Introduction to Time Series and Forecasting Book Complementary

Adhitya Arif Wibowo (AWibowo6@gatech.edu)

21 April 2022

Contents

Preface

This document is created based on *Introduction to Time Series and Forecasting, 3rd Edition* by *Peter J. Brockwell* and *Richard A. Davis* as one of the textbook suggested to be used in *ISYE 6402 - Time Series Analysis* course taught in Georgia Tech.

The scope of the document only discuss materials in the book that are also discussed in the lecturer.

How To Use The Document

The course related documents reference will be the documents used in Spring 2022 semester for the course. As for the book reference will be based on the 3rd edition of the book mentioned in Preface section above.

One should have both the book and the course related documents opened to refer to section, example, figures, module, lesson, slide etc., that are discussed in this document.

Most of the codes are implementation of equation or algorithm in the book, try to understand the code to support the understanding of the equation or vice versa and come up with your own code to achieve the same result.

Prerequisites

1. Basic probability and statistics, refer to *Probability, Statistics, and Stochastic Processes* by *Peter Olofsson* [https://g.co/kgs/tHPF95].
2. Basic programming in R or any language is expected to follow along with the code in this document, refer to *R for Everyone: Advanced Analytics and Graphics* by *Jared P. Lander* [https://g.co/kgs/qjgw2B].

R

The programming language that is used in the course is R, this section will deal with the most common setup throughout the document.

Common Libraries

Here are common libraries that will be used throughout the document:

```
library( dplyr )
library( ggplot2 )
library( ggthemes )
library( lubridate )
```

Book Related Library

Below is the library related with the book *Introduction to Time Series and Forecasting, 3rd Edition* by *Peter J. Brockwell* and *Richard A. Davis*:

```
install.packages("itsmr")
```

Helper Function

get_data_frame

Helper function to get data set as data frame:

```
get_data_frame <-
  function (
    v, start_date,
    period = "monthly",
    date_format = "%d-%b-%Y",
    multiplier = 1
  ) {
    data_t <- 1:length( v )
    t_to_date <- list(
        monthly = function( t, start_date, multiplier ) {
            return(
              as_date( start_date, format = date_format ) %m+%
                months( ( t - 1 ) * multiplier )
            )
        },
        yearly = function( t, start_date, multiplier ) {
            return(
              as_date( start_date, format = date_format ) %m+%
                months( ( t - 1 ) * 12 * multiplier )
            )
        }
    )
    return( data.frame(
        t = data_t,
        date = t_to_date[[ period ]]( data_t, start_date, multiplier ),
        value = v
    ) )
  }
```

plot_data

To get same look feel of graph:

```
plot_data <-
  function (
    data, title,
    x_label = NULL, y_label = NULL,
    line = TRUE, point = TRUE,
    x_ticks = 6, y_ticks = 7
  ) {
    data_plot <-
        ggplot( data, aes( date, value ) ) +
        ggtitle( title ) +
        xlab( x_label ) +
        ylab( y_label ) +
        scale_x_date(breaks = scales::pretty_breaks(n = x_ticks)) +
        scale_y_continuous(breaks = scales::pretty_breaks(n = y_ticks))
    if ( line ) {
        data_plot <- data_plot + geom_line( size = 0.3, color = "blue" )
    }
    if ( point ) {
        data_plot <- data_plot + geom_point( size = 1, shape = 0 )
    }
    data_plot <-
        data_plot +
            theme_tufte() +
            theme(
              text = element_text( family = "sans", size = 10 ),
              plot.title = element_text( size = 10 ),
              axis.title.x = element_text( size = 10 ),
              axis.title.y = element_text( size = 10 )
            )
    return( data_plot )
  }
```

Reference By Course Lecture

This section contains links that are connecting the book's complementary in reference [By Book] section to the course lecture videos slide.

Module 1 - Lesson 2 - Slide 14

Normality Transformation, variance stabilizing transformation example

1, Figure 1-17

Module 1 - Lesson 3 - Slide 4

Examples of time series.

1. Example 1.1.1
2. Figure 1-1
3. Example 1.1.3
4. Figure 1-3
5. Example 1.1.5
6. Figure 1-5

Module 1 - Lesson 4 - Slide 5

Estimating trend with moving average.

1. 1.5.1 Estimation and Elimination of Trend in the Absence of Seasonality
2. Smoothing with a finite moving average filter
3. Figure 1-18
4. Figure 1-19

Module 1 - Lesson 4 - Slide 6

Trend: Parametric Regression.

1. Example 1.3.4
2. Example 1.3.5
3. Example 1.5.4
4. Figure 1-8
5. Figure 1-9
6. Figure 1-10

Module 1 - Lesson 6 - Slide 5

Seasonality: Averaging

1. 1.5.2.1 Method S1: Estimation of Trend and Seasonal Components
2. Figure 1-24
3. Figure 1-25

Module 1 - Lesson 6 - Slide 7

Seasonality: Cosine-Sine Model

1. Example 1.3.6
2. Figure 1-11

Module 1 - Lesson 6 - Slide 8

Time Series: Trend and Seasonality

1. 1.5.2.1 Method S1: Estimation of Trend and Seasonal Components
2. Figure 1-24
3. Figure 1-25
4. Example 1.5.4

Module 1 - Lesson 6 - Slide 10

Differencing to Remove Trend and Seasonality

1. 1.5.1.1 Method 2: Trend Elimination by Differencing
2. Example 1.5.3
3. Figure 1-23
4. 1.5.2.2 Method S2: Elimination of Trend and Seasonal Components by Differencing
5. Figure 1-26
6. Figure 1-27

Module 1 - Lesson 8 - Slide 4, 5, 6, and 8

Examples of stationary time series, auto-covariance, and auto-correlation.

1. Example 1.4.1
2. Example 1.4.2
3. Example 1.4.4
4. Example 1.4.5

Module 1 - Lesson 8 - Slide 9

Sample auto-covariance and auto-correlation.

1. Definition 1.4.4
2. Sample Auto-Covariance Function
3. Sample Auto-Correlation Function
4. Figure 1-14
5. 1.4.2 A Model for the Lake Huron Data
6. Figure 1-10
7. AR(1) Model
8. Figure 1-16
9. AR(2) Model

Module 1 - Lesson 10 - Slide 6, 7, and 8

Prediction of stationary time series and best linear predictors.

1. Forecasting Stationary Time Series
2. 2.5.2 The Prediction Operator P(.|W)
3. Example 2.5.3

Module 1 - Lesson 10 - Slide 9

Durbin-Levinson Algorithm

1. Example 2.5.3
2. Durbin-Levinson Algorithm
3. Example 2.5.5

Module 1 - Lesson 10 - Slide 10

Innovations Algorithm

1. Example 2.5.3
2. Innovations Algorithm
3. Example 2.5.5

Module 2 - Lesson 1 - Slide 5 and 6

ARMA Model: Definition

1. Definition 3.1.1

Module 2 - Lesson 3 - Slide 4

ARMA Model: Stationarity

1. Existence and Uniqueness

Module 2 - Lesson 3 - Slide 5 and 9

Causal and Invertible ARMA Process

1. Example 3.1.1
2. Example 3.1.2
3. Example 3.1.3
4. Example 3.2.1

Module 2 - Lesson 4 - Slide 4, 5, and 6

ARMA Model: Auto-Covariance Function

1. 3.2.1 Calculation of the ACVF
2. Example 3.2.1
3. Example 3.2.2

Module 2 - Lesson 4 - Slide 7, 8, and 9

Partial Auto-Correlation Function

1. Example 3.2.6
2. Figure 3-7

Module 2 - Lesson 4 - Slide 10 and 12

ACF and MA(q) process

1. Example 3.2.2

Module 2 - Lesson 4 - Slide 11 and 12

PACF and AR(p) process

1. Example 3.2.6

Module 2 - Lesson 6 - Slide 5, 6, 7, and 8

Yule-Walker Equations, Estimates, and Properties

1. 5.1.1 Yule-Walker Estimation
2. Large-Sample Distribution of Yule-Walker Estimators
3. Example 5.1.1
4. Figure 5-1
5. Figure 5-2

Module 2 - Lesson 6 - Slide 9, 10 and 11

Innovation Algorithm for MA(q) and ARMA(p, q) process.

1. Example 5.1.5
2. Innovations Algorithm Estimates when p > 0 and q > 0
3. Example 5.1.6

Module 3 - Lesson 1 - Slide 7

Model Structure

1. 7.1 Historical Overview

   In the lecture slide the notion σ_t is equivalent to $\sqrt{h_t}$ in the book or $\sigma_t^2 = h_t$ where both denote **volatility** and R_t in the lecture slide is equivalent to e_t in the book where both are IID (0, 1) that is sequence of random variables with mean = 0 and variance = 1.

Module 3 - Lesson 2 - Slide 3

Simulation: Time varying Conditional Variance

1. 7.1 Historical Overview

   Here in the lecture slide we have the notion ε_t which is equivalent to Z_t in the book and w_t in lecture slide is equivalent to e_t in the book.

   The lecture slide is trying to simulate a GARCH(1, 1) euqation:

   $$\sigma_t^2 = 0.2 + 0.5\varepsilon_{t-1} + 0.3\sigma_{t-1}^2 \tag{7.1a}$$

   Where in the book it will be substituted to the GARCH equation:

$$
\begin{aligned}
h_t &= \alpha_0 + \sum_{i-1}^{p} \alpha_i Z_{t-i}^2 + \sum_{i-1}^{q} \beta_i h_{t-i} \\
h_t &= \alpha_0 + \sum_{i-1}^{1} \alpha_i Z_{t-i}^2 + \sum_{i-1}^{1} \beta_i h_{t-i},\ for\ p = 1, q = 1 \\
h_t &= \alpha_0 + \alpha_1 Z_{t-1}^2 + \beta_1 h_{t-1},\ with\ \alpha_0 = 0.2,\ \alpha_1 = 0.5,\ and \beta_1 = 0.3 \\
h_t &= 0.2 + 0.5 Z_{t-1}^2 + 0.3 h_{t-1}
\end{aligned}
\tag{7.1b}
$$

Where we reach the equivalence between equation 7.1a and 7.1b

Module 3 - Lesson 3 - Slide 8

Stationarity of ARCH(1)

1. ARCH(1) Process

Module 3 - Lesson 5 - Slide 7

Stationarity of GARCH(1,1)

1. GARCH(1, 1) Process

Reference By Book

This section contains the book's complementary that will be referenced by Reference By Course Lecture section.

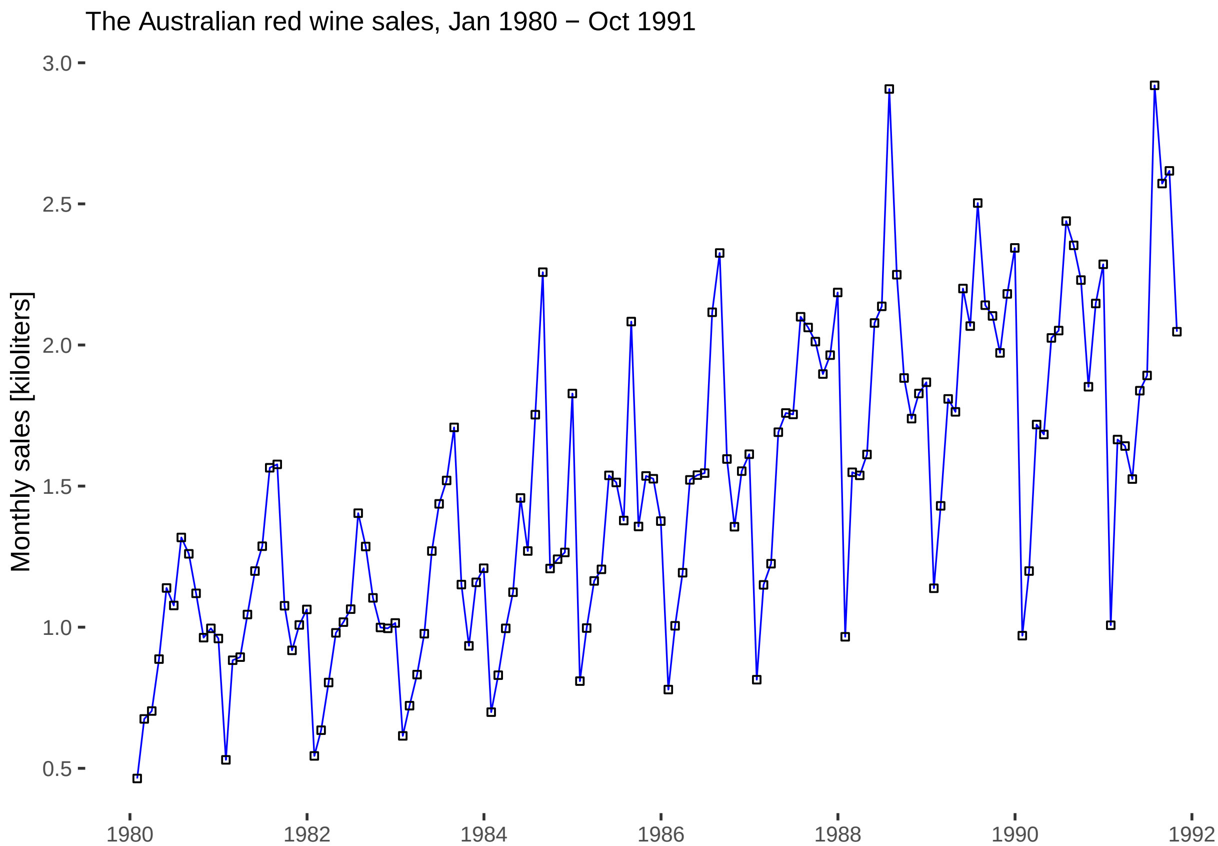
Example 1.1.1

```
wine <- get_data_frame( itsmr::wine, "31-Jan-1980" )
head( wine )
```

t	date	value
1	1980-01-31	464
2	1980-02-29	675
3	1980-03-31	703
4	1980-04-30	887
5	1980-05-31	1139
6	1980-06-30	1077

Figure 1-1

```
plot_data(
  # in order to get the kiloliters value
  data = wine %>% mutate( value = value / 1000 ),
  title = "The Australian red wine sales, Jan 1980 - Oct 1991",
  y_label = "Monthly sales [kiloliters]"
)
```



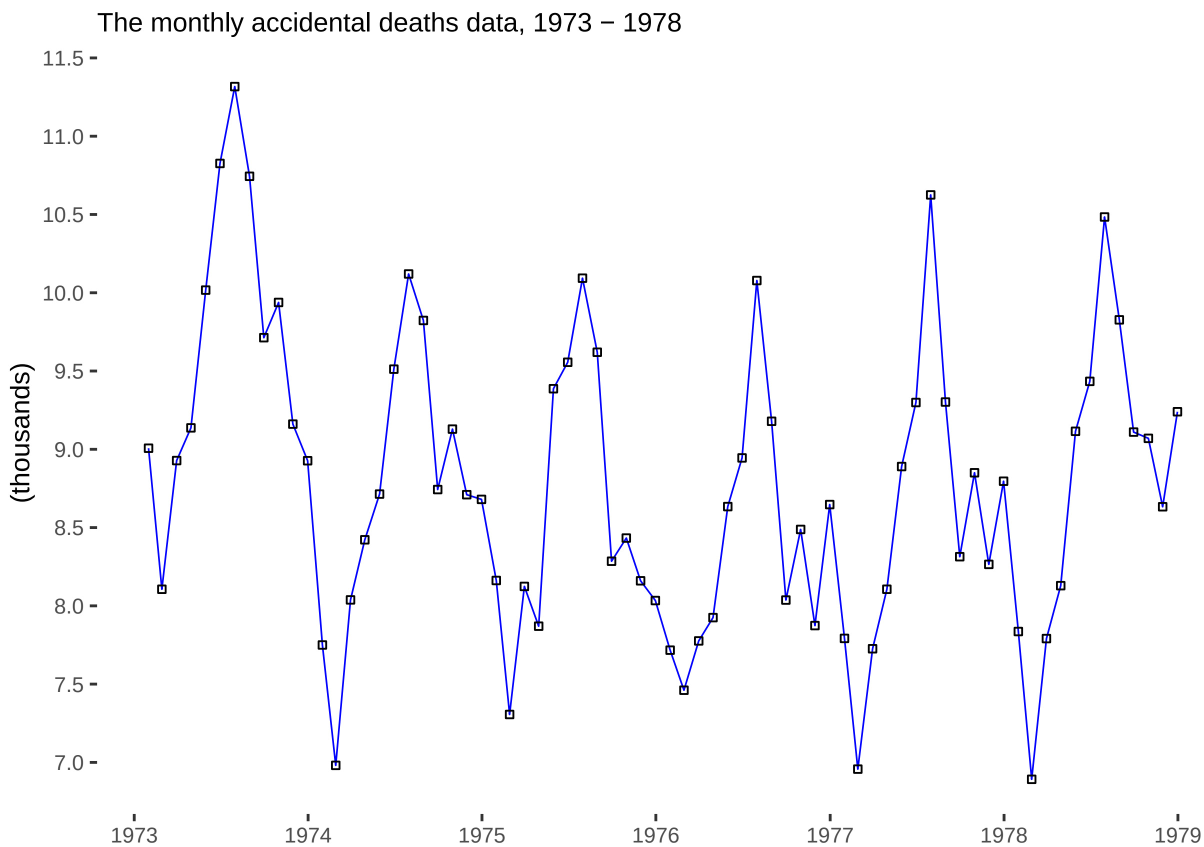
Example 1.1.3

```
deaths <- get_data_frame( itsmr::deaths, "31-Jan-1973" )
head( deaths )
```

t	date	value
1	1973-01-31	9007
2	1973-02-28	8106
3	1973-03-31	8928
4	1973-04-30	9137
5	1973-05-31	10017
6	1973-06-30	10826

Figure 1-3

```r
plot_data(
  # in order to get the value in thousands
  data = deaths %>% mutate( value = value / 1000 ),
  title = "The monthly accidental deaths data, 1973 - 1978",
  y_label = "(thousands)"
)
```



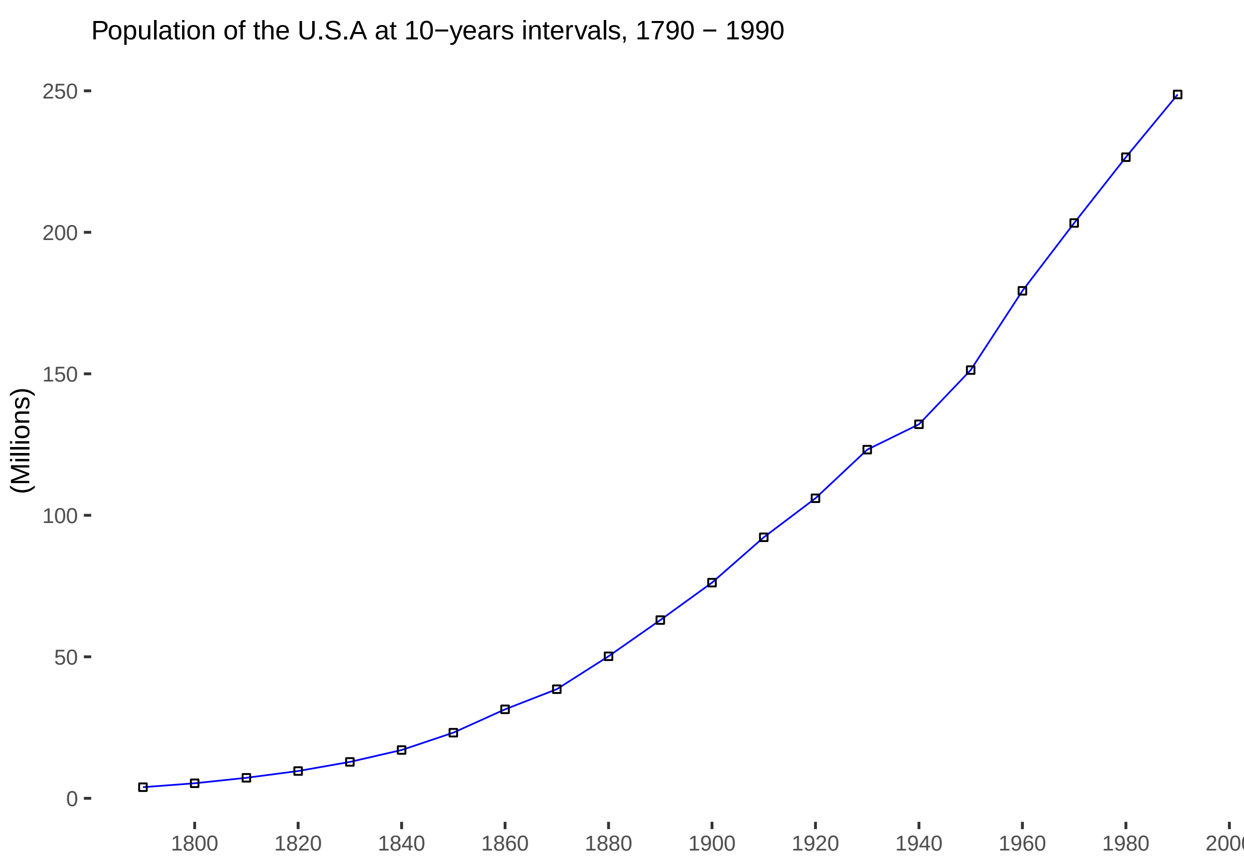
Example 1.1.5

```r
population <-
    get_data_frame( c(
        3929214, 5308483, 7239881, 9638453, 12860702,
        17063353, 23191876, 31443321, 38558371, 50189209,
        62979766, 76212168, 92228496, 106021537, 123202624,
        132164569, 151325798, 179323175, 203302031,
        226542203, 248709873
    ), "01-Jan-1790", period = "yearly", multiplier = 10 )
head( population )
```

t	date	value
1	1790-01-01	3929214
2	1800-01-01	5308483
3	1810-01-01	7239881
4	1820-01-01	9638453
5	1830-01-01	12860702
6	1840-01-01	17063353

Figure 1-5

```
plot_data(
  # in order to get the value in millions
  data = population %>% mutate( value = value / 1000000 ),
  title = "Population of the U.S.A at 10-years intervals, 1790 - 1990",
  y_label = "(Millions)",
  x_ticks = 10
)
```



Example 1.3.4

Set the squared t as one of the predictor:

```r
example.1.3.4.data <-
    population %>% mutate( t.squared = t ^ 2 )
```

Build the trend model:

```r
example.1.3.4.lm <-
    lm(
        value ~ t + t.squared,
        data = example.1.3.4.data
    )
summary( example.1.3.4.lm )
```

```
##
## Call:
## lm(formula = value ~ t + t.squared, data = example.1.3.4.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -6947521  -358167   436285  1481410  3391761
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6957920    1998526   3.482  0.00266 **
## t           -2159870     418437  -5.162 6.55e-05 ***
## t.squared     650634      18472  35.223  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2767000 on 18 degrees of freedom
## Multiple R-squared:  0.9989, Adjusted R-squared:  0.9988
## F-statistic:  8050 on 2 and 18 DF,  p-value: < 2.2e-16
```

Prediction of population of year 2000, $\hat{m}_{22}$:

```r
predict( example.1.3.4.lm, data.frame( t = 22, t.squared = 22 ^ 2 ) )
```

```
##         1
## 274347573
```

