Data Analysis and Simulation for Ratio and Regression Estimation

Longhai Li

Contents

1 Functions and packages for Analyzing Data **1**

2 Analysis of cherry.csv dataset using Ratio Estimation **2**
2.1 Importing cherry.csv data 2
2.2 SRS estimate 4
2.3 Step-by-step calculation with Ratio Estimation 5
2.4 Estimating the total volume of wood 7
2.5 Ratio estimation with a function 7

3 Analysis of cherry.csv dataset using Regression Estimation **8**
3.1 Step-by-step calculation 8
3.2 Using the function 10

1 Functions and packages for Analyzing Data

```
## ydata --- observations of the variable of interest
## xdata --- observations of the auxilliary variable
## N --- population size
## xbarU --- population mean of auxilliary variable

## the output is the estimate mean or total (est.total=TRUE)
srs_reg_est <- function (ydata, xdata, xbarU, N = Inf, est.total = FALSE)
{
  n <- length (ydata)
  lmfit <- lm (ydata ~ xdata)
  Bhat <- lmfit$coefficients
  efit <- lmfit$residuals
  SSe <- sum (efit^2) / (n - 2)
  yhat_reg <- Bhat[1] + Bhat[2] * xbarU
  se_yhat_reg <- sqrt ((1-n/N) * SSe / n)
  mem <- qt (0.975, df = n - 2) * se_yhat_reg
  output <- c(yhat_reg, se_yhat_reg, yhat_reg - mem, yhat_reg + mem)

  if (est.total) {
      if(!is.finite(N)) stop("N must be finite for estimating population total" )
      output <- output * N
  }

  names (output) <- c("Est.", "S.E.", "ci.low", "ci.upp" )
  output
}
```

```r
## ydata --- observations of the variable of interest
## xdata --- observations of the auxilliary variable
## N --- population size

## the output is the ratio of ybarU/xbarU
srs_ratio_est <- function (ydata, xdata, N = Inf)
{
  n <- length (xdata)
  xbar <- mean (xdata)
  ybar <- mean (ydata)
  B_hat <- ybar / xbar
  d <- ydata - B_hat * xdata
  var_d <- sum (d^2) / (n - 1)
  sd_B_hat <- sqrt ((1 - n/N) * var_d / n) / xbar
  mem <- qt (0.975, df = n - 1) * sd_B_hat
  output <- c (B_hat, sd_B_hat, B_hat - mem, B_hat + mem )


  names (output) <- c("Est.", "S.E.", "ci.low", "ci.upp" )
  output
}


## sdata --- a vector of original survey data
## N --- population size
## to find total, multiply N to the estimate returned by this function

srs_mean_est <- function (sdata, N = Inf)
{
    n <- length (sdata)
    ybar <- mean (sdata)
    se.ybar <- sqrt((1 - n / N)) * sd (sdata) / sqrt(n)
    mem <- qt (0.975, df = n - 1) * se.ybar
    c (Est. = ybar, S.E. = se.ybar, ci.low = ybar - mem, ci.upp = ybar + mem)
}
```

