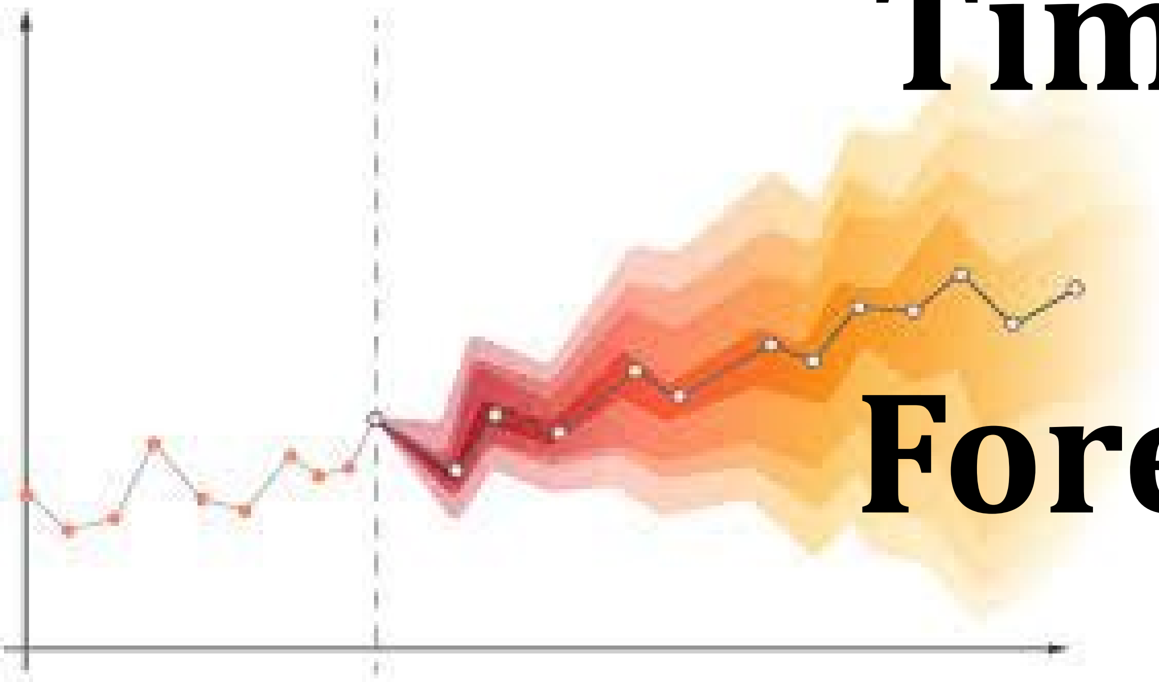
Engineering Statistics



Timeseries and Forecasting

Dr. Vvn Weian Chao (趙韋安)
https://ce.nctu.edu.tw/member/teachers/23
Department of Civil Engineering, National Yang Ming Chiao Tung University, Taiwan

Purpose

讀入時序資料
建立預測模型

R: Create Time Series

```
ts(x,
start=,
end=,
frequency=).
```

R: Create Time Series

The **ts()** function will convert a numeric vector into an R time series object. The format is **ts(***vector*, **start=, end=, frequency=)** where start and end are the times of the first and last observation and frequency is the number of observations per unit time (1=annual, 4=quartly, 12=monthly, etc.).

```
# save a numeric vector containing 72 monthly observations
# from Jan 2009 to Dec 2014 as a time series object
myts <- ts(myvector, start=c(2009, 1), end=c(2014, 12), frequency=12)
```

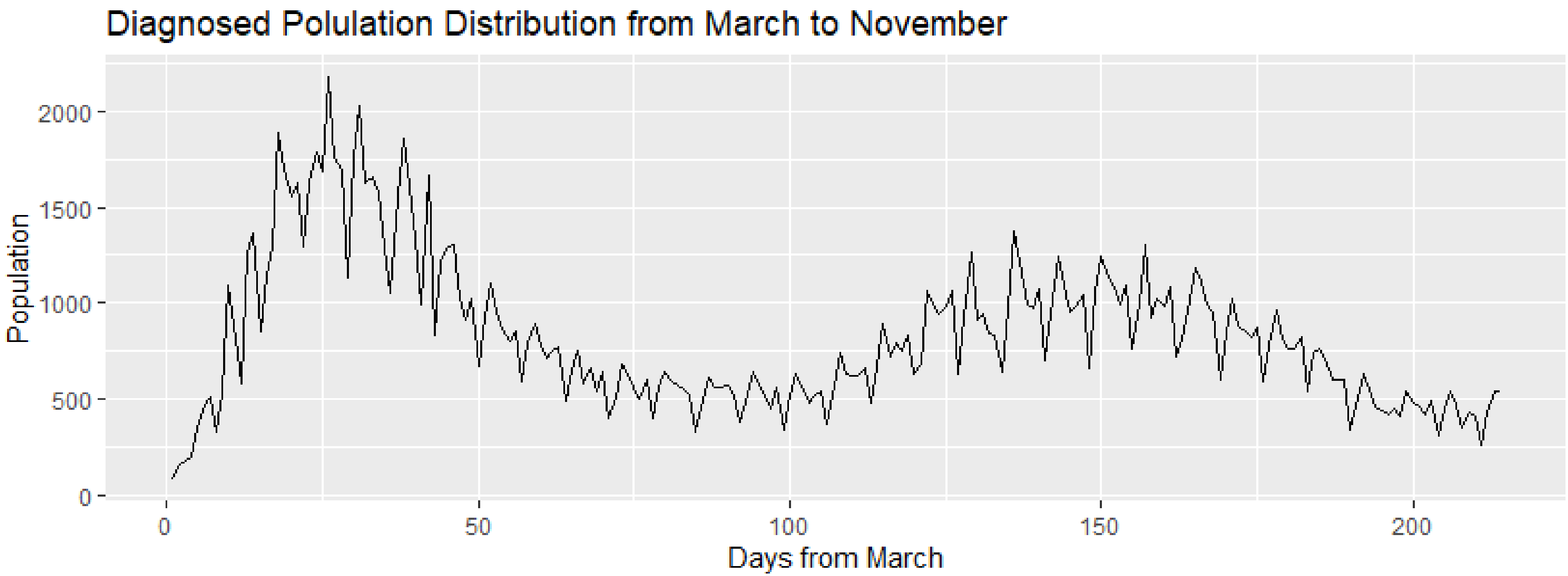
```
# subset the time series (June 2014 to December 2014)
myts2 <- window(myts, start=c(2014, 6), end=c(2014, 12))
```

