Introduction to Business Analytics

Lecture 10: Forecasting in R

Iegor Vyshnevskyi

Woosong University

May 2/8, 2023

Agenda

1. Intro to Forecasting
2. Time Series Data and Forecasting
3. Time Series Forecasting Applications
4. Time Series Components
5. Time Series Forecasting Methods
6. Time Series Forecasting Using the ARIMA Model
7. In-class Assignment

Acknowledgment: Used a number of open sources and materials from the web.

1. Intro to Forecasting

*What is **Forecasting?***

Forecasting is estimating how the sequence of observations will continue into the future.

Forecasting involves making predictions about the future.

Usefulness of Forecasting for Business Analytics

Forecasting is required in many situations:

- deciding whether to build another power generation plant in the next ten years requires forecasts of future demand.
- scheduling staff in a call centre next week requires forecasts of call volumes.
- stocking an inventory requires forecasts of stock requirements.

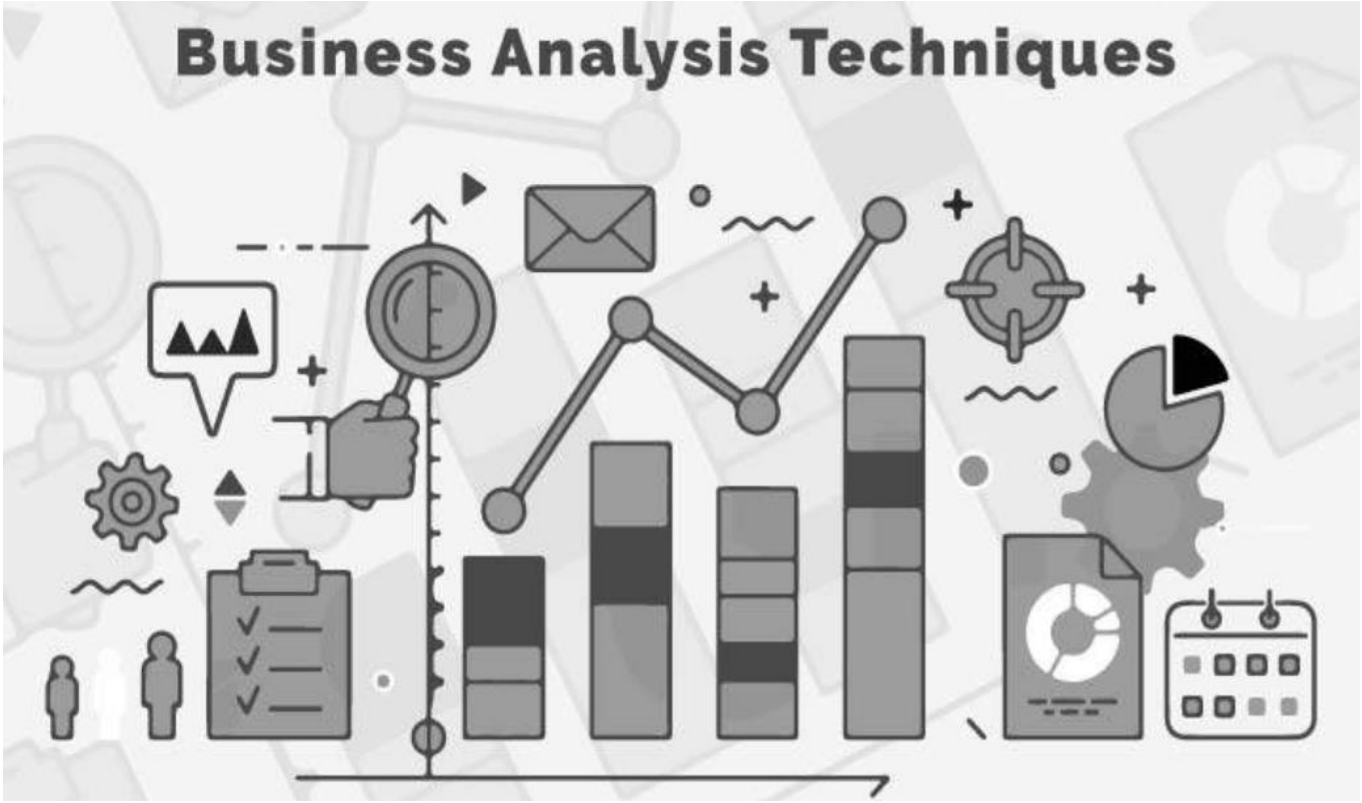
Forecasts can be required several years in advance (for the case of capital investments), or only a few minutes beforehand (for telecommunication routing).

Whatever the circumstances or time horizons involved, forecasting is an important aid to effective and efficient planning.

Usefulness of Forecasting for Business Analytics

Overall, forecasting can be a valuable tool for businesses looking to make data-driven decisions.

However, it should be used in conjunction with other analytical methods to ensure a comprehensive and accurate understanding of business data.



Advantages of using R for Time Series Forecasting

- ***Large community:*** R has a large and active community of users and developers, which means that there are many resources and packages available for time series forecasting, and it also allows for easy collaboration and sharing of knowledge.
- ***Flexibility:*** R provides a wide range of tools and packages for time series forecasting, which allows for flexibility in selecting the appropriate method for a given dataset.
- ***Open-source:*** R is an open-source programming language, which means that it is free to use and can be modified to fit specific needs.
- ***Easy to use:*** R has a simple and intuitive syntax, which makes it easy to learn and use.
- ***High-quality visualization:*** R has powerful data visualization capabilities, which allows for easy interpretation and analysis of time series data.

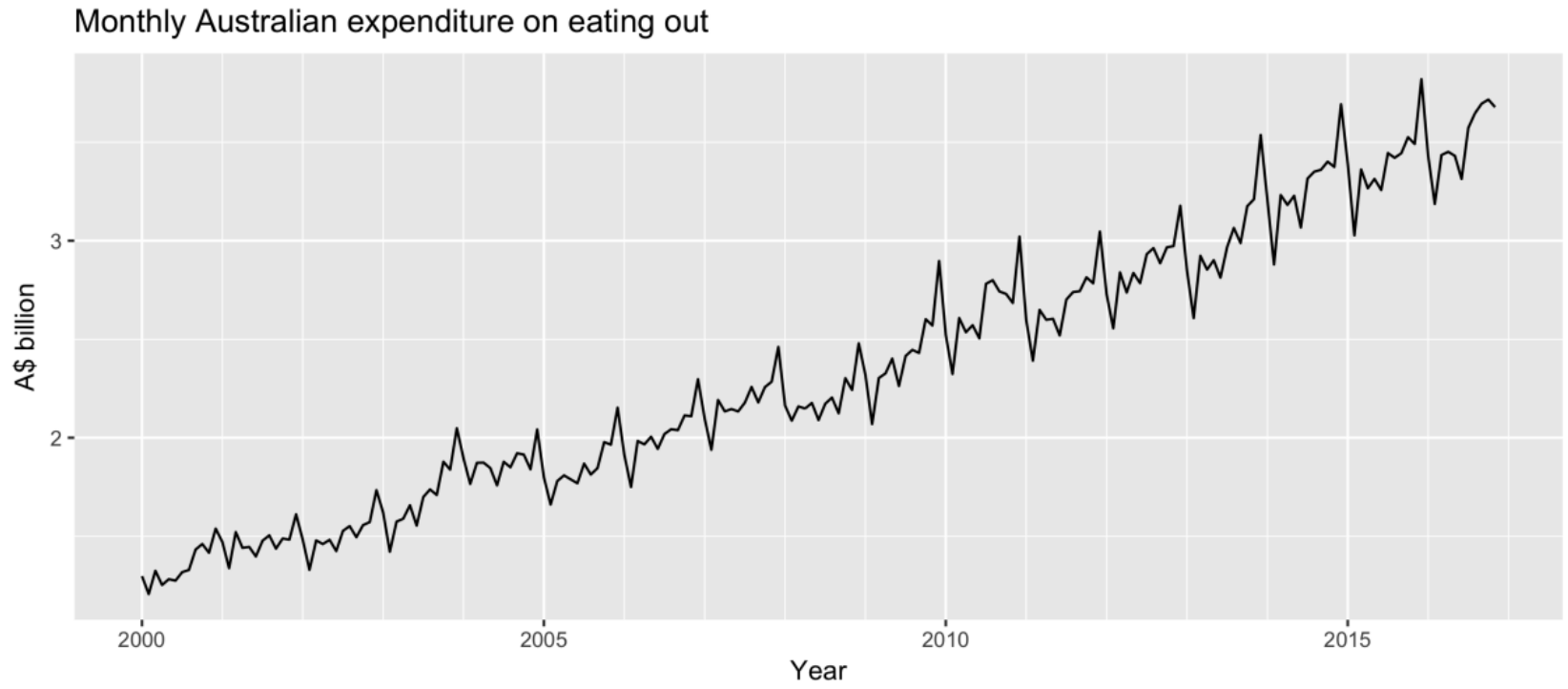
Disadvantages of using R for Time Series Forecasting

