Bootstrap and Jackknife

Dennis Do

11/17/2021

1)

```
load("water_qual.RData")
head(water_qual)
```

```
##   median_cl2 population median_income prop_children LO_health
## 1       2.02       2863        69.583    0.13726860  7.959975
## 2       0.96       3717        45.167    0.06887275  2.255012
## 3       2.70       4966        37.523    0.09967781  8.510571
## 4       2.86       7053        22.602    0.16347653  6.294698
## 5       2.89       9038        10.829    0.03098031  4.744040
## 6       2.70       5115        23.644    0.18670577  4.451189
```

```
library(boot)
```

```
cor.test(water_qual$median_cl2,water_qual$median_income)
```

```
##
##  Pearson's product-moment correlation
##
## data:  water_qual$median_cl2 and water_qual$median_income
## t = -2.9314, df = 146, p-value = 0.003919
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.38255414 -0.07736107
## sample estimates:
##        cor
## -0.2357622
```

From the correlation it is negative and there is not a strong correlation between income and median chlorine levels

2)

```
data(water_qual, package ="bootstrap")
```

```
## Warning in data(water_qual, package = "bootstrap"): data set 'water_qual' not
## found
```

```r
n <- nrow(water_qual)
z <- water_qual[,"median_cl2"]
y <- water_qual[,"median_income"]
theta.hat <- cor(y,z)
theta.jack <- numeric(n)
for (i in 1:n)
theta.jack[i] <- cor(y[-i], z[-i])
bias <- (n-1) * (mean(theta.jack) - theta.hat)
theta.calc <- theta.hat - bias
print(theta.calc)
```

```
## [1] -0.2363447
```

```r
print(bias)
```

```
## [1] 0.0005825009
```

3) With a small bias it can be concluded that we should not use our answer since there is insignificant amount of biasness.

4)

```r
duration = water_qual$median_cl2
quantile(duration, c(.90))
```

```
##    90%
## 2.5815
```

5)

```r
n <- 2e2
prob <- .9
B <- 1e3
x_star <- matrix(sample(water_qual$median_cl2, n*8, replace = T),n,B)
theta_star <- apply(x_star, 2, quantile, probs = prob)
mean(theta_star) + sd(theta_star) * 1.96 * c(-1,1)
```

```
## [1] 2.520521 2.632729
```

6)

```r
n <- 2e2
prob <- .1
B <- 1e3
x_star <- matrix(sample(water_qual$median_cl2, n*8, replace = T),n,B)
theta_star <- apply(x_star, 2, quantile, probs = prob)
mean(theta_star) + sd(theta_star) * 1.96 * c(-1,1)
```

```
## [1] 1.143906 1.563969
```

7)

```r
fits <- lm(median_cl2 ~ population + median_income + prop_children + LO_health, data = water_qual)
fits
```

```
##
## Call:
## lm(formula = median_cl2 ~ population + median_income + prop_children +
##     LO_health, data = water_qual)
##
## Coefficients:
##   (Intercept)     population  median_income  prop_children      LO_health
##     2.325e+00      2.336e-06     -5.824e-03     -8.378e-01      8.516e-03
```

8)

```r
confint(fits, level = .95)
```

```
##                       2.5 %        97.5 %
## (Intercept)    1.978148e+00  2.671096e+00
## population    -2.219046e-05  2.686167e-05
## median_income -1.139372e-02 -2.540114e-04
## prop_children -2.309725e+00  6.340881e-01
## LO_health     -3.909146e-02  5.612274e-02
```

9)

```r
get_regression_coefs <- function(data, ind){
  fit <- lm(median_cl2 ~ population + median_income + prop_children + LO_health, data = data[ind, ])
  coef(fit)
}
get_regression_coefs(water_qual, 1:10)
```

```
##   (Intercept)    population median_income prop_children     LO_health
##  3.5346120939 -0.0002160563 -0.0473334873 -2.3760093672  0.3081064103
```

10)

```r
boot_obj <- boot(data =water_qual, get_regression_coefs, R=1000)
boot.ci(boot.out = boot_obj, conf = .95, type = 'perc', index= 3)
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_obj, conf = 0.95, type = "perc", index = 3)
##
## Intervals :
## Level     Percentile
## 95%   (-0.0110, -0.0006 )
## Calculations and Intervals on Original Scale
```

11) The 95% confidence interval in 10 is narrower than the interval in 8. THis isn't surprising since the function we use finds the best fit with 1000 replicates. Using replicates allows the function to be more accurate to find the confidence interval.