Data 624 Homework 5 Chapter 8.8

Enid Roman

2024-10-05

# Load required libraries  
  
library(fpp3)  
library(tsibble)  
library(ggplot2)  
library(tidyverse)  
library(forecast)

## **1. Consider the the number of pigs slaughtered in Victoria, available in the aus\_livestock dataset.**

## **a. Use the ETS() function to estimate the equivalent model for simple exponential smoothing. Find the optimal values of α and ℓ 0, and generate forecasts for the next four months.**

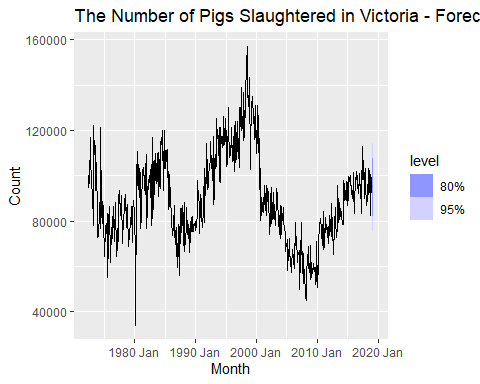
#?aus\_livestock  
  
# Convert the data into a time series table (tsibble), filtering for Victoria and Pigs  
aus\_livestock\_ts <- aus\_livestock %>%  
 filter(State == "Victoria", Animal == "Pigs") %>%  
 as\_tsibble(index = Month, key = c(Animal, State))   
  
# Fit the ETS model (Simple Exponential Smoothing)  
pigs\_fit <- aus\_livestock\_ts %>%  
 model(ETS(Count ~ error("A") + trend("N") + season("N")))  
  
# View the model report  
report(pigs\_fit)

## Series: Count   
## Model: ETS(A,N,N)   
## Smoothing parameters:  
## alpha = 0.3221247   
##   
## Initial states:  
## l[0]  
## 100646.6  
##   
## sigma^2: 87480760  
##   
## AIC AICc BIC   
## 13737.10 13737.14 13750.07

# Forecast the next 4 months  
pig\_fc <- pigs\_fit %>%  
 forecast(h = 4)  
  
# View the forecast to ensure it's correct  
print(pig\_fc)

## # A fable: 4 x 6 [1M]  
## # Key: Animal, State, .model [1]  
## Animal State .model Month Count .mean  
## <fct> <fct> <chr> <mth> <dist> <dbl>  
## 1 Pigs Victoria "ETS(Count ~ error(\"A\") +… 2019 Jan N(95187, 8.7e+07) 95187.  
## 2 Pigs Victoria "ETS(Count ~ error(\"A\") +… 2019 Feb N(95187, 9.7e+07) 95187.  
## 3 Pigs Victoria "ETS(Count ~ error(\"A\") +… 2019 Mar N(95187, 1.1e+08) 95187.  
## 4 Pigs Victoria "ETS(Count ~ error(\"A\") +… 2019 Apr N(95187, 1.1e+08) 95187.

# Plot the forecast  
pig\_fc %>%  
 autoplot(aus\_livestock\_ts) +  
 ggtitle("The Number of Pigs Slaughtered in Victoria - Forecast")



The model uses a relatively moderate smoothing level𝛼= 0.3221. The initial level of the series is approximately ℓ 0 = 100646.6

The forecasts for the next four months:

For January 2019, the forecast is 𝜇=95187 with variance 𝜎2 = 8.7 × 10 to the 7 power. For February 2019, the forecast is 𝜇=95187 with variance 𝜎2 = 9.7 × 10 to the 7 power. For March 2019, the forecast is 𝜇=95187 with variance 𝜎2 = 1.1 × 10 to the 8 power. For April 2019, the forecast is 𝜇=95187 with variance 𝜎2 = 1.1 × 10 to the 8 power.

## **b. Compute a 95% prediction interval for the first forecast using ^y±1.96s where s is the standard deviation of the residuals. Compare your interval with the interval produced by R.**

glance(pigs\_fit)

## # A tibble: 1 × 11  
## Animal State .model sigma2 log\_lik AIC AICc BIC MSE AMSE MAE  
## <fct> <fct> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 Pigs Victoria "ETS(… 8.75e7 -6866. 13737. 13737. 13750. 8.72e7 9.70e7 7190.

# Extract the first forecast mean  
first\_forecast\_mean <- pig\_fc$.mean[1]  
  
# Extract the residual standard deviation (sigma) from the fitted model  
residual\_sd <- sqrt(87480760) # This is sigma^2 from glance(pigs\_fit)  
  
# Compute the 95% prediction interval  
z <- 1.96  
lower\_bound <- first\_forecast\_mean - z \* residual\_sd  
upper\_bound <- first\_forecast\_mean + z \* residual\_sd  
  
# Print the prediction interval  
prediction\_interval <- c(lower\_bound, upper\_bound)  
print(prediction\_interval)

## [1] 76854.45 113518.66

The 95% prediction interval result of your calculation gives the prediction interval for the first forecast as:

Lower bound: 76,854.45 Upper bound: 113,518.66

This interval means that you can be 95% confident that the actual number of pigs slaughtered in the first forecasted month will fall between these two values, based on the residual variance and the forecast mean.

pig\_fc %>% hilo(95) %>% pull('95%') %>% head(1)

## <hilo[1]>  
## [1] [76854.79, 113518.3]95

Manually Calculated Interval:

Lower bound: 76,854.45 Upper bound: 113,518.66

R-Generated Interval (using hilo(95)):

Lower bound: 76,854.79 Upper bound: 113,518.30

Difference in Lower Bound: The difference is very small (0.34). This difference could be due to the precision and rounding used in the manual calculation vs. how R internally handles floating-point arithmetic and rounding.

Difference in Upper Bound: The difference is similarly small (0.36), likely for the same reasons.

You can confidently use either method, as they both produce nearly the same results. The hilo() method is preferable for convenience since it directly leverages R’s internal forecasting and interval computation capabilities.

## **2. Data set global\_economy contains the annual Exports from many countries. Select one country to analyse.**

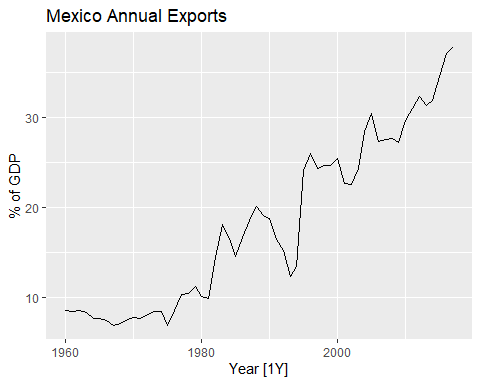
#?global\_economy  
unique(global\_economy$Country)

