

Exercise 3

TMA4300

Erling Fause Steen, Christian Oppegård Moen

Spring 2022

Contents

Introduction	2
Problem A	2
1	2
2	4
Problem B	5

Introduction

Problem A

In this problem we will analyze a dataset which contain a sequence of length $T = 100$ of a non-Gaussian time-series for which we will compare two different parameter estimators. Consider the AR(2) specified by the relation

$$x_t = \beta_1 x_{t-1} + \beta_2 x_{t-2} + e_t, \quad (1)$$

where e_t are independent and identically distributed (iid) random variables with zero mean and constant variance. Also, consider the loss functions with respect to $\beta = [\beta_1, \beta_2]^T$ given by

$$Q_{LS}(\mathbf{x}) = \sum_{t=3}^T (x_t - \beta_1 x_{t-1} - \beta_2 x_{t-2})^2$$
$$Q_{LA}(\mathbf{x}) = \sum_{t=3}^T |x_t - \beta_1 x_{t-1} - \beta_2 x_{t-2}|.$$

Then, the least sum residuals (LS) and least sum of absolute residuals (LA) are obtained by minimizing $Q_{LS}(\mathbf{x})$ and $Q_{LA}(\mathbf{x})$ respectively. We denote the minimisers by $\hat{\beta}_{LS}$ and $\hat{\beta}_{LA}$, and define the estimated residuals by $\hat{e} = x_t - \hat{\beta}_1 x_{t-1} - \hat{\beta}_2 x_{t-2}$ for $t = 3, \dots, T$ with mean \bar{e} . Then, $\hat{\varepsilon} = \hat{e} - \bar{e}$ is re-centered to have mean zero.

1

Now we will use the residual resampling bootstrap method to evaluate the relative performance of the two parameter estimators, $\hat{\beta}_{LS}$ and $\hat{\beta}_{LA}$, which are calculated by the given function `ARp.beta.est`. We consider the variance and mean of β_{LS}^* and β_{LA}^* obtained by minimizing $Q_{LS}(\mathbf{x}^*)$ and $Q_{LA}(\mathbf{x}^*)$ for bootstrap sample \mathbf{x}^* , respectively. In Table 1 the variance of β_{LA} is slightly lower than that of β_{LS} , and the absolute value of the bias is slightly lower as well. This suggest that the LS estimator is not optimal.

```
x = data3A$x
plot(x, type = "l", xlab = "t", ylab = "x")
```

```
detach("package:dplyr", unload = T)
rsBoot = function(B, x, p = 2) {
  T = length(x)
  # Estimate beta
  beta.hat = ARp.beta.est(x, p)
  # calculate observed residuals of AR(p) seq
  e.LS.observed = ARp.resid(x, beta.hat$LS)
  e.LA.observed = ARp.resid(x, beta.hat$LA)
  # Allocate memory
  beta.LS.star = matrix(nrow = B, ncol = 2)
  beta.LA.star = matrix(nrow = B, ncol = 2)
  # e.LS.star = vector(mode = 'double', length = B) x.LS.star = matrix(nrow =
  # B, ncol = 2) x.LA.star = matrix(nrow = B, ncol = 2)

  # Bootstrap
  for (b in 1:B) {
    # Resample from observed residuals to yield pseudo innovations
```