2 Analysis of cherry.csv dataset using Ratio Estimation

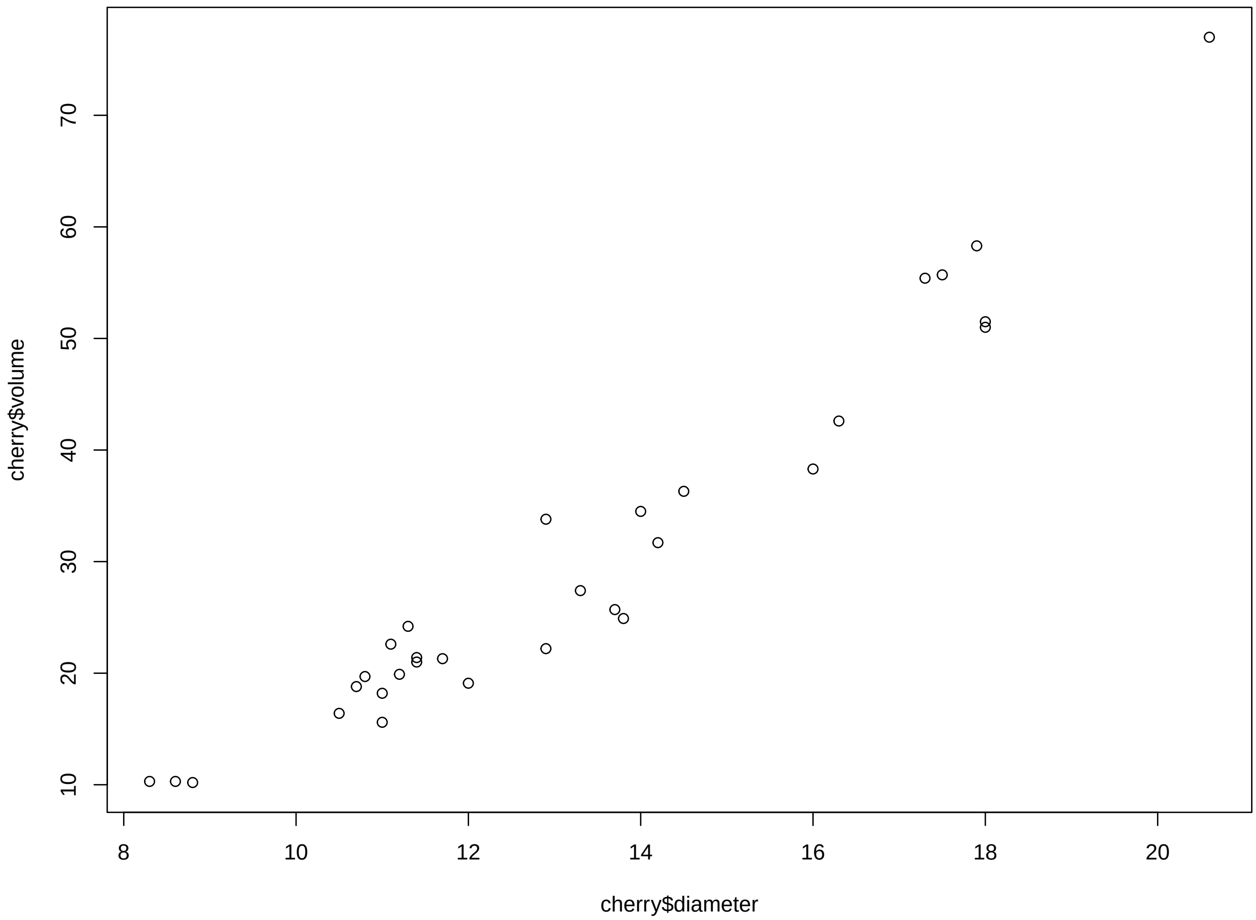
2.1 Importing cherry.csv data

```r
cherry <- read.csv ("data/cherry.csv", header = T)
cherry
```

```
##    diameter height volume
## 1       8.3     70   10.3
## 2       8.6     65   10.3
## 3       8.8     63   10.2
## 4      10.5     72   16.4
## 5      10.7     81   18.8
## 6      10.8     83   19.7
## 7      11.0     66   15.6
```

```
## 8       11.0     75   18.2
## 9       11.1     80   22.6
## 10      11.2     75   19.9
## 11      11.3     79   24.2
## 12      11.4     76   21.0
## 13      11.4     76   21.4
## 14      11.7     69   21.3
## 15      12.0     75   19.1
## 16      12.9     74   22.2
## 17      12.9     85   33.8
## 18      13.3     86   27.4
## 19      13.7     71   25.7
## 20      13.8     64   24.9
## 21      14.0     78   34.5
## 22      14.2     80   31.7
## 23      14.5     74   36.3
## 24      16.0     72   38.3
## 25      16.3     77   42.6
## 26      17.3     81   55.4
## 27      17.5     82   55.7
## 28      17.9     80   58.3
## 29      18.0     80   51.5
## 30      18.0     80   51.0
## 31      20.6     87   77.0
```

```
plot (cherry$volume ~ cherry$diameter)
```




```
summary (lm(cherry$volume ~ 0+cherry$diameter))
```

```
##
## Call:
## lm(formula = cherry$volume ~ 0 + cherry$diameter)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.104  -8.470  -6.199   1.883  27.129
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## cherry$diameter   2.4209     0.1253   19.32   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.493 on 30 degrees of freedom
## Multiple R-squared:  0.9256, Adjusted R-squared:  0.9231
## F-statistic: 373.1 on 1 and 30 DF,  p-value: < 2.2e-16
```

