Final Report- Time Series Analysis in R

Introduction

This report presents time series forecasting using the AirPassengers dataset (1949-1960), covering two methods:

1. ARIMA model with base R and the forecast package.
2. Prophet model developed by Facebook.

Both methods are used to analyze and predict trends and seasonality in monthly air passenger data.

Guide 1 Time Series using ARIMA

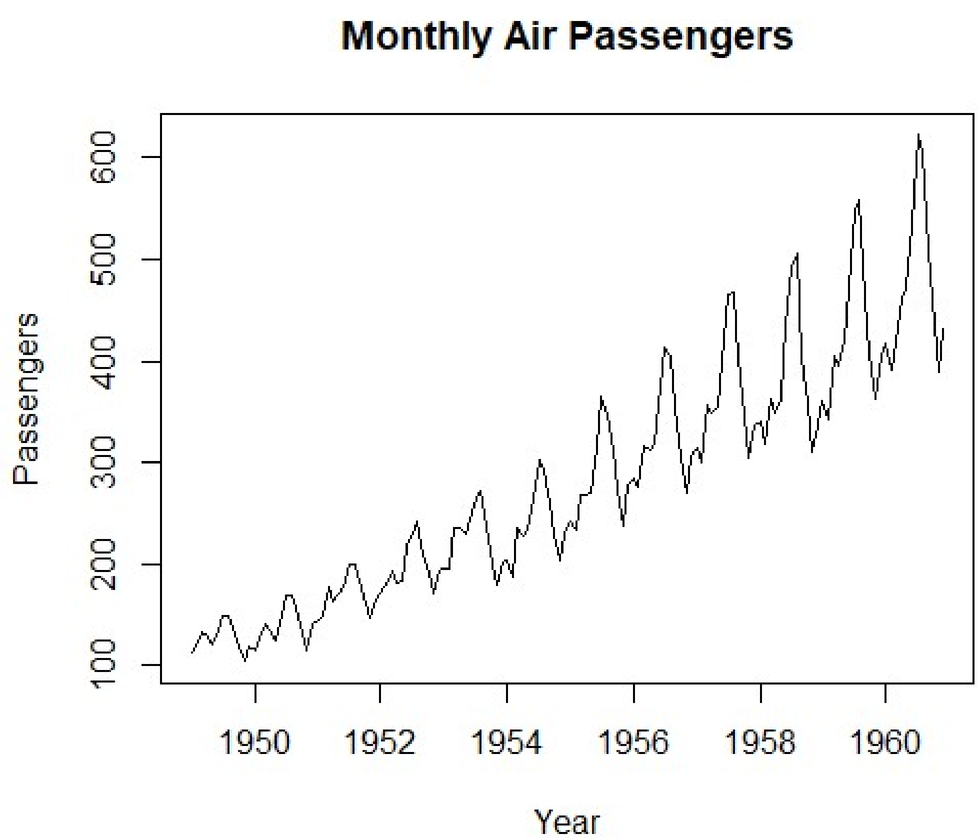
Steps followed:
- Converted the monthly data into a ts object with frequency 12.
- Plotted the time series to visualize trends.
- Decomposed the series into trend, seasonality, and noise.
- Used auto.arima() to find the best ARIMA model.
- Forecasted 24 months into the future.

