Assignment2

CMSC462

2024-09-19

```r
#install.packages("tidyverse")
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
#Read data, and display sample of dataset
Data = read_csv(file = "C:/Users/criss/Desktop/CMSC462/Assignments/[9-22]Assignment 2/myData.csv")
```

```
## New names:
## Rows: 100000 Columns: 3
## -- Column specification
## -------------------------------------------------------- Delimiter: "," dbl
## (3): ...1, x, y
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```r
Data
```

```
## # A tibble: 100,000 x 3
##     ...1     x     y
##    <dbl> <dbl> <dbl>
##  1     1 87.4      5
##  2     2 41.9     10
##  3     3 45.5     13
##  4     4 30.7      9
##  5     5 20.8     10
##  6     6 17.1     11
##  7     7  5.95     9
##  8     8 19.5      8
##  9     9 66.1      8
## 10    10 92.0     14
## # i 99,990 more rows
```

```r
#create an empty vector to store the point estimates of the samples to be taken for x&y,
#first 5 elements will be of sample 25, later 5 elements will be of sample 100.
meanx <- c()
meany <- c()
sdx <- c()
sdy <- c()

for(i in 1:10){ #iterate 10 times
  if(i<6){
    meanx[i] = mean(sample(Data$x,25)) #populate first 5 elements with PE of random sample of 25
    meany[i] = mean(sample(Data$y,25))
    sdx[i] = sd(sample(Data$x,25))
    sdy[i] = sd(sample(Data$y,25))

  }else{
    meanx[i] = mean(sample(Data$x,100)) #populate last 5 element with PE of random sample of 100
    meany[i] = mean(sample(Data$y,100))
    sdx[i] = sd(sample(Data$x,100))
    sdy[i] = sd(sample(Data$y,100))
  }
}
meanx
```

```
##  [1] 55.22881 46.63432 51.89651 47.99033 38.98472 54.69677 54.34758 48.08768
##  [9] 51.00309 51.26608
```

```r
meany
```

```
##  [1] 10.20 10.80 10.52  9.68 10.36  9.74 10.26  9.37 10.44 10.38
```

```r
#Calculate 95% CI for x and y, by applying 95% CI = Xbar +- 1.96*(s/sqrt(n))

#Upper and lower limit for 25 samples:
UL_X_25 <- c()
UL_Y_25 <- c()
LL_X_25 <- c()
LL_Y_25 <- c()

#Apply the confidence interval formula for 95% confidence: Xbar +- 1.96(sd/sqrt(n))
for(i in 1:5){
  UL_X_25[i] = meanx[i] + (1.96)*(sdx[i] / sqrt(25))
  UL_Y_25[i] = meany[i] + (1.96)*(sdy[i] / sqrt(25))
  LL_X_25[i] = meanx[i] - (1.96)*(sdx[i] / sqrt(25))
  LL_Y_25[i] = meany[i] - (1.96)*(sdy[i] / sqrt(25))
}

#Upper and lower limit for 100 samples:
UL_X_100 <- c()
UL_Y_100 <- c()
LL_X_100 <- c()
LL_Y_100 <- c()
```

```r
for(i in 6:10){
  UL_X_100[i-5] = meanx[i] + (1.96)*(sdx[i] / sqrt(100))
  UL_Y_100[i-5] = meany[i] + (1.96)*(sdy[i] / sqrt(100))
  LL_X_100[i-5] = meanx[i] - (1.96)*(sdx[i] / sqrt(100))
  LL_Y_100[i-5] = meany[i] - (1.96)*(sdy[i] / sqrt(100))
}

UL_X_25
```

```
## [1] 66.28655 58.73508 64.57241 59.84808 51.32745
```

```r
UL_Y_25
```

```
## [1] 11.49380 12.04723 11.67335 10.88886 11.22418
```

```r
LL_X_25
```

```
## [1] 44.17106 34.53357 39.22061 36.13257 26.64200
```

```r
LL_Y_25
```

```
## [1] 8.906202 9.552766 9.366649 8.471139 9.495820
```

```r
UL_X_100
```

```
## [1] 60.25964 60.25128 53.61980 56.63454 56.77940
```

```r
UL_Y_100
```

```
## [1] 10.32726 10.82547 10.00603 10.99550 11.07148
```

```r
LL_X_100
```

```
## [1] 49.13390 48.44387 42.55557 45.37164 45.75276
```

```r
LL_Y_100
```

```
## [1] 9.152740 9.694526 8.733969 9.884495 9.688524
```

```r
# Create a 2x4 matrix to display the intervals
table <- matrix(nrow = 4, ncol = 2)

# Name the columns and rows appropriately
colnames(table) <- c('Lower limit', 'Upper limit')
rownames(table) <- c('X 25 sample', 'Y 25 sample', 'X 100 sample', 'Y 100 sample')

# Calculate the means of each upper limit and lower limit for the CI's
limits <- c(mean(LL_X_25), mean(UL_X_25),
```

```r
            mean(LL_Y_25), mean(UL_Y_25),
            mean(LL_X_100), mean(UL_X_100),
            mean(LL_Y_100), mean(UL_Y_100))

# Fill the table with the calculated means
table[,] <- matrix(limits, nrow = 4, byrow = TRUE)

# Convert to table
limits_table <- as.table(table)
limits_table
```

```
##              Lower limit Upper limit
## X 25 sample    36.139960   60.153916
## Y 25 sample     9.158515   11.465485
## X 100 sample   46.251548   57.508932
## Y 100 sample    9.430851   10.645149
```

```r
#Calculate actual population means
mean(Data$x)
```

```
## [1] 50.01188
```

```r
mean(Data$y)
```

```
## [1] 10.00173
```

```r
cat("As we can see, the actual mean for x is ~50.01, and the actual mean for y
      is ~10, for the population. Both of these values fall into their sampled
      95% confidence intervals, for both 25 samples and 100 samples. 50.01 falls
      into the 25 x sample: (40.82,64.43) CI and into the 100 x sample:
      (42.79, 54.29) CI. And 10 falls into the 25 y sample: (9.06,11.42) CI and
      into the 100 y sample: (9.22, 10.41) CI. We can observe that for both X
      and Y, the 100 sample confidence intervals are more accurate in regards to
      the population mean, because both upper and lower limits are closer to the
      true mean compared to the 25 sample intervals.")
```

```
## As we can see, the actual mean for x is ~50.01, and the actual mean for y
##       is ~10, for the population. Both of these values fall into their sampled
##       95% confidence intervals, for both 25 samples and 100 samples. 50.01 falls
##       into the 25 x sample: (40.82,64.43) CI and into the 100 x sample:
##       (42.79, 54.29) CI. And 10 falls into the 25 y sample: (9.06,11.42) CI and
##       into the 100 y sample: (9.22, 10.41) CI. We can observe that for both X
##       and Y, the 100 sample confidence intervals are more accurate in regards to
##       the population mean, because both upper and lower limits are closer to the
##       true mean compared to the 25 sample intervals.
```