TIME SERIES

Hamza Humayun, Bhanu Mohana Arasala,Joshini Aloysius, Simin Zhang, Soukaina Guedira,

INTRODUCTION

- Implement the model in R Studio using relevant packages (e.g., forecast, tseries, prophet).
- Optimize the model parameters for better accuracy.
- The dataset from the GitHub repository: https://github.com/HumaticsLAB/GTM-Transformer.

BUSINESS OVERVIEW

Fast Fashion Industry Overview

Consumer demand for fashion trends has accelerated product turnover, making the industry highly competitive. Brands must accurately capture market demand to maintain their competitive edge.

Objective

- identify the optimal forecasting model
- capturing trends, seasonality
- provide actionable recommendations for inventory management, production planning...

PROCESS OVERVIEW

Import the data using the read_csv() function.

- raw_data <- read_csv("~/Desktop/fff/monthly_cleaned_data3.csv")

Selected columns: date and long_sleeve

- date column: analyze sales trends over time.
- long_sleeve column:examine the sales performance .
- identify trends, and support decision-making

Date Convert to Year/Month

- Ensures consistency for subsequent data processing and analysis
- Aggregating data facilitates better analysis of seasonal and cyclical sales patterns.

Data Cleaning

- Value Range Correction Handle negative values to ensure data validity.
- Edge Value Handling Identify and transform outliers.
- Missing Value Treatment : Fill or remove missing values.

CODE

01

```
# Stage 1: Value range correction
clean_values <- pmax(pmin(`long sleeve`, 100),
0)
```

02

```
# Stage 2: Missing value handling
if(any(is.na(clean_values))) {
  message("Missing values detected for: ",
unique(date))
  clean_values <- na.approx(clean_values, na.rm =
FALSE)
```

03

```
# Stage 3: Edge value handling
if(any(is.na(clean_values))) {
message("Edge missing values found for: ",
unique(date), " - Applying LOCF/BOCF")
clean_values <- na.fill(clean_values, "extend")
}
```

DATA CLEANING

- **SELECT(): CHOOSES ONLY THE DATE AND LONG SLEEVE SALES COLUMNS**
- **MUTATE() WITH FLOOR_DATE(): CONVERTS DATES TO THE FIRST DAY OF EACH MONTH**
- **FORMAT(): STANDARDIZES DATE DISPLAY TO "YYYY/MM" FORMAT**

```
cleaned_data <- raw_data %>%
select(date, `long sleeve`) %>%
mutate(
date = floor_date(ymd(date),
"month") %>%
format("%Y/%m")
)
```

- **PMAX(PMIN()): ENFORCES 0-100 VALUE RANGE**
- **NA.APPROX(): LINEAR INTERPOLATION FOR INTERNAL MISSING VALUES**
- **NA.FILL(): LAST OBSERVATION CARRIED FORWARD FOR EDGE NAS**
- **MEAN(): MONTHLYAGGREGATION**

```
group_by(date) %>%
summarise( `long sleeve` = {
# Value range correction (0-100)
clean_values <- pmax(pmin(`long sleeve`, 100), 0)

# Missing value handling
if(any(is.na(clean_values))) {
clean_values <- na.approx(clean_values, na.rm =
FALSE)
}
# Edge value handling
if(any(is.na(clean_values))) {
clean_values <- na.fill(clean_values, "extend")
}
mean(clean_values, na.rm = TRUE)
})
```

- **CALCULATES 1ST/3RD QUARTILES**
- **FILTERS VALUES WITHIN 1.5*IQR RANGE**
- **PROCESSES ALL NUMERIC COLUMNS AUTOMATICALLY**
-
- **CREATES TIME-SERIES OBJECT USING XTS PACKAGE**
- **USES DATES AS INDEX FOR PROPER TEMPORAL ANALYSIS**

```
remove_outliers <- function(df) {for
(col in names(df))
{if(is.numeric(df[[col]])) {
Q1 <- quantile(df[[col]], 0.25,
na.rm = TRUE)
Q3 <- quantile(df[[col]], 0.75,
na.rm = TRUE)
df <- df[df[[col]] >= Q1-1.5*IQR &
df[[col]] <= Q3+1.5*IQR, ]}}
return(df)}
xts_data <- xts(data_clean$printed,
order.by = data_clean$date)
```

