BootStrapping

```
# Example from
# An Introduction to the Bootstrap
# (Chapman & Hall/CRC Monographs on Statistics and Applied Probability) 1st Edition
# by Bradley Efron  (Author), R.J. Tibshirani (Author)


# Resampling Example for aspirin
#                     strokes      subjects
# aspirin group       119           11037
# placebo group       98            11034
#
# ratio of rates, theta_hat

defaultW <- getOption("warn")

options(warn = -1)





theta_hat = (119/11037) / (98/11034)
theta_hat
```

```
## [1] 1.213956
```

```
# We want to estimate the interval for true stroke ratio, theta, which is 0.93 to 1.59
# with 95% confidence level

aspirin_stroke_cases = rep(1,119)
#aspirin_stroke_cases

aspirin_no_stroke_cases = rep(0,11037-119)
#aspirin_no_stroke_cases

aspirin_group = c(aspirin_stroke_cases,aspirin_no_stroke_cases)
#aspirin_group

aspirin_group <- sample(aspirin_group)


placebo_stroke_cases = rep(1,98)
#placebo_stroke_cases

placebo_no_stroke_cases = rep(0,11034-98)
```

```r
#placebo_no_stroke_cases

placebo_group = c(placebo_stroke_cases,placebo_no_stroke_cases)
#placebo_group
placebo_group <- sample(placebo_group)


asp_bs_sample_case = sample(x= aspirin_group, size = 11037,replace = TRUE)
#asp_bs_sample_case

sum(asp_bs_sample_case)
```

```
## [1] 114
```

```r
plb_bs_sample_case = sample(x= placebo_group, size = 11037,replace = TRUE)
#sum(plb_bs_sample_case)

# lets take 1000 bootstrap samples

library(purrr)
k <- seq(1,1000)
asp_bs <- map(k,function(k){
  set.seed(k, sample.kind = "Rounding")
  return(sample(x= aspirin_group, size = 11037,replace = TRUE))
})

plb_bs <- map(k,function(k){
  set.seed(k, sample.kind = "Rounding")
  return(sample(x= placebo_group, size = 11034,replace = TRUE))
})


#str(asp_bs)
#typeof(asp_bs)
#class(asp_bs)

#str(asp_bs)

asp_bs[[1000]][45]
```

```
## [1] 0
```

```r
#summary(asp_bs)
k <- 1:1000
mean_asp <- map(k,function(k){
  numerator = sum(asp_bs[[k]])/11037
  denom = sum(plb_bs[[k]])/11034
  return(numerator/denom)
})

mean_asp1 <- unlist(mean_asp)
#mean_asp1
class(mean_asp1)
