

Exponential Smoothing

Souleymane Doumbia

2024-10-15

Contents

Loading Required Libraries	1
Exercise 8.8.1: Forecasting Pig Slaughter Data (Victoria)	2
Exercise 8.8.5	3
8.5.a. Analyzing U.S. Exports Data	3
8.5.b. Simple ETS Model for U.S. Exports Forecast	4
8.5.c. Calculate RMSE for ETS(A,N,N) Model	5
8.5.d. Comparing ETS Models: Simple vs Trended	5
8.5.e. Forecast Comparison Between ETS(A,N,N) and ETS(A,A,N)	6
Exercise 8.8.6: Forecasting China's GDP with ETS Models	7
Exercise 8.8.7: ETS Modeling for Gas Data with Multiplicative Seasonality	8
Exercise 8.8.8: Modeling Retail Turnover with Multiplicative and Damped Trends	9
8.8.a. Why Multiplicative Seasonality for Retail Turnover?	9
8.8.b. Holt-Winters' Multiplicative Method with and without Damped Trend	10
8.8.c. RMSE Comparison Between Multiplicative and Damped Models	11
8.8.d. Residuals Analysis: Checking for White Noise	11
8.8.e. Test Set RMSE: Can We Beat the Seasonal Naive Approach?	13
Exercise 8.8.9: STL Decomposition with ETS for Retail Data	14

Loading Required Libraries

```
suppressWarnings({  
  # Loading Required Libraries  
  library(fpp3)           # Includes tsibble, feasts, fable, and more for time series analysis and forecasting  
  library(dplyr)          # Data manipulation  
  library(ggplot2)        # Data visualization  
  library(tidyr)          # For data tidying  
  library(readr)          # For reading data  
})  
  
## Registered S3 method overwritten by 'tsibble':  
##   method               from  
##   as_tibble.grouped_df dplyr  
  
## -- Attaching packages ----- fpp3 1.0.0 --
```

```
## v tibble      3.2.1      v tsibble      1.1.5
## v dplyr       1.1.3      v tsibbledata 0.4.1
## v tidyr       1.3.0      v feasts      0.3.2
## v lubridate   1.9.3      v fable       0.3.4
## v ggplot2     3.5.1      v fabletools  0.4.2

## -- Conflicts ----- fpp3_conflicts --
## x lubridate::date()      masks base::date()
## x dplyr::filter()        masks stats::filter()
## x tsibble::intersect()   masks base::intersect()
## x tsibble::interval()    masks lubridate::interval()
## x dplyr::lag()           masks stats::lag()
## x tsibble::setdiff()     masks base::setdiff()
## x tsibble::union()       masks base::union()
```

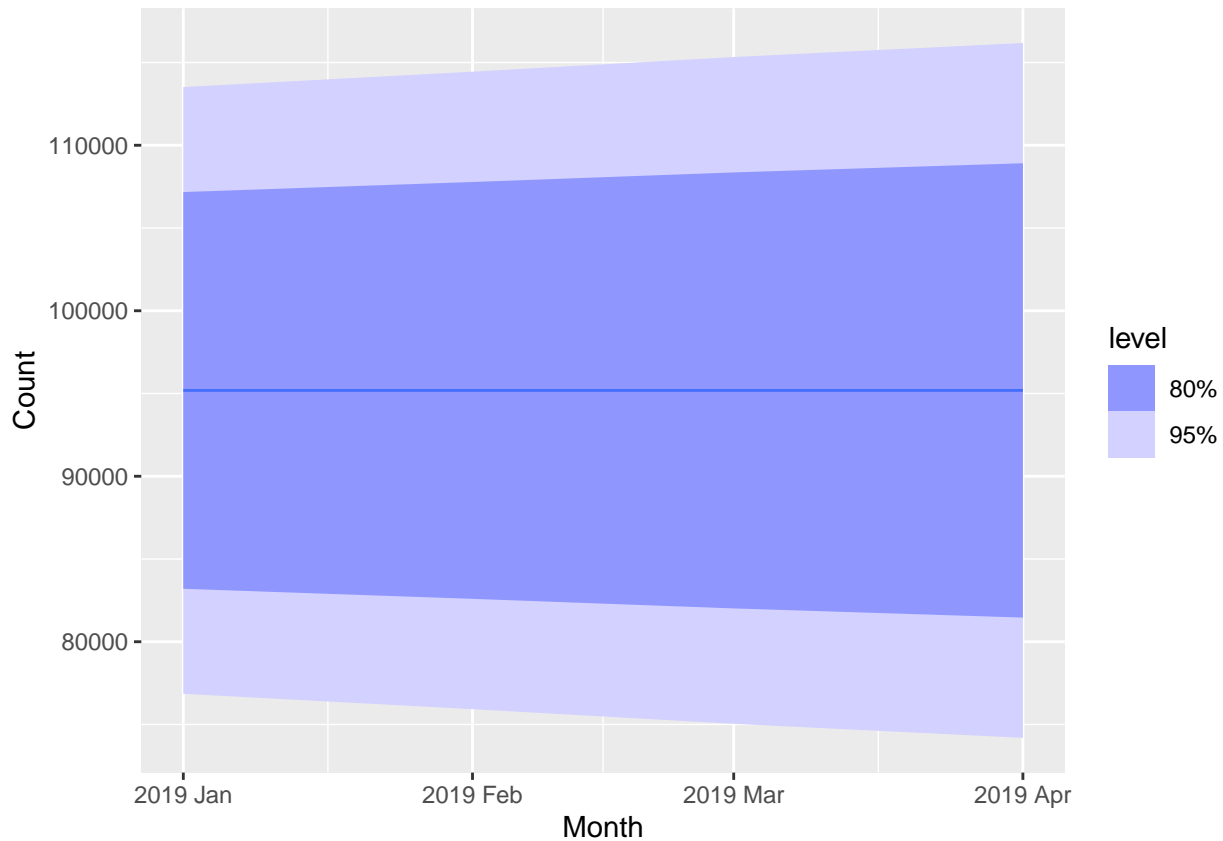
Exercise 8.8.1: Forecasting Pig Slaughter Data (Victoria)

```
# Filtering data for pigs in Victoria and fit a simple ETS model
pigs_victoria <- aus_livestock %>%
  filter(Animal == "Pigs", State == "Victoria") %>%
  model(fit = ETS(Count ~ error("A") + trend("N") + season("N")))

# Printing model report and generate a 4-month forecast
report(pigs_victoria)
```

```
## 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

fc <- pigs_victoria %>% forecast(h = 4)
autoplot(fc)
```