Code

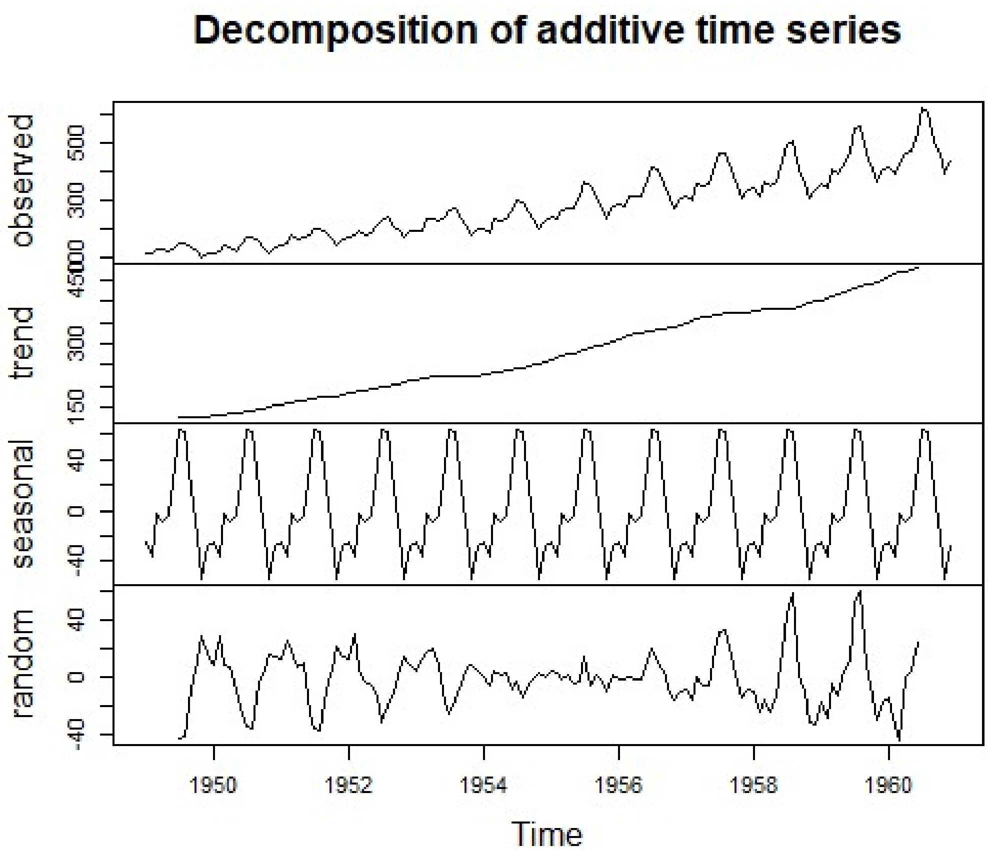
```
# Load required libraries
library(lubridate)
library(forecast)

# Read data
df <- read.csv("example_air_passengers.csv")
df$ds <- as.Date(df$ds)

# Create time series object
passenger_ts <- ts(df$y, start = c(1949, 1), frequency = 12)
```

```
# Plot time series
plot(passenger_ts, main = "Monthly Air Passengers", ylab = "Passengers", xlab = "Year")

# Decompose time series
decomposed <- decompose(passenger_ts)
plot(decomposed)

# Fit ARIMA model
fit <- auto.arima(passenger_ts)
summary(fit)

# Forecast next 24 months
forecast_values <- forecast(fit, h = 24)
plot(forecast_values)
```

