7.2 and 7.3 - Holt's Trend Methods

Hyndman and Anthanasopoulos (additional examples by Deppa)

March 15, 2018

7.2 - Trend Methods

Holt's Linear Trend Method

Holt (1957) extended simple exponential smoothing to allow the forecasting of data with a trend. This method involves a forecast equation and two smoothing equations (one for the level and one for the trend):

$$\begin{aligned}\text{Forecast equation} \quad & \hat{y}_{t+h|t} = \ell_t + h b_t \\ \text{Level equation} \quad & \ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\ \text{Trend equation} \quad & b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*) b_{t-1},\end{aligned}$$

where ℓ_t denotes an estimate of the level of the series at time t, b_t denotes an estimate of the trend (slope) of the time series at time t, α is the smoothing parameter for the level, $0 \le \alpha \le 1$, and β^* is the smoothing parameter for the trend, $0 \le \beta \le 1$.

As with simple exponential smooth (SES), the level equation here shows that ℓ_t is a weighted average of observation y_t and the one-step-ahead training forecast for time t, here given by $\ell_{t-1} + b_{t-1}$. The trend equation shows that b_t is weighted average of the estimated trend at time t based on $\ell_t - \ell_{t-1}$ and b_{t-1}, the previous estimate of the trend.

The forecast function is no longer flat but trending. The h-step-ahead forecast is equal to the last estimated level plus h times the last estimated trend value. Hence the forecasts are a linear function of h.

Example 7.0 - Simple Example of Holt's Linear Method Calculations

Below is a portion of a time series that Holt's Linear Method has been applied to. Suppose that after minimizing the SSE over the entire time series the optimal values for $\alpha, \beta_*, \ell_o, \text{and}\, b_o$ were as follows:

$$\alpha = 0.8,\ \beta_* = 0.2,\ \ell_o = 17.55,\ b_o = 4.31.$$

Let's use the table below to see how the calculations in Holt's Linear Method work.

Year	**t**	y_t	ℓ_t	b_t
1989	0		17.55	4.31
1990	1	17.55	18.41	3.62
1991	2	21.86	21.89	3.59
1992	3	23.89	24.21	3.33
1993	4	26.93	27.05	3.24

For $t=1$ **we have the following:**

$\hat{y}_1 = \ell_o + h(b_o) = 17.55 + 1(4.31) = 21.86$

Updating the level (ℓ_t) and slope (b_t) we have,

$\ell_1 = \alpha y_1 + (1-\alpha)(\ell_o + b_o) = .8(17.55) + (1-.80)(17.55+4.31) = 18.41$

$b_1 = \beta_*(\ell_1 - \ell_o) + (1-\beta_*)b_o = .2(18.41-17.55) + (1-.2)(4.31) = 3.62$

For $t=2$ **we have the following:**

$\hat{y}_1 = \ell_1 + h(b_1) = 18.41 + 1(3.62) = 22.03$

Updating the level (ℓ_t) and slope (b_t) we have,

$\ell_2 = \alpha y_2 + (1-\alpha)(\ell_1 + b_1) = .8(21.86) + (1-.8)(18.41+3.62) = 21.89$

$b_2 = \beta_*(\ell_2 - \ell_1) + (1-\beta_*)b_1 = .2(21.89-18.41) + (1-.2)3.62 = 3.59$

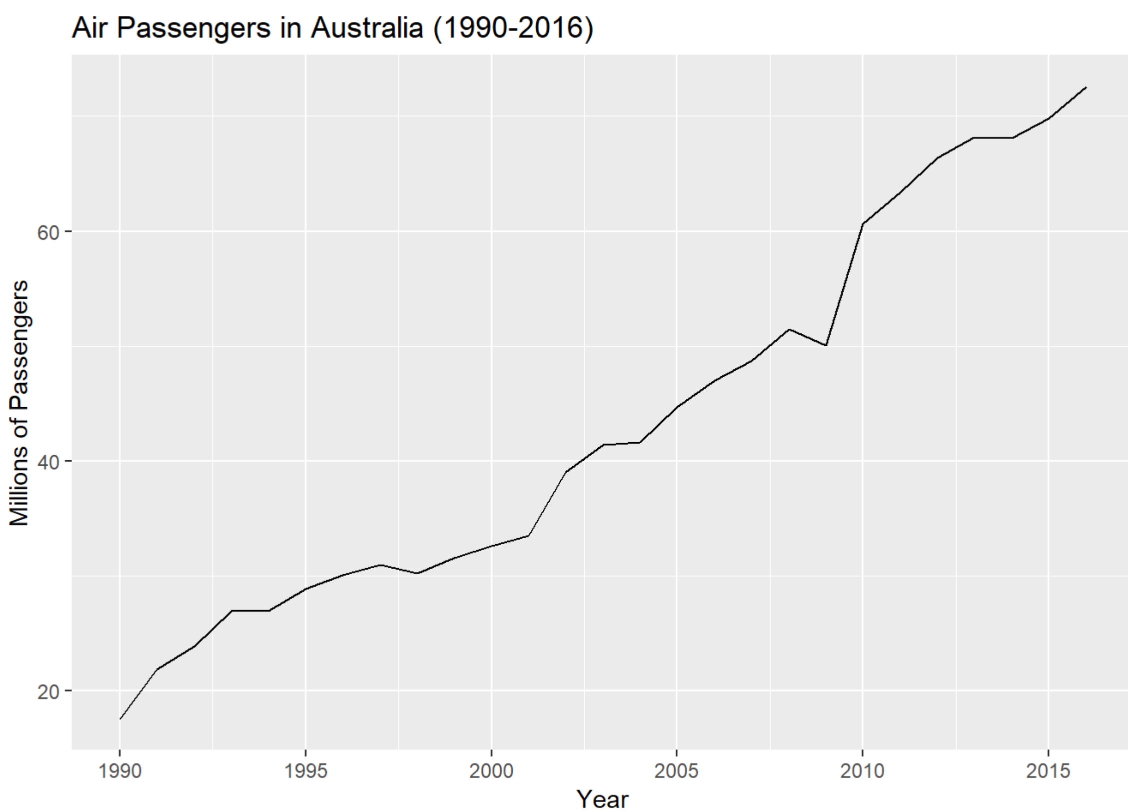
For $t=3$ **we have the following:**

etc…

Example 7.1 - Australian Airline Passengers (1990-2016)

We first we use `window` to create the subseries starting 1990.

```
require(fpp2)
## Loading required package: fpp2
## Loading required package: ggplot2
## Loading required package: forecast
## Loading required package: fma
## Loading required package: expsmooth
air = window(ausair,start=1990)
autoplot(air) + xlab("Year") + ylab("Millions of Passengers") + ggtitle
("Air Passengers in Australia (1990-2016)")
```

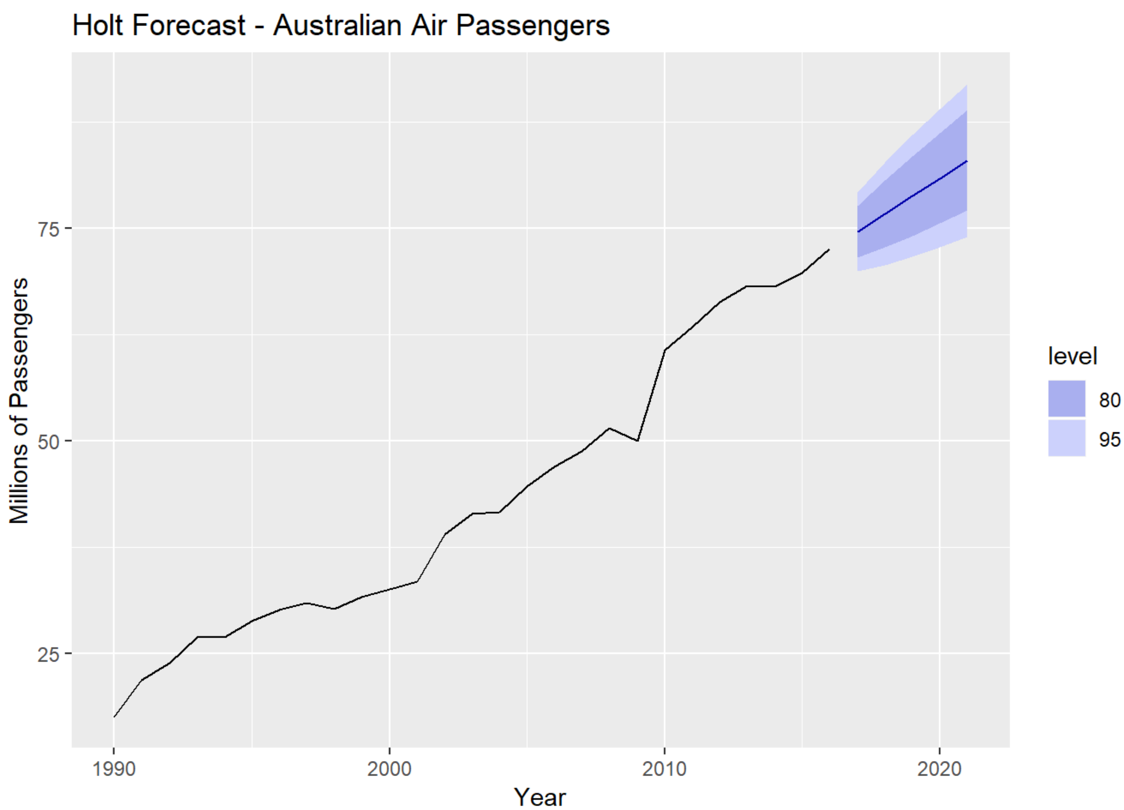


The table below demonstrates the application of Holt's method to these data. The smoothing parameters, αα and β_*β∗, and the initial values ℓ_0ℓo and $b_0$bo are estimated by minimizing the SSE for the one-step training errors as in SES in the previous section.

```
fc = holt(air,h=5)
summary(fc)
##
## Forecast method: Holt's method
##
## Model Information:
## Holt's method
##
## Call:
##  holt(y = air, h = 5)
##
##   Smoothing parameters:
##     alpha = 0.8302
##     beta  = 1e-04
##
##   Initial states:
##     l = 15.5715
##     b = 2.1017
##
##   sigma:  2.3645
##
##      AIC     AICc      BIC
## 141.1291 143.9863 147.6083
##
## Error measures:
##                       ME     RMSE     MAE        MPE     MAPE      M
ASE
## Training set 0.008359331 2.182343 1.52892 -0.3244107 3.820787 0.6654
839
##                     ACF1
## Training set -0.01335362
##
## Forecasts:
##      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2017       74.60130 71.57106 77.63154 69.96695 79.23566
```

```
## 2018       76.70304 72.76440 80.64169 70.67941 82.72668
## 2019       78.80478 74.13092 83.47864 71.65673 85.95284
## 2020       80.90652 75.59817 86.21487 72.78810 89.02494
## 2021       83.00826 77.13343 88.88310 74.02348 91.99305
```

```
autoplot(fc) + xlab("Year") + ylab("Millions of Passengers") + ggtitle
("Holt Forecast - Australian Air Passengers")
```