2.2 SRS estimate

```
N <- 2967
## estimating the mean of volume
srs_mean_est(cherry$volume, N = N)
```

```
##      Est.      S.E.    ci.low    ci.upp
## 30.170968  2.936861 24.173098 36.168837
```

```
## estimating the mean of volume
srs_mean_est(cherry$volume, N = N) * N
```

```
##       Est.       S.E.     ci.low     ci.upp
##  89517.261   8713.665  71721.583 107312.940
```

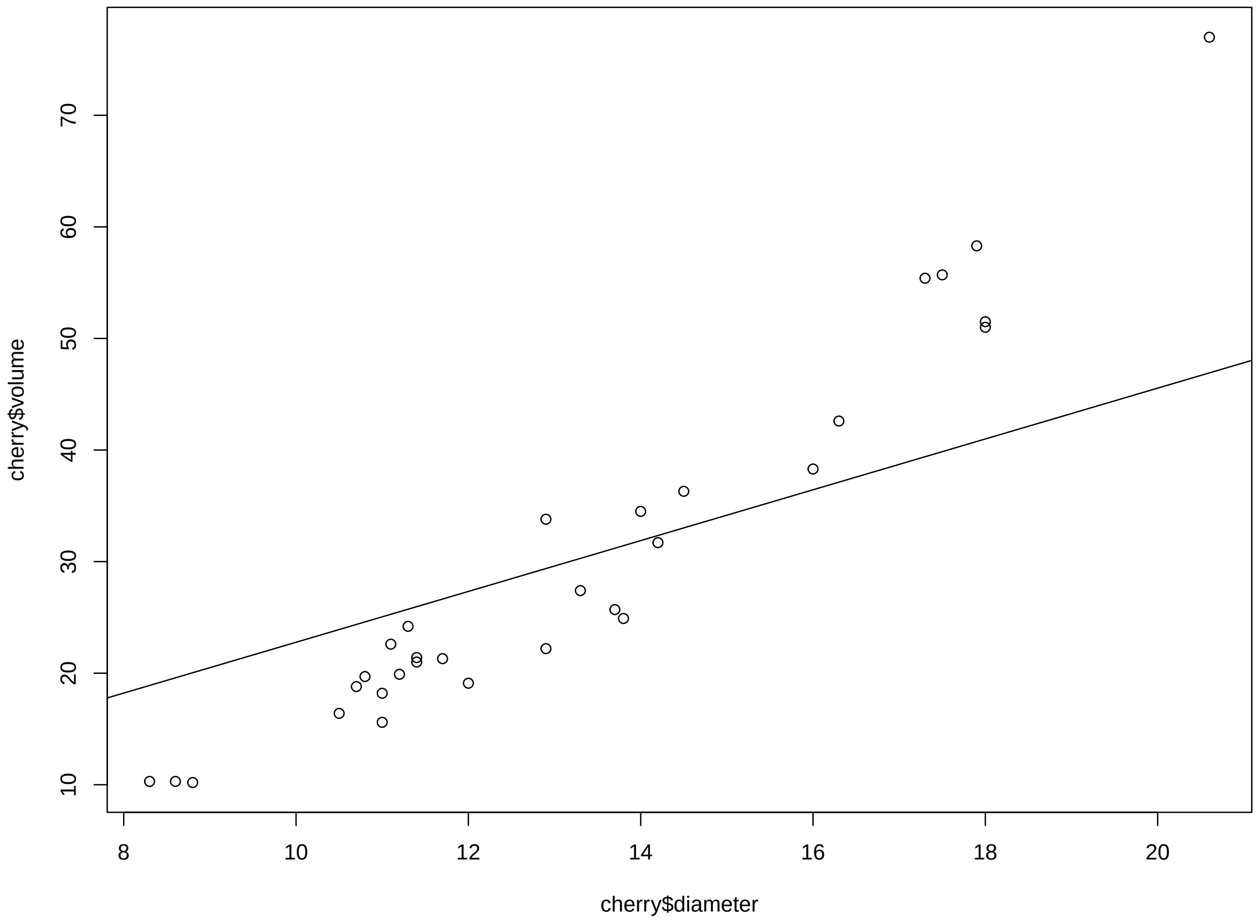
2.3 Step-by-step calculation with Ratio Estimation

