CASE STUDY#1
SALES FORECASTING

Identify the best forecasting model to predict monthly sales of a grocery chain

Alarmelu Pichu Mani – TJ6723

*Q1 - **Identify time series components and plot the data.***

1a- Create time series data set in R using the ts() function.

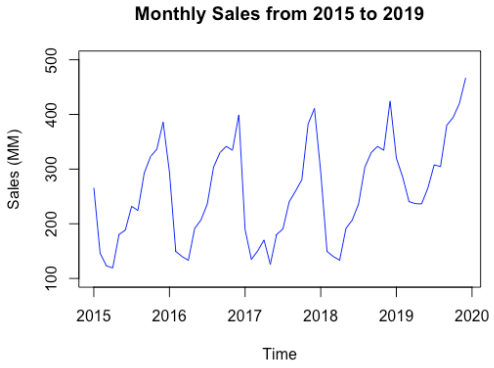
The ts() function in R creates a time series object by taking in our historical grocery chain sales dataset (a numeric vector). The code is as follows.

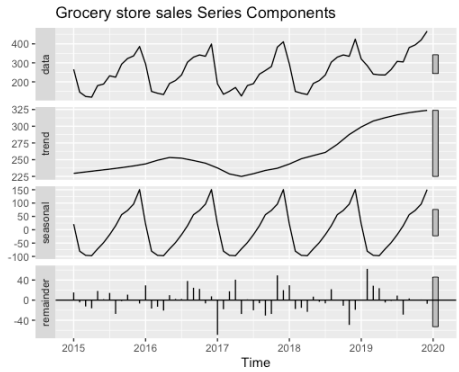
```
grocery.ts <- ts(grocery.data$Sales,start = c(2015,1), end = c(2019, 12), freq = 12)
```

where start and end are the times of the first and last observation and frequency is the number of observations per unit time (1=annual, 4=quarterly, 12=monthly).

1b-Employ the plot() function to create a data plot with the historical data, provide it in your report, and explain what data patterns can be visualized in this plot.

Plotting the time series data created using the historical sales data helps us visually notice trends and seasonality. From our plot here, we see that the sales follows an annual seasonal pattern. There is a spike in sales in December every year. As for the trend, we are seeing a linear upward trend over the time period in the historical data.





1c- Apply the Acf() function to identify possible time series components. Provide in the report the autocorrelation chart and explain the time series components existing in the historical data.

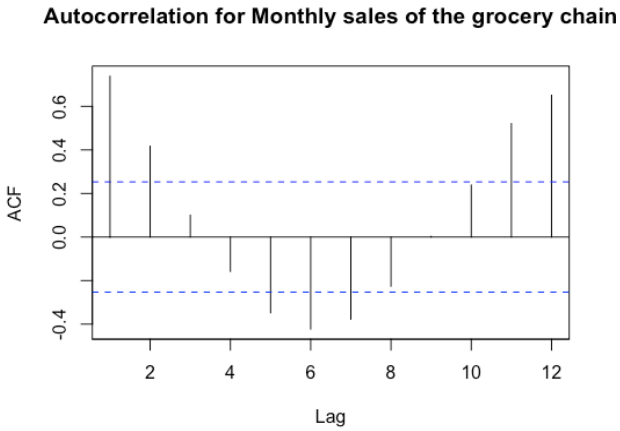


Figure 1 - Correlogram for the grocery.ts

Autocorrelation explains the correlation between a time series and the same lagged one or more periods. Autocorrelation coefficient is a measure of autocorrelation between a variable itself and the same variable lagged k periods.The Acf() in R identifies and plots multiple autocorrelation coefficients for various lags of a time series data into a correlogram.This helps to to answer whether the data has randomness,trend, seasonality and level components.

Interpretation of the correlogram for our dataset:

The blue dotted lines in the correlogram represents the statistical level of significance. If any correlation is above the level, it imples that there is a significant correlation between the data. In our data we see a siginificant level of correlation.

The first positive coefficients here indicate that the data has an upward trend. Since the coefficients for lag-1 and lag-12 are significantly high it shows the annual seasonality present in the dataset. The level of the data shows a decrease and increases after a certain point at lag 10. Some coefficients are lower than the level of significance which shows the randomness of the data.

*Q2 - **Use trailing MA for forecasting time series.***

2a- Use the rollmean() function to develop three trailing MAs (apply the entire data set with no partitioning) for the window width of 2, 6, and 12, respectively. Present the R code for these MAs in your report.

