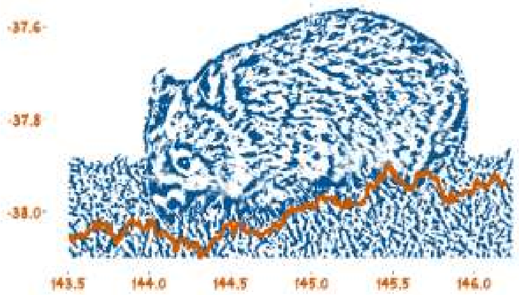


Workshop Organised by the Monash Business Analytics Team

WOMBAT

Open the World with Open Source

Time series Forecasting

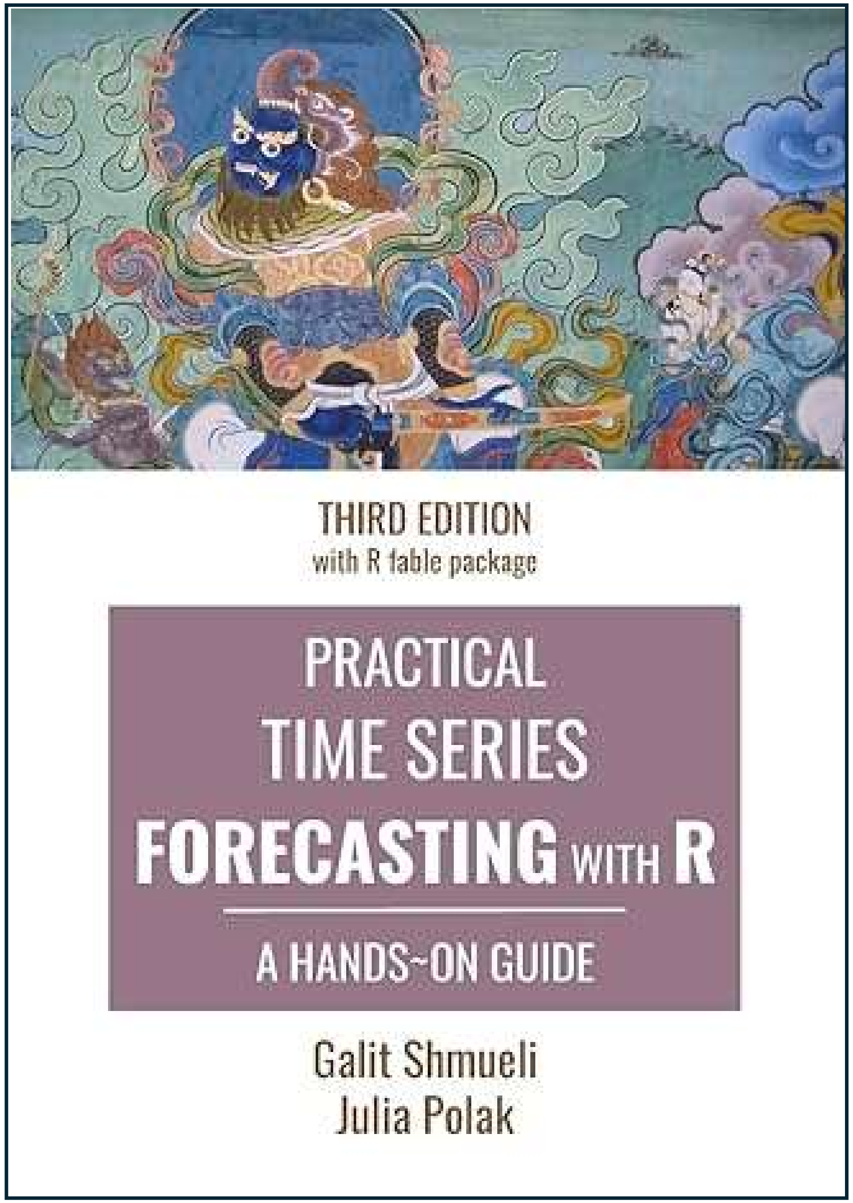
Part 1: Introduction

Dr Julia Polak

THE UNIVERSITY OF MELBOURNE

cebra
Centre of Excellence for Biosecurity Risk Analysis

This workshop will be based on ...



THIRD EDITION
with R fable package

PRACTICAL
TIME SERIES
FORECASTING WITH R

A HANDS~ON GUIDE

Galit Shmueli
Julia Polak

Program

1:30-3:00	Introduction and all you need to know about time series analysis Exponential smoothing
3:00-3:30	BREAK
3:30-5:00	Exponential smoothing Regression models ARIMA

