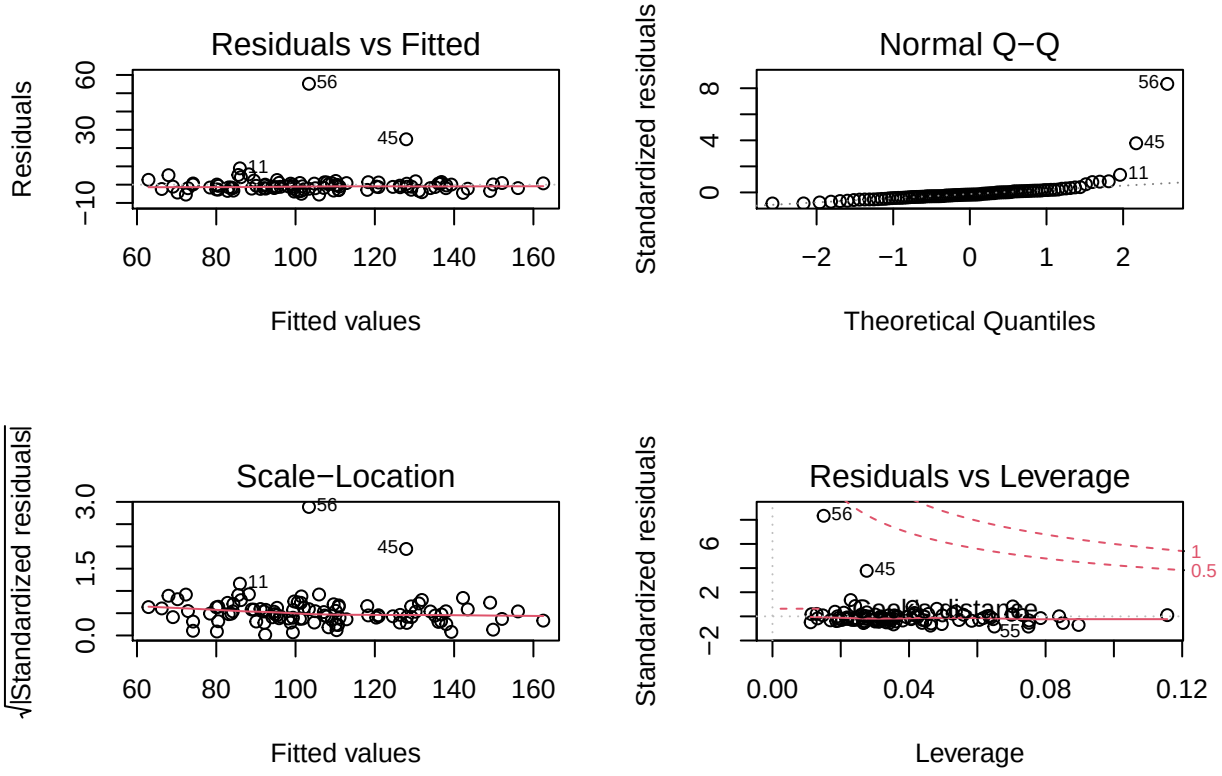
Bootstrap Project: Dataset 3

Arturo Prieto Tirado, Cesar Conejo Villalobos

Introduction

In this project, we study the dataset `data_3.csv` which contains four columns referring to a response variable `y` and 3 covariates `x1`, `x2`, and `x3` for 100 observations. 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 containing several outliers, it can be checked in the following picture that the error residuals are 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 $\beta_1 = 6.29$, (corresponding to the intercept) $\beta_2 = 0.91$, $\beta_3 = 2.02$, and $\beta_4 = 3.99$. In order to correct for the outliers, we build a robust linear regression model `rlm()` from `MASS` package and use the bootstrap to study the significance of the regressors.

Robust regression

(Build a robust linear regression model with the three covariates and use (95%) bootstrap confidence intervals on the regressors' coeffcients to study their significance.)

