Statistical Computing Project 2

Mario Ibanez

February 19, 2016

Introduction

Discussion

Illustrative Example ?

Other random topics

Conclusion

1) Using sampling create B bootsrapped samples of size n from teh origianl sample.

2) For each boot strapped sample calculate:

$$\hat{\theta}_b^* \quad s_{\hat{\theta}_b^*} \quad \text{and} \quad T_b^* = \frac{\hat{\theta}_b^* - \hat{\theta}}{s_{\hat{\theta}_b^*}}$$

3) Put the B values of T_b^* in order

4) $t_{\alpha/2}$ is the 97.5% and $t_{1-\alpha/2}$ is the 2.5% of the sorted list of test stats

5) confidence interval is then:

$$[samplemean - t_{\alpha/2} s_{\hat{\theta}_b^*}, samplemean - t_{1-\alpha/2} s_{\hat{\theta}_b^*}]$$

Appendix

```
library(ggplot2)
haircuts <- c(17, 16.49, 45,
              25, 12, 18,
              13, 20, 10,
              0, 15, 50)

statistic_generator <- function(x, theta_hat){
  mean_x <- mean(x)
  se_x <- sd(x)/sqrt(length(x))
  return(c(mean_x,
           sd(x),
           (mean_x - theta_hat)/se_x))
}
```

```
x_bar <- mean(haircuts)
s_x_bar <- sd(haircuts)/sqrt(length(haircuts))

B <- 10^5
```

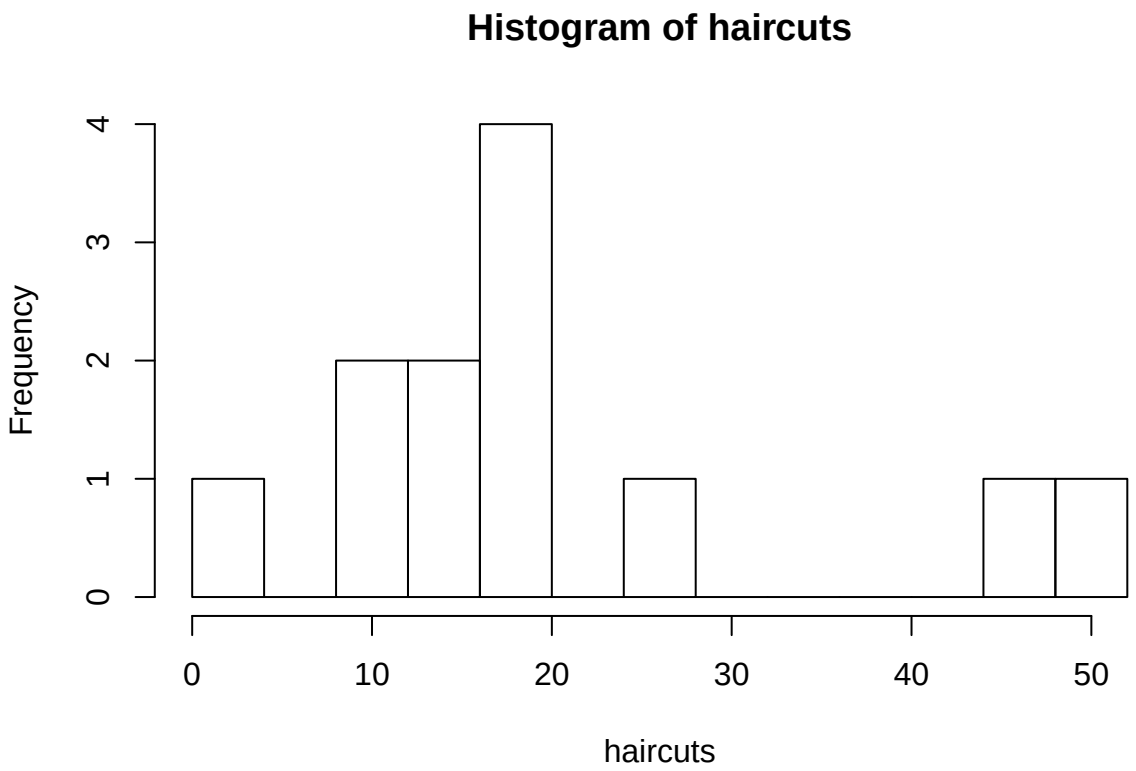
$$\hat{theta} = \overline{x} = 20.12417$$