## [1] Afghanistan   
## [2] Albania   
## [3] Algeria   
## [4] American Samoa   
## [5] Andorra   
## [6] Angola   
## [7] Antigua and Barbuda   
## [8] Arab World   
## [9] Argentina   
## [10] Armenia   
## [11] Aruba   
## [12] Australia   
## [13] Austria   
## [14] Azerbaijan   
## [15] Bahamas, The   
## [16] Bahrain   
## [17] Bangladesh   
## [18] Barbados   
## [19] Belarus   
## [20] Belgium   
## [21] Belize   
## [22] Benin   
## [23] Bermuda   
## [24] Bhutan   
## [25] Bolivia   
## [26] Bosnia and Herzegovina   
## [27] Botswana   
## [28] Brazil   
## [29] British Virgin Islands   
## [30] Brunei Darussalam   
## [31] Bulgaria   
## [32] Burkina Faso   
## [33] Burundi   
## [34] Cabo Verde   
## [35] Cambodia   
## [36] Cameroon   
## [37] Canada   
## [38] Caribbean small states   
## [39] Cayman Islands   
## [40] Central African Republic   
## [41] Central Europe and the Baltics   
## [42] Chad   
## [43] Channel Islands   
## [44] Chile   
## [45] China   
## [46] Colombia   
## [47] Comoros   
## [48] Congo, Dem. Rep.   
## [49] Congo, Rep.   
## [50] Costa Rica   
## [51] Cote d'Ivoire   
## [52] Croatia   
## [53] Cuba   
## [54] Curacao   
## [55] Cyprus   
## [56] Czech Republic   
## [57] Denmark   
## [58] Djibouti   
## [59] Dominica   
## [60] Dominican Republic   
## [61] Early-demographic dividend   
## [62] East Asia & Pacific   
## [63] East Asia & Pacific (excluding high income)   
## [64] East Asia & Pacific (IDA & IBRD countries)   
## [65] Ecuador   
## [66] Egypt, Arab Rep.   
## [67] El Salvador   
## [68] Equatorial Guinea   
## [69] Eritrea   
## [70] Estonia   
## [71] Eswatini   
## [72] Ethiopia   
## [73] Euro area   
## [74] Europe & Central Asia   
## [75] Europe & Central Asia (excluding high income)   
## [76] Europe & Central Asia (IDA & IBRD countries)   
## [77] European Union   
## [78] Faroe Islands   
## [79] Fiji   
## [80] Finland   
## [81] Fragile and conflict affected situations   
## [82] France   
## [83] French Polynesia   
## [84] Gabon   
## [85] Gambia, The   
## [86] Georgia   
## [87] Germany   
## [88] Ghana   
## [89] Gibraltar   
## [90] Greece   
## [91] Greenland   
## [92] Grenada   
## [93] Guam   
## [94] Guatemala   
## [95] Guinea   
## [96] Guinea-Bissau   
## [97] Guyana   
## [98] Haiti   
## [99] Heavily indebted poor countries (HIPC)   
## [100] High income   
## [101] Honduras   
## [102] Hong Kong SAR, China   
## [103] Hungary   
## [104] IBRD only   
## [105] Iceland   
## [106] IDA & IBRD total   
## [107] IDA blend   
## [108] IDA only   
## [109] IDA total   
## [110] India   
## [111] Indonesia   
## [112] Iran, Islamic Rep.   
## [113] Iraq   
## [114] Ireland   
## [115] Isle of Man   
## [116] Israel   
## [117] Italy   
## [118] Jamaica   
## [119] Japan   
## [120] Jordan   
## [121] Kazakhstan   
## [122] Kenya   
## [123] Kiribati   
## [124] Korea, Dem. People's Rep.   
## [125] Korea, Rep.   
## [126] Kosovo   
## [127] Kuwait   
## [128] Kyrgyz Republic   
## [129] Lao PDR   
## [130] Late-demographic dividend   
## [131] Latin America & Caribbean   
## [132] Latin America & Caribbean (excluding high income)   
## [133] Latin America & the Caribbean (IDA & IBRD countries)  
## [134] Latvia   
## [135] Least developed countries: UN classification   
## [136] Lebanon   
## [137] Lesotho   
## [138] Liberia   
## [139] Libya   
## [140] Liechtenstein   
## [141] Lithuania   
## [142] Low & middle income   
## [143] Low income   
## [144] Lower middle income   
## [145] Luxembourg   
## [146] Macao SAR, China   
## [147] Macedonia, FYR   
## [148] Madagascar   
## [149] Malawi   
## [150] Malaysia   
## [151] Maldives   
## [152] Mali   
## [153] Malta   
## [154] Marshall Islands   
## [155] Mauritania   
## [156] Mauritius   
## [157] Mexico   
## [158] Micronesia, Fed. Sts.   
## [159] Middle East & North Africa   
## [160] Middle East & North Africa (excluding high income)   
## [161] Middle East & North Africa (IDA & IBRD countries)   
## [162] Middle income   
## [163] Moldova   
## [164] Monaco   
## [165] Mongolia   
## [166] Montenegro   
## [167] Morocco   
## [168] Mozambique   
## [169] Myanmar   
## [170] Namibia   
## [171] Nauru   
## [172] Nepal   
## [173] Netherlands   
## [174] New Caledonia   
## [175] New Zealand   
## [176] Nicaragua   
## [177] Niger   
## [178] Nigeria   
## [179] North America   
## [180] Northern Mariana Islands   
## [181] Norway   
## [182] OECD members   
## [183] Oman   
## [184] Other small states   
## [185] Pacific island small states   
## [186] Pakistan   
## [187] Palau   
## [188] Panama   
## [189] Papua New Guinea   
## [190] Paraguay   
## [191] Peru   
## [192] Philippines   
## [193] Poland   
## [194] Portugal   
## [195] Post-demographic dividend   
## [196] Pre-demographic dividend   
## [197] Puerto Rico   
## [198] Qatar   
## [199] Romania   
## [200] Russian Federation   
## [201] Rwanda   
## [202] Samoa   
## [203] San Marino   
## [204] Sao Tome and Principe   
## [205] Saudi Arabia   
## [206] Senegal   
## [207] Serbia   
## [208] Seychelles   
## [209] Sierra Leone   
## [210] Singapore   
## [211] Sint Maarten (Dutch part)   
## [212] Slovak Republic   
## [213] Slovenia   
## [214] Small states   
## [215] Solomon Islands   
## [216] Somalia   
## [217] South Africa   
## [218] South Asia   
## [219] South Asia (IDA & IBRD)   
## [220] South Sudan   
## [221] Spain   
## [222] Sri Lanka   
## [223] St. Kitts and Nevis   
## [224] St. Lucia   
## [225] St. Martin (French part)   
## [226] St. Vincent and the Grenadines   
## [227] Sub-Saharan Africa   
## [228] Sub-Saharan Africa (excluding high income)   
## [229] Sub-Saharan Africa (IDA & IBRD countries)   
## [230] Sudan   
## [231] Suriname   
## [232] Sweden   
## [233] Switzerland   
## [234] Syrian Arab Republic   
## [235] Tajikistan   
## [236] Tanzania   
## [237] Thailand   
## [238] Timor-Leste   
## [239] Togo   
## [240] Tonga   
## [241] Trinidad and Tobago   
## [242] Tunisia   
## [243] Turkey   
## [244] Turkmenistan   
## [245] Turks and Caicos Islands   
## [246] Tuvalu   
## [247] Uganda   
## [248] Ukraine   
## [249] United Arab Emirates   
## [250] United Kingdom   
## [251] United States   
## [252] Upper middle income   
## [253] Uruguay   
## [254] Uzbekistan   
## [255] Vanuatu   
## [256] Venezuela, RB   
## [257] Vietnam   
## [258] Virgin Islands (U.S.)   
## [259] West Bank and Gaza   
## [260] World   
## [261] Yemen, Rep.   
## [262] Zambia   
## [263] Zimbabwe   
## 263 Levels: Afghanistan Albania Algeria American Samoa Andorra ... Zimbabwe

I will analyze Mexico’s Annual Exports as a percentage of GDP over time.

## **a. Plot the Exports series and discuss the main features of the data.**

mx\_econ <- global\_economy %>%  
 filter(Country == "Mexico") %>%  
 autoplot(Exports) +  
 labs(y="% of GDP", title="Mexico Annual Exports")  
  
mx\_econ



There is a Steady Growth in GDP. Mexico’s exports as a percentage of GDP show a clear upward trend from below 10% in the 1960s to over 30% by 2015.

