ConfidenceInterval_Simulation

```r
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 1.9.4     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 becon
```

```r
#read the data
CHIS_data <- read_csv("/Users/saurabhgupta/projects/github/StatisticsFundamentals/Assignment2
```

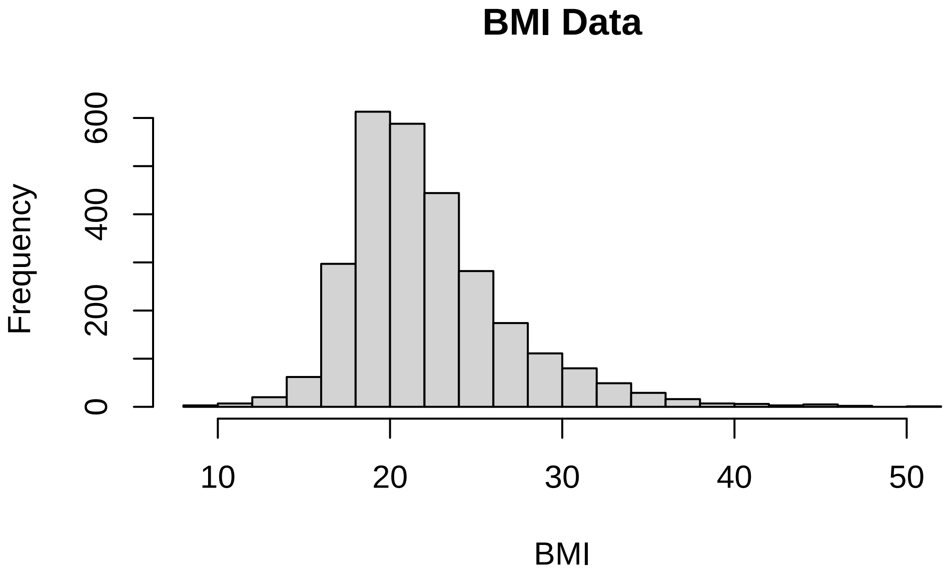
```
New names:
Rows: 2799 Columns: 5
-- Column specification
-------------------------------------------------------- Delimiter: "," dbl
(5): ...1, Height, Weight, BMI, Asian
i Use `spec()` to retrieve the full column specification for this data. i
Specify the column types or set `show_col_types = FALSE` to quiet this message.
* `` -> `...1`
```

```r
CHIS_data
```

```
# A tibble: 2,799 x 4
   Height Weight   BMI Asian
    <dbl>  <dbl> <dbl> <dbl>
```

```
 1      67     125  19.8      0
 2      70     145  20.5      0
 3      64     200  34.6      0
 4      63     112  20.1      0
 5      57      85  18.6      0
 6      70     150  21.7      0
 7      66     180  29.3      0
 8      61     110  20.5      0
 9      69     155  23.2      0
10      65     165  27.8      0
# i 2,789 more rows
```

```
hist(CHIS_data$BMI, main = "BMI Data", xlab = "BMI", breaks = 20)
```



```
mu <- mean(CHIS_data$BMI)
mu
```

```
[1] 22.28086
```

```
set.seed(3)
n <- 25
sample_data <- sample(CHIS_data$BMI, size = n, replace = FALSE)
sample_data
```

```
 [1] 23.68 19.30 19.86 16.98 20.73 27.94 20.65 13.14 32.85 22.99 34.02 26.03
[13] 18.51 25.42 21.26 23.18 19.49 17.50 17.95 27.65 24.54 22.66 29.57 27.58
[25] 16.77
```

```
mean(sample_data)
```

```
[1] 22.81
```

```
sd(sample_data)/sqrt(n)
```

```
[1] 1.03562
```

```
c(mean(sample_data)-qnorm(0.975)*sd(sample_data)/sqrt(n),
  mean(sample_data)+qnorm(0.975)*sd(sample_data)/sqrt(n))
```

```
[1] 20.78022 24.83978
```

```
N <- 100
observed_CIs <- replicate(N, {