Time series vs. Cross-sectional data

Time series data

Data collected over a period,
on one or many variables

Cross-sectional data

Data collected at one point of time,
but probable on many variables

- We will focus on time series at this workshop

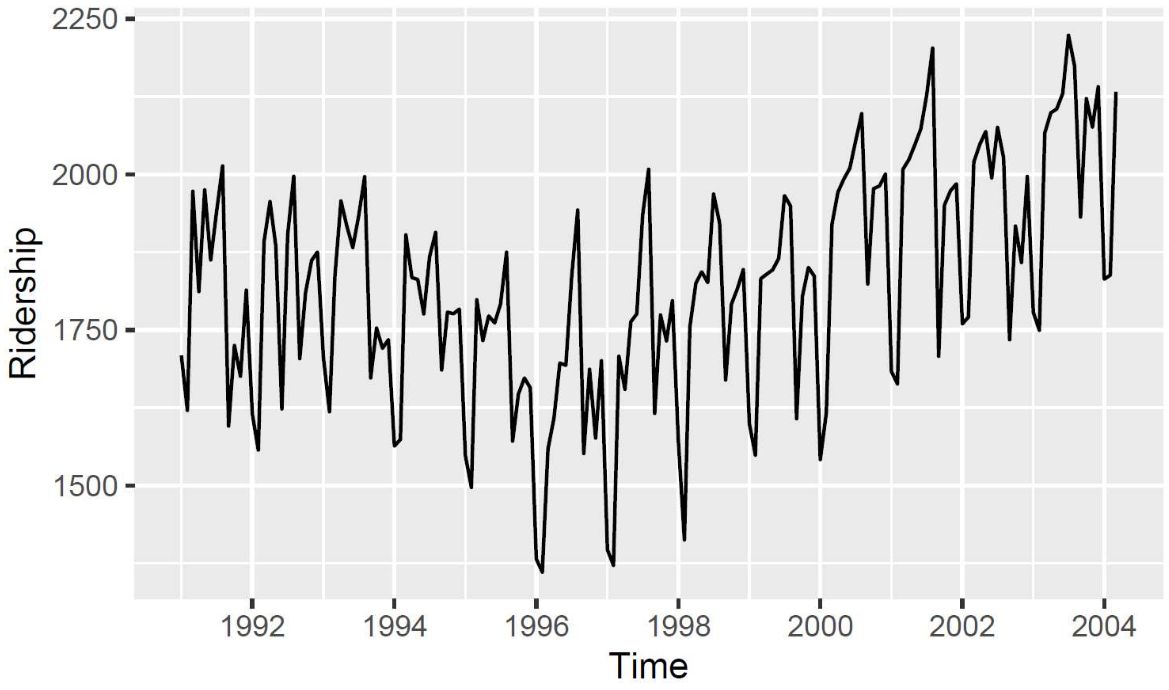
Predictive or descriptive

Predictive	Descriptive
Forecasting the future value	Understand the structure of the data
Out of sample	In sample
Learn => forecast	Learn => explain
	Detect all-time series components
Priority to methods that predict well the future (black-box methods are ok)	Priority to methods that provide explainable results (rather than black-box methods)

Example – Ridership on Amtrak Train

- Amtrak, a U.S. railway company, routinely collects data on ridership.
- Series of monthly Amtrak ridership between January 1991 and March 2004 in the United States.

Example – Ridership on Amtrak Train



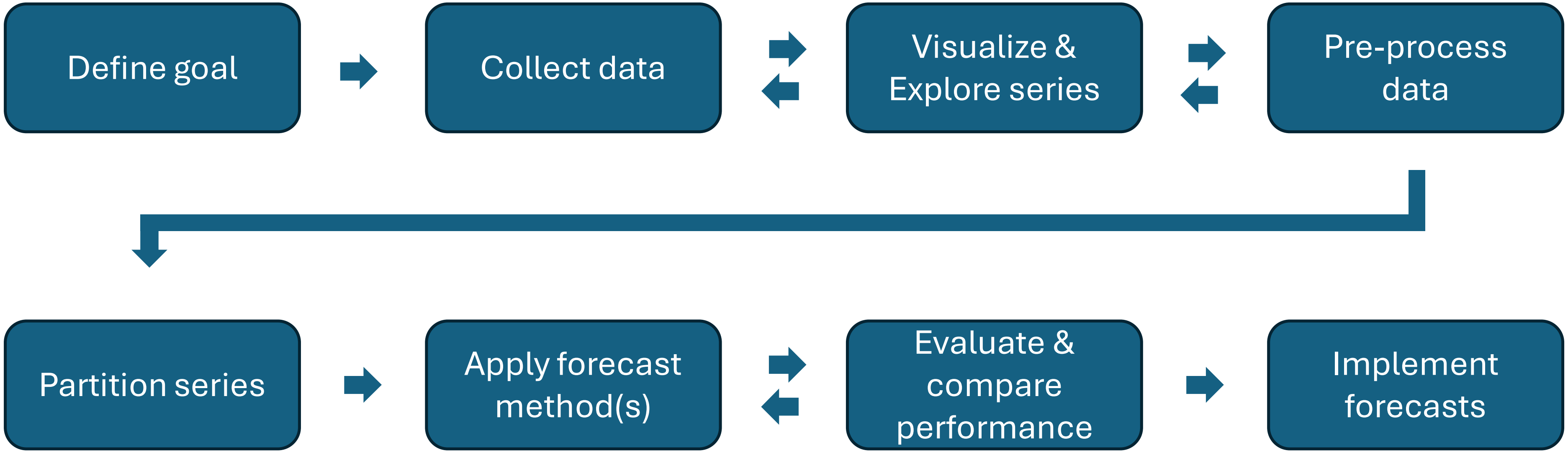
Potential goals:

Forecast future monthly ridership (for purposes of pricing).

Evaluate some effects in the past, for example, opening a new national highway.

Identify demands during different seasons for planning.

The forecasting process



Define goal

Collect data

Visualize & Explore series

Pre-process data

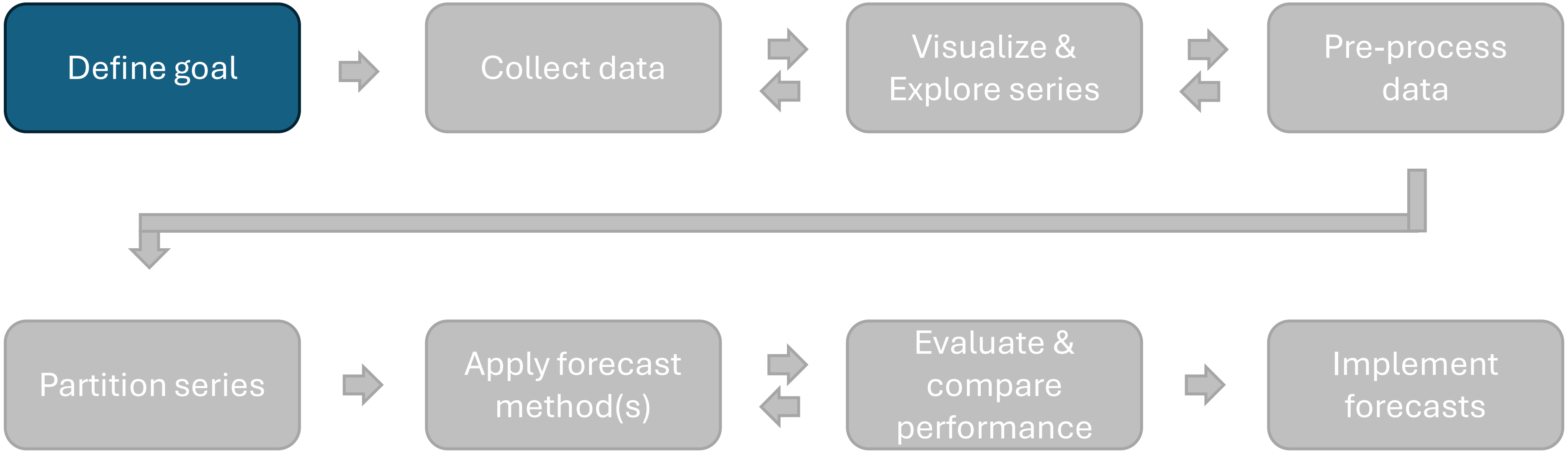
Partition series

Apply forecast method(s)

Evaluate & compare performance

Implement forecasts

The forecasting process



Define goal

Collect data

Visualize & Explore series

Pre-process data

Partition series

Apply forecast method(s)