2.3.1 Estimating B and calculating residuals

```
## input
ydata <- cherry$volume
xdata <- cherry$diameter
N <- 2967

## calculation
n <- length (xdata)
xbar <- mean (xdata)
ybar <- mean (ydata)
B_hat <- ybar / xbar ## ratio estimate


plot (cherry$volume ~ cherry$diameter)
abline (a = 0, b = B_hat)
```




```r
d <- ydata - B_hat * xdata ## errors
data.frame (cherry, d = d)
```

```
##    diameter height volume          d
## 1       8.3     70   10.3 -8.6018505
## 2       8.6     65   10.3 -9.2850499
## 3       8.8     63   10.2 -9.8405162
## 4      10.5     72   16.4 -7.5119795
## 5      10.7     81   18.8 -5.5674458
## 6      10.8     83   19.7 -4.8951790
## 7      11.0     66   15.6 -9.4506452
## 8      11.0     75   18.2 -6.8506452
## 9      11.1     80   22.6 -2.6783784
## 10     11.2     75   19.9 -5.6061115
## 11     11.3     79   24.2 -1.5338447
## 12     11.4     76   21.0 -4.9615778
## 13     11.4     76   21.4 -4.5615778
## 14     11.7     69   21.3 -5.3447772
## 15     12.0     75   19.1 -8.2279766
## 16     12.9     74   22.2 -7.1775749
## 17     12.9     85   33.8  4.4224251
## 18     13.3     86   27.4 -2.8885074
## 19     13.7     71   25.7 -5.4994400
## 20     13.8     64   24.9 -6.5271731
## 21     14.0     78   34.5  2.6173606
```

```
## 22      14.2     80   31.7 -0.6381057
## 23      14.5     74   36.3  3.2786949
## 24      16.0     72   38.3  1.8626978
## 25      16.3     77   42.6  5.4794984
## 26      17.3     81   55.4 16.0021670
## 27      17.5     82   55.7 15.8467008
## 28      17.9     80   58.3 17.5357682
## 29      18.0     80   51.5 10.5080351
## 30      18.0     80   51.0 10.0080351
## 31      20.6     87   77.0 30.0869735
```

2.3.2 Estimating SE of B

```
## estimating S^2_e
var_d <- sum (d^2) / (n - 1) ## variance of errors
sd_B_hat <- sqrt ((1 - n/N) * var_d / n) / xbar ## SE for B
mem <- qt (0.975, df = n - 1) * sd_B_hat ## margin error for B

## output
output_B <- c (B_hat, sd_B_hat, B_hat - mem, B_hat + mem )
names (output_B) <- c("Est.", "S.E.", "ci.low", "ci.upp" )
output_B
```

```
##     Est.     S.E.   ci.low   ci.upp
## 2.277331 0.130786 2.010231 2.544432
```

2.3.3 Estimating the mean volume of wood

```
mean_diameters <- 41835/N
output_B * mean_diameters
```

```
##      Est.      S.E.    ci.low    ci.upp
## 32.110603  1.844097 28.344455 35.876750
```

2.4 Estimating the total volume of wood

```
t_diameters <- 41835
output_B * t_diameters
```

```
##       Est.       S.E.     ci.low     ci.upp
##  95272.159   5471.434  84097.999 106446.318
```

2.5 Ratio estimation with a function

2.5.1 estimate ratio of volume to diameter

```
B_v2d <- srs_ratio_est (ydata = cherry$volume, xdata = cherry$diameter,  N = 2967)
B_v2d
```

```
##     Est.     S.E.   ci.low   ci.upp
## 2.277331 0.130786 2.010231 2.544432
```

2.5.2 Estimating total volume

```
xbarU <- 41835/N
srs_ratio_est (ydata = cherry$volume, xdata = cherry$diameter, N = 2967) * xbarU
```

```
##      Est.      S.E.    ci.low    ci.upp
## 32.110603  1.844097 28.344455 35.876750
```