- ***Speed:*** R is an interpreted language, which can make it slower than compiled languages such as C or C++ for large datasets.
- ***Memory usage:*** R can be memory-intensive, which can be a problem for large datasets.
- ***Limited scalability:*** R is not designed for large-scale parallel computing, so it may not be suitable for large-scale time series forecasting tasks.
- ***Steep learning curve:*** R is a powerful programming language, but it has a steep learning curve, which can make it difficult for beginners.
- ***Lack of standardization:*** R provides a wide range of tools and packages for time series forecasting, which can lead to a lack of standardization in the way that time series forecasting tasks are performed, this could make it difficult to compare results across different studies.

2. Time Series Data and Forecasting

Time Series Data

- Series of data observed over time.
  - E.g.: Daily Samsung stock prices, monthly rainfall in Seoul,...
- Not all data that have time values or date values as its features can be considered as a time series data.



Time Series Forecasting

- *Time Series Forecasting* is the method of ***exploring and analyzing time-series data*** recorded or collected over a set period of time.
- This technique is used to ***forecast values and make future predictions***.

Any data fit for time series forecasting should consist of observations over a ***regular, continuous interval***.

3. Time Series Forecasting Applications

Examples of Forecasting Analysis

- Time series forecasting is used in ***stock price prediction*** to predict the closing price of the stock on each given day.
- E-Commerce and retail companies use forecasting to ***predict sales and units sold*** for different products.
- ***Weather prediction*** is another application that can be done using time series forecasting.
- It is used by government departments to predict ***a state's population***, at any particular region, or the nation as a whole.

4. Time Series Components

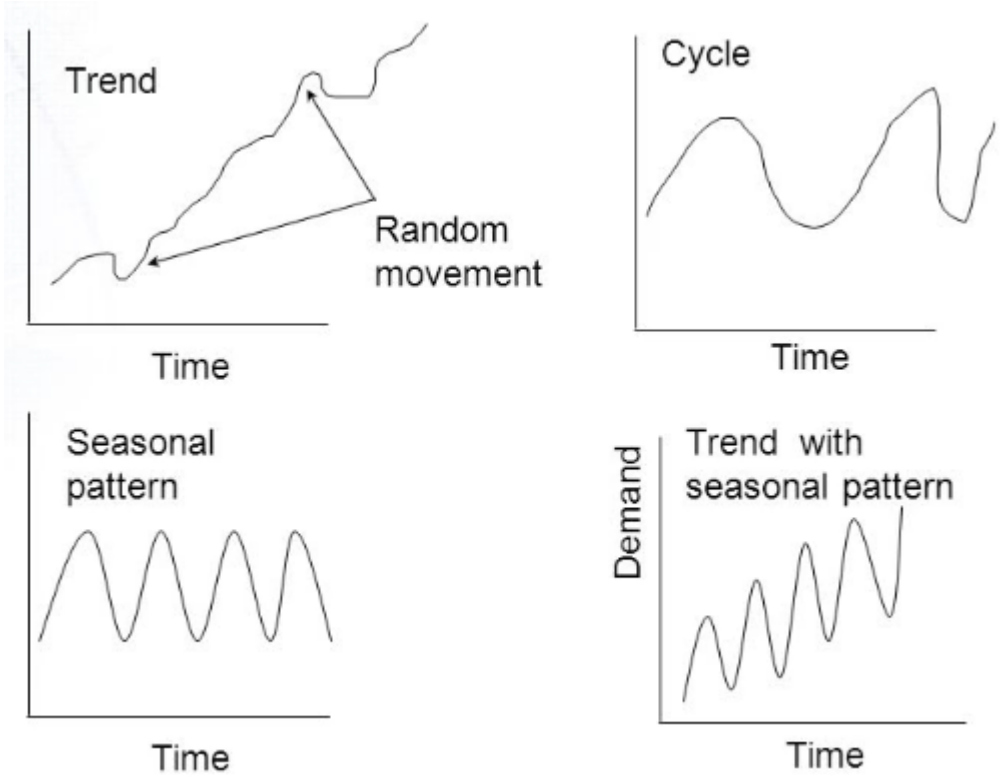
Components

- To use time-series data and develop a model, you *need to understand the patterns in the data over time*.

- These patterns are classified into four components, which are:
  - ***Trend:*** It represents the gradual change in the time series data. The trend pattern depicts long-term growth or decline.
  - ***Level:*** It refers to the baseline values for the series data if it were a straight line.
  - ***Seasonality:*** It represents the short-term patterns that occur within a single unit of time and repeats indefinitely.
  - ***Cyclic:*** A pattern exists where the data exhibits rises and falls that are not of fixed period (duration usually of at least 2 years)
  - ***Noise:*** It represents irregular variations and is purely random. These fluctuations are unforeseen, unpredictable, and cannot be explained by the model.

Seasonal vs cyclic

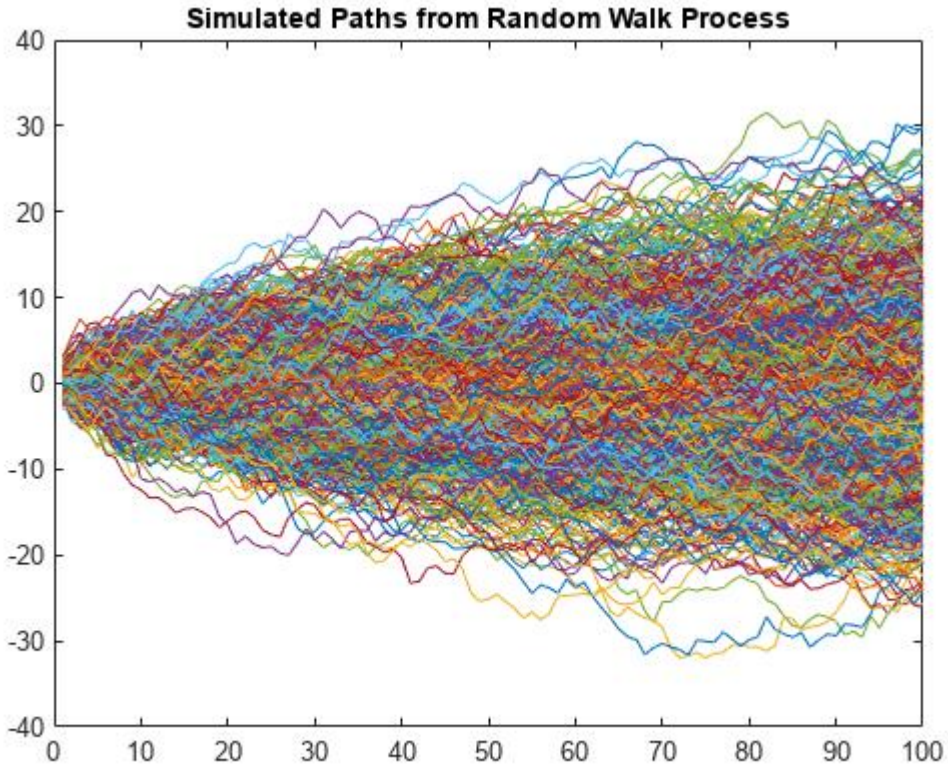
- Differences between seasonal and cyclic patterns:
  - Seasonal pattern constant length vs. cyclic pattern variable length
  - Average length of cycle longer than length of seasonal pattern
  - Magnitude of cycle more variable than magnitude of seasonal pattern
- The timing of peaks and troughs is predictable with seasonal data, but unpredictable in the long term with cyclic data.