Figure 1-8

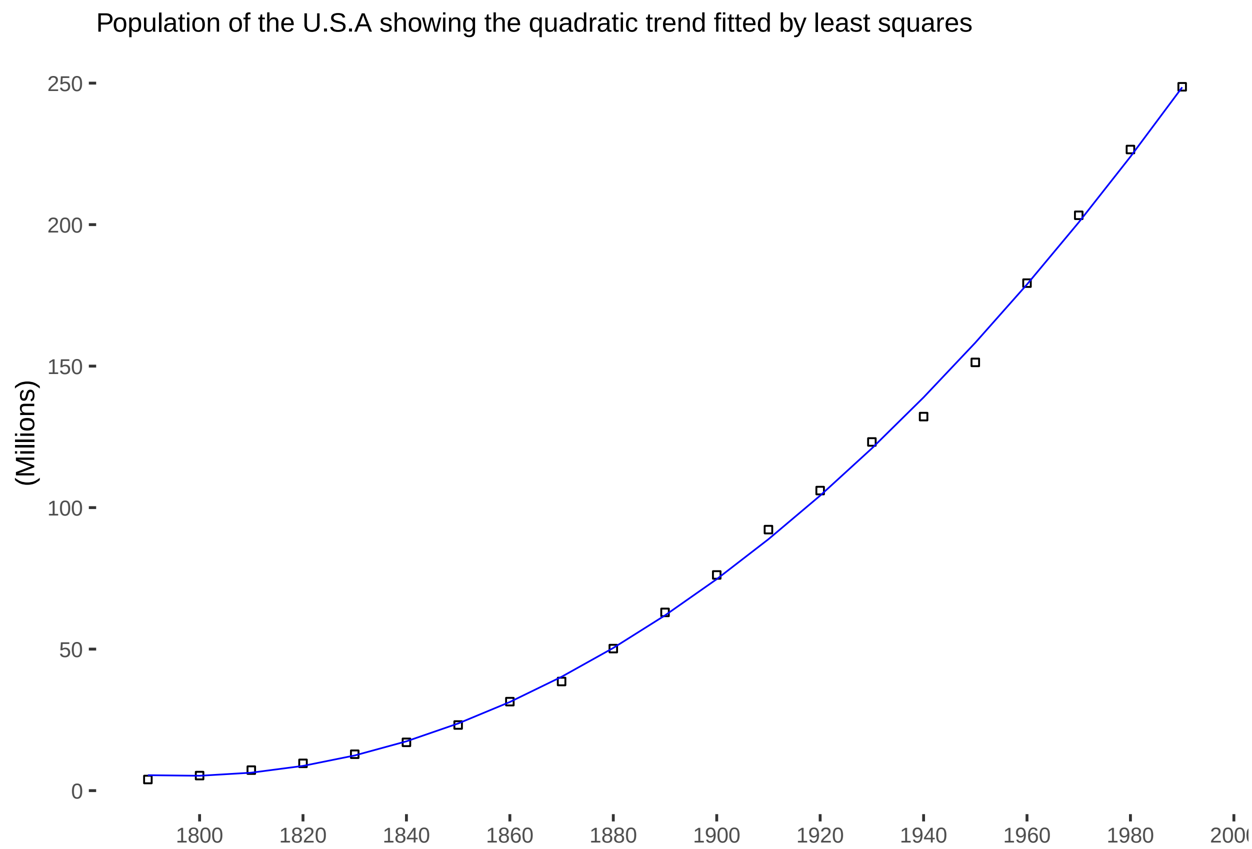
Quadratic trend fitted values:

```r
example.1.3.4.data$fitted <- example.1.3.4.lm$fitted.values
```

Plotting the result:

```r
plot_data(
    # in order to get the value in millions
    data = example.1.3.4.data %>% mutate( value = value / 1000000 ),
```

```
    title = "Population of the U.S.A showing the quadratic trend fitted by least squares",
    y_label = "(Millions)",
    x_ticks = 10,
    line = FALSE
) + geom_line( aes( y = fitted / 1000000 ), color = "blue", size = 0.3 )
```



Example 1.3.5

```
lake <- get_data_frame( itsmr::lake, "01-Jan-1875", "yearly" )

example.1.3.5.lm <- lm( value ~ t, lake )
summary( example.1.3.5.lm )
```

```
##
## Call:
## lm(formula = value ~ t, data = lake)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.50997 -0.72726  0.00083  0.74402  2.53565
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 10.202037   0.230111  44.335  < 2e-16 ***
## t           -0.024201   0.004036  -5.996 3.55e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.13 on 96 degrees of freedom
## Multiple R-squared:  0.2725, Adjusted R-squared:  0.2649
## F-statistic: 35.95 on 1 and 96 DF,  p-value: 3.545e-08
```

Figure 1-9

```
plot_data(
    data = lake %>% mutate( fitted = example.1.3.5.lm$fitted.values ),
    title = "Level of Lake Huron 1875-1972 showing the line fitted by least squares",
    x_ticks = 10
) + geom_line( aes( y = fitted ), size = 0.3 )
```

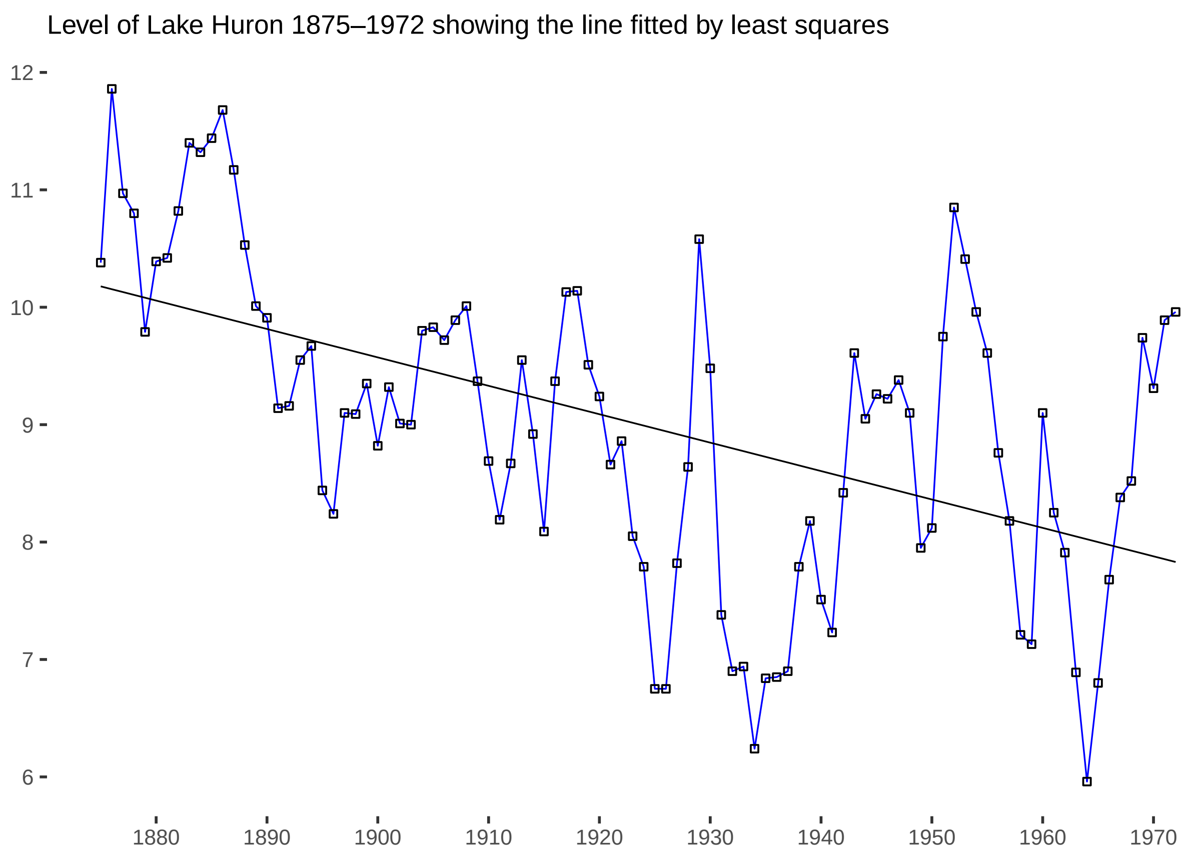
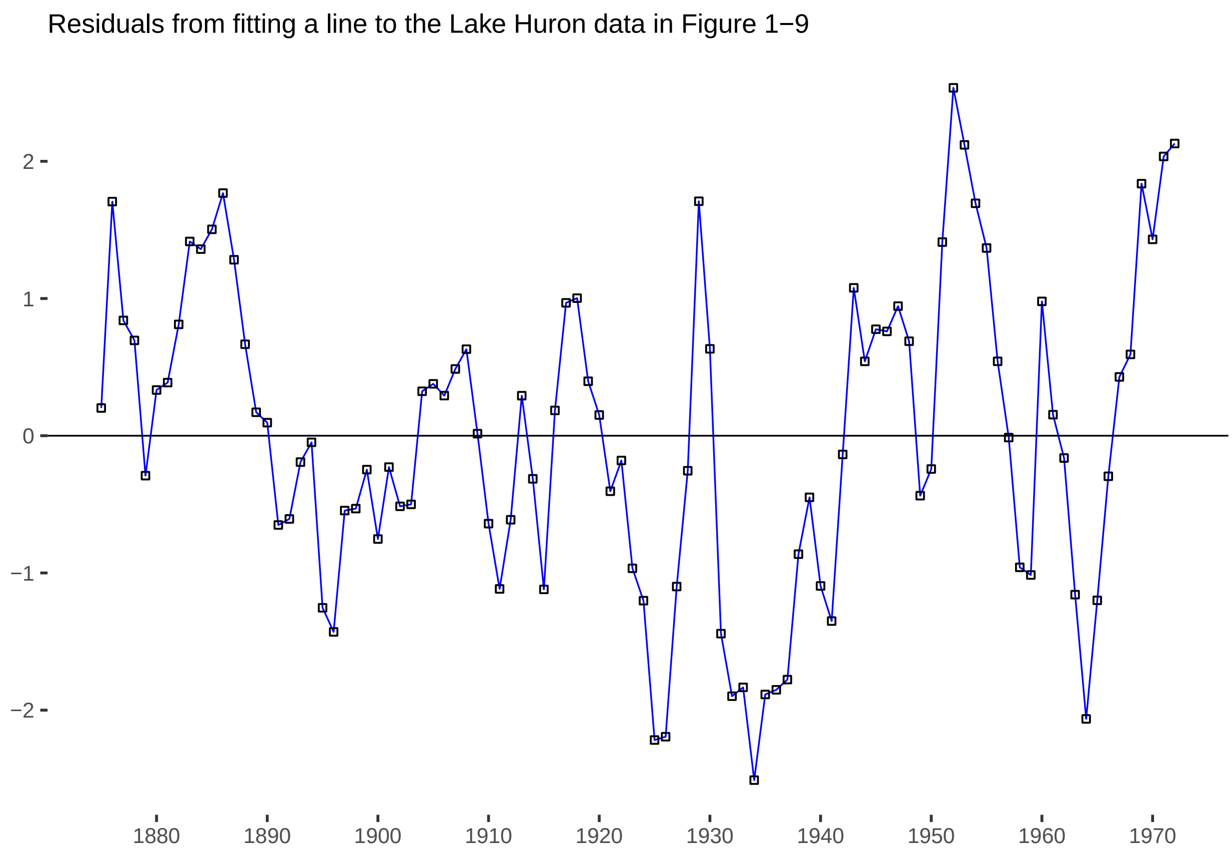


Figure 1-10

```
plot_data(
    data = lake %>% mutate( value = example.1.3.5.lm$residuals ),
```

```
    title = "Residuals from fitting a line to the Lake Huron data in Figure 1-9",
    x_ticks = 10
) + geom_hline( yintercept = 0, size = 0.3 )
```



Example 1.3.6

Equation 1.3.3 in the book is actually a type of multiple linear regression with predictors as combination of harmonic (cosine, sine) functions.

```
example.1.3.6.data <-
    deaths %>% mutate(
        lambda.1 = 1 * (2 * pi / 12),
        lambda.2 = 2 * (2 * pi / 12)
    ) %>% mutate(
        cos.1 = cos( lambda.1 * t ),
        sin.1 = sin( lambda.1 * t ),
        cos.2 = cos( lambda.2 * t ),
        sin.2 = sin( lambda.2 * t )
    )
```

In the code chunk above, `lambda.j` is the equivalent of $\lambda_j = j \times \frac{2\pi}{d}$ in equation 1.3.3 and in example 1.3.6 we have $k = 2$, $d = 12$, and $j = \{1, 2\}$.

The fitted trend model is:

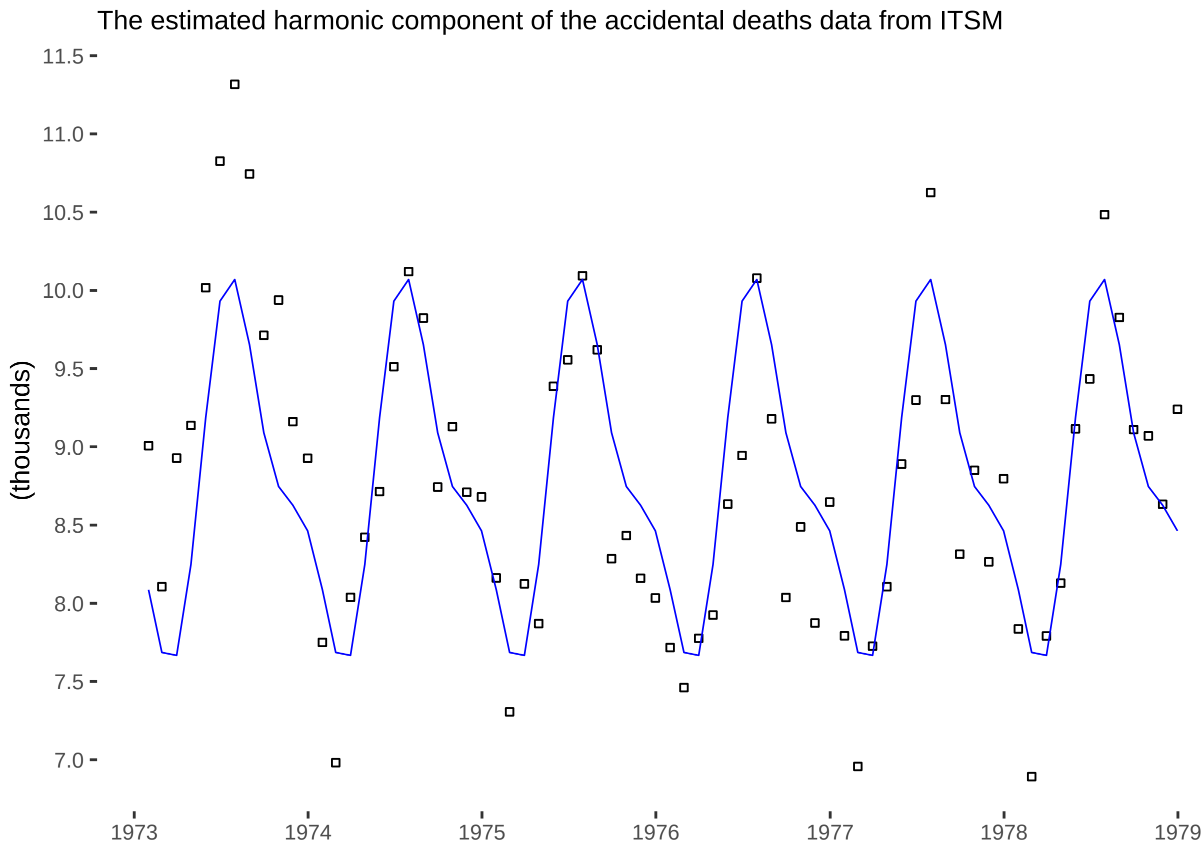
```r
example.1.3.6.lm <-
    lm( value ~ cos.1 + sin.1 + cos.2 + sin.2, example.1.3.6.data )
summary( example.1.3.6.lm )
```

```
##
## Call:
## lm(formula = value ~ cos.1 + sin.1 + cos.2 + sin.2, data = example.1.3.6.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1053.1  -375.8     7.3   342.6  1261.2
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8787.74      66.28 132.579  < 2e-16 ***
## cos.1        -734.04      93.74  -7.831 4.86e-11 ***
## sin.1        -711.64      93.74  -7.592 1.31e-10 ***
## cos.2         409.28      93.74   4.366 4.49e-05 ***
## sin.2          99.16      93.74   1.058    0.294
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 562.4 on 67 degrees of freedom
## Multiple R-squared:  0.675,  Adjusted R-squared:  0.6556
## F-statistic: 34.78 on 4 and 67 DF,  p-value: 1.053e-15
```

Figure 1-11

And the plot of the fitted value is:

```r
example.1.3.6.data$fitted <-
    example.1.3.6.lm$fitted.values

plot_data(
    data = example.1.3.6.data %>% mutate( value = value / 1000 ),
    title = "The estimated harmonic component of the accidental deaths data from ITSM",
    y_label = "(thousands)",
    x_ticks = 10,
    line = FALSE
) + geom_line( aes( y = fitted / 1000 ), color = "blue", size = 0.3 )
```



Example 1.4.1

By Definition 1.4.1

$\gamma_X(t+h,t) = E[(X_{t+h} - E(X_{t+h}))(X_t - E(X_t))]$

Since we have i.i.d noise, where $E(X_t) = 0$ for all t, then

$\gamma_X(t+h,t) = E[(X_{t+h} - E(X_{t+h}))(X_t - E(X_t))]$

$\gamma_X(t+h,t) = E[(X_{t+h} - 0)(X_t - 0)]$

$\gamma_X(t+h,t) = E[X_{t+h}X_t]$

If $h = 0$, we have:

$\gamma_X(t+0,t) = \gamma_X(t,t) = E[X_{t+0}X_t] = E[X_tX_t] = E[X_t^2] = Var(X_t) + E[X_t]^2$

Where $Var(X_t) = \sigma^2$ and $E(X_t) = 0$

$\gamma_X(t,t) = \sigma^2 + 0 = \sigma^2$

If $h \neq 0$, by independence:

$\gamma_X(t+h,t) = E[X_{t+h}X_t] = E[X_{t+h}]E[X_t] = 0 \times 0 = 0$

Thus we have:

$$\gamma_X(t+h,t) = \begin{cases} \sigma^2, if\ h = 0 \\ 0, if\ h \neq 0 \end{cases}$$

Example 1.4.2

Simplified, white noise differs from i.i.d the identically distributed part and independence part but keeping the uncorrelated part, thus the covariance function still the same as i.i.d noise in Example 1.4.2 above.

Example 1.4.4

By definition 1.4.3 and covariance linearity:

$\gamma_X(t+h,t) = Cov[X_{t+h}, X_t] = Cov[Z_{t+h} + \theta Z_{t+h-1}, Z_t + \theta Z_{t-1}]$

$\gamma_X(t+h,t) = Cov[Z_{t+h}, Z_t] + Cov[Z_{t+h}, \theta Z_{t-1}] + Cov[\theta Z_{t+h-1}, Z_t] + Cov[\theta Z_{t+h-1}, \theta Z_{t-1}]$

$\gamma_X(t+h,t) = Cov[Z_{t+h}, Z_t] + \theta Cov[Z_{t+h}, Z_{t-1}] + \theta Cov[Z_{t+h-1}, Z_t] + \theta^2 Cov[Z_{t+h-1}, Z_{t-1}]$

If $h = 0$:

$\gamma_X(t+0,t) = \gamma_X(t,t) = Cov[Z_{t+0}, Z_t] + \theta Cov[Z_{t+0}, Z_{t-1}] + \theta Cov[Z_{t+0-1}, Z_t] + \theta^2 Cov[Z_{t+0-1}, Z_{t-1}]$

$\gamma_X(t,t) = Cov[Z_t, Z_t] + \theta Cov[Z_t, Z_{t-1}] + \theta Cov[Z_{t-1}, Z_t] + \theta^2 Cov[Z_{t-1}, Z_{t-1}]$

$\gamma_X(t,t) = \sigma^2 + \theta \times 0 + \theta \times 0 + \theta^2 \times \sigma^2 = \sigma^2(1+\theta^2)$

$\gamma_X(t,t) = \sigma^2(1+\theta^2)$

If $h = 1$:

$\gamma_X(t+1,t) = Cov[Z_{t+1}, Z_t] + \theta Cov[Z_{t+1}, Z_{t-1}] + \theta Cov[Z_{t+1-1}, Z_t] + \theta^2 Cov[Z_{t+1-1}, Z_{t-1}]$

$\gamma_X(t+1,t) = Cov[Z_{t+1}, Z_t] + \theta Cov[Z_{t+1}, Z_{t-1}] + \theta Cov[Z_t, Z_t] + \theta^2 Cov[Z_t, Z_{t-1}]$

$\gamma_X(t+1,t) = 0 + \theta \times 0 + \theta \times \sigma^2 + \theta^2 \times 0$

$\gamma_X(t+1,t) = \theta\sigma^2$

If $h = -1$:

$\gamma_X(t-1,t) = Cov[Z_{t-1}, Z_t] + \theta Cov[Z_{t-1}, Z_{t-1}] + \theta Cov[Z_{t-1-1}, Z_t] + \theta^2 Cov[Z_{t-1-1}, Z_{t-1}]$

$\gamma_X(t-1,t) = Cov[Z_{t-1}, Z_t] + \theta Cov[Z_{t-1}, Z_{t-1}] + \theta Cov[Z_{t-2}, Z_t] + \theta^2 Cov[Z_{t-2}, Z_{t-1}]$

$\gamma_X(t-1,t) = 0 + \theta \times \sigma^2 + \theta \times 0 + \theta^2 \times 0$

$\gamma_X(t-1,t) = \theta\sigma^2$

If $|h| > 1$, none of the pair of covariance in equation below:

$\gamma_X(t+h,t) = Cov[Z_{t+h}, Z_t] + \theta Cov[Z_{t+h}, Z_{t-1}] + \theta Cov[Z_{t+h-1}, Z_t] + \theta^2 Cov[Z_{t+h-1}, Z_{t-1}]$

will have the same index, thus all covariance terms equal to 0:

$\gamma_X(t+h,t) = 0$

Thus we have:

$$\gamma_X(t+h,t) = \begin{cases} \sigma^2(1+\theta^2),\ if\ h = 0 \\ \sigma^2\theta,\ if\ h = \pm 1 \\ 0,\ if\ |h| > 1 \end{cases}$$

by definition 1.4.3,

$$\rho_X(h) = \frac{\gamma_X(h)}{\gamma_X(0)} = \frac{\gamma_X(t+h,t)}{\gamma_X(t,t)} = \frac{\gamma_X(t+h,t)}{\sigma^2(1+\theta^2)} = \begin{cases} 1,\ if\ h = 0 \\ \theta/(1+\theta^2),\ if\ h = \pm 1 \\ 0,\ if\ |h| > 1 \end{cases}$$

Example 1.4.5

First let's check for the mean of X_t:

$E(X_t) = E(\phi X_{t-1} + Z_t) = E(\phi X_{t-1}) + E(Z_t) = E(\phi X_{t-1}) + 0 = E(\phi X_{t-1})$

$E(X_t) = E(\phi X_{t-1}) = \phi E(X_{t-1}) = \phi E(\phi X_{t-2} + Z_{t-1}) = \phi E(\phi X_{t-2}) + E(Z_{t-1}) = \phi E(\phi X_{t-2}) + 0 = \phi E(\phi X_{t-2})$

$E(X_t) = \phi E(\phi X_{t-2}) = \phi^2 E(X_{t-2})$

...