Evaluate & compare performance

Implement forecasts

Goal definition

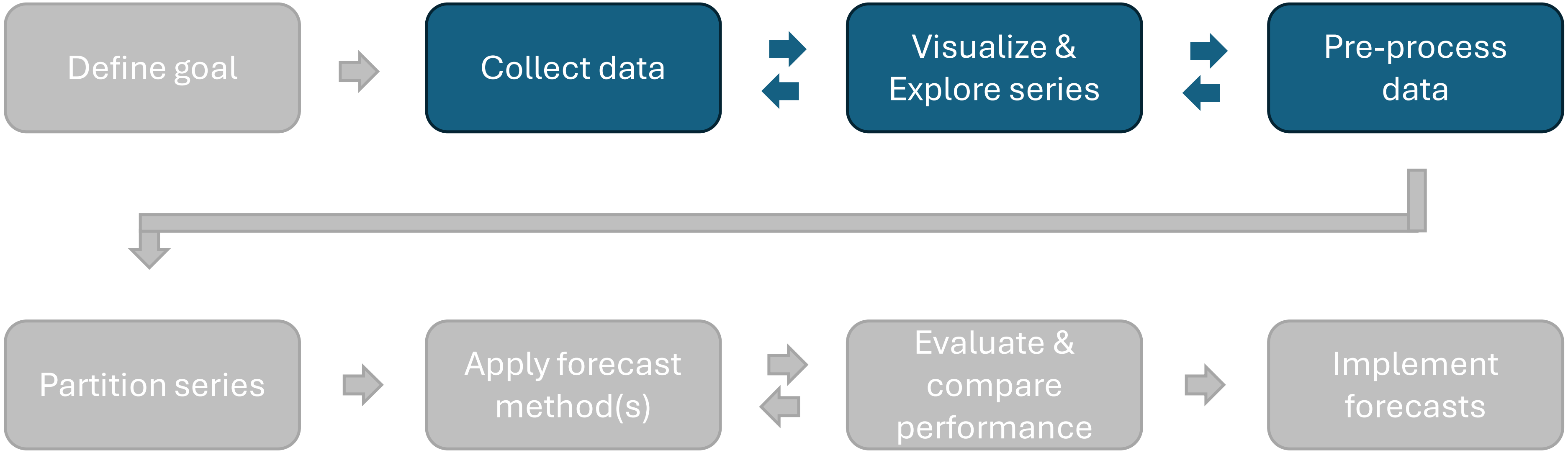
Issues to consider:

1. Forecast horizon (how far in the future)
2. Rolling forward or at a single time point?
3. One-time forecasting or ongoing tasks?
4. How will the forecast be used?
   - Who are the stakeholders
   - The cost of under-prediction & over-prediction
   - One-time forecasting or ongoing use?
5. Forecast expertise & automation
   - In-house forecasting or consultants?
   - How often re-fitting is planned?
   - How many time series?
   - Data and software availability

Implications:

1. How much data is needed
2. Chose of forecasting methods
3. Expected level of accuracy
4. Performance evaluations
5. Model deployment

The forecasting process



Define goal

Collect data

Visualize & Explore series

Pre-process data

Partition series

Apply forecast method(s)

Evaluate & compare performance

Implement forecasts

Data collection

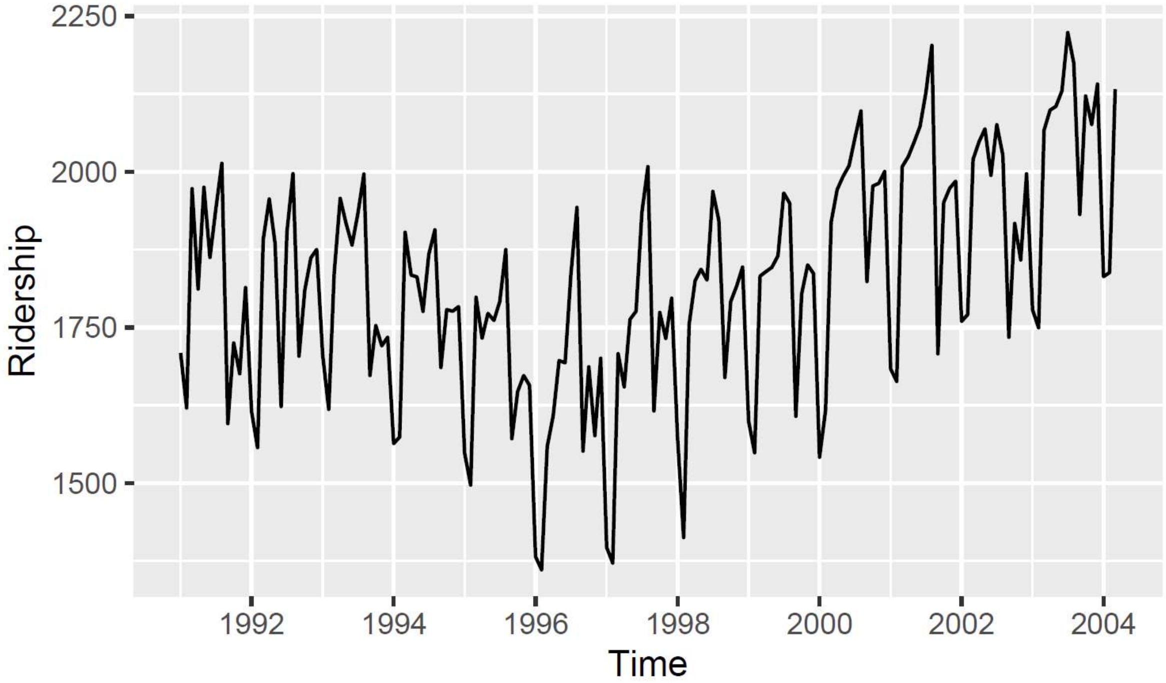
- Determined by our forecasting goal.
- Data quality -
  measurement accuracy, missing values, corrupted data and data entry errors.
- Temporal frequency –
  Today's technology allows us to collect data on a very frequent time scale (e.g., stock data on a minute level). But for modelling purposes, it is not always preferable (too much noise)

Visualizing Time series

Ridership on Amtrak Train

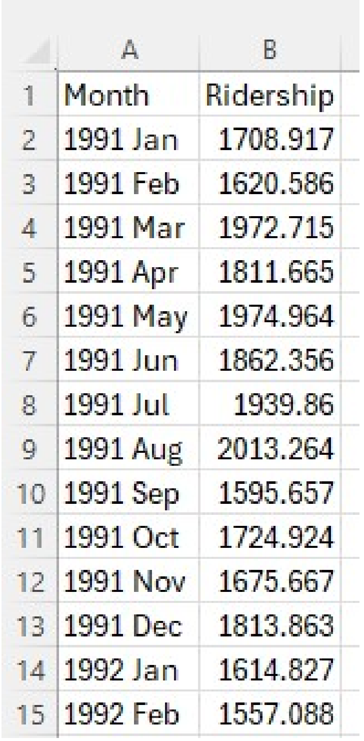
- Amtrak, a U.S. railway company, routinely collects data on ridership.
- Series of monthly Amtrak ridership in the U.S. between January 1991 and March 2004.