5. Time Series Forecasting Methods

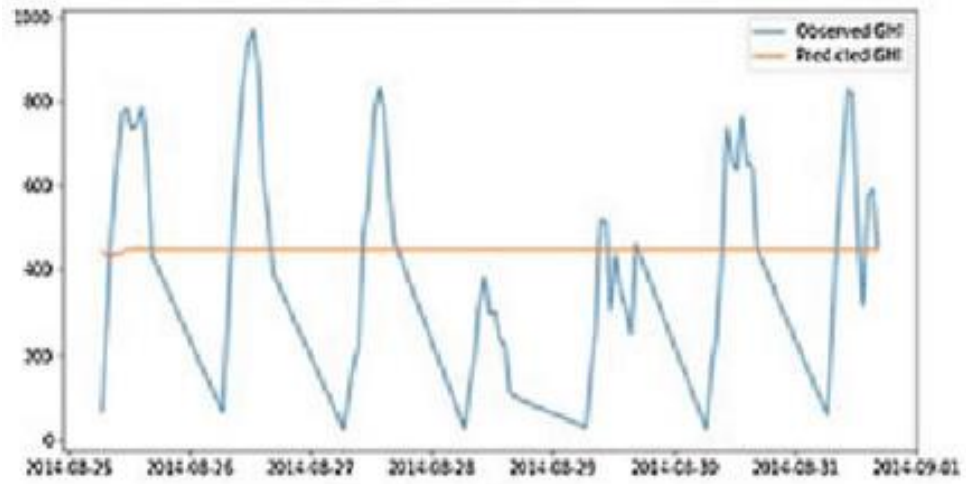
Methods: ARIMA Model

- ARIMA stands for Autoregressive Integrated Moving Average.
- It is a combination of the Autoregressive (AR) and Moving Average (MR) model.
- The AR model forecast corresponds to a linear combination of past values of the variable. The moving average model forecast corresponds to a linear combination of past forecast errors. The “I” represents the data values that are replaced by the difference between their values and the previous values.

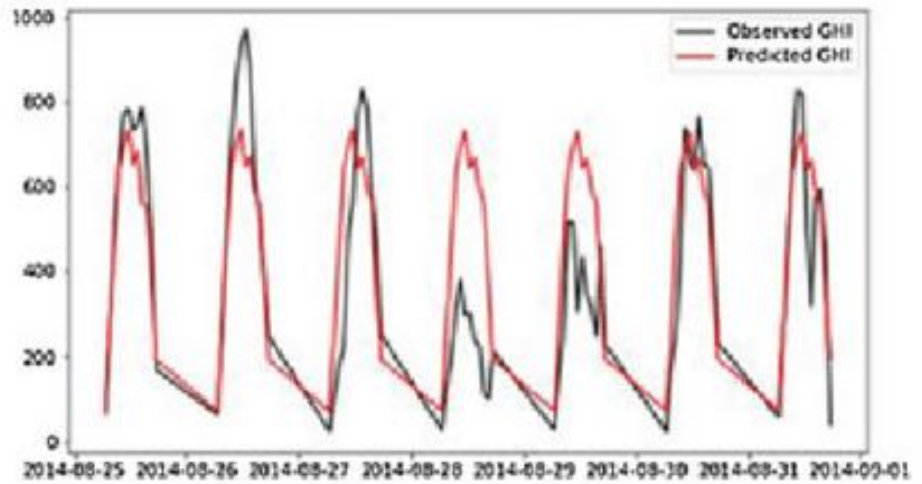


Methods: SARIMA Model

- SARIMA stands for Seasonal Autoregressive Integrated Moving Average.
- It extends the ARIMA model by adding a linear combination of seasonal past values and forecast errors.



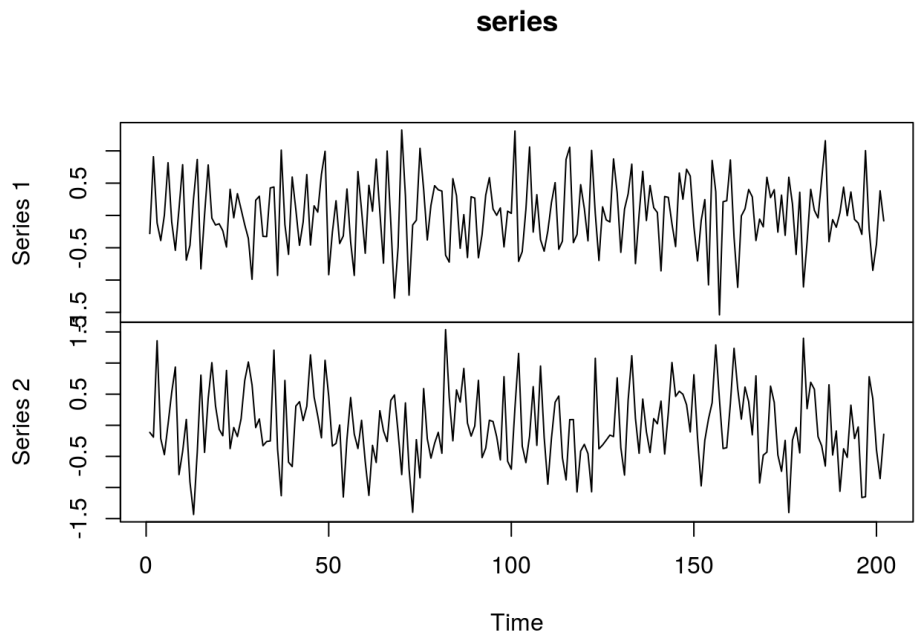
ARIMA 7 Day Ahead Prediction



SARIMA 7 Day Ahead Prediction

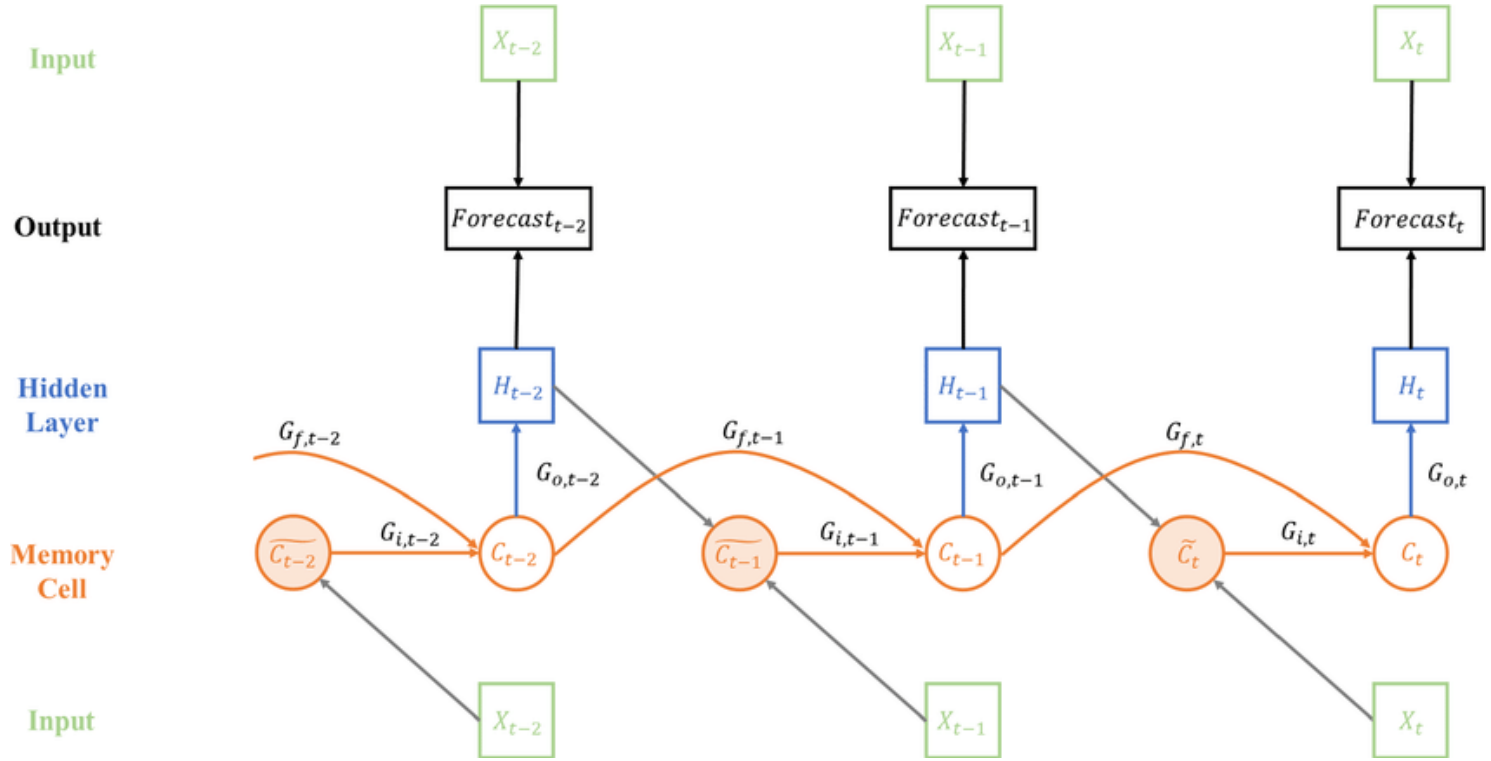
Methods: VAR

- The Vector Autoregression (VAR) method models the next step in each time series using an AR model.
- The VAR model is useful when you are interested in predicting multiple time series variables using a single model.



