Conducting Bootstrap and Jackknife Analysis from the real dataset

Yoon Joo Cho, Seung Wook Kim

December 5, 2016

Abstract

In this project, we develop a topic on the Bootstrap and Jackknife methods. There is a real data set regarding Boston housing prices whose distribution is non-normal. When we want to fit a multiple linear regression model from the data, normality assumption should be satisfied. In this case, the Bootstrap method can be used to overcome problems of unknown sampling distributions. By using the Bootstrap method, we can find point estimators of coefficients of the fitted linear model and the 95% confidence intervals for the coefficients. Furthermore, we can perform the same work by using the Jackknife method. Based on the results, we can compare Bootstrap with Jackknife.

Keywords: *Non-parametirc Bootstrap, Case Resampling, Residual Resampling, Jackknife, Boston Housing Data, Mutiple Linear Regression, Least Squared Regression Line*

1 Instructions

1.1 Data Set

Here is an explanation with more details about the data set we want to use. The Boston Housing data give information about housing values of Boston from D. Harrison and D.L. Rubenfeld (1978)'s article titled, "Hedonic Prices and the Demand for Clean Air", *Journal of Environmental Economics and Management* 5, 81–102. This data set is available in `R, MASS` package, and the sample size is $n = 506$. There are 5 predictors from the original data set: nitrogen oxides concentration (*nox*), average number of room per house (*rm*), weighted mean of distances to 5 employment centers (*dis*), student-teacher ratio by town (*ptratio*), and lower status of the population in percentage (*lstat*). The response variable (*medv*) is the median value of owner-occupied properties in \$1,000. As the data set includes 5 regressors, using those regressors, we can regress the price of house in the Boston area.

1.2 Our Goal and Procedure

There is Boston Housing Data whose sample size is $n = 506$. For each observation, there are 5 factors which might explain the Boston's housing price. Using this data, we can fit the least squared line to estimate mean value of *medv*, price of Boston's house, as follows;

$$medv = \beta_0 + \beta_1 nox + \beta_2 rm + \beta_3 dis + \beta_4 ptratio + \beta_5 lstat + \epsilon$$

where *nox, rm, dis, ptratio* and *lstat* are predictors. Suppose the above model roughly satisfies the normality assumption. Then, point estimators for coefficients and 95% confidence interval(CI) for them can be investigated.

However, note that this model does not completely satisfy the normality assumption. Since this model is a non-parametric model, non-parametric Bootstrap method can be performed to overcome this problem. After conducting Bootstrap methods, we can have six point estimators for $\beta_0, \beta_1, \beta_2, \beta_3, \beta_4$ and β_5 and 95% CIs for them. Similarly, Jackknife methods can be used to do the above same work. As a result, we have three results for six estimators and 95% CIs for those parameters in each method. Based on the results, there are possible questions that will be discussed. What are the results of t-test, Bootstrap and Jackknife? More specifically, can we convince that resampling methods are good enough to assess the accuracy? How can Jackknife be considered as an approximate to Bootstrap? Does the original data set of sample size $n = 506$ make Jackknife more computational intensive? More specifically, can smaller sample size, for example $n = 25$, work better?

2 Methods

2.1 Bootstrapping Regression

There are two ways of bootstrapping for a regression model. One is based on resampling from the original data, and the other is a residual resampling given fixed-x bootstrap procedure. We fit a regression model with the Boston housing median value as the response variable y and predictors *nox*, *rm*, *dis*, *ptratio*, and *lstat.* We used more than two packages of R for bootstrapping. `bootstrap` package (Efron, B., Tibshirani, R. J., 1993), and `boot` package (Davison, A. C., Hinkley, D. V, 1997) are associated with the subject. `Boot` with a capital "B" is a function in the `car` package (John Fox, Sanford Weisberg, 2016) in the `MASS` package, while the function `boot` in the `boot` package is also used.

2.1.1 Random x or Case Resampling

This Bootstrap method is to deal with predictors as random. Random sample **x** resampling is clearly case resampling of $n = 506$ observations $\mathbf{z_i'} = (y_i, x_{i1}, x_{i2}, \ldots, x_{i5}), i = 1, \ldots, n$ under which the Bootstrap data are

$$\mathbf{z_i'}^* = (y_i, x_{i1}, x_{i2}, \ldots, x_{i5})^*, i = 1, \ldots, n$$

taken independently with equal probabilities n^{-1} from the $\mathbf{z_i'} = (y_i, x_{i1}, x_{i2}, \ldots, x_{i5}), i = 1, \ldots, n$. The function `boot` in the `boot` library takes several arguments.