R: autoplot function

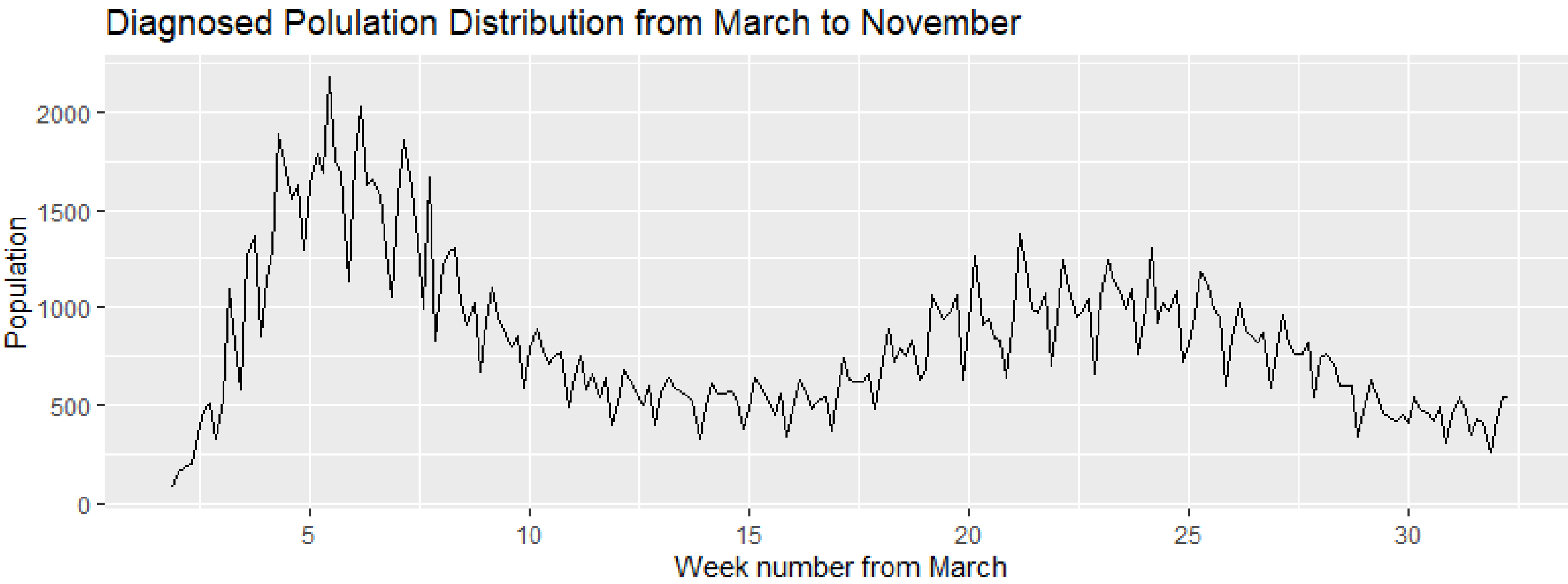
The **autoplot()** function in ggplot2 is a convenient way to generate various types of plots based on the type of data object passed to it.

```
library(ggplot2)
library(ggfortify)
autoplot(
)
```

R: autoplot function



R: autoplot function frequency=7



R: Decomposition

As a data analyst, time-series data is an important area for us to study, especially if you do a lot of web analytics. To be able to analyze time series effectively, it helps to understand the interaction between general seasonality in activity and the underlying trend. Time series decomposition works by splitting a time series into three components: **Seasonality, Trends and Random Fluctuation**. This article we will use one of the functions in R language, called **"Decompose()",** to transforms a time series into multiple different time series.

Seasonal: Patterns that repeat with a fixed period of time. For example, a website might receive more visits during weekends; this would produce data with a seasonality of 7 days.

Trend: How things are overall changing. A website increasing in popularity should show a general trend that goes up.

Random/error/residual: Also call “noise”, this is the residuals of the original time series after the seasonal and trend series are removed.

R: Decomposition

Choose Right Model: Additive or Multiplicative Decomposition?

To achieve successful decomposition result, it is important to understand what is the differences between the additive and multiplicative models. For example, does the magnitude of the seasonality increase when the time series increases?

Additive Model: the components add together to make the time series. If you have an increasing trend, you still see roughly the same size peaks throughout the time series.

$$Time\ Series = Seasonality + Trends + Random$$

Multiplicative Model: the components multiply together to make the time series. If you have an increasing trend, the amplitude of seasonal activity increases a lot.