Moving average is a common method of time series forecasting. As soon a new actual data point is available, a revised moving average is calculated for the next data period. Hence the name moving average. In R, the rollmean() method calculates the moving average with the specified parameters – k is the window width, align is the alignment of the moving average.

```
ma.trailing_2 <- rollmean(grocery.ts, k = 2, align = "right")
ma.trailing_6 <- rollmean(grocery.ts, k = 6, align = "right")
ma.trailing_12 <- rollmean(grocery.ts, k = 12, align = "right")
```

2b - Use the forecast() function to create a trailing MA forecast for each window width in 12 months of 2020, and present these forecasts in your report.

The forecast() method in R is a generic function for forecasting from time series or time series models. Trailing moving average forecast for a window of 2 in the moving average.

```
ma.trailing_2.pred <- forecast(ma.trailing_2, h=12, level = 0)
formattable(data.frame(ma.trailing_2.pred))
```

	Point.Forecast	Lo.0	Hi.0
Jan 2020	405.9923	405.9923	405.9923
Feb 2020	290.9559	290.9559	290.9559
Mar 2020	237.4785	237.4785	237.4785
Apr 2020	229.5528	229.5528	229.5528
May 2020	238.1631	238.1631	238.1631
Jun 2020	258.8753	258.8753	258.8753
Jul 2020	286.2924	286.2924	286.2924
Aug 2020	322.4193	322.4193	322.4193
Sep 2020	360.0991	360.0991	360.0991
Oct 2020	384.8573	384.8573	384.8573
Nov 2020	405.3897	405.3897	405.3897
Dec 2020	443.6000	443.6000	443.6000

Figure 2-Trailing MA forecast output for window width =2

Trailing moving average forecast for a window of 6 in the moving average.

ma.trailing_6.pred <- forecast(ma.trailing_6, h=12, level = 0)

formattable(data.frame(ma.trailing_6.pred))

	Point.Forecast	Lo.0	Hi.0
Jan 2020	385.8483	385.8483	385.8483
Feb 2020	369.9479	369.9479	369.9479
Mar 2020	344.8978	344.8978	344.8978
Apr 2020	314.1715	314.1715	314.1715
May 2020	284.4369	284.4369	284.4369
Jun 2020	250.6839	250.6839	250.6839
Jul 2020	246.4608	246.4608	246.4608
Aug 2020	265.3713	265.3713	265.3713
Sep 2020	291.2923	291.2923	291.2923
Oct 2020	318.9508	318.9508	318.9508
Nov 2020	347.5250	347.5250	347.5250
Dec 2020	379.0666	379.0666	379.0666

Figure 3-Trailing MA forecast output for window width =6

Trailing moving average forecast for a window of 12 in the moving average.

ma.trailing_12.pred <- forecast(ma.trailing_12, h=12, level = 0)

formattable(data.frame(ma.trailing_12.pred))

	Point.Forecast	Lo.0	Hi.0
Jan 2020	326.2019	326.2019	326.2019
Feb 2020	330.8036	330.8036	330.8036
Mar 2020	335.4054	335.4054	335.4054
Apr 2020	340.0071	340.0071	340.0071
May 2020	344.6088	344.6088	344.6088
Jun 2020	349.2106	349.2106	349.2106
Jul 2020	353.8123	353.8123	353.8123
Aug 2020	358.4141	358.4141	358.4141
Sep 2020	363.0158	363.0158	363.0158
Oct 2020	367.6175	367.6175	367.6175
Nov 2020	372.2193	372.2193	372.2193
Dec 2020	376.8210	376.8210	376.8210

Figure 4- Trailing MA forecast output for window width =12

From the forecasted data(for windows 2,6 and 12), we can see that the forecast is close to the most recent historical data – 2019 than it is to the previous years' data.

2c- Develop a seasonal naïve forecast for the entire historical data set, and apply the accuracy() function to compare accuracy of the four models: seasonal naïve forecast and trailing MAs with window width of 2, 6, and 12, respectively. Present the accuracy measures in your report, compare MAPE and RMSE of these forecasts, and identify the best forecasting model.