- `data`: The data as a vector, matrix or data frame. If it is a matrix or data frame, each row is considered as one multivariate observation

- `statistic`: A function which when applied to data returns a vector containing the statistic(s) of interest.

- `R`: The number of Bootstrap replicates. Usually this will be a single positive integer.

We also use the `boot.ci` function to get an confidence interval on each regression coefficient of 95% level of Normal, Percentile, and BCa methods.

2.1.2 Fixed-x resampling residuals

This is model-based re-sampling in linear regression and from obtained fitted values $\mathbf{X}\hat{\boldsymbol{\beta}}$, we take ε_i^b randomly from centered standardized residuals $e_1, \ldots, e_n$ and set

$$y^{b_i} = \mathbf{X}\boldsymbol{\beta} + \varepsilon_i^b, \quad i = 1, \ldots, n$$

Therefore, we can rewrite this as the fixed $\hat{\mathbf{Y}}$ and resample the residuals $\mathbf{e}$, to get

$$\mathbf{Y^b} = \hat{\mathbf{Y}} + \mathbf{e^b}$$

We refer to residual resampling method for `Boot` is slightly modified from this, by using modified residuals $e_i/\sqrt{1-h_{ii}}$, Thus,

$$r_i = \frac{e_i}{\sqrt{1-h_{ii}}} - \bar{r}$$

where, h_{ii} is the i-leverage in hat-matrix value for the i the observation. (Davison, A. C., Hinkley, D. V, 1997) (John Fox, 2011)

2.1.3 Comparing residuals between t-test and Bootstrap methods

In order to check if the resampling method, Bootstrap can give us more accuracy than t-test method can do, we will check the distribution of residuals. Let $y = medv$. From each method, we had two fitted line,

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 nox + \hat{\beta}_2 rm + \hat{\beta}_3 dis + \hat{\beta}_4 ptratio + \hat{\beta}_5 lstat,$$

$$\hat{y}^b = \hat{\beta}_0^{\ b} + \hat{\beta}_1^{\ b} nox + \hat{\beta}_2^{\ b} rm + \hat{\beta}_3^{\ b} dis + \hat{\beta}_4^{\ b} ptratio + \hat{\beta}_5^{\ b} lstat,$$

where $\hat{\beta}_i^{\ b}$ is point estimator of β_i using Bootstrap. When we use the t-test method, residuals can be written as $e_i = y_i - \hat{y}$ and when we use the Bootstrap, residuals can be written as $e_i^b = y_i - \hat{y}^b$. The difference between a distribution of e_i and e_i^b can be seen from the plot between residuals and fitted values and normal qq-plot.

2.2 Jackknife Regression

Similar to the Bootstrap method, Jackknife methods can be used to find point estimators and 95% CIs for coefficients. We need to use a package `bootstrap` and a function `jackknife` from the `R`. In addition, in order to check whether a smaller sample size is better or not, we will compare the case when $n = 25$ with the original case when $n = 506$. For each sample size, using Jackknife like before, point estimators for coefficients and estimate of standard error for coefficients will be calculated.

3 Results

Following our methods, for each method, we found the point estimators of coefficients, $\hat{\beta}_i$, the estimated standard errors of the point estimators of coefficients, $\hat{se}(\hat{\beta}_i)$ and 95% CIs of coefficients. When we use t-test and Bootstrap methods, the result for the $\hat{\beta}_i$, $\hat{se}(\hat{\beta}_i)$ can be shown in Table 1 and the 95% CIs for β_i can be shown in Table 2. In addition, when we use the Jackknife method, we could get each result displayed in Table 3 and Table 4 respectively. The tables we explained are as follows;

Table 1: $\hat{\beta}_i$ and $\hat{se}(\hat{\beta}_i)$ in each method

Coefficients	Origin. Data		Bootstrap			
	Estimate	Standard Error	Estimate		Standard Error	
			Random-x	Fixed-x	Random-x	Fixed-x
β_0	37.50	4.63	37.78	37.61	6.15	4.52
nox	-17.00	3.26	-17.90	-17.15	3.26	3.27
rm	4.16	0.41	4.14	4.17	0.72	0.41
dis	-1.18	0.17	-1.19	-1.19	0.18	0.17
pr	-1.05	0.11	-1.05	-1.05	0.11	0.11
ls	-0.58	0.48	-0.58	-0.58	0.09	0.05