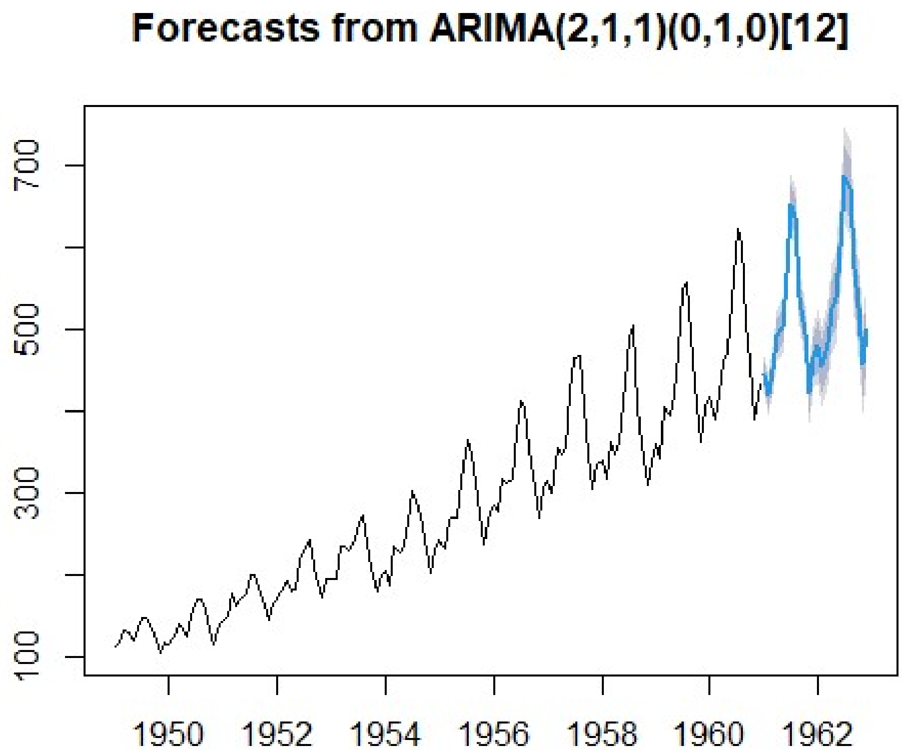
Outputs and Plots



Time Series Plot (Guide 1 - ARIMA)



Decomposition Plot (Guide 1 - ARIMA



ARIMA Forecast Plot (Guide 1)

<u>Guide 2: Time Series Forecasting using Facebook Prophet</u>

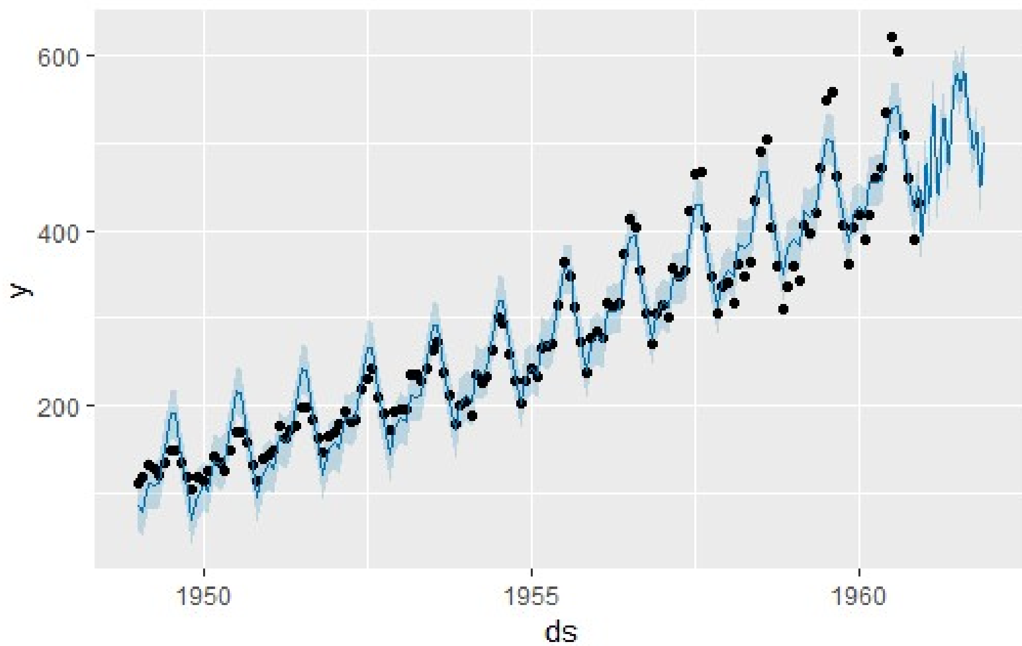
Steps followed:
- Prepared data with 'ds' (date) and 'y' (value) columns.
- Fitted Prophet model.
- Forecasted one year into the future.
- Visualized forecast and decomposed trend/seasonality.

Prophet handles holiday effects and multiple seasonality automatically.