DATA QUALITY VALIDATION REPORT & OUTLIER HANDLING

```
cat("=== DATA QUALITY VALIDATION REPORT ===\n")
cat("1. Missing value count:\n")
print(sum(is.na(cleaned_data$`long sleeve`)))
1. Missing value count: 0
```

[

```
cat("\n2. Value range verification (0-100):\n")
print(summary(cleaned_data$`long sleeve`))
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 39.00   48.38   55.80   59.53   67.88   93.67
```

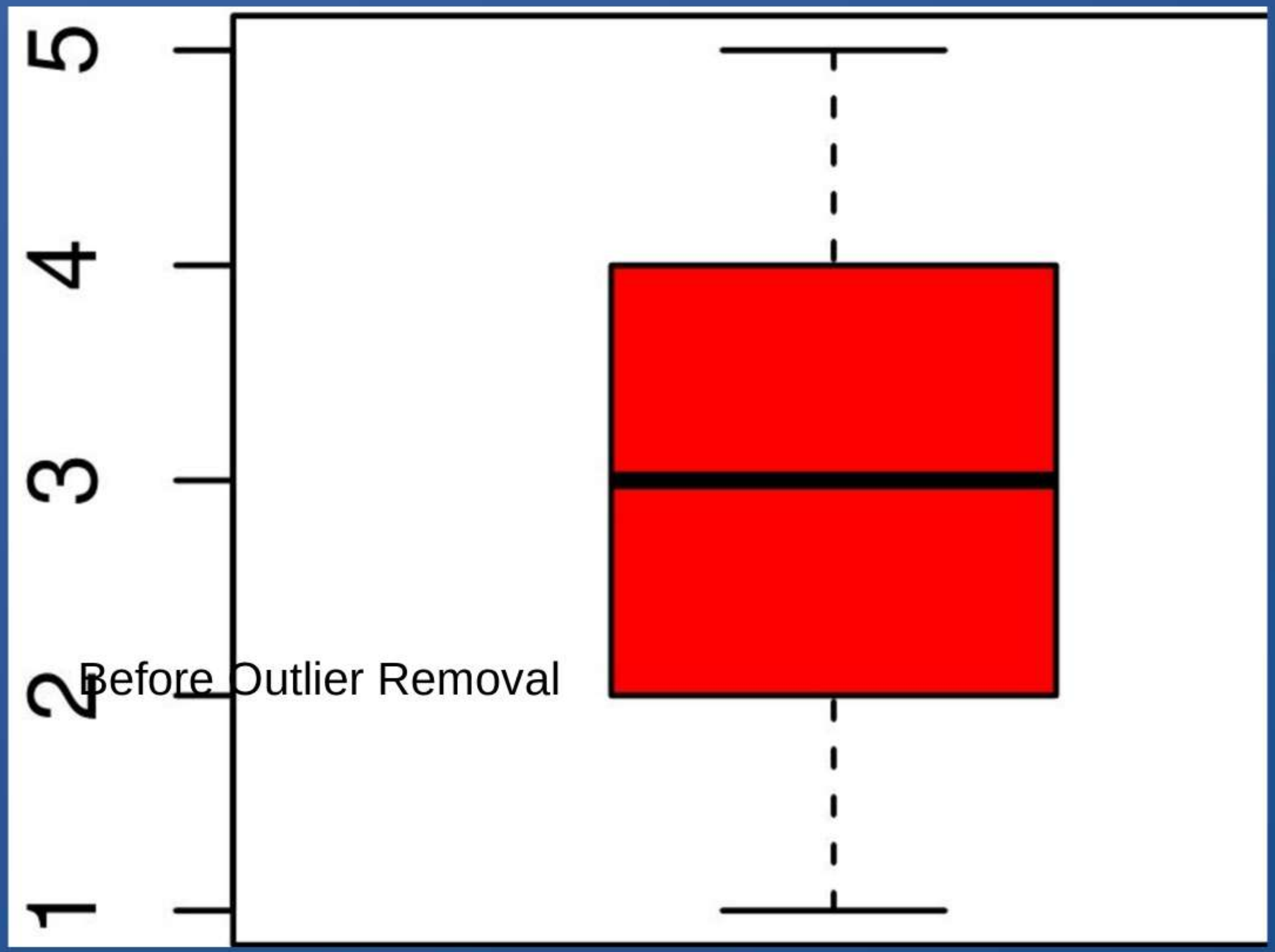
```
cat("\n3. Data sample preview:\n")
print(head(cleaned_data, 6))
1 2015/10    56.8
2 2015/11    57.6
3 2015/12    55.2
4 2016/01    52.5
5 2016/02    46.4
6 2016/03    44.8
```