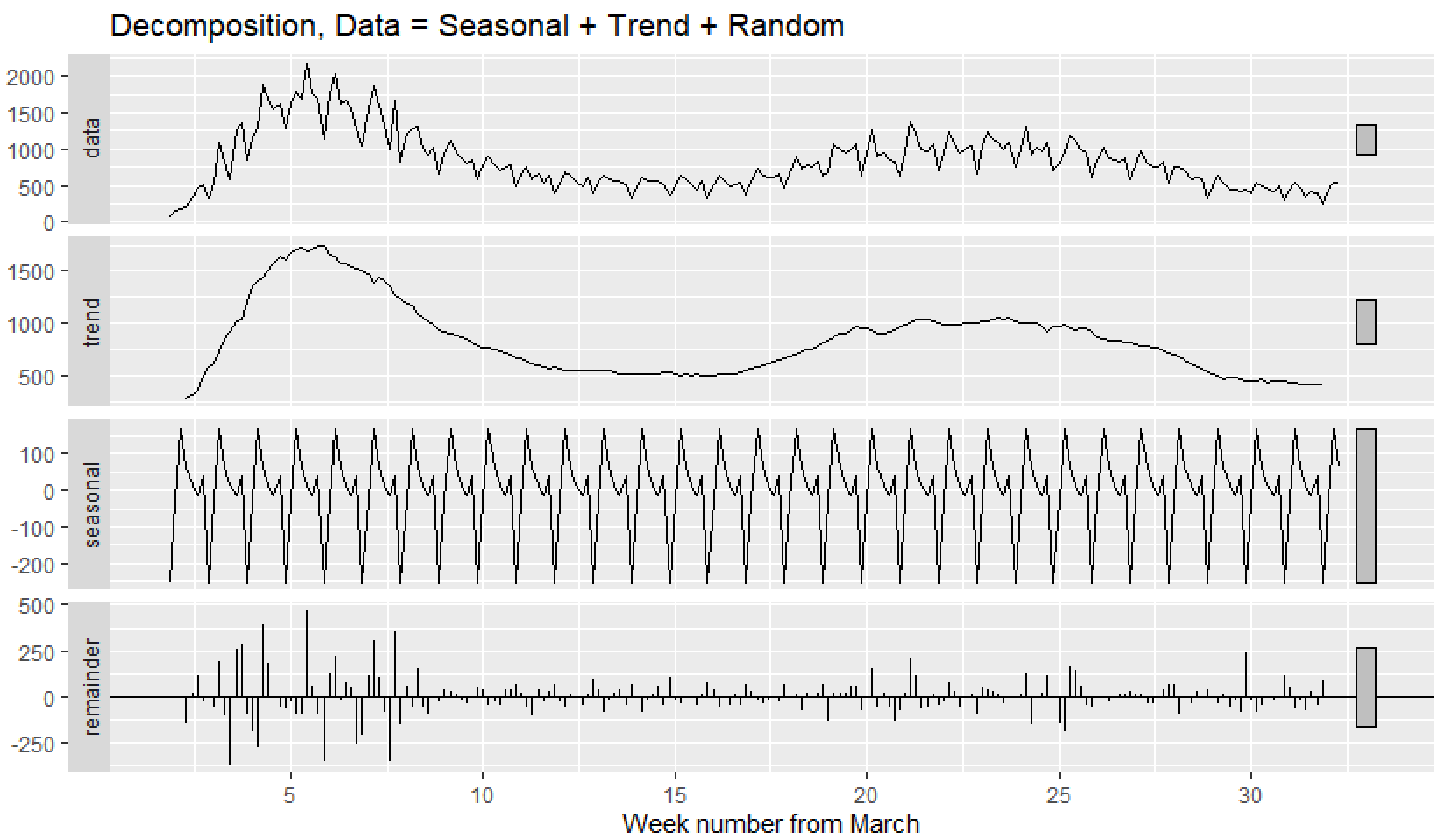
$$Time\ Series = Seasonality \times Trends \times Random$$