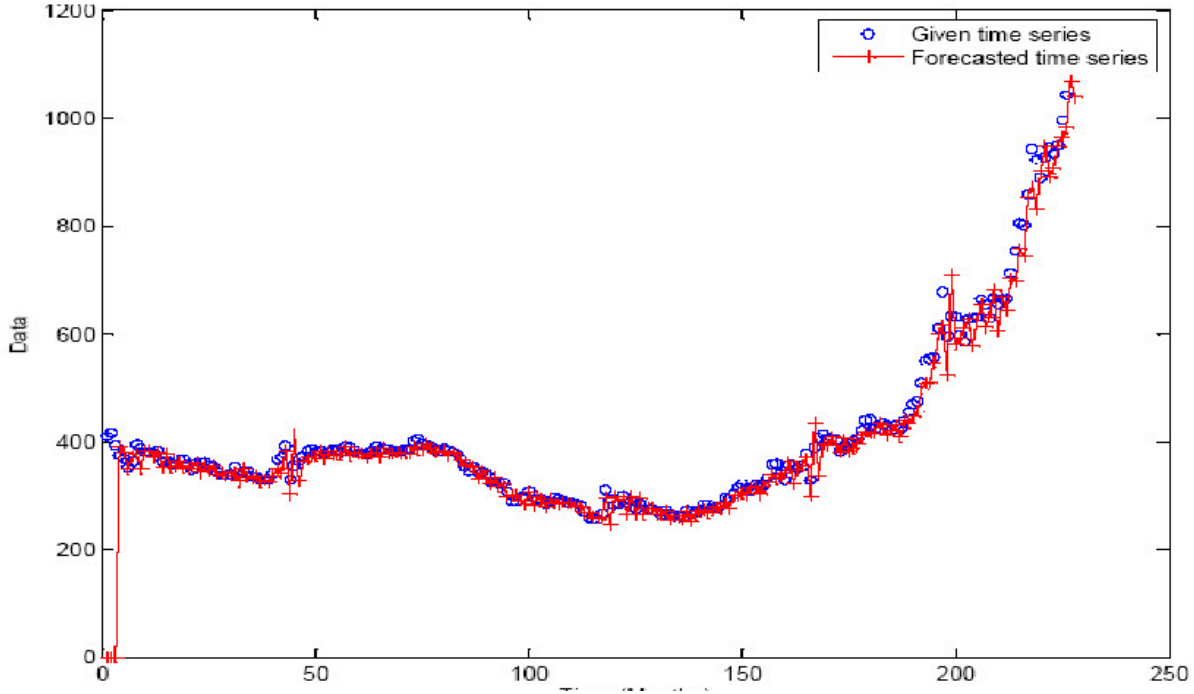
Methods: LSTM

- The Long Short Term Memory network or LSTM is a special kind of recurrent neural network that deals with long-term dependencies.
- It can remember information from past data and is capable of learning order dependence in sequence prediction problems.



Methods: GARCH

- The Generalised Autoregressive Conditional Heteroskedasticity (GARCH) models, most popular time series models used for forecasting conditional volatility.
- These models are conditional heteroskedastic as they take into account the conditional variance in a time series.
- GARCH models are one of the most widely used models for forecasting financial risk measures like VaR and Conditional VaR in financial risk modelling and management.



6. Time Series Forecasting Using the ARIMA Model

ARIMA Model: details

- ARIMA models are classified by three factors:
  - p = Number of autoregressive terms (AR).
  - d = How many non-seasonal differences are needed to achieve stationarity (I).
  - q = Number of lagged forecast errors in the prediction equation (MA).

What we do

- We’ll use a dataset with information about air-ticket sales of the airline industry from 1949-1960.
- We’ll predict the Airline tickets’ sales of 1961 using the ARIMA model in R.

- The idea for this analysis is to identify the time series components which are:
  - Trend
  - Seasonality
  - Random behavior of data

Then, we’ll forecast the values based on historical data.

Loading the data

```
# Install forecast library
install.packages('forecast')

# Load forecast library
library(forecast)
```

Install the FORECAST package first.

Load the data and check the class, because we need to have Time Series (i.e., ts) data

```
# Load the Air Passengers' dataset and view its class
data("AirPassengers")

class(AirPassengers)
```

```
> class(AirPassengers)
[1] "ts"
```

Runt AirPassengers to see our data

```
# Display the dataset
AirPassengers
```

```
> AirPassengers
     Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
1949 112 118 132 129 121 135 148 148 136 119 104 118
1950 115 126 141 135 125 149 170 170 158 133 114 140
1951 145 150 178 163 172 178 199 199 184 162 146 166
1952 171 180 193 181 183 218 230 242 209 191 172 194
1953 196 196 236 235 229 243 264 272 237 211 180 201
1954 204 188 235 227 234 264 302 293 259 229 203 229
1955 242 233 267 269 270 315 364 347 312 274 237 278
1956 284 277 317 313 318 374 413 405 355 306 271 306
1957 315 301 356 348 355 422 465 467 404 347 305 336
1958 340 318 362 348 363 435 491 505 404 359 310 337
1959 360 342 406 396 420 472 548 559 463 407 362 405
1960 417 391 419 461 472 535 622 606 508 461 390 432
```

Loading the data: exploration

```
# Let's check on our date values
start(AirPassengers) # start date

end(AirPassengers) # end date
```

```
> start(AirPassengers) # start date
[1] 1949    1
>
> end(AirPassengers) # end date
[1] 1960   12
```

Our start date is January 1949, while the end date is December 1960.

```
> # Find out if there are any missing values
> sum(is.na(AirPassengers))
[1] 0
```

Missing values check. None detected.

