MTH6139 Time Series

Coursework 1 – Air Passengers

Nabeel Asghar

Spring term 2025

Section 1.1: Understanding the Air Passenger Dataset ✈️

The Air Passengers dataset contains the monthly airline passenger counts (in thousands) in between the years of 1949 through to 1960. It is a useful dataset, often used to demonstrate time series forecasting

```
# Load necessary libraries
library(prophet)
```

```
## Loading required package: Rcpp
```

```
## Loading required package: rlang
```

```
library(zoo)
```

```
## Warning: package 'zoo' was built under R version 4.3.3
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
# Load and inspect the AirPassengers dataset
data("AirPassengers")
str(AirPassengers)  # Structure of the dataset
```

```
##  Time-Series [1:144] from 1949 to 1961: 112 118 132 129 121 135 148 148 136 1
19 ...
```

```
summary(AirPassengers)  # Summary statistics
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   104.0   180.0   265.5   280.3   360.5   622.0
```

Section 1.2: Preparing the Data for Prophet

Prophet requires a dataframe with: • A date column named ds (datetime format) • A values column named y (numeric format)

We will convert the AirPassengers dataset into this required format.

```
# Convert AirPassengers dataset into a dataframe suitable for Prophet
air_passengers_df <- data.frame(
  ds = as.Date(as.yearmon(time(AirPassengers))),  # Convert to date format
  y = as.numeric(AirPassengers)  # Convert passenger count to numeric
)

# Display first few rows of the dataset
head(air_passengers_df)
```

```
##           ds   y
## 1 1949-01-01 112
## 2 1949-02-01 118
## 3 1949-03-01 132
## 4 1949-04-01 129
## 5 1949-05-01 121
## 6 1949-06-01 135
```

Now that we have cleaned and formatted the data we can prepare for take off and fit the prophet model.

Section 1.3: Building a Prophet Model

Prophet handles seasonality, trend, and missing values very well. We will train the model and make a prediction for the next 24 months.

```
# Fit the Prophet model
model <- prophet(air_passengers_df)
```

```
## Disabling weekly seasonality. Run prophet with weekly.seasonality=TRUE to ove
rride this.
```

```
## Disabling daily seasonality. Run prophet with daily.seasonality=TRUE to overr
ide this.
```