  sample_data <- sample(CHIS_data$BMI, size = n, replace = FALSE)
  c(mean(sample_data)-qnorm(0.975)*sd(sample_data)/sqrt(n),
    mean(sample_data)+qnorm(0.975)*sd(sample_data)/sqrt(n))
})
observed_CIs[, 1:5]
```

```
        [,1]     [,2]     [,3]     [,4]     [,5]
[1,] 20.1040 20.53774 19.09465 20.01801 20.08299
[2,] 23.0712 24.29826 22.59255 23.15399 23.30581
```

```
#check how many contained the trut value
mean((observed_CIs[1, ] < mu) & (observed_CIs[2, ] > mu))
```

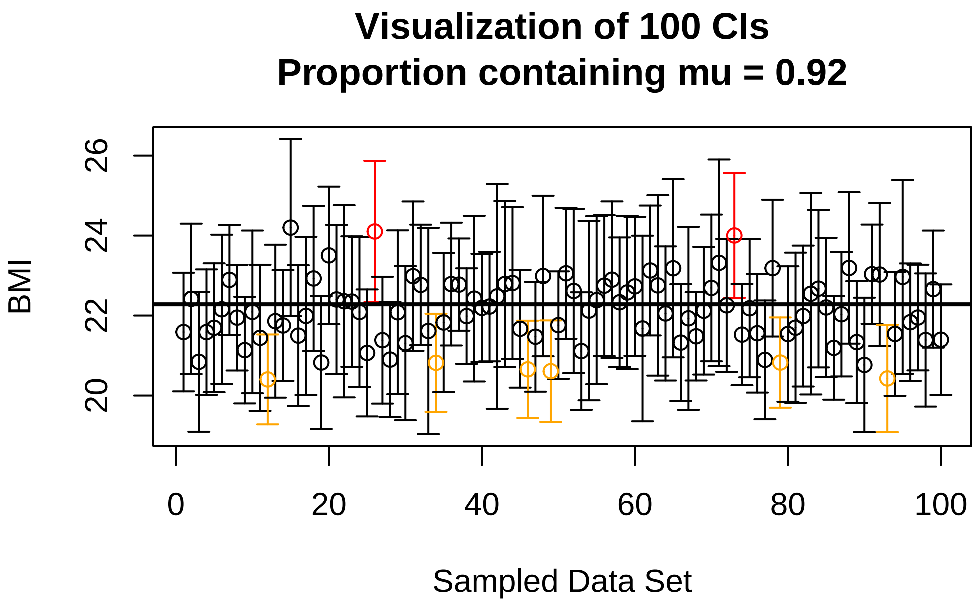
```
[1] 0.92
```

```r
#quick function to color our intervals based on how they hit or miss
mycolor <- function(endpoints, par) {
  if (par < endpoints[1])
    "Red"  # if the mean is below the left endpoint of the confidence interval
  else if (par > endpoints[2])
    "Orange"  # if the mean is above the right endpoint of the confidence interval
  else "Black"  # if the mean lies between the endpoints
}

#Load the plotrix package, which contains the plotCI function.
require(plotrix)
```

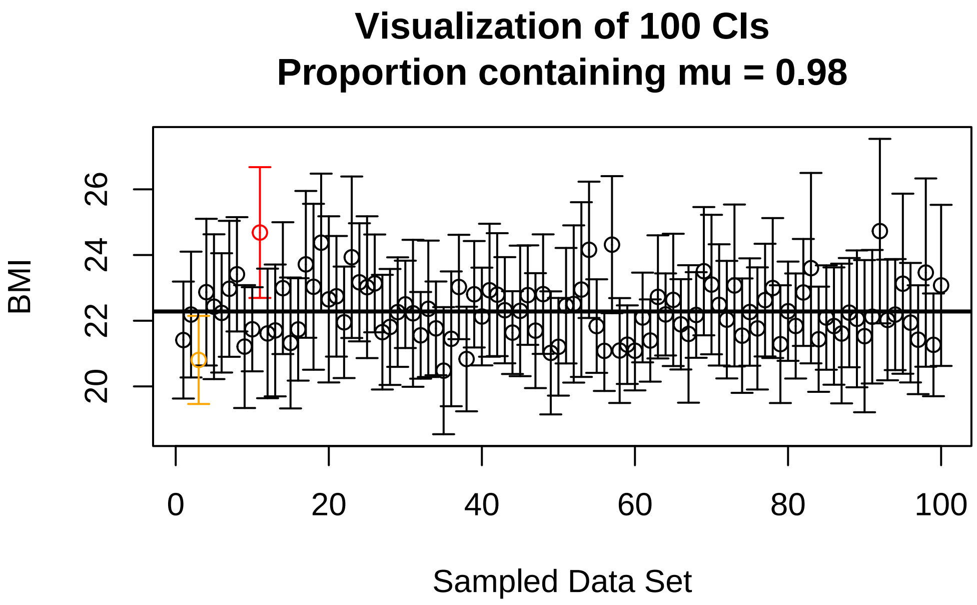
```
Loading required package: plotrix
```

```r
plotCI(x = 1:N,
       y = colMeans(observed_CIs),
       li = observed_CIs[1, ],
       ui = observed_CIs[2, ],
       col = apply(FUN = mycolor, X = observed_CIs, MARGIN = 2, par = mu),
       ylab = "BMI",
       xlab = "Sampled Data Set",
       main = paste0("Visualization of 100 CIs\nProportion containing mu = ",
              mean((observed_CIs[1, ] < mu) & (observed_CIs[2, ] > mu)))
)
#draw a line for true mean
abline(h = mu, lwd = 2)
```



```
N <- 100
observed_CIs <- replicate(N, {
  sample_data <- sample(CHIS_data$BMI, size = n, replace = FALSE)
  c(mean(sample_data)-qnorm(0.975)*sd(sample_data)/sqrt(n),
    mean(sample_data)+qnorm(0.975)*sd(sample_data)/sqrt(n))
})

plotCI(x = 1:N,
       y = colMeans(observed_CIs),
       li = observed_CIs[1, ],