Example – Ridership on Amtrak Train



Monthly ridership on Amtrak trains (in thousands) from Jan-1991 to March-2004

```
Amtrak.data <- read.csv("Amtrak.csv")
ridership <- Amtrak.data |>
            mutate(Month = yearmonth(as.character(Amtrak.data$Month))) |>
            as_tsibble(index = Month)
ridership |>  autoplot(Ridership) +
            xlab("Time") + ylab("Ridership")
```



	A	B
1	Month	Ridership
2	1991 Jan	1708.917
3	1991 Feb	1620.586
4	1991 Mar	1972.715
5	1991 Apr	1811.665
6	1991 May	1974.964
7	1991 Jun	1862.356
8	1991 Jul	1939.86
9	1991 Aug	2013.264
10	1991 Sep	1595.657
11	1991 Oct	1724.924
12	1991 Nov	1675.667
13	1991 Dec	1813.863
14	1992 Jan	1614.827
15	1992 Feb	1557.088

Reading data into R (& create *tsibble* object)

```
1  Amtrak.data <- read.csv("Amtrak.csv")
   ridership <- Amtrak.data |>
2                mutate(Month = yearmonth(as.character(Amtrak.data$Month))) |>
                 as_tsibble(index = Month)
   ridership |>  autoplot(Ridership) +
3                xlab("Time") + ylab("Ridership")
```

1. The first line reads the CSV file into a data frame called Amtrak.data.
2. We create a *tsibble* object from *Amtrak.data*.
   - It includes the original info about the variable values plus extra information (e.g., a short title summarising the stored data, variable names, and the expected data type of each variable)
   - The *tsibble* object must have at least two columns:
     - 1st column: time stamps (time index),
     - 2nd column: values of the series (value).

Reading data into R (& create *tsibble* object)

```
1  Amtrak.data <- read.csv("Amtrak.csv")
   ridership <- Amtrak.data |>
2              mutate(Month = yearmonth(as.character(Amtrak.data$Month))) |>
               as_tsibble(index = Month)
   ridership |>  autoplot(Ridership) +
3              xlab("Time") + ylab("Ridership")
```

2. We create a *tsibble* object from *Amtrak.data*.
   - The time index can contain different frequencies, such as annual, quarterly, monthly, or daily.
   - Amtrak dataset has monthly measurements.
   - We specify that we have monthly data by applying the *yearmonth()* function to the ‘timestamp’ column, *Amtrak.data$Month*.
   - Because *Amtrak.data* contains only two columns, R assumes the 2nd column is the value column.
   - The newly created *tsibble* object is saved under the name *ridership*.

Reading data into R (& create *tsibble* object)

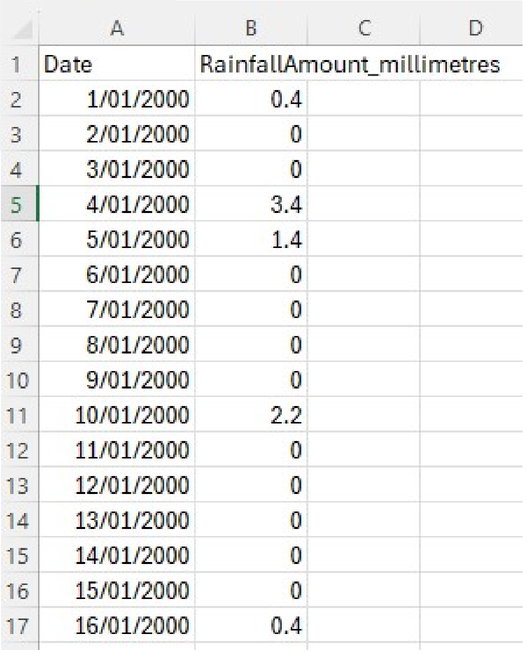
```
1  Amtrak.data <- read.csv("Amtrak.csv")
   ridership <- Amtrak.data |>
2               mutate(Month = yearmonth(as.character(Amtrak.data$Month))) |>
                as_tsibble(index = Month)
   ridership |>  autoplot(Ridership) +
3               xlab("Time") + ylab("Ridership")
```

3. Create a time series plot
   - *autoplot()* creates a ggplot of time series
   - We also add the x-axis and y-axis titles to be "Time" and "Ridership", respectively.

Creating *tsibble* object

- Examples of reading data sets with different time-frequency.
- Daily data

```
rain.df |>
     mutate(Date = dmy(as.character( rain.df$Date)) ) |>
     as_tsibble(index = Date)
```

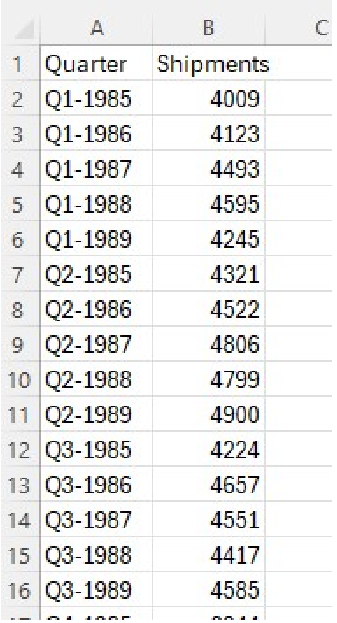


	A	B	C	D
1	Date	RainfallAmount_millimetres		
2	1/01/2000	0.4		
3	2/01/2000	0		
4	3/01/2000	0		
5	4/01/2000	3.4		
6	5/01/2000	1.4		
7	6/01/2000	0		
8	7/01/2000	0		
9	8/01/2000	0		
10	9/01/2000	0		
11	10/01/2000	2.2		
12	11/01/2000	0		
13	12/01/2000	0		
14	13/01/2000	0		
15	14/01/2000	0		
16	15/01/2000	0		
17	16/01/2000	0.4		