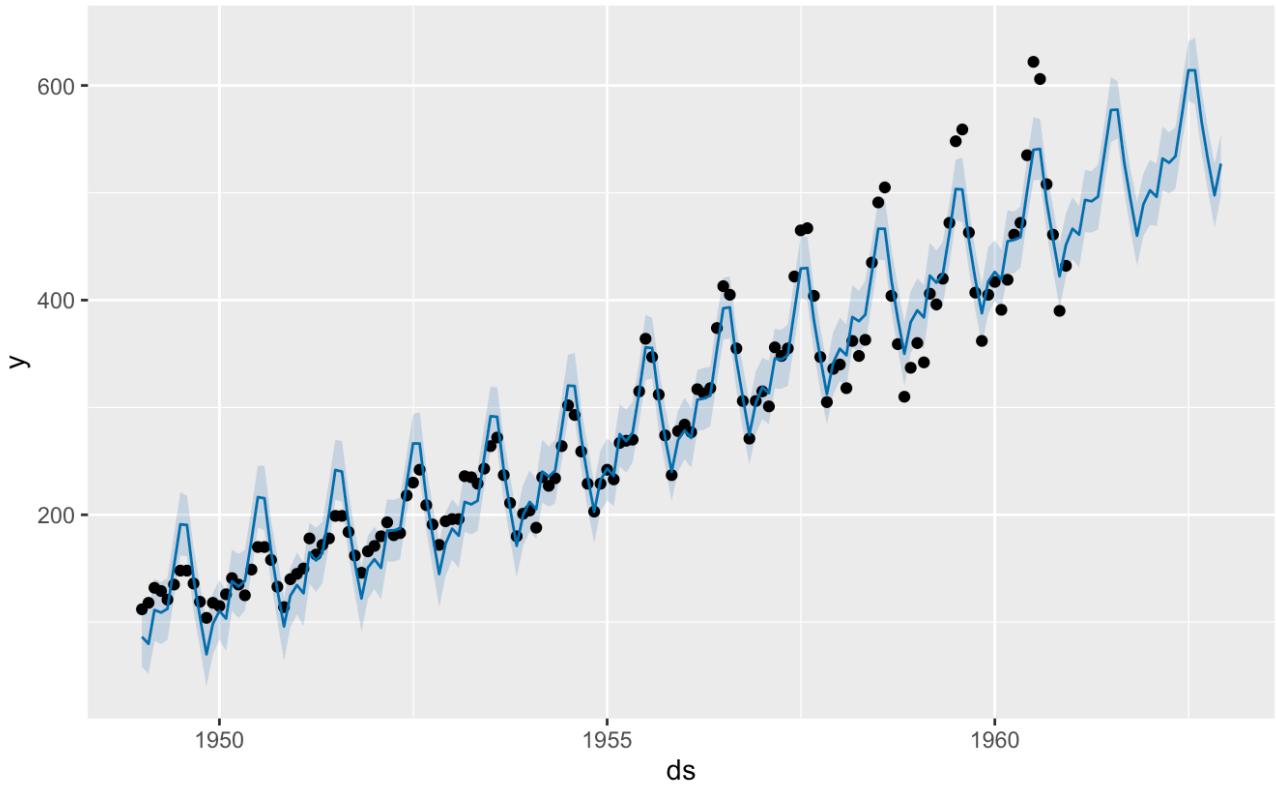
```
# Create future dates for forecasting (24 months ahead)
future_dates <- make_future_dataframe(model, periods = 24, freq = "month")

# Generate predictions
forecast <- predict(model, future_dates)

# Display the first few forecasted values
head(forecast)
```

```
##           ds    trend additive_terms additive_terms_lower additive_terms_uppe
r
## 1 1949-01-01 107.6954    -21.3502000          -21.3502000          -21.350200
0
## 2 1949-02-01 109.8329    -29.9939894          -29.9939894          -29.993989
4
## 3 1949-03-01 111.7635     -0.4636148           -0.4636148           -0.463614
8
## 4 1949-04-01 113.9010     -4.8997554           -4.8997554           -4.899755
4
## 5 1949-05-01 115.9695     -3.6071662           -3.6071662           -3.607166
2
## 6 1949-06-01 118.1070     34.4044256           34.4044256           34.404425
6
##        yearly yearly_lower yearly_upper multiplicative_terms
## 1 -21.3502000  -21.3502000  -21.3502000                    0
## 2 -29.9939894  -29.9939894  -29.9939894                    0
## 3  -0.4636148   -0.4636148   -0.4636148                    0
## 4  -4.8997554   -4.8997554   -4.8997554                    0
## 5  -3.6071662   -3.6071662   -3.6071662                    0
## 6  34.4044256   34.4044256   34.4044256                    0
##   multiplicative_terms_lower multiplicative_terms_upper yhat_lower yhat_upper
## 1                          0                          0   58.18991   115.1116
## 2                          0                          0   52.07052   108.3191
## 3                          0                          0   82.64520   140.2288
## 4                          0                          0   79.63162   137.2352
## 5                          0                          0   83.36537   140.7942
## 6                          0                          0  123.19383   181.5911
##   trend_lower trend_upper      yhat
## 1    107.6954    107.6954  86.34520
## 2    109.8329    109.8329  79.83889
## 3    111.7635    111.7635 111.29990
## 4    113.9010    113.9010 109.00124
## 5    115.9695    115.9695 112.36236
## 6    118.1070    118.1070 152.51143
```

Section 1.4: Visualising the Forecast 🗒️

Now, let's visualize the forecast to see what Prophet predicts for the next two years.

```
# Plot the forecast
plot(model, forecast)
```