$E(X_t) = \lim_{n\to\infty} \phi^n E(X_{t-n}) = 0$, since $|\phi| < 1$

Then we have

$\gamma_X(h) = Cov(X_t, X_{t-h}) = Cov(\phi X_{t-1} + Z_t, X_{t-h})$

by covariance linearity, and the given fact that Z_t is uncorrelated with X_s for each $s < t$,

$\gamma_X(h) = Cov(\phi X_{t-1}, X_{t-h}) + Cov(Z_t, X_{t-h}) = Cov(\phi X_{t-1}, X_{t-h}) + 0$

$\gamma_X(h) = Cov(\phi X_{t-1}, X_{t-h}) = \phi Cov(X_{t-1}, X_{t-h}) = \phi \gamma_X((t-1) - (t-h))$

$\gamma_X(h) = \phi \gamma_X(h-1) = \phi^2 \gamma_X(h-2) = ... = \phi^h \gamma_X(0)$

Next as $\gamma_x(h) = \gamma_x(-h)$ and using definition 1.4.3

$\rho_X(h) = \frac{\gamma_x(h)}{\gamma_x(0)} = \phi^{|h|},\ h = 0, \pm 1, ...$

with

$\gamma_X(0) = Cov(X_t, X_t) = Cov(\phi X_{t-1} + Z_t, \phi X_{t-1} + Z_t)$

by covariance linearity, and the given fact that Z_t is uncorrelated with X_s for each $s < t$,

$\gamma_X(0) = Cov(\phi X_{t-1}, \phi X_{t-1}) + Cov(\phi X_{t-1}, Z_t) + Cov(Z_t, \phi X_{t-1}) + Cov(Z_t, Z_t)$

$\gamma_X(0) = \phi^2 Cov(X_{t-1}, X_{t-1}) + \phi Cov(X_{t-1}, Z_t) + \phi Cov(Z_t, X_{t-1}) + Cov(Z_t, Z_t)$

$\gamma_X(0) = \phi^2 \gamma_X(0) + \phi \times 0 + \phi \times 0 + \sigma^2$

$\gamma_X(0) = \phi^2 \gamma_X(0) + \sigma^2$

$\gamma_X(0) = \frac{\sigma^2}{1-\phi^2}$

Definition 1.4.4

Try to recreate your own version of each of these two functions based on Definition 1.4.4

Sample Auto-Covariance Function

```
sacov <-
    function ( series, shift ) {
        series_length <- length( series )
        abs_shift <- abs( shift )
        avg <- sum( series ) / series_length
        calculate_acv <-
            function ( shift_index ) {
                index <- 1:(series_length - shift_index)
```

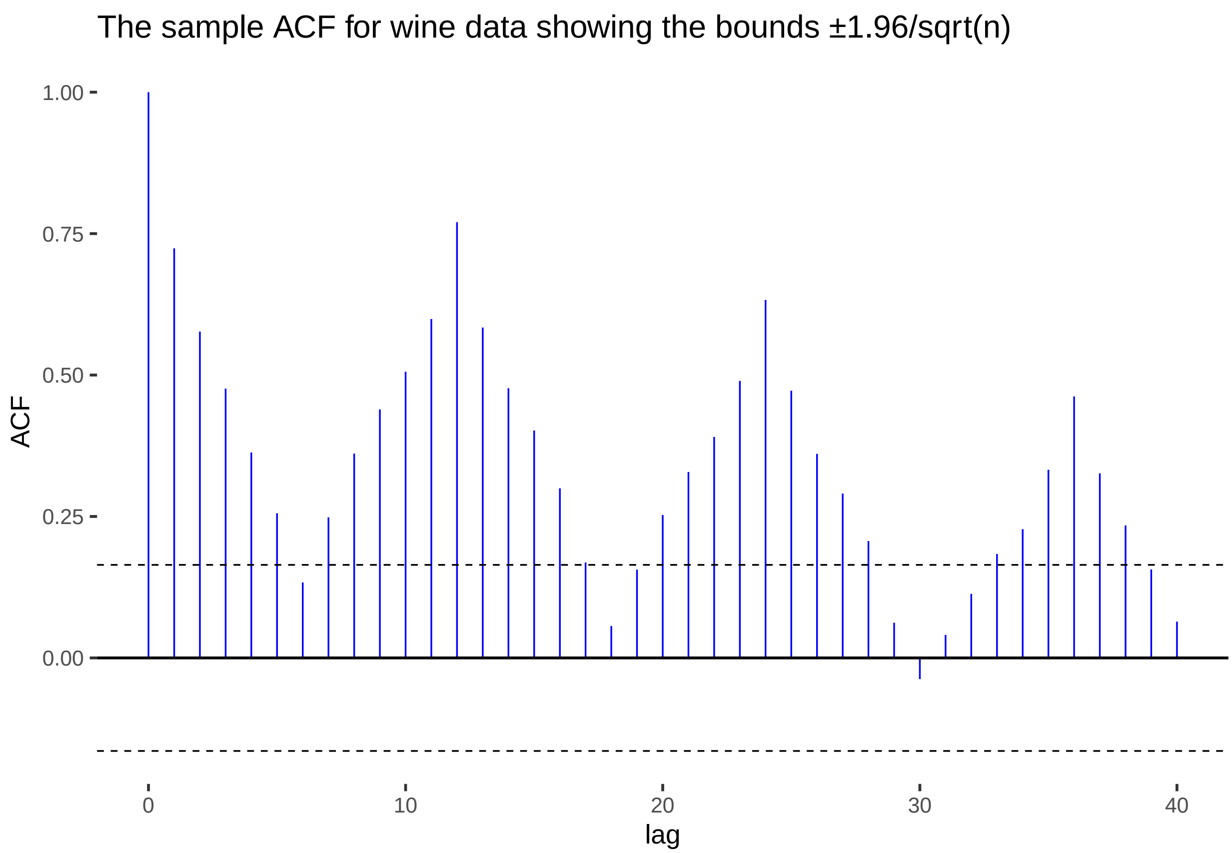
```
                sum(
                    ( series[ index + shift_index ] - avg ) *
                    ( series[ index ] - avg )
                ) / series_length
            }
        return( sapply( abs_shift, calculate_acv ) )
    }
```

Sample Auto-Correlation Function

```
sacor <-
    function( series, shift ) {
        return( sacov( series, shift ) / sacov( series, 0 ) )
    }

plot_acor <- function ( series, title, ci = 0.95, max_lag = 40 ) {
    sacor_index <- 0:min( length( series ), max_lag )
    max_plot_lag <- max( sacor_index )
    data_sacor <- data.frame(
        index = sacor_index
    )
    data_sacor$value <- sacor( series, sacor_index )
    ci_line <- qnorm( ( ( 1 - ci ) / 2 ) + c( 0, ci ) ) / sqrt( length( series ) )
    data_acor <-
        data.frame(
            lag = rep( data_sacor$index, 2 ),
            acor = c( rep( 0, max_plot_lag + 1 ), data_sacor$value )
        )
    data_plot <-
        ggplot( data_acor , aes( lag, acor ) ) +
        ggtitle( title ) +
        ylab("ACF") +
        geom_line(
            aes( group = lag ),
            size = 0.3,
            color = "blue"
        ) +
        geom_hline(
            yintercept = 0,
            color = "black"
        ) +
        geom_hline(
            yintercept = ci_line,
            linetype = "dashed",
            color = "black",
            size = 0.3
        ) +
        theme_tufte() +
        theme( text = element_text( family = "sans", size = 10 ) )
    return( data_plot )
}
```

Figure 1-14

```
plot_acor( wine$value, "The sample ACF for wine data showing the bounds \u00B11.96/sqrt(n)" )
```

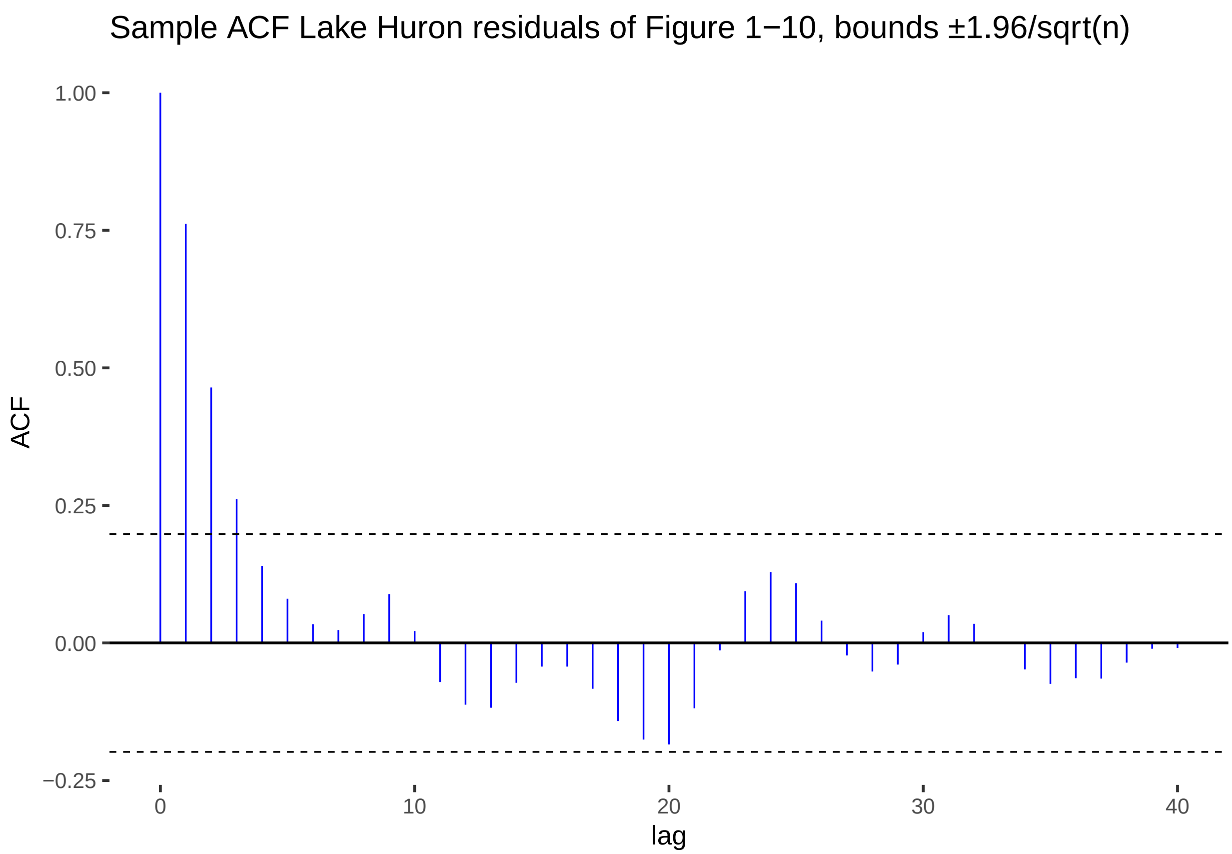


1.4.2 A Model for the Lake Huron Data

Figure 1-10

Here is the residual and the ACF plot of the residual just by fitting trend line directly in Lake Huron data:

```
lake_residual <-
    data.frame(
        t = 1:length(example.1.3.5.lm$residuals),
        value = example.1.3.5.lm$residuals )

plot_acor(
    lake_residual$value,
    "Sample ACF Lake Huron residuals of Figure 1-10, bounds \u00B11.96/sqrt(n)" )
```



AR(1) Model

Below are where the coefficient of equation 1.4.4 in the book comes from:

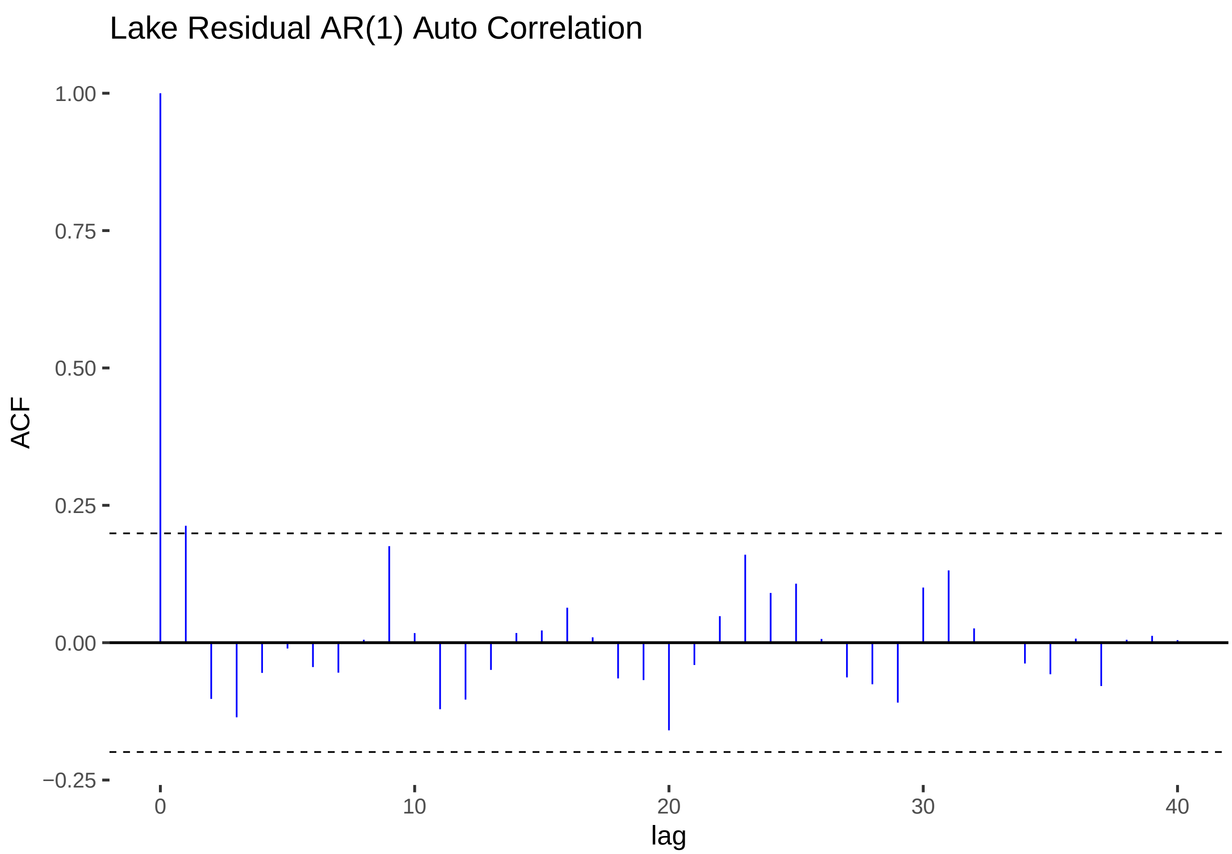
```
lake_residual_ar1 <-
    inner_join(
        lake_residual %>%
            mutate( value_0 = value ) %>%
            select( t, value_0 ),
        lake_residual %>%
            mutate( t = t - 1, value_1 = value ) %>%
            select( t, value_1 )
    )

lake_residual_ar1_model <- lm( value_1 ~ 0 + value_0, lake_residual_ar1 )
summary( lake_residual_ar1_model )
```

```
##
## Call:
## lm(formula = value_1 ~ 0 + value_0, data = lake_residual_ar1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.94335 -0.48386  0.01758  0.43251  1.91083
##
```

```
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## value_0  0.79084    0.06556   12.06   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7125 on 96 degrees of freedom
## Multiple R-squared:  0.6025, Adjusted R-squared:  0.5984
## F-statistic: 145.5 on 1 and 96 DF,  p-value: < 2.2e-16
```

As we can see above, the coefficient match the one described in equation 1.4.4. Now we get the residual ACF from the Auto-Regressive AR(1) model:

```
lake_residual_ar1$predicted <-
    predict( lake_residual_ar1_model, lake_residual_ar1 )

lake_residual_ar1$residual <-
    lake_residual_ar1$value_1 -
    lake_residual_ar1$predicted

plot_acor(
  lake_residual_ar1$residual,
  "Lake Residual AR(1) Auto Correlation"
)
```

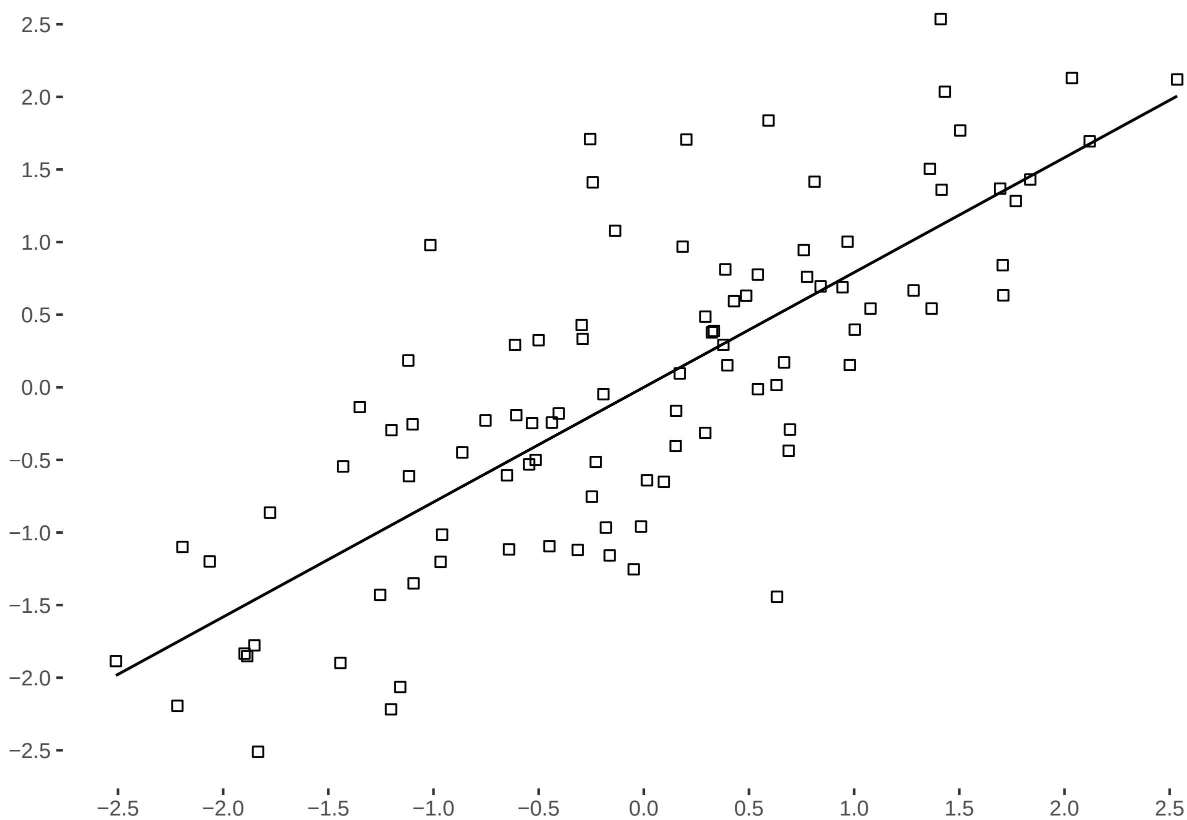


The ACF plot above is way much better than the one in Figure 1-10 where we fit trend directly in Lake Huron data.

Figure 1-16

Below is the residual plot after AR(1) model:

```
ggplot( lake_residual_ar1, aes( x = value_0 ) ) +
    xlab( NULL ) + ylab( NULL  ) +
    geom_point( aes( y = value_1 ), shape = 0 ) +
    geom_line( aes( y = predicted ) ) +
    theme_tufte(base_size = 10,base_family = "sans") +
    scale_x_continuous(breaks = scales::pretty_breaks(n = 11)) +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 8))
```



AR(2) Model

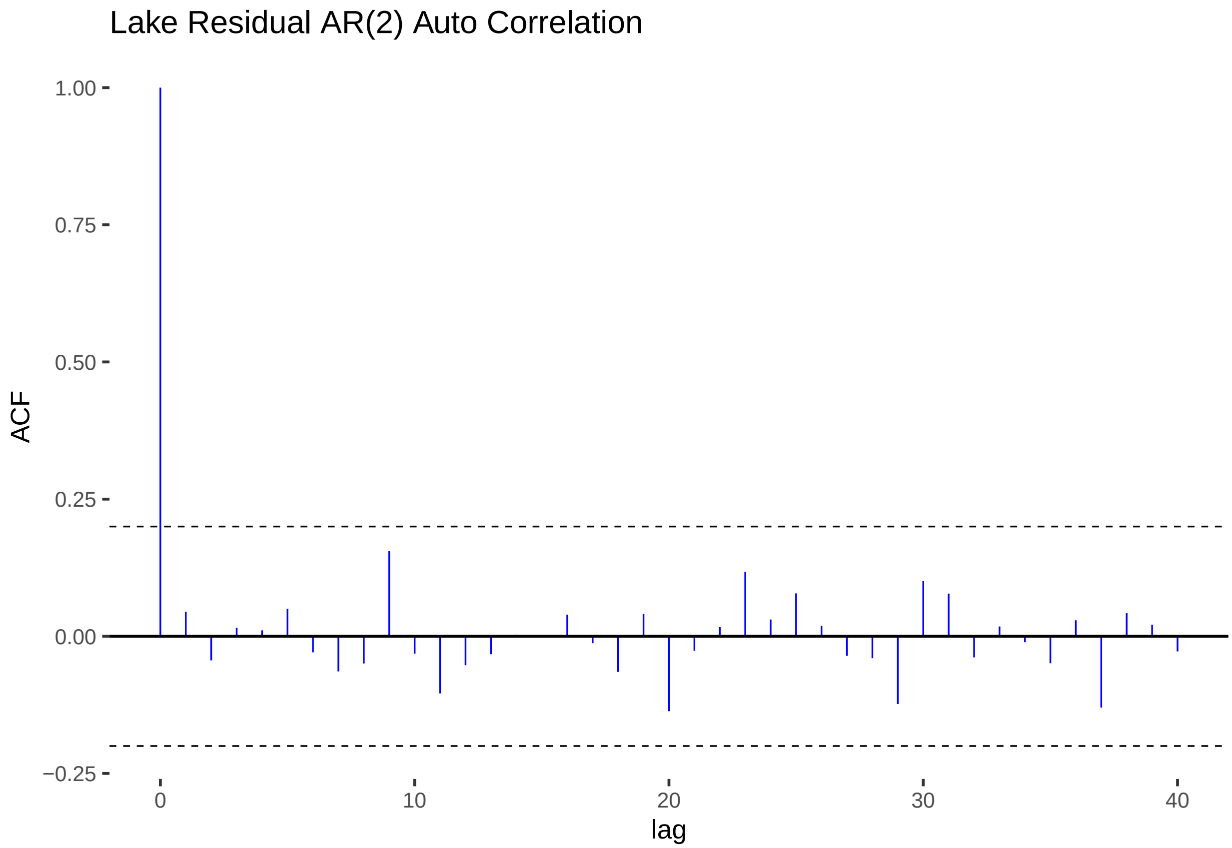
The discussion in the book goes on until creation of an AR(2) model, which we can get from the code below:

```
lake_residual_ar2 <-
    inner_join(
        lake_residual_ar1,
        lake_residual_ar1 %>%
            mutate( t = t - 1, value_2 = value_1 ) %>%
            select( t, value_2 )
    )
```

```r
lake_residual_ar2_model <- lm( value_2 ~ 0 + value_1 + value_0, lake_residual_ar2 )
summary( lake_residual_ar2_model )
```

```
##
## Call:
## lm(formula = value_2 ~ 0 + value_1 + value_0, data = lake_residual_ar2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.59273 -0.45997 -0.02406  0.39543  1.72440
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## value_1  1.00199    0.09669  10.362  < 2e-16 ***
## value_0 -0.28339    0.09842  -2.879  0.00493 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6731 on 94 degrees of freedom
## Multiple R-squared:  0.6442, Adjusted R-squared:  0.6366
## F-statistic:  85.1 on 2 and 94 DF,  p-value: < 2.2e-16
```

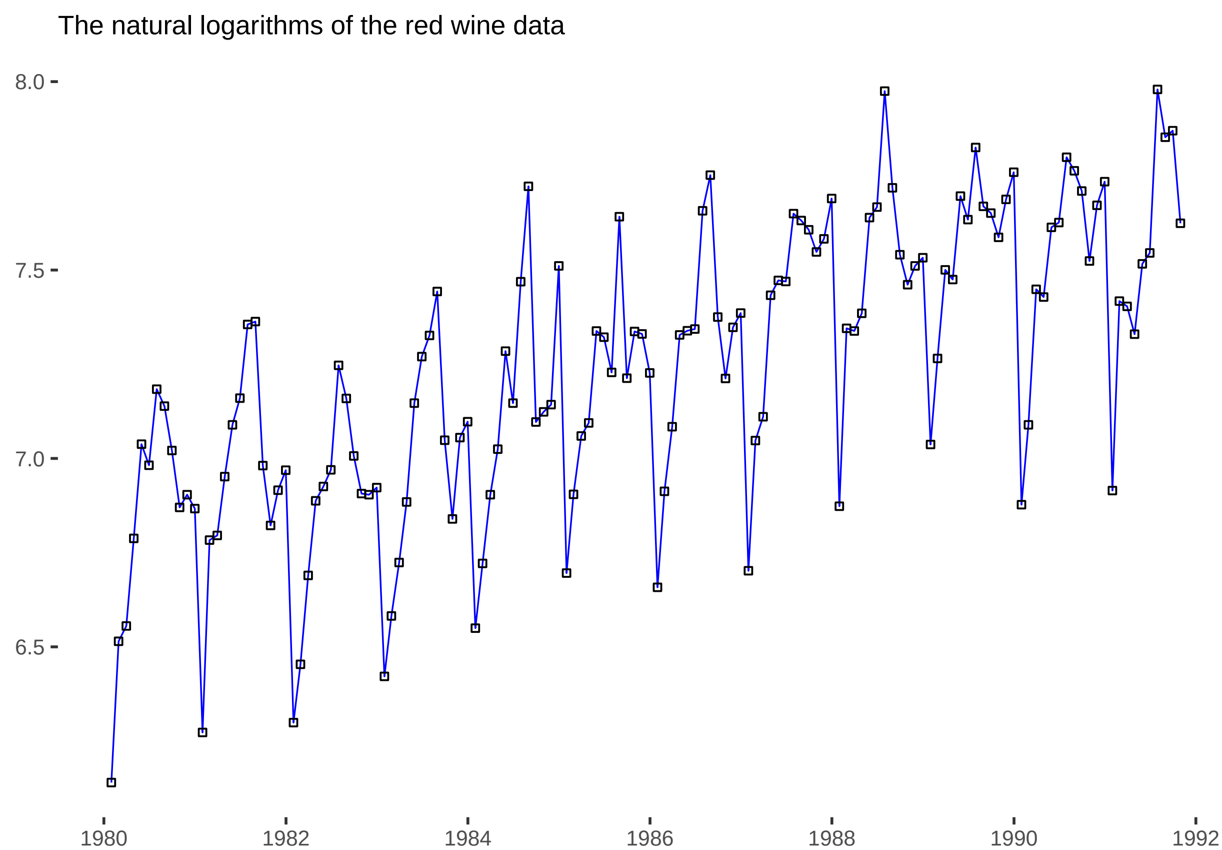
```r
plot_acor(
    lake_residual_ar2$value_2 -
    predict( lake_residual_ar2_model, lake_residual_ar2 ),
    "Lake Residual AR(2) Auto Correlation"
)
```



We can see it is a better fit compared to the AR(1) residual ACF plot with no significant value out the the threshold at all until lag 40.