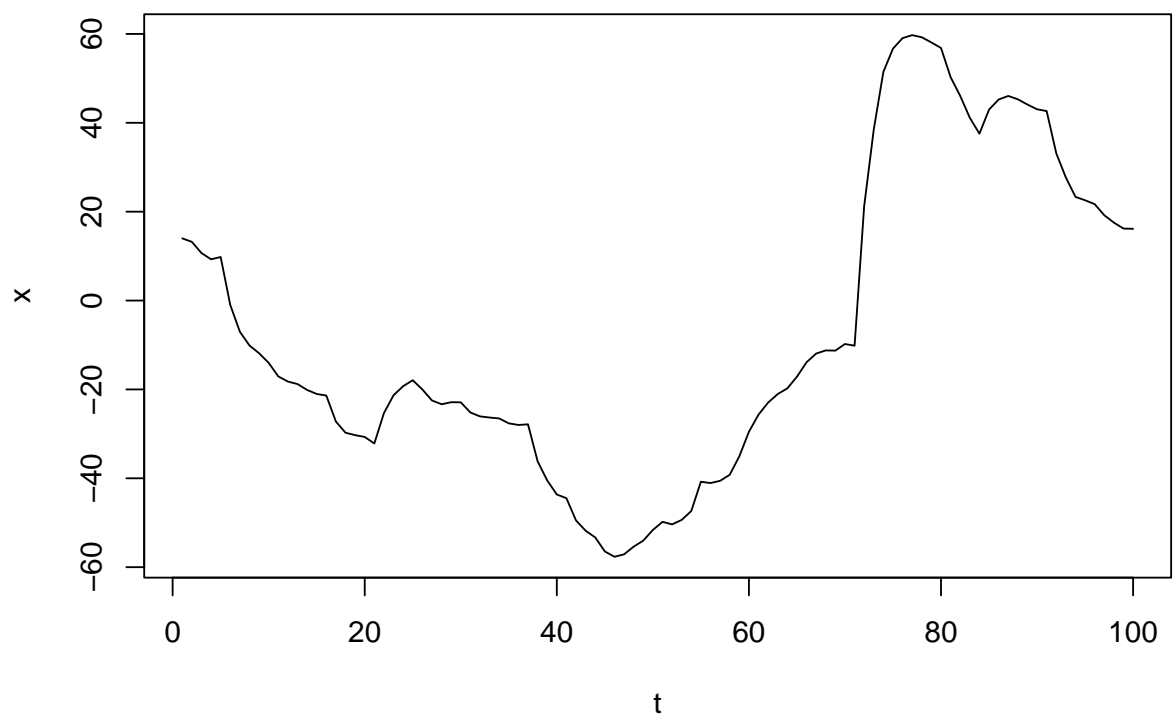


Figure 1: Given dataset. write more

Table 1: Variance and mean for β_1^* and β_2^* .

	Var ₁	Var ₂	bias ₁	bias ₂
LS	0.005451	0.005240	-0.012071	0.005988
LA	0.000362	0.000358	-0.001311	0.000829

```

e.LS.star = sample(e.LS.observed, size = T, replace = TRUE)
e.LA.star = sample(e.LA.observed, size = T, replace = TRUE)
# Generate pseudo data (Timeseries based on sampled residuals and
# beta.hat)
i = sample(T - 1, 1)
x.LS.star = ARp.filter(x[c(i, i + 1)], beta.hat$LS, e.LS.star)
x.LA.star = ARp.filter(x[c(i, i + 1)], beta.hat$LA, e.LA.star)
# compute beta star
beta.LS.star[b, ] = ARp.beta.est(x.LS.star, 2)$LS
beta.LA.star[b, ] = ARp.beta.est(x.LA.star, 2)$LA
}
return(list(beta.hat = beta.hat, beta.LS = beta.LS.star, beta.LA = beta.LA.star,
  e.LS = e.LS.observed, e.LA = e.LA.observed, x.LS = x.LS.star, x.LA = x.LA.star))
}

B = 1500
set.seed(420)
boot = rsBoot(B, x)

# Compute variance and bias

beta.var = rbind(apply(boot$beta.LS, 2, var), apply(boot$beta.LA, 2, var))
beta.bias = rbind(apply(boot$beta.LS, 2, mean) - boot$beta.hat$LS, apply(boot$beta.LA,
  2, mean) - boot$beta.hat$LA)
beta.vb = cbind(beta.var, beta.bias)
beta.vb

##           [,1]      [,2]      [,3]      [,4]
## [1,] 0.0054513391 0.0052403653 -0.012070714 0.0059875781
## [2,] 0.0003622196 0.0003583251 -0.001311029 0.0008292713

rownames(beta.vb) = c("LS", "LA")
# colnames(beta.vb) = c('var_1', 'var_2', 'bias_1', 'bias_2')

kable(round(beta.vb, 6), caption = "Variance and mean for  $\beta_1^*$  and  $\beta_2^*$ .",
  format = "latex") %>%
  kable_styling() %>%
  add_header_above(c(" ", "Var$_{1}$", "Var$_{2}$", "bias$_{1}$", "bias$_{2}$"),
    escape = FALSE)