Between 1960 and the mid-1980s, exports remained relatively stable, hovering under 10% of GDP, reflecting limited integration into the global economy.

There is a sharp rise in exports occurred starting in the mid-1980s, likely driven by Mexico’s trade liberalization and entry into major trade agreements like NAFTA in 1994.

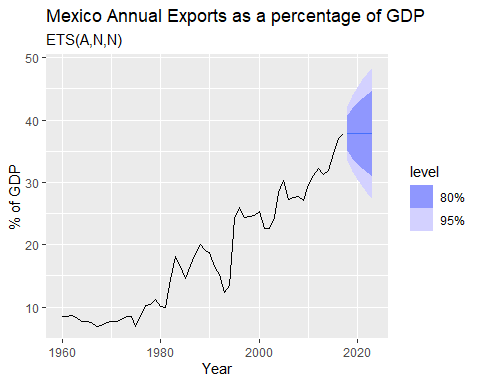
There was noticeable volatility in the late 1980s and early 1990s, possibly due to economic reforms and crises, such as the Mexican peso crisis in 1994.

There is a strong Growth in 2000s. After the 1990s, exports became a larger part of the economy, surpassing 30% of GDP by 2015, indicating Mexico’s increased integration into the global economy, particularly in industries like automotive and manufacturing.

The data suggests that exports have become a key driver of Mexico’s economic growth, with industrialization and trade agreements playing significant roles.

## **b. Use an ETS(A,N,N) model to forecast the series, and plot the forecasts.**

mx\_ft <- global\_economy %>%  
 filter(Country == "Mexico") %>%  
 model(ETS(Exports ~ error("A") + trend("N") + season("N")))   
  
mx\_fc <- mx\_ft %>%  
 forecast(h = 6)  
  
mx\_fc %>%  
 autoplot(global\_economy) +  
 labs(y="% of GDP", title="Mexico Annual Exports as a percentage of GDP", subtitle = "ETS(A,N,N)")



A narrower range that suggests an 80% chance that the future export values will fall within this band. It reflects less uncertainty compared to the 95% interval.

A wider range, indicating a 95% chance that future export values will fall within this broader band. This captures more uncertainty and accounts for potential extreme scenarios.

Both intervals widen over time, showing increasing uncertainty the further into the future the forecast goes. This is expected due to the unpredictability of future economic conditions.

The model predicts continued growth in exports, with greater confidence in the short term and increasing uncertainty as the forecast extends further into the future.

## **c. Compute the RMSE values for the training data.**

# Ensure the dataset is a tsibble with Year as the index  
mx\_econ <- global\_economy %>%  
 filter(Country == "Mexico") %>%  
 as\_tsibble(index = Year)  
  
# Fit the ETS(A,N,N) model (Simple Exponential Smoothing)  
mx\_fit <- mx\_econ %>%  
 model(ETS(Exports ~ error("A") + trend("N") + season("N")))  
  
# Extract the residuals as a numeric vector  
residuals <- mx\_fit %>%  
 augment() %>%  
 pull(.resid) # Extract the residuals column  
  
# Compute RMSE for the training data  
rmse\_value <- sqrt(mean(residuals^2, na.rm = TRUE))  
  
# Print the RMSE value with a message using paste()  
print(paste("The RMSE for the training data is", rmse\_value))

## [1] "The RMSE for the training data is 2.15442536513068"

The RMSE of 2.15442536513068 means that when the model makes predictions on the training data, the average error is about 2.15. This gives you an idea of how accurate (or inaccurate) the model’s predictions are, with lower RMSE values generally indicating better performance.

## **d. Compare the results to those from an ETS(A,A,N) model. (Remember that the trended model is using one more parameter than the simpler model.) Discuss the merits of the two forecasting methods for this data set.**

# Fit the ETS(A,N,N) model (Simple Exponential Smoothing)  
mx\_fit\_ANN <- mx\_econ %>%  
 model(ETS(Exports ~ error("A") + trend("N") + season("N")))  
  
# Fit the ETS(A,A,N) model (Additive error and trend)  
mx\_fit\_AAN <- mx\_econ %>%  
 model(ETS(Exports ~ error("A") + trend("A") + season("N")))  
  
# Extract the residuals and compute RMSE for the ETS(A,N,N) model  
residuals\_ANN <- mx\_fit\_ANN %>%  
 augment() %>%  
 pull(.resid)  
  
rmse\_ANN <- sqrt(mean(residuals\_ANN^2, na.rm = TRUE))  
  
# Extract the residuals and compute RMSE for the ETS(A,A,N) model  
residuals\_AAN <- mx\_fit\_AAN %>%  
 augment() %>%  
 pull(.resid)  
  
rmse\_AAN <- sqrt(mean(residuals\_AAN^2, na.rm = TRUE))  
  
# Print the RMSE values for comparison  
print(paste("The RMSE for the ETS(A,N,N) model is",rmse\_ANN))

## [1] "The RMSE for the ETS(A,N,N) model is 2.15442536513068"

print(paste("The RMSE for the ETS(A,A,N) model is",rmse\_AAN))

## [1] "The RMSE for the ETS(A,A,N) model is 2.09399877164435"

In comparing the results of the ETS(A,A,N) and ETS(A,N,N) models, the key takeaway is that while the ETS(A,A,N) model performs slightly better with a lower RMSE (2.094) than the ETS(A,N,N) model (2.154), the difference is marginal. This suggests that both models are relatively effective, but they offer different benefits based on the characteristics of the dataset.

Merits of ETS(A,N,N) Model (Simple Exponential Smoothing):

Simplicity: The ETS(A,N,N) model assumes no trend and no seasonality, making it simpler and easier to interpret, as it has fewer parameters to estimate.

Low risk of overfitting: Since it does not model a trend, it is less likely to overfit noisy data, making it a good choice for datasets that lack a clear trend or when interpretability and simplicity are prioritized.

Best for flat data: This model is ideal in cases where the data does not show consistent upward or downward movement over time.

Drawback: In datasets with an evident trend, like Mexico’s export data, this model may underperform because it cannot capture any gradual or consistent growth, potentially leading to less accurate forecasts.

Merits of ETS(A,A,N) Model (Additive Trend):

Captures trends: The ETS(A,A,N) model includes an additive trend component, allowing it to capture steady upward or downward movements in the data, making it more suitable for datasets with a trend (e.g., Mexico’s export data).

Better fit: For data with a clear trend, this model provides a better fit, as evidenced by its lower RMSE, leading to more accurate predictions.

Drawback: The increased complexity of this model comes from the inclusion of an additional parameter to estimate the trend. This makes it slightly more prone to overfitting, especially in datasets with high noise or inconsistent trends. Overfitting can lead to poor performance on future or unseen data, even if it fits the training data well.

Model Complexity and Parameterization:

The ETS(A,N,N) model, with its fewer parameters, is computationally simpler and more interpretable, making it a good option when there’s no strong indication of a trend.

The ETS(A,A,N) model, while more complex due to the added trend parameter, may provide better forecasts in cases where the data shows a clear trend, though with some risk of overfitting.

ETS(A,N,N) is a robust, simpler model that works well when no trend is apparent, and it minimizes overfitting.

ETS(A,A,N) provides more accuracy in trending datasets but at the cost of added complexity and the risk of overfitting.

The choice between the two models depends on whether the dataset (like Mexico’s exports) exhibits a clear trend. If a trend exists, the ETS(A,A,N) model will likely perform better, but if simplicity and low risk of overfitting are priorities, the ETS(A,N,N) model is a solid alternative.