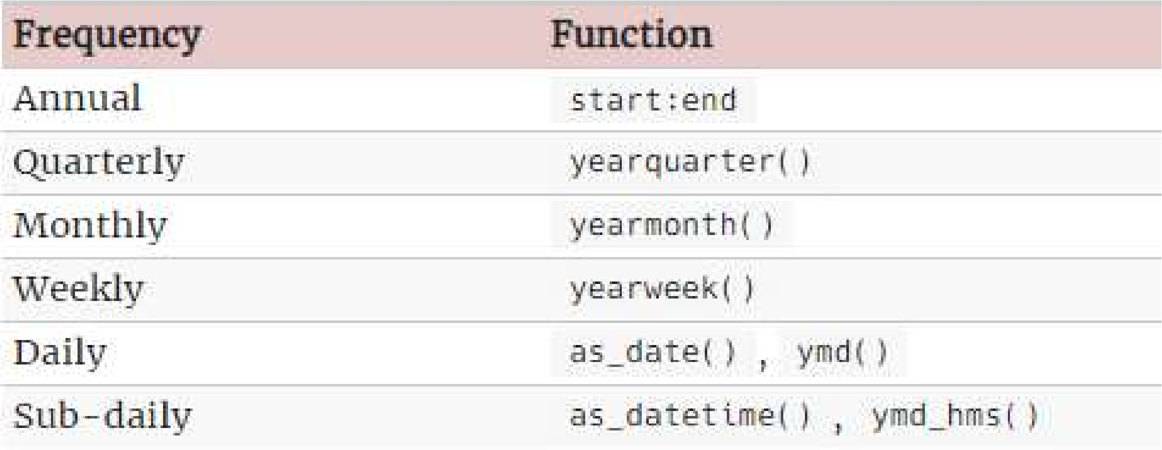
Creating *tsibble* object

- Examples of reading data sets with different time-frequency.
- Quarterly data

```
ApplianceShipments |>
  mutate(Quarter = yearquarter(as.character( ApplianceShipments$Quarter)) ) |>
 as_tsibble(index = Quarter)
```



	A	B	C
1	Quarter	Shipments	
2	Q1-1985	4009	
3	Q1-1986	4123	
4	Q1-1987	4493	
5	Q1-1988	4595	
6	Q1-1989	4245	
7	Q2-1985	4321	
8	Q2-1986	4522	
9	Q2-1987	4806	
10	Q2-1988	4799	
11	Q2-1989	4900	
12	Q3-1985	4224	
13	Q3-1986	4657	
14	Q3-1987	4551	
15	Q3-1988	4417	
16	Q3-1989	4585	

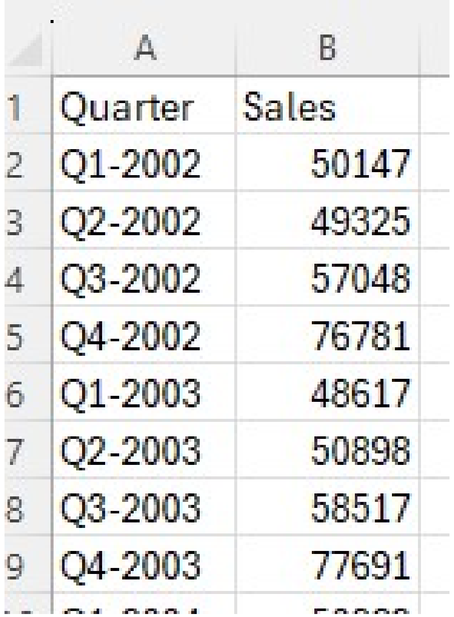


Frequency	Function
Annual	`start:end`
Quarterly	`yearquarter()`
Monthly	`yearmonth()`
Weekly	`yearweek()`
Daily	`as_date()` , `ymd()`
Sub-daily	`as_datetime()` , `ymd_hms()`

Forecasting: Principles and Practice (3rd ed)
Rob J Hyndman and George Athanasopoulos
https://otexts.com/fpp3/tsibbles.html

Hands-on

- Forecasting Department Store Sales
- The file *DepartmentStoreSales.csv* contains data on the quarterly sales for a department store over a 6-year period.
- Create a well-formatted time plot of the data.



	A	B
1	Quarter	Sales
2	Q1-2002	50147
3	Q2-2002	49325
4	Q3-2002	57048
5	Q4-2002	76781
6	Q1-2003	48617
7	Q2-2003	50898
8	Q3-2003	58517
9	Q4-2003	77691

Visualising Time series - Time series components

- To choose the model, we first need to understand the systematic and non-systematic parts of the series.

Systematic	
Level	The average value of the series
Trend	The change in series from one period to the next one
Seasonality	The short-term cyclical behaviour
Correlation	The correlation between neighbouring values in a series

Non-systematic	
Noise	

Visualising Time series - Time series components

- Relationships between the systematic components can be additive or multiplicative

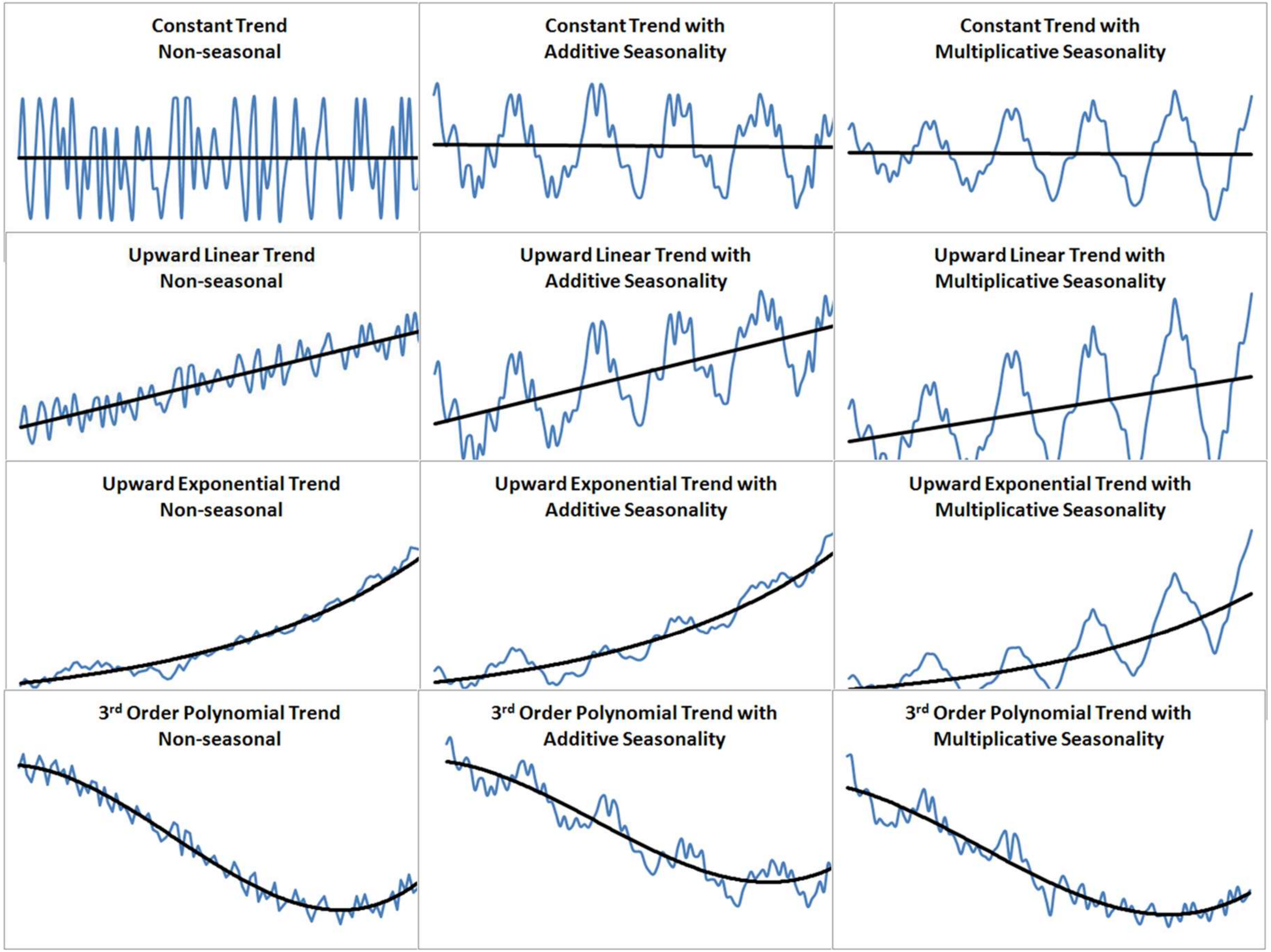
A time series with additive components can be written as:

$$y_t = \boxed{\text{Level}} + \boxed{\text{Trend}} + \boxed{\text{Seasonality}} + \boxed{\text{Noise}}$$

A time series with multiplicative components can be written as:

$$y_t = \boxed{\text{Level}} \times \boxed{\text{Trend}} \times \boxed{\text{Seasonality}} \times \boxed{\text{Noise}}$$

Visualising Time series



Constant Trend Non-seasonal

Constant Trend with Additive Seasonality

Constant Trend with Multiplicative Seasonality

Upward Linear Trend Non-seasonal

Upward Linear Trend with Additive Seasonality

Upward Linear Trend with Multiplicative Seasonality

Upward Exponential Trend Non-seasonal

Upward Exponential Trend with Additive Seasonality

Upward Exponential Trend with Multiplicative Seasonality

3rd Order Polynomial Trend Non-seasonal

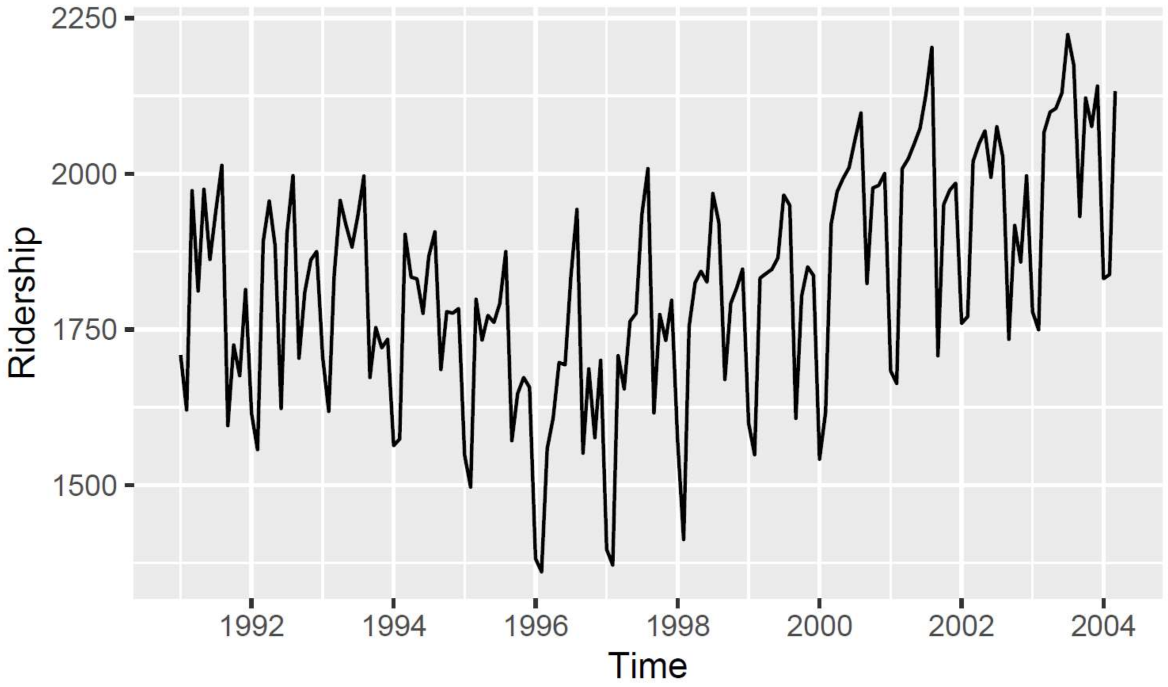
3rd Order Polynomial Trend with Additive Seasonality

3rd Order Polynomial Trend with Multiplicative Seasonality

Additive seasonality where values in different seasons vary by a constant amount.

Multiplicative seasonality where values in different seasons vary by a percentage.