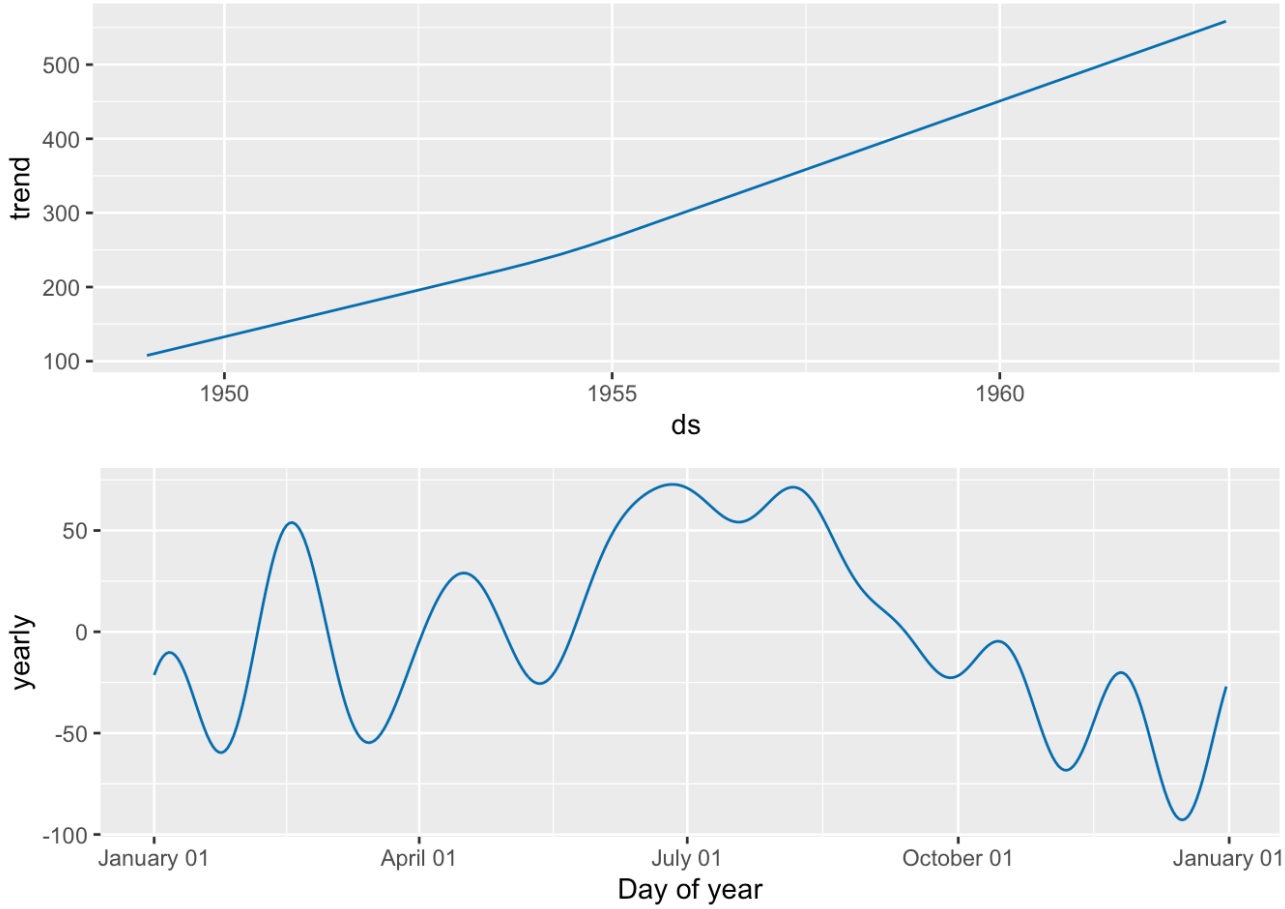


Prophet's predictions are depicted by the blue line in the graph, while the shaded area marks the uncertainty intervals.

🔍 Looking at Components

Prophet also allows us to break down the forecast into its components: trend, yearly seasonality, and weekly seasonality.

```
# Plot the forecast components
prophet_plot_components(model, forecast)
```



This breakdown helps us understand the underlying patterns in air travel demand. The strong yearly seasonality reflects how air travel fluctuates based on holidays, weather, and other seasonal factors.

Section 1.5: Insights and Interpretations 📊

We can clearly see that there is a:

Clear Upward Trend 📈 as Air Travel was increasing over time before 1960. This could possibly be the result of the rise of commercial aviation after World War 2 due to improvements in aircraft technology.

Strong Seasonality 🏝️ Peaks during summer months means that this was the most popular time to travel due to favourable flying conditions and more holidays are in the summer.

Section 1.6: Conclusions

In this project, we explored the Air Passengers dataset, built a Prophet forecasting model, and visualized trends in air travel. Prophet is an incredibly useful tool for making time series predictions, and this analysis could be expanded further by integrating external variables (like economic indicators or fuel prices) to improve accuracy.