R: Decomposition

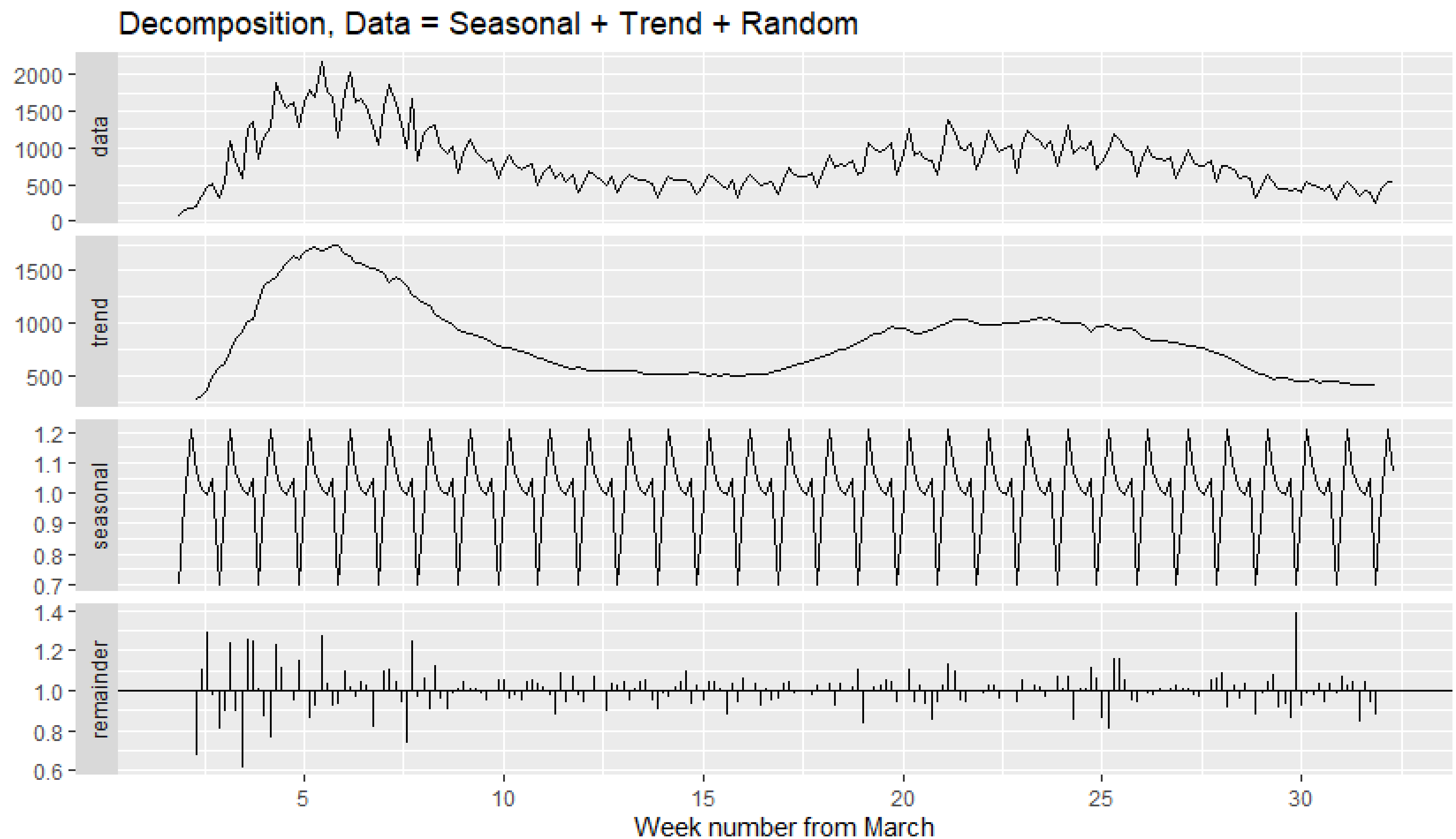
decompose(x, type=)

type= “additive” or “multiplicative”

R: Decomposition additive



R: Decomposition multiplicative



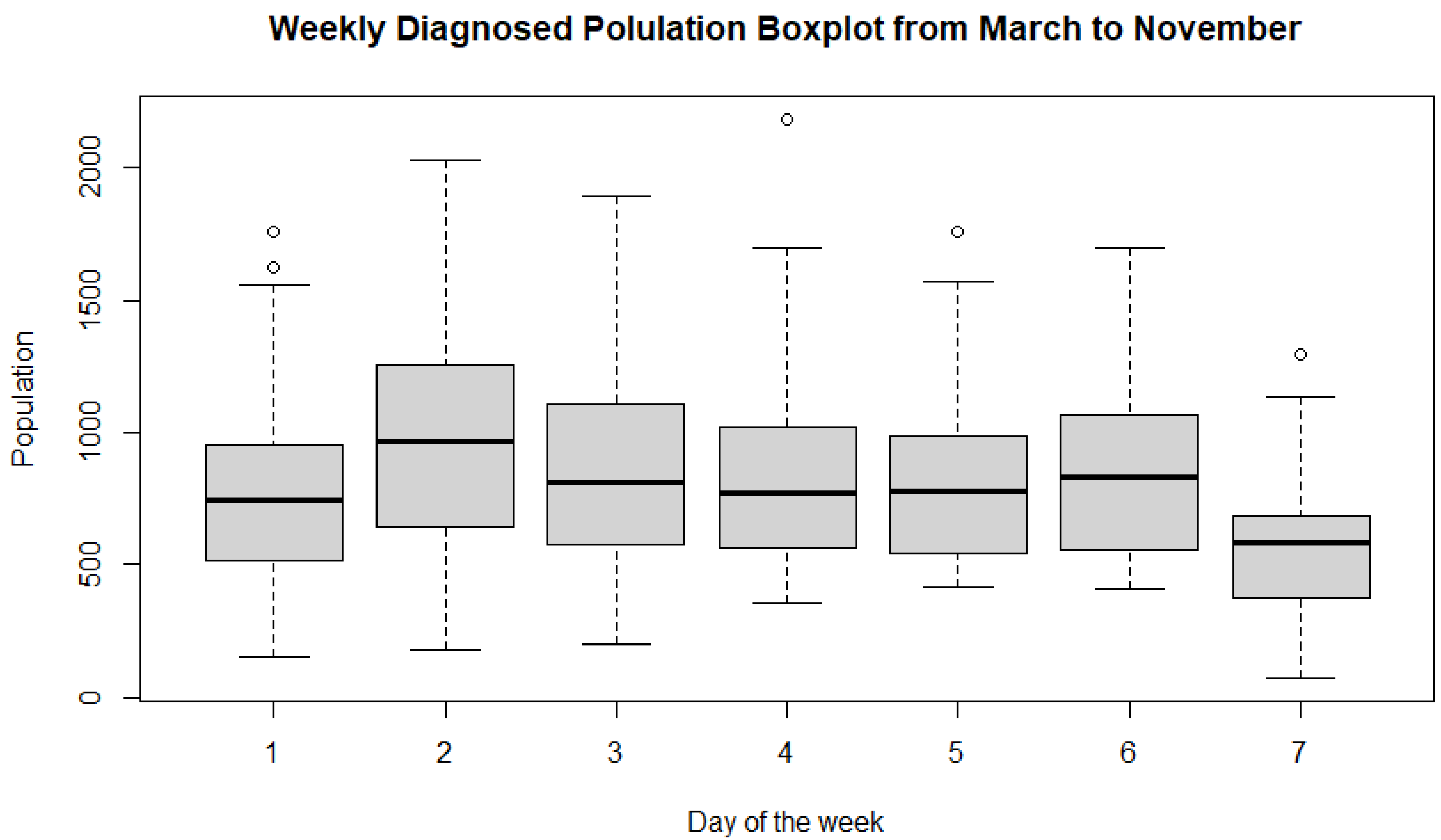
R: cycle

cycle(x)