Example 7.2 - U.S. Domestic Auto Sales (1000's of cars sold)

```
Auto = read.csv(file="http://course1.winona.edu/bdeppa/FIN%20335/Datase
ts/Domestic%20Auto%20Sales%20(thousands%20of%20units%20-%202010%20to%20
present).csv")

AutoSales = ts(Auto$AutoSales,start=2010,frequency=12)

AutoSales
```

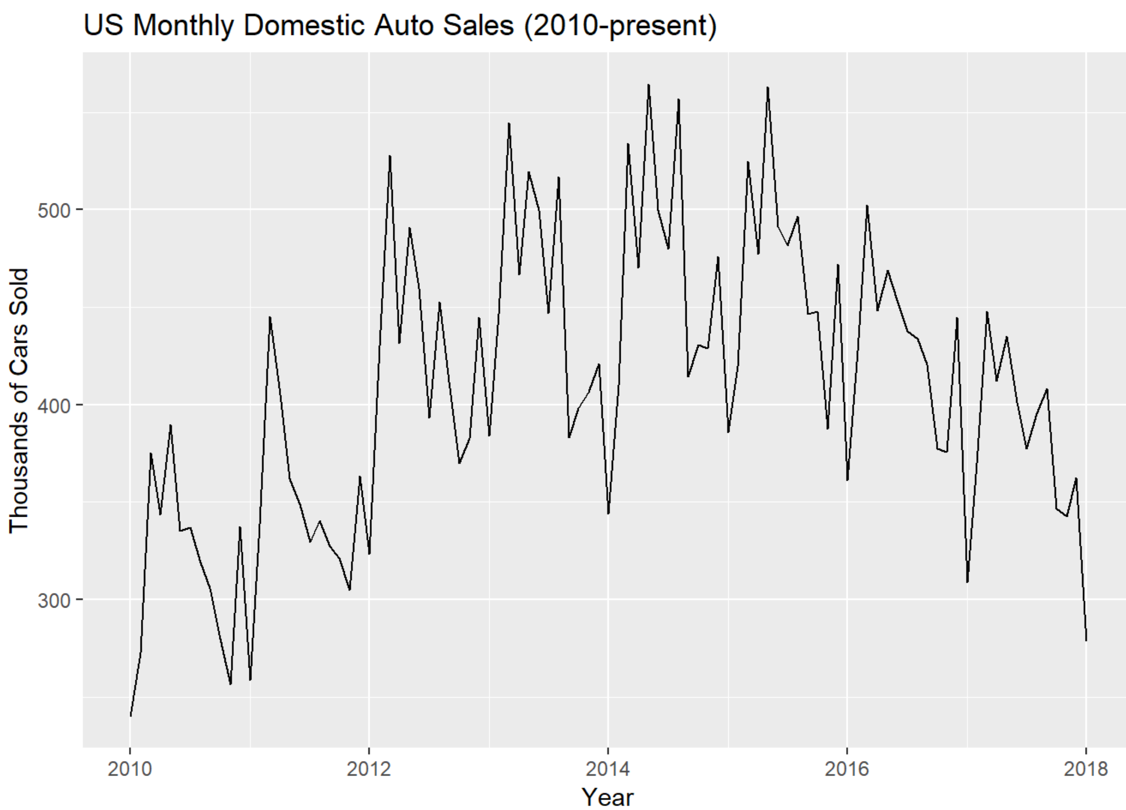
```
##         Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   N
ov

## 2010 240.0 272.7 375.2 343.5 389.7 335.3 336.8 319.1 305.6 279.7 25
6.3
```

```
## 2011 258.6 338.6 445.3 405.4 362.2 349.0 329.5 340.5 327.5 321.0 30
4.9
## 2012 323.4 430.2 528.1 431.8 491.1 459.9 393.3 452.8 412.0 369.8 38
2.7
## 2013 383.9 446.4 544.5 466.7 519.4 499.7 446.9 516.8 383.1 398.0 40
6.6
## 2014 344.2 409.9 534.3 470.2 564.6 499.8 480.0 556.9 414.5 430.9 42
8.8
## 2015 386.0 419.5 525.0 477.3 563.2 491.4 481.7 496.8 446.6 447.7 38
7.8
## 2016 361.4 427.8 502.6 448.3 468.8 453.4 437.7 433.8 420.5 377.6 37
5.7
## 2017 308.9 369.9 447.9 412.1 435.0 401.1 377.4 395.4 408.1 346.6 34
2.6
## 2018 278.4

##        Dec
## 2010 337.6
## 2011 363.5
## 2012 444.8
## 2013 421.2
## 2014 475.8
## 2015 472.1
## 2016 444.6
## 2017 362.7
## 2018
```

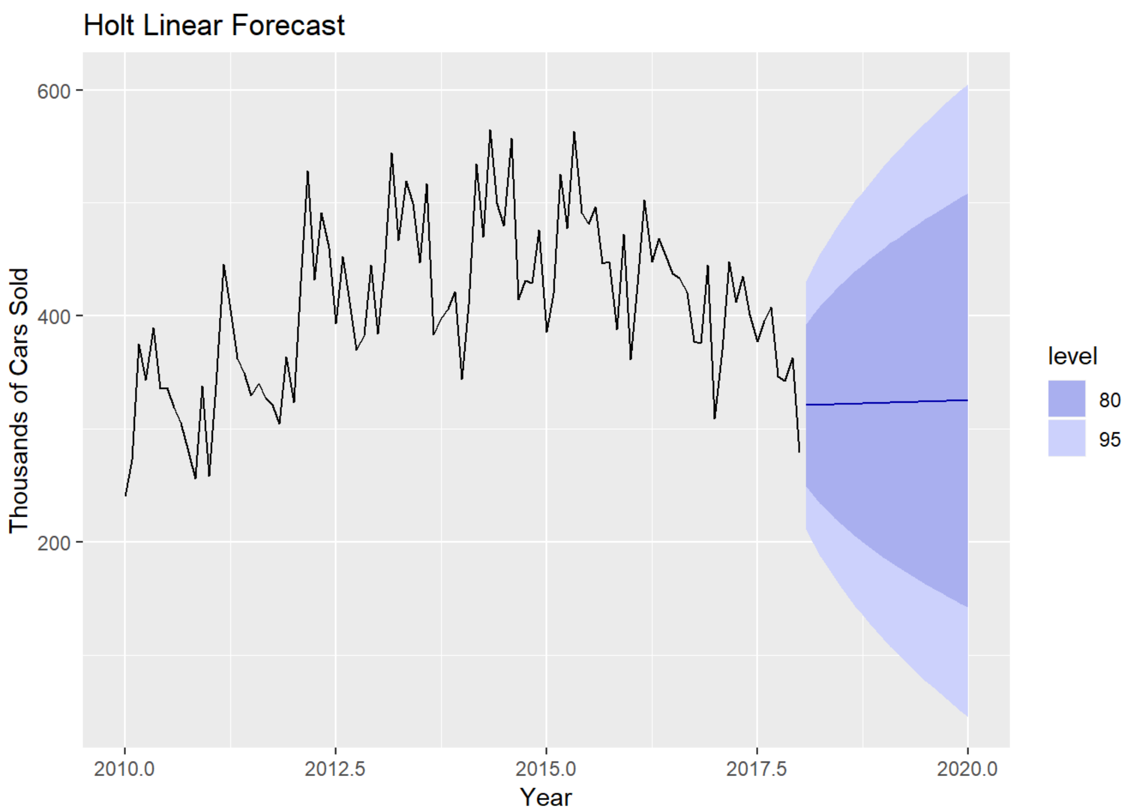
```
autoplot(AutoSales) + xlab("Year") + ylab("Thousands of Cars Sold") + g
gtitle("US Monthly Domestic Auto Sales (2010-present)")
```




```
Auto.holt = holt(AutoSales,h=24)
summary(Auto.holt)
##
## Forecast method: Holt's method
##
## Model Information:
## Holt's method
##
## Call:
##  holt(y = AutoSales, h = 24)
##
##   Smoothing parameters:
##     alpha = 0.4891
##     beta  = 1e-04
##
##   Initial states:
##     l = 310.436
```

```
##     b = 0.1967
##
##   sigma:  55.8839
##
##      AIC     AICc      BIC
## 1230.178 1230.837 1243.051
##
## Error measures:
##                       ME     RMSE      MAE       MPE     MAPE     MAS
E
## Training set -0.2228707 54.71951 42.60054 -1.420507 10.58087 1.05468
4
##                       ACF1
## Training set -7.313814e-05
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80     Lo 95    Hi 95
## Feb 2018       320.8998 249.2817 392.5178 211.36937 430.4302
## Mar 2018       321.0943 241.3646 400.8239 199.15832 443.0302
## Apr 2018       321.2888 234.1970 408.3806 188.09337 454.4842
## May 2018       321.4833 227.6023 415.3644 177.90462 465.0620
## Jun 2018       321.6778 221.4639 421.8917 168.41390 474.9418
## Jul 2018       321.8724 215.7003 428.0444 159.49624 484.2485
## Aug 2018       322.0669 210.2515 433.8822 151.06002 493.0737
## Sep 2018       322.2614 205.0720 439.4508 143.03566 501.4871
## Oct 2018       322.4559 200.1263 444.7855 135.36888 509.5429
## Nov 2018       322.6504 195.3860 449.9148 128.01634 517.2845
## Dec 2018       322.8449 190.8282 454.8617 120.94281 524.7471
## Jan 2019       323.0395 186.4338 459.6451 114.11916 531.9598
## Feb 2019       323.2340 182.1868 464.2811 107.52100 538.9470
## Mar 2019       323.4285 178.0738 468.7832 101.12766 545.7294
## Apr 2019       323.6230 174.0831 473.1630  94.92143 552.3246
## May 2019       323.8175 170.2047 477.4303  88.88702 558.7481
## Jun 2019       324.0121 166.4300 481.5941  83.01112 565.0130
## Jul 2019       324.2066 162.7513 485.6619  77.28202 571.1311
```

```
## Aug 2019       324.4011 159.1618 489.6404  71.68940 577.1128
## Sep 2019       324.5956 155.6556 493.5357  66.22410 582.9671
## Oct 2019       324.7901 152.2272 497.3531  60.87792 588.7023
## Nov 2019       324.9847 148.8719 501.0974  55.64350 594.3258
## Dec 2019       325.1792 145.5854 504.7729  50.51423 599.8441
## Jan 2020       325.3737 142.3637 508.3837  45.48410 605.2633
autoplot(Auto.holt) + xlab("Year") + ylab("Thousands of Cars Sold") + g
gtitle("Holt Linear Forecast")
```