A naive forecast is *Seasonal naïve forecast* for a seasonal time series is the value of the most recent identical season. The accuracy function in R give us the several error measures using the provided data, which include ME: Mean Error, RMSE: Root Mean Squared Error, MAE: Mean Absolute Error, MPE: Mean Percentage Error, MAPE: Mean Absolute Percentage Error, MASE: Mean Absolute Scaled Error, ACF1: Autocorrelation of errors at lag 1 and Theil's *U* statistic which is a relative accuracy measure that compares the forecasted results with the results of forecasting with minimal historical data. In all the cases, the general rule is the lower the value the better the forecast.

The error profiles are as follows:

Seasonal naïve forecasting

round(accuracy(grocery.snaive.pred$fitted,grocery.ts), 3)

	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil's U
Test set	21.702	54.943	44.606	6.166	17.942	0.452	0.851

Trailing MA with a window of 2

round(accuracy(ma.trailing_2,grocery.ts),3)

	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil's U
Test set	1.703	30.391	22.024	-1.271	9.792	0.182	0.5

Trailing MA with a window of 6

round(accuracy(ma.trailing_6,grocery.ts),3)

	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil's U
Test set	11.996	82.093	72.857	-5.669	32.559	0.736	1.73

Trailing MA with a window of 12

round(accuracy(ma.trailing_12,grocery.ts),3)

	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil's U
Test set	12.312	85.149	73.366	-7.468	31.116	0.681	1.612

As per the accuracy results, we can see that the lowest RMSE and MAPE is seen for the Trailing MA with a window of 2. Going by the RMSE and MAPE, the best forecasting model in this mix is the *Trailing MA with a with a window of 2.*

Q3 - ***Apply the two-level forecast with regression and trailing MA for residuals.***

De-trending and de-seasonalizing the historical data are used in 2-level forecasting methods. This is because, trailing MA is useful for data that lack seasonality and trend. In order to apply MA in data with trends and seasonality, it is good to first "de-seasonalize and de-trend" the time series data. Regression models can be used for this purpose. The regression residuals (noise and levels) can be forecasted using trailing MA. The final forecasting would be a sum of the trend and seasonality forecasting and the trailing MA residuals of regression forecast. Hence it is called 2 level forecasting.

3a- To de-trend and de-seasonalize the historical data for the entire data set, develop using the tslm() function a regression model with linear trend and seasonality and forecast sales in 2020 with the forecast() function. Present and briefly explain the model in your report.

De-trend and de-seasonalize the entire historical data using the tslm() function.

reg.trend.seas <- tslm(grocery.ts ~ trend + season)

summary(reg.trend.seas)

```
Call:
tslm(formula = grocery.ts ~ trend + season)

Residuals:
    Min      1Q  Median      3Q     Max
-83.340 -13.068   3.716  22.105  77.237

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  236.232     16.739  14.112  < 2e-16 ***
trend          1.464      0.262   5.588 1.12e-06 ***
season2     -101.284     21.787  -4.649 2.73e-05 ***
season3     -116.889     21.791  -5.364 2.43e-06 ***
season4     -118.593     21.799  -5.440 1.87e-06 ***
season5      -93.697     21.810  -4.296 8.66e-05 ***
season6      -70.501     21.824  -3.230  0.00226 **
season7      -41.226     21.842  -1.887  0.06528 .
season8       -7.750     21.862  -0.355  0.72455
season9       33.946     21.886   1.551  0.12760
season10      50.201     21.912   2.291  0.02649 *
season11      74.197     21.942   3.382  0.00146 **
season12     128.553     21.975   5.850 4.54e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 34.44 on 47 degrees of freedom
Multiple R-squared:  0.8884,    Adjusted R-squared:  0.8599
F-statistic: 31.18 on 12 and 47 DF,  p-value: < 2.2e-16
```

Figure 5- Output of the tslm() summary

Interpretation of the output:

This method provides the basis for the forecasting. Here we have the trend and seasonality attributes in the regression equation. Seasonality is given as part of the summary output, where in depending on the type of forecast (annual, quarterly or monthly) the season attribute gets added to the regression forecast. The characteristics of the linear trend is also given as part of the summary.

