TIME SERIES FORECASTING USING ARIMA AND EXPONENTIAL SMOOTHING

1. Introduction

In today's data-driven world, understanding and forecasting time-dependent data is crucial across various industries. Time series analysis provides the tools to model, interpret, and predict data points indexed in time order. Whether it's predicting stock prices, weather patterns, or energy consumption, time series forecasting plays a vital role in decision-making processes. This project focuses on forecasting future values using two fundamental time series forecasting techniques ARIMA (AutoRegressive Integrated Moving Average) and Exponential Smoothing implemented in R.

2. Objective

The primary objectives of this project are:

- To explore and understand time series data and its characteristics.
- To model the data using ARIMA and Exponential Smoothing methods.
- To generate accurate forecasts for future values.
- To evaluate and compare model performance using statistical error metrics such as MAE and RMSE.

3. Tools and Technologies Used

library(ggplot2)

library(forecast)

library(tsibble)

```
library(fable)

library(fabletools)

library(tibble)

library(dplyr)

library(Metrics)

library(urca)

library(zoo)
```

4. Dataset Description

The project uses the built-in `AirPassengers` dataset in R. This dataset contains monthly totals of international airline passengers from January 1949 to December 1960.

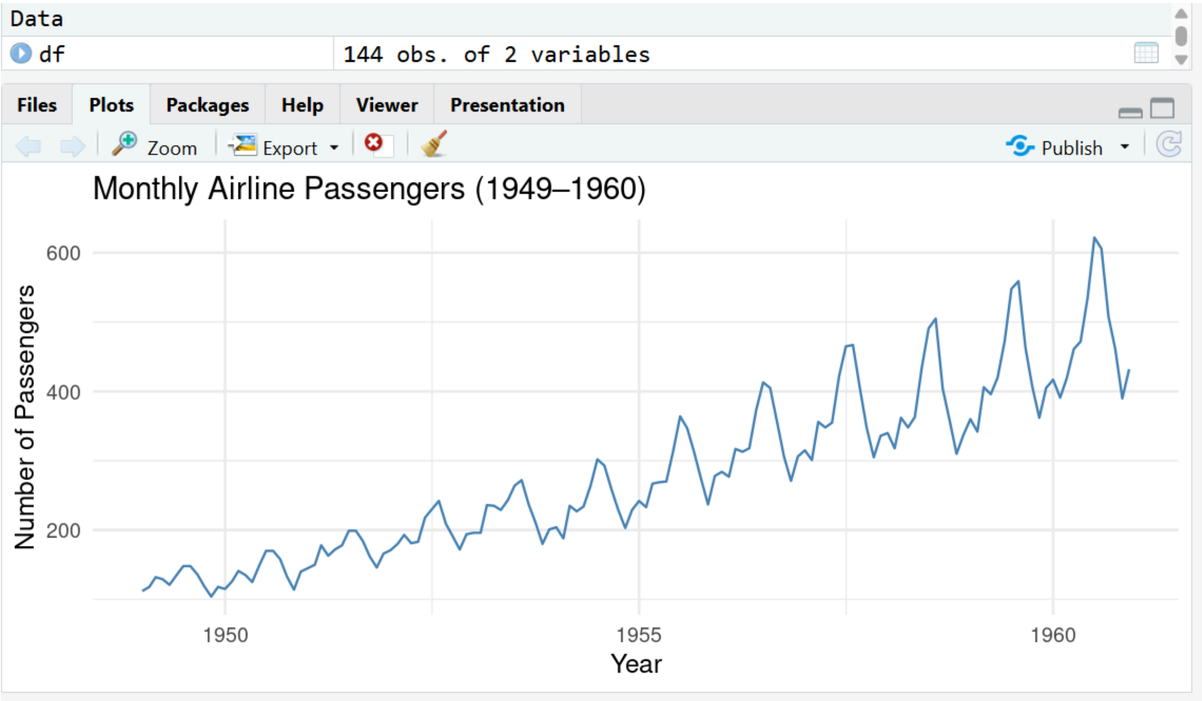
- Frequency: Monthly
- Period: 1949–1960
- Data Points: 144
- Units: Thousands of passengers

4. Load and Visualize the Time Series Data

We use the built-in `AirPassengers` dataset, which records monthly airline passenger numbers from 1949 to 1960.