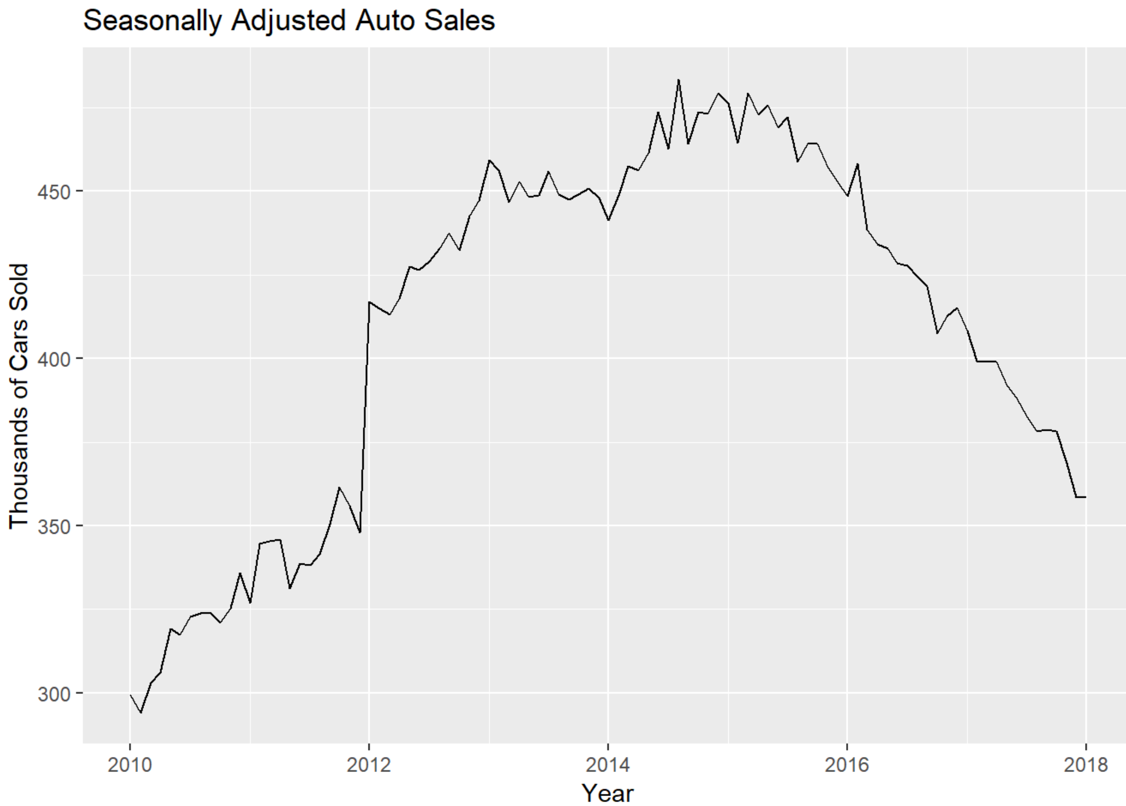


What is wrong with using this forecast for these data?

```
require(seasonal)
## Loading required package: seasonal
Auto.seats = seas(AutoSales)
summary(Auto.seats)
##
## Call:
## seas(x = AutoSales)
##
```

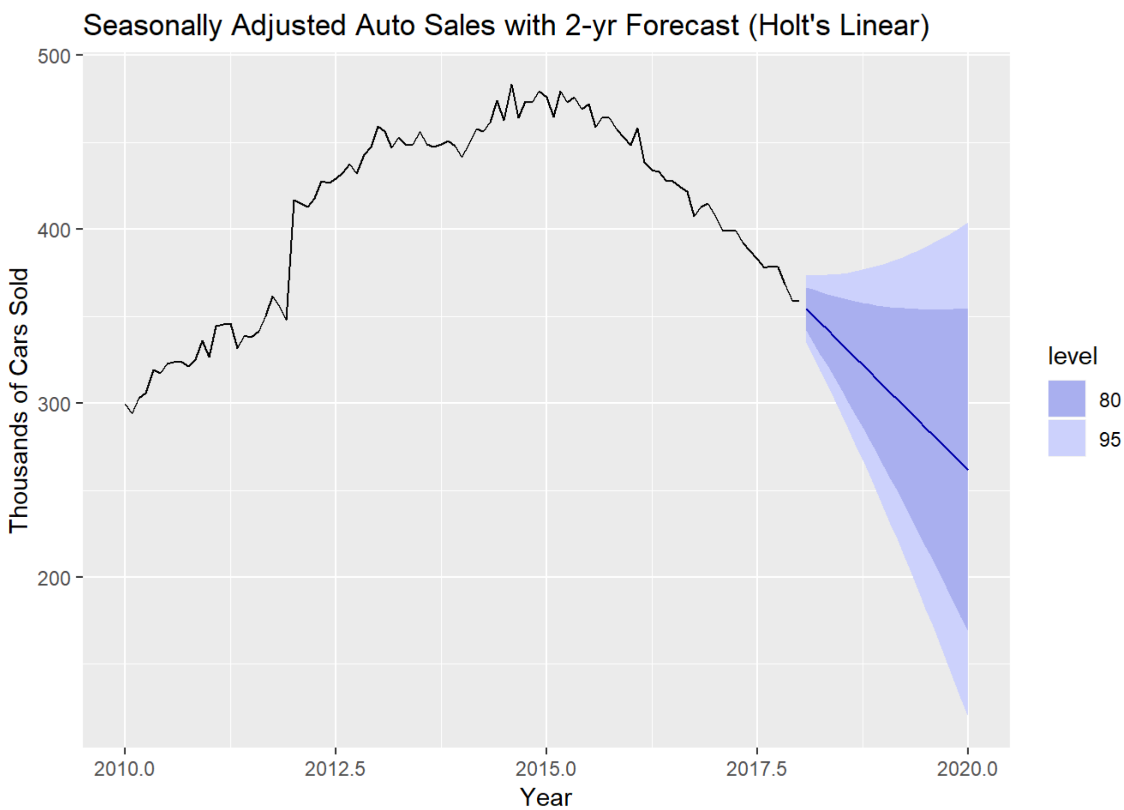
```
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## Constant          -0.0031601  0.0028185  -1.121    0.262
## Mon               -0.0312666  0.0070981  -4.405 1.06e-05 ***
## Tue               -0.0008744  0.0072099  -0.121    0.903
## Wed                0.0057486  0.0069613   0.826    0.409
## Thu                0.0060996  0.0068437   0.891    0.373
## Fri               -0.0077464  0.0072794  -1.064    0.287
## Sat                0.0502267  0.0073704   6.815 9.45e-12 ***
## LS2012.Jan         0.1501612  0.0312908   4.799 1.60e-06 ***
## MA-Nonseasonal-01  0.4838507  0.0959435   5.043 4.58e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## SEATS adj.  ARIMA: (0 1 1)(0 1 0)  Obs.: 97  Transform: log
## AICc: 771.3, BIC: 792.6  QS (no seasonality in final):    0
## Box-Ljung (no autocorr.): 22.98   Shapiro (normality): 0.9834
```

```r
Auto.SA = seasadj(Auto.seats)
Auto.seas = seasonal(Auto.seats)
autoplot(Auto.SA) + xlab("Year") + ylab("Thousands of Cars Sold") + ggtitle("Seasonally Adjusted Auto Sales")
```




```
AutoSA.holt = holt(Auto.SA,h=24)

autoplot(AutoSA.holt) + xlab("Year") + ylab("Thousands of Cars Sold") +
 ggtitle("Seasonally Adjusted Auto Sales with 2-yr Forecast (Holt's Lin
ear)")
```

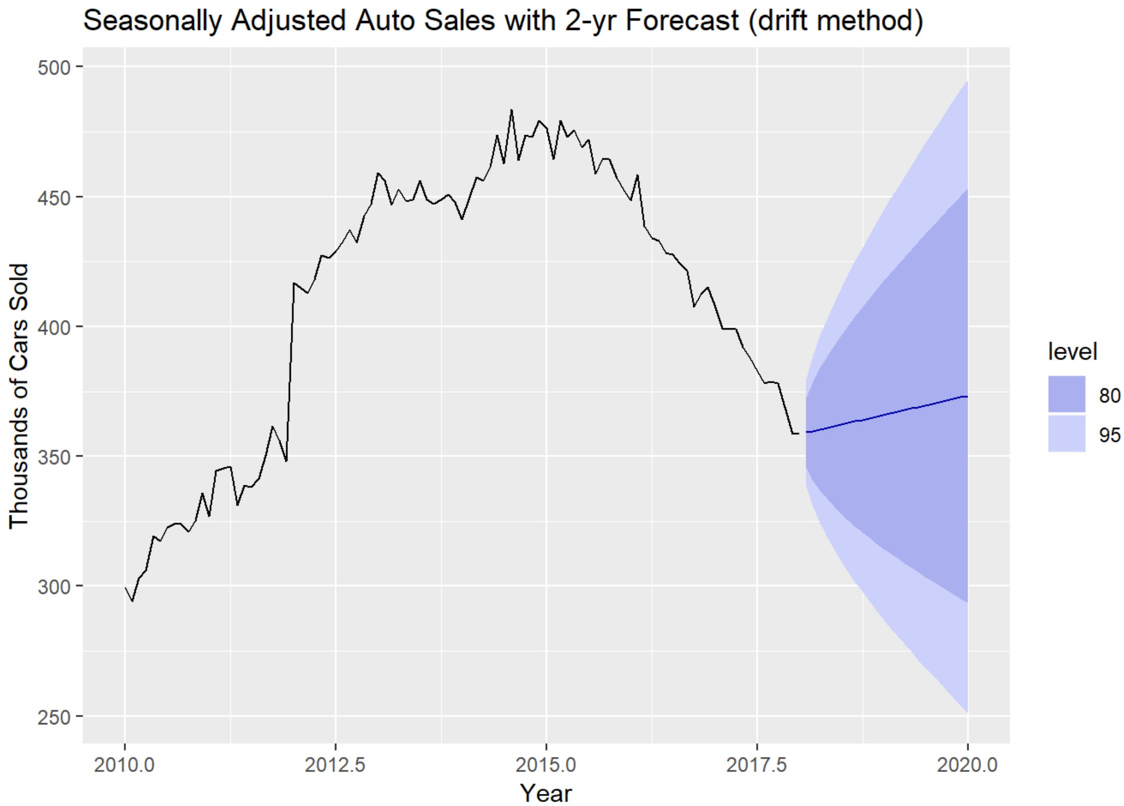


As with the best linear model for these data (*see Assignment 4*), the seasonally adjusted forecast for the general trend is gloomy for domestic auto sales through 2020.