Table 2: 95% CIs for β_i in each method

Coef.	Orig. Data	Bootstrap-Random-x			Fixed-x
		Norm.	Perc.	BCa	BCa
β_0	(28.44, 46.56)	(25.73, 49.82)	(24.94, 49.21)	(25.84, 50.32)	(28.28, 46.59)
nox	(-24.40, -11.59)	(-24.49, -11.70)	(-24.26, -11.27)	(-24.54, -11.80)	(-24.11, -11.10)
rm	(3.35, 4.97)	(2.72, 5.56)	(2.77, 5.57)	(2.65, 5.45)	(3.37, 4.97)
dis	(-1.52, -0.85)	(-1.55, -0.83)	(-1.57, -0.84)	(-1.58, -0.85)	(-1.51, -0.86)
pr	(-1.27, -0.82)	(-1.27, -0.83)	(-1.26, -0.81)	(-1.26, -0.81)	(-1.29, -0.82)
ls	(-0.68, -0.49)	(-0.75, -0.41)	(-0.75, -0.42)	(-0.76, -0.42)	(-0.67, -0.49)

Table 3: $\hat{\beta}_i$ and $\hat{se}(\hat{\beta}_i)$ of Jackknife method

Coefficients	Jackknife			
	Estimate		Standard Error	
	$n = 25$	$n = 506$	$n = 25$	$n = 506$
β_0	19.86	37.81	22.57	6.35
nox	-16.35	-18.00	20.91	3.29
rm	7.07	4.13	3.45	0.75
dis	-0.31	-1.187	0.92	0.19
pr	-1.49	-1.0469	0.77	0.12
ls	-0.54	-0.58	0.27	0.09

Table 4: 95% CIs for β_i of Jackknife method

Coefficients	Jackknife	
	$n = 25$	$n = 506$
β_0	(-27.99, 60.69)	(25.01, 49.98)
nox	(-53.50,28.64)	(-24.46, -11.53)
rm	(0.15,14.06)	(2.68, 5.64)
dis	(-1.96, 1.66)	(-1.55, -0.82)
pr	(-2.97, 0.05)	-1.28, -0.81)
ls	(-1.04,0.00)	(-0.76, -0.40)

3.1 Discussion from Results : Bootstrap

In this section, we can answer these following questions; What are the results of t-test, Bootstrap and Jackknife? More specifically, can we convince that resampling methods are good enough to assess the accuracy?

From the Table 1, when we consider the sign of $\hat{\beta}_i$, the estimate of mean *medv*, which is response variable of the model, has the negative relationship with the variables of *nox*, *dis*, *pr* and *ls* and the positive relationship with *rm* variable in both methods, t-test and Bootstrap. Also, the level of coefficients between the original sample value and the bootstrap estimates of both the random x resampling and the fixed-x residual resampling methods are very close as shown in the result Table 1. We can observe the method of the fixed-x residual resampling turns out very close values of on the standard errors of each coefficients of$4.52, 3.27, 0.41, 0.17, 0.11$ and 0.05 to the standard errors of the original sample values of $4.63.3.26, 0.41, 0.17, 0.11$ and 0.48, *intercept, nox, m, dis, pr* and *ls* respectively. The bootstrap can be used to produce approximate confidence intervals in an automatic way. `boot.ci` function provides confidence intervals using three methods which are Normal, Percentile and BC_a for the regression models. From `Boot` function, we can also find confidence interval for the fixed-x residual bootstrap method. In our median housing price of Boston model, we can observe both of fixed- x and residual resampling method has very similar confidence interval with the before one, for example, 95% CI's for β_0, (28.44, 46.56) and (28.28,46.59), which are the original sample confidence interval and the fixed-x residual resampling respectively.