gives the positions in the cycle of each observation

```
> cycle(ts.data)
Time Series:
Start = c(1, 7)
End = c(32, 3)
Frequency = 7
[1] 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2
[25] 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5
[49] 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1
[73] 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4
[97] 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7
[121] 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3
[145] 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6
[169] 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2
[193] 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3
```

R: boxplot


Weekly Diagnosed Polulation Boxplot from March to November
Population
0
500
1000
1500
2000
1
2
3
4
5
6
7
Day of the week

R: Auto-Regressive Integrated Moving Average (ARIMA)

library(forecast)

Auto.arima(
x)

R: Auto-Regressive Integrated Moving Average (ARIMA)

https://people.duke.edu/~rnau/411arim.htm

ARIMA (Auto-Regressive Integrated Moving Average) is a popular time series forecasting model. It is widely used in various fields to analyze and predict future values based on past observations in a time series dataset.

The ARIMA model combines three key components: ***Auto-regression (AR), Differencing (I), and Moving Average (MA).***

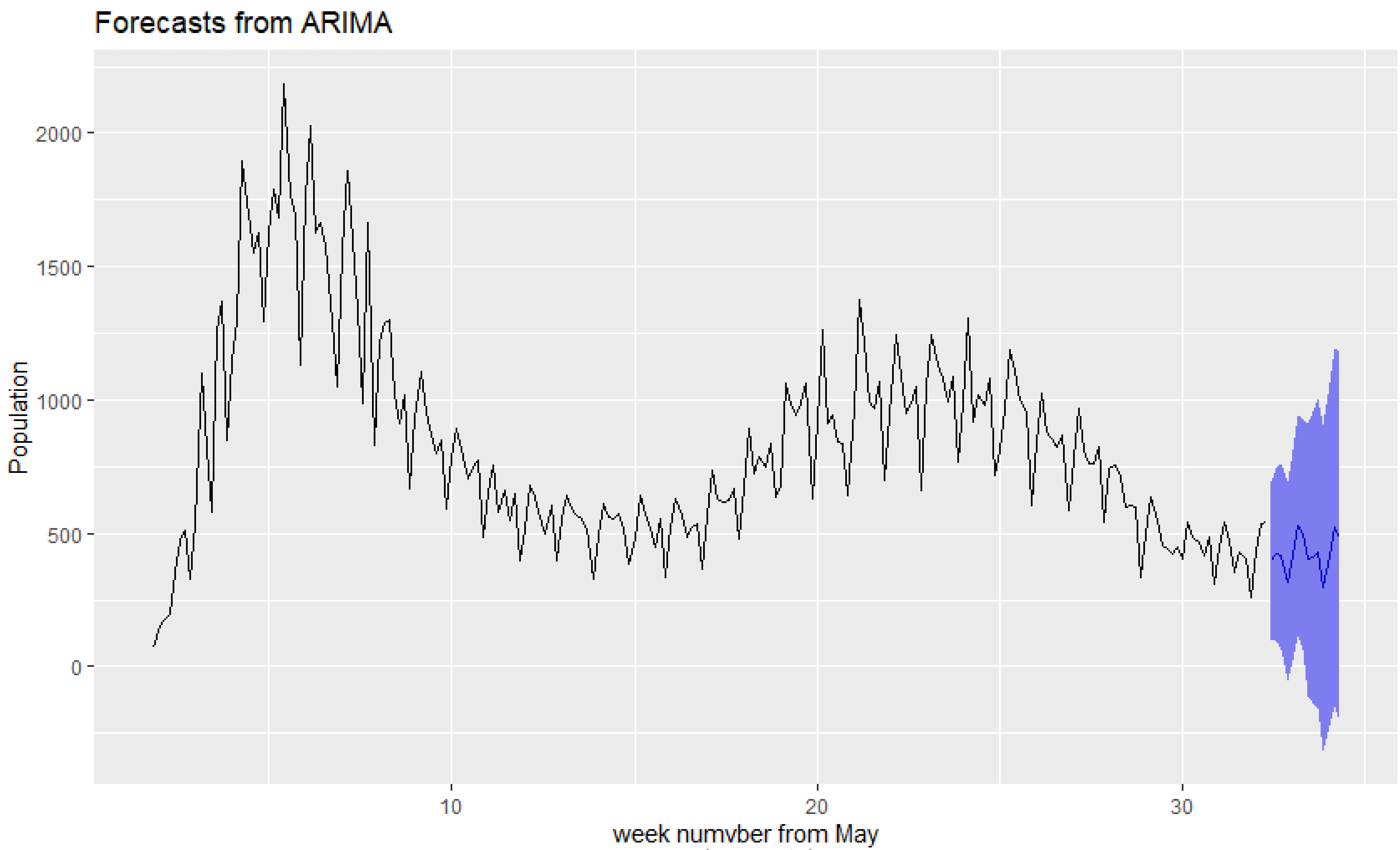
AR — Autoregressive (p): It refers to the dependence of the current value in a time series on previous values. The order of the AR component, denoted by **p**, represents the number of lagged observations used in the model.