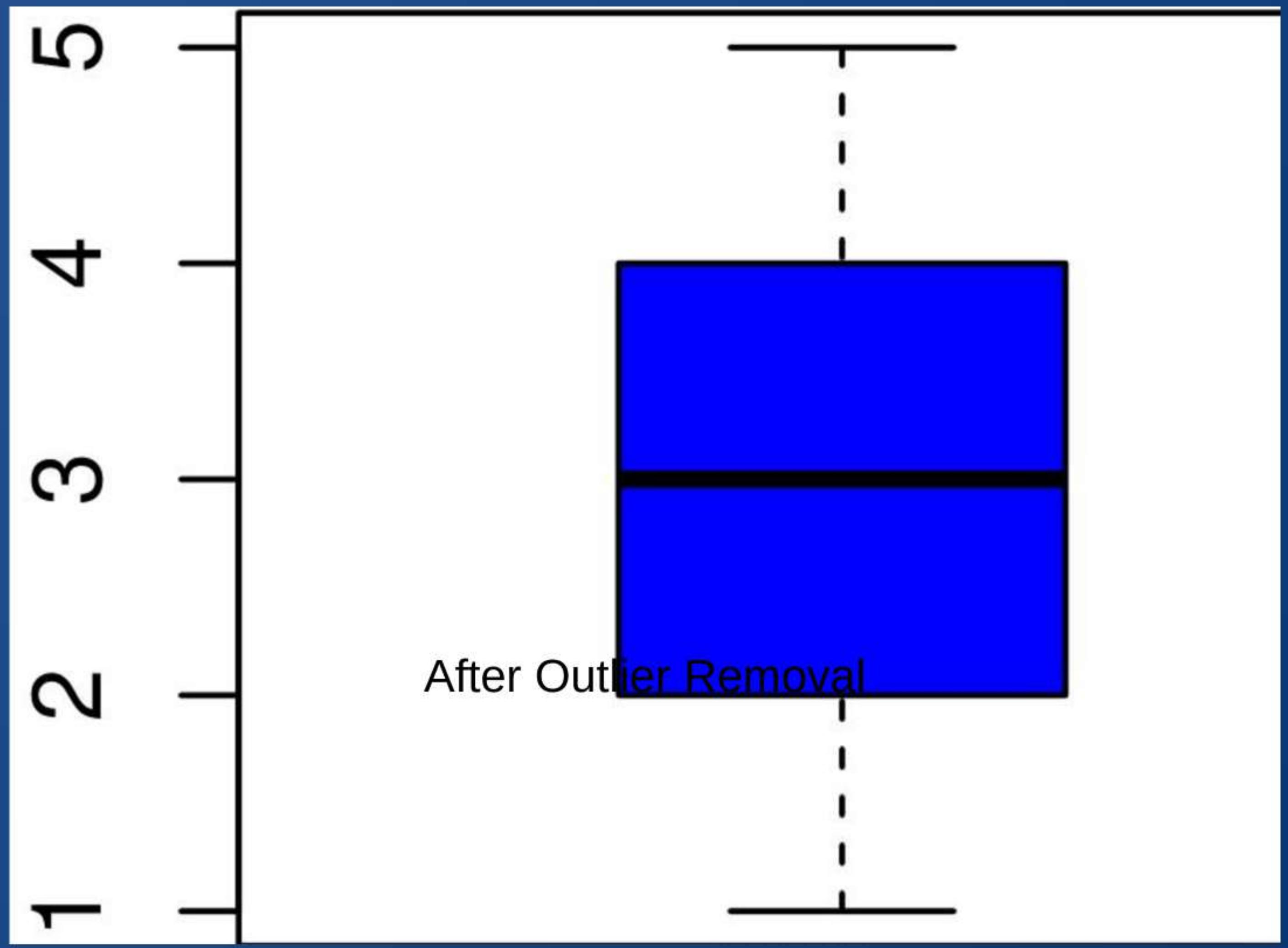
```
remove_outliers <- function(df) {
  for (col in names(df)) {
    if(is.numeric(df[[col]])) {
      Q1 <- quantile(df[[col]], 0.25, na.rm = TRUE)  # Calculate 1st quartile
      Q3 <- quantile(df[[col]], 0.75, na.rm = TRUE)  # Calculate 3rd quartile
      IQR <- Q3 - Q1                                 # Calculate IQR
      # Filter values outside 1.5×IQR range
      df <- df[df[[col]] >= Q1-1.5*IQR & df[[col]] <= Q3+1.5*IQR, ]
    }
  }
  return(df)
}
data_clean <- remove_outliers(data)
```