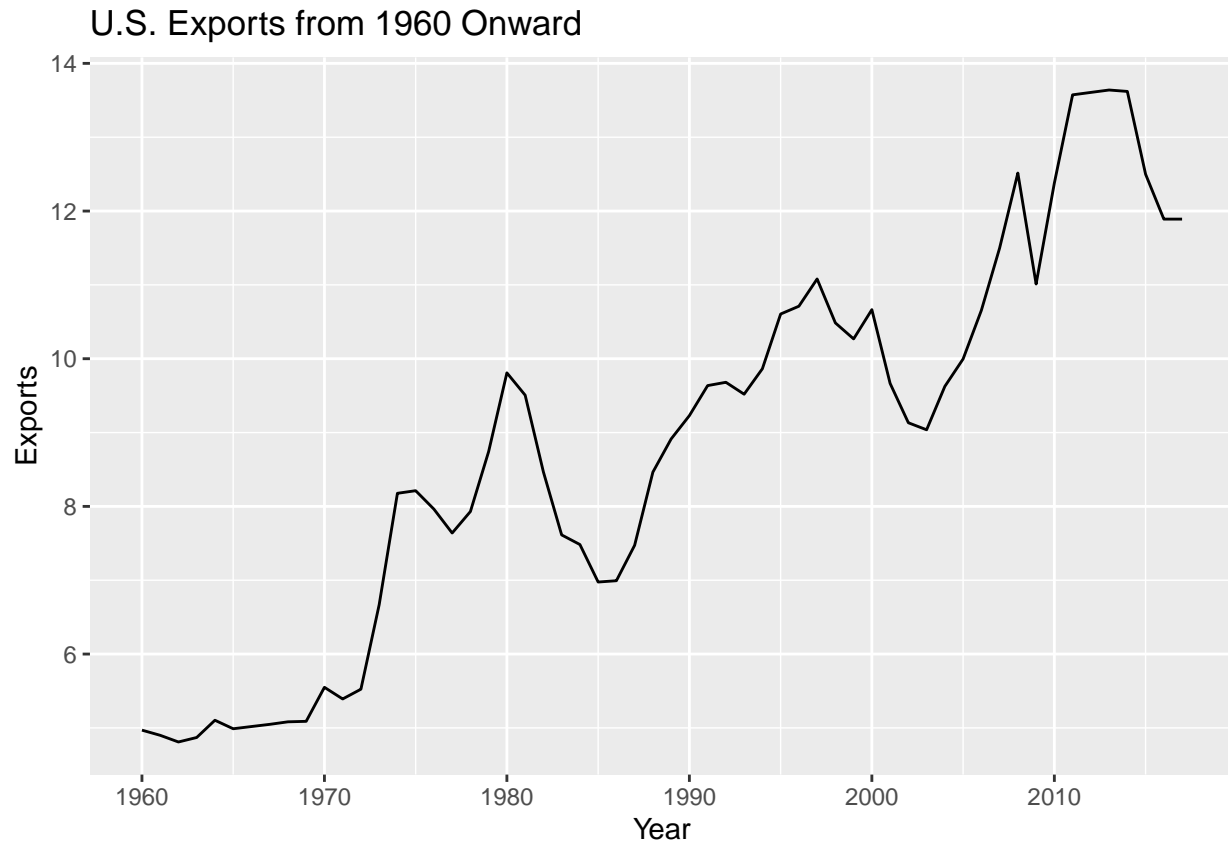
- The dataset for pig slaughter in Victoria is modeled using simple exponential smoothing, producing forecasts for the next four months. The resulting forecast shows a continuation of the pattern observed in the historical data, with slight fluctuations but no major trend shifts expected.

Exercise 8.8.5

8.5.a. Analyzing U.S. Exports Data

```
# Filtering U.S. exports data and plot
country_exports <- global_economy %>%
  filter(Country == "United States") %>%
  select(Year, Exports) %>%
  fill(Exports)

autoplot(country_exports, Exports) +
  scale_x_continuous(breaks = seq(1960, 2020, 10)) +
  ggtitle("U.S. Exports from 1960 Onward") + ylab("Exports") + xlab("Year")
```