I — Order of Integration (d): It is the number of differences we consider between the current value and past value in order to make the time series stationary. Stationarity means that the statistical properties of the series, such as mean and variance, remain constant over time.

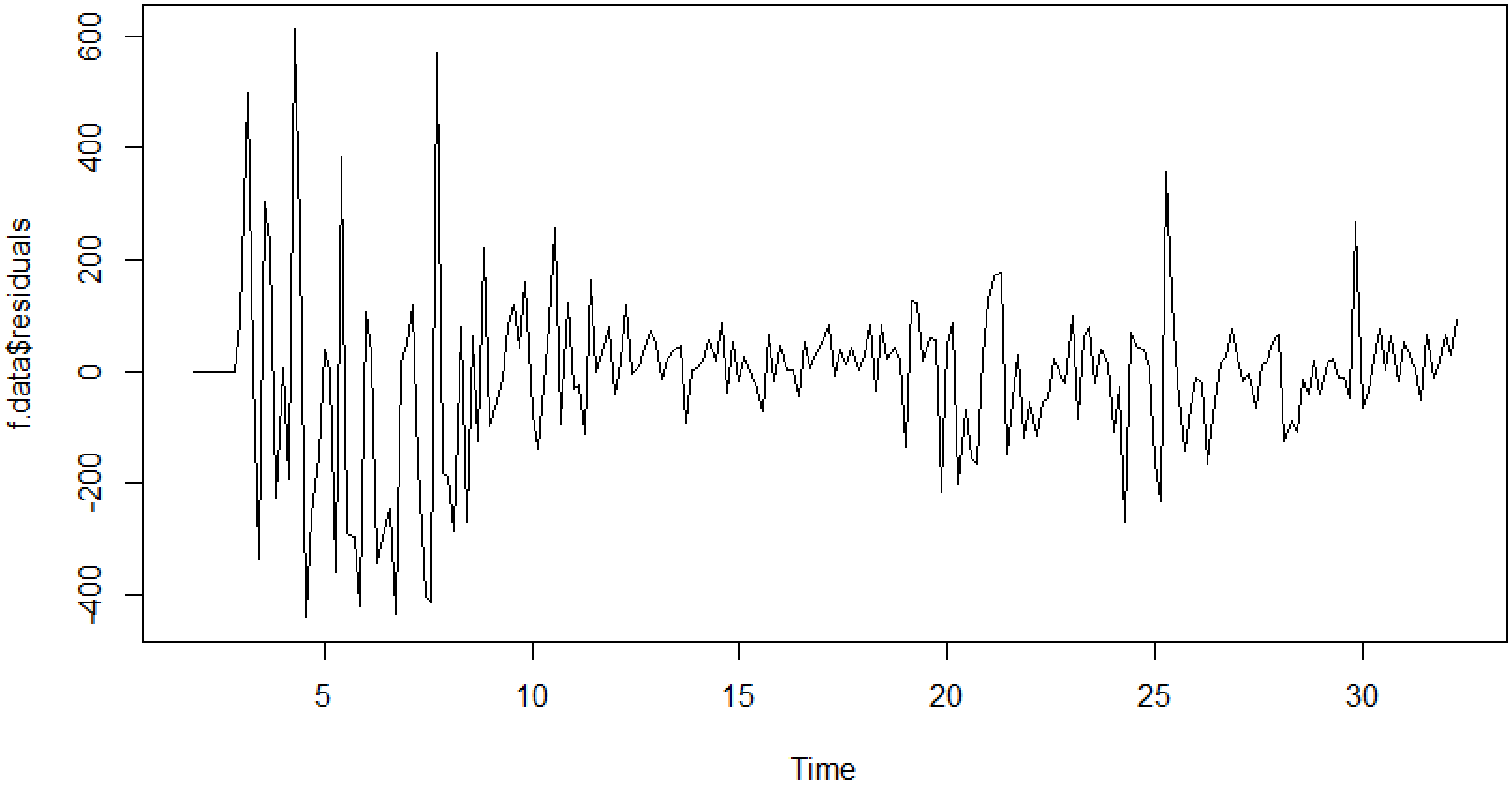
MA — Moving Average (q): It represents the dependency between the current value and the residual errors from past predictions. It calculates the weighted average of the previous forecast errors. The order of the MA component, denoted by **q**, represents the number of lagged forecast errors used in the model.

ARIMA model represented as **ARIMA (p, d, q)**.