Forecast sales in 2020 with the forecast() function.

reg.trend.seas.pred <- forecast(reg.trend.seas, h = 12, level = 0)

formattable(data.frame(reg.trend.seas.pred))

	Point.Forecast	Lo.0	Hi.0
Jan 2020	325.555	325.555	325.555
Feb 2020	225.735	225.735	225.735
Mar 2020	211.595	211.595	211.595
Apr 2020	211.355	211.355	211.355
May 2020	237.715	237.715	237.715
Jun 2020	262.375	262.375	262.375
Jul 2020	293.115	293.115	293.115
Aug 2020	328.055	328.055	328.055
Sep 2020	371.215	371.215	371.215
Oct 2020	388.935	388.935	388.935
Nov 2020	414.395	414.395	414.395
Dec 2020	470.215	470.215	470.215

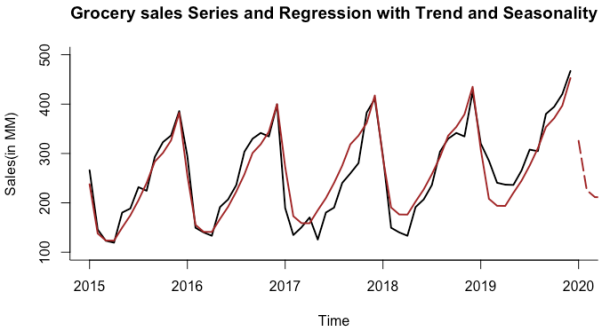


Figure 6 - Regression forecast data and plot

Though the regression forecast with trend and seasonality seems like a good fit to the original data, we can still see some gaps in the plot. This is because of the residuals left out of the fit. This explains the need for the second level of forecasting.

3b-Identify regression residuals, apply a trailing MA (window width of 2) for these residuals using the rollmean() function, and forecast worldwide monthly sales in 12 months of 2020 (use the forecast() function). Combine the regression and trailing MA residuals' forecast for 2020, and present in your report a table that contains regression forecast, trailing MA forecast for residuals, and total (combined) forecast in 2020.

Identify regression residuals

This is the difference between the time series historical data and the regression forecast.

reg.trend.seas.res <- reg.trend.seas$residuals

reg.trend.seas.res

	Jan	Feb	Mar	Apr	May	Jun
2015	28.3033333	8.0233333	-0.6366667	-4.1966667	30.4433333	13.9833333
2016	39.0316667	-5.9483333	-1.2083333	-7.7683333	23.9716667	14.9116667
2017	-83.3400000	-38.2200000	-8.4800000	11.7600000	-59.4000000	-29.2600000
2018	3.8883333	-41.0916667	-36.3516667	-42.9116667	-11.1716667	-20.2316667
2019	12.1166667	77.2366667	46.6766667	43.1166667	16.1566667	20.5966667
	Jul	Aug	Sep	Oct	Nov	Dec
2015	26.5433333	-15.6966667	9.4433333	21.8233333	9.9633333	3.5433333
2016	13.1716667	45.8316667	28.9716667	22.9516667	-9.6083333	-0.6283333
2017	-50.0000000	-35.0400000	-58.8000000	-55.7200000	21.0200000	-6.7000000
2018	-21.9716667	10.6883333	-6.1716667	-12.1916667	-44.7516667	-10.5716667
2019	32.2566667	-5.7833333	26.5566667	23.1366667	23.3766667	14.3566667

Figure 7 - Residuals

MA using rollmean()

ma.trailing.res_2 <- rollmean(reg.trend.seas.res, k = 2, align = "right")

Create forecast for residuals for the 12 periods into the future.

ma.trailing.res_2.pred <- forecast(ma.trailing.res_2, h =12, level = 0)

Combine regression forecast and trailing MA forecast for residuals.

ts.forecast.12 <- reg.trend.seas.pred$mean + ma.trailing.res_2.pred$mean

making a single data frame with regression, MA and forecast data

total.reg.ma.pred <- data.frame(reg.trend.seas.pred$mean, ma.trailing.res_2.pred$mean, ts.forecast.12)

formattable(data.frame(total.reg.ma.pred))

reg.trend.seas.pred.mean	ma.trailing.res_2.pred.mean	ts.forecast.12
325.555	18.86711	344.4221
225.735	18.86711	244.6021
211.595	18.86711	230.4621
211.355	18.86711	230.2221
237.715	18.86711	256.5821
262.375	18.86711	281.2421
293.115	18.86711	311.9821
328.055	18.86711	346.9221
371.215	18.86711	390.0821
388.935	18.86711	407.8021
414.395	18.86711	433.2621
470.215	18.86711	489.0821

Figure 8-regression forecast, trailing MA forecast for residuals, and total (combined) forecast in 2020.