In this section, we calibrate a robust linear model with the function `rlm()`. From the output, we can see a significant change in the coefficients of the linear `linear model` with respect to the `robust linear model`. First, the values related to the intercept and the variable `x3` decrease in the robust model. On the other hand, the weight assigned to the variables `x1` and `x2` increase considerably. As a result, the first insight that we can get is related to the covariate `x3`, in which the increase of one unit of the variable `x3` decreases from 4 in the linear model to 1.3.

```
model.rlm <- rlm(y ~  x1 + x2 + x3, data = data3, maxit = 200)
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

Additionally, we must consider that the standard error reported in `rml()` for the coefficients is based on asymptotic results. Moreover, the sample size of 100 is relatively small, resulting in not trustworthy estimations of the variability in the coefficient's regression. Therefore, we can explore a 95% bootstrap confidence interval on the regressors' 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 influential observations the bootstrap in pairs can lead to low-quality estimators. For example, in some samples, it is possible 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 residual bootstraping algorithm consists on: 1. Estimate $\hat{\beta}$ from the original dataset. 2. Calculate the approximate errors as $\hat{\epsilon}_i = y_i - \left(\hat{\beta}_0 + \sum_{j=1}^{k} \hat{\beta}_j x_j\right)$ 3. Sample with replacement n elements from $\{\hat{\epsilon}_1, \hat{\epsilon}_2, \ldots, \hat{\epsilon}_n\}$, denoted $\{\hat{\epsilon}_1^*, \hat{\epsilon}_2^*, \ldots, \hat{\epsilon}_n^*\}$ and build the date

$$\left\{\left(\mathbf{x}_1^t, \hat{\beta}_0 + \sum_{j=1}^{k} \hat{\beta}_j x_{1j} + \hat{\epsilon}_1^*\right), \ldots, \left(\mathbf{x}_n^t, \hat{\beta}_0 + \sum_{j=1}^{k} \hat{\beta}_j x_{nj} + \hat{\epsilon}_n^*\right)\right\}$$

4. Use each bootstrap sample built in Step 3. to estimate the parameters of a regression model.

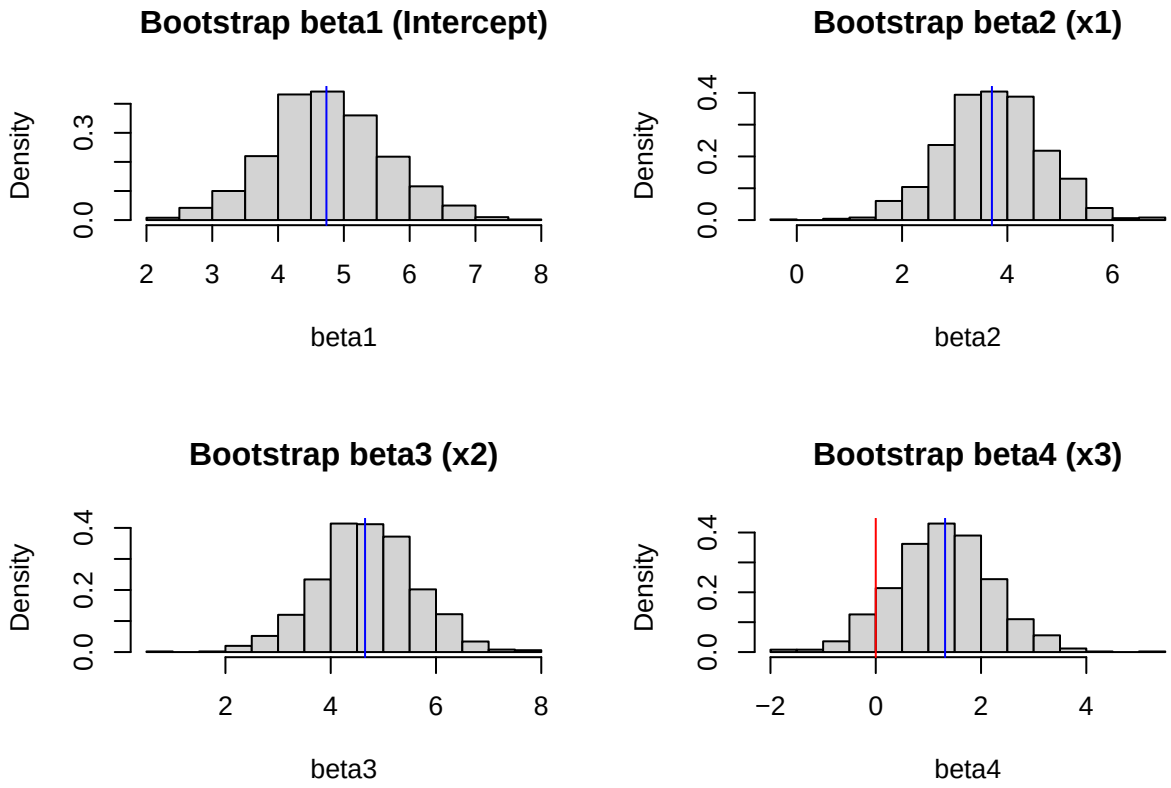
The following code shows how to realize the bootstrap configuration:

```
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 <- 1000
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. we can see how the distributions of β_1 and β_2 are symmetric, unimodal, and takes all values greater than zero. In the case of the distribution

of β_3, we can see notice a slightly right tail distribution, however, all the values are greater than zero. Finally, if we look at the distribution of β_4 corresponding to the variable `x3` we can see a left-skewed distribution taking positive and negative values. As a result, in some of the resampling scenarios, the coefficient β_4 can be not significant.

```
par(mfrow=c(2,2))
hist(coeff.res[1,], main = "Bootstrap beta1 (Intercept)", probability = T, xlab = "beta1")
abline(v = rbeta.h[1], col ="blue")
hist(coeff.res[2,], main = "Bootstrap beta2 (x1)", probability = T, xlab = "beta2")
abline(v = rbeta.h[2], col ="blue")
hist(coeff.res[3,], main = "Bootstrap beta3 (x2)", probability = T, xlab = "beta3")
abline(v = rbeta.h[3], col ="blue")
hist(coeff.res[4,], main = "Bootstrap beta4 (x3)", probability = T, xlab = "beta4")
abline(v = 0, col ="red")
abline(v = rbeta.h[4], col ="blue")
```



Moreover, we can compute the basic bootstrap confidence interval for each coefficient as

$$\mathrm{CI}_\alpha(\theta) = [2\hat{\theta} - F^{-1}_{\hat{\theta}^*}(1-\alpha/2), 2\hat{\theta} - F^{-1}_{\hat{\theta}^*}(\alpha/2)]$$

We can confirm that even in the space corresponding to 95% probability, β_4 overlaps with 0 and negative values.

```
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

(Use backward elimination to select the relevant covariates and provide the chosen regression model.)

As it was shown in the previous section, β_4 is compatible with 0. Since 0 is in the confidence interval, the p-value for the null hypothesis (coefficient not relevant) is greater than 0.05 (because we took 95% confidence). Therefore, the null hypothesis is not rejected. This means that we can consider the effect of the variable x_3 to be not significant and thus, remove it from the model. The final model will then be $y = \beta_1 + \beta_2 x_1 + \beta_3 x_2$, with the remaining coefficients being significantly different than 0, as it is shown in the following section.

Confidence Intervals

(Provide 95% confidence intervals on the regression coeffcients.)

The coefficients for the final model $y = \beta_1 + \beta_2 x_1 + \beta_3 x_2$ are shown in the following table. It can be seen how they are slightly different than the ones obtained for the full model.