$$s_{\hat{\theta}} = \frac{\overline{x}}{\sqrt{12}} = 14.18429$$

one interesting concern, what if all 12 values are the same and the sample standard deviation is 0. what would the test statistic be? Undefined.

histogram of haircuts

```
hist(haircuts, breaks = seq.int(from = 0, to = 52, by = 4))
```

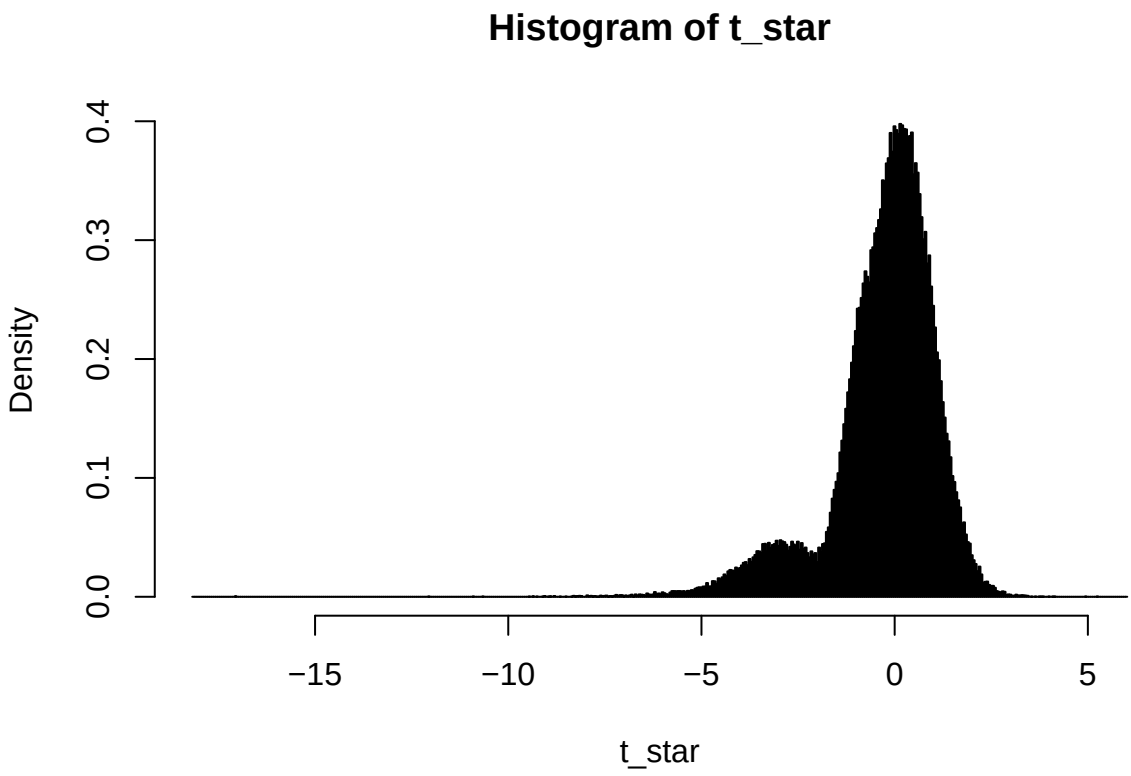


very skewed right, means most values are below the mean.