Particularly, the fixed-x residual resampling claims by force the assumption that the errors are identically distributed by resampling residuals from the distribution. It means that unusual observations are not carried over into the resampled data sets for the fixed-x residual resampling. So, in the case of the model which is not satisfied with the normal assumption, or the case the existence of non-constant error variance, or outliers, the fixed-x residual resampling will be preferred. (John Fox, 2011)

For now, we could compare the result between the methods of original data and with resampling method, Bootstrap. From the Table 1 and 2, the values of $\hat{\beta}_i$ between those methods are very similar, for example,

$$\boldsymbol{\beta}_{original} = (37.5, -17.0, 4.16, -1.18, -1.05, -0.58)^T \quad and$$

$$\boldsymbol{\beta}^b = (37.78, -17.9, 4.14, -1.19, -1.05, -0.58)^T$$

are quite similar. Then, can we say that the resampling method is better to get accuracy? Even though they look very similar, we can consider residual plot and normal qq-plot to check if there is any difference between two methods. From the

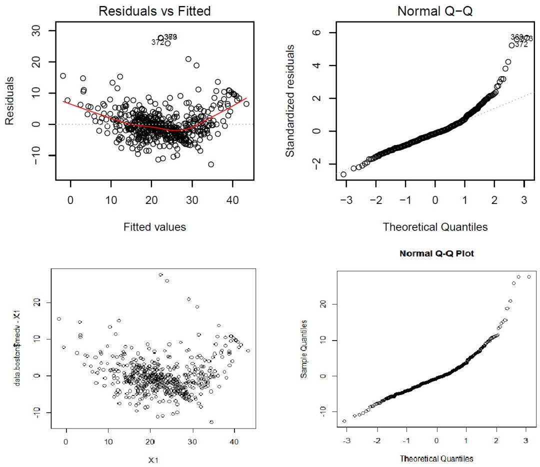

Figure 1: Residual plot and Normal qq-plot

Figure 1, there is no large difference between them when considering residual plot and normal qq-plot. However, when we see the normal qq-plot, even though both of them are skewed to the right, we can find the normal qq-plot of Bootstrap is slightly closer to the normal. Therefore, the fitted line of the Bootstrap is slightly better than the line without resampling method.

3.2 Discussion from Results : Jackknife

From the Table 1 and Table 3, when $n = 506$, we can see that the value of $\hat{\beta}_i$ are similar between Bootstrap and Jackknife methods. For example,

$$\boldsymbol{\beta}^b = (37.78, -17.9, 4.14, -1.19, -1.05, -0.58)^T and$$

$$\boldsymbol{\beta}^J = (37.81, -18, 4.13, -1.187, -1.0469, -0.58)^T$$

are quite similar. Based on this result, we can say that Jacknife can be considered as an approximation of Bootstrap.

From the Jackknife method, all of estimate standard error for coefficients, $\hat{se}(\hat{\beta}_i)$ under the $n = 25$ are much larger than those under the $n = 506$. For instance, $\hat{se}(\hat{\beta}_1)$ of $n = 25$, the estimate of standard error for $\hat{\beta}_1$, which is 20.91, is 6 times larger than $\hat{se}(\hat{\beta}_1)$ of $n = 506$, which is 6.35. Furthermore, even though we don't know the exact value of $\beta_0, \beta_1, \cdot\cdot\cdot, \beta_5$ of the population, when $n = 25$, point estimators of coefficient which came from the Jackknife method is too far from the those of original data model. We can assume the distribution of original data is not too much different from that of population because a sample size is not too small even though we don't know the shape of distribution of population. Then, the above result indicates that when we use Jackknife method if n is smaller, Jackknife method does not work well because when $n = 25$, the output is too much different than the output of the original data. In other words, we can conclude that unless n is large, Jackknife is less computationally intensive.

4 Further Study

The essential point of the resampling is how we can hold the relationship between the selection of the original sample from the population and the selection of each Bootstrap sample form the original sample. One of the benefits from the Bootstrap or Jackknife is that it can provide correct statistic inferences. From this rigorous method, Davison and Hinkley (Davison, A. C., Hinkley, D. V, 1997) developed the hypothesis test in the regression context. However, because testing using the Bootstrap or Jackknife is potentially complex, we leave this method for further discussion. Another study can be in use of `Boot` package for the generalized linear models (GLM), because `Boot` doesn't allow to use `method="residual"` for GLM. (Davison, A. C., Hinkley, D. V, 1997). We may leave for further discussion and try to build a new `R` function for GLM.