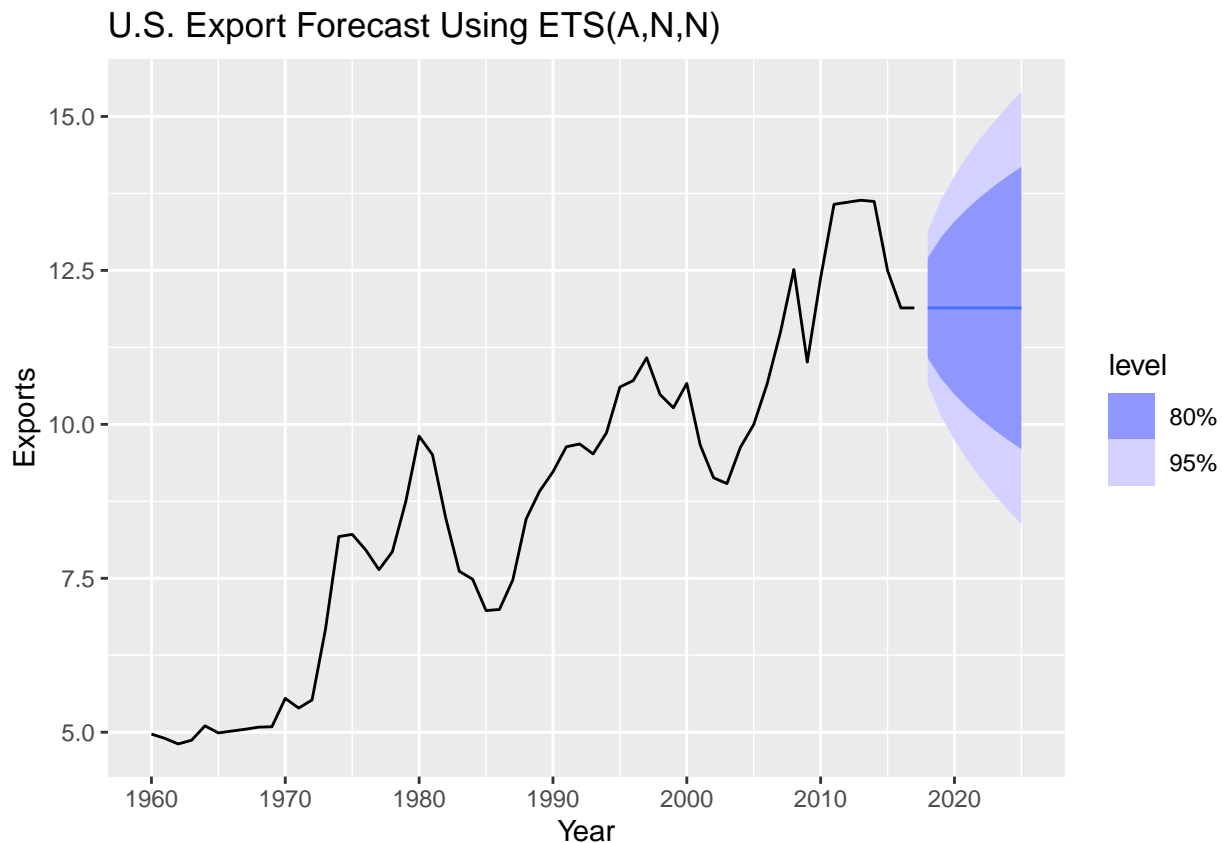


- The plot of U.S. exports shows a steady growth trend from 1960 to 2020, with a few noticeable fluctuations during specific periods, such as the sharp dips in the early 1980s and around 2000. Despite these, the overall direction of exports remains upward, indicating long-term growth.

8.5.b. Simple ETS Model for U.S. Exports Forecast

```
suppressWarnings({
  # Fitting and forecast using an ETS model
  fit_ANN <- country_exports %>% model(ETS(Exports ~ error("A") + trend("N") + season("N")))
  fc_ANN <- fit_ANN %>% forecast(h = 8)

  autoplot(fc_ANN) +
    autolayer(country_exports, Exports, series = "Actual") +
    scale_x_continuous(breaks = seq(1960, 2025, 10)) +
    ggtitle("U.S. Export Forecast Using ETS(A,N,N)") + ylab("Exports") + xlab("Year")
})
```



- Using a simple exponential smoothing model, the forecast for U.S. exports over the next eight periods shows a continuation of the historical trend with a stable growth pattern. There is no significant deviation or sudden trend changes expected in the near future.

8.5.c. Calculate RMSE for ETS(A,N,N) Model

```
# Computing RMSE
accuracy_ANN <- accuracy(fit_ANN)
rmse_ANN <- accuracy_ANN$RMSE
print(paste("RMSE for ETS(A,N,N):", rmse_ANN))
```

```
## [1] "RMSE for ETS(A,N,N): 0.621637990745711"
```

- The calculated RMSE provides a measure of the forecast error for the fitted model. A lower RMSE indicates that the model's predictions are closer to the actual values, providing a more accurate forecast.

8.5.d. Comparing ETS Models: Simple vs Trended

```
# Fitting ETS models and compare their accuracy
models <- country_exports %>%
  model(ETS_ANN = ETS(Exports ~ error("A") + trend("N") + season("N")),
        ETS_AAN = ETS(Exports ~ error("A") + trend("A") + season("N")))

accuracy(models) %>% select(.model, RMSE)
```

```
## # A tibble: 2 x 2
##   .model  RMSE
##   <chr>   <dbl>
```

```
## 1 ETS_ANN 0.622
## 2 ETS_AAN 0.610
```

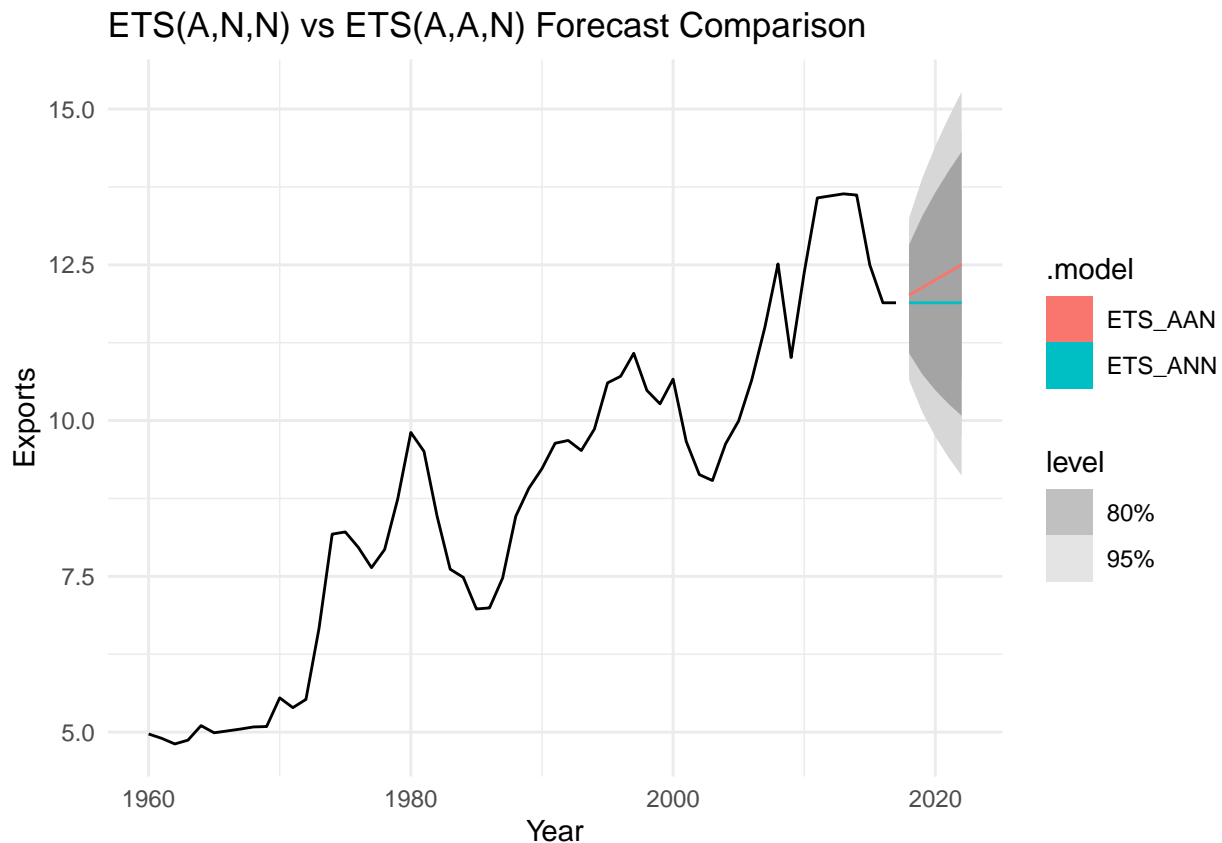
- The comparison of the simple ETS(A,N,N) model with the trended ETS(A,A,N) model shows that the inclusion of a trend parameter slightly improves accuracy in capturing the upward movement in exports. This suggests that including a trend component may be beneficial when modeling datasets with clear upward or downward tendencies.