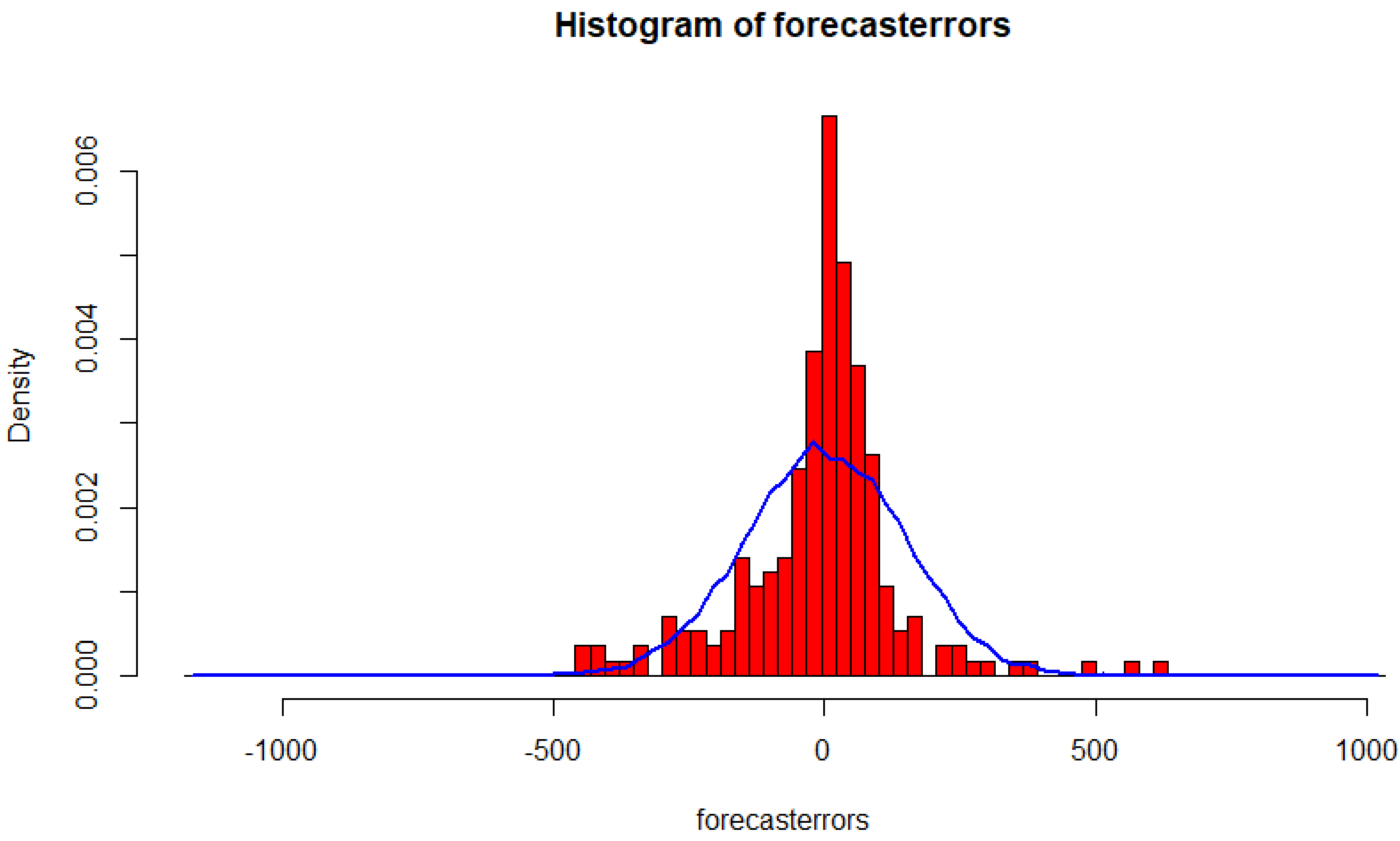
R: Auto-Regressive Integrated Moving Average (ARIMA)