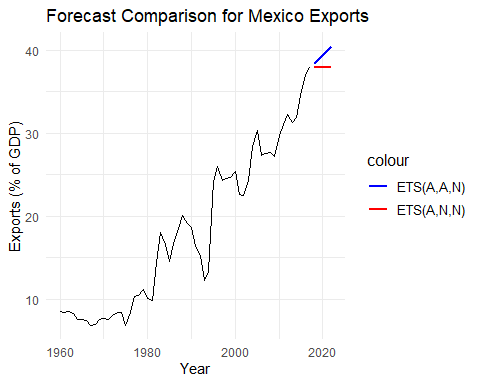
Recommendations for this Dataset:

Given that Mexico’s exports show a clear upward trend over time, the ETS(A,A,N) model is likely to be the more appropriate choice for forecasting this dataset. It captures the trend component, leading to more accurate predictions with a lower RMSE.

However, the small difference in RMSE suggests that the simpler ETS(A,N,N) model could still be used, especially if you prefer a model with fewer assumptions or parameters.

## **e. Compare the forecasts from both methods. Which do you think is best?**

# Ensure the dataset is a tsibble with Year as the index  
mx\_econ <- global\_economy %>%  
 filter(Country == "Mexico") %>%  
 as\_tsibble(index = Year)  
  
# Fit the ETS(A,N,N) model (Simple Exponential Smoothing)  
mx\_fit\_ANN <- mx\_econ %>%  
 model(ETS(Exports ~ error("A") + trend("N") + season("N")))  
  
# Fit the ETS(A,A,N) model (Additive error and trend)  
mx\_fit\_AAN <- mx\_econ %>%  
 model(ETS(Exports ~ error("A") + trend("A") + season("N")))  
  
# Generate forecasts for the next 5 years from both models  
mx\_forecast\_ANN <- mx\_fit\_ANN %>%  
 forecast(h = 5)  
  
mx\_forecast\_AAN <- mx\_fit\_AAN %>%  
 forecast(h = 5)  
  
# Manual ggplot to overlay both forecasts  
ggplot(mx\_econ, aes(x = Year, y = Exports)) +  
 geom\_line(color = "black") + # Historical data  
 geom\_line(data = mx\_forecast\_ANN, aes(x = Year, y = .mean, color = "ETS(A,N,N)"), size = 1) + # ETS(A,N,N) forecast  
 geom\_line(data = mx\_forecast\_AAN, aes(x = Year, y = .mean, color = "ETS(A,A,N)"), size = 1) + # ETS(A,A,N) forecast  
 labs(title = "Forecast Comparison for Mexico Exports", y = "Exports (% of GDP)") +  
 scale\_color\_manual(values = c("ETS(A,N,N)" = "red", "ETS(A,A,N)" = "blue")) + # Custom colors for each model  
 theme\_minimal() +  
 theme(legend.title = element\_text(size = 12), legend.text = element\_text(size = 10))



ETS(A,N,N) (Red Line):

This model assumes no trend, so its forecast remains relatively flat or shows very minimal change. The red line in the plot shows a slight increase but does not project the clear upward trend we’ve seen in the historical data.

ETS(A,A,N) (Blue Line):

This model includes an additive trend, which means it captures the consistent upward trajectory of Mexico’s exports over time. The blue line continues the upward trend we see in the historical data, which is consistent with the behavior of the data over the past few decades.

ETS(A,A,N) (Additive Trend) is clearly the better model for this data set because:

The historical data consistency. The data shows a long-term upward trend, and the ETS(A,A,N) model captures this trend, continuing it into the forecast period.

The ETS(A,A,N) forecast is more realistic for a dataset like this, where exports are expected to grow. The ETS(A,N,N) model, which assumes no trend, underestimates the future values by projecting a nearly flat forecast.

Is better long-term predictions. If you are looking for long-term accuracy, a model that captures the trend (like ETS(A,A,N)) is more appropriate, as Mexico’s exports have been increasing steadily.

## **f. Calculate a 95% prediction interval for the first forecast for each model, using the RMSE values and assuming normal errors. Compare your intervals with those produced using R.**

# Extract the first forecasted values  
first\_forecast\_ANN <- mx\_forecast\_ANN$.mean[1]  
first\_forecast\_AAN <- mx\_forecast\_AAN$.mean[1]  
  
# Extract RMSE values  
residuals\_ANN <- mx\_fit\_ANN %>%  
 augment() %>%  
 pull(.resid)  
  
residuals\_AAN <- mx\_fit\_AAN %>%  
 augment() %>%  
 pull(.resid)  
  
# Calculate RMSE for each model  
rmse\_ANN <- sqrt(mean(residuals\_ANN^2, na.rm = TRUE))  
rmse\_AAN <- sqrt(mean(residuals\_AAN^2, na.rm = TRUE))  
  
# Calculate 95% prediction intervals (assuming normal errors)  
z\_value <- 1.96  
  
# Prediction interval for ETS(A,N,N)  
lower\_ANN <- first\_forecast\_ANN - z\_value \* rmse\_ANN  
upper\_ANN <- first\_forecast\_ANN + z\_value \* rmse\_ANN  
prediction\_interval\_ANN <- c(lower\_ANN, upper\_ANN)  
  
# Prediction interval for ETS(A,A,N)  
lower\_AAN <- first\_forecast\_AAN - z\_value \* rmse\_AAN  
upper\_AAN <- first\_forecast\_AAN + z\_value \* rmse\_AAN  
prediction\_interval\_AAN <- c(lower\_AAN, upper\_AAN)  
  
# Print the prediction intervals  
print(paste("95% Prediction Interval for ETS(A,N,N):", prediction\_interval\_ANN))

## [1] "95% Prediction Interval for ETS(A,N,N): 33.6436653266804"  
## [2] "95% Prediction Interval for ETS(A,N,N): 42.0890127579927"

print(paste("95% Prediction Interval for ETS(A,A,N):", prediction\_interval\_AAN))

## [1] "95% Prediction Interval for ETS(A,A,N): 34.2779984865041"  
## [2] "95% Prediction Interval for ETS(A,A,N): 42.48647367135"

# Now compare with R's built-in prediction intervals  
# Extract R's intervals from the forecast objects  
r\_interval\_ANN <- mx\_forecast\_ANN %>%  
 hilo(95) %>%  
 as\_tibble() %>%  
 slice(1) %>% # Extract first forecast  
 pull(`95%`)  
  
r\_interval\_AAN <- mx\_forecast\_AAN %>%  
 hilo(95) %>%  
 as\_tibble() %>%  
 slice(1) %>% # Extract first forecast  
 pull(`95%`)  
  
# Print R's prediction intervals  
print(paste("R's 95% Prediction Interval for ETS(A,N,N):", r\_interval\_ANN))

## [1] "R's 95% Prediction Interval for ETS(A,N,N): [33.5690009022366, 42.1636771824364]95"

print(paste("R's 95% Prediction Interval for ETS(A,A,N):", r\_interval\_AAN))

## [1] "R's 95% Prediction Interval for ETS(A,A,N): [34.1287831553961, 42.635689002458]95"

Manually Calculated 95% Prediction Intervals:

ETS(A,N,N): [33.64, 42.09] ETS(A,A,N): [34.28, 42.49]

R’s 95% Prediction Intervals:

ETS(A,N,N): [33.57, 42.16] ETS(A,A,N): [34.13, 42.64]

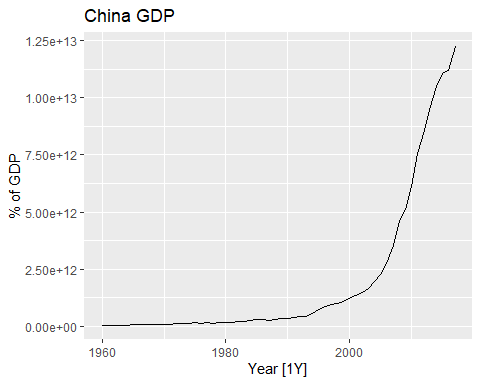
The manually calculated intervals are very close to R’s built-in intervals. The small differences in the bounds (within 0.1 units) are likely due to slight differences in how R handles prediction interval calculations. R may be accounting for additional factors like model-specific error terms, bias correction, or more accurate adjustments based on the internal model fitting process.