```

```
## [1] "numeric"
```

```
print("mean:")
```

```
## [1] "mean:"
```

```
mean(mean_asp1)
```

```
## [1] 1.23462
```

```
print("standard deviation:")
```

```
## [1] "standard deviation:"
```

```
sd(mean_asp1)
```

```
## [1] 0.1667418
```

```
# 95% confidence interval
print("95% confidence interval:")
```

```
## [1] "95% confidence interval:"
```

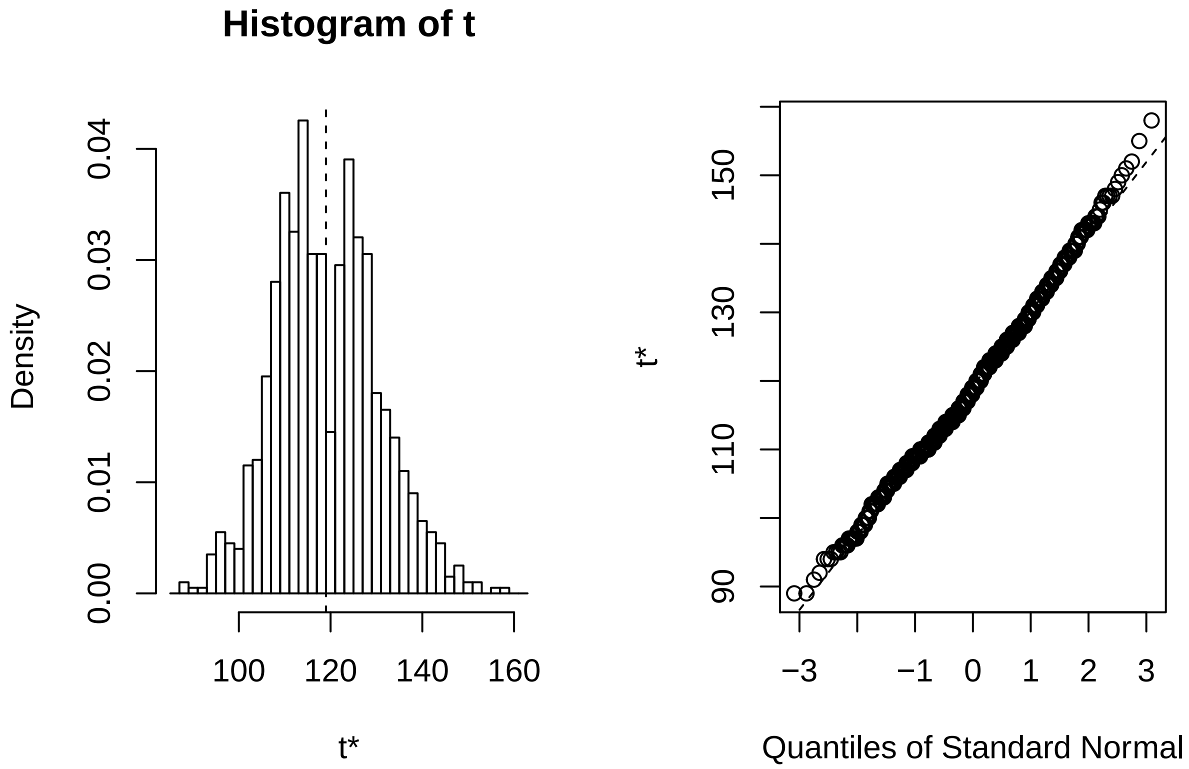
```
quantile(mean_asp1,c(25/1000,975/1000))
```

```
##      2.5%     97.5%
## 0.9240862 1.5829473
```

```
# Now we do the same analysis using bootstrap package boot

library(boot)

set.seed(100, sample.kind = "Rounding")

sum_ones_as <- function(data,indices){ return(sum(data[indices]))}

results_as <- boot(data = aspirin_group,statistic = sum_ones_as, R = 1000)
results_as
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = aspirin_group, statistic = sum_ones_as, R = 1000)
##
##
## Bootstrap Statistics :
##     original  bias    std. error
## t1*      119   0.255    10.89076
```

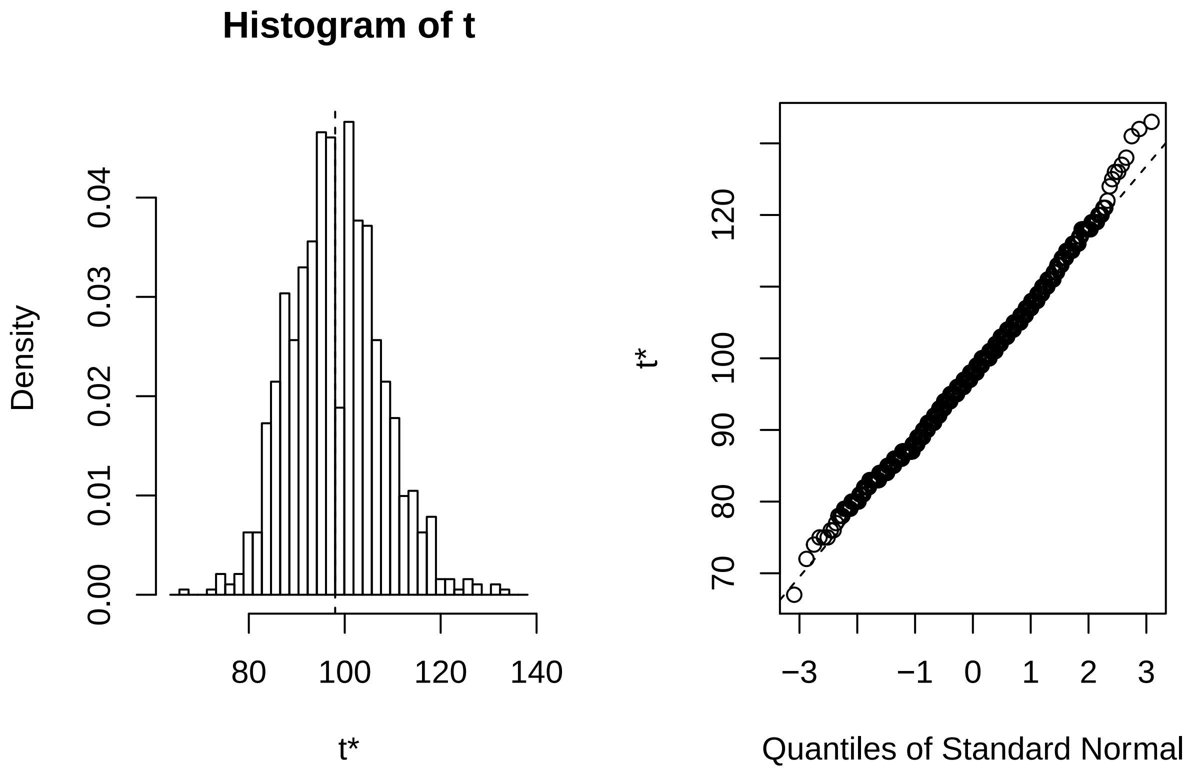
```
plot(results_as)
```



```
#results_as$t

sum_ones_pl <- function(data,indices){ return(sum(data[indices]))}

results_pl <- boot(data = placebo_group,statistic = sum_ones_pl, R = 1000)
results_pl
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = placebo_group, statistic = sum_ones_pl, R = 1000)
##
##
## Bootstrap Statistics :
##     original  bias    std. error
## t1*       98   0.155    9.555327
```

```
plot(results_pl)
```



```r
#results_pl$t

ratio <- results_as$t/results_pl$t
#ratio

print("mean:")
```

```
## [1] "mean:"
```

```r
mean(ratio)
```

```
## [1] 1.22675
```

```r
print("standard deviation:")
```

```
## [1] "standard deviation:"
```

```r
sd(ratio)
```

```
## [1] 0.1670607
```

```
# 95% confidence interval
print("95% Confidence Interval:")
```

```
## [1] "95% Confidence Interval:"
```

```
quantile(ratio,c(25/1000,975/1000))
```

```
##      2.5%     97.5%
## 0.9411471 1.6048290
```

```
#Reference
# https://www.statmethods.net/advstats/bootstrapping.html

options(warn = defaultW)
```