       ui = observed_CIs[2, ],
       col = apply(FUN = mycolor, X = observed_CIs, MARGIN = 2, par = mu),
       ylab = "BMI",
       xlab = "Sampled Data Set",
       main = paste0("Visualization of 100 CIs\nProportion containing mu = ",
                     mean((observed_CIs[1, ] < mu) & (observed_CIs[2, ] > mu)))
)
#draw a line for true mean
abline(h = mu, lwd = 2)
```

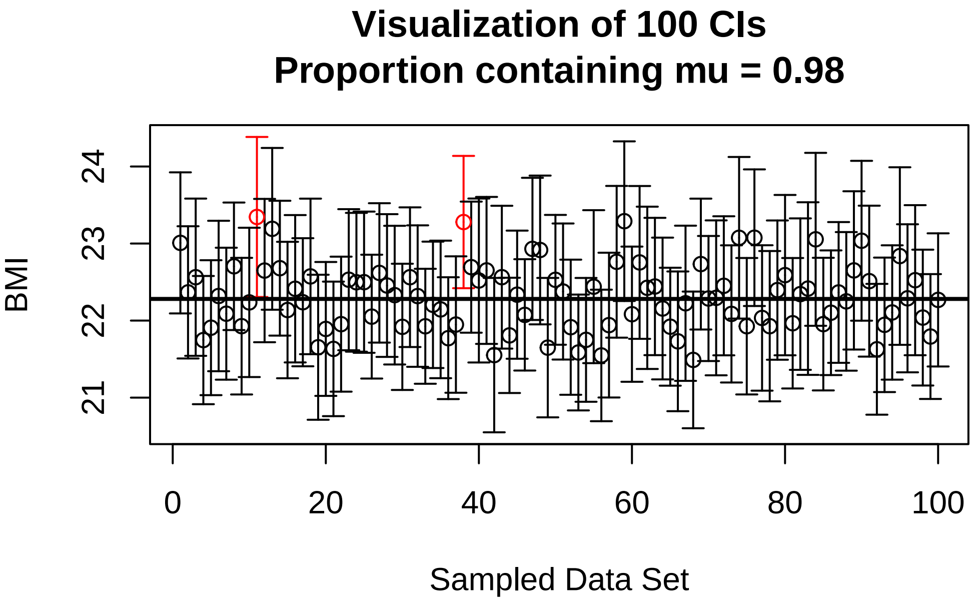


```
N <- 10000
observed_CIs <- replicate(N, {
  sample_data <- sample(CHIS_data$BMI, size = n, replace = FALSE)
  c(mean(sample_data)-qnorm(0.975)*sd(sample_data)/sqrt(n),
    mean(sample_data)+qnorm(0.975)*sd(sample_data)/sqrt(n))
})
#check how many contained the true value
mean((observed_CIs[1, ] < mu) & (observed_CIs[2, ] > mu))
```

```
[1] 0.9259
```

```
N <- 100
n <- 100
observed_CIs <- replicate(N, {
  sample_data <- sample(CHIS_data$BMI, size = n, replace = FALSE)
  c(mean(sample_data)-qnorm(0.975)*sd(sample_data)/sqrt(n),
    mean(sample_data)+qnorm(0.975)*sd(sample_data)/sqrt(n))
})

plotCI(x = 1:N,
       y = colMeans(observed_CIs),
       li = observed_CIs[1, ],
       ui = observed_CIs[2, ],
       col = apply(FUN = mycolor, X = observed_CIs, MARGIN = 2, par = mu),
       ylab = "BMI",
       xlab = "Sampled Data Set",
       main = paste0("Visualization of 100 CIs\nProportion containing mu = ",
                    mean((observed_CIs[1, ] < mu) & (observed_CIs[2, ] > mu)))
)
#draw a line for true mean
abline(h = mu, lwd = 2)
```



```
N <- 10000
n <- 100
observed_CIs <- replicate(N, {
  sample_data <- sample(CHIS_data$BMI, size = n, replace = FALSE)
  c(mean(sample_data)-qnorm(0.975)*sd(sample_data)/sqrt(n),
    mean(sample_data)+qnorm(0.975)*sd(sample_data)/sqrt(n))
})
#check how many contained the true value
mean((observed_CIs[1, ] < mu) & (observed_CIs[2, ] > mu))
```

```
[1] 0.9476
```