Bayesian Learning - Lab 03

Anub Dikshit (anudi287) and Maximilian Pfundstein (maxpf364)

2019-09-07

Contents

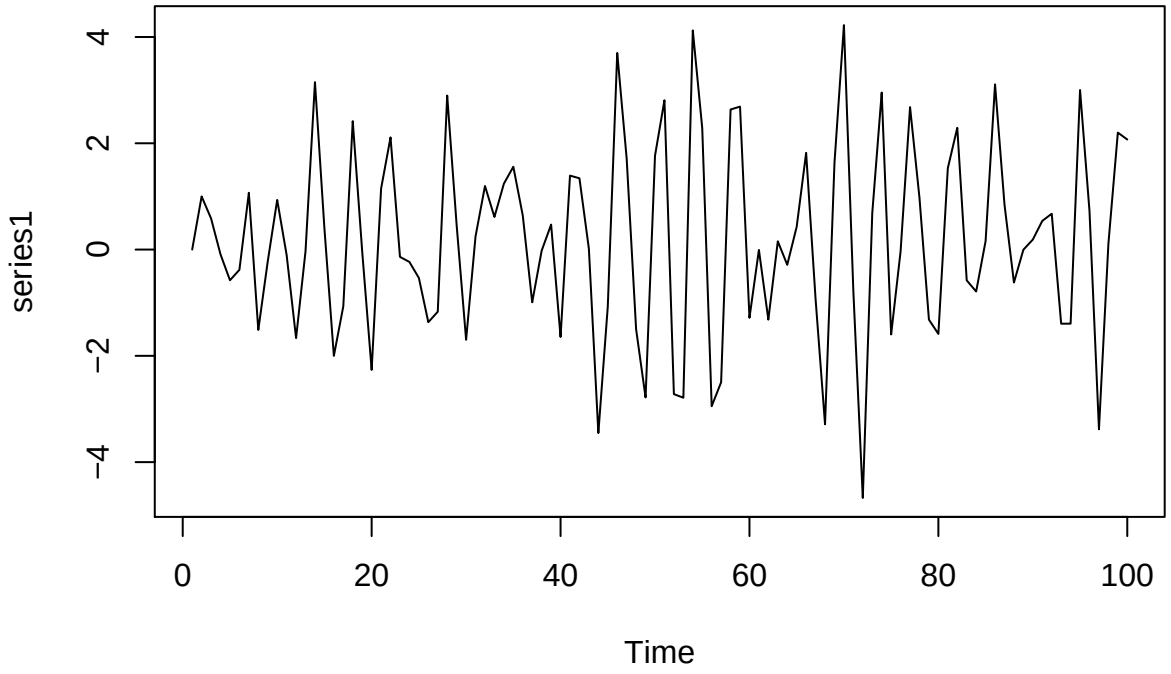
1 Computations with Simulated Data

Exercise:

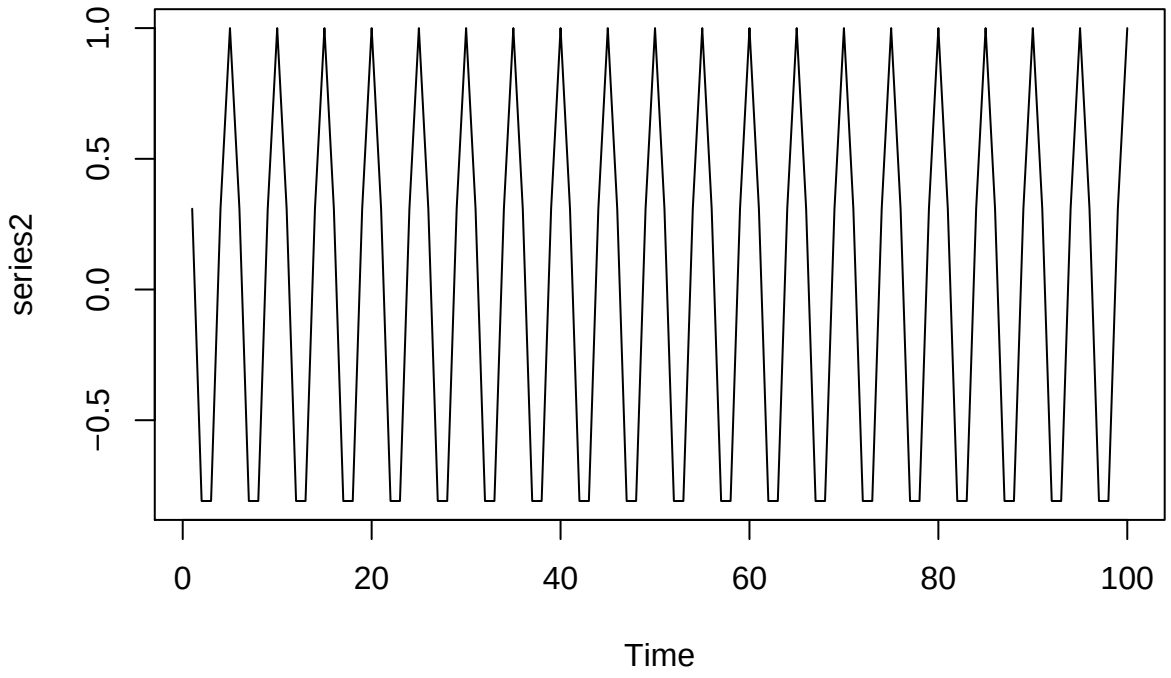
a) Generate two time series $x_t = -0.8x_{t-2} + w_t$, where $x_0 = x_1 = 0$ and $x_t = cos(\frac{2\pi t}{5})$ with 100 observations each.

```
################################################################################
# Exercise 1.a)
################################################################################

x0 = 0
x1 = 0

n = 100

# Series 1
generate_S1 = function(t, x0=0, x1=1) {

  series = vector(length = t)
  series[1] = x0
  series[2] = x1

  for (i in 3:t) {
    series[i] = -0.8 * series[i-2] + rnorm(n=1, mean=0, sd=1)
  }

  return(ts(series))
}

# Series 2
generate_S2 = function(t) {
  series = vector(length = t)

  for (i in 1:t) {
    series[i] = cos(2 * pi * i / 5)
```

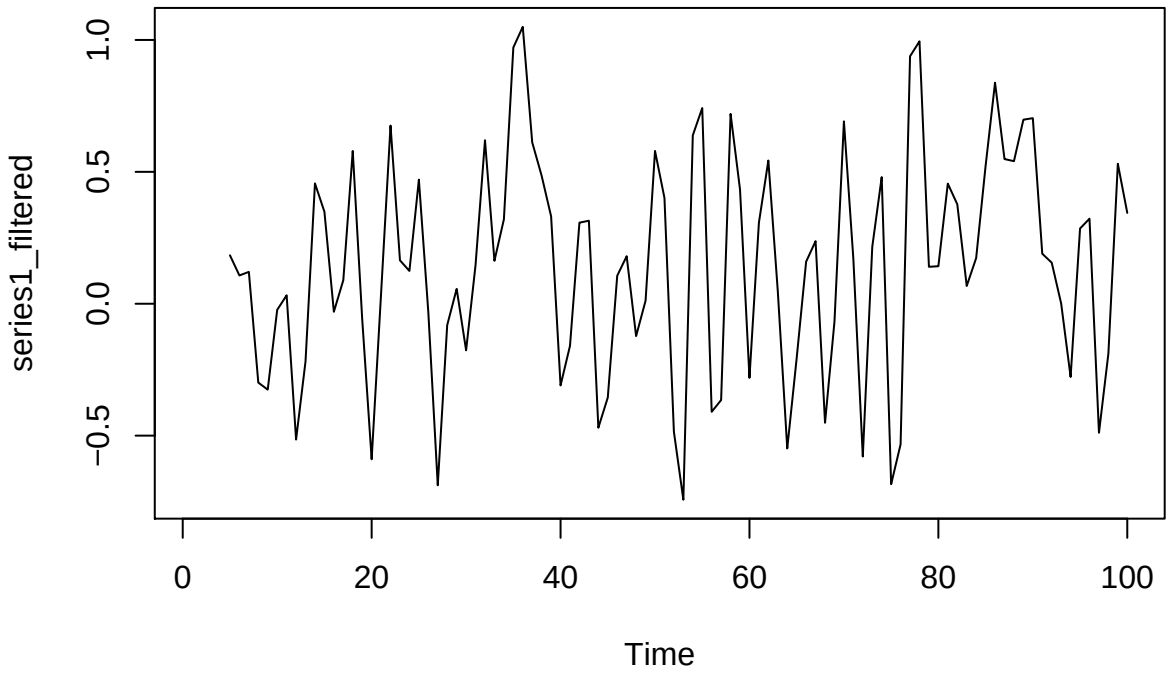
```
  }

  return(ts(series))
}

series1 = generate_S1(n)
series2 = generate_S2(n)

plot(series1)
```