The combined forecast – ts.forecast.12 is the sum of the regression forecast of the linear trend and seasonality and the moving average forecast for the noise and level from based on the historical data.

3c- Apply the accuracy() function to compare accuracy of the three forecasting models: seasonal naïve forecast (applied in question 2c), regression model with trend and seasonality, and two-level (combined) model with regression and trailing MA for residuals. Present the accuracy measures in your report, compare MAPE and RMSE of these forecasts, and identify the best forecasting model.

The required error profiles are as follows:

Seasonal naïve forecasting

round(accuracy(grocery.snaive.pred$fitted,grocery.ts), 3)

	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil's U
Test set	21.702	54.943	44.606	6.166	17.942	0.452	0.851

Regression model with trend and seasonality

round(accuracy(reg.trend.seas.pred$fitted, grocery.ts), 3)

	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil's U
Test set	0	30.486	24.128	-1.963	10.868	0.525	0.592

Two-level (combined) model with regression and trailing MA for residuals

round(accuracy(reg.trend.seas.pred$fitted+ma.trailing.res_2, grocery.ts), 3)

	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil's U
Test set	-0.118	14.838	11.453	-0.513	5.218	-0.363	0.294

From the error profiles, by comparing the MAPE and RMSE of the 3 models we have in the mix, the two-level (combined) model with regression and trailing MA for residuals perform the best. Hence this is the best forecasting model among the 3.

Q4 - Use advanced exponential smoothing methods.

4a - Develop data partition with the validation partition of 12 historical periods and training partition of the rest of the historical periods. Present in your report the data associated with each partition.

The dataset is partitioned using the following R code.

```
nValid <- 12
nTrain <- length(grocery.ts) - nValid
train.ts <- window(grocery.ts, start = c(2015, 1), end = c(2015, nTrain))
valid.ts <- window(grocery.ts, start = c(2015, nTrain + 1),
                   end = c(2015, nTrain + nValid))
```

Since we have specified the validation dataset to have 12 historical periods, the training dataset will be the rest of the 48 historical periods.

```
> train.ts
       Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
2015 266.0 145.9 123.1 119.3 180.3 188.5 231.8 224.5 292.8 322.9 336.5 385.9
2016 294.3 149.5 140.1 133.3 191.4 207.0 236.0 303.6 329.9 341.6 334.5 399.3
2017 189.5 134.8 150.4 170.4 125.6 180.4 190.4 240.3 259.7 280.5 382.7 410.8
2018 294.3 149.5 140.1 133.3 191.4 207.0 236.0 303.6 329.9 341.6 334.5 424.5
> valid.ts
       Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
2019 320.1 285.4 240.7 236.9 236.3 265.4 307.8 304.7 380.2 394.5 420.2 467.0
```

Figure 9- Training dataset - train.ts, Validation data set - valid.ts

4b-For the training partition, use the ets() function to develop a Holt-Winter's model with multiplicative error, multiplicative trend, and multiplicative seasonality options, and automated selection of smoothing parameters for the training partition. Present and explain the model in your report. Use the model to forecast worldwide sales for the validation period using the forecast() function, and present the forecast in your report.

Holt-Winter's model is used for time series forecasting of data that contains trend and seasonality. The model can be additive or multiplicative nature depending on the time series. The ets() in R is used to build exponential smoothening models. The inputs for the model parameters decide the type of model we are building. Since we need a model with multiplicative trend, and multiplicative seasonality options, and automated selection of smoothing parameters, the following piece of code is appropriate, where "MMM" specify multiplicative error(M), multiplication trend(M) and multiplicative seasonality (M). As for the smoothening parameters, they are supposed to be automatically selected by R, hence we do not explicitly specify any inputs for it. The model when printed in R provides the following information on what smoothening parameters have been selected.

ses.orig <- ets(train.ts, model = "MMM")

```
> ses.orig
ETS(M,M,M)

Call:
 ets(y = train.ts, model = "MMM")

  Smoothing parameters:
    alpha = 0.0329
    beta  = 1e-04
    gamma = 1e-04

  Initial states:
    l = 228.5661
    b = 1.0021
    s = 1.6115 1.414 1.3034 1.2362 1.1039 0.9278
           0.8093 0.7278 0.5741 0.571 0.6063 1.1147

  sigma:  0.125

     AIC     AICc      BIC
521.8267 542.2267 553.6371
```