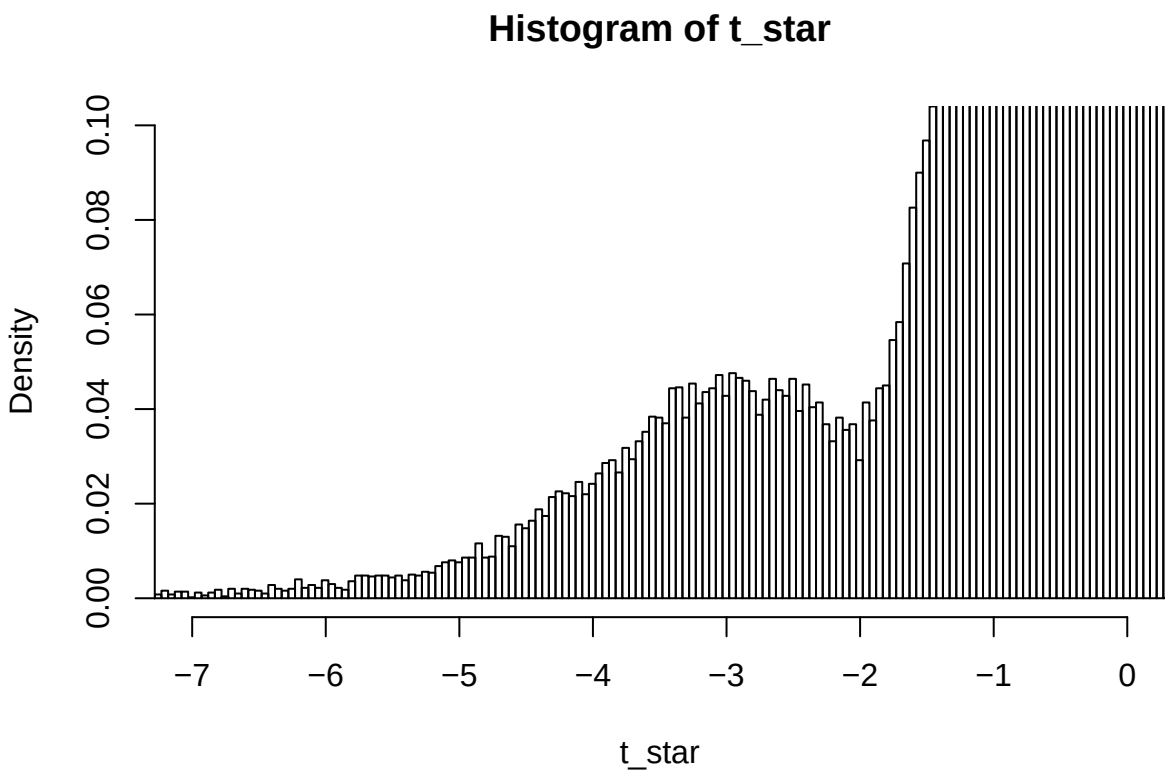
```
boot_strapped_haircuts <- matrix(data = sample(x = haircuts, size = 12*B, replace = TRUE),
                                 nrow = 12)

boot_statistics <- matrix(apply(X = boot_strapped_haircuts, MARGIN = 2, FUN = statistic_generator, theta
t_star <- boot_statistics[3, ]
sorted_t_star <- sort(t_star)
sample_means <- boot_statistics[1, ]
sample_stan_errors <- boot_statistics[2, ]
```

```r
left <- round(sorted_t_star[1]-1)
right <- round(sorted_t_star[B]+1)
hist(t_star, breaks = seq.int(left*101, right*101, by = 5)/100, freq = FALSE)
```




```r
hist(t_star, breaks = seq.int(left*101, right*101, by = 5)/100, xlim = c(-7, -0), freq = FALSE, ylim = c
```



Why are there two humps ?

Most values being below the mean means most test statistics are going to be negative.

```
t(boot_strapped_haircuts[, sorted_t_star < -12])
```

```
##      [,1] [,2] [,3] [,4]  [,5]  [,6]  [,7] [,8] [,9] [,10] [,11] [,12]
## [1,]   20   20   10   50 50.00 16.49 45.00   12   13    10    12 16.49
## [2,]   20   25   10    0 16.49 16.49 25.00   12   45    15    13 13.00
## [3,]   45   15   18   25  0.00 45.00 16.49   25   12    15    50 45.00
```

```
apply(t(boot_strapped_haircuts[, sorted_t_star < -12]), MARGIN = 1, FUN = statistic_generator, theta_ha
```

```
##            [,1]       [,2]      [,3]
## [1,] 22.9150000 17.5816667 25.957500
## [2,] 15.7384113 10.9397531 16.319031
## [3,]  0.6142761 -0.8050893  1.238263
```

probability of choosing a sample of all the same values?