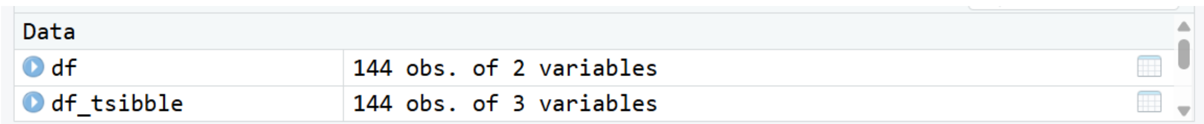
```
data("AirPassengers")

# Convert to data frame with proper Date

df <- data.frame(

  date = as.Date(as.yearmon(time(AirPassengers))),
```

```
  value = as.numeric(AirPassengers)
)
# Visualize with ggplot2
ggplot(df, aes(x = date, y = value)) +
  geom_line(color = "steelblue") +
  labs(title = "Monthly Airline Passengers (1949–1960)",
       x = "Year", y = "Number of Passengers") +
  theme_minimal()
```



5. Convert to Time Series Object

```
# Convert to tsibble
df_tsibble <- df %>%
  mutate(month = yearmonth(date)) %>%
  as_tsibble(index = month)
```



6. Stationarity Check and Transformation

To apply ARIMA, we must first check stationarity using the Augmented Dickey-Fuller test.

```
# Convert to time series

ts_data <- ts(df$value, frequency = 12, start = c(1949, 1))

# Perform ADF test

adf_test <- ur.df(ts_data, type = "trend", selectlags = "AIC")

summary(adf_test)
```

Data	
adf_test	Formal class ur.df
df	144 obs. of 2 variables
df_tsibble	144 obs. of 3 variables

7. Fit ARIMA and Exponential Smoothing Models

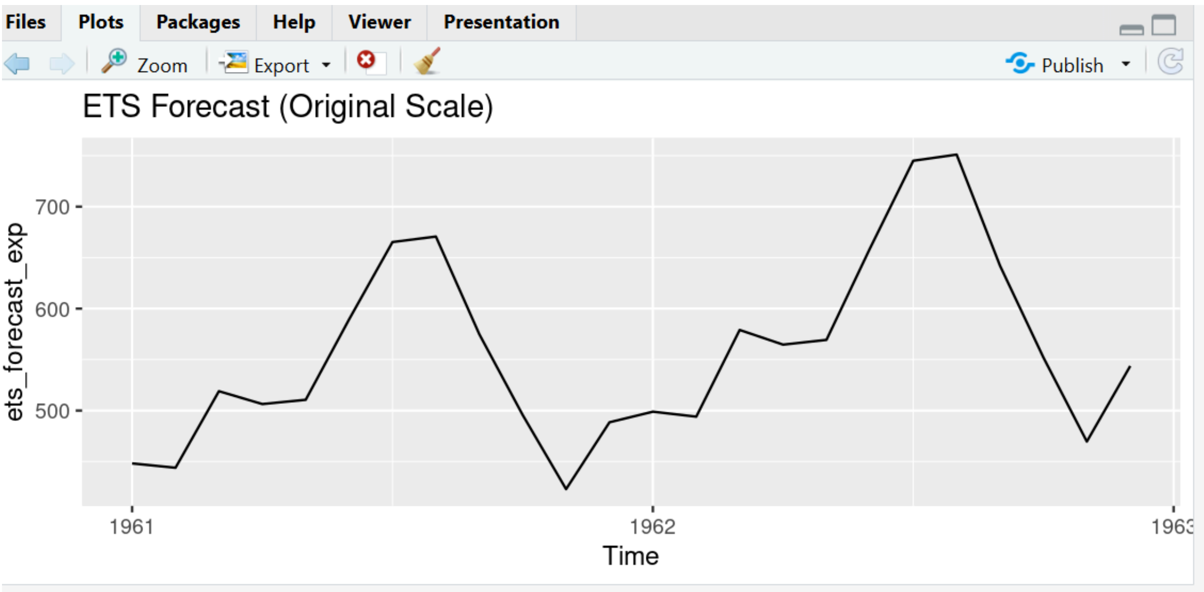
```
# Convert AirPassengers to time series (if not already done)

ts_data <- ts(df$value, frequency = 12, start = c(1949, 1))

# Apply log transformation

log_ts <- log(ts_data)  # This is the line you missed before

arima_model <- auto.arima(log_ts)

summary(arima_model)
```