This coursework provided a great opportunity to apply time series forecasting techniques and practice working with real-world datasets. Hopefully, this knowledge I have applied will be useful in my studies and in my future career! 🎯

Section 1.7: References 🔗

https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/AirPassengers.html (https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/AirPassengers.html)

```
## Warning: package 'astsa' was built under R version 4.3.3
```

```
## [1] 1951    1
```

```
## [1] 2022   10
```

```
## [1] 12
```

Section 2: This is the second section

2.1 Including image files

You can include images with `![](images/QMlogo.png)`

2.2 Including hyperlinks

You can write an URL by surrounding it by `<` and `>`, this will render like this: https://www.qmul.ac.uk (https://www.qmul.ac.uk)

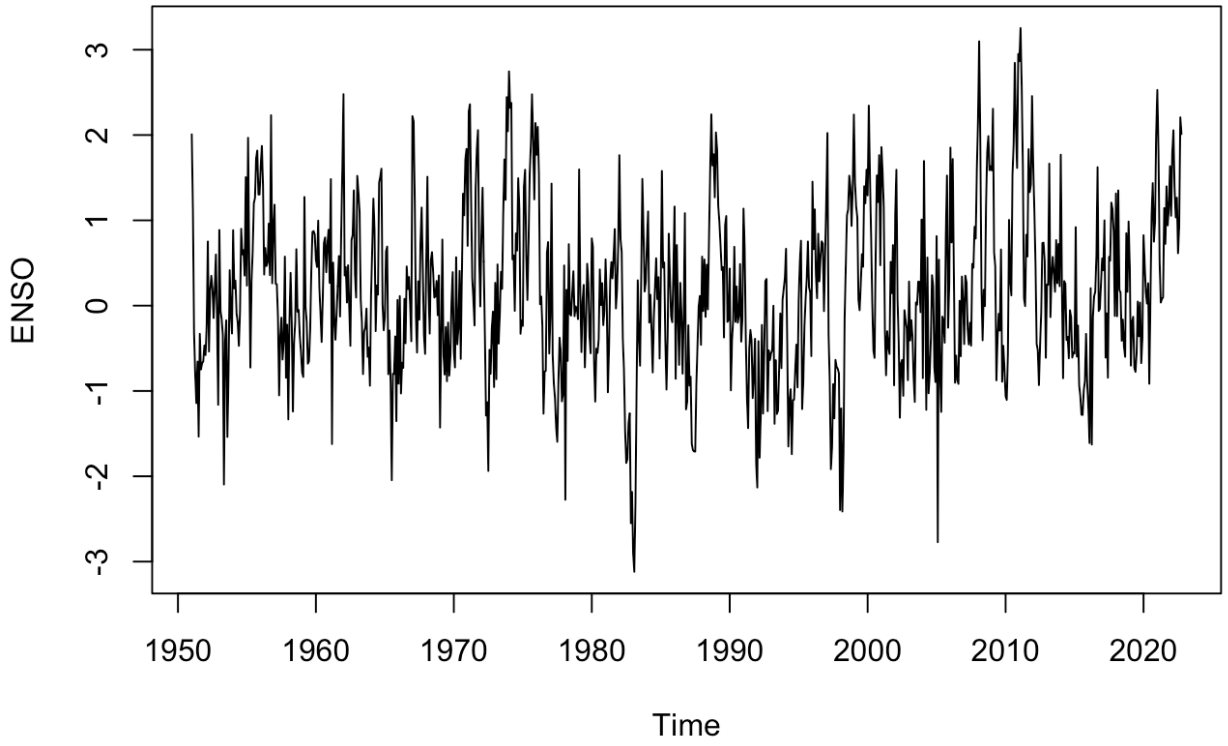
You can also hyperlink text to a URL using `[text](url)`. For example: `[Click here to go to the QMUL website](https://www.qmul.ac.uk)` results in

Click here to go to the QMUL website (https://www.qmul.ac.uk)

2.2 Displaying plots

You can also display plots. For example, the code below will display a plot of the ENSO data.

```
plot(ENSO)
```



2.3 Writing equations

In the course we will not use this much, but in case you want, you can also write equations using $\LaTeX$. For example, the equation of a straight line is $y = mx + b$.

$$z = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$$

etc.

References

- Datacamp course: https://www.datacamp.com/courses/reporting-with-rmarkdown (https://www.datacamp.com/courses/reporting-with-rmarkdown)
- RStudio reference: https://rmarkdown.rstudio.com/lesson-1.html (https://rmarkdown.rstudio.com/lesson-1.html)
- More sophisticated reference: https://bookdown.org/yihui/rmarkdown/ (https://bookdown.org/yihui/rmarkdown/)