How does Holt's linear method compare to the **drift method** covered in Chapter 2 for the seasonally adjusted auto sales?

```
AutoSA.drift = rwf(Auto.SA,drift=T,h=24)

autoplot(AutoSA.drift) + xlab("Year") + ylab("Thousands of Cars Sold")
+ ggtitle("Seasonally Adjusted Auto Sales with 2-yr Forecast (drift met
hod)")
```



Clearly not adjusting the slope dynamically leads to a very different forecast! I am sure the U.S. automakers would like to believe this is the seasonally adjusted trend through 2020.

Even we will not do it, it should clear that if we used the last year of the seasonally adjusted auto sales as a test set, Holt's linear method would yield much more accurate forecasts.

Rather than forecast the seasonally adjusted auto sales, can we use the exponential smoothing idea to model the seasonal time series directly? Adding a seasonal component estimation step to Holt's linear method would take care of this. This is precisely what Holt-Winter's seasonal method does.

7.3 - Holt-Winter's Seasonal Method

Holt (1957) and Winters (1960) extended Holt's method to capture seasonality. The Holt-Winters seasonal method comprises the forecast equation and three smoothing equations - one for the level (ℓ_t)(ℓt), one for trend or slope (b_t)(bt), and one for the seasonal component (s_t)(st), with corresponding smoothing parameters αα, β_*β∗, and γγ. We mm to denote the frequency of the seasonality, i.e., the number of seasons in a year. For example, for quarterly data
$m=4$m=4, and for monthly data $m=12$m=12 as usual.
There are two variations to this method that differ in the nature of the seasonal component. The additive method is preferred when the seasonal variations are roughly constant through the series, while the multiplicative method is preferred when the seasonal variations are changing proportional to the level of the series, i.e. the seasonal swings are increasing in magnitude over time. With the additive method, the seasonal component is expressed in absolute terms in the scale of the observed series, and in the level equation the series is

seasonally adjusted by subtracting the seasonal component. Within each year, the seasonal component will add up to approximately zero. With the multiplicative method, the seasonal component is expressed in relative terms (percentages), and the series is seasonally adjusted by dividing through by the seasonal component. Within each year, the seasonal component will sum up to approximately mm.

Holt-Winter's Additive Seasonal Method $(y_t=T_t+S_T+R_t)$(yt=Tt+ST+Rt)

The component form for the additive method is:

$$\hat{y}_{t+h|t}\ \ell_t\ b_t\ s_t = \ell_t+hb_t+s_{t-m+h_m^+} = \alpha(y_t-s_{t-m})+(1-\alpha)(\ell_{t-1}+b_{t-1}) = \beta^*(\ell_t-\ell_{t-1})+(1-\beta^*)b_{t-1} = \gamma(y_t-\ell_{t-1}-b_{t-1})+(1-\gamma)s_{t-m},$$

$$\hat{y}_{t+h|t}=\ell_t+hb_t+s_{t-m+h_m^+}\quad \ell_t=\alpha(y_t-s_{t-m})+(1-\alpha)(\ell_{t-1}+b_{t-1})\quad b_t=\beta^*(\ell_t-\ell_{t-1})+(1-\beta^*)b_{t-1}\quad s_t=\gamma(y_t-\ell_{t-1}-b_{t-1})+(1-\gamma)s_{t-m},$$

as $h_m^+=\lfloor(h-1)/m\rfloor+1$ $h_m^+=\lfloor(h-1)/m\rfloor+1$, which ensures that the estimates of the seasonal indices used for forecasting come from the final year of the sample.

As before the $\hat{y}_t$ $\hat{y}_t$ is a equal to level plus trend, but this time we also add in a seasonal level. The level equation shows that level is estimated by adding taking a weighted average of the seasonally adjusted time series value at time t t (y_t-s_{t-m}) (y_t-s_{t-m}) and the previous level plus trend $(\ell_{t-1}+b_{t-1})$ $(\ell_{t-1}+b_{t-1})$. The trend equation shows that b_t b_t is estimated by taking a weighted average of the slope between to the two most recent level estimates $(\ell_t-\ell_{t-1})$ $(\ell_t-\ell_{t-1})$ and the previous slope (b_{t-1}) (b_{t-1}), and finally the seasonal equation shows that the level of the seasonal component (s_t) (s_t) is estimated by taking a weighted average of the current seasonal component estimate $(y_t-\ell_{t-1}-b_{t-1})$ $(y_t-\ell_{t-1}-b_{t-1})$ and the previous seasonal component estimate (s_{t-m}) (s_{t-m}), i.e. the seasonal index of the same season last year (i.e., m m time periods ago). Thus the level, trend, and seasonal components are all estimated/updated via exponential smoothing.

The equation for the seasonal component is often expressed as

$$s_t=\gamma^*(y_t-\ell_t)+(1-\gamma^*)s_{t-m}. \quad s_t=\gamma^*(y_t-\ell_t)+(1-\gamma^*)s_{t-m}.$$

If we substitute ℓ_t ℓ_t from the smoothing equation for the level of the component form above, we get

$$s_t=\gamma^*(1-\alpha)(y_t-\ell_{t-1}-b_{t-1})+[1-\gamma^*(1-\alpha)]s_{t-m}, \quad s_t=\gamma^*(1-\alpha)(y_t-\ell_{t-1}-b_{t-1})+[1-\gamma^*(1-\alpha)]s_{t-m},$$

which is identical to the smoothing equation for the seasonal component we specify here, with

$\gamma=\gamma^*(1-\alpha).$ $\gamma=\gamma^*(1-\alpha)$. The usual parameter restriction is $0\le\gamma^*\le1$ $0\le\gamma^*\le1$, which translates to $0\le\gamma\le(1-\alpha)$ $0\le\gamma\le(1-\alpha)$.

Holt's Multiplicative Seasonal Model $(y_t=T_t\times S_t\times R_t)$(yt=Tt×St×Rt)

The component form for the multiplicative method is:

$$\hat{y}_{t+h|t}\ \ell_t\ b_t\ s_t=(\ell_t+hb_t)s_{t-m+h_m^+}=\alpha\frac{y_t}{s_{t-m}}+(1-\alpha)(\ell_{t-1}+b_{t-1})=\beta^*(\ell_t-\ell_{t-1})+(1-\beta^*)b_{t-1}=\gamma\frac{y_t}{(\ell_{t-1}+b_{t-1})}+(1-\gamma)s_{t-m}$$

$$\hat{y}_{t+h|t}=(\ell_t+hb_t)s_{t-m+h_m^+}\quad \ell_t=\alpha\frac{y_t}{s_{t-m}}+(1-\alpha)(\ell_{t-1}+b_{t-1})\quad b_t=\beta^*(\ell_t-\ell_{t-1})+(1-\beta^*)b_{t-1}\quad s_t=\gamma\frac{y_t}{(\ell_{t-1}+b_{t-1})}+(1-\gamma)s_{t-m}$$

The main difference is the adjustments are made via division vs. subtraction, which is clear if you carefully examine the formulae above. Again estimates of the level, slope (trend), and seasonality at time tt are found using exponential smoothing.

Example 7.3 - U.S. Monthly Clothing Sales (in millions, 1992-present)

n this example we employ the Holt-Winters method with both additive and multiplicative seasonality to forecast monthly clothing sales in U.S. in millions of dollars from 1992 - present. In order to compare the performance of these two methods we will use the last $h=24$h=24 months of this time series as a test set and fit both additive and multiplicative Holt-Winter's models to the training set.

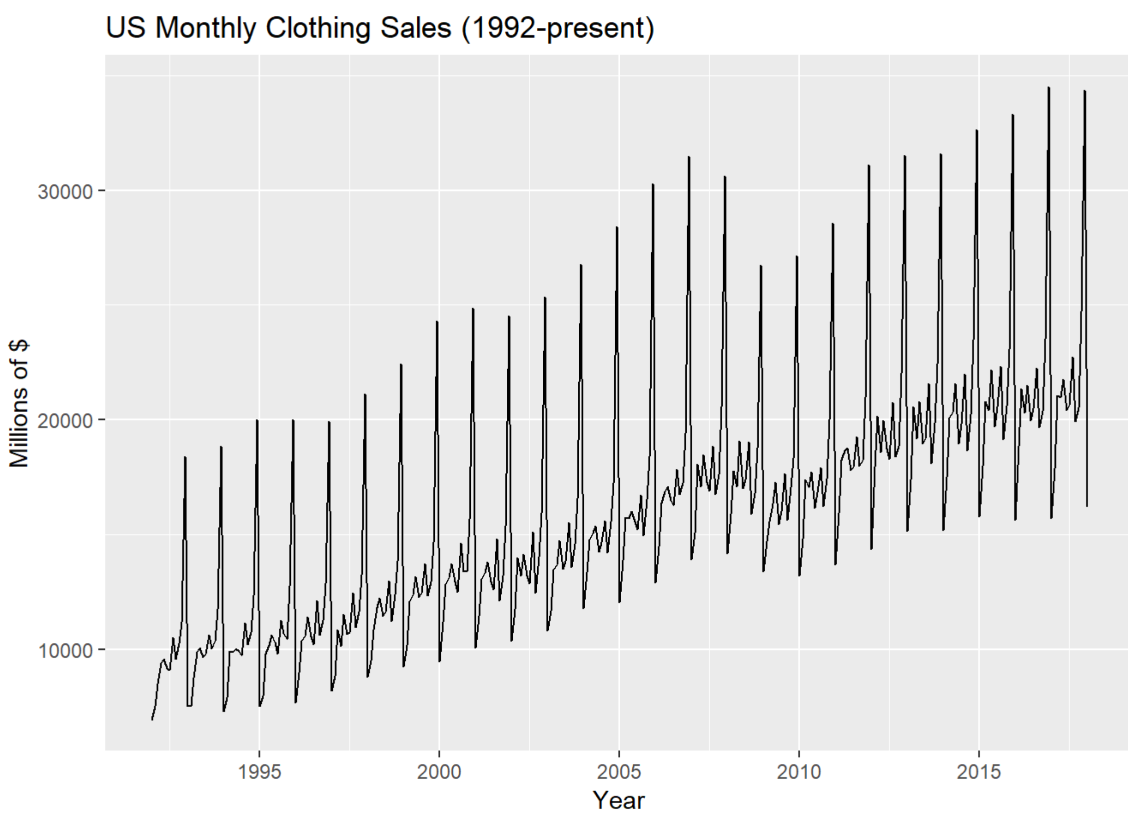
```
Cloth = read.csv(file="http://course1.winona.edu/bdeppa/FIN%20335/Datas
ets/US%20Clothing%20Sales%20(millions%20of%20dollars%20-%201992%20to%20
present).csv")

names(Cloth)

## [1] "DATE"      "Clothing"

ClothSales = ts(Cloth$Clothing,start=1992,frequency=12)

autoplot(ClothSales) + xlab("Year") + ylab("Millions of $") + ggtitle("
US Monthly Clothing Sales (1992-present)")
```