Figure 1-17

```
plot_data(
    wine %>% mutate( value = log( value ) ),
    "The natural logarithms of the red wine data"
)
```



1.5.1 Estimation and Elimination of Trend in the Absence of Seasonality

Smoothing with a finite moving average filter

Below are the code related with equation 1.5.5 and 1.5.12 (note that the equation 1.5.12 is under section 1.5.2.1 Method S1: Estimation of Trend and Seasonal Components):

```
ma <- function ( series, ma_window, extended = TRUE ) {
    series_length <- length( series )
    q <- floor( ma_window / 2 )
    nominator_length <- (2 * q) + 1
    multiplier <- rep( 1, nominator_length )
    if ( (ma_window %% 2) != 1 ) {
        multiplier[ 1 ] <- 0.5
        multiplier[ nominator_length ] <- 0.5
    }
    if ( extended ) {
        ma_index <- (1:series_length) + q
        ma_series <-
            c(
                rep( series[ 1 ], q ),
                series,
                rep( series[ series_length ], q )
            )
    } else {
```

```
        ma_index <- ((1 + q):(series_length - q))
        ma_series <- series
    }
    return(sapply(
        ma_index,
        function ( i ) {
            return(
                sum( ma_series[(i - q):(i + q)] * multiplier ) /
                    ma_window
            )
        }
    ))
}
```

Figure 1-18

```
strikes <- get_data_frame( itsmr::strikes, "01-Jan-1950", "yearly" )

strikes$ma_5 <- ma( strikes$value, 5 )

plot_data(
  # to get value in thousands
  strikes %>% mutate( value = value / 1000 ),
  "Simple 5-term moving average of Strike data",
  line = FALSE
) +
  geom_line( aes( y = ma_5 / 1000 ), color = "blue", size = 0.3 ) +
  ylab("(thousands)")
```

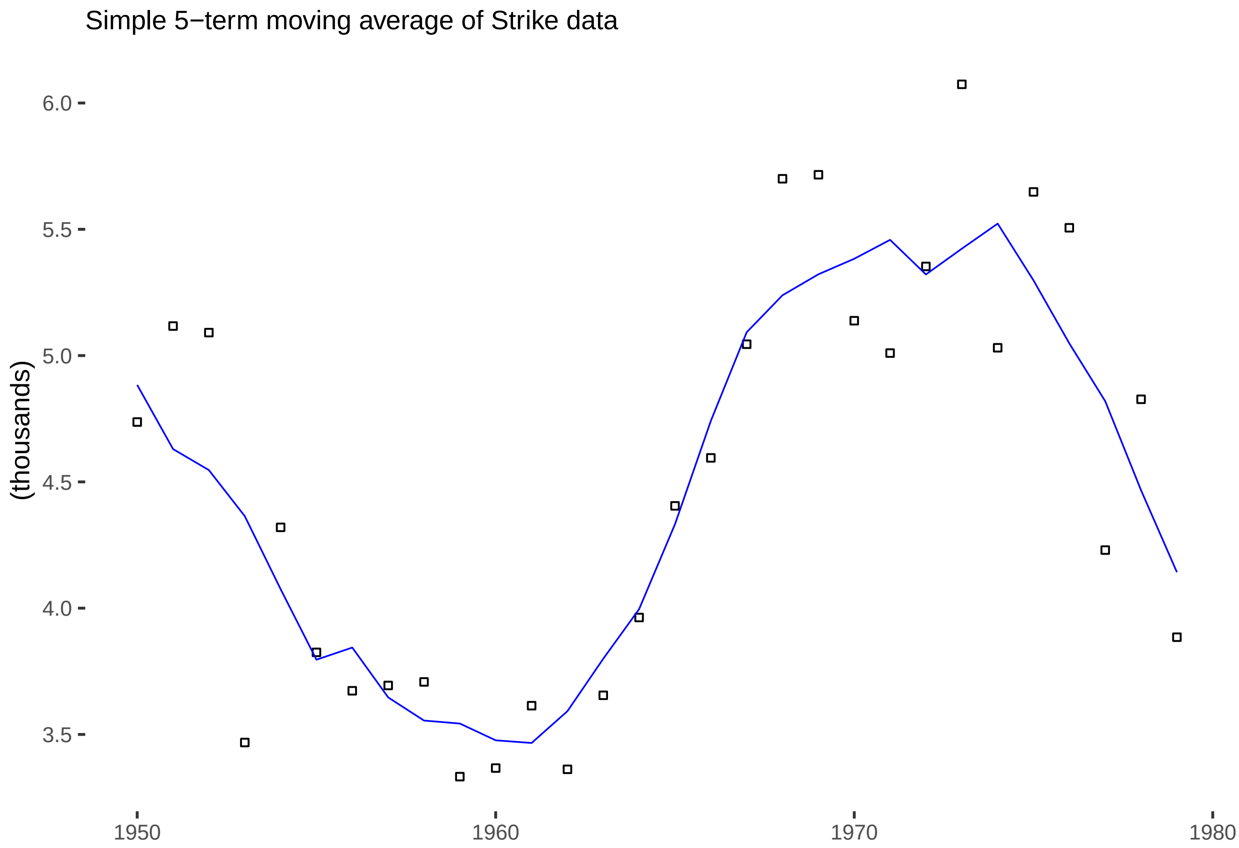
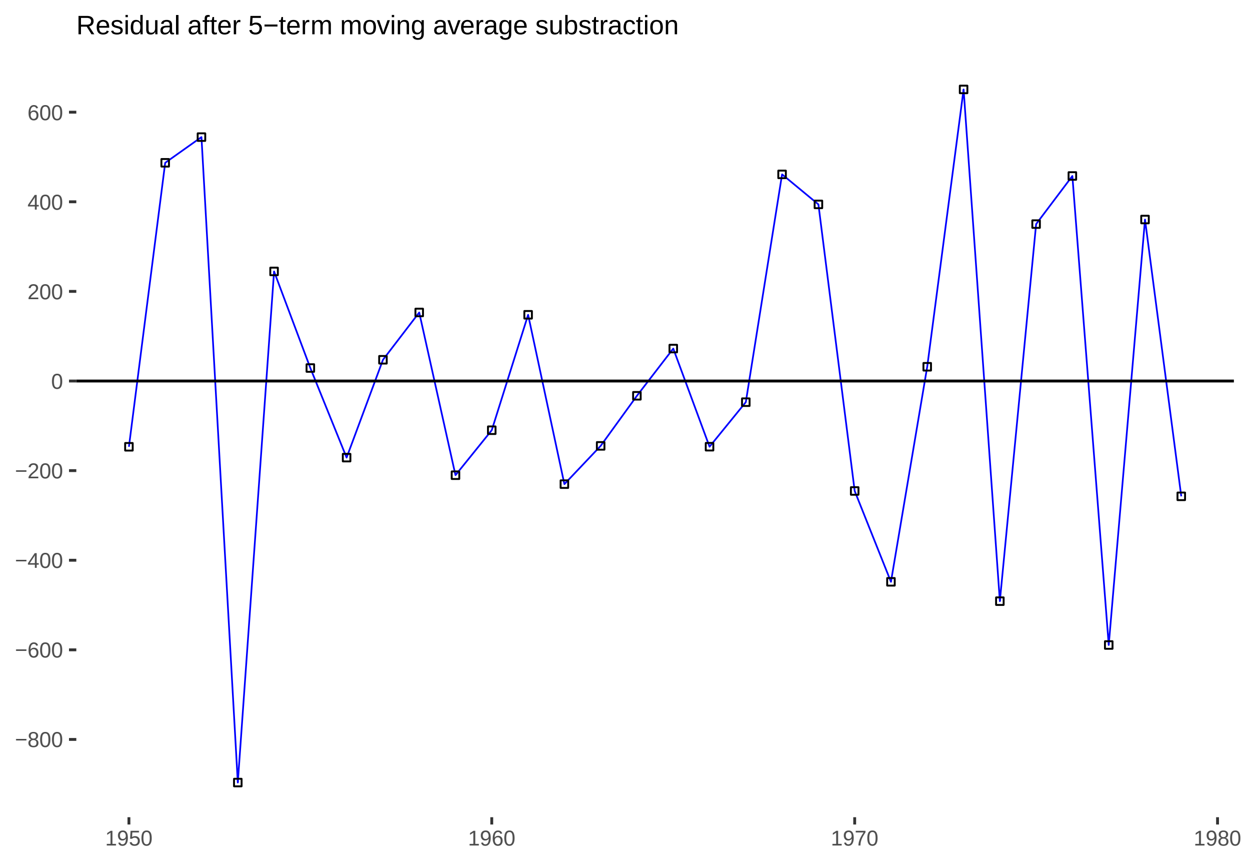


Figure 1-19

```
plot_data(
    strikes %>% mutate( value = value - ma_5 ),
    "Residual after 5-term moving average substraction"
) + geom_hline( yintercept = 0 )
```



1.5.1.1 Method 2: Trend Elimination by Differencing

Try to write the code of differencing by yourself to get the sense of how differencing work, here I got my interpretation of the code:

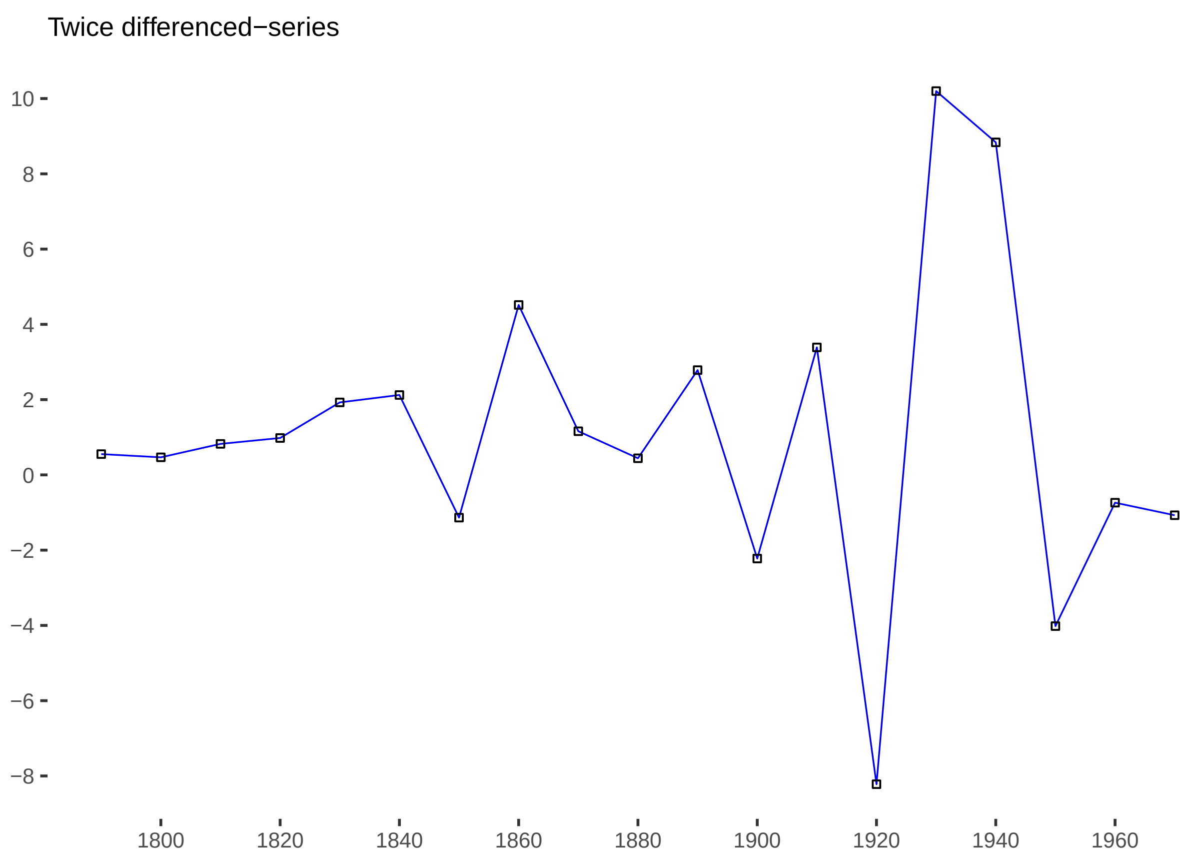
```
backshift <- function ( series, repetition, shift = 1 ) {
    result <- NULL
    if ( repetition > 0 ) {
        series_length <- length( series )
        result <-
            backshift(
                series[ (1 + shift):series_length ] -
                    series[ 1:(series_length - shift) ],
                repetition - 1
            )
    } else {
        result <- series
    }
    return( result )
}
```

Example 1.5.3

```
diff_rep <- 2
population_bs_2 <-
    data.frame(
        index = 1:(length( population$value ) - diff_rep),
        date = population$date[ 1:(length( population$value ) - diff_rep)],
        value = backshift( population$value, diff_rep ) / (10 ^ 6)
    )
```

Figure 1-23

```
plot_data( population_bs_2, "Twice differenced-series", y_ticks = 10, x_ticks = 10 )
```



1.5.2.1 Method S1: Estimation of Trend and Seasonal Components

The code related to equation 1.5.5 and 1.5.12 for moving average has been discussed in Smoothing with a finite moving average filter, which we will use to deseasonalize the data as stated in equation 1.5.13:

```
classic_deseasonalize <- function( series, period ) {
    series_length <- length( series )
```

```
    q <- floor( period / 2 )
    xm_index <- (q + 1):(series_length - q)
    ma_trend <- ma( series, period )
    season_index <- 1:period
    w <- sapply( season_index, function ( i ) {
        xmi <- xm_index[ (xm_index %% period) == (i %% period) ]
        return( mean(series[ xmi ] - ma_trend[ xmi ]) )
    })
    w <- (w - mean( w ))
    seasonal <- sapply( 1:series_length, function ( i ) {
        index <- i %% period
        return(ifelse(
            index == 0,
            w[[ period ]],
            w[[ index ]]
        ))
    })
    return(list(
        seasonal = seasonal,
        deseasonalized = series - seasonal
    ))
}
```

Figure 1-24

```
deaths_deseasonalized_classically <- classic_deseasonalize( deaths$value, 12 )

plot_data(
    data.frame(
        index = 1:nrow( deaths ),
        date = deaths$date,
        value = deaths_deseasonalized_classically$deseasonalized / 1000
    ),
    "Deseasonalized Deaths"
)
```

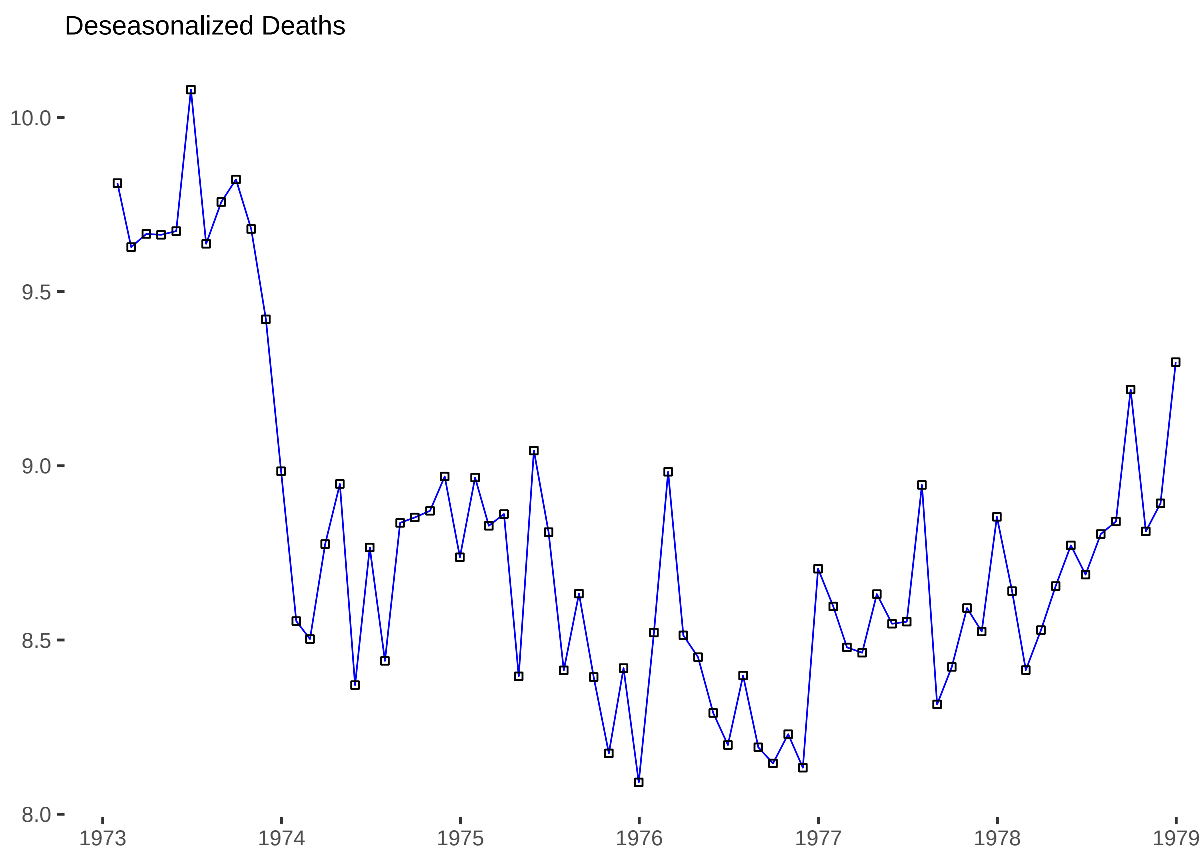
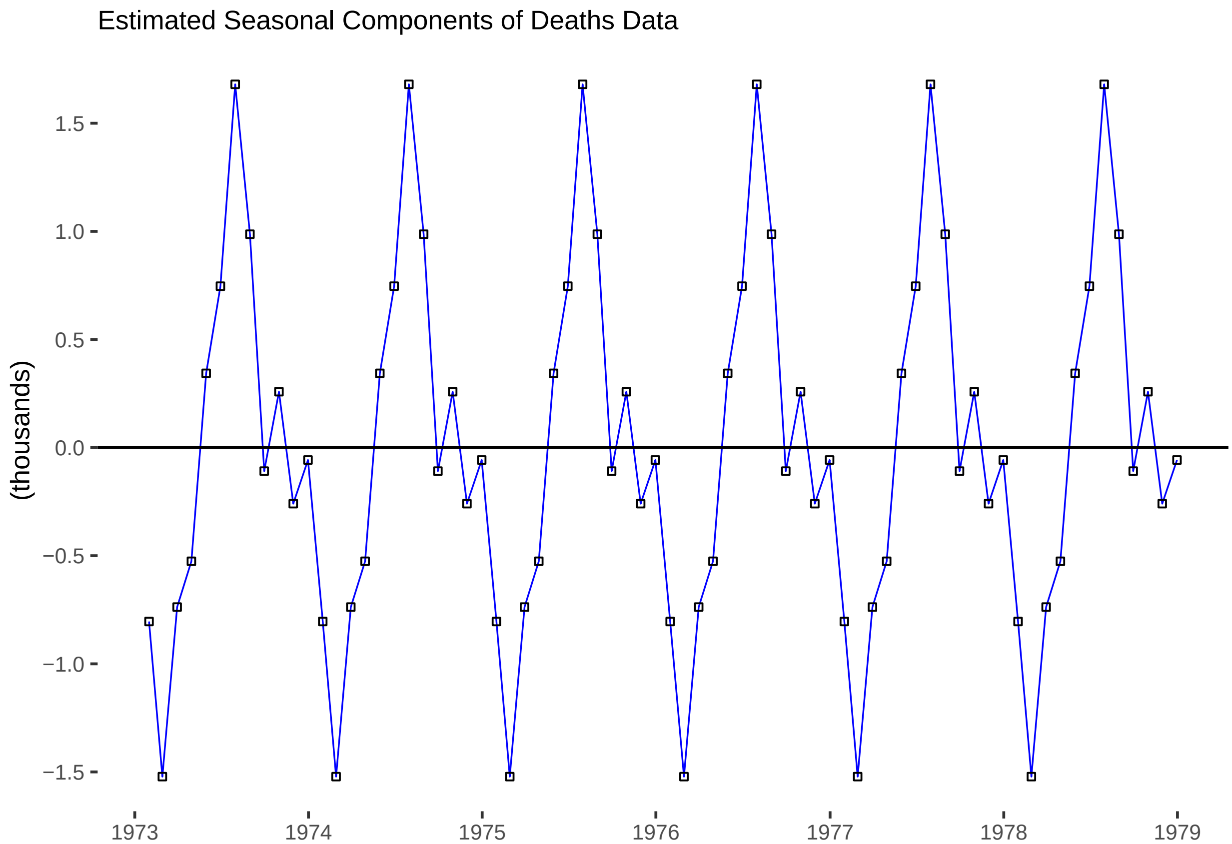


Figure 1-25

```
plot_data(
    data.frame(
        index = 1:nrow( deaths ),
        date = deaths$date,
        value = deaths_deseasonalized_classically$seasonal / 1000
    ),
    "Estimated Seasonal Components of Deaths Data"
) + geom_hline( yintercept = 0 ) + ylab("(thousands)")
```



Example 1.5.4

Getting the quadratic polynomial trend equation in Example 1.5.4:

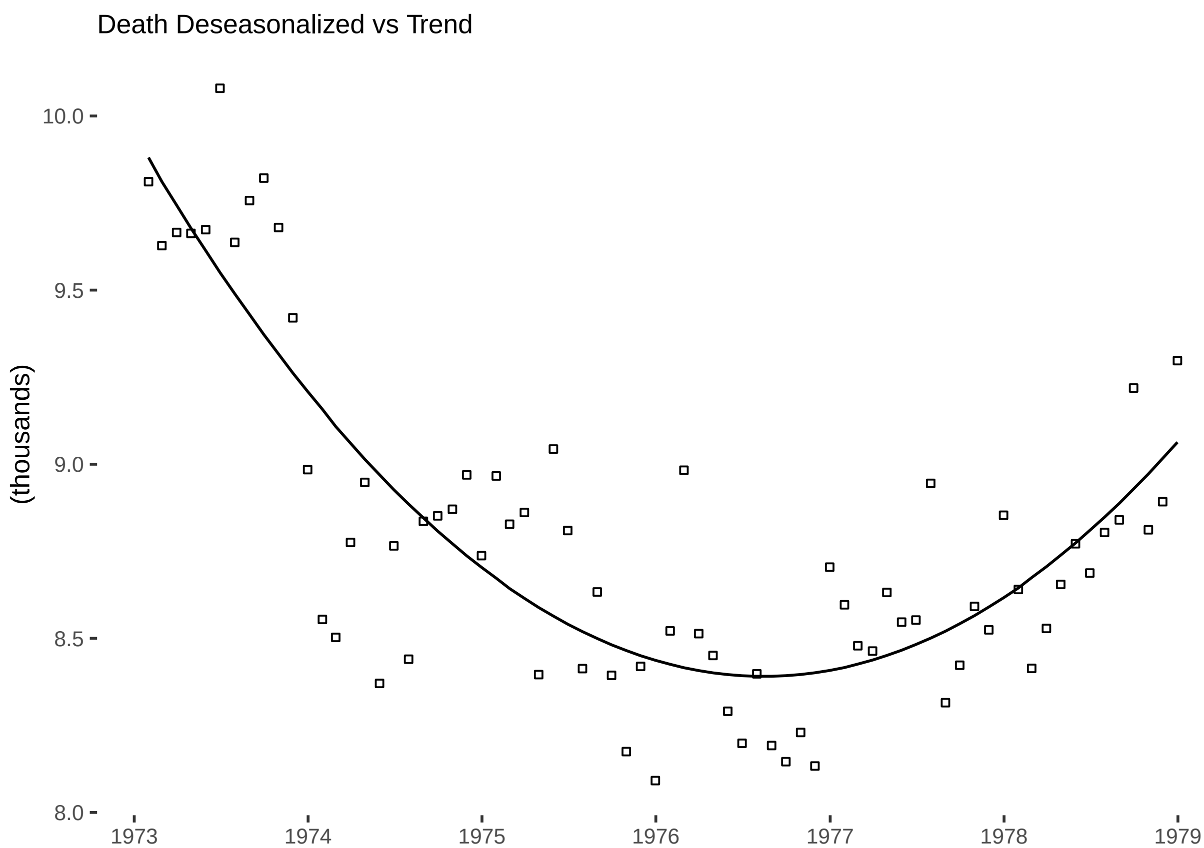
```
deaths_trend <- data.frame(
    index = 1:nrow( deaths ),
    date = deaths$date,
    value = deaths_deseasonalized_classically$deseasonalized
) %>% mutate( index2 = index ^ 2 )

deaths_trend_model <- lm( value ~ index + index2, deaths_trend )
summary( deaths_trend_model )
```

```
##
## Call:
## lm(formula = value ~ index + index2, data = deaths_trend)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -605.55 -162.47   -7.01  164.01  567.19
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) 9951.82201   92.62731  107.44  < 2e-16 ***
## index        -71.81717    5.85570  -12.26  < 2e-16 ***
```

```
## index2          0.82602    0.07774   10.63  3.6e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 254.7 on 69 degrees of freedom
## Multiple R-squared:  0.7189, Adjusted R-squared:  0.7108
## F-statistic: 88.24 on 2 and 69 DF,  p-value: < 2.2e-16
```