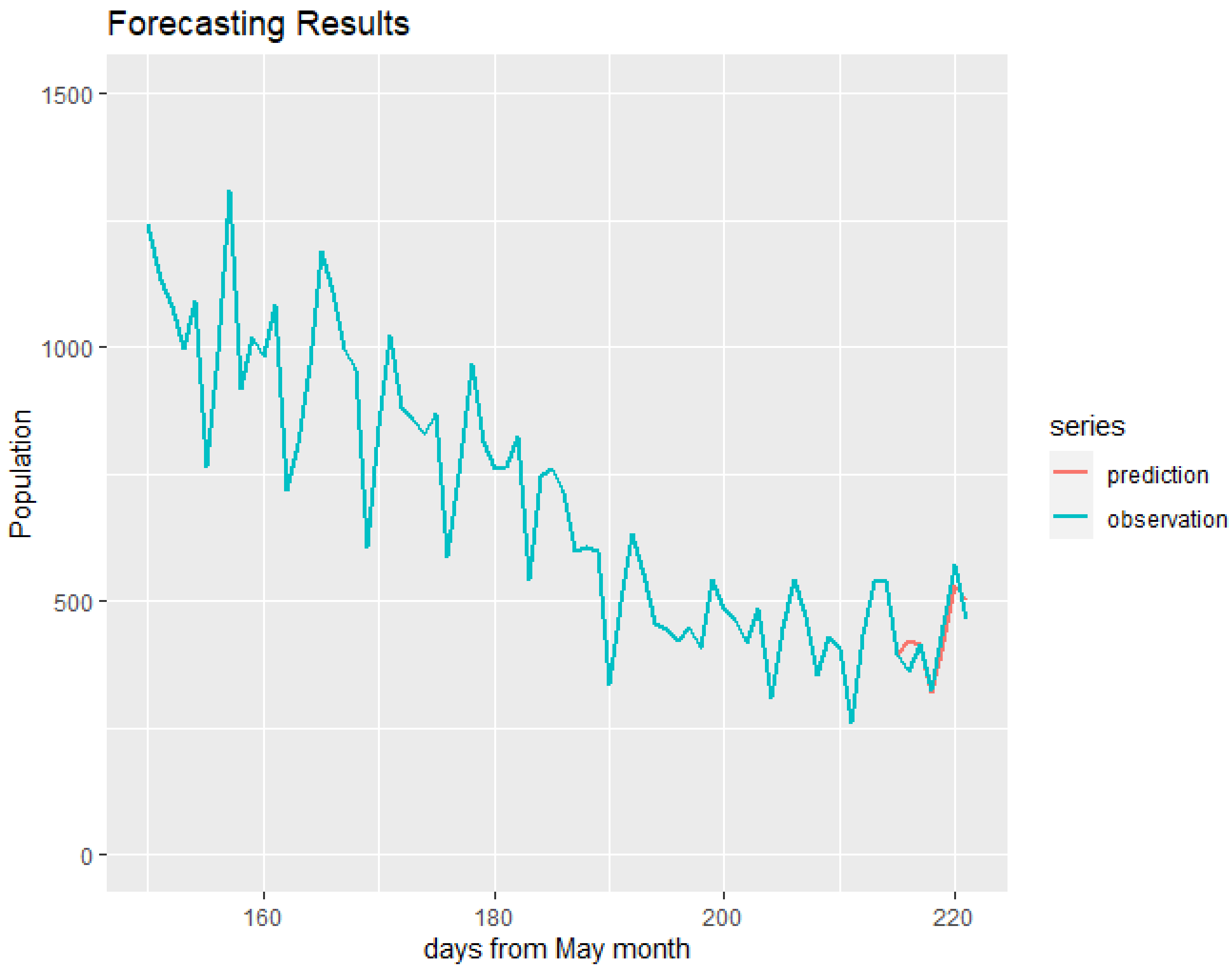
R: ARIMA, residuals check



R: ARIMA, residuals check



R: Comparison prediction with observations



TRY
it
in
R

R: Hypothesis Testing

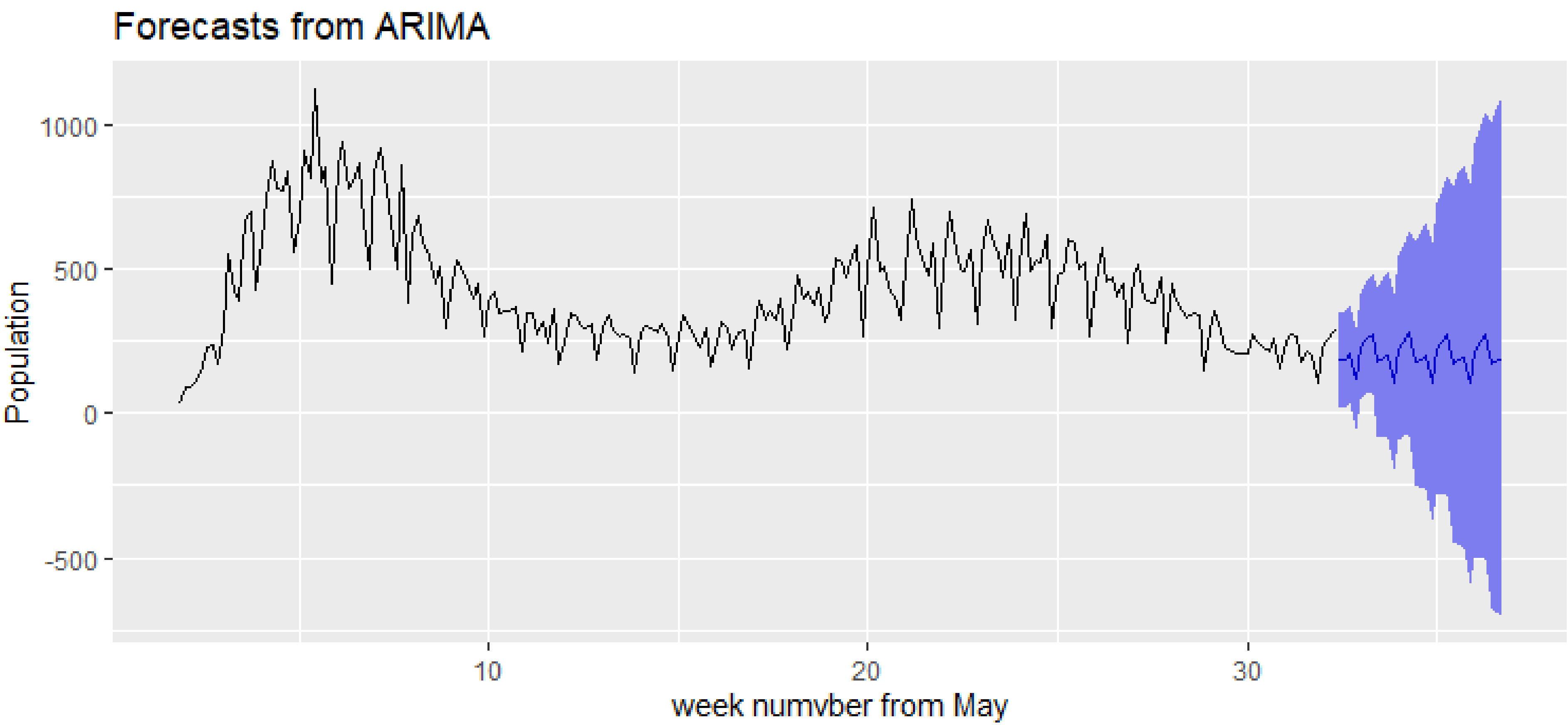
R_TimeSeries_and_Forecasting_a.R

課堂練習: 學號-姓名-ch12-forecasting.R

請參考以下程式:
R_TimeSeries_and_Forecasting_a.R

利用2022年5月1日至11月30日新竹市東區時序確診人數數據，進行預測12月1日至12月31日的確診人數，並將成果以作圖展示(Confidence level 95%)

課堂練習: 學號-姓名-ch12-forecasting.R



課堂練習: 學號-姓名-ch12-forecasting.R