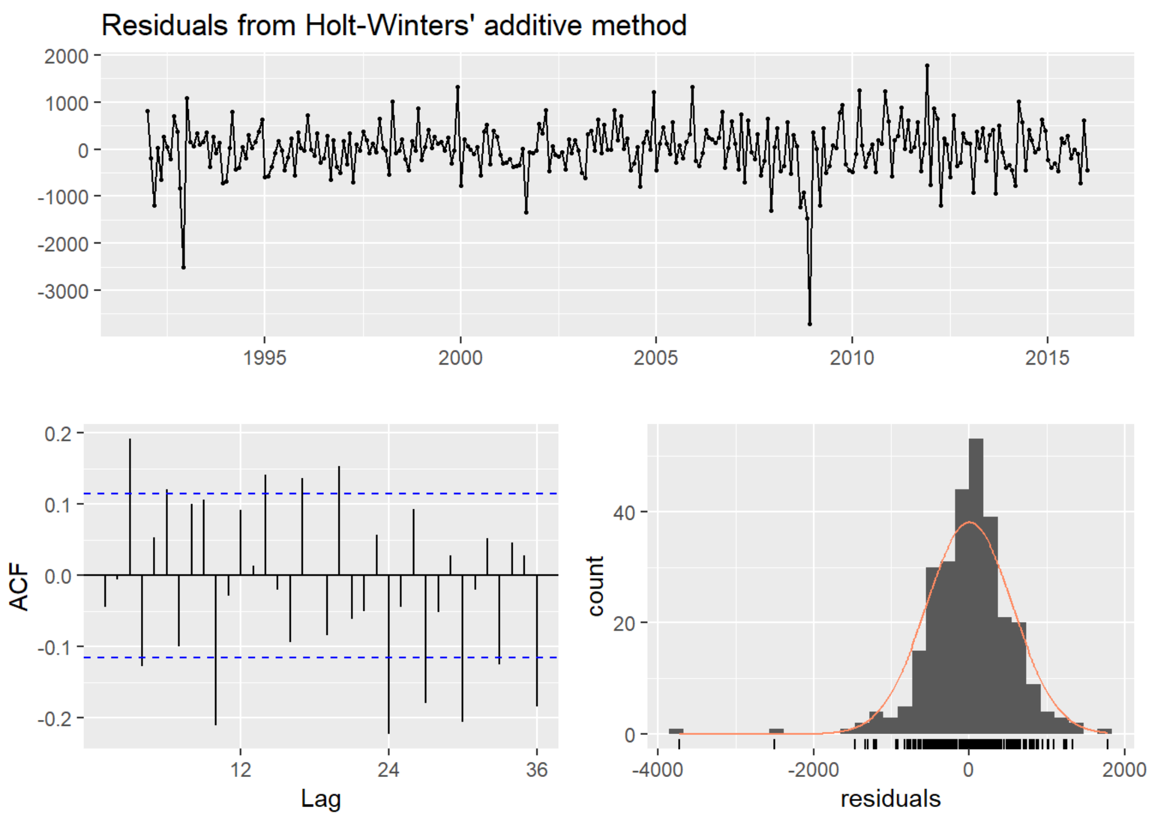

```
Cloth.test = tail(ClothSales,24)
```

```r
Cloth.train = head(ClothSales,289)
hw.linear = hw(Cloth.train,seasonal="additive",h=24)
summary(hw.linear)
```

```
##
## Forecast method: Holt-Winters' additive method
##
## Model Information:
## Holt-Winters' additive method
##
## Call:
##  hw(y = Cloth.train, h = 24, seasonal = "additive")
##
##   Smoothing parameters:
##     alpha = 0.2218
##     beta  = 1e-04
##     gamma = 0.6918
##
##   Initial states:
##     l = 10124.3241
##     b = 39.281
##     s = 10490.53 1465.19 -552.2822 -1360.9 483.8566 -1145.296
##            -1185.023 -3.5062 -784.7891 -712.0521 -2659.721 -4036.011
##
##   sigma:  570.6937
##
##      AIC     AICc      BIC
## 5323.618 5325.877 5385.947
##
## Error measures:
##                      ME     RMSE      MAE       MPE     MAPE       MA
SE
## Training set -0.1916868 554.6711 389.1126 -0.107157 2.600867 0.56826
93
##                     ACF1
## Training set -0.04434422
```

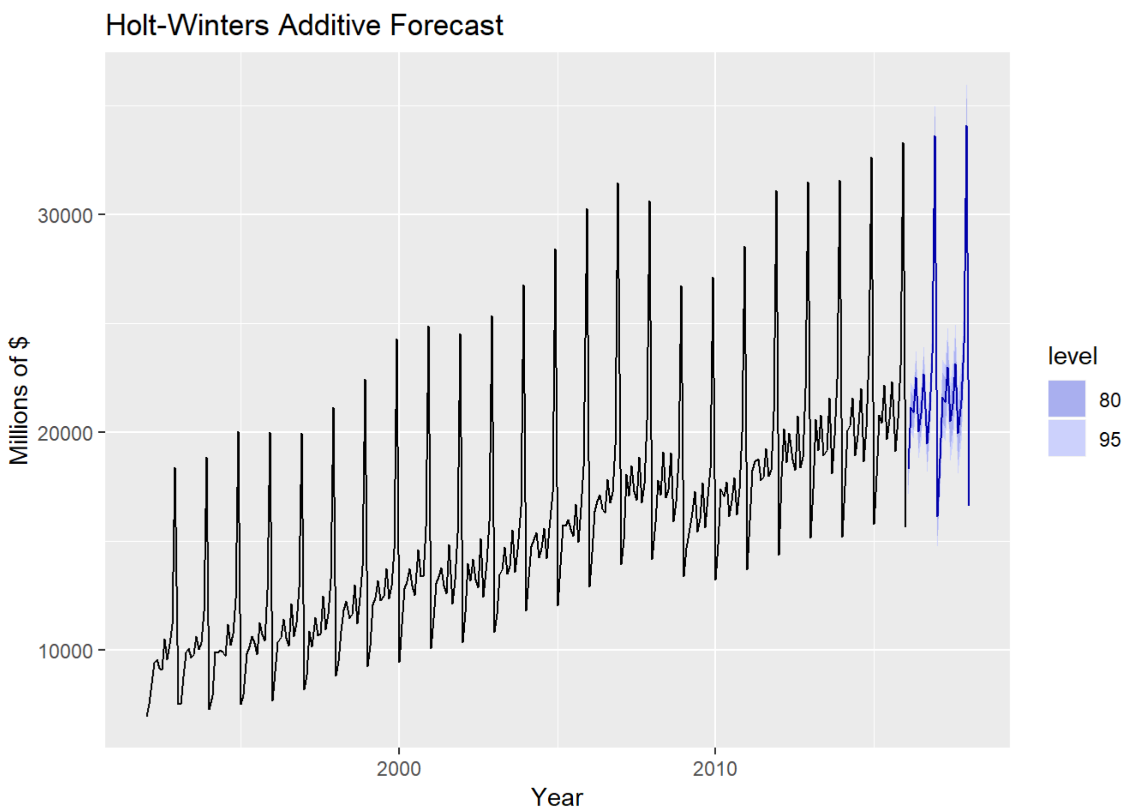
```
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Feb 2016       18292.65 17561.28 19024.02 17174.11 19411.19
## Mar 2016       21130.81 20381.64 21879.97 19985.06 22276.56
## Apr 2016       20884.21 20117.65 21650.77 19711.86 22056.56
## May 2016       22530.07 21746.49 23313.65 21331.68 23728.46
## Jun 2016       20026.57 19226.31 20826.83 18802.67 21250.46
## Jul 2016       20841.27 20024.66 21657.88 19592.37 22090.17
## Aug 2016       22659.68 21827.03 23492.34 21386.25 23933.12
## Sep 2016       19475.49 18627.08 20323.90 18177.96 20773.02
## Oct 2016       20975.56 20111.66 21839.45 19654.35 22296.76
## Nov 2016       23627.99 22748.88 24507.10 22283.50 24972.48
## Dec 2016       33628.02 32733.93 34522.11 32260.63 34995.41
## Jan 2017       16136.38 15227.55 17045.22 14746.45 17526.32
## Feb 2017       18763.96 17635.39 19892.52 17037.97 20489.95
## Mar 2017       21602.11 20461.81 22742.41 19858.17 23346.05
## Apr 2017       21355.52 20203.59 22507.44 19593.80 23117.24
## May 2017       23001.37 21837.93 24164.82 21222.03 24780.72
## Jun 2017       20497.87 19323.01 21672.74 18701.07 22294.68
## Jul 2017       21312.57 20126.39 22498.76 19498.46 23126.69
## Aug 2017       23130.99 21933.58 24328.40 21299.71 24962.26
## Sep 2017       19946.79 18738.26 21155.33 18098.50 21795.09
## Oct 2017       21446.86 20227.29 22666.43 19581.69 23312.03
## Nov 2017       24099.30 22868.78 25329.81 22217.39 25981.21
## Dec 2017       34099.33 32857.95 35340.70 32200.81 35997.85
## Jan 2018       16607.69 15355.54 17859.84 14692.69 18522.69
```

```
checkresiduals(hw.linear)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Holt-Winters' additive method
## Q* = 89.881, df = 8, p-value = 4.441e-16
##
## Model df: 16.   Total lags used: 24
```

```
autoplot(hw.linear) + xlab("Year") + ylab("Millions of $") + ggtitle("H
olt-Winters Additive Forecast")
```