```
model.rlm <- rlm(y ~  x1 + x2, data = data3, maxit = 200)
knitr::kable(coef(summary(model.rlm)), digits = 3)
```

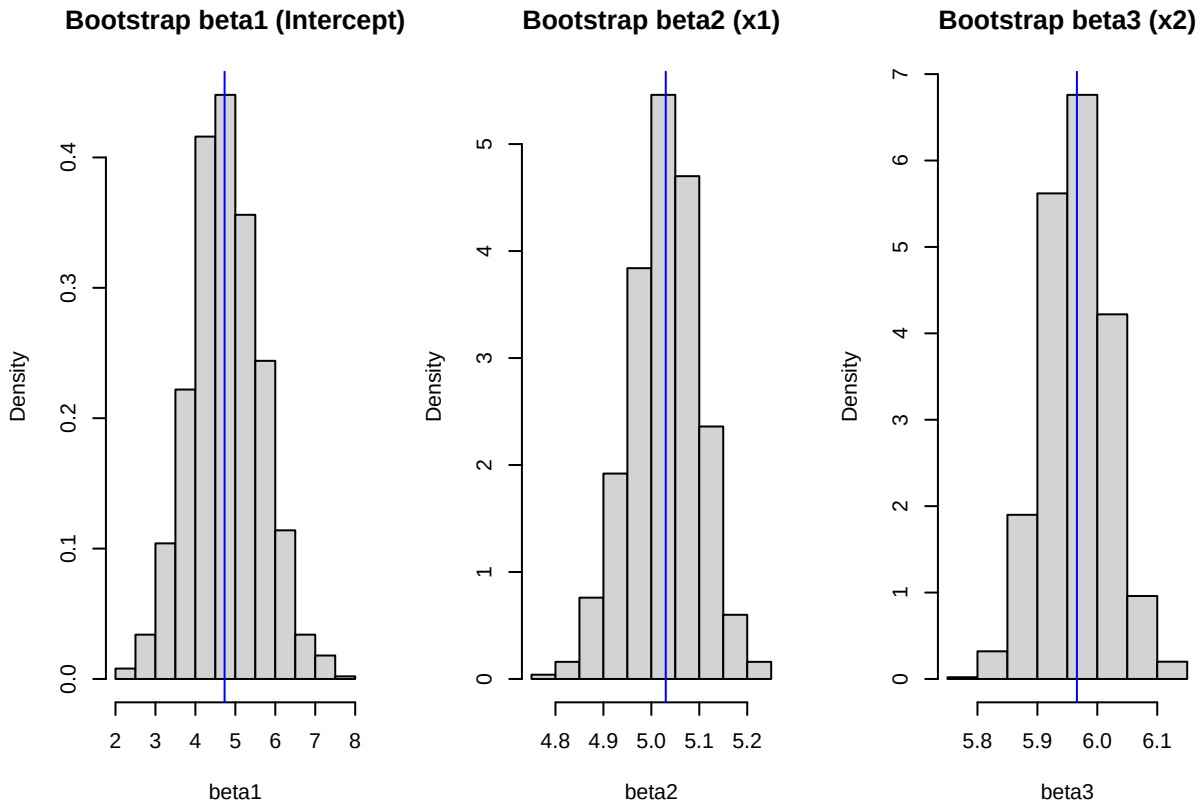
	Value	Std. Error	t value
(Intercept)	4.731	0.898	5.266
x1	5.030	0.076	66.555
x2	5.966	0.059	101.300

The goal now is to obtain confidence intervals for the parameters of the simple model. In order to do this, we make use of the same residual bootstrap fashion used in part one to diminish the effect of outliers and reduce variability.

```
res_rob_rg <- function(x, beta, xdata){
  y_fit  <- beta[1] + beta[2]*xdata$x1 + beta[3]*xdata$x2 + x
  bmodel <- rlm(y_fit ~ x1 + x2, data = xdata)
  return(coef(bmodel))
}
rres    <- model.rlm$residuals
rbeta.h <- coef(model.rlm)
B <- 1000
set.seed(1)
coeff.res <- bootstrap(rres, B, res_rob_rg, beta = rbeta.h, xdata = data3)$thetastar
```

The bootstrap distributions obtained are shown in the following figures.

```r
par(mfrow=c(1,3))
hist(coeff.res[1,], main = "Bootstrap beta1 (Intercept)", probability = T, xlab = "beta1")
abline(v = rbeta.h[1], col ="blue")
hist(coeff.res[2,], main = "Bootstrap beta2 (x1)", probability = T, xlab = "beta2")
abline(v = rbeta.h[2], col ="blue")
hist(coeff.res[3,], main = "Bootstrap beta3 (x2)", probability = T, xlab = "beta3")
abline(v = rbeta.h[3], col ="blue")
```



And the final confidence intervals obtained in the same way as before are:

```r
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))
knitr::kable(rbind(basic_beta1, basic_beta2, basic_beta3), digits = 3)
```

	97.5%	2.5%
basic_beta1	2.927	6.395
basic_beta2	4.891	5.178
basic_beta3	5.857	6.074

It can be seen that none of the coefficients is compatible with 0, as we wanted.

Mean Response

(Build a 95% confidence interval on the mean response when $(x_1; x_2; x_3) = (14; 14; 14)$.)

The main idea now is to use again the same bootstraping in residuals procedure but, instead of returning the coefficients each time and thus getting a bootstrap sample of coefficients as we did before, use this coefficients to predict a response. In order to do that, the simplest model will be used, that is, the model with x_1 and x_2 obtained in the previous parts.

The following code shows how to obtain a bootstrap sample of predicted values when $x_i = 14$ in the bootstraping residuals fashion. The main difference is the use of the `predict` function to calculate $\hat{y} = \hat{\beta}_0 + 14\hat{\beta}_1 + 14\hat{\beta}_2$.

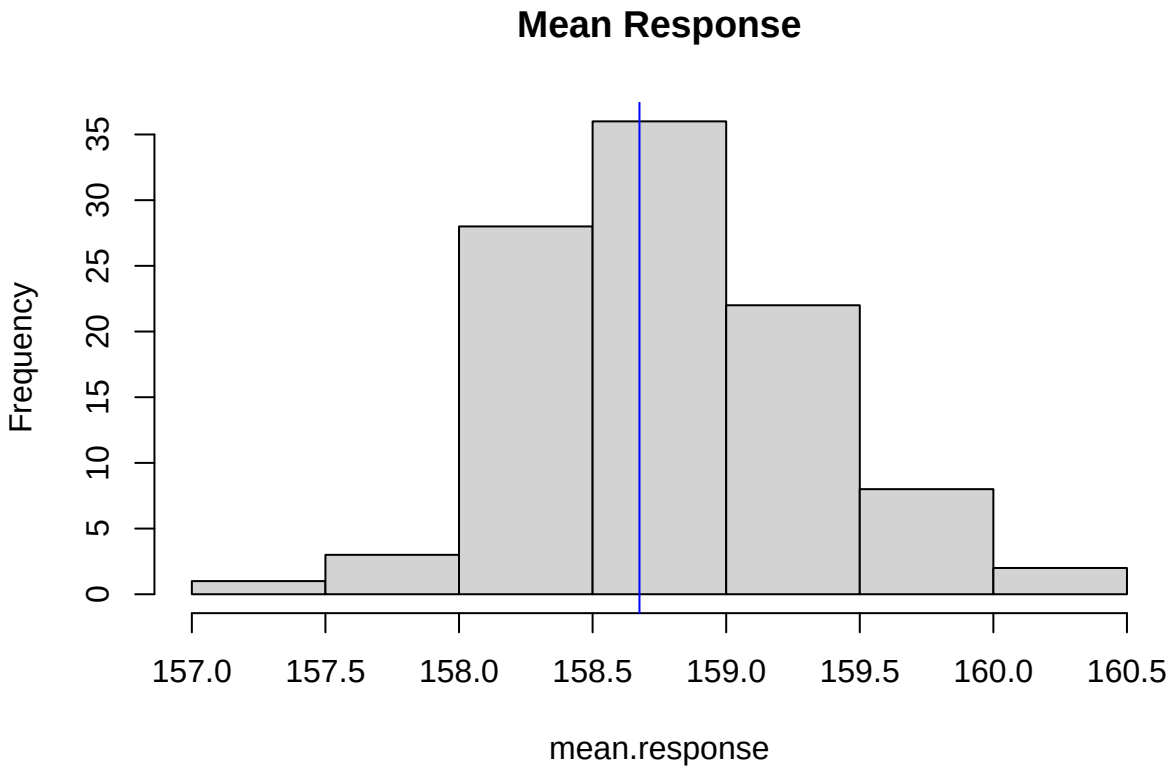
```
# Model with only x1 and x2
model.rlm <- rlm(y ~  x1 + x2, data = data3)
coef(summary(model.rlm))
```

```
##                Value Std. Error   t value
## (Intercept) 4.730836 0.89835999   5.26608
## x1          5.030091 0.07557784  66.55511
## x2          5.965955 0.05889420 101.29954
```

```
fit_value <- predict(model.rlm, newdata = data.frame(cbind(x1 = 14, x2 = 14, x3 = 14)))
fit_value
```

```
##        1
## 158.6755
```

```
res_rob_mean <- function(x,beta,xdata){
  y_fit  <- beta[1] + beta[2]*xdata$x1 + beta[3]*xdata$x2 + x
  bmodel <- rlm(y_fit ~ x1 + x2, data = xdata)
  return(predict(bmodel, newdata = data.frame(cbind(x1 = 14, x2 = 14, x3 = 14))))
}
#take the best model in 3 (change this)
rres    <- model.rlm$residuals
rbeta.h <- coef(model.rlm)
B <- 100
set.seed(1)
mean.response <- bootstrap(rres, B, res_rob_mean, beta = rbeta.h, xdata = data3)$thetastar
hist(mean.response, main = "Mean Response")
abline(v = fit_value, col = "blue")
```




```
# Basic Bootstrap
2*fit_value - quantile(mean.response, c(0.975,0.025))
```

```
##   97.5%    2.5%
## 157.437 159.462
```

The results show a mean response of 158.676 with a confidence interval of [157.437,159.462].