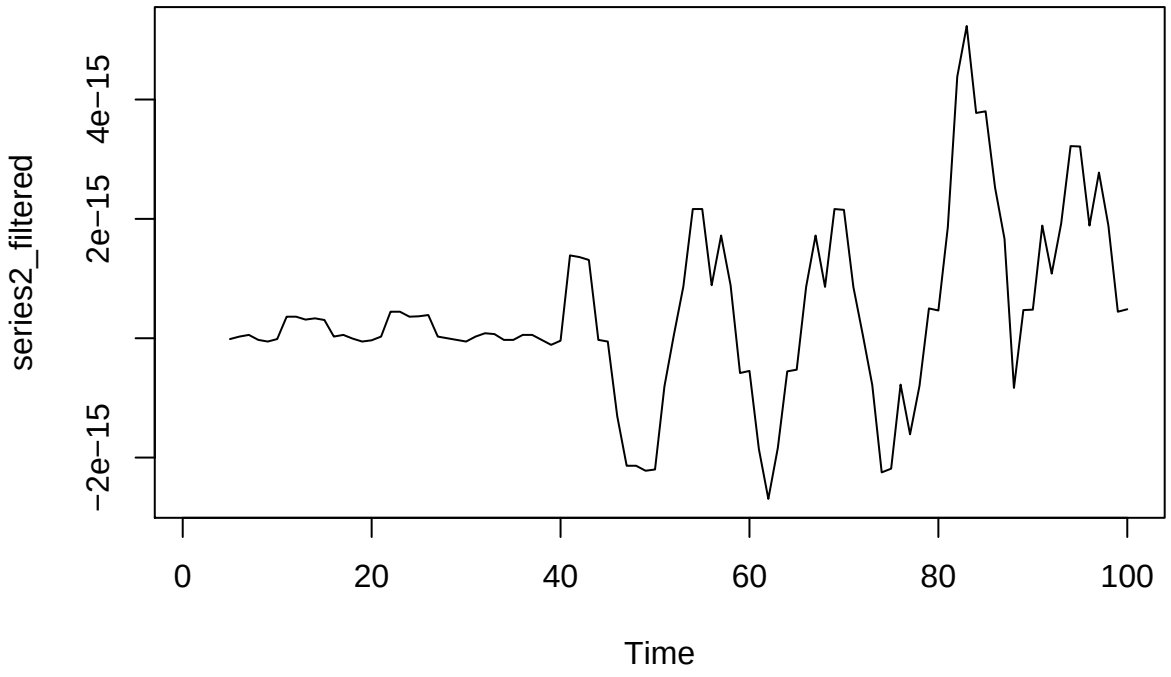

```
plot(series2)
```



```r
series1_filtered = filter(series1, filter = rep(0.2, 5), sides = 1)
series2_filtered = filter(series2, filter = rep(0.2, 5), sides = 1)

plot(series1_filtered, type = 'l')
```