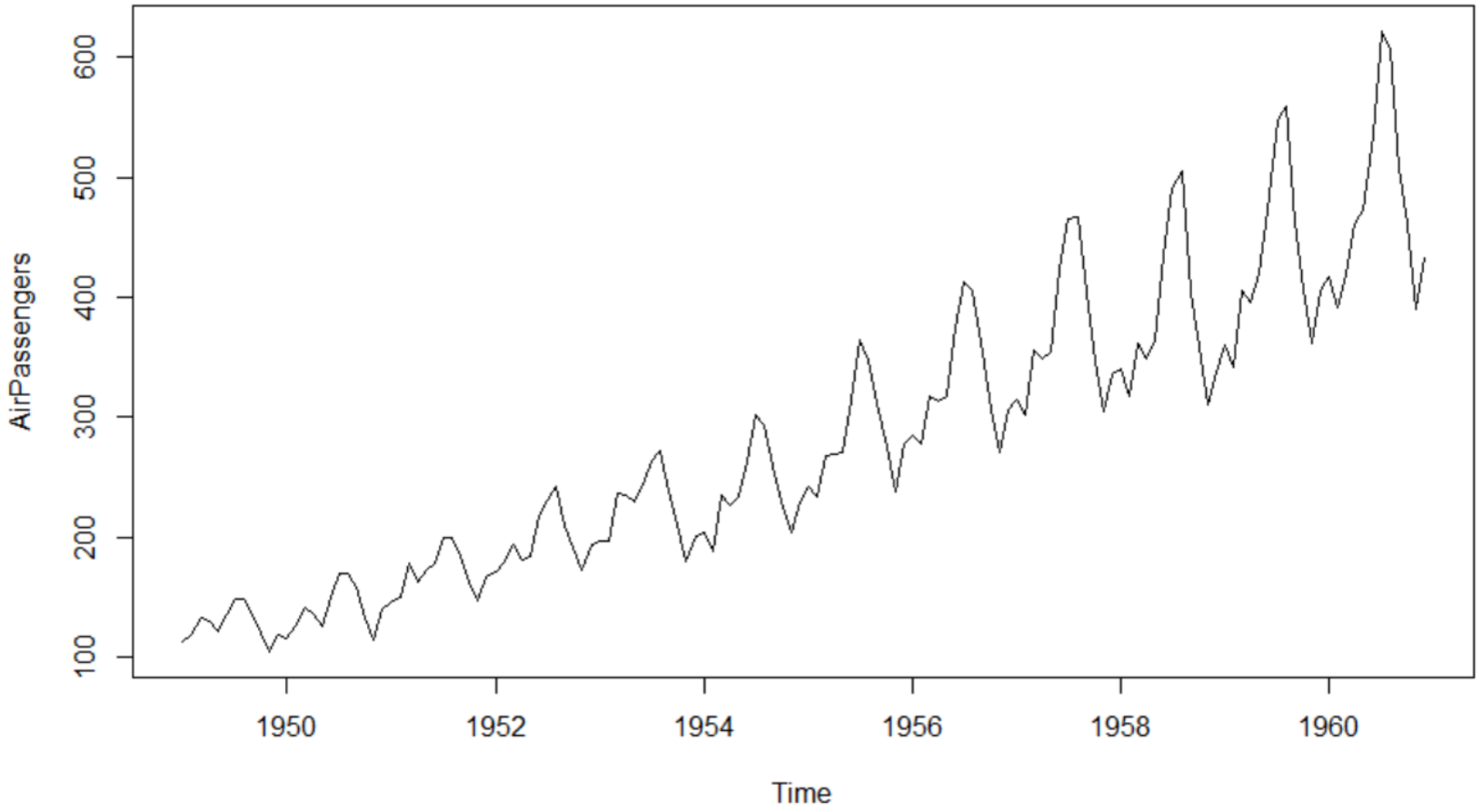
Checking the summary of our data

```
> # Check the summary of the dataset
> summary(AirPassengers)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  104.0   180.0   265.5   280.3   360.5   622.0
```

Loading the data: exploration

```
# Plot the dataset
plot(AirPassengers)
```

Visualization of our data.



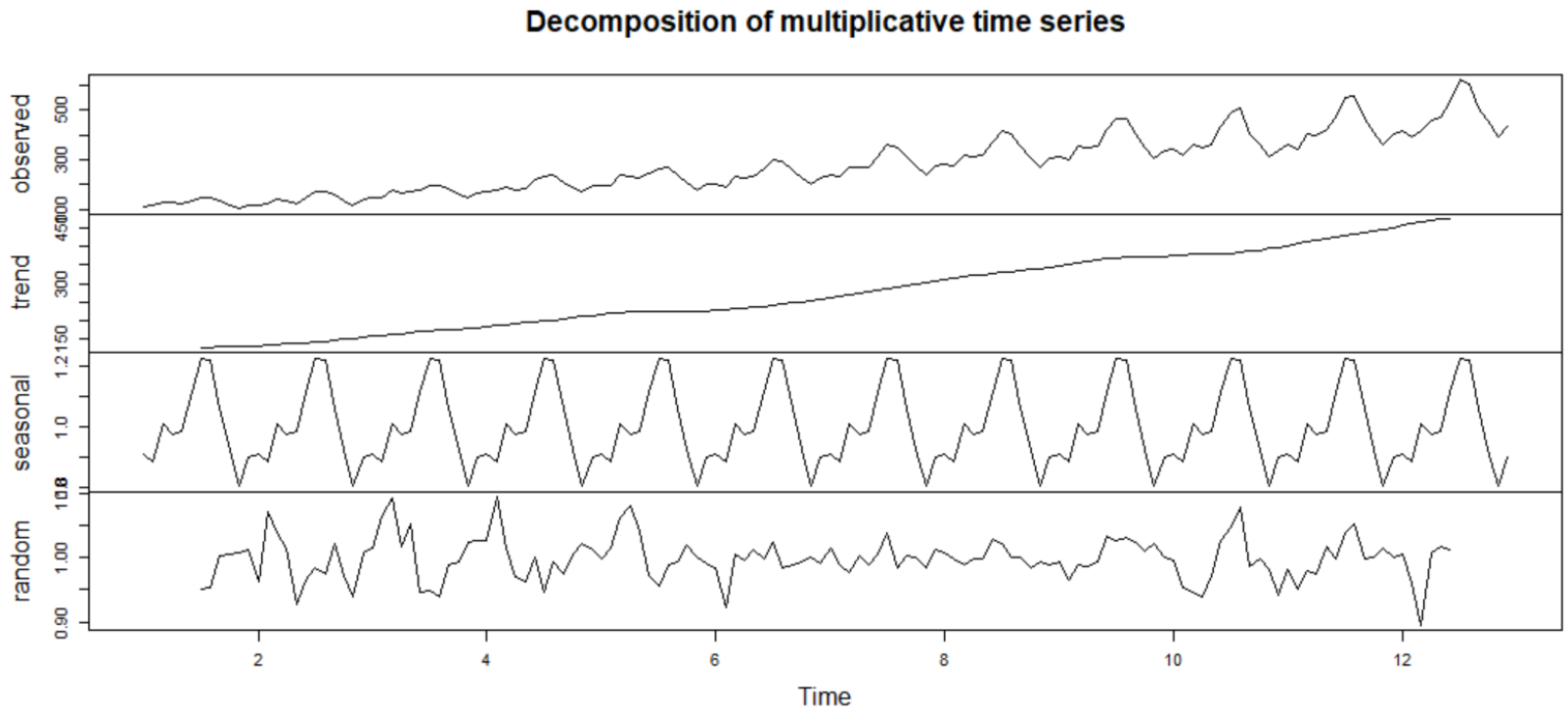
Decompose the data into four components

- Time series data are decomposed into three components :
  - *Seasonal* – Patterns that show how data is being changed over a certain period of time. Example – A clothing e-commerce website will have heavy traffic during festive seasons and less traffic during normal times. Here it is a seasonal pattern as value is being increased only at a certain period of time.
  - *Trend* – It is a pattern that shows how values are being changed. For example how a website is running overall if running successfully trend goes up, if not, the trend comes down.
  - *Random* – The remaining data of the time series after seasonal trends are removed is a random pattern. This is also known as noise.

Decompose the data into four components

```
# Decompose the data into four components and plot
tsdata <- ts(AirPassengers, frequency = 12)
ddata <- decompose(tsdata, "multiplicative")
plot(ddata)
```

The parameter *multiplicative* is added because time series data changes with the trend, if not so, such kinds of data are called “additive”.



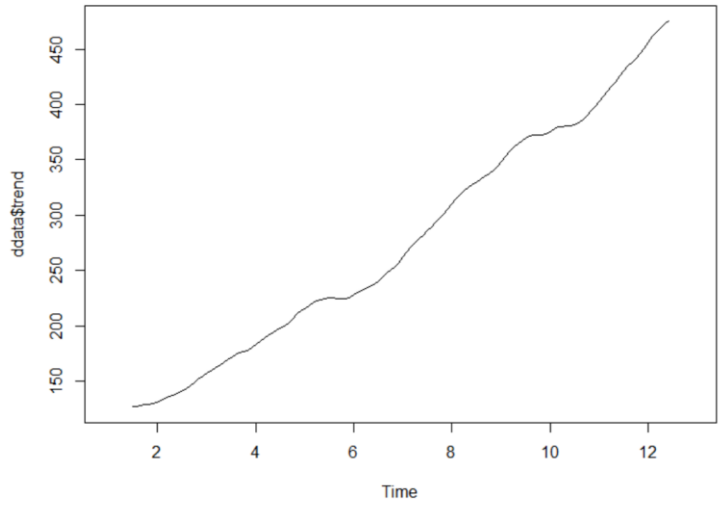
Our original data (in panel 1) is now broken into 3 components.