1 in:

```
1 / (12 / (12^12))
```

```
## [1] 743008370688
```

Probability of avoiding this issue with 10^{5} bootstrapped samples?

```
(1 - (12 / (12^12)))^B
```

```
## [1] 0.9999999
```

extremely high...

how many unique bootstrapped samples are there?

$$\binom{n+k-1}{n-1} = \binom{n+k-1}{k}$$

where n and k are both 12. 23 choose 11

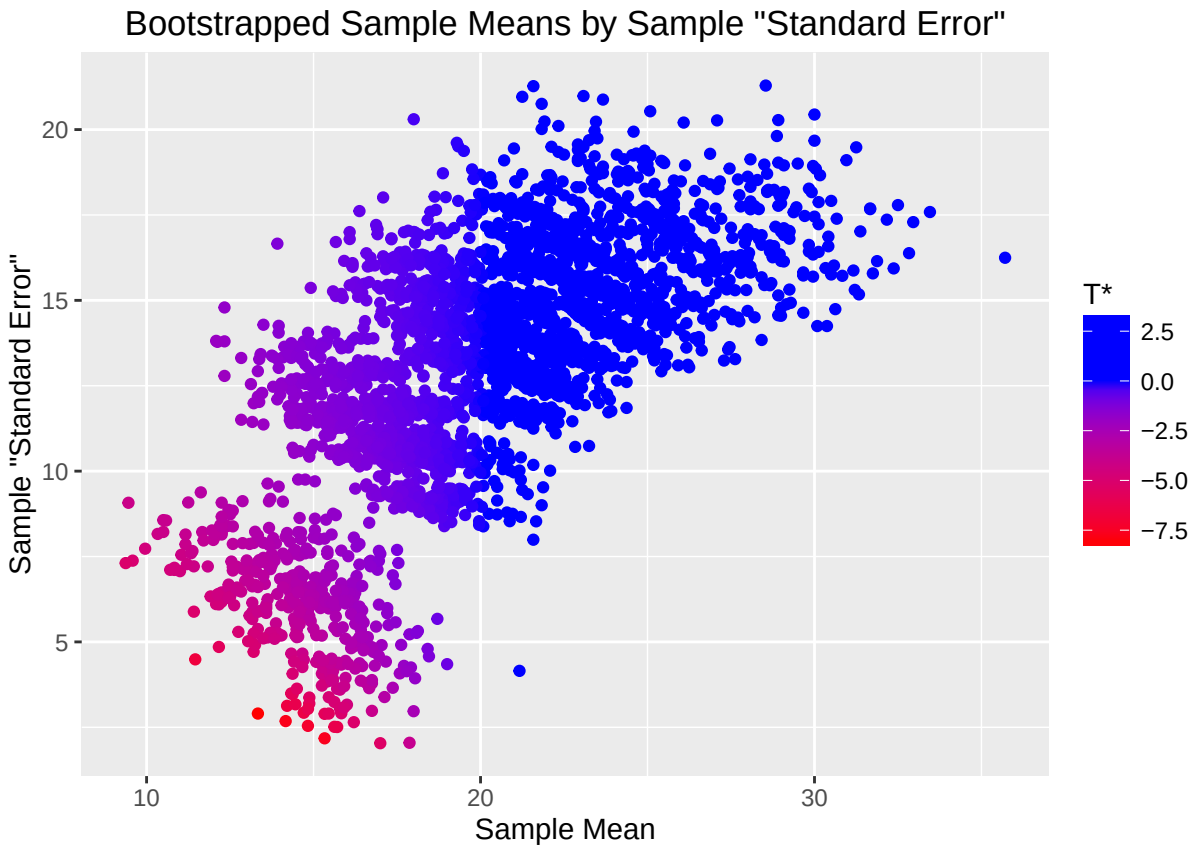
```
choose(23, 11)
```

```
## [1] 1352078
```

1,352,078 different bootstrapped samples are possible.

