced models there is a need to test the models forecasting power. A way to do this is to split the data into a training data set and into a validation data set in order to get an out-of-sample accuracy/ error. Therefore, the 103 observations of the original time series get split into a training data set of 83 observations (80% of the original data) and into a validation data set with 20 observations (20% of the original data).

Illustration 71 below shows the forecast, in-sample-error and out-of- error for the splitted data set for the original Holt Winters modelling technique.

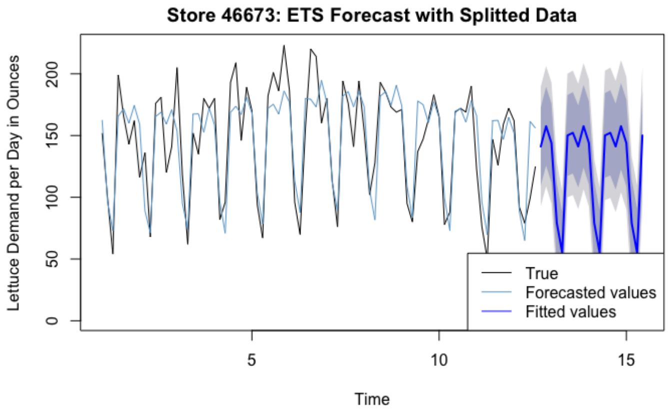
Illustration 71: Store 46673 Results Holt Winters Forecast Splitted Data



Error Type	Result
RMSE In-Sample	26.94
RMSE Out-Of-Sample	37.09

Illustration 72 below shows the forecast, in-sample-error and out-of-error for the splitted data set for the ETS modelling technique.

Illustration 72: Store 46673 Results ETS Forecast Splitted Data

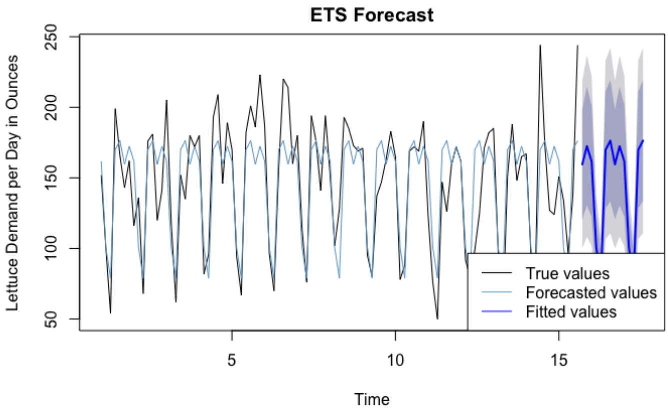


Error Type	Result
RMSE In-Sample	23.22
RMSE Out-Of-Sample	33.28

From a purely visual perspective, a slight difference between the modelling techniques can be observered by having a look at the spikes. Moreover, the out-of-sample error and the in-sample errors indicate that the ETS model performs better. Based on that, the choice is to proceed with the ETS model and to compare this model to the best fitting ARIMA model.

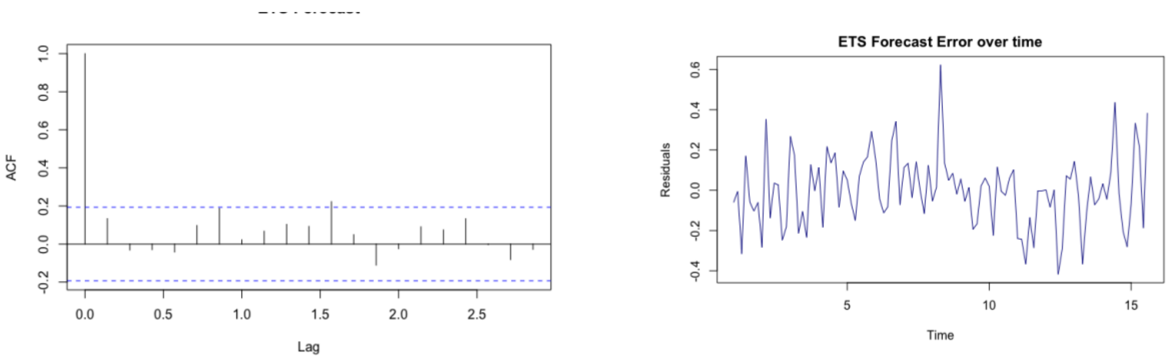
As a next step, ETS is getting retrained with the entire data set (103 observations) and a 14-day forecast is made. The resulting forecast is illustrated in *Illustration 73*.