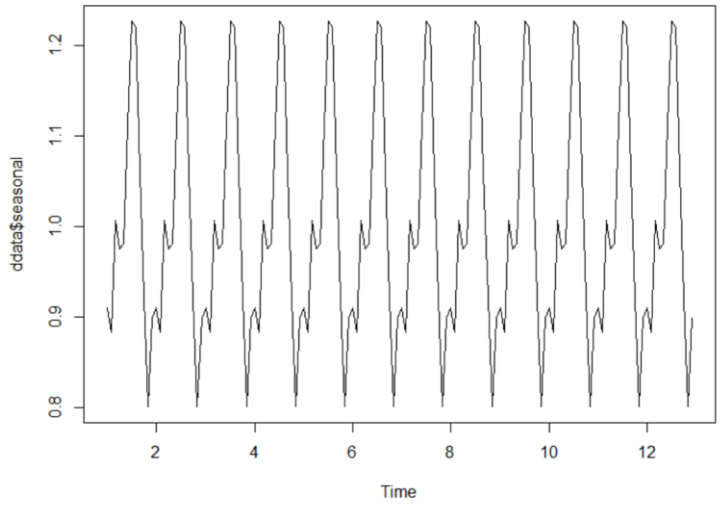
1. a trend component (panel 2)
2. a seasonal component (panel 3)
3. a random/remained (panel 4)

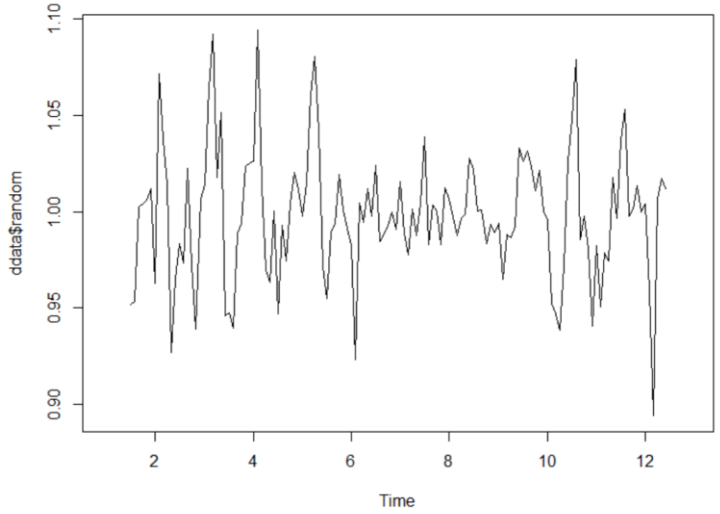
Decompose the data into four components

```
# Plot the different components individually
plot(ddata$trend)

plot(ddata$seasonal)

plot(ddata$random)
```