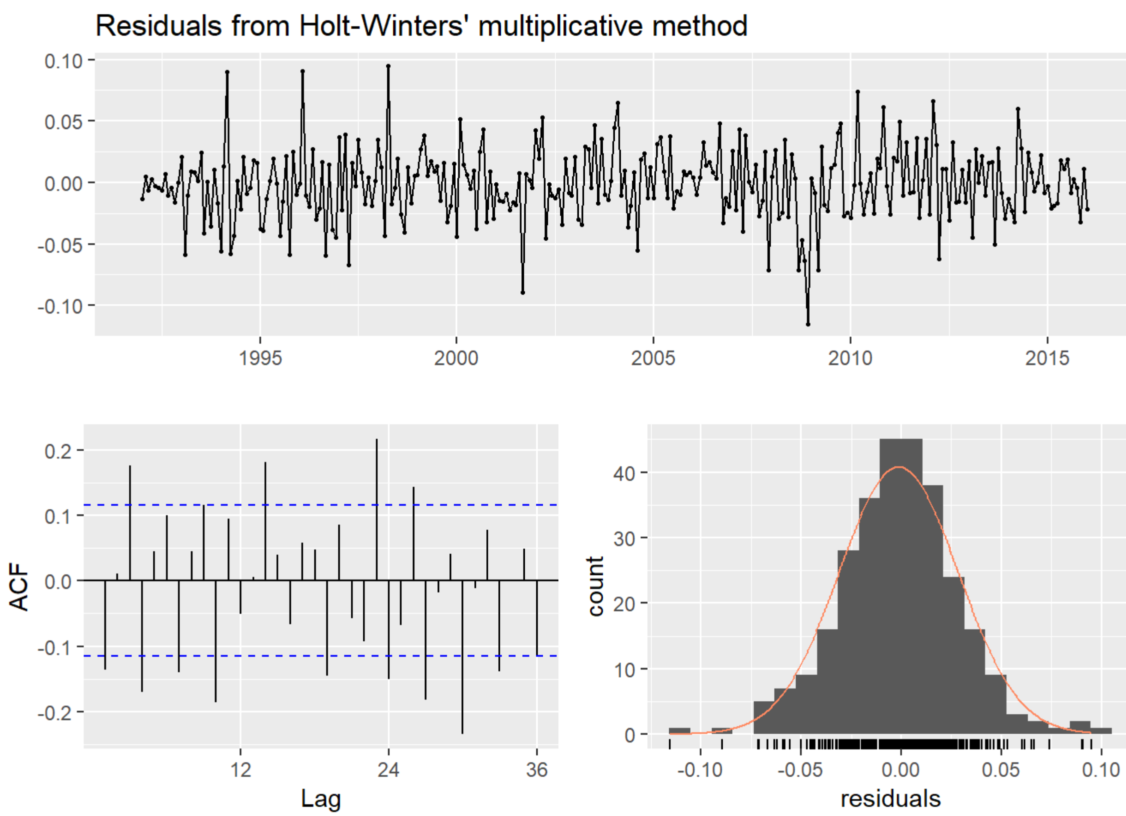



```
hw.mult = hw(Cloth.train,seasonal="multiplicative",h=24)
summary(hw.mult)
##
## Forecast method: Holt-Winters' multiplicative method
##
## Model Information:
## Holt-Winters' multiplicative method
##
## Call:
##  hw(y = Cloth.train, h = 24, seasonal = "multiplicative")
##
##   Smoothing parameters:
##     alpha = 0.3113
##     beta  = 0.0022
##     gamma = 0.6003
##
##   Initial states:
```

```
##     l = 9712.5503
##     b = 57.0034
##     s = 1.7978 1.1117 1.011 0.9516 1.0351 0.9117
##            0.9215 0.9622 0.9477 0.8649 0.7652 0.7197
##
##   sigma:  0.0306
##
##      AIC     AICc      BIC
## 5187.786 5190.044 5250.115
##
## Error measures:
##                     ME     RMSE      MAE        MPE     MAPE      MA
SE
## Training set -36.16646 496.9546 353.4097 -0.2272557 2.299883 0.51612
79
##                     ACF1
## Training set -0.07603236
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Feb 2016       18250.27 17534.50 18966.03 17155.60 19344.93
## Mar 2016       21186.53 20315.87 22057.19 19854.97 22518.10
## Apr 2016       20920.18 20022.58 21817.79 19547.41 22292.95
## May 2016       22525.02 21519.04 23531.00 20986.51 24063.53
## Jun 2016       19941.23 19016.66 20865.79 18527.22 21355.23
## Jul 2016       20707.76 19713.29 21702.22 19186.86 22228.66
## Aug 2016       22566.01 21445.78 23686.24 20852.76 24279.26
## Sep 2016       19324.81 18334.90 20314.72 17810.88 20838.74
## Oct 2016       20849.61 19749.30 21949.93 19166.83 22532.40
## Nov 2016       23613.99 22331.93 24896.05 21653.24 25574.73
## Dec 2016       33852.69 31964.28 35741.11 30964.61 36740.77
## Jan 2017       16039.38 15121.15 16957.62 14635.07 17443.70
## Feb 2017       18695.63 17426.44 19964.82 16754.57 20636.69
## Mar 2017       21702.51 20202.46 23202.57 19408.38 23996.65
## Apr 2017       21428.66 19921.37 22935.95 19123.46 23733.86
```

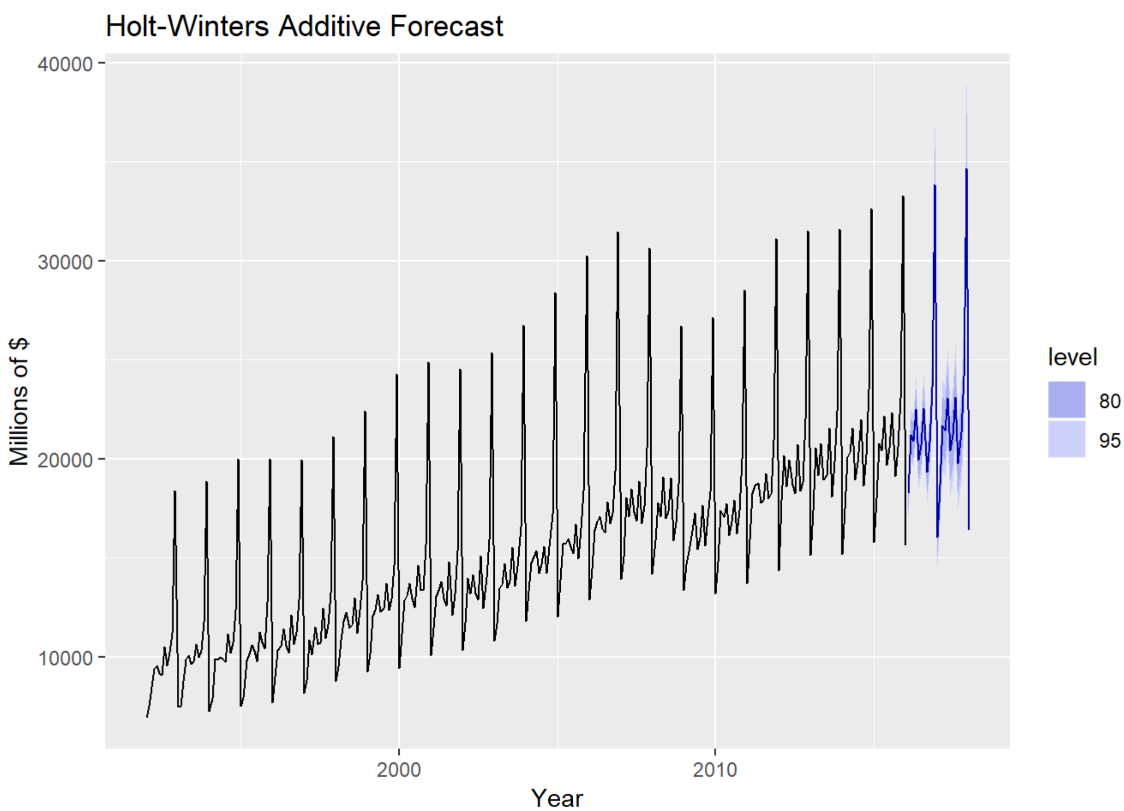
```
## May 2017       23071.41 21420.63 24722.19 20546.76 25596.06
## Jun 2017       20423.98 18938.10 21909.87 18151.52 22696.45
## Jul 2017       21208.08 19639.87 22776.29 18809.71 23606.45
## Aug 2017       23110.15 21373.95 24846.35 20454.86 25765.44
## Sep 2017       19789.87 18279.87 21299.88 17480.52 22099.23
## Oct 2017       21350.38 19696.39 23004.38 18820.82 23879.95
## Nov 2017       24180.04 22278.80 26081.27 21272.35 27087.73
## Dec 2017       34662.58 31897.19 37427.98 30433.28 38891.89
## Jan 2018       16422.36 15093.37 17751.35 14389.85 18454.87
```

```
checkresiduals(hw.mult)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Holt-Winters' multiplicative method
## Q* = 99.313, df = 8, p-value < 2.2e-16
##
## Model df: 16.   Total lags used: 24
```

```
autoplot(hw.mult) + xlab("Year") + ylab("Millions of $") + ggtitle("Hol
t-Winters Additive Forecast")
```




```
accuracy(hw.linear,Cloth.test)
```

```
##                        ME     RMSE      MAE       MPE     MAPE      
MASE
## Training set   -0.1916868 554.6711 389.1126 -0.107157 2.600867 0.568
2693
## Test set     -267.0027695 588.4365 492.7033 -1.441568 2.327744 0.719
5556
##                     ACF1 Theil's U
## Training set -0.04434422        NA
## Test set     -0.01796978 0.1307722
```

```
accuracy(hw.mult,Cloth.test)
```

```
##                      ME     RMSE      MAE        MPE     MAPE      M
ASE
## Training set  -36.16646 496.9546 353.4097 -0.2272557 2.299883 0.5161
279
## Test set     -256.58984 558.4255 461.6377 -1.3018875 2.181082 0.6741
865
```

```
##                       ACF1 Theil's U
## Training set -0.07603236        NA
## Test set     -0.05962277 0.1244884
```

Which method is more accurate, Holt-Winter’s addtive or Holt-Winter’s multiplicative? Explain.