5 R codes

This is a R code for finding least squared regression line.

```
 library(MASS)
> data(Boston)
> nox<-Boston[,5]
> rm<-Boston[,6]
> dis<-Boston[,8]
> pr<-Boston[,11]
> ls<-Boston[,13]
> medv<-Boston[,14]
> data.boston<- data.frame(medv, nox, rm, dis, pr, ls)
>
> lm.boston<-lm(medv~nox+rm+dis+pr+ls,data=data.boston)
> summary(lm.boston)
Call:
lm(formula = medv ˜ nox + rm + dis + pr + ls, data = data.boston)

Residuals:
     Min       1Q   Median       3Q      Max
-12.7765  -3.0186  -0.6481   1.9752  27.7625

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  37.49920    4.61295   8.129 3.43e-15 ***
nox         -17.99657    3.26095  -5.519 5.49e-08 ***
rm            4.16331    0.41203  10.104  < 2e-16 ***
dis          -1.18466    0.16842  -7.034 6.64e-12 ***
pr           -1.04577    0.11352  -9.212  < 2e-16 ***
ls           -0.58108    0.04794 -12.122  < 2e-16 ***

Residual standard error: 4.994 on 500 degrees of freedom
Multiple R-squared:  0.7081,Adjusted R-squared:  0.7052
F-statistic: 242.6 on 5 and 500 DF,  p-value: < 2.2e-16
```

This is a R code for the Bootstrap method.

```
> boot.huber<-function(data,indices,maxit=20){
+  data<-data[indices,]
+  mod<-lm(medv˜nox+rm+dis+pr+ls,data=data, maxit=maxit)
+  coefficients(mod)
+ }
>
> library(boot)
> boston.boot<-boot(data.boston, boot.huber,1999,maxit=100)
There were 50 or more warnings (use warnings() to see the first 50)
> boston.boot

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
```

```
boot(data = data.boston, statistic = boot.huber, R = 1999, maxit = 100)


Bootstrap Statistics :
       original       bias    std. error
t1*  37.4991961 -0.277448662  6.14628473
t2* -17.9965715  0.097974099  3.26387578
t3*   4.1633074  0.022944043  0.72419450
t4*  -1.1846623  0.002925215  0.18363489
t5*  -1.0457738  0.003852122  0.11415168
t6*  -0.5810836 -0.001905865  0.08705555
>
> ci.b0<-boot.ci(boston.boot,index=1,type=c("norm","perc","bca"))
> ci.nox<-boot.ci(boston.boot,index=2,type=c("norm","perc","bca"))
> ci.rm<-boot.ci(boston.boot,index=3,type=c("norm","perc","bca"))
> ci.dis<-boot.ci(boston.boot,index=4,type=c("norm","perc","bca"))
> ci.pr<-boot.ci(boston.boot,index=5,type=c("norm","perc","bca"))
> ci.ls<-boot.ci(boston.boot,index=6,type=c("norm","perc","bca"))
> normal.ci<-cbind(c(ci.b0$normal), c(ci.nox$normal),c(ci.rm$normal),
c(ci.dis$normal),c(ci.pr$normal),c(ci.ls$normal))
> rownames(normal.ci) <- c("Level.Normal","Lower","Upper")
> colnames(normal.ci) <- c( "B0","B1","B2","B3","B4","B5")
> print(normal.ci)