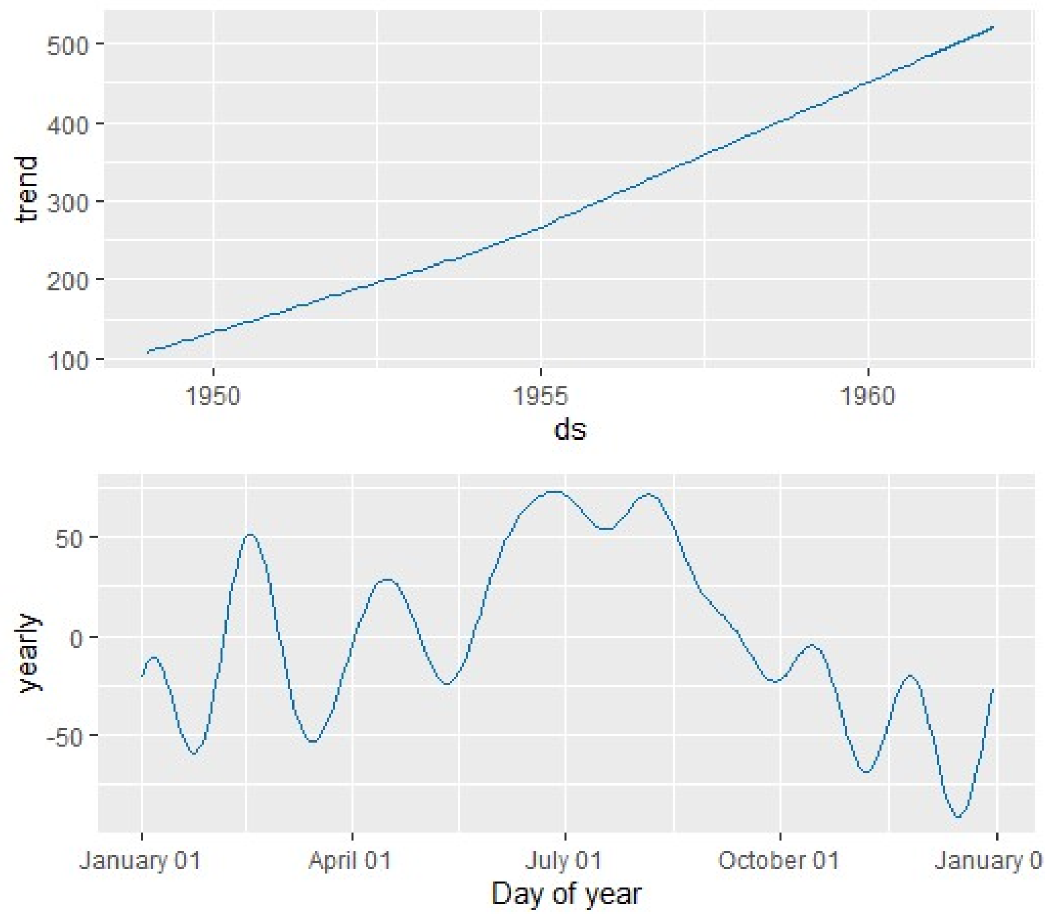
Code

```r
# Load library
library(prophet)

# Read data
df <- read.csv("example_air_passengers.csv")
df$ds <- as.Date(df$ds)

# Fit model
m <- prophet(df)

# Future dataframe: forecast 365 days
future <- make_future_dataframe(m, periods = 365)
forecast <- predict(m, future)

# Plot forecast
plot(m, forecast)

# Plot components
prophet_plot_components(m, forecast)

# View forecast
tail(forecast[c("ds", "yhat", "yhat_lower", "yhat_upper")])
```

Output & Plots



Forecast Plot (Guide 2 - Prophet)



Trend and Yearly Seasonality (Guide 2 - Prophet)

Conclusion

ARIMA models are strong for shorter-term forecasts and allow parameter tuning. Prophet excels in ease-of-use and interpretability, especially for longer time series with clear seasonality.

Both methods showed strong upward trends in air passenger traffic with seasonal spikes between June and August.

Recommendation: Use ARIMA for fine-tuned statistical forecasting; use Prophet for automated business-friendly forecasting with visualization.