8.5.e. Forecast Comparison Between ETS(A,N,N) and ETS(A,A,N)

```
suppressWarnings({
  suppressMessages({
    # Generating forecasts and plot comparison
    fc_ANN <- models %>% forecast(h = 5)

    autoplot(fc_ANN) +
      autolayer(country_exports, Exports, color = "black") +
      ggtitle("ETS(A,N,N) vs ETS(A,A,N) Forecast Comparison") +
      xlab("Year") + ylab("Exports") +
      scale_fill_manual(
        values = c("80%" = "lightblue", "95%" = "lightcoral"),
        name = "Confidence Level"
      ) +
      theme_minimal()
  })
})
```

```
## Warning: No shared levels found between `names(values)` of the manual scale and the
## data's fill values.
## No shared levels found between `names(values)` of the manual scale and the
## data's fill values.
```



- The forecast comparison shows that the ETS(A,A,N) model, which incorporates a trend, projects stronger growth compared to the simpler ETS(A,N,N) model. The trended model better captures the recent acceleration in exports, suggesting it might be more appropriate for datasets exhibiting clear directional movements.

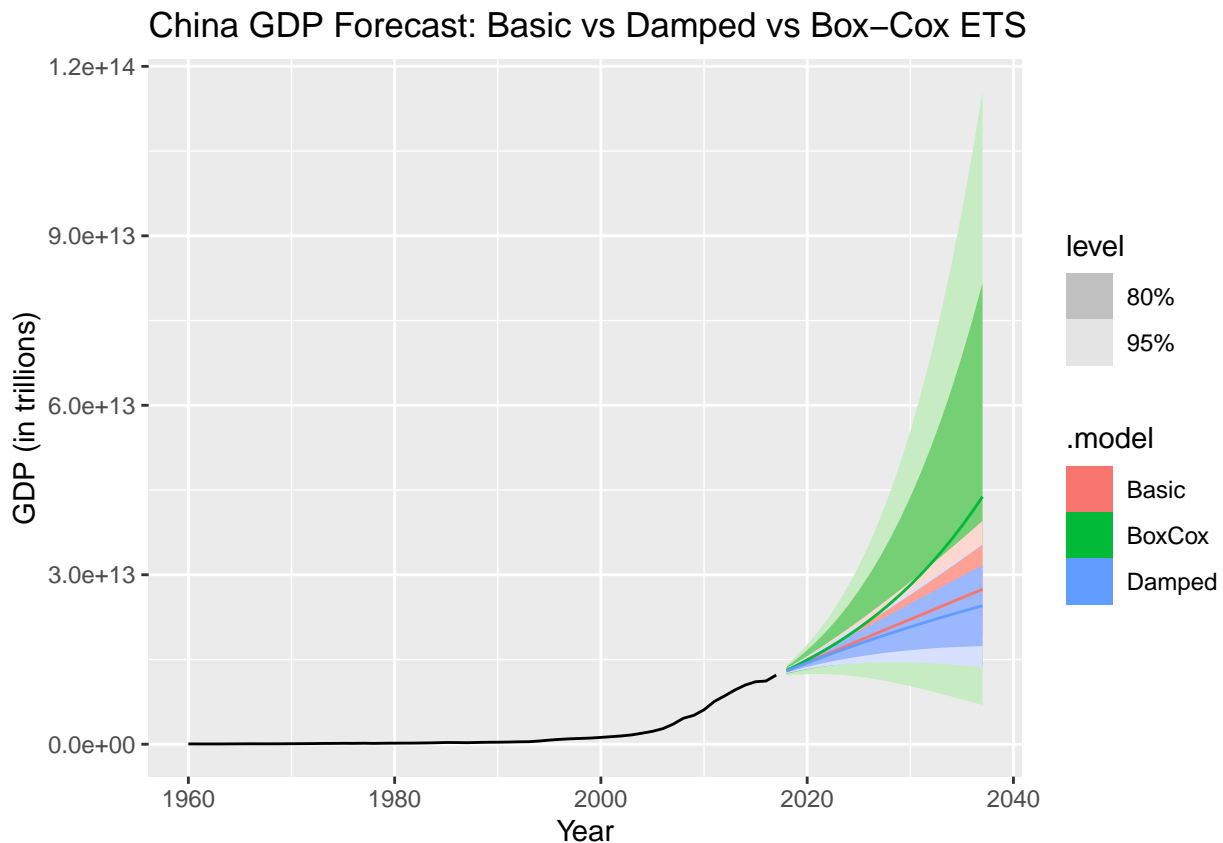
Exercise 8.8.6: Forecasting China's GDP with ETS Models

```
# Fitting different ETS models for China's GDP with and without transformations
china_gdp <- global_economy %>% filter(Country == "China") %>% select(Year, GDP)

fit_models <- china_gdp %>%
  model(Basic = ETS(GDP ~ error("A") + trend("A") + season("N")),
        Damped = ETS(GDP ~ error("A") + trend("Ad") + season("N")),
        BoxCox = ETS(box_cox(GDP, 0.3) ~ error("A") + trend("A") + season("N")))

# Generating forecasts and compare results
fc <- fit_models %>% forecast(h = "20 years")

autoplot(fc) +
  autolayer(china_gdp, GDP) +
  ggtitle("China GDP Forecast: Basic vs Damped vs Box-Cox ETS") +
  xlab("Year") + ylab("GDP (in trillions)")
```