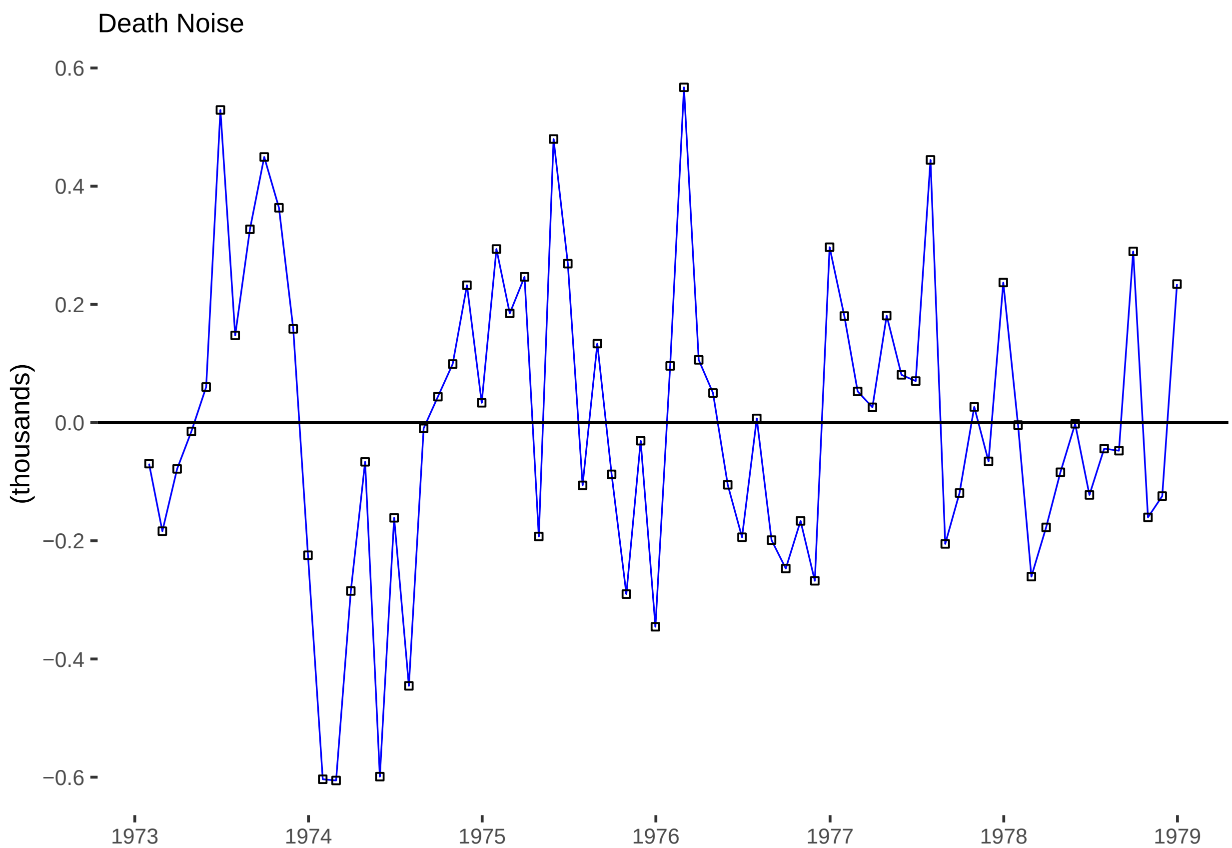
Check the fitted value of quadratic polynomial:

```
death_trend_fitted <-
    deaths_trend %>%
    mutate(
        fitted = predict( deaths_trend_model, deaths_trend )
    )
plot_data(
    death_trend_fitted %>% mutate( value = value / 1000 ),
    "Death Deseasonalized vs Trend",
    line = FALSE
) + geom_line( aes( y = fitted / 1000 ) ) + ylab("(thousands)")
```



Check the residual after seasonal and trend are removed from deaths data, this is the value of $\hat{Y}_t$ after phrase *At this point the data stored in ITSM consists of the estimated noise*:

```
death_noise <-
    data.frame(
        index = 1:nrow( deaths ),
        date = deaths$date,
        value =
            death_trend_fitted$value -
            death_trend_fitted$fitted
    )
plot_data(
    death_noise %>% mutate( value = value / 1000 ),
    "Death Noise"
) + geom_hline( yintercept = 0 ) + ylab("(thousands)")
```



1.5.2.2 Method S2: Elimination of Trend and Seasonal Components by Differencing

Figure 1-26

```
deaths_deseasonalized_by_diff <-
    backshift( deaths$value, repetition = 1, shift = 12 )
plot_data(
    data.frame(
```

```r
        index = 1:length( deaths_deseasonalized_by_diff ),
        date = deaths$date[ 1:length( deaths_deseasonalized_by_diff )],
        value = deaths_deseasonalized_by_diff / 1000
    ),
    expression( paste( "Figure 1-26 The differenced series { ", nabla[12], "..." ) )
)
```

Figure 1−26 The differenced series { ∇_{12}...

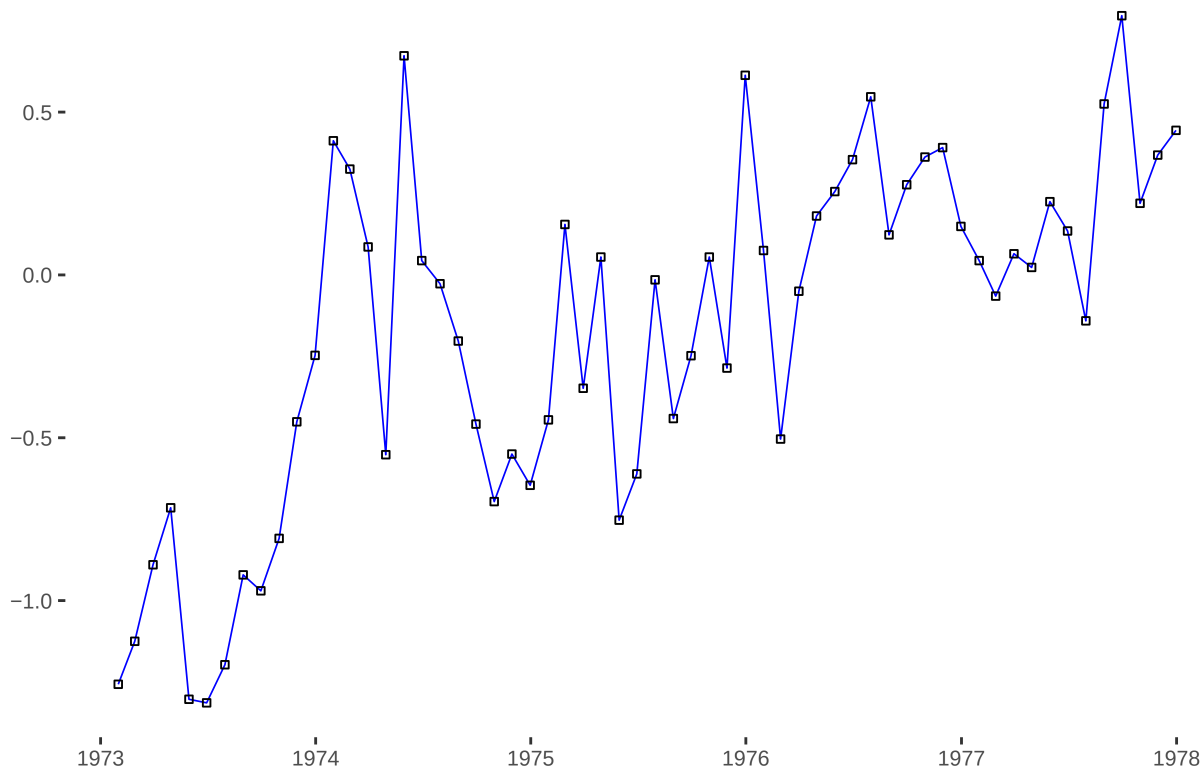
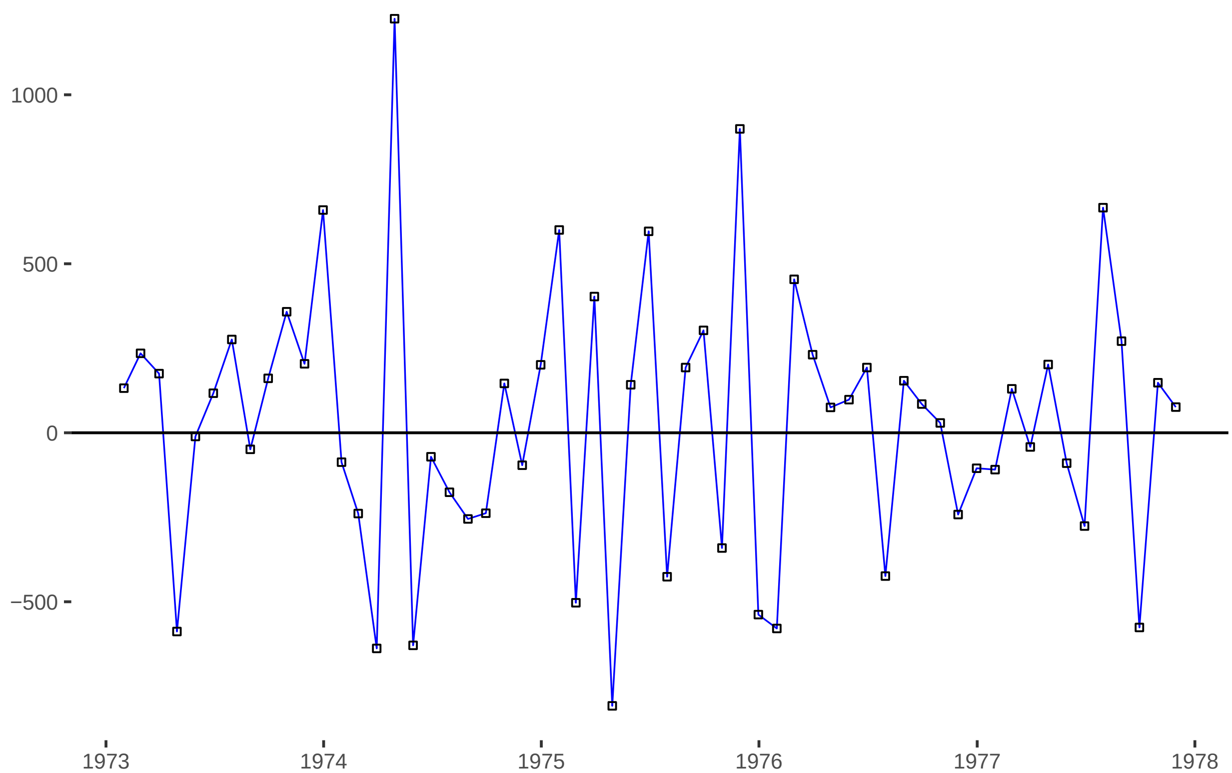


Figure 1-27

```r
deaths_detrended_by_diff <-
    backshift( deaths_deseasonalized_by_diff, repetition = 1, shift = 1 )
plot_data(
    data.frame(
        index = 1:length( deaths_detrended_by_diff ),
        date = deaths$date[1:length( deaths_detrended_by_diff )],
        value = deaths_detrended_by_diff
    ),
    expression( paste( "Figure 1-27 The differenced series { ", nabla, nabla[12], "..." ) )
) + geom_hline( yintercept = 0 )
```

Figure 1−27 The differenced series { $\nabla\nabla_{12}$...



Forecasting Stationary Time Series

Here the derivation from equation 2.5.4 and equation 2.5.5, to equation 2.5.6 and equation 2.5.7 will be given:

Explanation how to go from 2.5.4 to 2.5.6:

$$E[X_{n+h} - a_0 - \sum_{i=1}^{n} a_i X_{n+1-i}] = 0 \tag{2.5.4}$$

$$E[X_{n+h}] - E[a_0] - E[\sum_{i=1}^{n} a_i X_{n+1-i}] = 0$$

$$\mu - a_0 - \sum_{i=1}^{n} E[a_i X_{n+1-i}] = 0$$

$$\mu - a_0 - \sum_{i=1}^{n} a_i E[X_{n+1-i}] = 0$$

$$\mu - a_0 - \sum_{i=1}^{n} a_i \mu = 0$$

$$\mu - \sum_{i=1}^{n} a_i \mu = a_0$$

$$a_0 = \mu(1 - \sum_{i=1}^{n} a_i) \tag{2.5.6}$$

Now substitute a_0 in equation 2.5.5 with a_0 from equation 2.5.6, we have:

$$E[(X_{n+h} - a_0 - \sum_{i=1}^{n} a_i X_{n+1-i}) X_{n+1-j}] = 0 \tag{2.5.5}$$

$$E[(X_{n+h} - \mu(1 - \sum_{i=1}^{n} a_i) - \sum_{i=1}^{n} a_i X_{n+1-i}) X_{n+1-j}] = 0$$

$$E[(X_{n+h} - \mu + \mu \sum_{i=1}^{n} a_i - \sum_{i=1}^{n} a_i X_{n+1-i}) X_{n+1-j}] = 0$$

$$E[(X_{n+h} - \mu - \sum_{i=1}^{n} a_i (X_{n+1-i} - \mu)) X_{n+1-j}] = 0$$

$$E[X_{n+1-j}(X_{n+h} - \mu) - \sum_{i=1}^{n} a_i (X_{n+1-i} - \mu) X_{n+1-j}] = 0$$

$$E[X_{n+1-j}(X_{n+h} - \mu)] - E[\sum_{i=1}^{n} a_i (X_{n+1-i} - \mu) X_{n+1-j}] = 0$$

$$E[\sum_{i=1}^{n} a_i (X_{n+1-i} - \mu) X_{n+1-j}] = E[X_{n+1-j}(X_{n+h} - \mu)] \tag{2.5.7a}$$

Now notice that:

$$Z_1 = E[\mu \sum_{i=1}^{n} a_i (X_{n+1-i} - \mu)] \tag{2.5.7b}$$

$$Z1 = \mu \sum_{i=1}^{n} a_i (E[X_{n+1-i}] - \mu)] = \mu \sum_{i=1}^{n} a_i (\mu - \mu)] = 0$$

$$Z_2 = E[\mu(X_{n+h} - \mu)] \tag{2.5.7c}$$

$$Z_2 = \mu(E[X_{n+h}] - \mu) = \mu(\mu - \mu) = 0$$

Then combining equation 2.5.7b and 2.5.7c to 2.5.7a:

$$E[\sum_{i=1}^{n} a_i(X_{n+1-i} - \mu)X_{n+1-j}] - 0 = E[X_{n+1-j}(X_{n+h} - \mu)] - 0$$
$$E[\sum_{i=1}^{n} a_i(X_{n+1-i} - \mu)X_{n+1-j}] - Z_1 = E[X_{n+1-j}(X_{n+h} - \mu)] - Z_2$$
$$E[\sum_{i=1}^{n} a_i(X_{n+1-i} - \mu)X_{n+1-j}] - E[\mu \sum_{i=1}^{n} a_i(X_{n+1-i} - \mu)] = E[X_{n+1-j}(X_{n+h} - \mu)] - E[\mu(X_{n+h} - \mu)]$$
$$E[\sum_{i=1}^{n} a_i(X_{n+1-i} - \mu)X_{n+1-j} - \mu \sum_{i=1}^{n} a_i(X_{n+1-i} - \mu)] = E[X_{n+1-j}(X_{n+h} - \mu) - \mu(X_{n+h} - \mu)]$$
$$E[\sum_{i=1}^{n} a_i(X_{n+1-i} - \mu)X_{n+1-j} - \mu \sum_{i=1}^{n} a_i(X_{n+1-i} - \mu)] = E[X_{n+1-j}(X_{n+h} - \mu) - \mu(X_{n+h} - \mu)]$$
$$E[\sum_{i=1}^{n} (X_{n+1-i} - \mu)(X_{n+1-j} - \mu)a_i] = E[(X_{n+h} - \mu)(X_{n+1-j} - \mu)]$$
$$\vec{\gamma}_{i-j}^{T}\vec{a}_n = \gamma(h + j - 1) \tag{2.5.7d}$$

where

$$\vec{\gamma}_{i-j} = \begin{bmatrix} E[(X_{n+1-1} - \mu)(X_{n+1-j} - \mu)] \\ E[(X_{n+1-2} - \mu)(X_{n+1-j} - \mu)] \\ ... \\ E[(X_{n+1-n} - \mu)(X_{n+1-j} - \mu)] \end{bmatrix}, \vec{a}_n = \begin{bmatrix} a_1 \\ a_2 \\ ... \\ a_n \end{bmatrix}$$

and $\vec{\gamma}_{i-j}^{T}$ is the row of matrix Γ_n in which we derive:

$$\Gamma_n \vec{a} = \vec{\gamma}_n(h) \tag{2.5.7}$$

2.5.2 The Prediction Operator P(.|W)

Example 2.5.3

In this example we will demonstrate the use of **Properties of the Prediction Operator P(.|W)** for One-Step Prediction of an AR(p) series.

Let

$$U = X_{n+1} = \phi_1 X_n + \phi_2 X_{n-1} + ... + \phi_p X_{n-(p-1)} + Z_{n+1} = \sum_{i=1}^{p} \phi_i X_{n+1-i} + Z_{n+1}$$
$$\vec{W} = \begin{bmatrix} X_n \\ X_{n-1} \\ ... \\ X_{n+1-p} \end{bmatrix}$$

by **Properties of the Prediction Operator P(.|W)** point (4):

$P(U|\vec{W}) = P(\sum_{i=1}^{p} \phi_i X_{n+1-i} + Z_{n+1}|\vec{W}) = P(\sum_{i=1}^{p} \phi_i X_{n+1-i}|\vec{W}) + P(Z_{n+1}|\vec{W})$

then by **Properties of the Prediction Operator P(.|W)** point (6):

$P(Z_{n+1}|\vec{W}) = 0$

so we are left with

$P(U|\vec{W}) = P(\sum_{i=1}^{p} \phi_i X_{n+1-i}|\vec{W})$

in which by **Properties of the Prediction Operator P(.|W)** point (5):

$P(U|\vec{W}) = P(\sum_{i=1}^{p} \phi_i X_{n+1-i}|\vec{W}) = \sum_{i=1}^{p} \phi_i X_{n+1-i}$

Thus overall we have

$$P(U|\vec{W}) = P_n X_{n+1} = \phi_1 X_n + \phi_2 X_{n-1} + ... + \phi_p X_{n-(p-1)}$$

two recursive algorithm that are discussed in the lecture, the *Durbin-Levinson* and *Innovations* algorithm deal with above equation in different way.

The *Durbin-Levinson* algorithm deals with finding the coefficients of $\phi_1, ..., \phi_p$, while the *Innovations* algorithm directly find the prediction value $P_n X_{n+1}$

Durbin-Levinson Algorithm

```
dla <- function ( number_of_predictors, acvf.or.series ) {
    acvf <- NA
    if ( typeof(acvf.or.series) == "closure" ) {
        acvf <- acvf.or.series
    } else {
        acvf <- function ( h ) {
            return( sacov( acvf.or.series, h ) )
        }
    }
    result <- list()
    result$v <- list()
    result$phi <- list()
    result$v[[ "0" ]] <- acvf( 0 )
    result$phi[[ "1" ]] <- c( acvf( 1 ) / result$v[[ "0" ]] )
    for ( n in 1:number_of_predictors ) {
        if (n > 1) {
            result$phi[[ as.character( n ) ]] <- numeric( n )
            result$phi[[ as.character( n ) ]][ n ] <-
                (
                    acvf( n ) -
                        sum( sapply( 1:(n - 1), function ( j ) {
                            return(
                                result$phi[[ as.character( n - 1 ) ]][ j ] *
                                    acvf( n - j )
                            )
                        } ) )
                ) / result$v[[ as.character( n - 1 ) ]]
            result$phi[[ as.character( n ) ]][ 1:(n - 1) ] <-
                result$phi[[ as.character( n - 1 ) ]][ 1:(n - 1) ] -
                (
                    result$phi[[ as.character( n ) ]][ n ] *
```

```
                        result$phi[[ as.character( n - 1 ) ]][ (n - 1):1 ]
                    )
            }
            result$v[[ as.character( n ) ]] <-
                result$v[[ as.character( n - 1 ) ]] *
                (1 - (result$phi[[ as.character( n ) ]][ n ] ^ 2))
        }
        if ( typeof(acvf.or.series) != "closure" ) {
            result[["n+1 prediction"]] <-
                sum(
                    result$phi[[ as.character( number_of_predictors ) ]] *
                    rev( tail( acvf.or.series, number_of_predictors ) )
                )
        }
        return( result )
}
```

Innovations Algorithm

```
ia <- function ( number_of_predictors, acvf.or.series ) {
    acvf <- NA
    is.series <- FALSE
    if ( typeof(acvf.or.series) == "closure" ) {
        acvf <- acvf.or.series
    } else {
        acvf <- function ( h ) {
            return( sacov( acvf.or.series, h ) )
        }
        is.series <- TRUE
    }
    result <- list()
    result$v <- list()
    result$v[[ "0" ]] <- acvf( 0 )
    result$theta[[ "1" ]] <- c( acvf( 1 ) / result$v[[ "0" ]] )
    npo.prediction <- 0
    tail.series <- NA
    if ( is.series ) {
        tail.series <- tail( acvf.or.series, number_of_predictors )
    }
    for ( n in 1:number_of_predictors ) {
        if (n > 1) {
            result$theta[[ as.character( n ) ]] <- numeric( n )
            for ( k in 0:(n - 1) ) {
                result$theta[[ as.character( n ) ]][ n - k ] <-
                    (
                        acvf(n - k) -
                            ifelse(
                                k > 0,
                                sum( sapply( 0:(k - 1), function ( j ) {
                                    return(
                                        result$theta[[ as.character( k ) ]][ k - j ] *
                                            result$theta[[ as.character( n ) ]][ n - j ] *
```

```
                                        result$v[[ as.character( j ) ]]
                                ) } ) ),
                            0
                        )
                ) / result$v[[ as.character( k ) ]]
            }
            if ( is.series ) {
                npo.prediction <-
                    c(
                        npo.prediction,
                        sum(
                            result$theta[[ as.character( n ) ]] *
                            rev(tail.series[ 1:n ] - npo.prediction)
                        )
                    )
            }
        } else {
            if ( is.series ) {
                npo.prediction <-
                    c(
                        npo.prediction,
                        result$theta[[ as.character( n ) ]][ 1 ] *
                        tail.series[ n ]
                    )
            }
        }
        result$v[[ as.character( n ) ]] <-
            result$v[[ "0" ]] -
            sum( sapply( 0:(n - 1), function ( j ) {
                return(
                    (result$theta[[ as.character( n ) ]][ n - j ] ^ 2) *
                        result$v[[ as.character( j ) ]]
                )
            } ) )
    }
    if ( is.series ) {
        result[["n+1 prediction"]] <- tail( npo.prediction, 1 )
    }
    return( result )
}
```