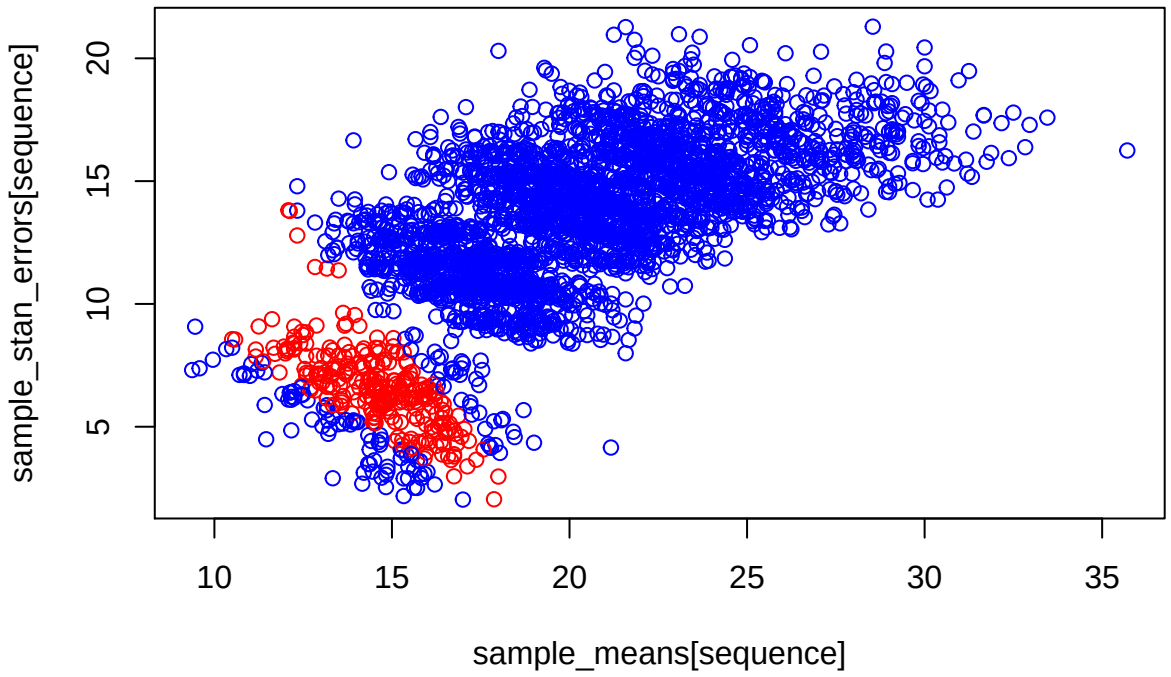
why are there two humps

```
sequence <- seq.int(from = 1, to = B, by = 30)
ggplot()+
  geom_point(aes(x = sample_means[sequence], y = sample_stan_errors[sequence],
                 colour = t_star[sequence]))+
  scale_colour_gradient2(low = "red", mid = "blue", high = "blue")+
  ggtitle("Bootstrapped Sample Means by Sample \"Standard Error\"")+
  labs(x="Sample Mean", y="Sample \"Standard Error\"", size = 20, colour = expression("T*"))+
  theme_grey(base_size = 11)
```



The low values of t* are all from the lower left.

```
colors <- rep("blue", B)
for(i in 1:B){
  if((t_star[i] < -2) * (t_star[i] > -4)){
    colors[i] <- "red"
  }
}
plot(sample_stan_errors[sequence] ~ sample_means[sequence], col = colors[sequence])
```


sample_stan_errors[sequence]
5
10
15
20
10
15
20
25
30
35
sample_means[sequence]