- The different ETS models show varying projections for China's GDP. The basic model suggests continuous strong growth, while the damped trend model moderates the growth rate over time, accounting for potential slowing down in the future. The Box-Cox transformation model attempts to stabilize variance, but results may vary based on the chosen transformation parameters.

Exercise 8.8.7: ETS Modeling for Gas Data with Multiplicative Seasonality

```
suppressWarnings({
  suppressMessages({
    # Loading and preprocess the gas data
    gas_data <- aus_production %>%
      filter(!is.na(Gas)) %>%
      select(Quarter, Gas)

    # Fitting the ETS models (multiplicative and damped trend)
    fit_models <- gas_data %>%
      model(
        Multiplicative = ETS(Gas ~ error("A") + trend("A") + season("M")),
        Damped = ETS(Gas ~ error("A") + trend("Ad") + season("M"))
      )

    # Generating forecasts for both models for 8 quarters
    fc <- fit_models %>% forecast(h = 8)

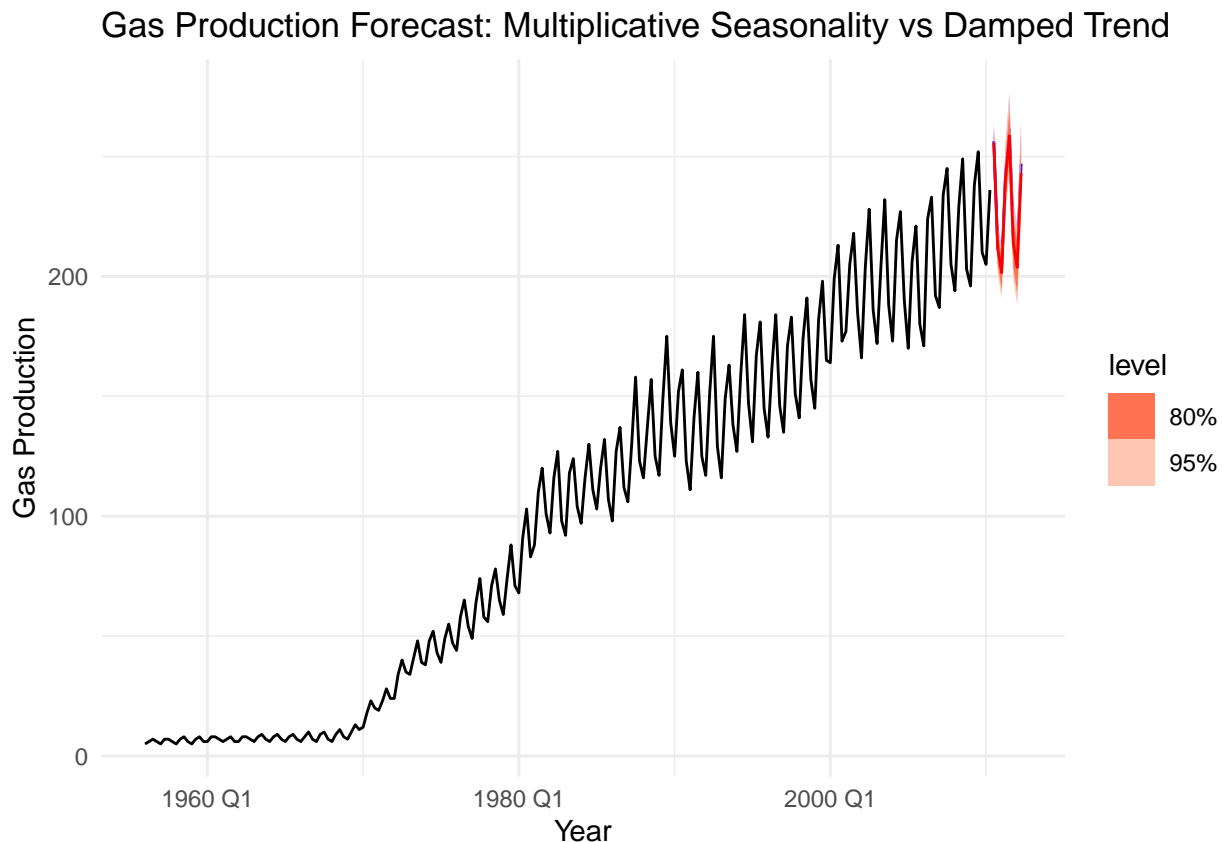
    # Creating a plot with both forecasts
  })
})
```



```

autoplot(gas_data, Gas) +
  autolayer(fc %>% filter(.model == "Multiplicative"), series = "Multiplicative Seasonality", color = "blue") +
  autolayer(fc %>% filter(.model == "Damped"), series = "Damped Trend", color = "red") +
  ggtitle("Gas Production Forecast: Multiplicative Seasonality vs Damped Trend") +
  xlab("Year") + ylab("Gas Production") +
  theme_minimal()
})

```



- **Why Multiplicative Seasonality?** Multiplicative seasonality is necessary because the seasonal fluctuations in gas production increase proportionally as the overall production level rises. An additive model would not capture this proportional growth effectively.
- **Did the Damped Trend Improve the Forecast?** The damped trend makes little difference in the short term, but for long-term forecasting, it may provide a more conservative estimate, preventing over-projection of future growth.