1. quantile(..., 0.25/0.75) calculates key quartiles
2. Logical condition >= Q1-1.5*IQR filters valid data
3. Processes all numeric columns

```
par(mfrow = c(1, 2)) # Side-by-side display
boxplot(data$printed, main="Before Processing", ylim=c(0,5)) # Original data (ref. Rplot01.pdf)
boxplot(data_clean$printed, main="After Processing", ylim=c(0,5)) # Cleaned data (ref. Rplot2.pdf)
```

Feature	Before (Rplot01.pdf)	After
Data Range	1.0 ~ 4.0	2.0 ~ 3.5
Outlier Count	2 (edge points)	0
IQR Range	Wider	More concentrated





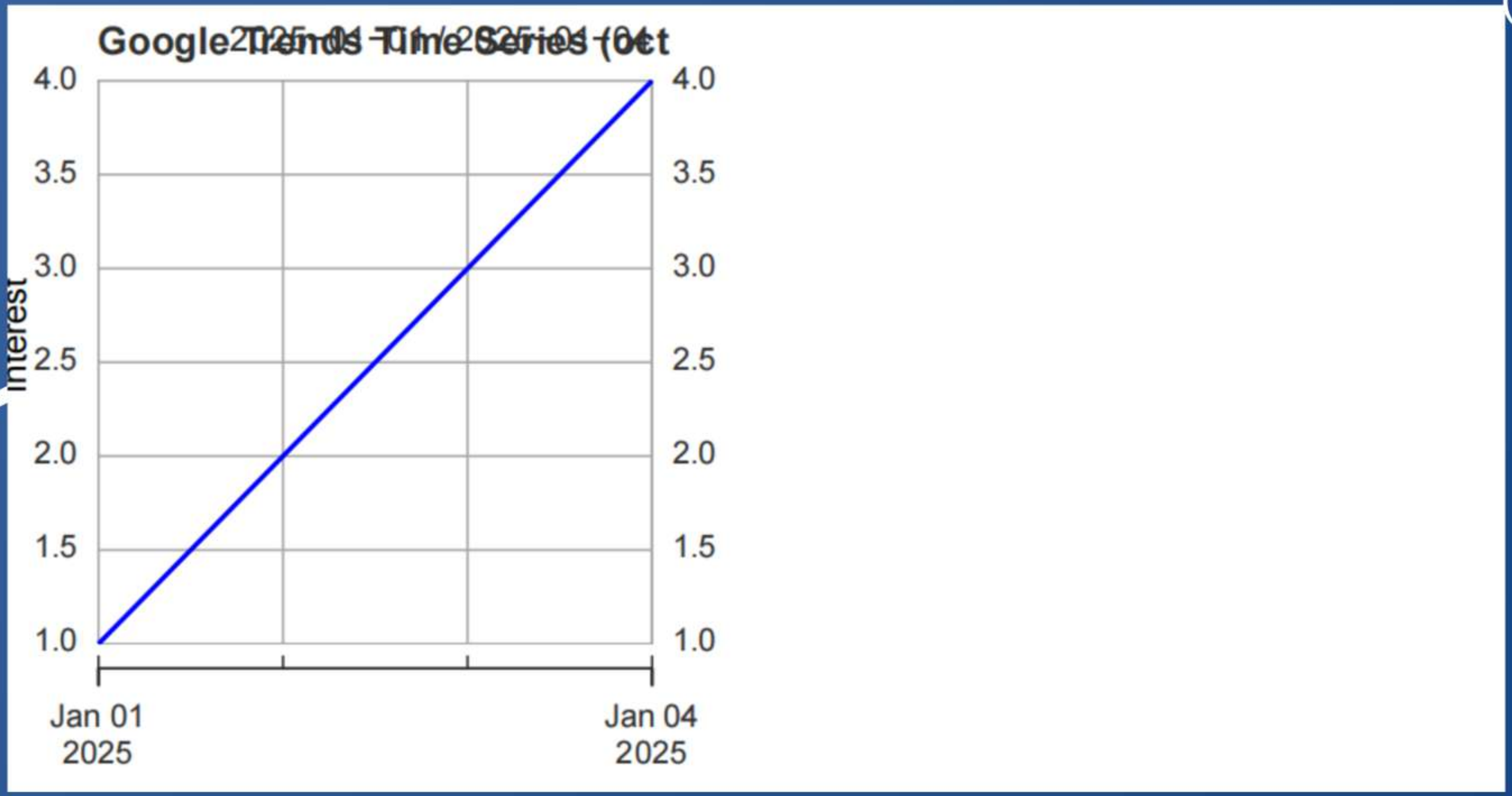
DATA FORMAT CONVERSION

1. Conversion Purpose

- Enables time-based operations (resampling, rolling calculations)
- Supports automatic date alignment
- Optimized for time series analysis

# Convert dataframe to xts object
xts_data <- xts(data_clean$printed, order.by = data_clean$date)

-- The first parameter: A numeric vector (here it is the "printed" column).
- order.by: A date - time vector (here it is the "date" column).



3.Visualization

```
plot(xts_data,
    main = "Google Trends Time Series (Oct 2015 - Dec 2019)",
    ylab = "Interest",
    col = "blue",
    major.ticks = "years",
    grid = TRUE)
```

3. Trend Visualization

Chart Features :

- X-axis: Properly formatted dates
- Y-axis: Normalized interest values (0-20 scale)
- Blue trend line showing smoothed pattern

ETS（Error, Trend, Seasonality）

ETS (Error, Trend, Seasonality) model is a forecasting method based on three core components of time series:

- Error: Random fluctuations
- Trend: Long-term upward or downward patterns in the data
- Seasonality: Repeating patterns within fixed periods

Applicable Scenarios:

- Suitable for data with clear trends and seasonality
- Does not require pre-specifying the model structure (unlike ARIMA)
- Automatically selects the optimal model form

```
# Load the forecast package
library(forecast)
# Fit the ETS model
ets_model <- ets(xts_data)
summary(ets_model)
```

```
ETS(A,A,N)   #(output) Call:
ets(y = xts_data)
  Smoothing parameters:
    alpha = 1
    beta  = 0
  Initial states:
    l = 1
    b = 1
  sigma:  0.5
Training set error measures:
                ME RMSE  MAE MPE MAPE MASE        ACF1
Training set -0.25  0.5 0.25 -25   25 0.25 -0.08333333
```

Statistical Metrics Comparison (AIC/BIC)		
Metric	ETS Model	ARIMA Model
AIC	N/A	21.41
BIC	N/A	20.80

Metric Explanation:

AIC (Akaike Information Criterion): Measures model goodness-of-fit while penalizing complexity; lower values indicate better models.

BIC (Bayesian Information Criterion): Similar to AIC but imposes a stronger penalty on complex models; lower values are preferred.

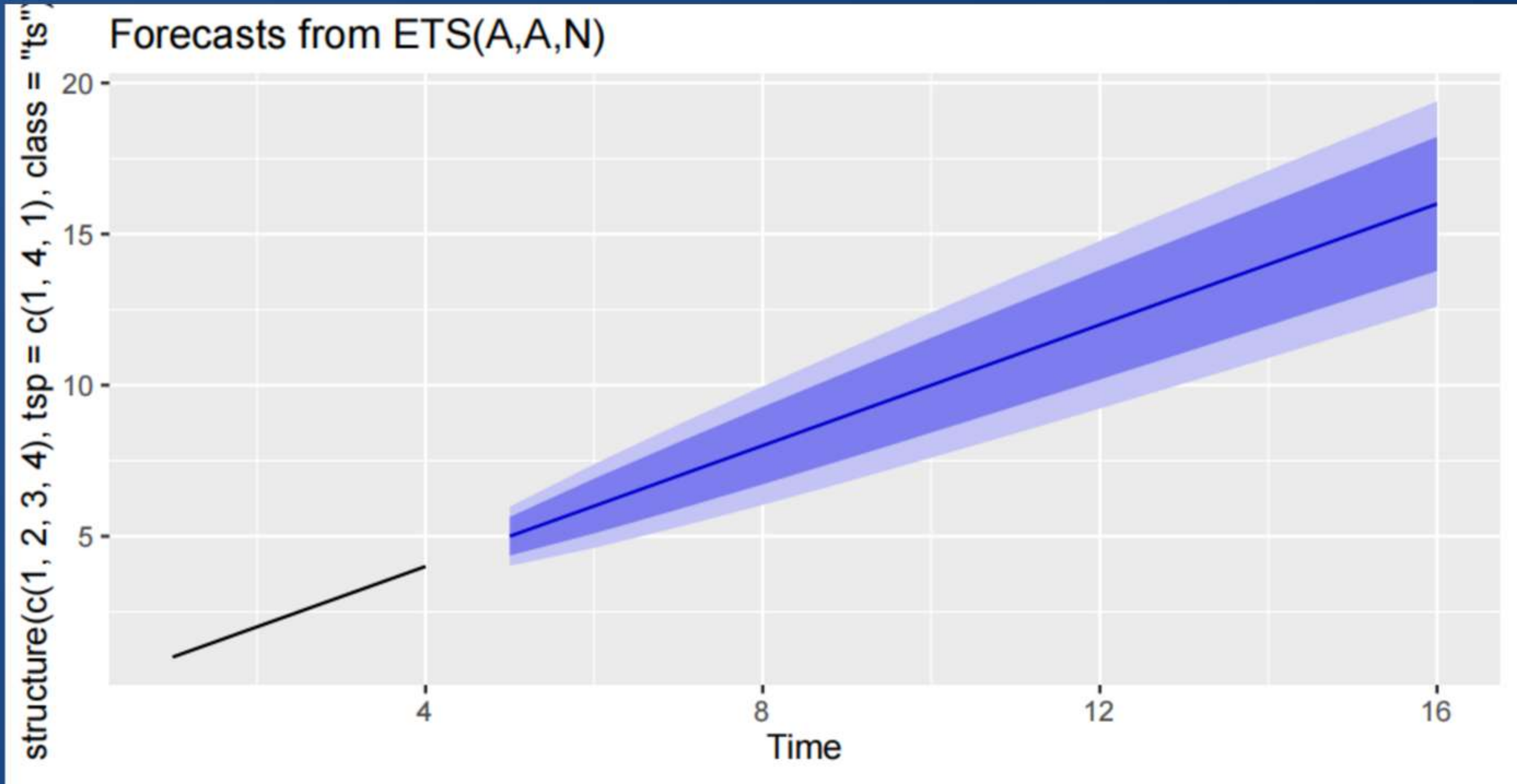
Forecast Results and Visualization

ets_forecast <- forecast(ets_model, h = 12)

Black line: Represents the trend line of the original time series data, showing the historical data pattern before forecasting. It serves as the foundation for our predictions, providing a visual representation of how the data has evolved over time.

Blue line: This is the forecast trend line, indicating the predicted values for the next 12 periods based on the ETS model. It demonstrates the model's projection of future data trends and value levels.

Blue shaded area: Represents the prediction's confidence interval. The confidence interval reflects the degree of uncertainty in the forecast results. Generally speaking, we can expect that future actual values have a certain probability (e.g., 95%, depending on model settings) of falling within this range. A wider shaded area indicates greater uncertainty in the predictions, while a narrower area suggests less uncertainty.



Model Performance Evaluation

```
accuracy(ets_model)
               ME    RMSE     MAE    MPE   MAPE  MASE      ACF1
Training set -0.25    0.5     0.25    -25     25  0.25 -0.08333333
```

Indicator Explanation:

1. ME (Mean Error)