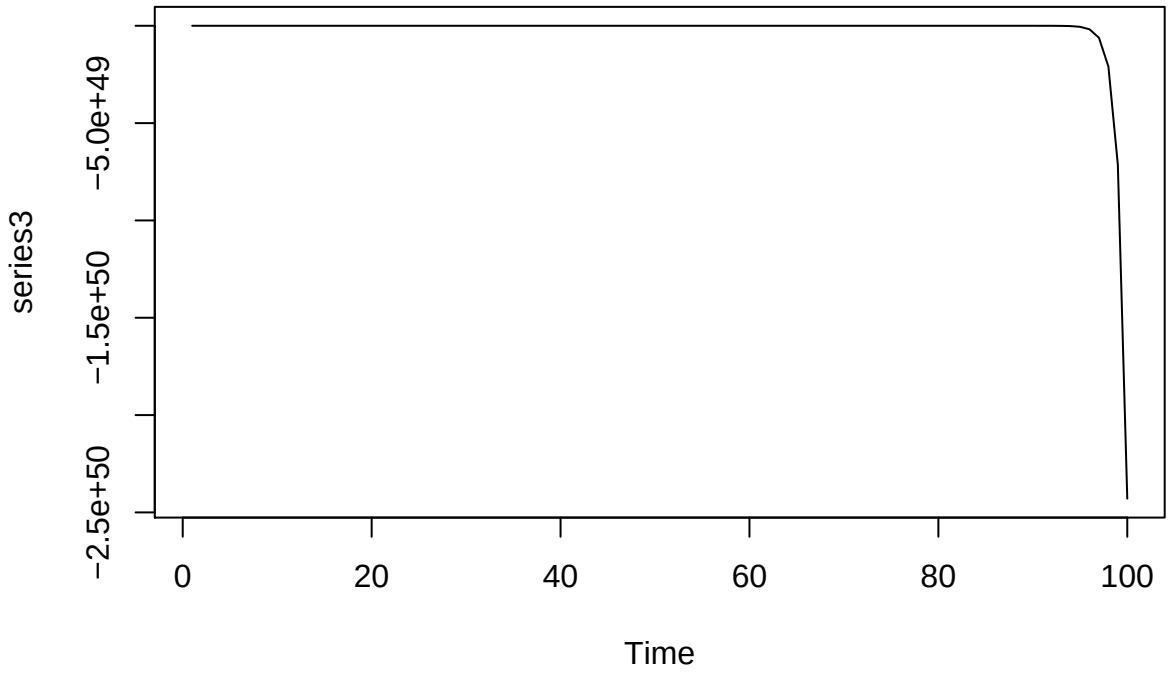



```r
plot(series2_filtered, type = 'l')
```




```r
################################################################################
# Exercise 1.b)
################################################################################
```

```r
generate_S3 = function(t, X, W) {
  series = vector(length=t)
  white_noise = vector(length=t)

  series[1:length(X)] = X
  white_noise[1:length(W)] = W

  for (i in 7:t) {
    W[1:6] = W[2:7]
    W[7] = rnorm(1, mean=0, sd=1)
    series[i] = 4 * series[i-1] - 2 * series[i-2] - series[i-5] +
                W[7] + 3 * W[5] + W[2] - 4 * W[1]
  }

  return(ts(series))
}

series3 = generate_S3(t=n, X = rnorm(7, mean=0, sd=1), W = rnorm(7, mean=0, sd=1))

plot(series3)
```



2 Visualization, detrending and residual analysis of Rhine data.

3 Analysis of oil and gas time series.

4 Source Code

```r
knitr::opts_chunk$set(echo = TRUE)
set.seed(12345)

################################################################################
# Exercise 1.a)
################################################################################

x0 = 0
x1 = 0

n = 100

# Series 1
generate_S1 = function(t, x0=0, x1=1) {

  series = vector(length = t)
  series[1] = x0
  series[2] = x1

  for (i in 3:t) {
    series[i] = -0.8 * series[i-2] + rnorm(n=1, mean=0, sd=1)
  }

  return(ts(series))
}

# Series 2
generate_S2 = function(t) {
  series = vector(length = t)

  for (i in 1:t) {
    series[i] = cos(2 * pi * i / 5)
  }

  return(ts(series))
}

series1 = generate_S1(n)
series2 = generate_S2(n)

plot(series1)
plot(series2)


series1_filtered = filter(series1, filter = rep(0.2, 5), sides = 1)
series2_filtered = filter(series2, filter = rep(0.2, 5), sides = 1)

plot(series1_filtered, type = 'l')
plot(series2_filtered, type = 'l')


################################################################################
# Exercise 1.b)
```

```
################################################################################

generate_S3 = function(t, X, W) {
  series = vector(length=t)
  white_noise = vector(length=t)

  series[1:length(X)] = X
  white_noise[1:length(W)] = W

  for (i in 7:t) {
    W[1:6] = W[2:7]
    W[7] = rnorm(1, mean=0, sd=1)
    series[i] = 4 * series[i-1] - 2 * series[i-2] - series[i-5] +
                W[7] + 3 * W[5] + W[2] - 4 * W[1]
  }

  return(ts(series))
}

series3 = generate_S3(t=n, X = rnorm(7, mean=0, sd=1), W = rnorm(7, mean=0, sd=1))

plot(series3)
```