P8160 Homework 4: Bootstrapping

Renjie Wei

4/11/2022

In this homework, we require the use of parallel computing codes for your implementations.

```
library(parallel)
library(foreach)
```

```
## Warning: package 'foreach' was built under R version 4.0.3
```

```
library(doParallel)
```

```
## Warning: package 'doParallel' was built under R version 4.0.5
```

```
## Loading required package: iterators
```

```
## Warning: package 'iterators' was built under R version 4.0.3
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.5
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
## Warning: package 'tibble' was built under R version 4.0.5
```

```
## Warning: package 'tidyr' was built under R version 4.0.5
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
```

```
## Warning: package 'forcats' was built under R version 4.0.5
```

```
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x purrr::accumulate() masks foreach::accumulate()
## x tidyr::expand()     masks Matrix::expand()
## x dplyr::filter()     masks stats::filter()
## x dplyr::lag()        masks stats::lag()
## x tidyr::pack()       masks Matrix::pack()
## x dplyr::select()     masks MASS::select()
## x tidyr::unpack()     masks Matrix::unpack()
## x purrr::when()       masks foreach::when()
```

Problem 1: a randomized trial on an eye treatment

An ophthalmologist designed a randomized clinical trial to evaluate a new laser treatment in comparison to the traditional one. The response is visual acuity, measured by the number of letters correctly identified in a standard eye test. 20 patients have both eyes eligible for laser treatment. The ophthalmologist randomized the two laser treatments (new vs traditional) to the two eyes of those patients (i.e. one eye received the new laser treatment and the other receive traditional laser treatment). Another 20 patients had only one suitable eye, so they received one treatment allocated at random.So we have a mixture of paired comparison and two-sample data.

```
> blue <- c(4,69,87,35,39,79,31,79,65,95,68,
           62,70,80,84,79,66,75,59,77,36,86,
           39,85,74,72,69,85,85,72)
> red <-c(62,80,82,83,0,81,28,69,48,90,63,
        77,0,55,83,85,54,72,58,68,88,83,78,
        30,58,45,78,64,87,65)
> acui<-data.frame(str=c(rep(0,20),
            rep(1,10)),red,blue)
```

Answer the following question:

(1) The treatment effect of the new laser treatment is defined as

$$E(Y \mid \text{trt} = \text{new}) - E(Y \mid \text{trt} = \text{traditional}).$$

Estimate the treatment effect using the collected data.

(2) Use bootstrap to construct 95 % confidence interval of the treatment effect. Describe your bootstrap procedure, and what is your conclusion from the bootstrap CI?

Problem 1.1

```r
blue <- c(4,69,87,35,39,79,31,79,65,95,68,
           62,70,80,84,79,66,75,59,77,36,86,
           39,85,74,72,69,85,85,72)
red <- c(62,80,82,83,0,81,28,69,48,90,63,
        77,0,55,83,85,54,72,58,68,88,83,78,
        30,58,45,78,64,87,65)
acui <- data.frame(str = c(rep(0,20),rep(1,10)),red,blue)
```

The treatment effect is defined as $E(Y \mid \text{trt} = \text{new}) - E(Y \mid \text{trt} = \text{traditional})$. So we calculate the raw treatment effect based on the data. Let blue laser be the new treatment.

```
raw_trt_eff <- mean(blue) - mean(red)
```

The raw treatment effect (the observed value of mean difference) is 3.067.