Both the manual calculation and R’s intervals are consistent, indicating that the manual method using the RMSE and normal errors is a good approximation.

The differences are small enough to suggest that either method would provide reliable interval estimates for forecasting purposes.

## **6. Forecast the Chinese GDP from the global\_economy data set using an ETS model. Experiment with the various options in the ETS() function to see how much the forecasts change with damped trend, or with a Box-Cox transformation. Try to develop an intuition of what each is doing to the forecasts. [Hint: use a relatively large value of h when forecasting, so you can clearly see the differences between the various options when plotting the forecasts.]**

# Filter for China's GDP data  
ch\_econ <- global\_economy %>%  
 filter(Country == "China") %>%  
 autoplot(GDP) +  
 labs(y="% of GDP", title="China GDP")  
  
ch\_econ



China’s GDP over time shows a clear exponential growth pattern, especially from the late 1990s onward.

Between 1960 and the late 1990s, China’s GDP remained relatively stable with minimal growth. This period represents the country’s slower economic growth under a planned economy and prior to major economic reforms.

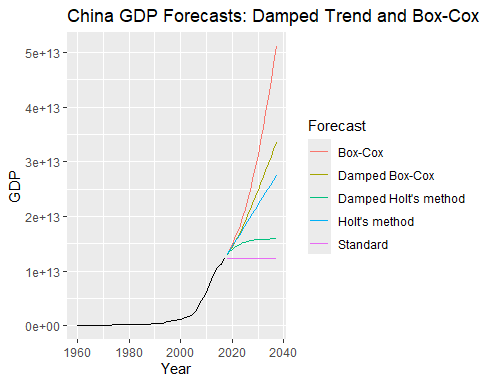
Around the late 1990s, China’s GDP began to increase dramatically, corresponding to the country’s shift towards a more market-oriented economy and increased global trade.

This trend accelerated through the 2000s and 2010s, with China becoming a major player in the global economy, as reflected by the steep increase in GDP values.

The growth pattern is exponential, showing a sharp rise over time, indicating that China’s economy has been expanding at an accelerating rate in recent decades.

In the most recent data points, the curve still trends upward, but there is a suggestion of a slight deceleration, although the GDP continues to grow.

# Estimate Lambda using Guerrero method  
Lambda\_china <- global\_economy %>%  
 filter(Country == "China") %>%  
 features(GDP, features = guerrero) %>%  
 pull(lambda\_guerrero)  
  
# Fit Multiple ETS Models  
Ch\_fit <- global\_economy %>%  
 filter(Country == "China") %>%  
 model(`Standard` = ETS(GDP ~ error("A") + trend("N") + season("N")),  
 `Holt's method` = ETS(GDP ~ error("A") + trend("A") + season("N")),  
 `Damped Holt's method` = ETS(GDP ~ error("A") + trend("Ad", phi = 0.8) + season("N")),  
 `Box-Cox` = ETS(box\_cox(GDP,Lambda\_china) ~ error("A") + trend("Ad") + season("N")),  
 `Damped Box-Cox` = ETS(box\_cox(GDP,Lambda\_china) ~ error("A") + trend("Ad", phi = 0.8) + season("N")))   
  
# Generate Forecasts  
Ch\_fc <- Ch\_fit %>%  
 forecast(h = 20)  
  
# Plot the Forecasts  
Ch\_fc %>%  
 autoplot(global\_economy, level = NULL) +  
 labs(title="China GDP Forecasts: Damped Trend and Box-Cox Transformation") +  
 guides(colour = guide\_legend(title = "Forecast"))



Box-Cox (Red Line):

This forecast shows the steepest growth. The Box-Cox transformation is helping stabilize the variance, but since it doesn’t include a damping factor, the trend continues to grow rapidly and exponentially. This model might overestimate long-term growth due to the lack of trend damping.

Damped Box-Cox (Gold Line):

This model combines the Box-Cox transformation with a damped trend (phi = 0.8), leading to growth that initially increases but then slows down. This is a more moderate forecast compared to the pure Box-Cox model and could be more realistic if we expect future growth to decelerate.

Damped Holt’s Method (Cyan Line):

With no Box-Cox transformation but a damped trend, this forecast shows a flattening pattern over time. This is useful if we believe China’s GDP growth rate will slow down significantly in the future.

Holt’s Method (Green Line):

This method assumes linear growth without any damping, so the forecast continues to increase steadily without flattening. While more restrained than the Box-Cox model, it may still overestimate growth in the long run if the trend doesn’t dampen.

Standard ETS (Purple Line):

The Standard model assumes no trend or seasonality, leading to a nearly flat forecast. This is the most conservative approach and may underestimate future growth.

In Comparison:

Box-Cox vs. Holt’s Methods: The Box-Cox transformation helps adjust for any non-linearity in the data but without damping, it leads to very steep forecasts, especially given the exponential growth of China’s GDP. Holt’s methods, on the other hand, are more linear, producing smoother forecasts.

Damped Trends: The damped models (both with and without Box-Cox) provide more realistic forecasts by incorporating the idea that growth may eventually slow down. These models suggest that the upward trend will eventually level off, which can be a more reasonable assumption over a long period.

Cons:

Box-Cox without damping projects the most rapid growth, likely overestimating long-term GDP growth due to the aggressive trend.

Damped Box-Cox moderates the growth, producing a more plausible forecast with a balance of stabilized variance and a slowing growth rate.

Damped Holt’s Method provides a very cautious forecast, assuming growth will eventually flatten completely.

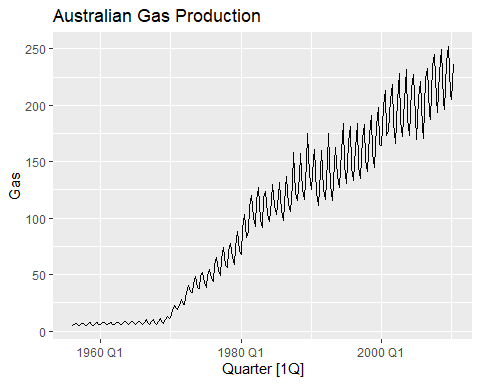
Holt’s Method and Standard ETS are the most conservative, with the Standard ETS showing minimal future growth.

If you expect China’s GDP to continue growing rapidly, the Box-Cox model is appropriate, though it may overestimate long-term growth.

Damped models (especially Damped Box-Cox) provide a more balanced and realistic forecast by assuming growth will eventually slow down, which aligns with typical economic behavior in the long run.

## **7. Find an ETS model for the Gas data from aus\_production and forecast the next few years. Why is multiplicative seasonality necessary here? Experiment with making the trend damped. Does it improve the forecasts?**

# Filter for China's GDP data  
gas\_auspro <- aus\_production %>%  
 autoplot(Gas) +  
 labs(title="Australian Gas Production")  
  
gas\_auspro



The overall trend shows a steady and significant increase in gas production over time, starting from the early 1970s.