Illustration 73: Store 46673 14-Days Lettuce ETC Forecast



As a last step, there is a need to examine the resulting residual to assess the quality of the model. More precisely, there needs to be checked whether the error term behaves heteroscedastic and whether there is a correlation between the modelled error terms and the forecasted values. For these reasons, *Illustration 74* shows the residuals over time as well as the autocorrelation of the error term.

Illustration 74: Store 46673 Residual Analysis ETS



By examine the autocorrelation function of the residuals on the left-hand side of *Illustration 74*, one can see that only for a few of the 20 examined lags a slightly significant correlation can be observed. Moreover, by examining the residuals over time on the right-hand side of *Illustration 74*, one can see that except an outlier around week 9 the residuals are more or less homogenously distributed (no trend in variance) and are distributed around the zero mean. To confirm these results quantitatively the Ljung-Box, the Argumented Dickey-Fuller and the Box-Pierce tests were run. As shown in *Illustration 75*, there is little evidence that the errors are non-stationary or correlated/ dependent.

Illustration 75: Store 46673 Quantitative Residual Analysis for ETS (A,N,A)

Test	Null Hypothesis	p-value	Result (5% Significance)
Box-Ljung	Indendetly Distributed Residuals	0.3077	Independent Residuals
Argumented Dickey-Fuller	Non-Trend-Stationary Residuals	0.02438	Trend-Stationary Residuals
Box-Pierce	Rero Autocorrelated Residuals	0.4041	Zero Autocorrelated Residuals

5.4 ARIMA

To start the construction of the ARIMA model, certain tests have to be run in order to check whether there is a need to preprocess the original data in order to make the data fit for the ARIMA model. More precisely, we need to check whether the original data is stationary, independently distributed, non-correlated and has a constant variance. Therefore, the tests as outlined in *Illustration 76* were conducted.

Illustration 76: Store 46673 Quantitative Time Series Analysis

Test	Null Hypothesis	p-value	Result (5% Significance)
Box-Ljung	Indendetly Distributed Data	0.01505	(H1) Dependent Data
Argumented Dickey-Fuller	Existence of Unit Root	0.01	(H1) Non-Existence of Unit Root
Kwiatkowski-Philips-Schmidt-Shin	Trend Stationarity	0.1	(H0) Trend Stationary Data
Box-Pierce	zero autocorrelated data	0.01657	(H1) Autocorrelated Data
Philips-Perrron	Integrated of order 1	0.01	(H1) Not Itegrated of Order 1

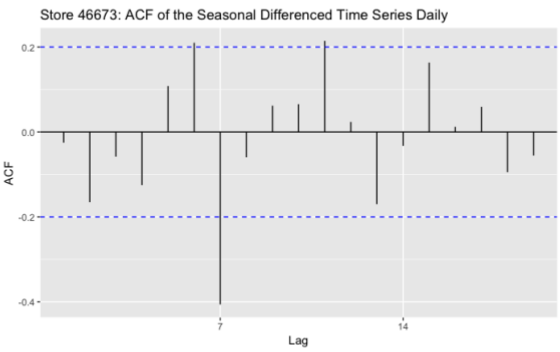
Having a look at all of the above test results, it can be concluded that the original time series is trend-stationary in variance and mean but also dependent/ autocorrelated. In addition to the above test, the built-in R-Functions 'ndiffs' and 'nsdiffs' which help to determine how many order differences, respectively how many seasonal differences one has to take in order to make the time series stationary were run as outlined in *Illustration 77*.

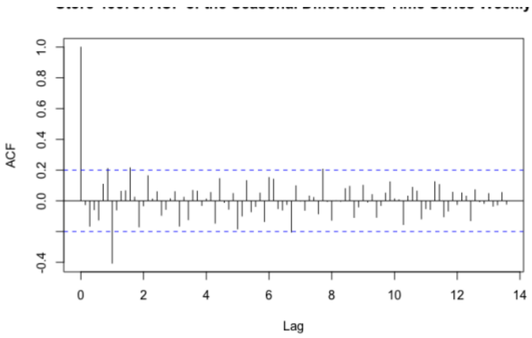
Illustration 77: Store 46673 Output R Preprocessing Functions

R-Function	Result
ndiff	0
nsdiff	1

The result of 'ndiff' confirms that the times series is trend-stationary and does not need to be first-ordered differenced. However, the result of 'nsdiff' indicates that the first order difference of the seasonal component, which is equal to seven days due to the weekly time intervals, needs to be taken in order to make the time series both trend- and seasonal-stationary. Therefore, it can be derived from the so far collected information that possible ARIMA models need to be of the form ARIMA (p,0,q)(P,1,Q)(7). As a next step, the ACF and the PACF of the time series need to be inspected.