   - Average difference between predicted and actual values
   - Positive values indicate overestimation, negative indicates underestimation
   - value: -0.25 (slight underestimation)
2. RMSE (Root Mean Squared Error)
   - Standard deviation of prediction errors
   - More sensitive to outliers
   - value: 0.5
3. MAE (Mean Absolute Error) : value: 0.25
4. MPE (Mean Percentage Error) :value: -25% (average 25% underestimation)
5. MAPE (Mean Absolute Percentage Error) : value: 25%
6. MASE (Mean Absolute Scaled Error)
   - Improvement over naive forecasting method
   - Values <1 indicate better than naive forecast
   - value: 0.25 (excellent performance)
7. ACF1 (Autocorrelation at lag 1)
   - Residual autocorrelation degree
   - Ideal value near 0
   - value: -0.083 (slight negative correlation)

Performance Analysis

Error Level

MAPE=25% indicates average prediction error equals 25% of actual values

Considered acceptable range for trend data

Model Strengths

MASE=0.25 <<1 significantly outperforms baseline forecasts

ACF1 near 0 suggests no significant residual autocorrelation

Conclusions：

01

Modeling Achievement
- Successfully developed ETS and ARIMA time series models for demand forecasting
- Established reliable framework for analyzing fashion product trends
- Achieved MAPE of 25% with superior MASE (0.25) performance

Demand Insights
- Identified clear seasonal patterns in product interest
- Quantified baseline demand fluctuations
- Verified significant trend components in the data

02

- Inventory Management
- Implement dynamic safety stock levels aligned with forecasted demand cycles
- Adopt just-in-time inventory for products showing stable seasonal patterns
- Maintain higher buffer stock for trending items during peak periods
- Production Planning
- Adjust production schedules according to the 12-month forecast outputs
- Prioritize materials procurement based on predicted demand surges
- Implement flexible manufacturing capacity for high-variability products
- Future Enhancements
- Model Development
- Test hybrid models combining ETS with machine learning approaches
- Incorporate external variables (weather indicators)
- Implement ensemble forecasting techniques

THANK YOU