Example 7.2 - U.S. Domestic Auto Sales (cont’d)

```
AutoSales
```

```
##        Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   N
ov
## 2010 240.0 272.7 375.2 343.5 389.7 335.3 336.8 319.1 305.6 279.7 25
6.3
## 2011 258.6 338.6 445.3 405.4 362.2 349.0 329.5 340.5 327.5 321.0 30
4.9
## 2012 323.4 430.2 528.1 431.8 491.1 459.9 393.3 452.8 412.0 369.8 38
2.7
## 2013 383.9 446.4 544.5 466.7 519.4 499.7 446.9 516.8 383.1 398.0 40
6.6
## 2014 344.2 409.9 534.3 470.2 564.6 499.8 480.0 556.9 414.5 430.9 42
8.8
## 2015 386.0 419.5 525.0 477.3 563.2 491.4 481.7 496.8 446.6 447.7 38
7.8
## 2016 361.4 427.8 502.6 448.3 468.8 453.4 437.7 433.8 420.5 377.6 37
5.7
## 2017 308.9 369.9 447.9 412.1 435.0 401.1 377.4 395.4 408.1 346.6 34
2.6
## 2018 278.4

##        Dec
## 2010 337.6
## 2011 363.5
## 2012 444.8
## 2013 421.2
## 2014 475.8
## 2015 472.1
## 2016 444.6
## 2017 362.7
## 2018
```

```
length(AutoSales)
```

```
## [1] 97
```

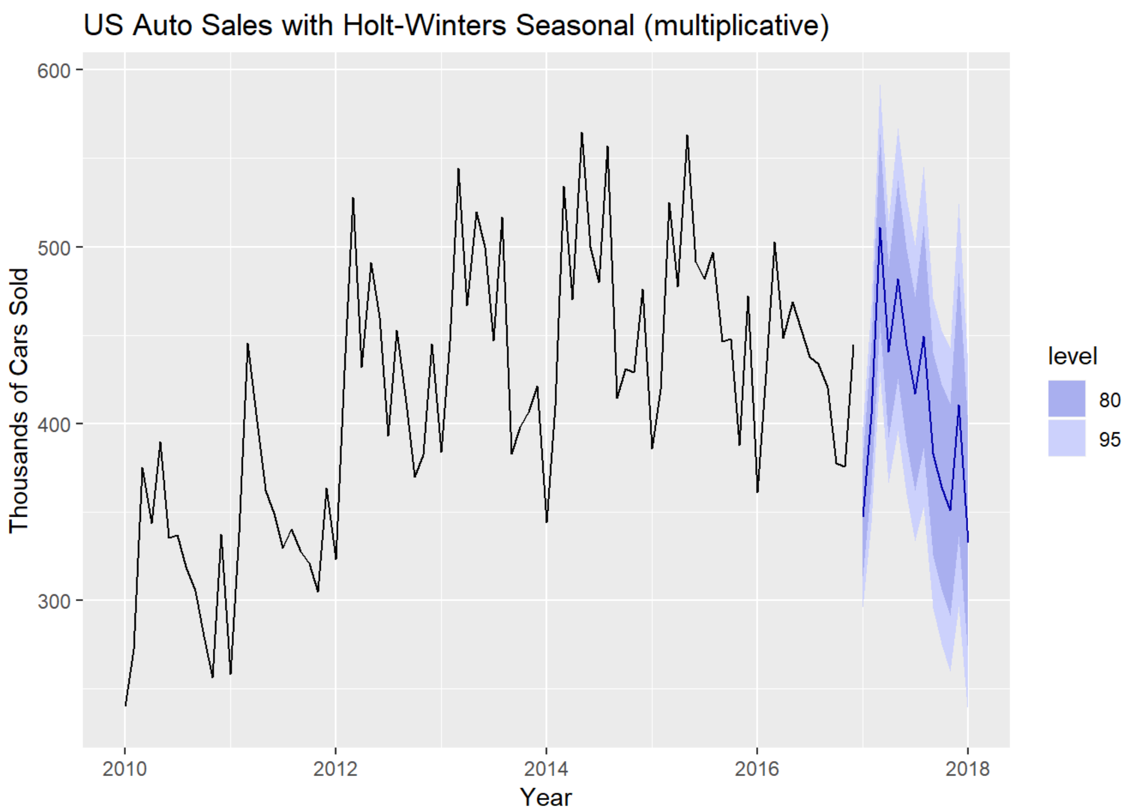
```
Auto.test = tail(AutoSales,13)
Auto.train = head(AutoSales,84)
hw.add = hw(Auto.train,seasonal="additive",h=13)
hw.mult = hw(Auto.train,seasonal="multiplicative",h=13)
accuracy(hw.add,Auto.test)
```

```
##                      ME     RMSE      MAE         MPE      MAPE
 MASE
## Training set  -1.621898 25.45453 20.15591  -0.5917106  5.072157 0.50
77232
## Test set     -72.302735 75.91779 72.30273 -19.7235043 19.723504 1.82
12913
##                     ACF1 Theil's U
## Training set -0.03699112        NA
## Test set     -0.07098146  1.664592
```

```
accuracy(hw.mult,Auto.test)
```

```
##                     ME     RMSE      MAE        MPE      MAPE      M
ASE
## Training set  -1.69851 27.39812 22.23781 -0.6868033  5.488451 0.5601
658
## Test set     -34.69418 41.30596 38.53013 -9.4245447 10.364500 0.9705
662
##                   ACF1 Theil's U
## Training set 0.2798302        NA
## Test set     0.0366181  0.888674
```

```
autoplot(hw.mult) + xlab("Year") + ylab("Thousands of Cars Sold") + ggt
itle("US Auto Sales with Holt-Winters Seasonal (multiplicative)")
```



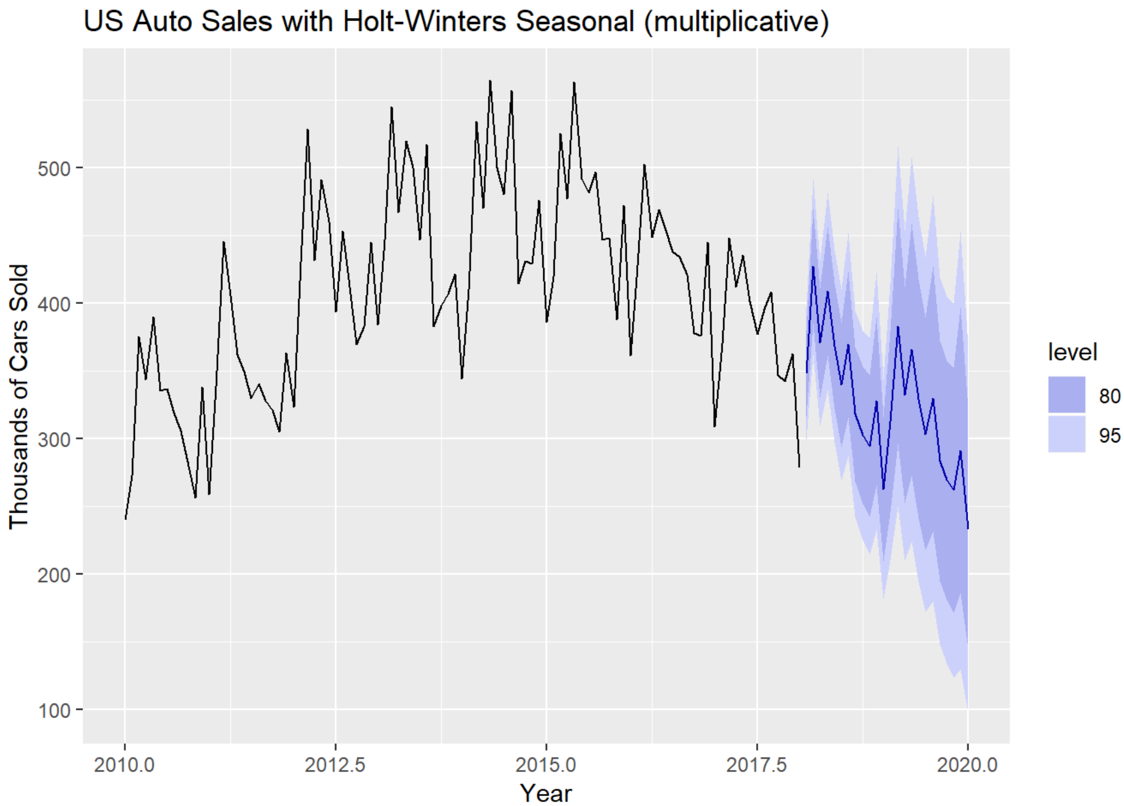

```
summary(hw.mult)
##
## Forecast method: Holt-Winters' multiplicative method
##
## Model Information:
## Holt-Winters' multiplicative method
##
## Call:
##  hw(y = Auto.train, h = 13, seasonal = "multiplicative")
##
##   Smoothing parameters:
##     alpha = 0.2497
##     beta  = 0.0347
##     gamma = 2e-04
##
##   Initial states:
##     l = 330.4766
```

```
##     b = 3.4317
##     s = 1.0044 0.8557 0.8844 0.9271 1.083 1.0018
##            1.0632 1.1491 1.0472 1.2102 0.9571 0.8168
##
##   sigma:  0.0746
##
##      AIC     AICc      BIC
##  962.9794  972.2521 1004.3033
##
## Error measures:
##                    ME     RMSE      MAE        MPE     MAPE      MAS
E
## Training set -1.69851 27.39812 22.23781 -0.6868033 5.488451 0.560165
8
##                   ACF1
## Training set 0.2798302
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 2017       347.0537 313.8601 380.2474 296.2884 397.8191
## Feb 2017       405.2750 364.9560 445.5941 343.6124 466.9376
## Mar 2017       510.6797 457.4902 563.8691 429.3334 592.0260
## Apr 2017       440.3977 392.1030 488.6924 366.5373 514.2581
## May 2017       481.5952 425.7295 537.4608 396.1560 567.0343
## Jun 2017       444.0370 389.3523 498.7217 360.4040 527.6700
## Jul 2017       416.9218 362.2675 471.5760 333.3353 500.5082
## Aug 2017       449.1230 386.3457 511.9003 353.1134 545.1326
## Sep 2017       383.1663 326.0050 440.3275 295.7457 470.5869
## Oct 2017       364.2256 306.2166 422.2346 275.5085 452.9428
## Nov 2017       351.1548 291.4581 410.8515 259.8565 442.4530
## Dec 2017       410.6923 336.2115 485.1732 296.7838 524.6009
## Jan 2018       332.8021 268.4697 397.1345 234.4142 431.1900
```

```r
# Fit Holt-Winter's seasonal multiplicative model to full data set and
forecast 2-yrs. ahead

hw.mult.full = hw(AutoSales,seasonal="mult",h=24)
```

```
autoplot(hw.mult.full) + xlab("Year") + ylab("Thousands of Cars Sold")
+ ggtitle("US Auto Sales with Holt-Winters Seasonal (multiplicative)")
```




```
# Display table of forecasts (h = 24)
hw.mult.full
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Feb 2018       348.0713 316.3271 379.8155 299.5227 396.6199
## Mar 2018       426.9713 384.5914 469.3513 362.1567 491.7860
## Apr 2018       370.5191 330.6070 410.4312 309.4788 431.5594
## May 2018       408.6242 360.9867 456.2617 335.7689 481.4795
## Jun 2018       368.1945 321.8628 414.5262 297.3363 439.0527
## Jul 2018       339.5818 293.5752 385.5885 269.2207 409.9429
## Aug 2018       369.2819 315.5457 423.0181 287.0994 451.4643
## Sep 2018       318.0752 268.4745 367.6759 242.2174 393.9329
## Oct 2018       302.5396 252.0894 352.9899 225.3826 379.6967
## Nov 2018       294.5470 242.1280 346.9660 214.3791 374.7149
## Dec 2018       327.8428 265.6937 389.9919 232.7939 422.8916
## Jan 2019       262.6335 209.6930 315.5739 181.6680 343.5989
```

```
## Feb 2019       312.4250 245.5615 379.2885 210.1661 414.6839
## Mar 2019       382.8684 296.0205 469.7163 250.0460 515.6908
## Apr 2019       331.9149 252.2312 411.5986 210.0493 453.7805
## May 2019       365.6769 272.8920 458.4618 223.7747 507.5792
## Jun 2019       329.1546 240.9975 417.3116 194.3299 463.9792
## Jul 2019       303.2547 217.6266 388.8828 172.2978 434.2115
## Aug 2019       329.4222 231.4692 427.3751 179.6160 479.2283
## Sep 2019       283.4310 194.7763 372.0857 147.8453 419.0167
## Oct 2019       269.2857 180.7704 357.8011 133.9132 404.6583
## Nov 2019       261.8723 171.5008 352.2438 123.6610 400.0836
## Dec 2019       291.1351 185.7490 396.5212 129.9610 452.3093
## Jan 2020       232.9502 144.5749 321.3254  97.7919 368.1084
```