                   B0        B1       B2         B3         B4         B5
Level.Normal  0.95000   0.95000 0.950000  0.9500000  0.9500000  0.9500000
Lower        25.73015 -24.49162 2.720968 -1.5475053 -1.2733591 -0.7498035
Upper        49.82314 -11.69747 5.559758 -0.8276697 -0.8258928 -0.4085520
>
> perc.ci<-cbind(c(ci.b0$percent),c(ci.nox$percent),c(ci.rm$percent),
c(ci.dis$percent),c(ci.pr$percent),c(ci.ls$percent))[-c(2,3),]
> rownames(perc.ci) <- c("Level.Percent","Lower","Upper")
> colnames(perc.ci) <- c( "B0","B1","B2","B3","B4","B5")
> print(perc.ci)
                    B0        B1       B2         B3         B4         B5
Level.Percent  0.95000   0.95000 0.950000  0.9500000  0.9500000  0.9500000
Lower         24.94099 -24.26457 2.773954 -1.5655196 -1.2593001 -0.7490329
Upper         49.21359 -11.27481 5.570935 -0.8355352 -0.8064689 -0.4161756
> bca.ci<-cbind(c(ci.b0$bca), c(ci.nox$bca),c(ci.rm$bca),c(ci.dis$bca),
c(ci.pr$bca),c(ci.ls$bca))[-c(2,3),]
> rownames(bca.ci) <- c("Level.BCa","Lower","Upper")
> colnames(bca.ci) <- c( "B0","B1","B2","B3","B4","B5")
> print(bca.ci)
                B0        B1       B2         B3         B4         B5
Level.BCa  0.95000   0.95000 0.950000  0.9500000  0.9500000  0.9500000
Lower     25.84048 -24.54421 2.650669 -1.5838713 -1.2599990 -0.7563293
Upper     50.32128 -11.80295 5.453341 -0.8531867 -0.8081859 -0.4224423

#. Bootstrap that resamples the residuals instead of the whole observations can
also be used.
> library(car)
> summary(duncan.fix.boot<-Boot(lm.boston,R=1999,method="residual"))
               R  original   bootBias   bootSE   bootMed
(Intercept) 1999  37.49920 -0.1046049 4.521593  37.48841
nox         1999 -17.99657  0.1526540 3.269043 -17.94022
rm          1999   4.16331 -0.0023713 0.410380   4.16111
dis         1999  -1.18466  0.0046584 0.167546  -1.18416
pr          1999  -1.04577  0.0020244 0.114007  -1.04517
ls          1999  -0.58108 -0.0016091 0.047706  -0.58302

> confint(duncan.fix.boot)
Bootstrap quantiles, type =  bca

                  2.5 %      97.5 %
(Intercept)  28.2762510  46.5859385
nox         -24.1145012 -11.0985037
rm            3.3749515   4.9654761
dis          -1.5114378  -0.8609910
pr           -1.2861148  -0.8234678
ls           -0.6743186  -0.4897070

#.
> data.boston1<-data.frame(nox,rm,dis,pr,ls)
> N1<- c(-17.9942628,4.1101246,-1.1865082,-1.0446642,-0.5851452)
>
> f<-function(x)
+ {x*N1}
```

```
> X<-t(apply(data.boston1,1,f))
> f1<- function(x)
+ {sum(x)+37.8956220}
> X1 <- apply(X, 1, f1)
> plot(X1,data.boston$medv-X1)
> qqnorm(data.boston$medv-X1)
```