Example 2.5.5

```
acvf.example.2.5.5 <- function( h ) {
    s <- 1
    theta <- -0.9
    acv <- list()
    acv[[ as.character( 0 ) ]] <- ( s ^ 2 ) * ( 1 + ( theta ^ 2 ) )
    acv[[ as.character( 1 ) ]] <- theta * ( s ^ 2 )
    result <- ifelse( any( h == 0:1 ), acv[[ as.character( h ) ]], 0 )
    return( result )
}
```

```r
# Innovation in action
ia( 4, acvf.example.2.5.5 )
```

```
## $v
## $v$`0`
## [1] 1.81
##
## $v$`1`
## [1] 1.362486
##
## $v$`2`
## [1] 1.215499
##
## $v$`3`
## [1] 1.143607
##
## $v$`4`
## [1] 1.101715
##
##
## $theta
## $theta$`1`
## [1] -0.4972376
##
## $theta$`2`
## [1] -0.6605572  0.0000000
##
## $theta$`3`
## [1] -0.7404369  0.0000000  0.0000000
##
## $theta$`4`
## [1] -0.7869838  0.0000000  0.0000000  0.0000000
```

```r
# Durbin-Levinson in action
dla( 4, acvf.example.2.5.5 )
```

```
## $v
## $v$`0`
## [1] 1.81
##
## $v$`1`
## [1] 1.362486
##
## $v$`2`
## [1] 1.215499
##
## $v$`3`
## [1] 1.143607
##
## $v$`4`
## [1] 1.101715
##
```

```
##
## $phi
## $phi$`1`
## [1] -0.4972376
##
## $phi$`2`
## [1] -0.6605572 -0.3284538
##
## $phi$`3`
## [1] -0.7404369 -0.4891009 -0.2431993
##
## $phi$`4`
## [1] -0.7869838 -0.5827118 -0.3849145 -0.1913939
```

The example above uses the theoretical auto-covariance function, we will now do simulation with the underlying MA(1) process $X_t = Z_t - 0.9Z_{t-1}$ and see if we get the coefficients close to the theoretical underlying MA(1) process and also check whether the `n+1 prediction` value from both algorithm match.

```
set.seed( 42 )
wn.example.2.5.5 <- rnorm(n = 50001,mean = 0,sd = 1)
ma.example.2.5.5 <- stats::filter( wn.example.2.5.5, filter = c( 1, -0.9 ), side = 1 )[2:50001]
ia( 4, ma.example.2.5.5 )
```

```
## $v
## $v$`0`
## [1] 1.833535
##
## $v$`1`
## [1] 1.37719
##
## $v$`2`
## [1] 1.228508
##
## $v$`3`
## [1] 1.157458
##
## $v$`4`
## [1] 1.113134
##
##
## $theta
## $theta$`1`
## [1] -0.4988866
##
## $theta$`2`
## [1] -0.662807287  0.002092469
##
## $theta$`3`
## [1] -0.741825237  0.003716711  0.001401517
##
## $theta$`4`
## [1] -0.788886254  0.001754547 -0.001865328 -0.005617684
##
```

```
##
## $`n+1 prediction`
## [1] 0.2105799
```

```r
dla( 4, ma.example.2.5.5 )
```

```
## $v
## $v$`0`
## [1] 1.833535
##
## $v$`1`
## [1] 1.37719
##
## $v$`2`
## [1] 1.228508
##
## $v$`3`
## [1] 1.157458
##
## $v$`4`
## [1] 1.113134
##
##
## $phi
## $phi$`1`
## [1] -0.4988866
##
## $phi$`2`
## [1] -0.6628073 -0.3285732
##
## $phi$`3`
## [1] -0.7418252 -0.4879705 -0.2404881
##
## $phi$`4`
## [1] -0.7888863 -0.5834612 -0.3856556 -0.1956896
##
##
## $`n+1 prediction`
## [1] 0.2105799
```

We can see that the coefficients using sampling auto-covariance are very close to the theoretical value and that the `n+1 prediction` values are match between *Durbin-Levinson* and *Innovations* algorithms.

ARMA (p,q) Processes

Definition 3.1.1

$\{X_t\}$ is an **ARMA(p, q) process** if $\{X_t\}$ is stationary and if for every t,

$$X_t - \phi_1 X_{t-1} - ... - \phi_p X_{t-p} = Z_t + \theta_1 Z_{t-1} + ... + \theta_q Z_{t-q}$$

where $\{Z_t\} \sim WN(0, \sigma^2)$ and the polynomials $(1 - \phi_1 z - ... - \phi_p z^p)$ and $1 + \theta_1 z + ... + \theta_q z^q$ have no common factors.

No further explanation for this section but one thing to keep in mind is that from the book, the *Moving Average* part of the **ARMA(**p,q**)** process definition is different in the coefficient signs compared to the course's slide.

In the the book the signs for coefficients $\theta_1, ..., \theta_q$ are all positive (+) whilst in the course's slide is negative (-).

Another thing that should be noticed is that z in either pth degree Auto-Regressive polynomial $\phi(z)$ or qth degree Moving Average polynomial $\theta(z)$ is that $z \in \mathbb{C}$ where $\mathbb{C}$ is complex numbers, thus if any of the polynomial has root of let's say $z = 3 - 4i$, than $|z| = \sqrt{3^2 + 4^2} = 5$ which is outside the unit circle where all value of $z \in \mathbb{C}$ such that $|z| = 1$.

Existence and Uniqueness

Remember as explained in Definition 3.1.1 above that $z \in \mathbb{C}$, where $\mathbb{C}$ is complex number, which will make the equation 3.1.4 in the book rewritten below makes more sense:

$$\phi(z) = 1 - \phi_1 z - ... - \phi_p z^p \neq 0 \; for \; all \; |z| = 1 \tag{3.1.4}$$

Example 3.1.1

An ARMA(1, 1) Process:

$$X_t - 0.5X_{t-1} = Z_t + 0.4Z_{t-1}, \quad \{Z_t\} \sim WN(0, \sigma^2) \tag{3.1.9}$$

Where we have the Auto-Regression and Moving Average polynomial representation:

$$\begin{aligned} \phi(z)X_t &= \theta(z)Z_t & \text{(3.1.9a)} \\ (1 - 0.5z)X_t &= (1 + 0.4z)Z_t \\ \phi(z) &= 1 - 0.5z & \text{(3.1.9b)} \\ \theta(z) &= 1 + 0.4z & \text{(3.1.9c)} \end{aligned}$$

with the root of Auto-Regressive polynomial:

$$\begin{aligned} \phi(z) &= (1 - 0.5z) = 0 \\ \iff & 1 = 0.5z \\ \iff & z = 2 \end{aligned}$$

Since the root of the Auto-Regressive polynomial is outside the unit circle, $|z| = 2 > 1$, then this ARMA(1, 1) is stationary and causal. We will then find the causal equivalent of this ARMA(1,1). Remember that the causal representation is:

$$X_t = \sum_{j=0}^{\infty} \psi_j Z_{t-j} = \psi(z)Z_t = (\sum_{j=0}^{\infty} \psi_j z^j)Z_t = (\psi_0 + \psi_1 z + \psi_2 z^2 + \psi_3 z^3 + ...)Z_t \tag{3.1.9d}$$

And since we have from 3.1.9a:

$$
\begin{aligned}
&\phi(z)X_t = \theta(z)Z_t \\
\iff &X_t = \frac{\theta(z)}{\phi(z)}Z_t
\end{aligned} \tag{3.1.9e}
$$

Thus from 3.1.9d and 3.1.9e we have:

$$
\begin{aligned}
&\frac{\theta(z)}{\phi(z)} = \psi(z) \\
&\theta(z) = \psi(z)\phi(z) \\
&\theta(z) = (\psi_0 + \psi_1 z + \psi_2 z^2 + \psi_3 z^3 + ...)\phi(z)
\end{aligned} \tag{3.1.9f}
$$

where we get the value of ψ_j by equating the coefficients of z^j for $j = 0, 1, 2, ...$ by substituting 3.1.9b and 3.1.9c to 3.1.9f, we have:

$$
\begin{aligned}
\theta(z) &= (\psi_0 + \psi_1 z + \psi_2 z^2 + \psi_3 z^3 + ...)\phi(z) \\
1 + 0.4z &= (\psi_0 + \psi_1 z + \psi_2 z^2 + \psi_3 z^3 + ...)(1 - 0.5z) \\
1 + 0.4z &= (\psi_0 + \psi_1 z + \psi_2 z^2 + \psi_3 z^3 + ...) - 0.5z(\psi_0 + \psi_1 z + \psi_2 z^2 + \psi_3 z^3 + ...) \\
1 + 0.4z &= \psi_0 + \quad \psi_1 z \ + \quad \psi_2 z^2 + \quad \psi_3 z^3 + \ ... \\
&\quad - 0.5\psi_0 z \ - 0.5\psi_1 z^2 - \ 0.5\psi_2 z^3 - \ ... \\
1 + 0.4z &= \psi_0 + (\psi_1 - 0.5\psi_0)z + (\psi_2 - 0.5\psi_1)z^2 + \sum_{j=3}^{\infty}(\psi_j - 0.5\psi_{j-1})z^j
\end{aligned}
$$

where we can find:

$$
\begin{aligned}
\psi_0 &= 1 \\
\psi_1 - 0.5\psi_0 = 0.4 \iff \psi_1 &= 0.5\psi_0 + 0.4 = 0.5(1) + 0.4 = 0.9 = 0.5^0 \times 0.9 \\
\psi_2 - 0.5\psi_1 = 0 \iff \psi_2 &= 0.5\psi_1 = 0.5^1 \times 0.9 \\
\psi_3 - 0.5\psi_2 = 0 \iff \psi_3 &= 0.5\psi_2 = 0.5^1 \times (0.5^1 \times 0.9) = 0.5^2 \times 0.9 \\
\psi_j &= 0.5^{j-1} \times 0.9, \quad j = 1, 2, 3, ...
\end{aligned}
$$

Now we check for the root of Moving Average polynomial:

$$
\begin{aligned}
&\theta(z) = (1 + 0.4z) = 0 \\
\iff &1 = -0.4z \\
\iff &z = -2.5
\end{aligned}
$$

Since the root of the Moving Average is outside the unit circle, $|z| = 2.5 > 1$, then this ARMA(1, 1) is invertible which means we can find the invertible equivalence as:

$$
Z_t = \sum_{j=0}^{\infty} \pi_j X_{t-j} = \pi(z)X_t = (\sum_{j=0}^{\infty} \pi_j z^j)X_t = (\pi_0 + \pi_1 z + \pi_2 z^2 + \pi_3 z^3 + ...)X_t \tag{3.1.9g}
$$

And since we have from 3.1.9a:

$$\begin{aligned}
&\phi(z)X_t = \theta(z)Z_t \\
\iff\ &Z_t = \frac{\phi(z)}{\theta(z)}X_t
\end{aligned} \tag{3.1.9h}$$

Thus from 3.1.9g and 3.1.9h we have:

$$\begin{aligned}
&\frac{\phi(z)}{\theta(z)} = \pi(z) \\
&\phi(z) = \pi(z)\theta(z) \\
&\phi(z) = (\pi_0 + \pi_1 z + \pi_2 z^2 + \pi_3 z^3 + ...)\theta(z)
\end{aligned} \tag{3.1.9i}$$

where we get the value of π_j by equating the coefficients of z^j for $j = 0, 1, 2, ...$ by substituting 3.1.9b and 3.1.9c to 3.1.9i, we have:

$$\begin{aligned}
\phi(z) &= (\pi_0 + \pi_1 z + \pi_2 z^2 + \pi_3 z^3 + ...)\theta(z) \\
1 - 0.5z &= (\pi_0 + \pi_1 z + \pi_2 z^2 + \pi_3 z^3 + ...)(1 + 0.4z) \\
1 - 0.5z &= \pi_0 + \quad \pi_1 z \ + \quad \pi_2 z^2 + \quad \pi_3 z^3 + \ ... \\
&\qquad\quad + 0.4\pi_0 z \ + \ 0.4\pi_1 z^2 + \ 0.4\pi_2 z^3 + \ ... \\
1 - 0.5z &= \pi_0 + (\pi_1 + 0.4\pi_0)z + (\pi_2 + 0.4\pi_1)z^2 + \sum_{j=3}^{\infty}(\pi_j + 0.4\pi_{j-1})z^j
\end{aligned}$$

where we can find:

$$\begin{aligned}
\pi_0 &= 1 \\
\pi_1 + 0.4\pi_0 = -0.5 \iff \pi_1 = -0.4\pi_0 - 0.5 = -0.4(1) - 0.5 = -0.9 &= (-0.4)^0 \times (-0.9) \\
\pi_2 + 0.4\pi_1 = 0 \iff \pi_2 = -0.4\pi_1 &= (-0.4)^1 \times (-0.9) \\
\pi_3 + 0.4\pi_2 = 0 \iff \pi_3 = -0.4\pi_2 = (-0.4)^1 \times ((-0.4)^1 \times (-0.9)) &= (-0.4)^2 \times (-0.9) \\
\pi_j &= (-0.4)^{j-1} \times (-0.9), \quad j = 1, 2, 3, ...
\end{aligned}$$

Example 3.1.2

The AR(2) process of interest:

$$\begin{aligned}
&X_t = 0.7X_{t-1} - 0.1X_{t-2} + Z_t \\
\iff\ &Z_t = X_t - 0.7X_{t-1} + 0.1X_{t-2}
\end{aligned}$$

is already in invertible form.

The corresponding Auto-Regression and Moving Average polynomials are:

$$
\begin{aligned}
X_t &= 0.7X_{t-1} - 0.1X_{t-2} + Z_t \\
X_t - 0.7X_{t-1} + 0.1X_{t-2} &= Z_t \\
(1 - 0.7z + 0.1z^2)X_t &= (1)Z_t \\
\phi(z) &= 1 - 0.7z + 0.1z^2 \\
\theta(z) &= 1
\end{aligned}
$$

The root of Auto-Regressive polynomial:

$$
\begin{aligned}
\phi(z) &= 1 - 0.7z + 0.1z^2 \\
\phi(z) &= (1 - 0.2z)(1 - 0.5z) \\
z_1 &= 2 \\
z_2 &= 5
\end{aligned}
$$

Since the root of Auto-Regression are both outide the unit circle, then the AR(2) process is stationary and causal, where we then implement the same technique of equating coefficient of z^j from equation 3.1.9f substituting the Auto-Regression and Moving Average polynomials accordingly:

$$
\begin{aligned}
\theta(z) &= (\psi_0 + \psi_1 z + \psi_2 z^2 + \psi_3 z^3 + ...)\phi(z) \\
1 &= (\psi_0 + \psi_1 z + \psi_2 z^2 + \psi_3 z^3 + ...)(1 - 0.7z + 0.1z^2) \\
1 &= \psi_0 + \quad \psi_1 z \ + \quad \psi_2 z^2 \ + \quad \psi_3 z^3 \ + \quad \psi_4 z^4 \ + ... \\
& \quad - \ 0.7\psi_0 z \ - \ 0.7\psi_1 z^2 \ - \ 0.7\psi_2 z^3 \ - \ 0.7\psi_3 z^4 \ - ... \\
& \qquad\qquad + \ 0.1\psi_0 z^2 \ + \ 0.1\psi_1 z^3 \ + \ 0.1\psi_2 z^4 \ + ... \\
1 &= \psi_0 + (\psi_1 - 0.7\psi_0)z + (\psi_2 - 0.7\psi_1 + 0.1\psi_0)z^2 + (\psi_3 - 0.7\psi_2 + 0.1\psi_1)z^3 + (\psi_4 - 0.7\psi_3 + 0.1\psi_2)z^4 + ...
\end{aligned}
$$

where we have

$$
\begin{aligned}
\psi_0 &= 1 \\
\psi_1 - 0.7\psi_0 = 0 \iff \psi_1 = 0.7\psi_0 \iff \psi_1 &= 0.7 \times 1 = 0.7 \\
\psi_j - 0.7\psi_{j-1} + 0.1\psi_{j-2} = 0 \iff \psi_j &= 0.7\psi_{j-1} - 0.1\psi_{j-2}, \quad j = 2, 3, ...
\end{aligned}
$$

Example 3.1.3

As what has been explained in Definition 3.1.1, that the polynomial root $z \in \mathbb{C}$, thus the AR polynomial $\phi(z)$ roots magnitude $|z|$ with roots $z = 2(1 \pm i\sqrt{3})/3$ is:

$$
|z| = \sqrt{(\frac{2}{3})^2 + (\frac{2\sqrt{3}}{3})^2} = \sqrt{\frac{4}{9} + \frac{12}{9}} = \sqrt{\frac{16}{9}} = \frac{4}{3} > 1
$$

which means the Auto-Regressive polynomial roots lie outside the unit circle.

3.2.1 Calculation of the ACVF

Recall that causality implies that:

$$X_t = \sum_{j=0}^{\infty} \psi_j Z_{t-j}, \quad \{Z_t\} \sim WN(0, \sigma^2) \tag{3.2.2}$$

Then, the Auto-Covariance of lag h will be:

$$\begin{aligned}
\gamma(h) &= Cov(X_{t+h}, X_t) \\
\gamma(h) &= Cov(\sum_{i=0}^{\infty} \psi_i Z_{t+h-i}, \sum_{j=0}^{\infty} \psi_j Z_{t-j}), \quad \textit{by substituting } 3.2.2 \\
\gamma(h) &= Cov(\sum_{(j+h)=0}^{\infty} \psi_{j+h} Z_{t+h-(j+h)}, \sum_{j=0}^{\infty} \psi_j Z_{t-j}), \ \textit{by setting } i = j+h \\
\gamma(h) &= Cov(\sum_{j=-h}^{\infty} \psi_{j+h} Z_{t-j}, \sum_{j=0}^{\infty} \psi_j Z_{t-j}) \qquad (3.2.3a)
\end{aligned}$$

By covariance linearity and uncorrelation of white noises $Cov(Z_m, Z_n) = 0, \ m \neq n$ we have:

$$\begin{aligned}
\gamma(h) &= \sum_{j=0}^{\infty} Cov(\psi_j Z_{t-j}, \psi_{j+h} Z_{t-j}) \\
\gamma(h) &= \sum_{j=0}^{\infty} \psi_j \psi_{j+h} Cov(Z_{t-j}, Z_{t-j})
\end{aligned}$$

Since covariance of random variable with itself is its variance, we have:

$$\begin{aligned}
\gamma(h) &= \sum_{j=0}^{\infty} \psi_j \psi_{j+h} Var(Z_{t-j}) \\
\gamma(h) &= \sum_{j=0}^{\infty} \psi_j \psi_{j+h} \sigma^2 \\
\gamma(h) &= \sigma^2 \sum_{j=0}^{\infty} \psi_j \psi_{j+h} \qquad (3.2.3)
\end{aligned}$$

Example 3.2.1

We need to find the causal process by equating coefficients of z^j just like Example 3.1.1 and Example 3.1.2 of the ARMA(1, 1) process below:

$$\begin{aligned}
X_t - \phi X_{t-1} &= Z_t + \theta Z_{t-1}, \quad \{Z_t\} \sim WN(0, \sigma^2) \qquad (3.2.4) \\
(1 - \phi z) X_t &= (1 + \theta) Z_t \\
\phi(z) &= 1 - \phi z \\
\theta(z) &= 1 + \theta z
\end{aligned}$$

Recalling equation 3.1.9f, then substituting the Auto-Regression and Moving Average polynomials above, we have:

$$\begin{aligned}
\theta(z) =&(\psi_0 + \psi_1 z + \psi_2 z^2 + \psi_3 z^3 + ...)\phi(z)\\
1+\theta z =&(\psi_0 + \psi_1 z + \psi_2 z^2 + \psi_3 z^3 + ...)(1-\phi z)\\
1+\theta z =&\ \psi_0 + \quad \psi_1 z + \quad \psi_2 z^2 + \quad \psi_3 z^3 + ...\\
&\quad - \quad \phi\psi_0 z \ - \quad \phi\psi_1 z^2 - \quad \phi\psi_2 z^3 - ...\\
1+\theta z =&\ \psi_0 + (\psi_1 - \phi\psi_0)z + (\psi_2 - \phi\psi_1)z^2 + (\psi_3 - \phi\psi_2)z^3 + ...
\end{aligned}$$

where we have:

$$\psi_0 = 1 \tag{3.2.4a}$$

$$\begin{aligned}
\psi_1 - \phi\psi_0 = \psi_1 - \phi \times 1 = \psi_1 - \phi = \theta \iff& \psi_1 = \phi + \theta = \phi^0(\phi+\theta)\\
\psi_2 - \phi\psi_1 = \psi_2 - \phi(\phi+\theta) = 0 \iff& \psi_2 = \phi(\phi+\theta) = \phi^1(\phi+\theta)\\
\psi_3 - \phi\psi_2 = \psi_3 - \phi(\phi(\phi+\theta)) = 0 \iff& \psi_3 = \phi^2(\phi+\theta)
\end{aligned}$$

$$\psi_j = \phi^{j-1}(\phi+\theta),\ j = 1,2,3,... \tag{3.2.4b}$$

Substituting equation 3.2.4a and 3.2.4b to equation 3.2.3, we have:

$$\begin{aligned}
\gamma(h) &= \sigma^2 \sum_{j=0}^{\infty} \psi_j \psi_{j+h} = \sigma^2[\psi_0\psi_{0+h} + \sum_{j=1}^{\infty} \psi_j\psi_{j+h}]\\
\gamma(h) &= \sigma^2[\psi_0\psi_{0+h} + \sum_{j=1}^{\infty} \phi^{j-1}(\phi+\theta)\phi^{j+h-1}(\phi+\theta)]\\
\gamma(h) &= \sigma^2[(1)\psi_h + \sum_{j=1}^{\infty}(\phi+\theta)^2\phi^h\phi^{2(j-1)}]\\
\gamma(h) &= \sigma^2[\psi_h + (\phi+\theta)^2\phi^h \sum_{j=1}^{\infty}\phi^{2(j-1)}]\\
\gamma(h) &= \sigma^2[\psi_h + (\phi+\theta)^2\phi^h(1+\phi^2+\phi^4+\phi^6+...)]\\
\gamma(h) &= \sigma^2[\psi_h + (\phi+\theta)^2\phi^h \lim_{n\to\infty}\frac{1-\phi^{2n}}{1-\phi^2}]\\
\gamma(h) &= \sigma^2[\psi_h + (\phi+\theta)^2\phi^h \frac{1-\lim_{n\to\infty}\phi^{2n}}{1-\phi^2}],\ \mathit{since}\ |\phi|<1\ \mathit{then}\ \lim_{n\to\infty}\phi^{2n} = 0,\ \mathit{thus}\\
\gamma(h) &= \sigma^2[\psi_h + (\phi+\theta)^2\phi^h \frac{1}{1-\phi^2}]\\
\gamma(h) &= \sigma^2[\psi_h + \frac{(\phi+\theta)^2\phi^h}{1-\phi^2}]
\end{aligned} \tag{3.2.4c}$$

Where we have for $h = 0$:

$$\gamma(0) = \sigma^2[\psi_0 + \frac{(\phi+\theta)^2\phi^0}{1-\phi^2}] = \sigma^2[1 + \frac{(\phi+\theta)^2}{1-\phi^2}]$$

and for $h > 0$, substituting equation 3.2.4b into equation 3.2.4c:

$$\gamma(h) = \sigma^2[\phi^{h-1}(\phi+\theta) + \frac{(\phi+\theta)^2\phi^h}{1-\phi^2}] = \phi^{h-1}\sigma^2[(\phi+\theta) + \frac{(\phi+\theta)^2\phi}{1-\phi^2}] \tag{3.2.4d}$$

$$\gamma(1) = \sigma^2[\phi^{1-1}(\phi+\theta) + \frac{(\phi+\theta)^2\phi^1}{1-\phi^2}] = \sigma^2[(\phi+\theta) + \frac{(\phi+\theta)^2\phi}{1-\phi^2}] \tag{3.2.4e}$$

combining equation 3.2.4d and 3.2.4e, we have:

$$\gamma(h) = \phi^{h-1}\gamma(1),\ for\ h \geq 2$$

Example 3.2.2

For process MA(q) process:

$$X_t = Z_t + \theta_1 Z_{t-1} + ... + \theta_q Z_{t-q}, \quad \{Z_t\} \sim WN(0, \sigma^2)$$

Equation 3.2.3 immediately gives the result

$$\gamma(h) = \begin{cases} \sigma^2 \sum_{j=0}^{q-|h|} \theta_j \theta_{j+|h|},\ if\ |h| \leq q \\ 0,\ if\ |h| > q \end{cases}$$

where θ_0 is defined to be 1.