```

2

Next, we will compute a 95% prediction interval for x_{101} for both the LS and the LA estimator. That is, we bootstrap sample B times

$$x_{101} = \beta_1^* x_{100} + \beta_2^* x_{99} + \hat{e}_i \quad (2)$$

Table 2: Prediction interval of x_{101} .

	2.5%	97.5%
LS	7.44	23.78
LA	7.30	23.54

where β_1^* and β_2^* are sampled from all 1500 bootstrap samples found in section A1, and \hat{e}_i is sampled from the estimated residuals \hat{e}_t for $t = 2, \dots, 100$. The prediction intervals are found in Table 2.

```
set.seed(420)
# Bootstrap x_101 using beta.star and e.observed
x_101.LS = vector(mode = "double", length = B)
x_101.LA = vector(mode = "double", length = B)
for (b in 1:B) {
  i.e = sample(98, 1)
  i.beta = sample(1500, 1)
  x_101.LS[b] = boot$beta.LS[i.beta, ] %*% x[c(100, 99)] + boot$e.LS[i.e]
  x_101.LA[b] = boot$beta.LS[i.beta, ] %*% x[c(100, 99)] + boot$e.LA[i.e]
}

pi.boot = rbind(LS = quantile(x_101.LS, probs = c(0.025, 0.975)), LA = quantile(x_101.LA,
  probs = c(0.025, 0.975)))
kable(round(pi.boot, 2), caption = "Prediction interval of  $x_{101}$ $.")
```

Problem B

We will investigate the concentration of bilirubin (mg/dL) in blood samples taken from three young men.

Individual	Concentration (mg/dL)											
1	0.14	0.20	0.23	0.27	0.27	0.34	0.41	0.41	0.55	0.61	0.66	
2	0.20	0.27	0.32	0.34	0.34	0.38	0.41	0.41	0.48	0.55		
3	0.32	0.41	0.41	0.55	0.55	0.62	0.71	0.91				

```
bilirubin <- read.table("./additionalFiles/bilirubin.txt", header = T)
bilirubin
```

```
##      meas pers
## 1  0.14  p1
## 2  0.20  p1
## 3  0.23  p1
## 4  0.27  p1
## 5  0.27  p1
## 6  0.34  p1
## 7  0.41  p1
## 8  0.41  p1
## 9  0.55  p1
## 10 0.61  p1
## 11 0.66  p1
## 12 0.20  p2
```

```
## 13 0.27 p2
## 14 0.32 p2
## 15 0.34 p2
## 16 0.34 p2
## 17 0.38 p2
## 18 0.41 p2
## 19 0.41 p2
## 20 0.48 p2
## 21 0.55 p2
## 22 0.32 p3
## 23 0.41 p3
## 24 0.41 p3
## 25 0.55 p3
## 26 0.55 p3
## 27 0.62 p3
## 28 0.71 p3
## 29 0.91 p3
```

```
kable(bilirubin)
```

meas	pers
0.14	p1
0.20	p1
0.23	p1
0.27	p1
0.27	p1
0.34	p1
0.41	p1
0.41	p1
0.55	p1
0.61	p1
0.66	p1
0.20	p2
0.27	p2
0.32	p2
0.34	p2
0.34	p2
0.38	p2
0.41	p2
0.41	p2
0.48	p2
0.55	p2
0.32	p3
0.41	p3
0.41	p3
0.55	p3
0.55	p3
0.62	p3
0.71	p3
0.91	p3