Example – Ridership on Amtrak Train



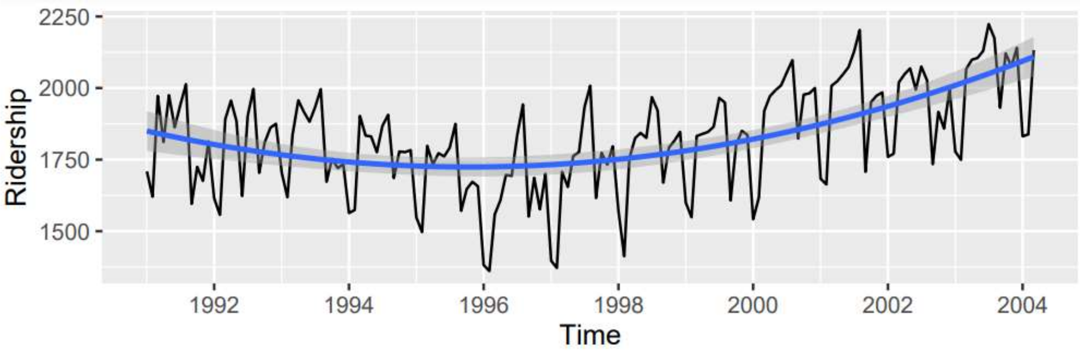
Monthly ridership on Amtrak trains (in thousands) from Jan-1991 to March-2004

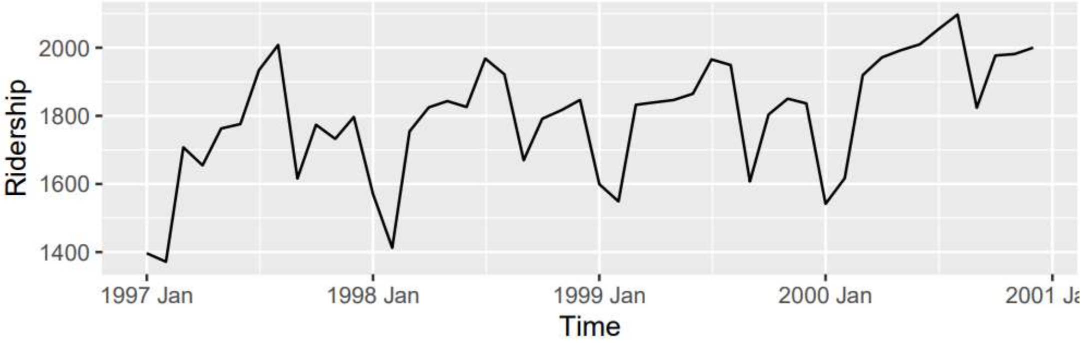
- This is a *time plot*.
- Plots help to understand the systematic parts of the time series.
- Helps to spot issues with time series, like missing values, extreme values, and unusual observations.
- On the plot, we can see a U-shape trend, and
  the general level at about 1800 passengers.
  Clear annual seasonality, peaks in summer (Jul & Aug).

```
Amtrak.data <- read.csv("Amtrak.csv")
ridership <- Amtrak.data |>
            mutate(Month = yearmonth(as.character(Amtrak.data$Month))) |>
            as_tsibble(index = Month)
ridership |>  autoplot(Ridership) +
            xlab("Time") + ylab("Ridership")
```

Example – Ridership on Amtrak Train

- Changing the scale helps us better identify the trend and seasonality.
- Here, we can see a clear U-shape trend.
- Other common trends are linear & exponential.
- Trend is easily spotted when seasonality is suppressed.
- Suppressing seasonality can be done by
  - Plotting the time series on a cruder time scale (yearly instead of monthly)
  - Separate plots for each season.
  - Moving average plots.

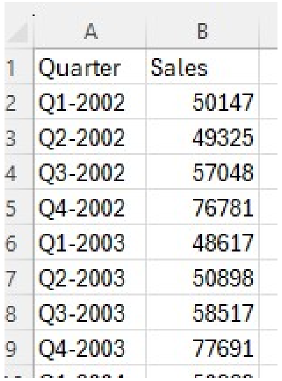




Visualising Time series - Time series components

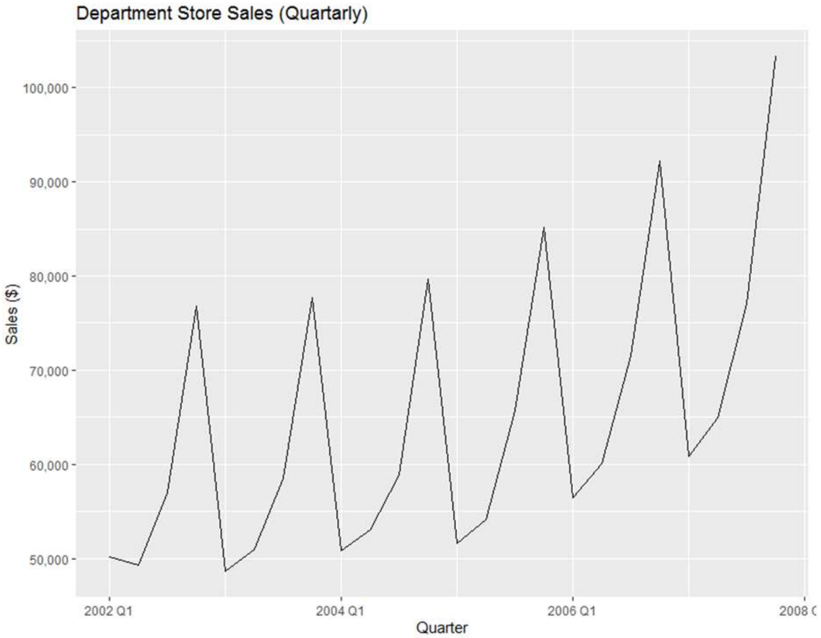
- The forecast method attempts to isolate the systematic parts and quantifies the noise level.
- The systematic parts are used to generate point forecast.
- The noise level helps assess the uncertainty associated with the point forecast.

Hands-on



	A	B
1	Quarter	Sales
2	Q1-2002	50147
3	Q2-2002	49325
4	Q3-2002	57048
5	Q4-2002	76781
6	Q1-2003	48617
7	Q2-2003	50898
8	Q3-2003	58517
9	Q4-2003	77691

- Forecasting Department Store Sales
- The file *DepartmentStoreSales.csv* contains data on the quarterly sales for a department store over a 6-year period.