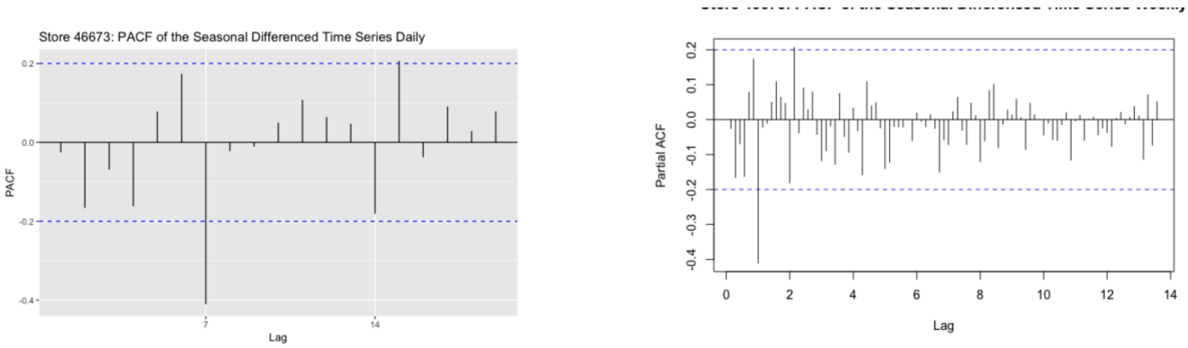
Illustration 78: Store 46673 Autocorrelation Function on Daily and Weekly Basis





From the autocorrelation function on a daily basis one can see that there no or at most one statistically significant lag that likely should be included in the model (so p and q are likely 0 or 1). Additionally, as the seasonal pattern is already accounted for with the first seasonal difference, no further seasonal component can be observed which can be see with the decline in statistical significance in the weekly ACF plot on the right-hand side of *Illustration 78*

Illustration 79: Store 46673 Partial Autocorrelation Function on Daily and Weekly Basis



For the Partial Autocorrelation Function on a daily and weekly basis one can see that there is a statistically significant impact for the lag at day seven and a less significant one at day 14 which implies that a P between 1 and 2 could make sense.

R has the built-in function “auto ARIMA” which automatically tries to fit the best parameters. The two best performing models according to autoarima are ARIMA (0,0,0)(2,1,0)(7) and ARIMA (0,0,1)(2,1,0)(7). Additionally, multiple other models based on the before made findings derived from the ACF and PACF plot were tried out. The best performing model based on these findings is ARIMA (1,0,0)(2,1,0)(7). As a next, step the goal is to find out which of these models performs best.

Again, as with the Holt-Winters methods, in order to critically and meaningfully assess the accuracy and precision of each models we need to test the models’ forecasting power. A way to do this is to split the data into a training data set and into a validation data set in order to get an out-of-sample accuracy/ error of each model. Therefore, we again use the splitted data to asses the performance of the models. *Illustration 80* sums up the in-sample and out-of-sample performance of each model.

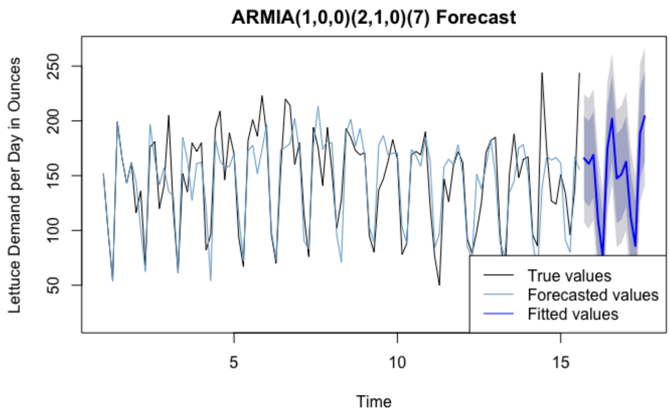
Illustration 80: Store 46673 Results ARIMA Forecast Splitted Data

	ARIMA (0,0,0)(2,1,0)(7)	**ARIMA (0,0,1)(2,1,0)(7)**	**ARIMA (1,0,0)(2,1,0)(7)**
RMSE In-Sample	25.63	25.09	25.07
RMSE Out-Of-Sample	38.61	38.09	38.01

Based on the above results, the model ARIMA (1,0,0)(2,1,0)(7) performs best. Therefore, the decision is to proceed with this model.