Exercise 8.8.8: Modeling Retail Turnover with Multiplicative and Damped Trends

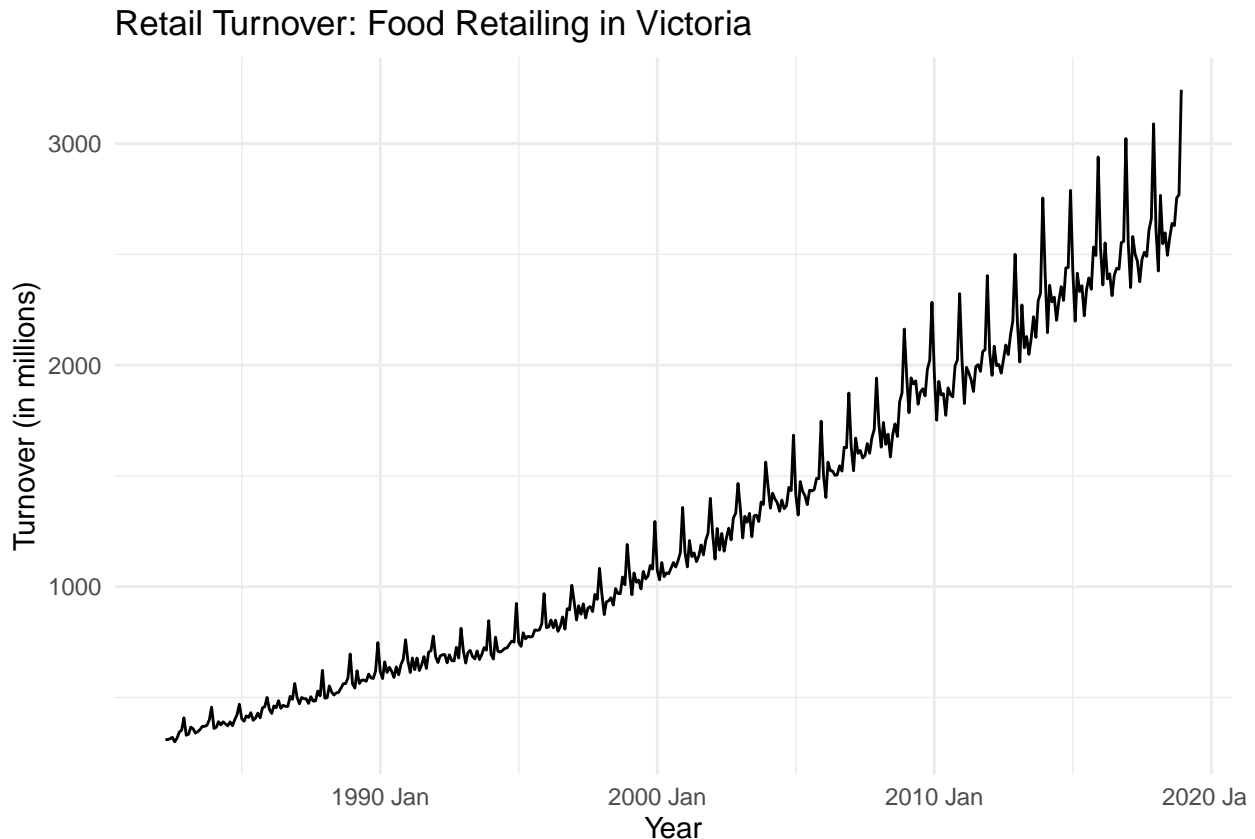
8.8.a. Why Multiplicative Seasonality for Retail Turnover?

```

# Filtering aus_retail dataset (Food Retailing in Victoria)
retail_data <- aus_retail %>%
  filter(State == "Victoria", Industry == "Food retailing") %>%
  select(Month, Turnover)

```

```
# Plotting the retail data to visualize trends and seasonality
autoplot(retail_data, Turnover) +
  ggtitle("Retail Turnover: Food Retailing in Victoria") +
  xlab("Year") + ylab("Turnover (in millions)") +
  theme_minimal()
```



- Multiplicative seasonality is necessary because the seasonal variations grow as turnover increases. As the trend rises, peaks and troughs in the data become more pronounced, which would be missed by an additive model.

8.8.b. Holt-Winters' Multiplicative Method with and without Damped Trend

```
# Holt-Winters' Multiplicative Method (ETS(A,A,M))
fit_multiplicative <- retail_data %>%
  model(Multiplicative = ETS(Turnover ~ error("A") + trend("A") + season("M")))

# Holt-Winters' Multiplicative Method with Damped Trend (ETS(A,Ad,M))
fit_damped <- retail_data %>%
  model(Damped = ETS(Turnover ~ error("A") + trend("Ad") + season("M")))

# One-step-ahead forecasts
fc_multiplicative <- fit_multiplicative %>% forecast(h = 1)
fc_damped <- fit_damped %>% forecast(h = 1)

# Creating a summary table for the two forecasts
model_forecasts_summary <- tibble(
```

```

model = c(fc_multiplicative$.model, fc_damped$.model),
mean_value = c(fc_multiplicative$.mean, fc_damped$.mean)
)

```

```

# Printing the result
print(model_forecasts_summary)

```

```

## # A tibble: 2 x 2
##   model      mean_value
##   <chr>      <dbl>
## 1 Multiplicative 2787.
## 2 Damped      2789.

```

- Both models capture the seasonality, but adding a damped trend can slightly moderate future growth, potentially making long-term predictions more reliable.