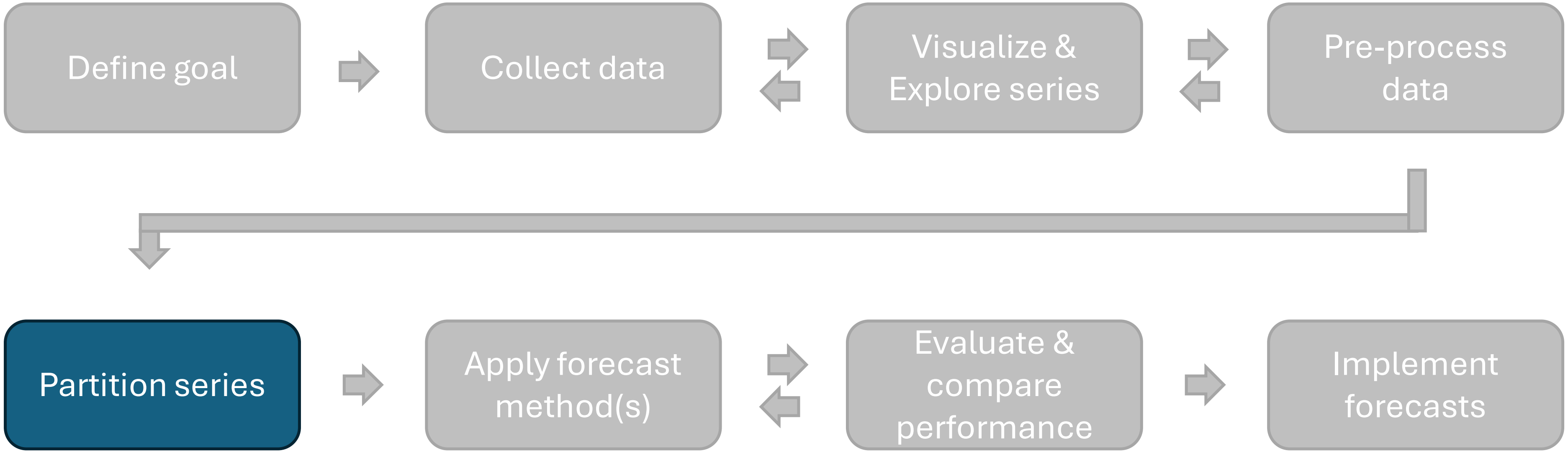


- Which of the four components (level, trend, seasonality, noise) are present in this series?

Data pre-processing

- **Missing values** – some models, such as ARIMA and smoothing methods, cannot be directly applied to time series because the relationship between consecutive periods is modelled directly.
- Be aware that **missing data** can affect the ability to forecast or evaluate the forecast.
  Solution: **imputation** (fitting the missing values with available past data or averaging neighbouring values)
- **Unequally spaced series** – similar to missing values, not all methods can handle unequally spaced series
- **Extreme values** – can affect different forecasting methods.
  The decision to remove or leave these values should be taken after understanding the information beyond the data.
- **Choose of time spam** – how far into the past data we need to consider.
  (consider significant changes that happened in the past)

The forecasting process



Partition series

- To avoid overfitting, the model performance is examined on a period different from the one the data was fitted on.

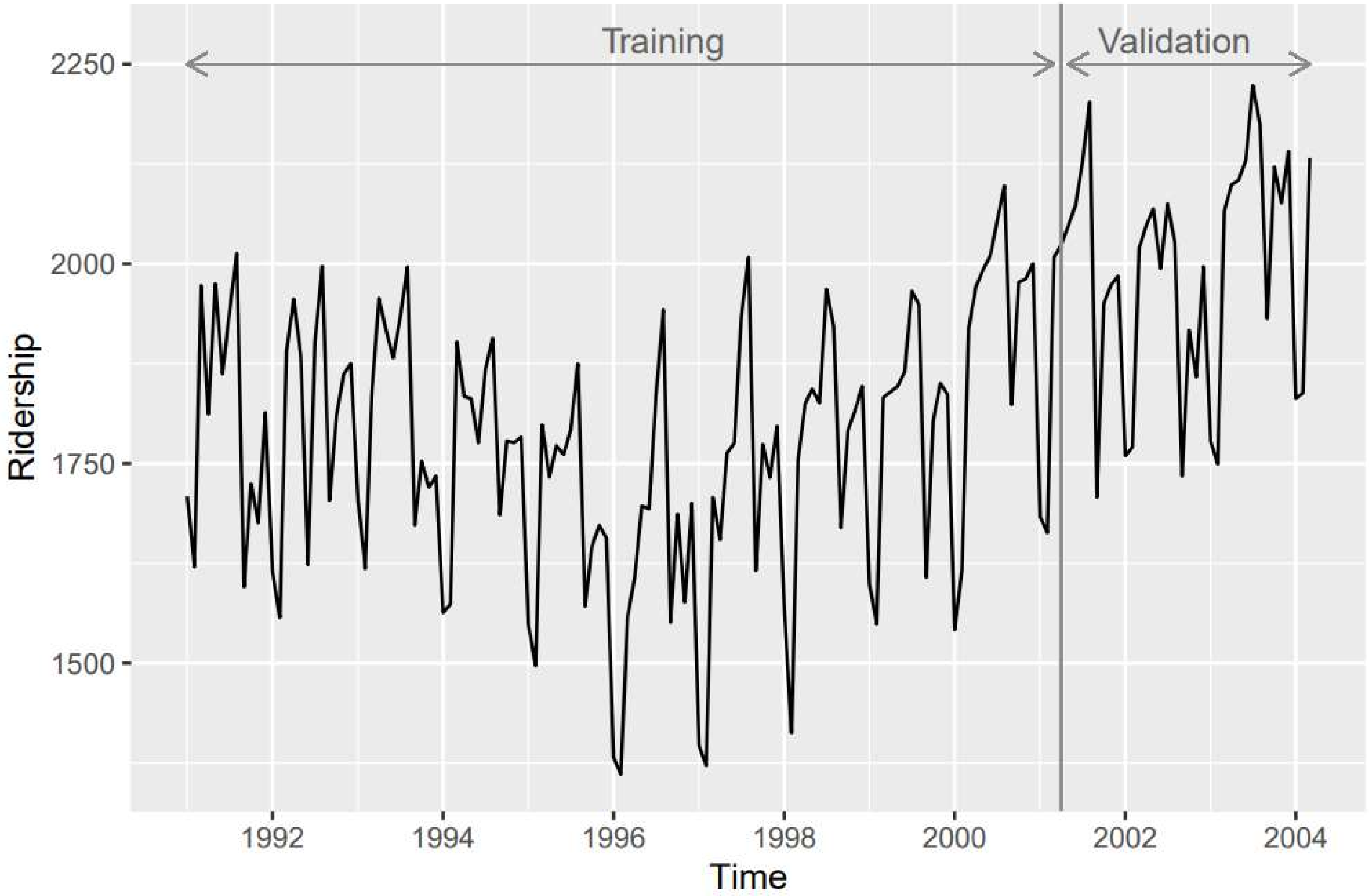
Training set – to fit the models

Validation set – to examine the performance of the fitted models and chose the 'best' model.

Testing set – to assess the performance of the chosen model with new data.

- In time series partitioning, we avoid using the testing set. Keeping the most reason data as the testing set will compromise our fitted model as we avoid using the most reason information.

Partition series



Partition series

Choosing the validation period.

- The validation period must mimic the forecasting horizon
- For example, we aim to forecast next year in the Amtrak data. Therefore, our validation period is going to be one year.
- Also, it is important to make sure we capture all the ‘futures’ of the time series in our validation period
  (weekends seasonality, at least one weekend in the validation period).