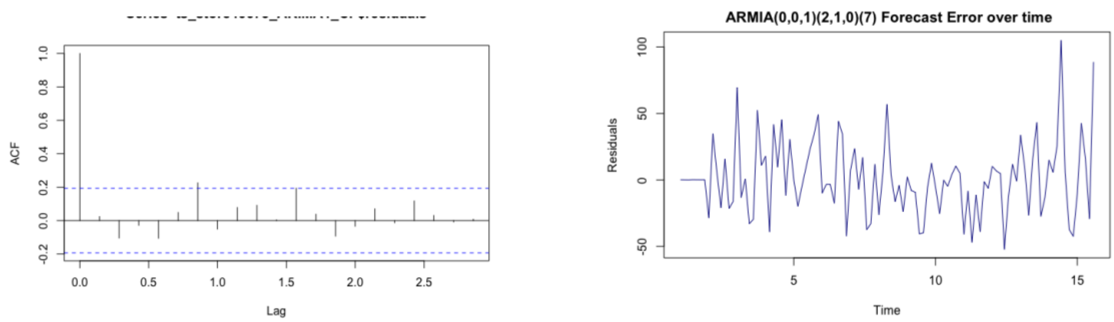
As a next step, the ARIMA (1,0,0) (2,1,0)(7) model is getting trained with the entire time series data and a forecast for the next 14 days was made. The forecast is displayed in *Illustration 81*:

Illustration 81: Store 46673 14-Days Lettuce Forecast with ARIMA (1,0,0)(2,1,0)(7)



As a last step, there is a need to again examine the resulting residuals to assess the quality of the ARIMA(1,0,0) (2,1,0)(7) model. More precisely, there is a need to check whether the error term is heteroscedastic and whether there is correlation between the modelled error terms and the forecasted values. For these reasons, *Illustration 82* shows the residuals over time as well as the autocorrelation of the error term.

Illustration 82: Store 46673 Residual Analysis ARIMA (1,0,0)(2,1,0)(7)



By examine the autocorrelation function of the residuals on the left-hand side of *Illustration 82*, one can see that at only a very few residuals of the 20 examined lags have a significant, although the correlations are very small and therefore it could be coincidentally. Moreover, by examine the residual over time on the right-hand side of *Illustration 82*, one can see that except few outliers at the very end, the residuals are more or less homogenously distributed (no trend in variance) and are distributed around the zero mean. To confirm this result quantitatively the

Ljung-Box, the Argumented Dickey-Fuller and the Box-Pierce test were run. As shown in *Illustration 83* there is little evidence that the errors are non-stationary or correlated/ dependent. Therefore, the forecast with ARIMA(1,0,0) (2,1,0)(7) seems valid.

Illustration 83: Store 46673 Quantitative Residual Analysis ARIMA

Test	Null Hypothesis	p-value	Result (5% Significance)
Box-Ljung	Indendetly Distributed Residuals	0.539	Independent Residuals
Argumented Dickey-Fuller	Non-Trend-Stationary Residuals	0.01	Trend-Stationary Residuals
Box-Pierce	Zero Autocorrelated Residuals	0.8071	Zero Autocorrelated Residuals

5.5 Conclusion

To conclude the analysis of the 14-days lettuce forecast for store 46673, a decision needs to be made whether to use the ETS model or the ARIMA (1,0,0)(2,1,0)(7) for the final forecast. As outlined in the previous analysis, both models meet the standards to do meaningful and correct forecasting. It comes now down to decide which model will likely be more precise. *Illustration 84* shows the in-sample performance on the splitted series data set and the out-of-sample performance for the splitted data of the two models.

Illustration 84: Performance Comparison ETS and ARIMA (1,0,1)(0,1,1)(7)

	ETS	**ARIMA (0,0,1)(2,1,0)(7)**
RMSE In-Sample (Entire Data)	25.48	28.38
RMSE Out-Of-Sample (Splitted Data)	30.03	33.20

As one can see, the ETS model performs better. Based on that, the final forecast is done with the ETS model as outlined in *Illustration 85*.

Illustration 85: Final Lettuce Forecast Lettuce for Store 46673 (in ounces)

Day	Forecast	Low (80% Confidence)	High (80% Confidence)
1	160	121	198
2	172	131	214
3	162	122	201
4	101	76	125
5	79	60	98
6	170	129	211
7	176	133	219
8	160	121	198
9	172	131	214
10	162	122	201
11	101	76	125
12	79	60	98
13	170	129	211
14	176	133	219