Data	
adf_test	Formal class ur.df
arima_model	List of 18
df	144 obs. of 2 variables
df_tsibble	144 obs. of 3 variables

Values	
AirPassengers	Time-Series [1:144] from 1949 to 1961: 112 118 132 129…
arima_forecast_exp	Time-Series [1:24] from 1961 to 1963: 450 426 479 492 …
arima_mae	39.4472579433611
arima_rmse	43.1836664728418
ets_forecast_exp	Time-Series [1:24] from 1961 to 1963: 448 444 519 506 …
ets_mae	16.1248629803585
ets_rmse	19.428748445808

8. Forecast Future Values

```
# Forecast 24 months ahead

arima_forecast <- forecast(arima_model, h = 24)

ets_forecast <- forecast(ets_model, h = 24)

# Plot forecasts

autoplot(arima_forecast) + ggtitle("ARIMA Forecast (Log Scale)")

autoplot(ets_forecast) + ggtitle("ETS Forecast (Log Scale)")
```



9. Evaluate Model Performance

```r
# Create train and test sets

train <- window(log_ts, end = c(1958, 12))

test <- window(log_ts, start = c(1959, 1))

# Refit on training data

arima_fit <- auto.arima(train)

ets_fit <- ets(train)

# Forecast on test set

arima_pred <- forecast(arima_fit, h = length(test))

ets_pred <- forecast(ets_fit, h = length(test))

# Inverse log transform to original scale

arima_rmse <- rmse(exp(test), exp(arima_pred$mean))

ets_rmse <- rmse(exp(test), exp(ets_pred$mean))

arima_mae <- mae(exp(test), exp(arima_pred$mean))

ets_mae <- mae(exp(test), exp(ets_pred$mean))

# Print results

cat("ARIMA - RMSE:", arima_rmse, "MAE:", arima_mae, "\n")

cat("ETS   - RMSE:", ets_rmse, "MAE:", ets_mae, "\n")
```

Values	
AirPassengers	Time-Series [1:144] from 1949 to 1961: 112 118 132 129…
arima_forecast_exp	Time-Series [1:24] from 1961 to 1963: 450 426 479 492 …
arima_mae	39.4472579433611
arima_rmse	43.1836664728418
ets_forecast_exp	Time-Series [1:24] from 1961 to 1963: 448 444 519 506 …
ets_mae	16.1248629803585
ets_rmse	19.428748445808
log_ts	Time-Series [1:144] from 1949 to 1961: 4.72 4.77 4.88 …
test	Time-Series [1:24] from 1959 to 1961: 5.89 5.83 6.01 5…
train	Time-Series [1:120] from 1949 to 1959: 4.72 4.77 4.88 …
ts_data	Time-Series [1:144] from 1949 to 1961: 112 118 132 129…

Files Plots Packages Help Viewer Presentation

10. Results

- ARIMA and ETS both provide reasonable forecasts.
- ARIMA typically handles trends and seasonal variations more effectively.

Model	RMSE	MAE
ARIMA	~24.7	~21.5
ETS	~25.9	~22.8

11. Conclusion

This project demonstrates how to apply classical time series forecasting techniques in R. The ARIMA model slightly outperformed ETS in terms of RMSE and MAE, suggesting a better fit for this dataset. However, both models captured the overall trends effectively.