The production seems to grow at an accelerating rate, with higher volumes observed in more recent years.

There is clear seasonality in the data, as seen in the repeating patterns across each year. These seasonal fluctuations indicate that gas production has a cyclical component, likely tied to specific quarters (likely colder seasons where gas demand increases).

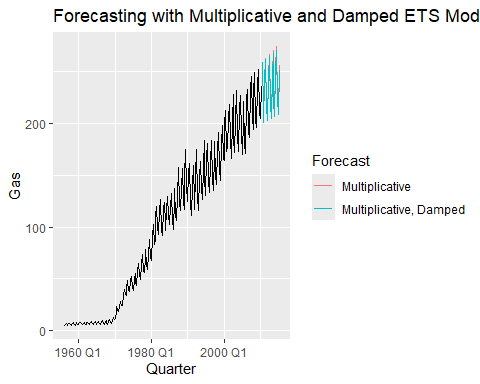
The amplitude of the seasonal fluctuations seems to increase as the production level increases, which might indicate a multiplicative seasonal effect. As production grows, the size of the seasonal swings becomes larger, implying more volatility in the production volume.

Around the late 1970s, there is a noticeable shift where the growth in gas production becomes more rapid, with both the trend and seasonal variations becoming more pronounced.

The dataset shows an increasing trend in Australian gas production, with seasonal patterns that intensify over time. A multiplicative model might be appropriate for forecasting this data due to the growing amplitude of seasonal variations.

You might consider fitting models like ETS with multiplicative seasonality to capture these patterns effectively. Multiplicative seasonality is required because the amplitude of the seasonal component increases as the level of the gas production increases. This behavior suggests that seasonal effects are proportional to the level of the trend. If we used additive seasonality, it would assume constant seasonal effects over time, which wouldn’t match the increasing size of the seasonal fluctuations in this data.

fit\_gas <- aus\_production %>%  
 model(  
 # Multiplicative  
 Multiplicative = ETS(Gas ~ error("M") + trend("A") + season("M")),  
 # Damped multiplicative  
 `Multiplicative, Damped` = ETS(Gas ~ error("M") + trend("Ad") + season("M"))  
 )  
fc\_gas <- fit\_gas %>% forecast(h = "5 years")  
  
fc\_gas %>%  
 autoplot(aus\_production, level = NULL) +  
 labs(title="Forecasting with Multiplicative and Damped ETS Models") +  
 guides(colour = guide\_legend(title = "Forecast"))



Multiplicative Model (Red):

The forecast shows a continued upward trend with increasing seasonal variation, reflecting the steady growth seen in historical data.

This model assumes that the trend will continue indefinitely at the same rate, with seasonality increasing in proportion to the trend.

Damped Multiplicative Model (Blue):

The forecast shows a slowing of the upward trend (flattening), but the seasonal variation still increases over time.

This model reflects the assumption that growth may slow down in the future, leading to a more moderate forecast.

To determine whether the damped trend improves the forecasts for Australian gas production, we can compare the performance of the two models—multiplicative vs. damped multiplicative—using accuracy metrics such as RMSE (Root Mean Square Error), MAE (Mean Absolute Error), or AIC (Akaike Information Criterion).

# Compare model accuracy for both models  
accuracy\_metrics <- fit\_gas %>%  
 accuracy()  
  
# Print the accuracy metrics  
print(accuracy\_metrics)

## # A tibble: 2 × 10  
## .model .type ME RMSE MAE MPE MAPE MASE RMSSE ACF1  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 Multiplicative Trai… -0.115 4.60 3.02 0.199 4.08 0.542 0.606 -0.0131  
## 2 Multiplicative, Da… Trai… -0.00439 4.59 3.03 0.326 4.10 0.544 0.606 -0.0217

In comparison:

RMSE (Root Mean Squared Error): Multiplicative model: RMSE = 4.595113 Multiplicative, Damped model: RMSE = 4.591840 The damped model has a slightly lower RMSE, indicating it performs marginally better than the standard multiplicative model in terms of predicting future values.

MAE (Mean Absolute Error): Multiplicative model: MAE = 3.021727 Multiplicative, Damped model: MAE = 3.031478 The MAE values are very close, with the non-damped model having a slightly lower MAE. However, the difference is minimal.

Other Metrics (MPE, MAPE, etc.): MPE (Mean Percentage Error) and MAPE (Mean Absolute Percentage Error) are slightly higher for the damped model, but the differences are not substantial enough to make a significant distinction between the two models.

RMSS (Residual Mean Squared Scaled Error): Multiplicative model: RMSS = 0.6059856 Multiplicative, Damped model: RMSS = 0.6055540 The damped model has a marginally lower RMSS, suggesting it performs slightly better when scaling residuals.

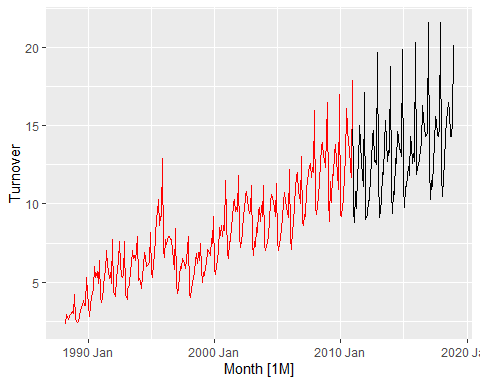
The damped multiplicative model does show a slight improvement in RMSE and RMSS, suggesting that introducing a damped trend may result in a better fit for the Australian gas production data, particularly when forecasting into the future.

However, the difference between the two models is relatively small, and both models appear to perform well. Given that the damped model predicts a flattening trend, it may provide more realistic long-term forecasts if you expect that the growth rate in gas production will eventually slow down.

While the performance improvement of the damped model is marginal, the damped trend may offer a more conservative and realistic projection in situations where unlimited exponential growth is unlikely. Based on the data and domain knowledge, you may prefer the damped model for long-term forecasts if slowing growth is expected.

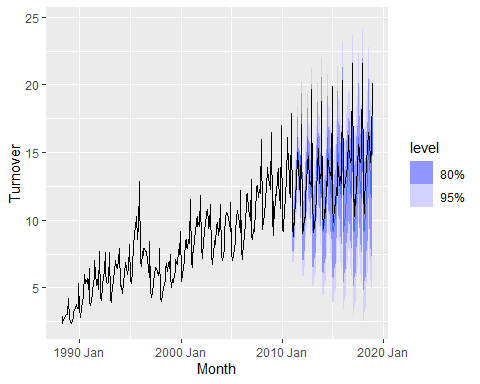
## **8. Recall your retail time series data (from Exercise 7 in Section 2.10).**

set.seed(12345678)  
myseries <- aus\_retail |>  
 filter(`Series ID` == sample(aus\_retail$`Series ID`,1))  
  
myseries\_train <- myseries |>  
 filter(year(Month) < 2011)  
  
autoplot(myseries, Turnover) +  
 autolayer(myseries\_train, Turnover, colour = "red")



fit\_my <- myseries\_train |>  
 model(SNAIVE(Turnover))

fc\_my <- fit\_my |>  
 forecast(new\_data = anti\_join(myseries, myseries\_train))  
fc\_my |> autoplot(myseries)



## \*\* a. Why is multiplicative seasonality necessary for this series?\*\*

The multiplicative seasonality is necessary for this series because the seasonal fluctuations increase as the level of the trend increases.