2.5.3 Estimating the total of volume

```
total_diameters <- 41835
srs_ratio_est (ydata = cherry$volume, xdata = cherry$diameter, N = 2967) * total_diameters
```

```
##       Est.       S.E.     ci.low     ci.upp
##  95272.159   5471.434  84097.999 106446.318
```

3 Analysis of cherry.csv dataset using Regression Estimation

3.1 Step-by-step calculation

3.1.1 Importing data

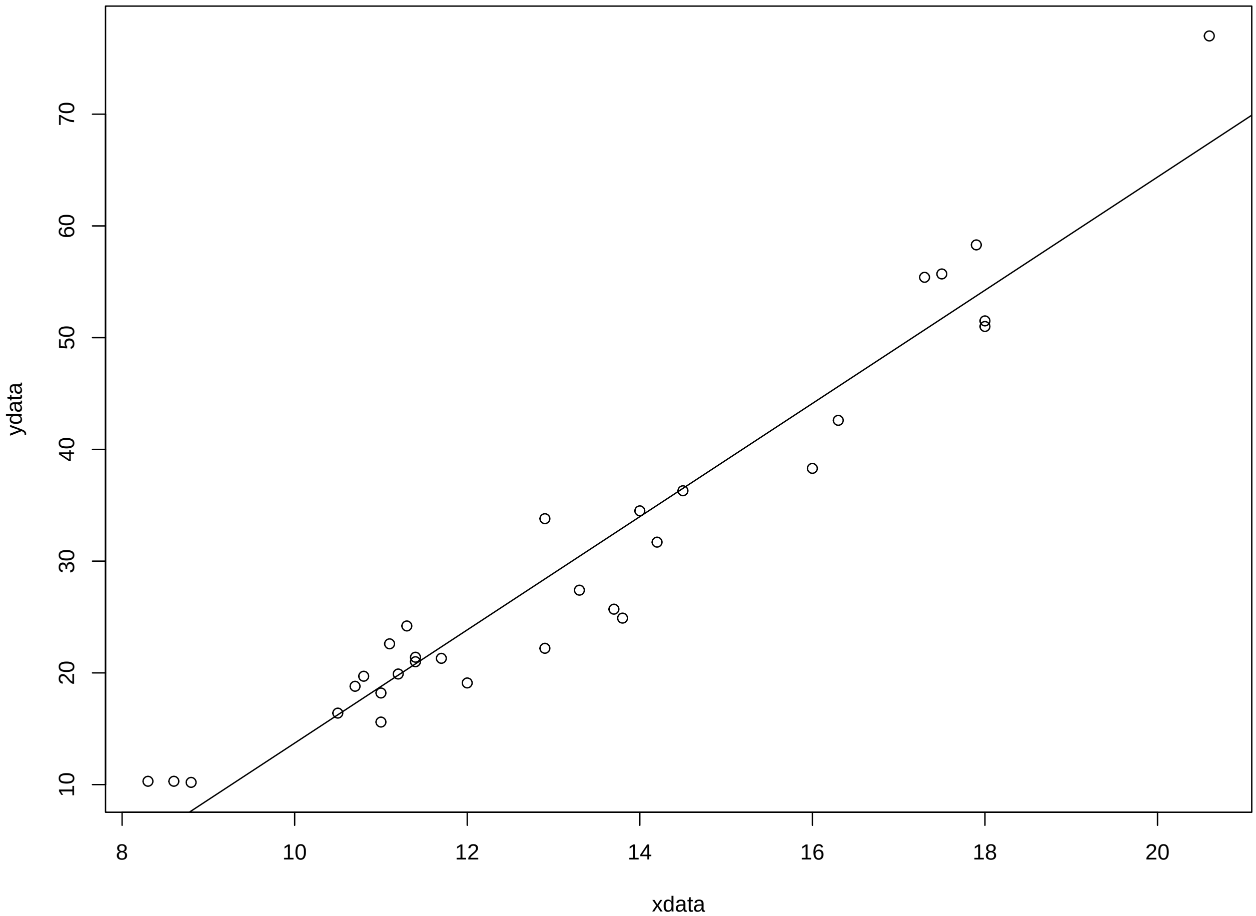
```
ydata <- cherry$volume
xdata <- cherry$diameter
t_diameters <- 41835
xbarU <- t_diameters/2967
N <- 2967
```

