Bootstrap Project: Dataset 3

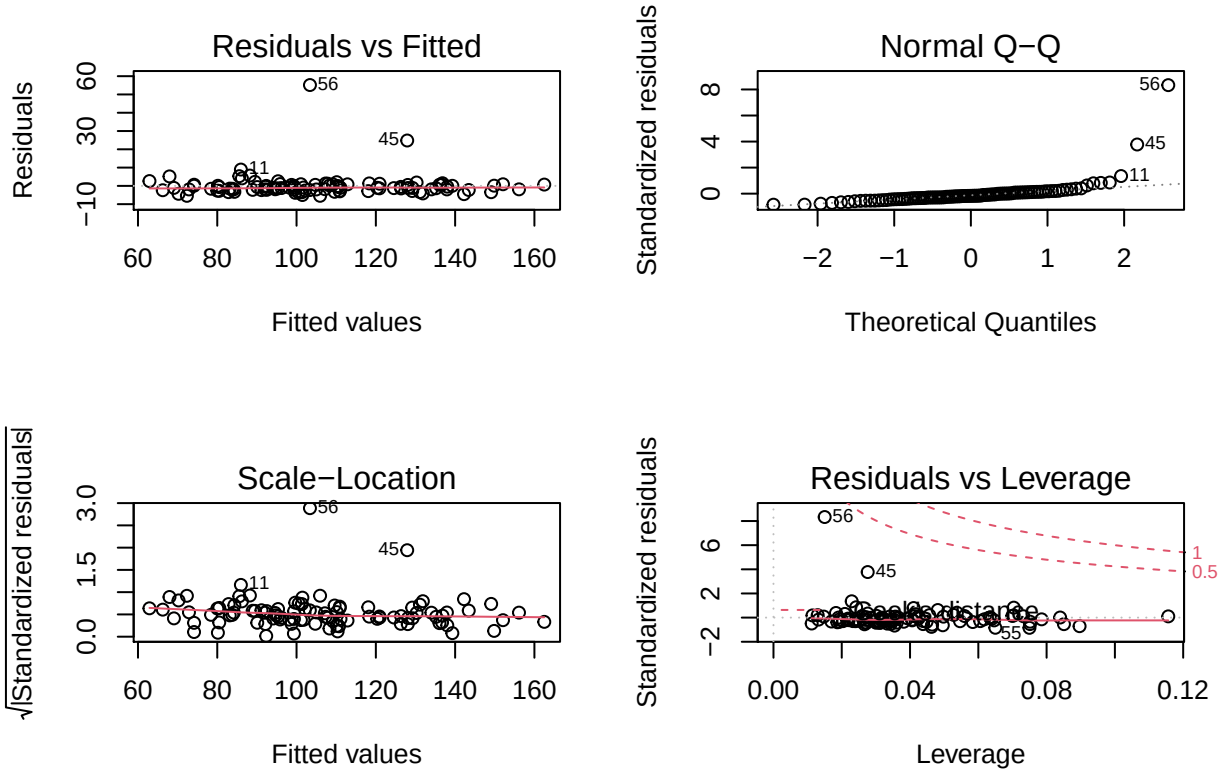
Cesar Conejo, Arturo Prieto

3/10/2021

Introduction

In this task, we study the dataset `data_3.csv` which contains four columns refering to a response variable `y` and 3 covariates `x1`, `x2`, and `x3`. The goal of the task is to build a linear regression model to `y` in terms of some of the other three variables. However, if we apply a linear model, due to the data constains several outliers, it can be checked in the following picture how the error residuals are no not normally distributed.

```
model.lm <- lm(y ~  x1 + x2 + x3, data = data3)
par(mfrow=c(2,2))
plot(model.lm)
```



The coefficients for the linear model are given by $\beta_1 = 6.29$, $\beta_2 = 0.91$, $\beta_3 = 2.02$, and $\beta_4 = 3.99$. As a result, we build a robust linear regression model `rlm()` from `MASS` package and use the bootstrap to study the significance of the regressors.

Robust regression

In this section, we calibrate a robust linear model with the function `rlm()`. From the output, we can

```
model.rlm <- rlm(y ~  x1 + x2 + x3, data = data3)
knitr::kable(coef(summary(model.rlm)), digits = 3)
```

	Value	Std. Error	t value
(Intercept)	4.735	0.905	5.230
x1	3.707	0.959	3.864
x2	4.653	0.959	4.851
x3	1.317	0.958	1.374

Then, we can explore a 95% bootstrap confidence interval on the regressors's coefficients in order to study their significance. In this case, we have two alternatives: bootstrap in pairs and bootstrap in residuals. However, given the presence of outliers and influencial observations the bootstap in pairs can lead to low quality estimator. For example, in some samples, it is posible to exclude some influential observations, given a high variability in the results. In conclusion, we tackle this estimation problem using the bootstrap in residuals technique. The following code shows how to realize the bootstrap configuration:

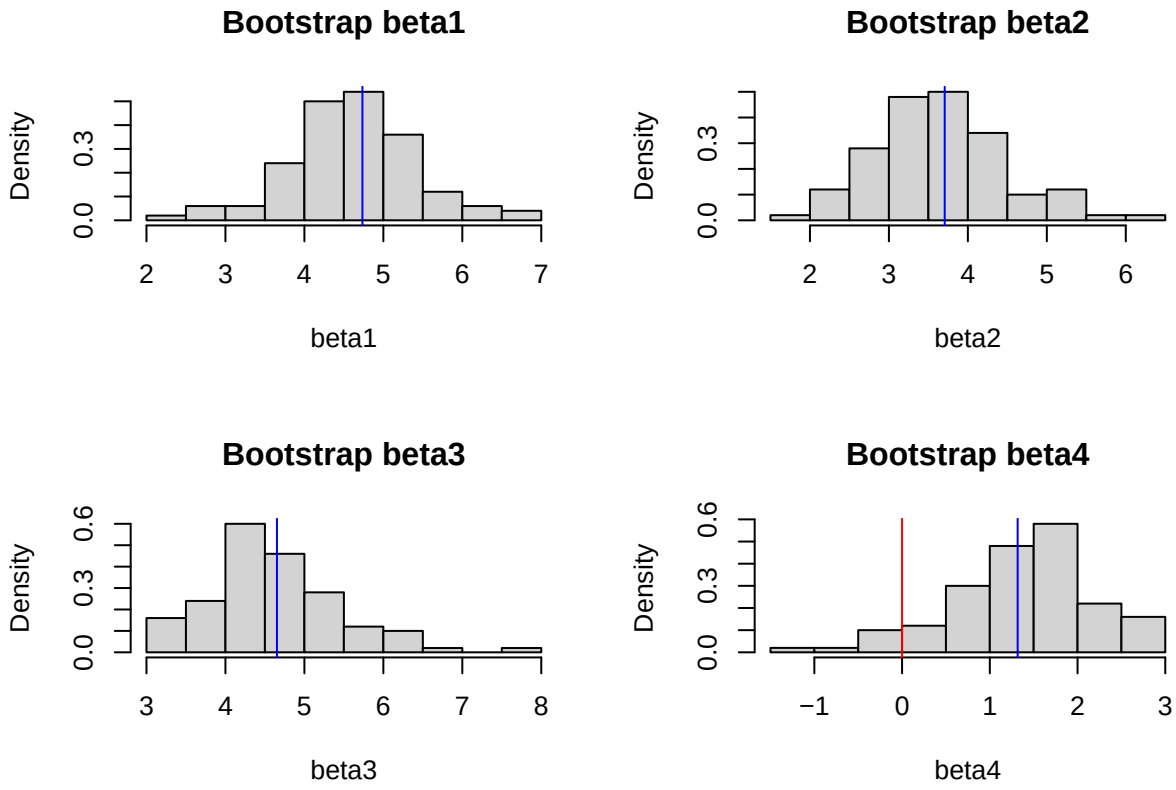
```
res_rob_rg <- function(x, beta, xdata){
  y_fit  <- beta[1] + beta[2]*xdata$x1 + beta[3]*xdata$x2 + beta[4]*xdata$x3 + x
  bmodel <- rlm(y_fit ~ x1 + x2 + x3, data = xdata)
  return(coef(bmodel))
}

rres    <- model.rlm$residuals
rbeta.h <- coef(model.rlm)

B <- 100
set.seed(1)
coeff.res <- bootstrap(rres, B, res_rob_rg, beta = rbeta.h, xdata = data3)$thetastar
```

Then, the following figure shows the distribution of the estimate parameters.

```
par(mfrow=c(2,2))
hist(coeff.res[1,], main = "Bootstrap beta1", probability = T, xlab = "beta1")
abline(v = rbeta.h[1], col ="blue")
hist(coeff.res[2,], main = "Bootstrap beta2", probability = T, xlab = "beta2")
abline(v = rbeta.h[2], col ="blue")
hist(coeff.res[3,], main = "Bootstrap beta3", probability = T, xlab = "beta3")
abline(v = rbeta.h[3], col ="blue")
hist(coeff.res[4,], main = "Bootstrap beta4", probability = T, xlab = "beta4")
abline(v = 0, col ="red")
abline(v = rbeta.h[4], col ="blue")
```



In this case, the previous graphic show how the coefficient related to β_4 contains the value of 0 inside the interval. Moreover we can compute the basic bootstrap confidence interval

```
basic_beta1 <- 2*rbeta.h[1] - quantile(coeff.res[1,], c(0.975,0.025))
basic_beta2 <- 2*rbeta.h[2] - quantile(coeff.res[2,], c(0.975,0.025))
basic_beta3 <- 2*rbeta.h[3] - quantile(coeff.res[3,], c(0.975,0.025))
basic_beta4 <- 2*rbeta.h[4] - quantile(coeff.res[4,], c(0.975,0.025))
knitr::kable(rbind(basic_beta1, basic_beta2, basic_beta3, basic_beta4), digits = 3)
```

	97.5%	2.5%
basic_beta1	3.217	6.613
basic_beta2	2.195	5.169
basic_beta3	3.107	6.128
basic_beta4	-0.213	2.872

Backward elimination

From the previous section, we are suspicious about the significance of the coefficient β_4. For this reason, we will explore the significance of this covariate deeply with the bias corrected accelerated BC_a bootstrap. In this case, we use again the residual bootstrap, but we fix the values corresponding to $\beta 1$, $\beta 2$ and $\beta 3$. In other words, we compute only the effect of the residuals only over the covariate `x3`.

```
# Fitted values for theoretical betas
y_predict <-  predict(model.rlm)
```

```r
res_rob_rg_x3 <- function(x, beta, xdata){

  y_bt <-  y_predict + x
  y_bt <-  y_bt - (beta[1] + beta[2]*xdata$x1 + beta[3]*xdata$x2)
  data1_BT <-  data.frame(x3 = data3$x3,
                          y = y_bt)
  fit_BT <- rlm(y ~ x3 -1, data = data1_BT)$coefficients
  return(fit_BT)
}

B = 1000
bcanon(rres, B, res_rob_rg_x3, beta = rbeta.h, xdata = data3,alpha = c(0.025, 0.975))$confpoints
```

```
##      alpha bca point
## [1,] 0.025  1.296253
## [2,] 0.975  1.339254
```

Confidence Intervals

Mean Response