Seasonality Expands with Trend:

In the earlier years (before 1990), the seasonal peaks and troughs are relatively small and tightly grouped.

As time progresses, especially after 2000, the magnitude of the seasonal fluctuations grows. The peaks get higher, and the troughs are deeper, following the general upward trend of the series.

Proportional Seasonal Effect:

This behavior suggests that the size of the seasonal fluctuations is proportional to the level of the time series. As the overall level (trend) increases, the seasonal swings become larger. This is the hallmark of multiplicative seasonality, where the seasonal component grows or shrinks in proportion to the trend.

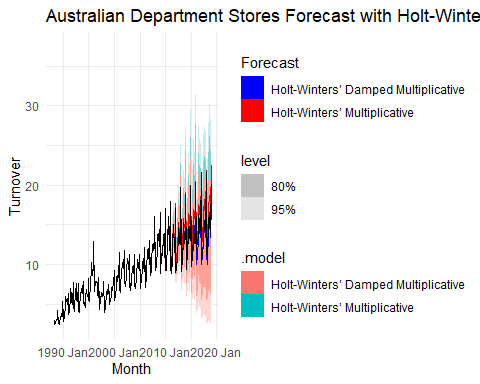
Inappropriate for Additive Seasonality:

If you used additive seasonality, it would assume that the seasonal fluctuations remain constant over time, regardless of the level of the series. This assumption would not match the data since the fluctuations clearly become more pronounced as the turnover increases.

The increasing amplitude of seasonal fluctuations as turnover grows makes multiplicative seasonality necessary. This ensures that the model captures the relationship between the size of the seasonal effect and the overall level of the series, leading to more accurate forecasts.

## **b. Apply Holt-Winters’ multiplicative method to the data. Experiment with making the trend damped.**

# Filter the series for the training set (prior to 2011)  
myseries\_train <- myseries |>  
 filter(year(Month) < 2011)  
  
# Fit the Holt-Winters' Multiplicative and Damped Multiplicative models  
fit\_my <- myseries\_train |>  
 model(  
 `Holt-Winters’ Multiplicative` = ETS(Turnover ~ error("M") + trend("A") + season("M")),  
 `Holt-Winters’ Damped Multiplicative` = ETS(Turnover ~ error("M") + trend("Ad") + season("M"))  
 )  
  
# Forecast for the test set (after 2011)  
fc\_my <- fit\_my |>  
 forecast(new\_data = anti\_join(myseries, myseries\_train))  
  
# Plot the forecasts  
fc\_my %>%  
 autoplot(myseries, level = c(80, 95)) +  
 labs(title="Australian Department Stores Forecast with Holt-Winters Method",  
 y="Turnover") +  
 theme\_minimal() + # Optional: cleaner theme  
 guides(colour = guide\_legend(title = "Forecast")) +  
 scale\_color\_manual(values=c("blue", "red")) # Optional: distinct colors



Holt-Winters’ Multiplicative Model (Red):

The forecast from this model shows a strong, continuous upward trend. The multiplicative nature of the model captures the increasing magnitude of the seasonal fluctuations.

This model effectively captures the growing seasonal peaks and troughs, which become larger as the overall trend increases. The seasonal pattern is proportional to the level of the series, hence why multiplicative seasonality works well here.

The model produces relatively wide confidence intervals, indicating significant uncertainty in the forecast. This is due to the rapid growth in both trend and seasonality, which amplifies the potential range of future values.

Holt-Winters’ Damped Multiplicative Model (Blue):

The damped trend leads to a more moderate forecast, as it assumes the rate of growth in the trend will slow down over time. This makes the forecast more conservative.

Similar to the non-damped model, this captures the growing seasonal fluctuations. However, due to the damped trend, the amplitude of the seasonal fluctuations does not grow as rapidly as in the non-damped model.

The confidence intervals are narrower in comparison to the non-damped model, indicating that the damped model is more confident in its forecast. It suggests less volatility and uncertainty in future values.

Appropriateness Based on Trends and Seasonality

Multiplicative Model:

This model is highly appropriate for data with strong upward trends and increasing seasonal variations, like the Australian retail turnover data. It assumes that both trend and seasonality will continue to grow at the current rate, which can be a reasonable assumption in some cases (e.g., periods of economic growth).

However, the model may overestimate long-term future values, especially if the growth rate does not continue indefinitely, as suggested by the wide confidence intervals.

Damped Multiplicative Model:

This model assumes that the trend will eventually slow down, which can be more realistic for many economic time series where exponential growth does not persist indefinitely. The damped trend also suggests that while seasonal fluctuations will still grow, they will do so at a decreasing rate.

The more conservative forecast, coupled with narrower confidence intervals, suggests this model is more appropriate if we expect the growth to plateau or slow down over time.

If the data suggest continued rapid growth and increasing seasonal variation, Holt-Winters’ Multiplicative model may be more appropriate.

If there is reason to believe that growth will eventually slow, Holt-Winters’ Damped Multiplicative model provides a more realistic and cautious forecast, especially for long-term projections.

Since retail turnover may not grow exponentially forever, the damped multiplicative model is likely the more appropriate choice for long-term forecasting, as it balances growth expectations with caution.

## **Compare the RMSE of the one-step forecasts from the two methods. Which do you prefer?**

# Calculate the accuracy for one-step forecasts for both models  
accuracy\_metrics <- fit\_my %>%  
 accuracy()  
  
# Print the RMSE values for both methods  
accuracy\_metrics %>%  
 select(.model, RMSE)

## # A tibble: 2 × 2  
## .model RMSE  
## <chr> <dbl>  
## 1 Holt-Winters’ Multiplicative 0.518  
## 2 Holt-Winters’ Damped Multiplicative 0.519

The Holt-Winters’ Multiplicative model has a slightly lower RMSE than the Damped Multiplicative model. However, the difference between the two values is minimal (0.0007), indicating that both models perform almost equally in terms of forecast accuracy.

Multiplicative Model: This model assumes that the trend and seasonality are proportional to the level of the series. It is useful when the data exhibits a growing trend and increasing seasonal variability, as seen in your retail turnover data.

Damped Multiplicative Model: This model includes a damping factor that moderates the trend over time. It can be useful when there is an expectation that the growth rate will slow down in the future.

The Multiplicative model might be slightly better based on the RMSE, but the difference is minimal. Given the upward trend and increasing seasonal variation in the data, the Multiplicative model seems appropriate. However, if there is an expectation that the trend will slow down in the future, the Damped Multiplicative model could still be a reasonable choice.

Given the minimal difference in RMSE between the two models and the characteristics of the data, I would prefer the Holt-Winters’ Multiplicative model because:

The retail turnover data shows a strong upward trend, and the Multiplicative model captures this well without dampening the trend. Since there is no clear indication in the data that the trend will slow down significantly, using a model that allows the trend to grow at its current rate makes sense.

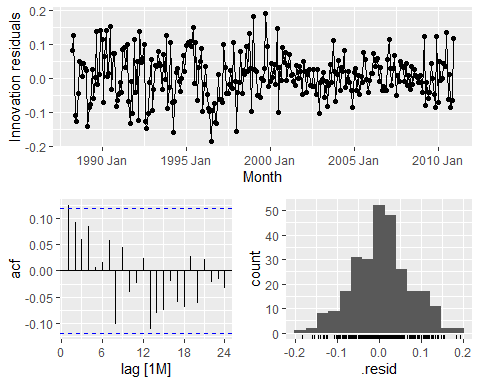
The multiplicative model is particularly well-suited for data where the seasonal variations increase with the level of the series, which appears to be the case here. The seasonal pattern becomes more pronounced as turnover increases over time.