This is a R code for the Jackknife method.

```
> library(MASS)
> nox <- Boston[,5]
> rm <- Boston[,6]
> dis <- Boston[,8]
> pr <- Boston[,11]
> ls <- Boston[,13]
> medv <- Boston[,14]
>
> Boss<-cbind(medv,nox,rm,dis,pr,ls)
> B<-cbind(1:506, Boss)
> data <- B[,1]
>
> f.nox<- function(x){
+   model<-lm(B[x,2]~B[x,3]+B[x,4]+B[x,5]+B[x,6]+B[x,7])
+   model$coefficient[2]}
>
> f.rm<- function(x){
+   model<-lm(B[x,2]~B[x,3]+B[x,4]+B[x,5]+B[x,6]+B[x,7])
+   model$coefficient[3]}
>
> f.dis<- function(x){
+   model<-lm(B[x,2]~B[x,3]+B[x,4]+B[x,5]+B[x,6]+B[x,7])
+   model$coefficient[4]}
>
> f.pr<- function(x){
+   model<-lm(B[x,2]~B[x,3]+B[x,4]+B[x,5]+B[x,6]+B[x,7])
+   model$coefficient[5]}
>
> f.ls<- function(x){
+   model<-lm(B[x,2]~B[x,3]+B[x,4]+B[x,5]+B[x,6]+B[x,7])
+   model$coefficient[6]}
>
> f.b0<- function(x){
+   model<-lm(B[x,2]~B[x,3]+B[x,4]+B[x,5]+B[x,6]+B[x,7])
+   model$coefficient[1]}
>
> library(bootstrap)
> myjack.f <- function(x){
+   jack.nox <- jackknife(x, f.nox)
+   jack.rm <- jackknife(x, f.rm)
+   jack.dis <- jackknife(x, f.dis)
+   jack.pr <- jackknife(x, f.pr)
+   jack.ls <- jackknife(x, f.ls)
+   jack.b0 <- jackknife(x, f.b0)
+
+   mean.b0<-mean(jack.b0$jack.values)
+   mean.nox<-mean(jack.nox$jack.values)
+   mean.rm<-mean(jack.rm$jack.values)
+   mean.dis<-mean(jack.dis$jack.values)
+   mean.pr<-mean(jack.pr$jack.values)
+   mean.ls<-mean(jack.ls$jack.values)
+
+   bias.b0<-jack.b0$jack.bias
+   bias.nox<-jack.nox$jack.bias
+   bias.rm<-jack.rm$jack.bias
+   bias.dis<-jack.dis$jack.bias
+   bias.pr<-jack.pr$jack.bias
+   bias.ls<-jack.ls$jack.bias
+
+   se.b0<-jack.b0$jack.se
+   se.nox<-jack.nox$jack.se
+   se.rm<-jack.rm$jack.se
+   se.dis<-jack.dis$jack.se
+   se.pr<-jack.pr$jack.se
+   se.ls<-jack.ls$jack.se
+
+   results <- rbind( c( mean.b0, mean.nox, mean.rm, mean.dis, mean.pr, mean.ls),
+     c(bias.b0, bias.nox, bias.rm, bias.dis, bias.pr, bias.ls),
       c(se.b0,se.nox,se.rm,se.dis,se.pr,se.ls))
+   colnames(results) <- c("B0","B1","B2","B3","B4","B5")
+   rownames(results) <- c( "Est","bias","se")
+
+   CI.b0 <- c(mean.b0-qt(0.975,505)*se.b0, mean.b0+qt(0.975,505)*se.b0)
+   CI.nox <- c(mean.nox-qt(0.975,505)*se.nox, mean.nox+qt(0.975,505)*se.nox)
+   CI.rm <- c(mean.rm-qt(0.975,505)*se.rm, mean.rm+qt(0.975,505)*se.rm)
+   CI.dis <- c(mean.dis-qt(0.975,505)*se.dis, mean.dis+qt(0.975,505)*se.dis)
+   CI.pr <- c(mean.pr-qt(0.975,505)*se.pr, mean.pr+qt(0.975,505)*se.pr)
+   CI.ls <- c(mean.ls-qt(0.975,505)*se.ls, mean.ls+qt(0.975,505)*se.ls)
+
+   results2 <- cbind(c(CI.b0),c(CI.nox),c(CI.rm),c(CI.dis),c(CI.pr),c(CI.ls))
+   colnames(results2) <- c("B0","B1","B2","B3","B4","B5")
+   rownames(results2) <- c( "Lower","Upper")
+
+   print(results)
+   print(results2)
+ }
>
> myjack.f(data)
             B0            B1        B2          B3          B4          B5
Est  37.4985856 -17.996566235 4.1633806 -1.184657498 -1.045771476 -0.581079481
bias -0.3082935   0.002654018 0.0369517  0.002416203  0.001182984  0.002080027
se    6.3548753   3.290271301 0.7500144  0.185504387  0.117677890  0.090862674
            B0        B1       B2         B3         B4         B5
Lower 25.01334 -24.46087 2.689848 -1.5491129 -1.2769700 -0.7595949
Upper 49.98384 -11.53226 5.636913 -0.8202021 -0.8145729 -0.4025641
>
> set.seed(7)
> mysample <- sample(data, 25, replace=FALSE)
> myjack.f(mysample)
            B0         B1        B2         B3          B4          B5
Est  16.347759 -12.427031 7.2896451 -0.1472991 -1.45996020 -0.52178163
bias -3.509645   3.920147 0.2212455  0.1632556 -0.03441305 -0.01396542
se   22.569114  20.904865 3.4482833  0.9223861  0.76852592  0.26599138
             B0        B1         B2        B3          B4           B5
Lower -27.99316 -53.49825  0.5148972 -1.959486 -2.96986204 -1.044367618
Upper  60.68868  28.64418 14.0643929  1.664888  0.04994164  0.000804359
```

References

Davison, A. C., Hinkley, D. V (1997). *Bootstrap Methods and their Application.* Cambridge University Press.

Efron, B., Tibshirani, R. J. (1993). *An Introduction to the Bootstrap.* New York; Chapman & Hall.

John Fox, S. W. (2011). *An R Companion to Applied Regression.* Sage, Thousand Oaks CA, second edition.

John Fox, Sanford Weisberg (2016). *Companion to Applied Regression.*