3.1.2 Fitting a linear regression model

```
n <- length (ydata)
lmfit <- lm (ydata ~ xdata)
summary (lmfit)
```

```
##
## Call:
## lm(formula = ydata ~ xdata)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.065 -3.107  0.152  3.495  9.587
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -36.9435     3.3651  -10.98 7.62e-12 ***
## xdata         5.0659     0.2474   20.48  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.252 on 29 degrees of freedom
## Multiple R-squared:  0.9353, Adjusted R-squared:  0.9331
## F-statistic: 419.4 on 1 and 29 DF,  p-value: < 2.2e-16
```

```
plot (xdata, ydata)
abline (lmfit)
```




```
Bhat <- lmfit$coefficients
efit <- ydata - (Bhat[1] + Bhat[2] * xdata)
data.frame (cherry, residual=efit) ## for visualization
```

```
##    diameter height volume   residual
## 1       8.3     70   10.3  5.1968508
## 2       8.6     65   10.3  3.6770939
## 3       8.8     63   10.2  2.5639226
## 4      10.5     72   16.4  0.1519667
## 5      10.7     81   18.8  1.5387954
## 6      10.8     83   19.7  1.9322098
## 7      11.0     66   15.6 -3.1809615
## 8      11.0     75   18.2 -0.5809615
## 9      11.1     80   22.6  3.3124528
## 10     11.2     75   19.9  0.1058672
## 11     11.3     79   24.2  3.8992815
## 12     11.4     76   21.0  0.1926959
## 13     11.4     76   21.4  0.5926959
## 14     11.7     69   21.3 -1.0270610
## 15     12.0     75   19.1 -4.7468179
## 16     12.9     74   22.2 -6.2060887
## 17     12.9     85   33.8  5.3939113
```

```
## 18     13.3     86   27.4 -3.0324313
## 19     13.7     71   25.7 -6.7587739
## 20     13.8     64   24.9 -8.0653595
## 21     14.0     78   34.5  0.5214692
## 22     14.2     80   31.7 -3.2917021
## 23     14.5     74   36.3 -0.2114590
## 24     16.0     72   38.3 -5.8102436
## 25     16.3     77   42.6 -3.0300006
## 26     17.3     81   55.4  4.7041430
## 27     17.5     82   55.7  3.9909717
## 28     17.9     80   58.3  4.5646292
## 29     18.0     80   51.5 -2.7419565
## 30     18.0     80   51.0 -3.2419565
## 31     20.6     87   77.0  9.5868168
```

```
SSe <- sum (efit^2) / (n - 2)
```

3.1.3 Estiamte the mean

```
yhat_reg <- Bhat[1] + Bhat[2] * xbarU
se_yhat_reg <- sqrt ((1-n/N) * SSe / n)
mem <- qt (0.975, df = n - 2) * se_yhat_reg
output <- c(yhat_reg, se_yhat_reg, yhat_reg - mem, yhat_reg + mem)
names (output) <- c("Est.", "S.E.", "ci.low", "ci.upp" )
output
```

```
##       Est.       S.E.     ci.low     ci.upp
## 34.4856287  0.7596795 32.9319097 36.0393476
```

3.1.4 Estiamte the total

```
output * N
```

```
##       Est.       S.E.     ci.low     ci.upp
## 102318.860   2253.969  97708.976 106928.744
```

3.2 Using the function

```
## estimating the mean
srs_reg_est(ydata = cherry$volume, xdata = cherry$diameter,
                 xbarU=t_diameters/2967, N = 2967)
```

```
##       Est.       S.E.     ci.low     ci.upp
## 34.4856287  0.7596795 32.9319097 36.0393476
```

```
## estimating the total
t_diameters <- 41835
srs_reg_est(ydata = cherry$volume, xdata = cherry$diameter,
                 xbarU=t_diameters/2967, N = 2967, est.total = TRUE)
```

```
##       Est.       S.E.     ci.low     ci.upp
## 102318.860   2253.969  97708.976 106928.744
```