Since Auto-Correlation $\rho(h) = \frac{\gamma(h)}{\gamma(0)}$, the MA(q) process Auto-Correlation has the distinctive feature of vanishing at lags greater than q.

Example 3.2.6

The causal AR(p) defined by:

$$X_t - \phi_1 X_{t-1} - ... - \phi_p X_{t-p} = Z_t, \quad \{Z_t\} \sim WN(0, \sigma^2)$$

as we know from Example 2.5.3 that for $h \geq p$ the best linear predictor of X_{h+1} in terms of $X_1, ..., X_h$ is

$$\hat{X}_{h+1} = \phi_1 X_h + \phi_2 X_{h-1} + ... + \phi_p X_{h+1-p}$$

Since the coefficient of ϕ_{hh} of X_1 is ϕ_p if $h = p$ and 0 if $h > p$, we conclude that the PACF $\alpha(.)$ of the process $\{X_t\}$ has the properties

$$\alpha(p) = \phi_p$$

and

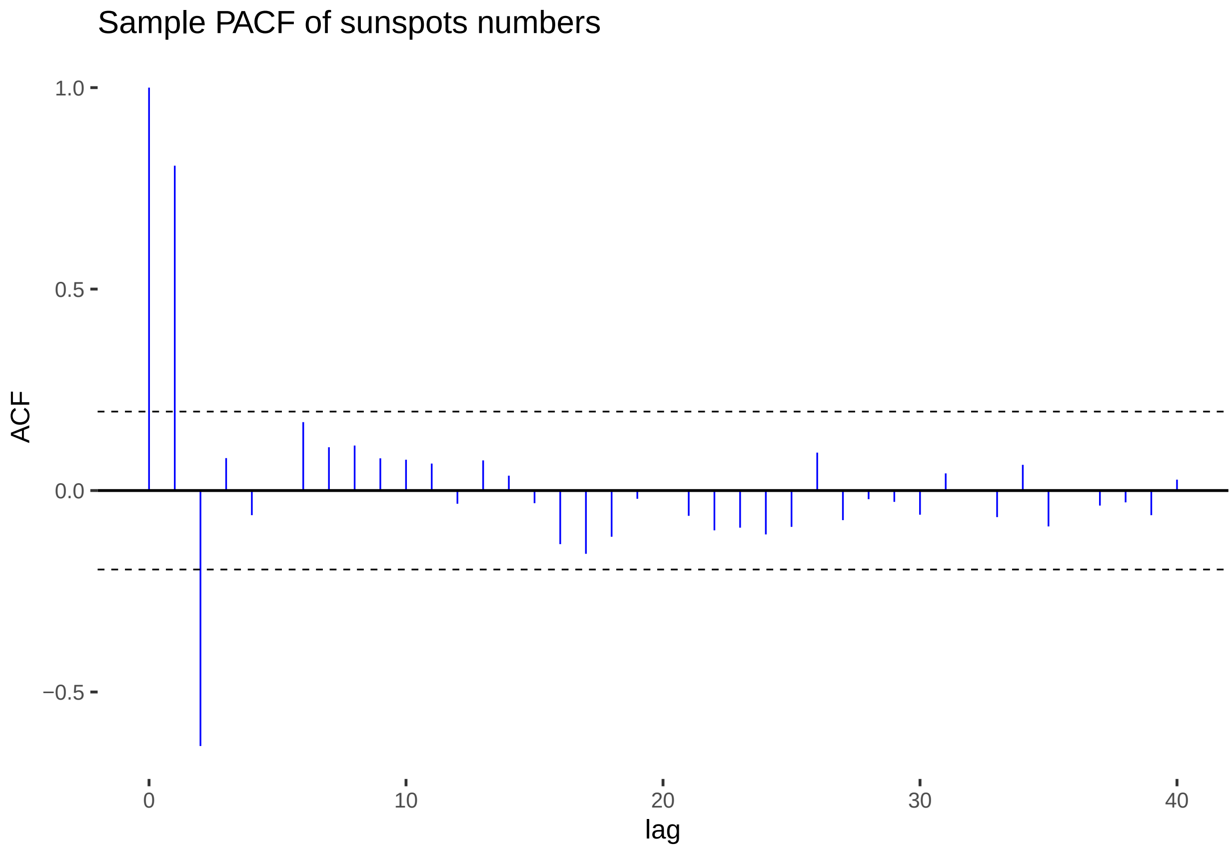
$$\alpha(h) = 0\ for\ h > p$$

For $h < p$, recall that the coefficient ϕ_{hh} of X_1 above is the last coefficient of *Durbin-Levinson Algorithm* which we will show in the way we build Figure 3-7.

Figure 3-7

Notice that the function `plot_partial_acor` below calls *Durbin-Levinson* `dla` function to calculate the partial auto-correlation values.

```
plot_partial_acor <- function ( series, title, ci = 0.95, max_lag = 40 ) {
    sacor_index <- 0:min( length( series ), max_lag )
    max_plot_lag <- max( sacor_index )
    data_sacor <- data.frame(
        index = sacor_index
    )
    data_sacor$value <- c(
        1,
        sapply(
            dla( max_plot_lag, series )$phi,
            function ( dl.coefficients ) {
                return( dl.coefficients[length(dl.coefficients)])
            }
        )
    )
    ci_line <- qnorm( ( ( 1 - ci ) / 2 ) + c( 0, ci ) ) / sqrt( length( series ) )
    data_acor <-
        data.frame(
            lag = rep( data_sacor$index, 2 ),
            acor = c( rep( 0, max_plot_lag + 1 ), data_sacor$value )
        )
    data_plot <-
        ggplot( data_acor , aes( lag, acor ) ) +
        ggtitle( title ) +
        ylab("ACF") +
        geom_line(
            aes( group = lag ),
            size = 0.3,
            color = "blue"
        ) +
        geom_hline(
            yintercept = 0,
            color = "black"
        ) +
        geom_hline(
            yintercept = ci_line,
            linetype = "dashed",
            color = "black",
            size = 0.3
        ) +
        theme_tufte() +
        theme( text = element_text( family = "sans", size = 10 ) )
    return( data_plot )
}

plot_partial_acor(
    series = itsmr::Sunspots,
    title = "Sample PACF of sunspots numbers"
)
```



5.1.1 Yule-Walker Estimation

For causal AR(p) defined by:

$$\begin{aligned} X_t - \phi_1 X_{t-1} - ... - \phi_p X_{t-p} =& Z_t, \quad \{Z_t\} \sim WN(0, \sigma^2) \\ & Z_t \ is \ uncorrelated \ with \ X_s \ for \ each \ s < t \\ (1 - \phi_1 z - \phi_2 z^2 - ... - \phi_p z^p) X_t =& Z_t \\ \phi(z) =& 1 - \phi_1 z - \phi_2 z^2 - ... - \phi_p z^p, \ auto-regressive \\ \theta(z) =& 1, \ moving \ average \end{aligned} \tag{5.5.1a}$$

with equating $z^j, \ j = 1, 2, 3, ...$ coefficients technique, substituting above polynomial to equation 3.1.9f, we have:

$$\begin{aligned} 1 &= (\psi_0 + \psi_1 z + \psi_2 z^2 + \psi_3 z^3 + ...)(1 - \phi_1 z - \phi_2 z^2 - ... - \phi_p z^p) \\ \implies \psi_0 &= 1 \end{aligned}$$

By causality we have:

$$
\begin{aligned}
X_t &= \sum_{j=0}^{\infty} \psi_j Z_{t-j} \\
X_t &= \psi_0 Z_t + \sum_{j=1}^{\infty} \psi_j Z_{t-j} \\
X_t &= (1) Z_t + \sum_{j=1}^{\infty} \psi_j Z_{t-j} \\
X_t &= Z_t + \sum_{j=1}^{\infty} \psi_j Z_{t-j}
\end{aligned} \tag{5.1.1b}
$$

where we have:

$$
E[X_t] = E[\sum_{j=0}^{\infty} \psi_j Z_{t-j}] = \sum_{j=0}^{\infty} \psi_j E[Z_{t-j}] = \sum_{j=0}^{\infty} \psi_j (0) = 0
$$

Then by definition of covariance we have:

$$
\begin{aligned}
&\gamma(h) = Cov(X_t, X_{t-h}) = E[(X_t - E[X_t])(X_{t-h} - E[X_{t-h}])] \\
&\gamma(h) = E[(X_t - 0)(X_{t-h} - 0)] \\
&\gamma(h) = E[X_t X_{t-h}]
\end{aligned} \tag{5.1.1c}
$$

If we multiply both sides of equation 5.1.1a with $X_{t-j},\ j = 1, 2, 3, ...$ then take the expectation, we have:

$$
\begin{aligned}
E[(X_t - \phi_1 X_{t-1} - ... - \phi_p X_{t-p}) X_{t-j}] =& E[Z_t X_{t-j}] \\
E[X_t X_{t-j} - \phi_1 X_{t-1} X_{t-j} - ... - \phi_p X_{t-p} X_{t-j}] =& E[Z_t X_{t-j}]
\end{aligned} \tag{5.1.1d}
$$

since Z_t is uncorrelated with X_s for each s < t, the right hand side of equation 5.1.1d $E[Z_t X_{t-j}] = 0$ thus we have:

$$
\begin{aligned}
E[(X_t - \phi_1 X_{t-1} - ... - \phi_p X_{t-p}) X_{t-j}] =& E[Z_t X_{t-j}] \\
E[X_t X_{t-j} - \phi_1 X_{t-1} X_{t-j} - ... - \phi_p X_{t-p} X_{t-j}] =& 0 \\
\phi_1 X_{t-1} X_{t-j} + ... + \phi_p X_{t-p} X_{t-j}] =& E[X_t X_{t-j}] \\
\textit{substituting equation } 5.1.1c: \\
\phi_1 \gamma(j-1) + \phi_2 \gamma(j-2) + ... + \phi_p \gamma(j-p) =& \gamma(j) \\
\begin{bmatrix} \gamma(j-1) & \gamma(j-2) & ... & \gamma(j-p) \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ ... \\ \phi_p \end{bmatrix} =& \gamma(j)
\end{aligned} \tag{5.1.1e}
$$

If we think equation 5.1.1e as row of matrix multiplied by AR(p) coefficients we get for $j = 1, 2, 3...$

$$\begin{bmatrix}\gamma(1-1) & \gamma(1-2) & ... & \gamma(1-p) \\ \gamma(2-1) & \gamma(2-2) & ... & \gamma(2-p) \\ & & ... & \\ \gamma(p-1) & \gamma(p-2) & ... & \gamma(p-p)\end{bmatrix}\begin{bmatrix}\phi_1 \\ \phi_2 \\ ... \\ \phi_p\end{bmatrix} = \begin{bmatrix}\gamma(1) \\ \gamma(2) \\ ... \\ \gamma(p)\end{bmatrix}$$

$$\Gamma_p\vec{\phi} = \vec{\gamma}_p \tag{5.1.3}$$

Now if we multiply equation 5.1.1a with equation 5.1.1b for the same side respectively, and take the expectation we have:

$$\begin{aligned}
E[(X_t - \phi_1 X_{t-1} - ... - \phi_p X_{t-p})X_t] =& E[Z_t(Z_t + \sum_{j=1}^{\infty}\psi_j Z_{t-j})] \\
E[X_tX_t - \phi_1 X_tX_{t-1} - ... - \phi_p X_tX_{t-p}] =& E[(Z_tZ_t + \sum_{j=1}^{\infty}\psi_j Z_tZ_{t-j})] \\
E[X_tX_t] - \phi_1 E[X_tX_{t-1}] - ... - \phi_p E[X_tX_{t-p}] =& E[Z_tZ_t] + \sum_{j=1}^{\infty}\psi_j E[Z_tZ_{t-j}] \\
\textit{substituting equation } 5.1.1c: & \\
\gamma(0) - \phi_1\gamma(1) - ... - \phi_p\gamma(p) =& \sigma^2 + \sum_{j=1}^{\infty}\psi_j(0) \\
\gamma(0) - \begin{bmatrix}\phi_1 & \phi_2 & ... & \phi_p\end{bmatrix}\begin{bmatrix}\gamma(1) \\ \gamma(2) \\ ... \\ \gamma(p)\end{bmatrix} =& \sigma^2
\end{aligned}$$

$$\gamma(0) - \vec{\phi}^{\,T}\vec{\gamma}_p = \sigma^2 \tag{5.1.4}$$

Large-Sample Distribution of Yule-Walker Estimators

For a large sample from an AR(p) process,

$$\hat{\vec{\phi}} \approx N(\vec{\phi}, \frac{\sigma^2}{n}\Gamma_p^{-1})$$

If we replace σ^2 and Γ_p by their estimates $\hat{\sigma^2}$ and $\hat{\Gamma}_p$ we get:

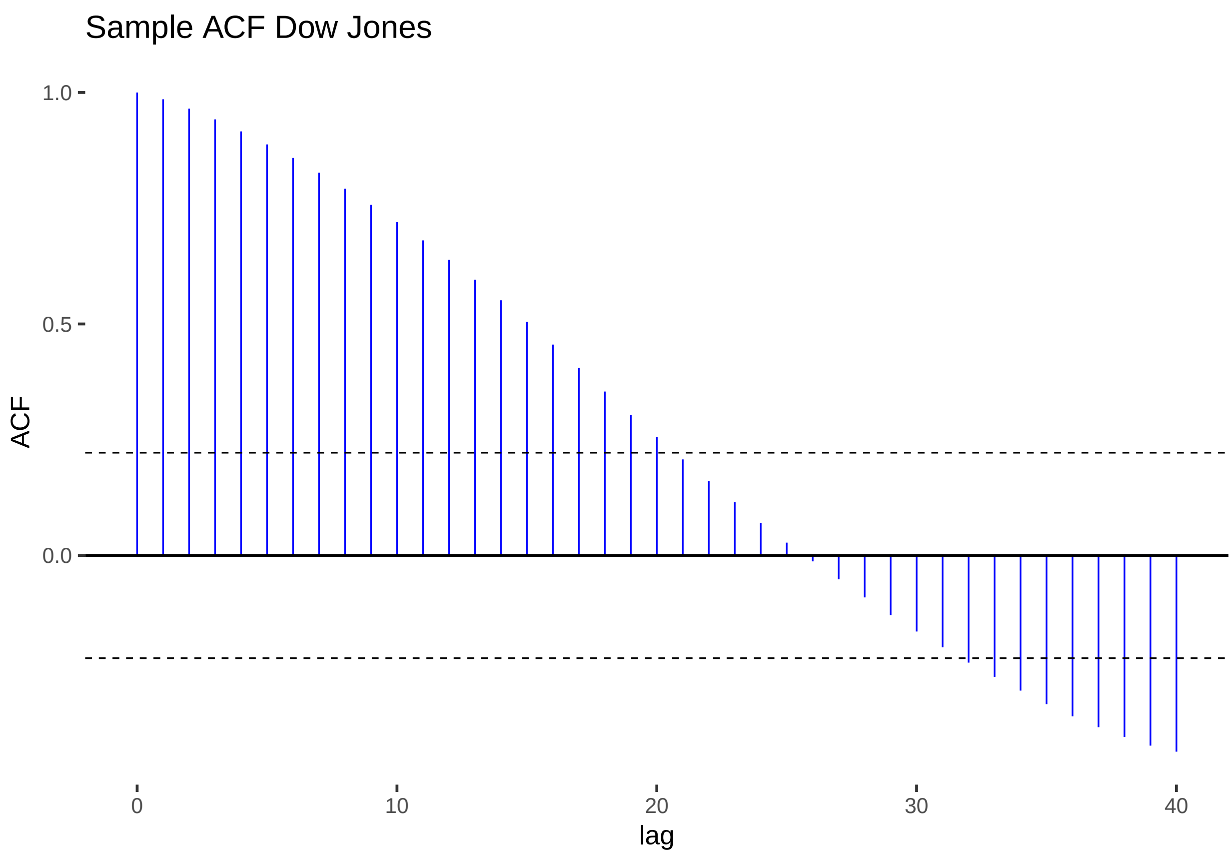
$$\hat{\phi}_{pj} \pm \Phi_{1-\alpha/2}\frac{\hat{v}_{jj}^{1/2}}{n} \tag{5.1.13}$$

where $\hat{v}_{jj}$ is the j$^{\text{th}}$ diagonal element of $\hat{v}_p\hat{\Gamma}_p^{-1}$, contains ϕ_{pj} with probability close to $1-\alpha$

Example 5.1.1

We are analyzing Dow Jones from `itsmr` package.

```r
plot_acor(series = itsmr::dowj,title = "Sample ACF Dow Jones")
```



we can see indication of trend, let's difference the series to remove the trend and check the result:

```r
dowjones.diff <- backshift(itsmr::dowj,1,1)
```

Sample auto-covariance until lag 2:

```r
sacov(dowjones.diff, 0:2)
```

```
## [1] 0.17991924 0.07590408 0.04885042
```

Applying Durbin-Levinson algorithm of order 2:

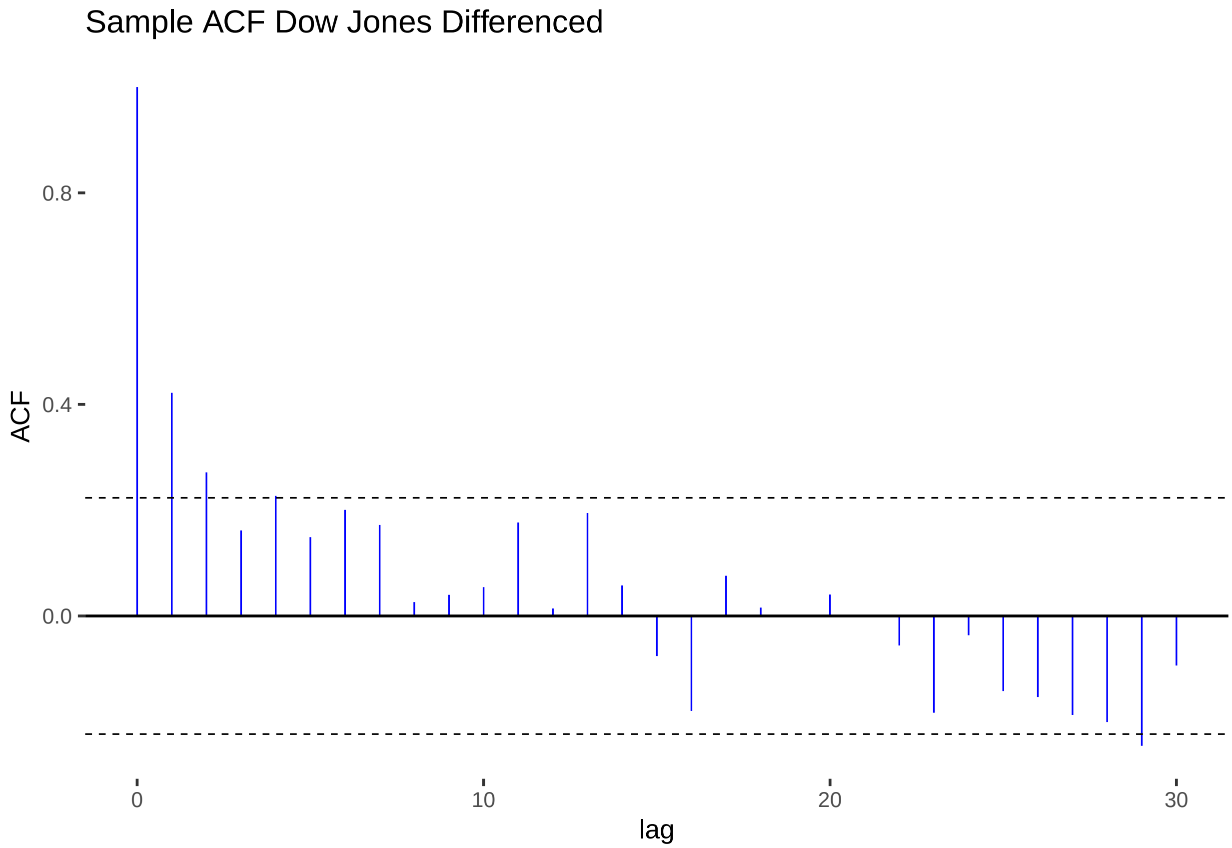
```r
dla.example.5.1.1 <- dla( 2, dowjones.diff )
dla.example.5.1.1
```

```
## $v
## $v$`0`
## [1] 0.1799192
##
## $v$`1`
## [1] 0.1478969
```

```
##
## $v$`2`
## [1] 0.1459822
##
##
## $phi
## $phi$`1`
## [1] 0.4218786
##
## $phi$`2`
## [1] 0.3738761 0.1137827
##
##
## $`n+1 prediction`
## [1] -0.298125
```

Figure 5-1

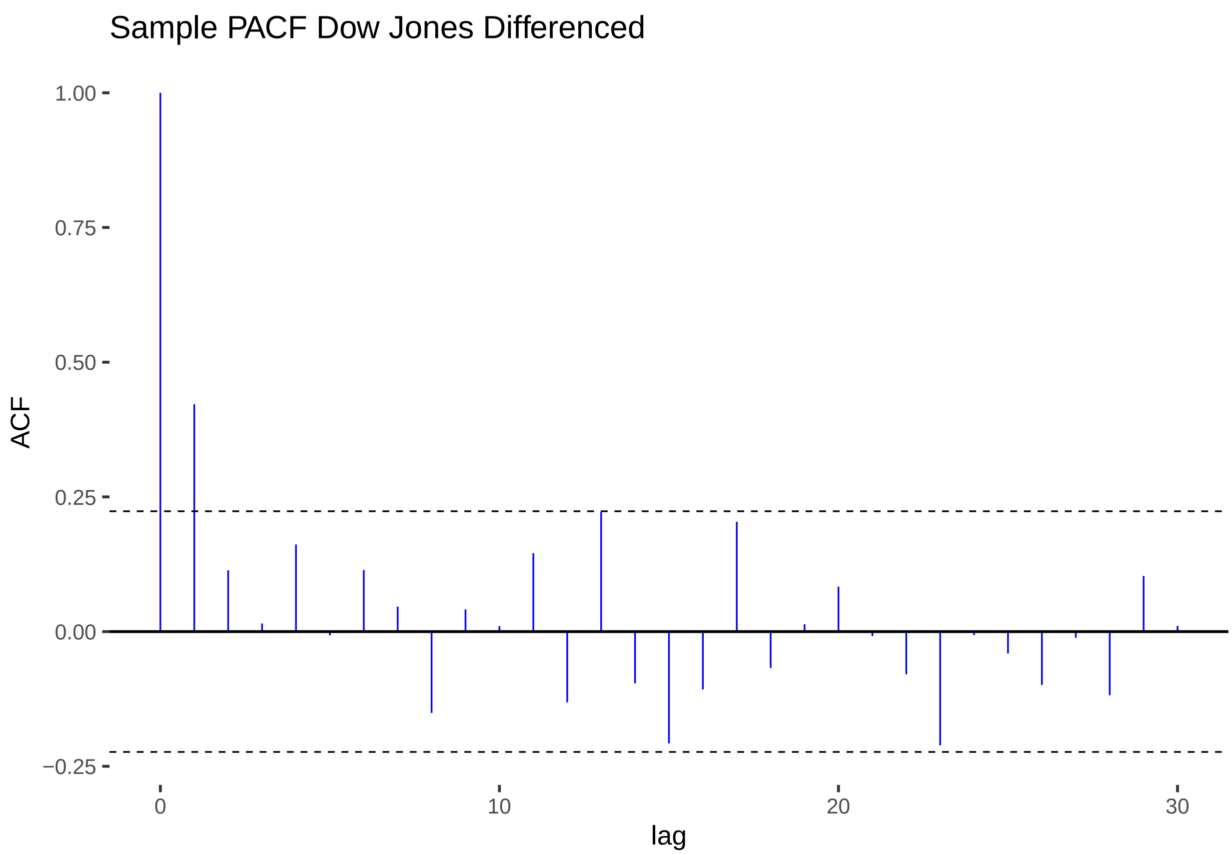
```
plot_acor(
  series = dowjones.diff,
  title = "Sample ACF Dow Jones Differenced",
  max_lag = 30)
```



From ACF plot, the trend has been removed.

Figure 5-2

```
plot_partial_acor(
  series = dowjones.diff,
  title = "Sample PACF Dow Jones Differenced",
  max_lag = 30)
```



PACF plot suggest an AR(1) model.