Holt-Winter’s Seasonal Method with Damped Trend

Damping is possible with both additive and multiplicative Holt-Winters’ methods. A method that often provides accurate and robust forecasts for seasonal data is the Holt-Winters method with a damped trend and multiplicative seasonality:

$$\begin{aligned}\hat{y}_{t+h|t} &= [\ell_t + (\phi+\phi^2+\cdots+\phi^h)b_t]s_{t-m+h_m^+}.\\ \ell_t &= \alpha(y_t/s_{t-m}) + (1-\alpha)(\ell_{t-1}+\phi b_{t-1})\\ b_t &= \beta^*(\ell_t-\ell_{t-1}) + (1-\beta^*)\phi b_{t-1}\\ s_t &= \gamma\frac{y_t}{(\ell_{t-1}+\phi b_{t-1})} + (1-\gamma)s_{t-m}.\end{aligned}$$

The parameter ϕ controls the amount of dampening, the smallter ϕ is the more the trend is dampened/decreased. The value of ϕ will be estimated along the initial $(t=0)$ values for the level, slope, and seasonal effects. To introduce a damped-trend we added the argument `damped=TRUE` to the call to the `hw()` function, e.g. `hw(AutoSales,damped=TRUE,seasonal="multiplicative")` will fit the Holt-Winter’s multiplicative seasonal model to the auto sales data with damped trend.

Example 7.2 - U.S. Domestic Auto Sales (cont’d)

```
Auto.test = tail(AutoSales,13)

Auto.train = head(AutoSales,84)

hw.add.damp = hw(Auto.train,seasonal="additive",damped=TRUE,h=13)

hw.mult.damp = hw(Auto.train,seasonal="multiplicative",damped=TRUE,h=1
3)

accuracy(hw.add.damp,Auto.test)
```

```
##                          ME      RMSE        MAE          MPE       MAPE
 MASE
```

```
## Training set  -1.129668 25.01176 19.76083  -0.5411722  4.932653 0.49
77712
## Test set     -59.630133 63.49083 59.75899 -16.2196357 16.251211 1.50
53169
##                     ACF1 Theil's U
## Training set -0.03636566        NA
## Test set     -0.07778679  1.379807
```

```
accuracy(hw.mult.damp,Auto.test)
```

```
##                      ME     RMSE      MAE         MPE      MAPE
MASE
## Training set  -1.824641 24.77461 20.05645  -0.7519949  4.996755 0.50
5218
## Test set     -63.858531 67.65422 63.85853 -17.2767676 17.276768 1.60
8584
##                     ACF1 Theil's U
## Training set -0.03919155        NA
## Test set     -0.06300179  1.473287
```

```
# Compare to non-damped trend HW models
accuracy(hw.add,Auto.test)
```

```
##                      ME     RMSE      MAE         MPE      MAPE
 MASE
## Training set  -1.621898 25.45453 20.15591  -0.5917106  5.072157 0.50
77232
## Test set     -72.302735 75.91779 72.30273 -19.7235043 19.723504 1.82
12913
##                     ACF1 Theil's U
## Training set -0.03699112        NA
## Test set     -0.07098146  1.664592
```

```
accuracy(hw.mult,Auto.test)
```

```
##                     ME     RMSE      MAE        MPE      MAPE       M
ASE
## Training set  -1.69851 27.39812 22.23781 -0.6868033  5.488451 0.5601
658
## Test set     -34.69418 41.30596 38.53013 -9.4245447 10.364500 0.9705
662
##                   ACF1 Theil's U
## Training set 0.2798302        NA
## Test set     0.0366181  0.888674
```

For these data the non-damped multiplicative Holt-Winter's model predicts the test cases most accurately. Despite this fact, we will examine plots the 2-year forecasts using both the damped and non-damped multiplicative Holt-Winter's models fit the entire auto sales time series.

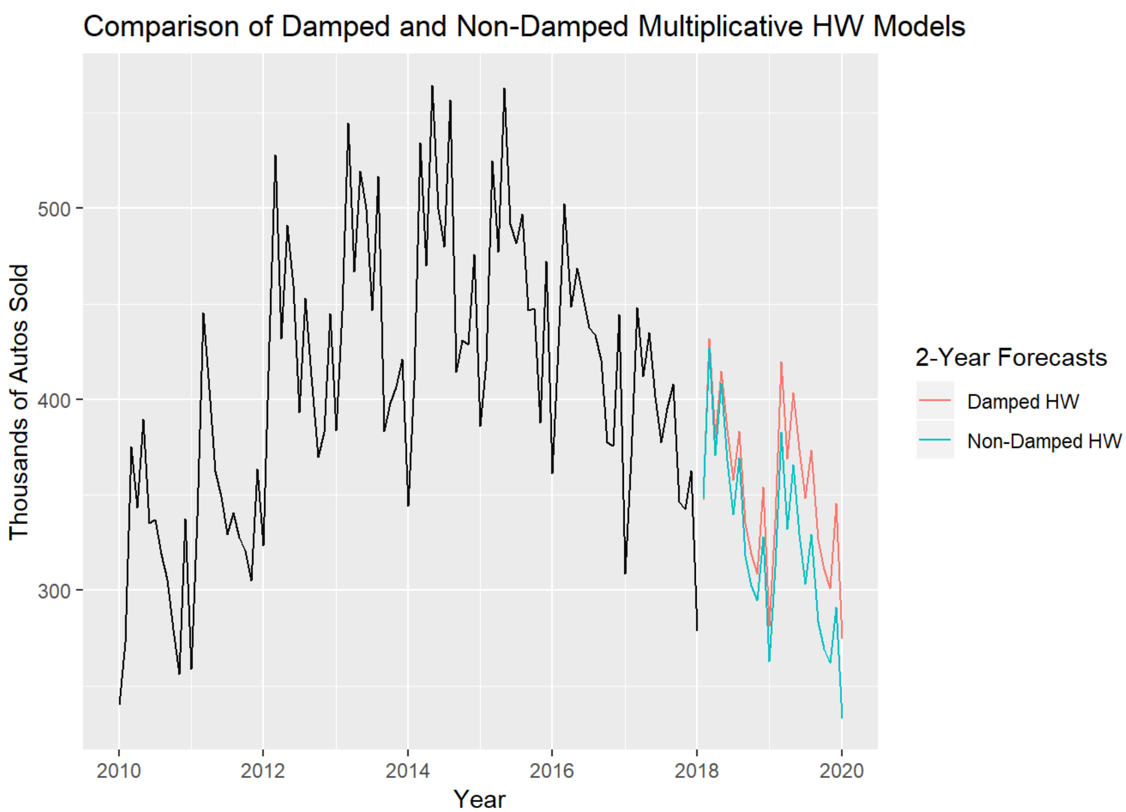
```
hw.mult = hw(AutoSales,seasonal="mult",h=24)

hw.mult.damped = hw(AutoSales,seasonal="mult",damped=TRUE,h=24)

autoplot(AutoSales) + xlab("Year") + ylab("Thousands of Autos Sold") +
ggtitle("Comparison of Damped and Non-Damped Multiplicative HW Models")
 +
  autolayer(hw.mult.damped$mean,series="Damped HW") +

  autolayer(hw.mult$mean,series="Non-Damped HW") +

  guides(colour=guide_legend(title="2-Year Forecasts"))
```



Here you can clearly see the what damped trend means. The non-damped HW multiplicative model continues the decreasing trend in auto sales, possibly to unrealistically low values (at least some of you thought so on HW 4). The damped trend, while still suggesting a continued decrease in sales, is not nearly as steep, i.e. the trend has been dampened/decreased.