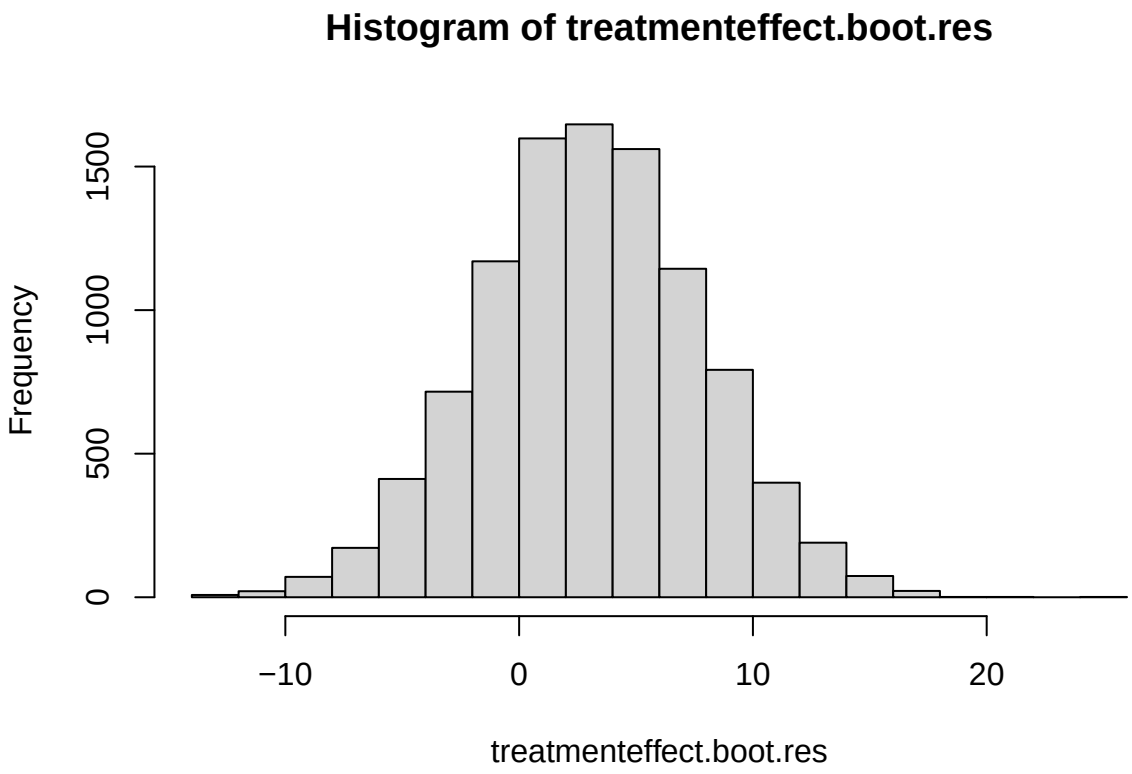
However, raw treatment is not a proper way to estimate the mean difference. I am going to use bootstrap sample to estimate the mean difference and its standard error.

Since there are paired structures in our data, when doing bootstrap, we're going to preserve this structure, so instead of bootstrap each observation, we bootstrap subjects for paired data. For un-paired data, we use the simple bootstrap.

```
set.seed(2022)
# return whole bootstrap sample for future use
pairedboot <- function(paired, unpaired, nboot = 2000){
    numCores <- detectCores()
    registerDoParallel(numCores)
    # parallel computing implementation using foreach
        res <- foreach(icount(nboot), .combine=rbind) %dopar% {
           # bootstrap for paired data
           subject <- nrow(paired)
           pairedboot.ind <- sample(subject, subject, replace = T)
           pairedsamp <- paired[pairedboot.ind,]
           pairedboot.trt <- pairedsamp$blue
           pairedboot.ctrl <- pairedsamp$red
           # bootstrap for unpaired data
           unpaired.trt <- unpaired$blue
           unpaired.ctrl <- unpaired$red
           unpairedboot.trt <- sample(unpaired.trt, replace = T)
           unpairedboot.ctrl <- sample(unpaired.ctrl, replace = T)
           # combine two parts of bootstrap
           boot.trt <- c(pairedboot.trt, unpairedboot.trt)
           boot.ctrl <- c(pairedboot.ctrl, unpairedboot.ctrl)
           # b-th bootstrap estimate of treatment effect
           mean(boot.trt) - mean(boot.ctrl)
        }
        return(res)
}


acui.paired <- acui[which(acui$str == 0),]
acui.unpaired <- acui[which(acui$str == 1),]

system.time({treatmenteffect.boot.res <- pairedboot(acui.paired, acui.unpaired, 1e4)})
```

```
##    user  system elapsed
##    3.61    0.89    5.33
```

```
treatmenteffect.boot.se <- sqrt(var(treatmenteffect.boot.res))


hist(treatmenteffect.boot.res)
```



From 10000 bootstrap sample, we have a estimation of the treatment effect is 3.048 with a standard error 4.705.

Problem 1.2

The bootstrap method for this problem is discussed in Problem 1.1 above.

And the implementation of this paired bootstrap is shown in the `pairedboot()` function.