Centralize the mean for differenced data $X_t = Y_t - 0.1336$:

```
dowjones.diff.centralized <- dowjones.diff - mean( dowjones.diff )
```

Get the estimated coefficient using Durbin-Levinson algorithm:

```
dla.example.5.1.1.centralized <- dla(1,dowjones.diff.centralized)
dla.example.5.1.1.centralized
```

```
## $v
## $v$`0`
## [1] 0.1799192
##
## $v$`1`
## [1] 0.1478969
##
```

```
##
## $phi
## $phi$`1`
## [1] 0.4218786
##
##
## $`n+1 prediction`
## [1] -0.3812248
```

We can see that the first coefficient estimate for AR(1) both are the same for differenced data, either having the mean centralized or not:

```
dla.example.5.1.1$phi[[1]]
```

```
## [1] 0.4218786
```

```
dla.example.5.1.1.centralized$phi[[1]]
```

```
## [1] 0.4218786
```

Following equation 5.1.13, we need to find $\hat{v}_{jj}$ for $j = 1$, v_{11}:

```
v.1 <- dla.example.5.1.1.centralized$v[["1"]]
v.1
```

```
## [1] 0.1478969
```

```
gamma.matrix.1.inverse <- sacov(dowjones.diff.centralized,0)
gamma.matrix.1.inverse
```

```
## [1] 0.1799192
```

```
v.11 <- v.1/ gamma.matrix.1.inverse
v.11
```

```
## [1] 0.8220184
```

Notice that in the code chunk above since $p = 1$ the inverse of matrix Γ_p is just its reciprocal $\Gamma_1^{-1} = \frac{1}{\Gamma_1}$

The 95% confidence interval for the AR(1) coefficient is then:

```
dla.example.5.1.1.centralized$phi[[1]] +
    ( c(-1,1) * abs(qnorm((1-0.95)/2)) * sqrt(v.11) / sqrt(length(dowjones.diff.centralized)) )
```

```
## [1] 0.2193700 0.6243872
```

Notice that the difference of number results are caused by the book rounding to 4 decimal digits.

Example 5.1.5

In Example 5.1.1, Figure 5-1 suggests that MA(2) model might also provide a good fit for the data.

Showing the coefficient of MA(2) from innovation algorithm of order 17:

```r
ia.example.5.1.5 <- ia( 17, dowjones.diff.centralized )
ia.example.5.1.5$theta[["17"]][1:2]
```

```
## [1] 0.4269274 0.2703894
```

The whole MA(17) coefficients:

```r
ia.example.5.1.5$theta[["17"]]
```

```
##  [1]  0.426927405  0.270389381  0.118292048  0.158903846  0.135491764
##  [6]  0.156813594  0.128359762 -0.005994138  0.014755083 -0.001652658
## [11]  0.197386903 -0.046330757  0.202282155  0.128519260 -0.021317802
## [16] -0.257511184  0.075988619
```

Innovations Algorithm Estimates when p > 0 and q > 0

Here is an example of function to get parameter estimation of an ARMA(p, q) process with p > 0 and q > 0:

```r
ia.arma <- function ( number_of_predictors, series, p, q ) {
    ia.coefficients <-
        ia( number_of_predictors, series )$theta[[
            as.character( number_of_predictors )
        ]]
    theta.vector <- as.matrix( ia.coefficients[(q+1):(q+p)] )
    theta.matrix <-  t(as.matrix(
        sapply(q:(q+p-1),function(i){
            return(sapply( i:(i+1-p), function ( j ){
                return(ifelse(j>0,ia.coefficients[j],0))
            } ) )
        } ) ) )
    ar.vector <- as.vector(solve(theta.matrix) %*% theta.vector)
    ma.vector <- sapply( 1:q, function ( j ) {
        return(ia.coefficients[ j ] - sum(sapply( 1:min(j,p), function ( i ){
            return(
                ar.vector[ i ] *
                ifelse( j -i >=0,
                        ifelse(
                            j - i == 0, 1,
                            ia.coefficients[j-i])
                        , 0 ) )
        } ) ) )
    } )
    return( list(
        ar = ar.vector,
        ma = ma.vector
    ) )
}
```

Example 5.1.6

The coefficients of ARMA(1, 1) model using the innovations methods as in the example of the book are given by:

```
ia.arma( 17, itsmr::lake, 1, 1 )
```

```
## $ar
## [1] 0.7234365
##
## $ma
## [1] 0.3596418
```

pay attention to the consensus of the coefficient sign that the book used in Definition 3.1.1, the ARMA(1, 1) model is then given by:

$$X_t - 0.7234365X_{t-1} = Z_t + 0.3596418Z_{t-1}$$

where in the book the coefficients are rounded up to 4 decimal points.

7.1 Historical Overview

The explanation has been discussed in Module 3 - Lesson 1 - Slide 7 and Module 3 - Lesson 2 - Slide 3.

7.2 GARCH Models

ARCH(1) Process

First thing first let's rewrite the equation mentioned in the book for ARCH(p) process $\{Z_t\}$:

$$Z_t = \sqrt{h_t}e_t,\ \{e_t\} \sim IID\ N(0,1) \tag{7.2.1}$$

$$h_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i Z_{t-i}^2,\ \alpha_0 > 0,\ \alpha_i \geq 0,\ i = 1, ..., p \tag{7.2.2}$$

Now let's focus on $e_t \sim IID\ N(0,1)$, recall from basic statistic lesson we have below relationship:

$$Var[e_t] = E[e_t^2] - E[e_t]^2 \tag{7.2a}$$

since $e_t \sim IID\ N(0,1)$ simply means $E[e_t] = 0$ and $Var[e_t] = 1$, we then substitute it to equation 7.2a:

$$\begin{aligned} Var[e_t] &= E[e_t^2] - E[e_t]^2 \\ 1 &= E[e_t^2] - 0^2 \\ E[e_t^2] &= 1 \end{aligned} \tag{7.2b}$$

Now for ARCH(1) process, $p = 1$ in equation 7.2.2, thus we have:

$$
\begin{aligned}
h_t &= \alpha_0 + \sum_{i=1}^{1} \alpha_i Z_{t-i}^2 \\
h_t &= \alpha_0 + \alpha_1 Z_{t-1}^2 \qquad (7.2c)
\end{aligned}
$$

alternately substituting equation 7.2c and equation 7.2.1:

$$
\begin{aligned}
Z_t &= \sqrt{h_t} e_t \\
Z_t &= \sqrt{\alpha_0 + \alpha_1 Z_{t-1}^2 e_t} \\
Z_t^2 &= (\alpha_0 + \alpha_1 Z_{t-1}^2) e_t^2 \\
Z_t^2 &= \alpha_0 e_t^2 + \alpha_1 Z_{t-1}^2 e_t^2 = \alpha_0 \sum_{j=0}^{0} \alpha_1^j (\prod_{k=0}^{j} e_{t-k}^2) + \alpha_1^1 Z_{t-1}^2 (\prod_{j=0}^{0} e_{t-j}^2) \qquad (7.2\text{d cycle } 1) \\
Z_t^2 &= \alpha_0 e_t^2 + \alpha_1 (\sqrt{h_{t-1}} e_{t-1})^2 e_t^2 \\
Z_t^2 &= \alpha_0 e_t^2 + \alpha_1 h_{t-1} e_t^2 e_{t-1}^2 \\
Z_t^2 &= \alpha_0 e_t^2 + \alpha_1 (\alpha_0 + \alpha_1 Z_{t-2}^2) e_t^2 e_{t-1}^2 \\
Z_t^2 &= \alpha_0 e_t^2 + \alpha_0 \alpha_1 e_t^2 e_{t-1}^2 + \alpha_1^2 Z_{t-2}^2 e_t^2 e_{t-1}^2 = \alpha_0 \sum_{j=0}^{1} \alpha_1^j (\prod_{k=0}^{j} e_{t-k}^2) + \alpha_1^2 Z_{t-2}^2 (\prod_{j=0}^{1} e_{t-j}^2) \qquad (7.2\text{d cycle } 2) \\
&\ldots \\
Z_t^2 &= \alpha_0 \sum_{j=0}^{n} \alpha_1^j (\prod_{k=0}^{j} e_{t-k}^2) + \alpha_1^{n+1} Z_{t-(n+1)}^2 (\prod_{j=0}^{n} e_{t-j}^2), \ \textit{after } n+1 \textit{ cycles} \qquad (7.2d)
\end{aligned}
$$

for $\{Z_t\}$ to be stationary, we need $E[Z_t^2] < \infty$, taking expectation from equation 7.2d:

$$
\begin{aligned}
E[Z_t^2] &= E[\alpha_0 \sum_{j=0}^{n} \alpha_1^j (\prod_{k=0}^{j} e_{t-k}^2) + \alpha_1^{n+1} Z_{t-(n+1)}^2 (\prod_{j=0}^{n} e_{t-j}^2)] \\
E[Z_t^2] &= \alpha_0 \sum_{j=0}^{n} \alpha_1^j (\prod_{k=0}^{j} E[e_{t-k}^2]) + \alpha_1^{n+1} Z_{t-(n+1)}^2 (\prod_{j=0}^{n} E[e_{t-j}^2]), \ \textit{due to independence of } \{e_t\} \\
E[Z_t^2] &= \alpha_0 \sum_{j=0}^{n} \alpha_1^j (\prod_{k=0}^{j} 1) + \alpha_1^{n+1} Z_{t-(n+1)}^2 (\prod_{j=0}^{n} 1), \ \textit{substituting equation 7.2b}, \ E[e_{t-k}^2] = E[e_{t-j}^2] = 1 \\
E[Z_t^2] &= \alpha_0 \sum_{j=0}^{n} \alpha_1^j + \alpha_1^{n+1} Z_{t-(n+1)}^2 \qquad (7.2e)
\end{aligned}
$$

From equation 7.2e, especially the $\alpha_0 \sum_{j=0}^{n} \alpha_1^j$ part, it is clear that as $n \to \infty$, $E[Z_t^2] < \infty$ if and only if $\alpha_1 < 1$ which also means that for $\{Z_t\}$ to be stationary, it must be that $\alpha_1 < 1$.

Taking limit $n \to \infty$ from equation 7.2e and considering the constraint $0 < \alpha_1 < 1$ yields:

$$
\begin{aligned}
E[Z_t^2] &= \lim_{n\to\infty}\left[\alpha_0\sum_{j=0}^{n}\alpha_1^j + \alpha_1^{n+1}Z_{t-(n+1)}^2\right] \\
E[Z_t^2] &= \lim_{n\to\infty}\left[\alpha_0(\frac{1-\alpha_1^{n+1}}{1-\alpha_1}) + \alpha_1^{n+1}Z_{t-(n+1)}^2\right] \\
E[Z_t^2] &= \alpha_0(\frac{1-0}{1-\alpha_1}) + 0\times Z_{t-(n+1)}^2 \\
E[Z_t^2] &= \frac{\alpha_0}{1-\alpha_1} \qquad (7.2f)
\end{aligned}
$$

If we take the expectation from equation 7.2.1:

$$
\begin{aligned}
E[Z_t] = E[\sqrt{h_t}e_t] = E[\sqrt{h_t}]E[e_t] = E[\sqrt{h_t}]\times 0,\ &since\ e_t\ and\ h_t\ are\ independent \\
E[Z_t] = 0 & \qquad (7.2g)
\end{aligned}
$$

Recalling the fact that $Var[Z_t] = E[Z_t^2] - E[Z_t]^2$, substituting 7.2f and 7.2g yields:

$$
\begin{aligned}
Var[Z_t] = E[Z_t^2] - E[Z_t]^2 &= \frac{\alpha_0}{1-\alpha_1} - 0^2 \\
Var[Z_t] &= \frac{\alpha_0}{1-\alpha_1} \qquad (7.2h)
\end{aligned}
$$

The covariance:

$$
\begin{aligned}
Cov(Z_{t+h}, Z_t) =& E[(Z_{t+h} - E[Z_{t+h}])(Z_t - E[Z_t])] \\
Cov(Z_{t+h}, Z_t) =& E[(Z_{t+h} - 0)(Z_t - 0)],\ from\ equation\ 7.2g \\
Cov(Z_{t+h}, Z_t) =& E[Z_{t+h}Z_t],\ recall\ from\ basic\ statistic\ we\ have\ E[Y] = E[E[Y|X]],\ thus: \\
Cov(Z_{t+h}, Z_t) =& E[E[Z_{t+h}Z_t|(Z_{t+h-1} = z_{t+h-1}, Z_{t+h-2} = z_{t+h-2}, ..., Z_t = z_t)]] \\
Cov(Z_{t+h}, Z_t) =& E[E[Z_{t+h}z_t|(Z_{t+h-1} = z_{t+h-1}, Z_{t+h-2} = z_{t+h-2}, ..., Z_t = z_t)]] \\
Cov(Z_{t+h}, Z_t) =& E[z_tE[Z_{t+h}|(Z_{t+h-1} = z_{t+h-1}, Z_{t+h-2} = z_{t+h-2}, ..., Z_t = z_t)]] \\
Cov(Z_{t+h}, Z_t) =& E[z_tE[\sqrt{h_{t+h}}e_{t+h}|(Z_{t+h-1} = z_{t+h-1}, Z_{t+h-2} = z_{t+h-2}, ..., Z_t = z_t)]] \\
& since\ e_{t+h}\ is\ independent\ of\ h_{t+h},\ Z_{t+h-1}, Z_{t+h-2}, ..., Z_t\ thus: \\
Cov(Z_{t+h}, Z_t) =& E[z_tE[\sqrt{h_{t+h}}|(Z_{t+h-1} = z_{t+h-1}, Z_{t+h-2} = z_{t+h-2}, ..., Z_t = z_t)]E[e_{t+h}]] \\
Cov(Z_{t+h}, Z_t) =& E[z_tE[\sqrt{h_{t+h}}|(Z_{t+h-1} = z_{t+h-1}, Z_{t+h-2} = z_{t+h-2}, ..., Z_t = z_t)]\times 0] \\
Cov(Z_{t+h}, Z_t) =& E[0] \\
Cov(Z_{t+h}, Z_t) =& 0 \qquad (7.2i)
\end{aligned}
$$

Remember the notation equivalence between the lecture slide and book as explained in Module 3 - Lesson 1 - Slide 7 and Module 3 - Lesson 2 - Slide 3.

GARCH(1, 1) Process

The general GARCH(p, q) equation:

$$h_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i Z_{t-i}^2 + \sum_{j=1}^{q} \beta_j h_{t-j};\ \alpha_0 > 0,\ \alpha_i \geq 0, i = 1, 2, ..., p;\ \beta_j \geq 0,\ j = 1, 2, ..., q \tag{7.2.6}$$

for GARCH(1, 1) we have:

$$h_t = \alpha_0 + \alpha_1 Z_{t-1}^2 + \beta_1 h_{t-1} \tag{7.2j}$$

substituting equation 7.2j to equation 7.2.1:

$$\begin{aligned}
Z_t &= \sqrt{h_t} e_t \\
Z_t &= \sqrt{\alpha_0 + \alpha_1 Z_{t-1}^2 + \beta_1 h_{t-1}} e_t \\
Z_t^2 &= (\alpha_0 + \alpha_1 Z_{t-1}^2 + \beta_1 h_{t-1}) e_t^2 \\
Z_t^2 &= (\alpha_0 + \alpha_1 (\sqrt{h_{t-1}} e_{t-1})^2 + \beta_1 h_{t-1}) e_t^2 \\
Z_t^2 &= (\alpha_0 + \alpha_1 h_{t-1} e_{t-1}^2 + \beta_1 h_{t-1}) e_t^2 \\
Z_t^2 &= \alpha_0 e_t^2 + \alpha_1 h_{t-1} e_t^2 e_{t-1}^2 + \beta_1 h_{t-1} e_t^2 \\
E[Z_t^2] &= E[\alpha_0 e_t^2 + \alpha_1 h_{t-1} e_t^2 e_{t-1}^2 + \beta_1 h_{t-1} e_t^2],\ \textit{taking expectation} \\
E[Z_t^2] &= \alpha_0 E[e_t^2] + \alpha_1 E[h_{t-1}] E[e_t^2] E[e_{t-1}^2] + \beta_1 E[h_{t-1}] E[e_t^2],\ \textit{due to } \{e_t\} \textit{ independence} \\
E[Z_t^2] &= \alpha_0 \times 1 + \alpha_1 E[h_{t-1}] \times 1 \times 1 + \beta_1 E[h_{t-1}] \times 1,\ \textit{from equation 7.2b} \\
E[Z_t^2] &= \alpha_0 + \alpha_1 E[h_{t-1}] + \beta_1 E[h_{t-1}] \\
E[Z_t^2] &= \alpha_0 + (\alpha_1 + \beta_1) E[h_{t-1}] \\
E[Z_t^2] &= \alpha_0 + (\alpha_1 + \beta_1)(\alpha_0 + (\alpha_1 + \beta_1) E[h_{t-2}]) \\
E[Z_t^2] &= \alpha_0 + \alpha_0(\alpha_1 + \beta_1) + (\alpha_1 + \beta_1)^2 E[h_{t-2}] \\
E[Z_t^2] &= \alpha_0 \sum_{j=0}^{1} (\alpha_1 + \beta_1)^j + (\alpha_1 + \beta_1)^2 E[h_{t-2}] && \text{(7.2k 1st expansion)} \\
E[Z_t^2] &= \alpha_0 \sum_{j=0}^{1} (\alpha_1 + \beta_1)^j + (\alpha_1 + \beta_1)^2 (\alpha_0 + (\alpha_1 + \beta_1) E[h_{t-3}]) \\
E[Z_t^2] &= \alpha_0 \sum_{j=0}^{1} (\alpha_1 + \beta_1)^j + \alpha_0(\alpha_1 + \beta_1)^2 + (\alpha_1 + \beta_1)^3 E[h_{t-3}]) \\
E[Z_t^2] &= \alpha_0 \sum_{j=0}^{2} (\alpha_1 + \beta_1)^j + (\alpha_1 + \beta_1)^3 E[h_{t-3}]) && \text{(7.2k 2nd expansion)} \\
&... \\
E[Z_t^2] &= \alpha_0 \sum_{j=0}^{n} (\alpha_1 + \beta_1)^j + (\alpha_1 + \beta_1)^{n+1} E[h_{t-(n+1)}]),\ \textit{after } n^{th} \textit{ expansion} && \text{(7.2k)}
\end{aligned}$$

From equation 7.2k it is clear that when $n \to \infty$ in order to fulfill the stationary requirements of $\{Z_t\}$ that is $E[Z_t^2] < \infty$, we must have $\alpha_1 + \beta_1 < 1$, and since by binomial expansion we have:

$$\alpha_0 \sum_{j=0}^{n} (\alpha_1 + \beta_1)^j = \alpha_0 + \alpha_0 \sum_{j=1}^{n} (\alpha_1 + \beta_1)^j = \alpha_0 + \alpha_0 \sum_{j=1}^{n} \left[\sum_{k=0}^{j} \binom{j}{k} \alpha_1^{j-k} \beta_1^k \right]$$
$$= \alpha_0 + \alpha_0 \sum_{j=1}^{n} \left[\binom{j}{0} \alpha_1^{j-0} \beta_1^0 + \sum_{k=1}^{j-1} \binom{j}{k} \alpha_1^{j-k} \beta_1^k + \binom{j}{j} \alpha_1^{j-j} \beta_1^j \right]$$
$$= \alpha_0 + \alpha_0 \sum_{j=1}^{n} \left[\alpha_1^j + \sum_{k=1}^{j-1} \binom{j}{k} \alpha_1^{j-k} \beta_1^k + \beta_1^j \right]$$
$$= \alpha_0 + \alpha_0 \sum_{j=1}^{n} \alpha_1^j + \alpha_0 \sum_{j=1}^{n} \sum_{k=1}^{j-1} \binom{j}{k} \alpha_1^{j-k} \beta_1^k + \alpha_0 \sum_{j=1}^{n} \beta_1^j \tag{7.2l}$$

We can see also from equation 7.2l that, as $n \to \infty$ in order to fulfill the stationary requirements $E[Z_t^2] < \infty$ of $\{Z_t\}$, we must have $\alpha_1, \beta_1 < 1$.

If we take limit $n \to \infty$ in equation 7.2k with the constraint $\alpha_1 + \beta_1 < 1,\ 0 < \alpha_1, \beta_1 < 1$ we have:

$$E[Z_t^2] = \lim_{n \to \infty} \left[\alpha_0 \sum_{j=0}^{n} (\alpha_1 + \beta_1)^j + (\alpha_1 + \beta_1)^{n+1} E[h_{t-(n+1)}]) \right]$$
$$E[Z_t^2] = \lim_{n \to \infty} \left[\alpha_0 \frac{1 - (\alpha_1 + \beta_1)^{n+1}}{1 - (\alpha_1 + \beta_1)} + (\alpha_1 + \beta_1)^{n+1} E[h_{t-(n+1)}]) \right]$$
$$E[Z_t^2] = \alpha_0 \frac{1 - 0}{1 - (\alpha_1 + \beta_1)} + 0 \times E[h_{t-(n+1)}])$$
$$E[Z_t^2] = \frac{\alpha_0}{1 - (\alpha_1 + \beta_1)}$$
$$E[Z_t^2] = \frac{\alpha_0}{1 - \alpha_1 - \beta_1} \tag{7.2m}$$

Similar to calculation of mean and variance of ARCH(1) process in equation 7.2g and 7.2h, the mean and variance of GARCH(1, 1) process is:

$$E[Z_t] = E[\sqrt{h_t} e_t] = E[\sqrt{h_t}] E[e_t] = E[\sqrt{h_t}] \times 0,\ \textit{since } e_t \textit{ and } h_t \textit{ are independent}$$
$$E[Z_t] = 0 \tag{7.2n}$$

Recalling the fact that $Var[Z_t] = E[Z_t^2] - E[Z_t]^2$, substituting 7.2m and 7.2n yields:

$$Var[Z_t] = E[Z_t^2] - E[Z_t]^2 = \frac{\alpha_0}{1 - \alpha_1 - \beta_1} - 0^2$$
$$Var[Z_t] = \frac{\alpha_0}{1 - \alpha_1 - \beta_1} \tag{7.2o}$$

The covariance:

$$
\begin{aligned}
Cov(Z_{t+h}, Z_t) =& E[(Z_{t+h} - E[Z_{t+h}])(Z_t - E[Z_t])] \\
Cov(Z_{t+h}, Z_t) =& E[(Z_{t+h} - 0)(Z_t - 0)],\ from\ equation\ 7.2n \\
Cov(Z_{t+h}, Z_t) =& E[Z_{t+h}Z_t],\ recall\ from\ basic\ statistic\ we\ have\ E[Y] = E[E[Y|X]],\ thus: \\
Cov(Z_{t+h}, Z_t) =& E[E[Z_{t+h}Z_t|(Z_{t+h-1} = z_{t+h-1}, Z_{t+h-2} = z_{t+h-2}, ..., Z_t = z_t)]] \\
Cov(Z_{t+h}, Z_t) =& E[E[Z_{t+h}z_t|(Z_{t+h-1} = z_{t+h-1}, Z_{t+h-2} = z_{t+h-2}, ..., Z_t = z_t)]] \\
Cov(Z_{t+h}, Z_t) =& E[z_tE[Z_{t+h}|(Z_{t+h-1} = z_{t+h-1}, Z_{t+h-2} = z_{t+h-2}, ..., Z_t = z_t)]] \\
Cov(Z_{t+h}, Z_t) =& E[z_tE[\sqrt{h_{t+h}}e_{t+h}|(Z_{t+h-1} = z_{t+h-1}, Z_{t+h-2} = z_{t+h-2}, ..., Z_t = z_t)]] \\
& since\ e_{t+h}\ is\ independent\ of\ h_{t+h},\ Z_{t+h-1}, Z_{t+h-2}, ..., Z_t\ thus: \\
Cov(Z_{t+h}, Z_t) =& E[z_tE[\sqrt{h_{t+h}}|(Z_{t+h-1} = z_{t+h-1}, Z_{t+h-2} = z_{t+h-2}, ..., Z_t = z_t)]E[e_{t+h}]] \\
Cov(Z_{t+h}, Z_t) =& E[z_tE[\sqrt{h_{t+h}}|(Z_{t+h-1} = z_{t+h-1}, Z_{t+h-2} = z_{t+h-2}, ..., Z_t = z_t)] \times 0] \\
Cov(Z_{t+h}, Z_t) =& E[0] \\
Cov(Z_{t+h}, Z_t) =& 0 \qquad \text{(7.2p)}
\end{aligned}
$$

Remember the notation equivalence between the lecture slide and book as explained in Module 3 - Lesson 1 - Slide 7 and Module 3 - Lesson 2 - Slide 3.