To check decomposition separately



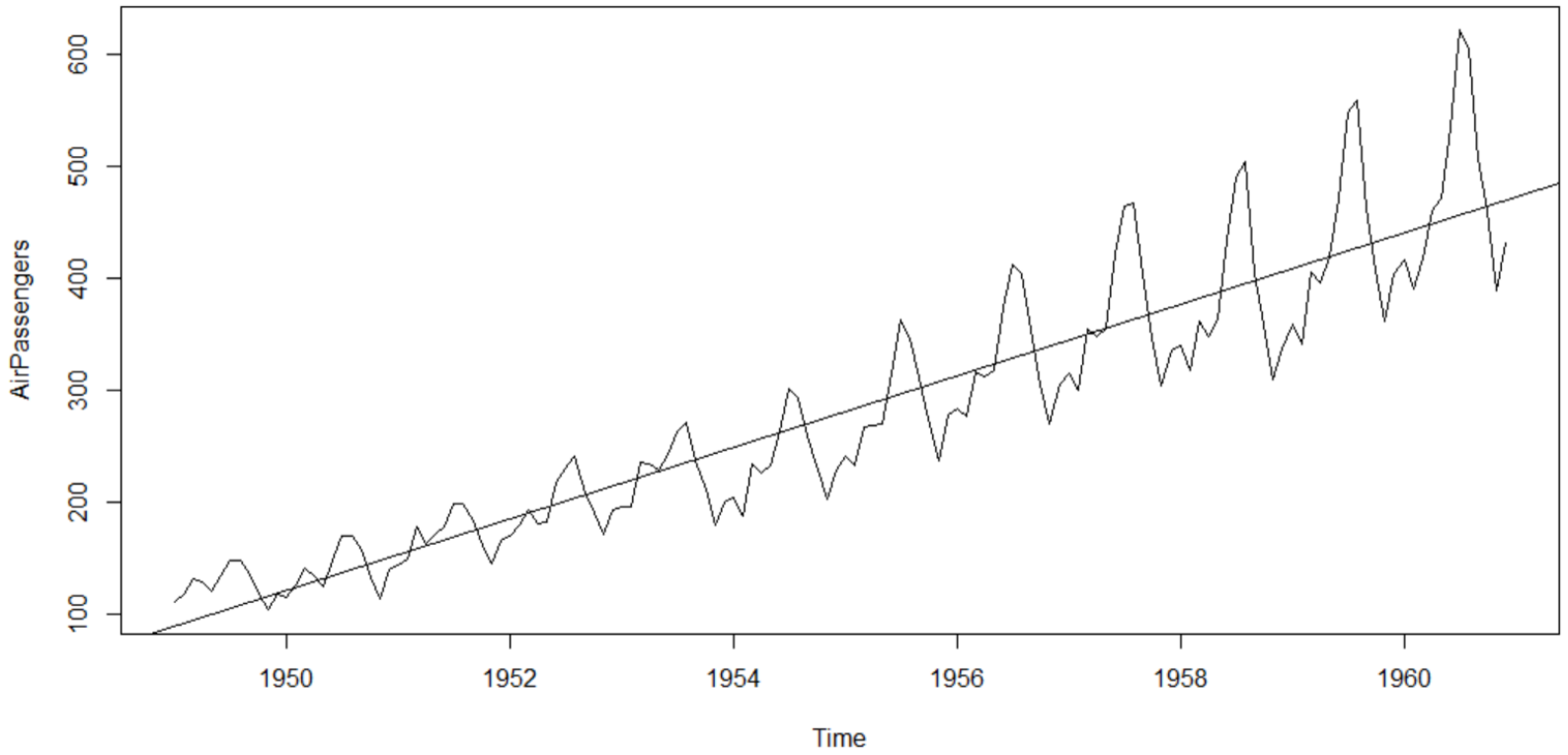




Plot a trendline on the original dataset

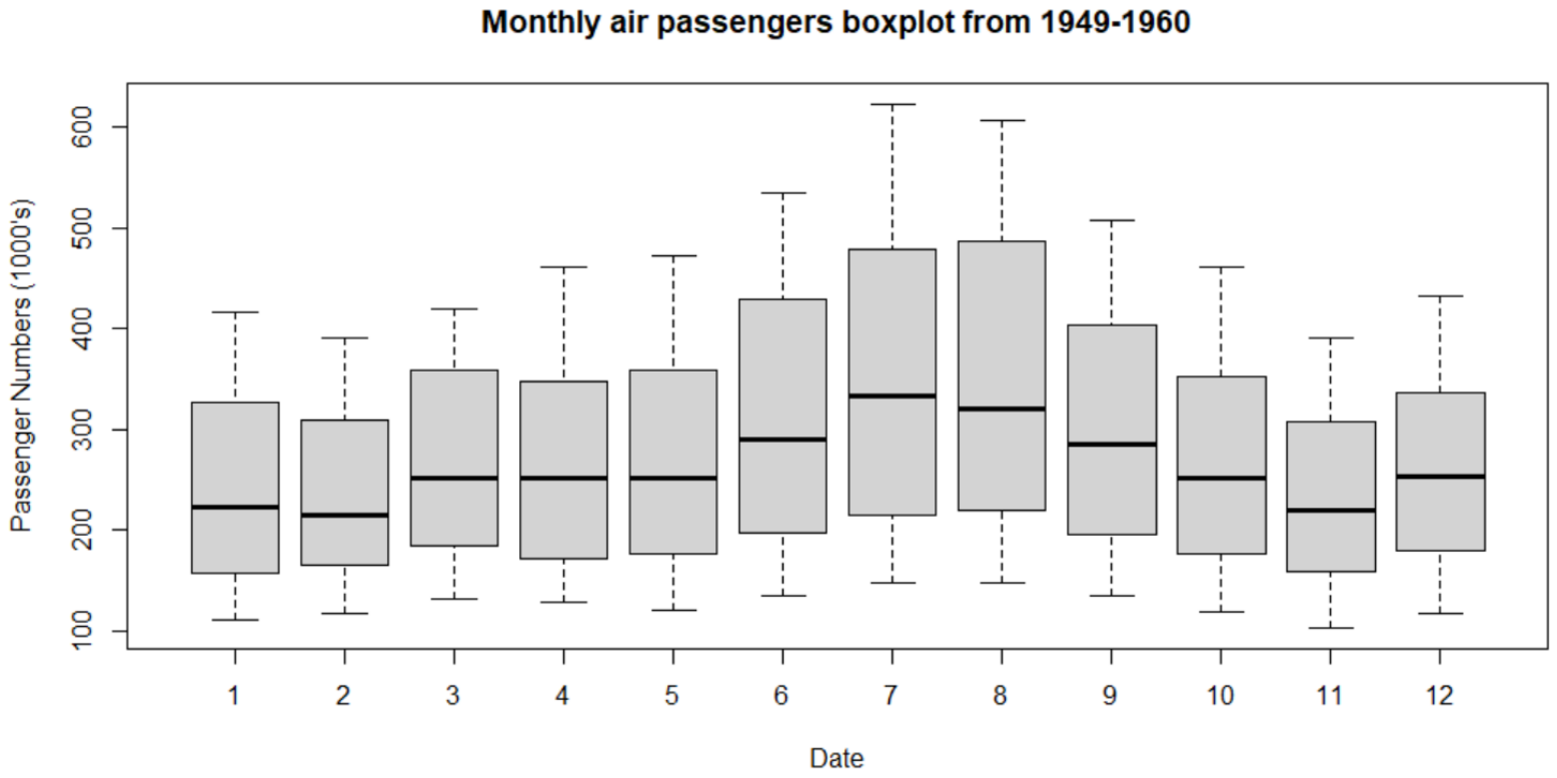
```
# Plot a trend line on the original dataset
plot(AirPassengers)
abline(reg=lm(AirPassengers~time(AirPassengers)))
```

To check the trend



Create a box plot by cycle

```
# Create a box plot by cycle
boxplot(AirPassengers~cycle(AirPassengers),
                        xlab="Date",
                        ylab = "Passenger Numbers (1000's)",
                        main = "Monthly air passengers boxplot from 1949-1960")
```



From the plot, you can see that the number of ticket sales goes higher in **June, July,** and **August** as compared to the other months of the years.

Build the ARIMA Model: combining into ARIMA

- An ARIMA model is simply the sum of the AR (Autoregression), differencing, and MA components (Moving Average).

- We abbreviate an ARIMA model as follows arima(p, d, q)(p, d, q)
  - p indicates the order of the autoregression
  - d indicates the number of times differencing takes place
  - q indicates the number of previous values we use for the moving average
- The first set of parenthesis indicates the non-seasonal (i.e. previous) values in the model.
- The second set of parenthesis indicates the seasonal values used in the model.

- The function Arima(data, order = (p, d, q), seasonal = (p, d, q))

Build the ARIMA Model: using auto.arima() function

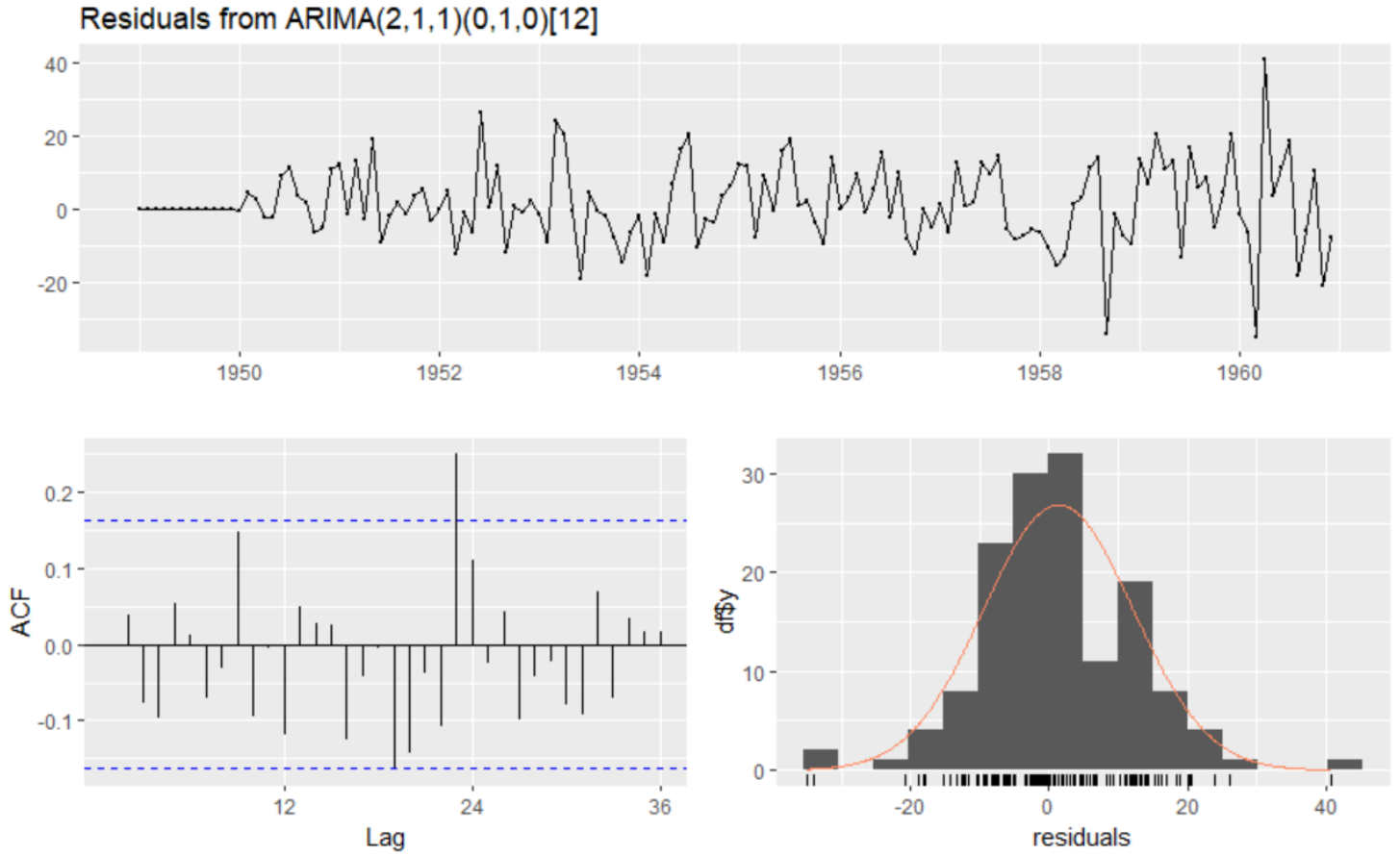
```
# Build the ARIMA Model: using auto.arima() function
mymodel <- auto.arima(AirPassengers)
mymodel
```

```
> mymodel
Series: AirPassengers
ARIMA(2,1,1)(0,1,0)[12]

Coefficients:
          ar1     ar2      ma1
      0.5960  0.2143  -0.9819
s.e.  0.0888  0.0880   0.0292

sigma^2 = 132.3:  log likelihood = -504.92
AIC=1017.85   AICc=1018.17   BIC=1029.35
```

The auto.arima() function has recommended that we use 2 past values in our regression, the past value for differencing, 1 error in the moving average, and the past value of the same season for seasonal differencing.