The $100(1-\alpha)\%$ confidence limits for the basic bootstrap confidence interval are

$$(2\hat{\theta} - \hat{\theta}^*_{1-\alpha/2}, 2\hat{\theta} - \hat{\theta}^*_{\alpha/2})$$

```
set.seed(2022)
alpha = 0.05
boot.ci <- function(boot_est, alpha, raw_est){
    # use quantile function to get the upper and lower bound
    qt <- quantile(boot_est, c(alpha/2, 1-alpha/2), type = 1)
    names(qt) <- rev(names(qt))
    CI <- rev(2 * raw_est - qt)
    return(CI)
}

boot.CI <- boot.ci(treatmenteffect.boot.res, 0.05, raw_trt_eff)
t(boot.CI) %>%  knitr::kable(digits = 3, caption = "95% confidence interval of bootstrap estimate of tr
```

Table 1: 95% confidence interval of bootstrap estimate of treatment effect

2.5%	97.5%
-6.167	12.367

The 95% confidence interval of the treatment effect $\hat{\theta}$ is shown in the table 1. Since the confidence interval contains 0, so we conclude that at 95% confident level, we cannot say that there is significant difference in treatment effect between the newer treatment and the older one.

Problem 2

The Galaxy data consist of the velocities (in km/sec) of 82 galaxies from 6 well-separated conic sections of an unfilled survey of the Corona region. The structure in the distribution of velocities corresponds to the spatial distribution of galaxies in the far universe. In particular, a multimodal distribution of velocities indicates a strong heterogeneity in the spatial distribution of the galaxies and thus is seen as evidence for the existence of voids and super clusters in the far universe.

Statistically, the question of multimodality can be formulated as a test problem

$$H_0 : n_{\text{mode}} = 1 \quad \text{vs} \quad H_a : n_{\text{mode}} \geq 1$$

where $ n_{mode} $ is the number of modes of the density of the velocities.

Considered nonparametric kernel density estimates

$$\widehat{f}_{K,h}(x) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{x - X_i}{h})$$

It can be shown that the number of modes in $\widehat{f}_{K,h}(x)$ decreases as h increase. Let H_1 be the minimal bandwidth for which $\widehat{f}_{K,H_1}(x)$ is unimodal. In the galaxy data, $h_1 = 3.05$

Since multimodal densities need more smoothing to become unimodal, the minimal bandwidth H_1 can be used as a test statistic, and one reject the null hypothesis if

$$\text{Prob}(H_1 > h_1) \leq \alpha$$

To evaluating the distribution of H_1 under the null, one could use the following bootstrap algorithm

1. draw B bootstrap samples if size n from $\widehat{f}_{K,h_1}(x)$
2. for each bootstrap, find $h_1^{*(b)}$, the smallest h for which this bootstrap sample has just 1 mode
3. approximate p-value of test is $\frac{\#h_1^{*(b)} > h_1}{B}$

Implement the algorithm above in R, apply it to the galaxy data, and report your findings. You may find the following R codes helpful.

```
library(MASS)
data(galaxies)
plot(density(galaxies/1000, bw=1.5))
plot(density(galaxies/1000, bw=3.5))
```

```
#calculate the number of modes in the density
  den <- density(galaxies/1000, bw=1.5)
  den.s <- smooth.spline(den$x, den$y, all.knots=TRUE, spar=0.8)
  s.1 <- predict(den.s, den.s$x, deriv=1)
  nmodes <- length(rle(den.sign <- sign(s.1$y))$values)/2
```

Problem 2

$$\widehat{f}_{K,h}(x) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x - X_i}{h})$$

Problem 3 (the breast cancer sutdy):

The data *breast-cancer2.csv* have 569 patients. The first column **ID** lables individual breast tissue images; The second column **Diagnonsis** indentifies if the image is coming from cancer tissue or benign cases (M=malignant, B = benign). There are 357 benign and 212 malignant cases. The other 10 columns correspond to mean of the distributions of the following 10 features computed for the cellnuclei;

- radius (mean of distances from center to points on the perimeter)
- texture (standard deviation of gray-scale values)
- perimeter
- area
- smoothness (local variation in radius lengths)
- compactness ($perimeter^2/area - 1$)
- concavity (severity of concave portions of the contour)
- concave points (number of concave portions of the contour)
- symmetry
- fractal dimension ("coastline approximation" - 1)

Consider a logistic LASSO regression to predict cancer cases based on the image features.

1. Propose and implement a 5-fold cross-validation aglorithm to select the turning parameter in the logistic LASSO regression. We call the logistic-LASSO with CV-selected λ as the "optimal" logistic LASSO; The R function for logistic-LASSO
2. Using the selected predictors from the "optimal" logistic LASSO to predict the probability of malignant for each of the images (Note that estimates from logistic-Lasso are biased. You need to re-fit the logistic regression with the selected predictors to estimate the probability.) How well the predictors classify the images?
3. Using the bootstraping smoothing idea to re-evaluate the probabilities of malignant. How well the new predictors classify the images?
4. Writ a summary of your findings.