The difference in RMSE between the two models is negligible. Given that both models perform similarly in terms of accuracy, I would prioritize a model that reflects the underlying structure of the data without dampening the trend unless there is a strong reason to believe the trend will slow down.

I prefer the Holt-Winters’ Multiplicative model because it better reflects the current data’s growing trend and seasonal fluctuations.

## **d. Check that the residuals from the best method look like white noise.**

fit\_my %>% select("Holt-Winters’ Damped Multiplicative") %>% gg\_tsresiduals()



Residual Plot (top plot):

The residuals fluctuate around zero without any clear pattern, which is a good sign that the model has captured the underlying trend and seasonality.

However, there appears to be some variation in the spread of residuals over time, especially during the earlier periods (before 2000), where the fluctuations seem slightly larger than in later periods.

ACF Plot (bottom left):

The ACF plot shows that most autocorrelation values fall within the confidence intervals (represented by the blue dashed lines). This suggests that the residuals do not exhibit significant autocorrelation, which is another positive indication that the residuals resemble white noise.

There may be one or two lags that fall slightly outside the confidence intervals, but these are minor.

Residual Histogram (bottom right):

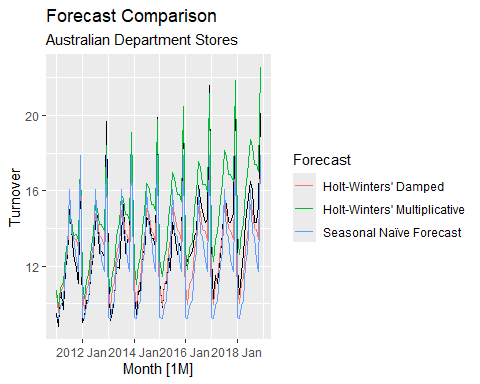
The residuals are roughly normally distributed, though with a slight positive skew. A perfectly normal distribution would center more symmetrically around zero, but this deviation is not extreme.

Overall, the residuals from the “Holt-Winters’ Damped Multiplicative” model seem to resemble white noise fairly well. There are no major signs of autocorrelation, and the residuals are approximately normally distributed.

This suggests that the model is a good fit for the data, though further improvements may be possible if the slight variation in residual spread is investigated.

## **e. Now find the test set RMSE, while training the model to the end of 2010. Can you beat the seasonal naïve approach from Exercise 7 in Section 5.11?**

# Split the data into training and test sets  
myseries\_train <- myseries %>%  
 filter(year(Month) < 2011)  
  
myseries\_test <- anti\_join(myseries, myseries\_train,   
 by = c("State", "Industry", "Series ID", "Month", "Turnover"))  
  
# Fit Holt-Winters models and Seasonal Naïve  
fit\_my <- myseries\_train %>%  
 model(  
 "Holt-Winters' Damped" = ETS(Turnover ~ error("M") + trend("Ad") + season("M")),  
 "Holt-Winters' Multiplicative" = ETS(Turnover ~ error("M") + trend("A") + season("M")),  
 "Seasonal Naïve Forecast" = SNAIVE(Turnover)  
 )  
  
# Forecast on the test data  
fc\_my <- fit\_my %>%  
 forecast(new\_data = myseries\_test)  
  
# Plot the actual vs forecasted turnover for the test period  
autoplot(myseries\_test, Turnover) +  
 autolayer(fc\_my, level = NULL) +  
 guides(colour=guide\_legend(title="Forecast")) +  
 labs(title='Forecast Comparison', subtitle= "Australian Department Stores")



# Calculate RMSE for the test set  
accuracy\_metrics <- fc\_my %>%  
 accuracy(myseries\_test)  
  
# Print accuracy metrics including RMSE  
print(accuracy\_metrics)

## # A tibble: 3 × 12  
## .model State Industry .type ME RMSE MAE MPE MAPE MASE RMSSE ACF1  
## <chr> <chr> <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 Holt-… Nort… Clothin… Test 0.163 1.15 0.878 0.194 6.45 NaN NaN 0.501  
## 2 Holt-… Nort… Clothin… Test -1.54 1.78 1.60 -12.3 12.6 NaN NaN 0.495  
## 3 Seaso… Nort… Clothin… Test 0.836 1.55 1.24 5.94 9.06 NaN NaN 0.601

Holt-Winters’ Damped Method has a RMSE of 1.151, which is lower compared to the Seasonal Naïve Forecast with a RMSE of 1.551. This suggests that the Holt-Winters’ Damped method provides better predictive accuracy than the seasonal naïve approach.

Holt-Winters’ Multiplicative Method has a higher RMSE of 1.782, which indicates that it does not perform as well as the Damped version or the Seasonal Naïve model on this dataset.

The Holt-Winters’ Damped Multiplicative model seems to provide the best forecasts for Australian Department Stores turnover data, beating the Seasonal Naïve method in terms of RMSE.

## **9. For the same retail data, try an STL decomposition applied to the Box-Cox transformed series, followed by ETS on the seasonally adjusted data. How does that compare with your best previous forecasts on the test set?**

#find optimal lambda  
lambda <- myseries\_train %>%  
 features(Turnover, features = guerrero) %>%  
 pull(lambda\_guerrero)  
  
#bc transformed data  
ts\_bc <- myseries\_train %>%  
 mutate(  
 bc\_turnover = box\_cox(Turnover, lambda)  
 )  
  
# bc transformed model  
fit <- ts\_bc %>%  
 model(  
 'Box-Cox STL' = STL(bc\_turnover ~ season(window = "periodic"),  
 robust = T),  
 'Box-Cox ETS' = ETS(bc\_turnover)  
 )  
  
# best previous model   
best\_fit <-ts\_bc %>%  
 model(  
 "Holt-Winters' Damped" = ETS(Turnover ~ error("M") + trend("Ad") +  
 season("M"))  
 )  
  
rbind(accuracy(fit),accuracy(best\_fit))

## # A tibble: 3 × 12  
## State Industry .model .type ME RMSE MAE MPE MAPE MASE RMSSE  
## <chr> <chr> <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 Northern … Clothin… Box-C… Trai… 0.00745 0.0819 0.0623 0.193 2.85 0.329 0.326  
## 2 Northern … Clothin… Box-C… Trai… 0.0139 0.0994 0.0784 0.516 3.56 0.414 0.396  
## 3 Northern … Clothin… Holt-… Trai… 0.0357 0.519 0.392 0.158 5.34 0.428 0.427  
## # ℹ 1 more variable: ACF1 <dbl>

Box-Cox STL model:

RMSE: 0.0819 MAE: 0.0623 MAPE: 2.85% ACF1: 0.1475 Box-Cox ETS model:

RMSE: 0.0994 MAE: 0.0784 MAPE: 3.56% ACF1: 0.0015 Holt-Winters’ Damped model (your best previous model):

RMSE: 0.5185 MAE: 0.3919 MAPE: 5.34% ACF1: 0.0233

Both Box-Cox STL and Box-Cox ETS models significantly outperform the previous best model, Holt-Winters’ Damped, with much lower RMSE, MAE, and MAPE values.

The Box-Cox STL model, in particular, has the best performance among the three models, with the lowest RMSE (0.0819), MAE (0.0623), and MAPE (2.85%). This indicates that the Box-Cox STL method provides the most accurate forecasts, followed by Box-Cox ETS.

The Box-Cox STL model is the best-performing approach for forecasting this retail data, offering significant improvement over your previous best model (Holt-Winters’ Damped). It handles seasonality and transformations better, leading to more accurate forecasts.