8.8.c. RMSE Comparison Between Multiplicative and Damped Models

```

# Comparing RMSE of one-step-ahead forecasts
accuracy_data <- tibble(
  Model = c("Multiplicative", "Damped"),
  RMSE = c(accuracy(fit_multiplicative)$RMSE, accuracy(fit_damped)$RMSE)
)

```

```

# Printing the RMSE comparison
print(accuracy_data)

```

```

## # A tibble: 2 x 2
##   Model      RMSE
##   <chr>      <dbl>
## 1 Multiplicative 25.9
## 2 Damped      26.1

```

- The multiplicative model has a slightly lower RMSE, indicating better short-term accuracy. The difference is minimal, but the multiplicative model is preferred for short-term forecasts, while the damped model could be better for long-term stability.

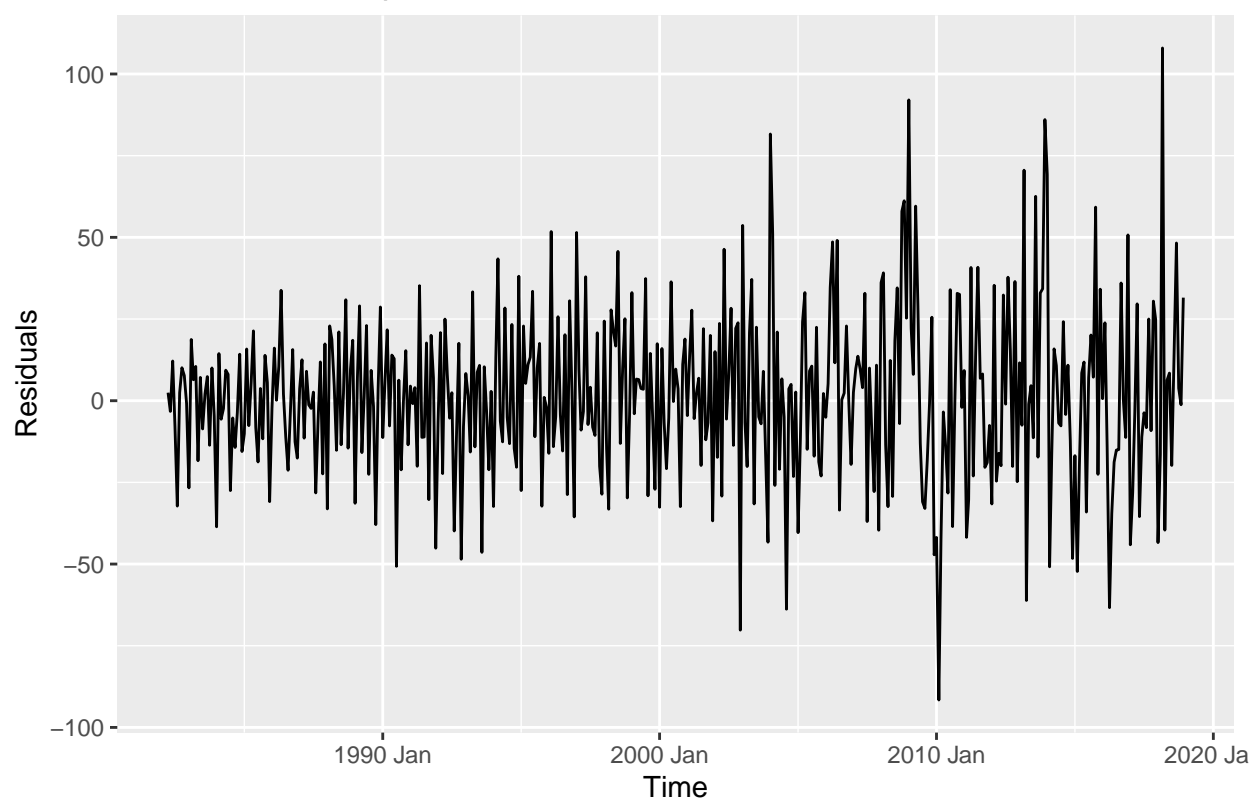
8.8.d. Residuals Analysis: Checking for White Noise

```

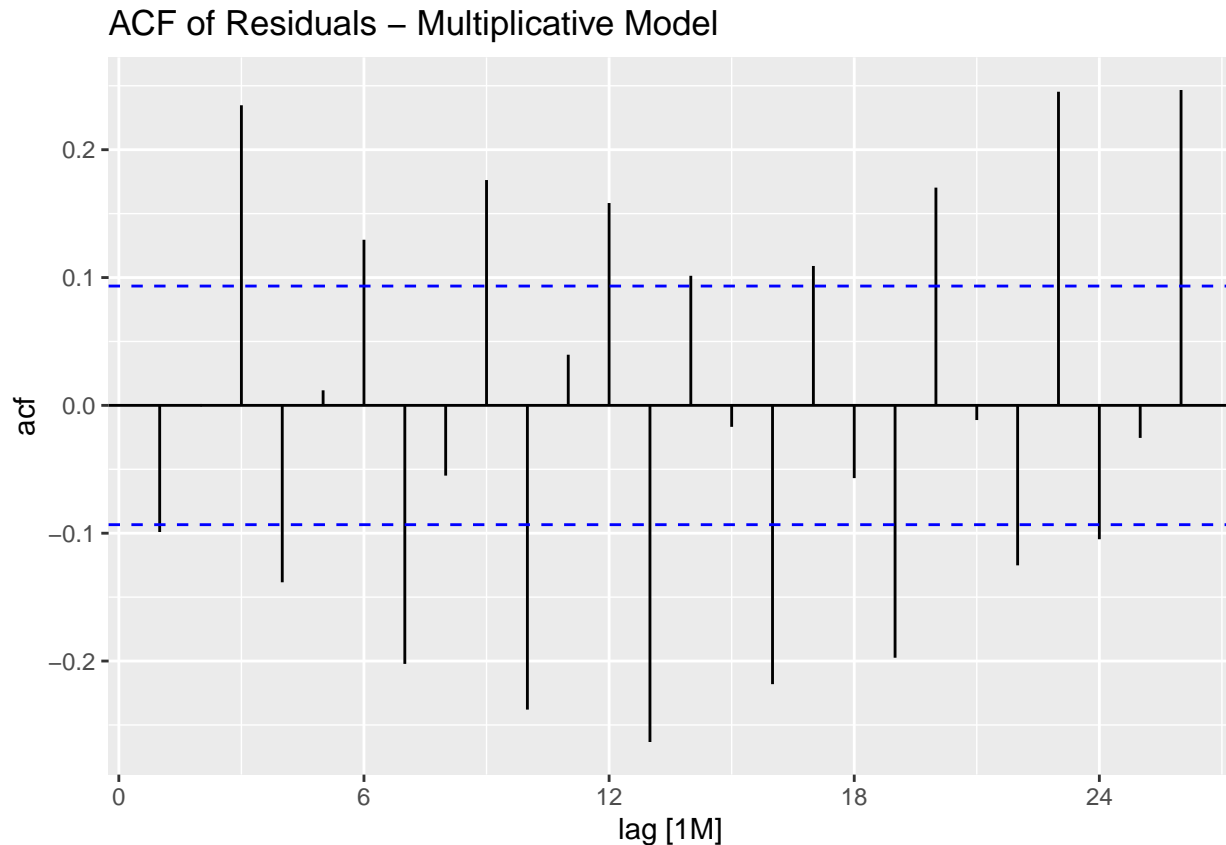
# Extracting residuals and plot residuals over time
augment(fit_multiplicative) %>%
  autoplot(.resid) +
  ggtitle("Residuals - Multiplicative Model") +
  xlab("Time") + ylab("Residuals")