Forecasting:

ses.orig.pred <- forecast(ses.orig, h = nValid, level = 0)

formattable(data.frame(ses.orig.pred))

	Point.Forecast	Lo.0	Hi.0
Jan 2019	283.3120	282.1421	282.1421
Feb 2019	154.4293	154.7728	154.7728
Mar 2019	145.7254	146.0438	146.0438
Apr 2019	146.8208	146.7157	146.7157
May 2019	186.5228	185.3252	185.3252
Jun 2019	207.8478	208.9623	208.9623
Jul 2019	238.7822	238.5292	238.5292
Aug 2019	284.6867	284.0084	284.0084
Sep 2019	319.4830	319.5322	319.5322
Oct 2019	337.5579	336.7526	336.7526
Nov 2019	366.9474	366.2768	366.2768
Dec 2019	419.0808	421.0580	421.0580

4c- To make a forecast for the 12 months of 2020, use the entire data set (no partitioning) to develop the Holt-Winter's model using the ets() function with the automated selection of error, trend, and seasonality options, and automated selection of smoothing parameters. Present and explain the model in your report. Use the model to forecast worldwide sales for the 12 months of 2020 using the forecast() function, and present the forecast in your report.

In this case, since the selection of error, trend and seasonality options need to be selected automatically by R, we just specify the input to the model parameter as "ZZZ".

```
hw.ZZZ <- ets(grocery.ts, model = "ZZZ")
```

Forecasting:

```
hw.ZZZ.pred <- forecast(hw.ZZZ, h = 12, level = 0)
formattable(data.frame(hw.ZZZ.pred))
```

	Point.Forecast	Lo.0	Hi.0
Jan 2020	337.3377	337.3377	337.3377
Feb 2020	242.9455	242.9455	242.9455
Mar 2020	227.5951	227.5951	227.5951
Apr 2020	226.4937	226.4937	226.4937
May 2020	243.4364	243.4364	243.4364
Jun 2020	269.2489	269.2489	269.2489
Jul 2020	300.4341	300.4341	300.4341
Aug 2020	334.1278	334.1278	334.1278
Sep 2020	370.6418	370.6418	370.6418
Oct 2020	392.1157	392.1157	392.1157
Nov 2020	416.1190	416.1190	416.1190
Dec 2020	469.1503	469.1503	469.1503

4d- Apply the accuracy() function to compare the two forecasting models: seasonal naïve forecast (applied in question 2c) and Holt-Winter's model developed in question 4c. Present the accuracy measures in your report, compare MAPE and RMSE of these forecasts, and identify the best forecasting model.

Comparison of error profiles:

Holt-Winter's model developed in question 4c:

round(accuracy(hw.ZZZ.pred$fitted,grocery.ts), 3)

	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil's U
Test set	2.516	27.143	20.296	-0.445	9.273	0.09	0.518

Seasonal naïve forecasting from question 2c:

round(accuracy(grocery.snaive.pred$fitted,grocery.ts), 3)

	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil's U
Test set	21.702	54.943	44.606	6.166	17.942	0.452	0.851

By comparing the RMSE and MAPE of the Seasonal naïve forecasting from question 2c and Holt-Winter's model developed in question 4c, we can see that the Holt-winter model performs better than the seasonal naïve. Hence it would be the best among these two models.

4e- Compare the best forecasts identified in questions 3c and 4c. Explain what your final choice of the forecasting model in this case will be.

The best model from question 3c is the 2 level combined forecast model

Two-level (combined) model with regression and trailing MA for residuals

round(accuracy(reg.trend.seas.pred$fitted+ma.trailing.res_2, grocery.ts), 3)

	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil's U
Test set	-0.118	14.838	11.453	-0.513	5.218	-0.363	0.294

The best model from question 4c is the Holt winter model with automated selection of smoothening parameters.

Holt-Winter's model developed in question 4c:

round(accuracy(hw.ZZZ.pred$fitted,grocery.ts), 3)

	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil's U
Test set	2.516	27.143	20.296	-0.445	9.273	0.09	0.518

Among these 2, the RMSE and MAPE for Two-level (combined) model with regression and trailing MA for residuals is the lowest and performance in terms of error profile is better. Hence the Two-level (combined) model with regression and trailing MA is the best choice for this dataset.