Build the ARIMA Model: check the residuals

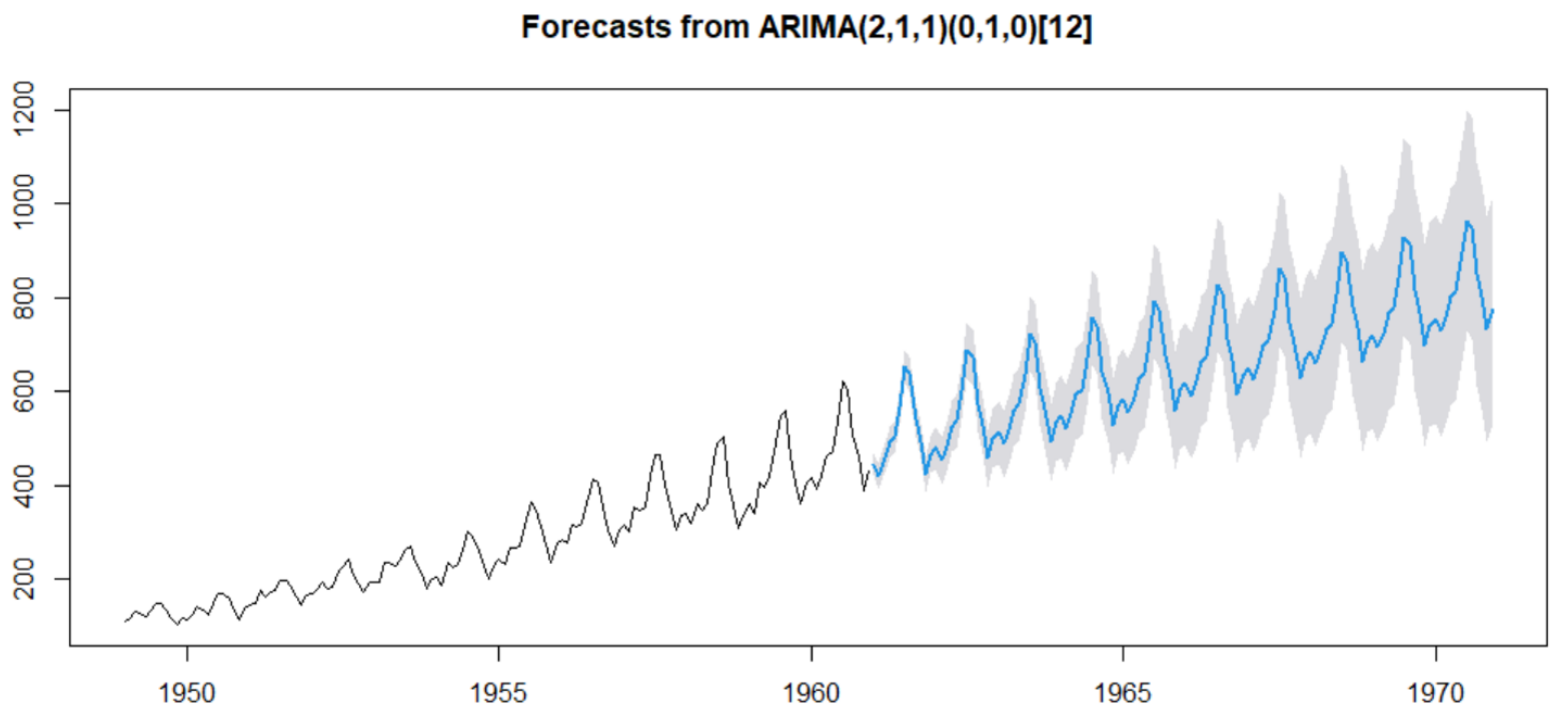
```
# Check that the residuals look like white noise
checkresiduals(mymodel)
```



Residuals are normally distributed.

Build the ARIMA Model: Forecast the values for the next 10 years

```
# Forecast the values for the next 10 years
myforecast <- forecast(mymodel, level=c(95), h=10*12)
plot(myforecast)
```



The shaded region covers all the values that can possibly occur in the future10 years and the blue color pattern is the average of all values in the shaded part. This is how we can forecast values using any time series dataset.

h: Number of periods for forecasting.

level: Confidence level for prediction intervals.

https://www.rdocumentation.org/packages/forecast/versions/8.4/topics/forecast

Build the ARIMA Model: Validate the model by selecting lag values

```
# Validate the model by selecting lag values
Box.test(mymodel$resid, lag=5, type="Ljung-Box")
Box.test(mymodel$resid, lag=10, type="Ljung-Box")
Box.test(mymodel$resid, lag=15, type="Ljung-Box")
```

```
> Box.test(mymodel$resid, lag=5, type="Ljung-Box")

        Box-Ljung test

data:  mymodel$resid
X-squared = 2.9244, df = 5, p-value = 0.7116

> Box.test(mymodel$resid, lag=10, type="Ljung-Box")

        Box-Ljung test

data:  mymodel$resid
X-squared = 8.6878, df = 10, p-value = 0.562

> Box.test(mymodel$resid, lag=15, type="Ljung-Box")

        Box-Ljung test

data:  mymodel$resid
X-squared = 11.582, df = 15, p-value = 0.7104
```

Looking at the lower p values, we can say that our model is relatively accurate, and we can conclude that from the ARIMA model, that the parameters (2, 1, 1) adequately fit the data.

5. In-class Assignment

You're going to be working with some retail data from “**Sales.RData**”. The data set contains information retail sales of automotive parts, accessory, and tire stores.

The dataset contains 2 columns:

DATE: monthly data

SALES: sales in Millions of USD,

Seasonally Adjusted

```
> head(sales)
      Jan  Feb  Mar  Apr  May  Jun
1992 3311 3360 3445 3415 3510 3521
```

```
> sales
      Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
1992 3311 3360 3445 3415 3510 3521 3470 3461 3521 3576 3592 3579
1993 3600 3636 3525 3672 3739 3721 3811 3815 3861 3837 3810 3761
1994 3984 4037 4040 4100 4017 4066 4138 4179 4127 4185 4189 4223
1995 4233 4153 4199 4274 4276 4340 4312 4397 4405 4404 4520 4476
1996 4484 4557 4539 4569 4596 4626 4645 4573 4617 4732 4704 4766
1997 4787 4807 4860 4856 4785 4890 4909 4952 4979 4915 4996 4818
1998 4943 4959 5013 4946 5040 4958 4994 4981 5015 5073 5015 5073
1999 5117 5144 5087 5185 5165 5165 5195 5150 5358 5344 5434 5350
2000 5373 5284 5406 5220 5289 5293 5210 5236 5490 5191 5166 5317
2001 4956 5020 5036 5130 5117 5144 5151 5167 5072 5103 5088 5111
2002 5180 5189 5110 5226 5204 5250 5284 5280 5272 5234 5213 5159
2003 5219 5199 5265 5250 5307 5341 5369 5381 5500 5476 5499 5422
2004 5439 5507 5577 5481 5541 5476 5530 5526 5514 5610 5603 5743
2005 5935 5824 5728 5853 5754 5828 5784 5833 5769 5799 6022 6024
2006 6210 6031 6013 5989 5901 5962 6014 6030 6103 6074 6001 6118
2007 6105 6047 6248 6131 6268 6194 6229 6261 6295 6411 6283 6219
2008 6228 6230 6237 6287 6374 6422 6463 6298 6349 6346 6442 6395
2009 6411 6366 6247 6277 6262 6204 6143 6102 6051 6098 6016 6089
2010 6097 6218 6314 6491 6419 6452 6475 6648 6641 6657 6720 6771
2011 6609 6664 6806 6757 6828 7029 6947 7120 7134 7025 7057 7022
2012 7193 7067 6980 7004 6959 6935 7044 6905 7047 7051 6965 6978
2013 7001 7104 7005 6983 7088 7037 7084 7162 7132 7282 7332 7374
2014 7227 7319 7314 7309 7200 7108 7119 7189 7176 7184 7595 7156
2015 7292 7389 7450 7559 7593 7685 7725 7627 7581 7597 7727 7652
2016 7771 7748 7665 7554 7624 7726 7599 7696 7635 7705 7609 8175
2017 7773 7678 7760 7814 7767 7721 7805 7705 7891 7753 7884 8120
2018 7783 7823 7800 7814 7938 7923 8006 7925 8046 8008 7961 7922
2019 7721 7994 8113 8090 8108 8110 8089 8189 8066 8096 8132 8054
2020 7971 7866 7128 6743 8391 8950 8671 8713 8589 8251 8240 8183
2021 8446 8358 9467 9487 9299 9283 9349 9460 9413 9642 9669 9974
2022 9975 9943 10066 10212 10417 10353 10423 10487 10662 10543 10608 10655
2023 10648 10837
```