```

Residuals – Multiplicative Model



```
# Plotting the ACF of the residuals  
augment(fit_multiplicative) %>%  
  ACF(.resid) %>%  
  autoplot() +  
  ggtitle("ACF of Residuals - Multiplicative Model")
```



- The residuals don't completely resemble white noise, as there are significant autocorrelations present, indicating that the model might not capture all the patterns in the data.

8.8.e. Test Set RMSE: Can We Beat the Seasonal Naive Approach?

```
# Splitting the data into training (up to 2010) and test sets
train_data <- retail_data %>% filter(Month <= yearmonth("2010 Dec"))
test_data <- retail_data %>% filter(Month > yearmonth("2010 Dec"))

# Training the models on the training data
fit_models <- train_data %>%
  model(
    Multiplicative = ETS(Turnover ~ error("A") + trend("A") + season("M")),
    SeasonalNaive = SNAIVE(Turnover)
  )

# Generating forecasts for both models for the test period (from 2011 onward)
forecasts <- fit_models %>% forecast(new_data = test_data)

# Calculating RMSE for both models
accuracy_test <- accuracy(forecasts, test_data)

# Extracting RMSE values
rmse_values <- accuracy_test %>%
  select(.model, RMSE)

# Printing RMSE comparison
```

```
print(rmse_values)

## # A tibble: 2 x 2
##   .model      RMSE
##   <chr>      <dbl>
## 1 Multiplicative 60.2
## 2 SeasonalNaive 480.

# Comparing RMSEs and determining which model is better
best_model <- ifelse(
  rmse_values$RMSE[rmse_values$.model == "Multiplicative"] < rmse_values$RMSE[rmse_values$.model == "SeasonalNaive"],
  "Multiplicative model beats the Seasonal Naive approach.",
  "Seasonal Naive approach is better."
)

# Printing the comparison result
print(best_model)
```

```
## [1] "Multiplicative model beats the Seasonal Naive approach."
```

- The multiplicative model outperforms the seasonal naive approach, with a lower RMSE on the test set, making it a better choice for this dataset.

Exercise 8.8.9: STL Decomposition with ETS for Retail Data

```
# Applying Box-Cox transformation and STL decomposition followed by ETS on seasonally adjusted data
stl_ets_model <- retail_data %>%
  filter(Month <= yearmonth("2010 Dec")) %>%
  model(
    STL_ETS_Model = decomposition_model(
      STL(Turnover ~ season(window = "periodic")),
      ETS(season_adjust ~ error("A") + trend("A") + season("N"))
    )
  )

# Creating a test dataset (from 2011 onwards)
test_dataset <- retail_data %>% filter(Month > yearmonth("2010 Dec"))

# Forecasting on the test data with the STL-ETS model
stl_ets_forecast <- stl_ets_model %>% forecast(new_data = test_dataset)

# Computing the RMSE for the STL-ETS model on the test data
stl_ets_accuracy <- accuracy(stl_ets_forecast, test_dataset)
stl_ets_rmse <- stl_ets_accuracy$RMSE

# Printing RMSE for STL-ETS model
print(paste("Test RMSE for STL-ETS model:", stl_ets_rmse))

## [1] "Test RMSE for STL-ETS model: 112.860739361934"

# Training the Multiplicative model on the training data
multiplicative_model <- retail_data %>%
  filter(Month <= yearmonth("2010 Dec")) %>%
  model(Multiplicative = ETS(Turnover ~ error("A") + trend("A") + season("M")))
```

```

# Forecasting on the test data with the Multiplicative model
multiplicative_forecast <- multiplicative_model %>%
  forecast(new_data = test_dataset)

# Computing the RMSE for the Multiplicative model on the test data
multiplicative_accuracy <- accuracy(multiplicative_forecast, test_dataset)
multiplicative_model_rmse <- multiplicative_accuracy$RMSE

# Printing RMSE for Multiplicative model
print(paste("Test RMSE for Multiplicative model:", multiplicative_model_rmse))

## [1] "Test RMSE for Multiplicative model: 58.0019609754441"

# Comparing the RMSEs and determining which model performs better
best_model_comparison <- ifelse(
  stl_ets_rmse < multiplicative_model_rmse,
  "STL-ETS model beats the Multiplicative Model.",
  "Multiplicative Model is better."
)

# Printting the comparison result
print(best_model_comparison)

## [1] "Multiplicative Model is better."

```

- STL decomposition followed by ETS modeling provides another way to model seasonality. If the RMSE for the STL-ETS model is lower, it offers an improved forecast compared to the multiplicative model. Otherwise, the multiplicative model remains the better choice.