Probabilità e Statistica 2

Execution Time

Claudio Agostinelli
Dipartimento di Scienze Ambientali, Informatica e Statistica
Università Ca' Foscari, Venezia

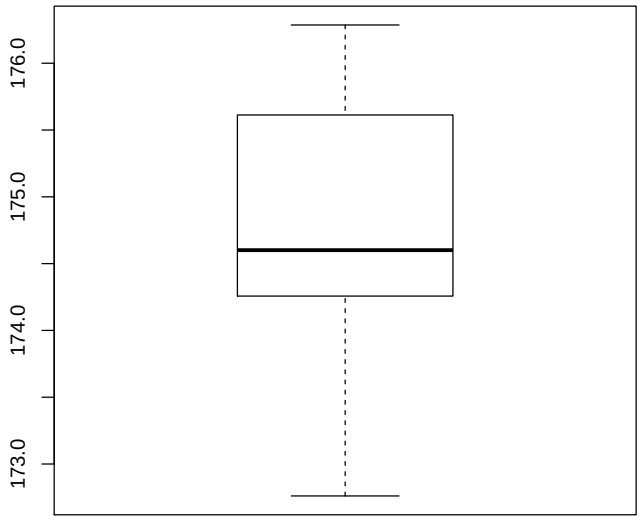
Ver 1.1 – 06 Febbraio 2014

1 Introduzione

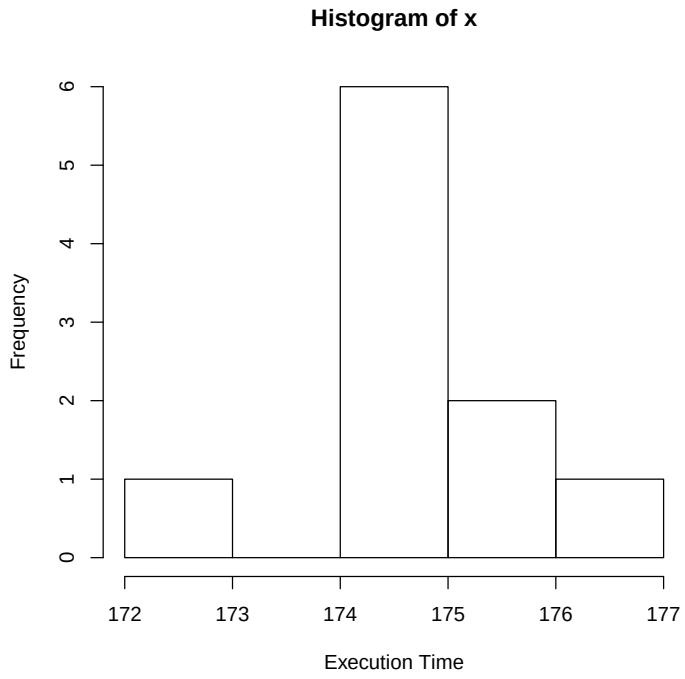
```
> load(file = "./data/ExecutionTime.RData")
```

2 Analisi grafica

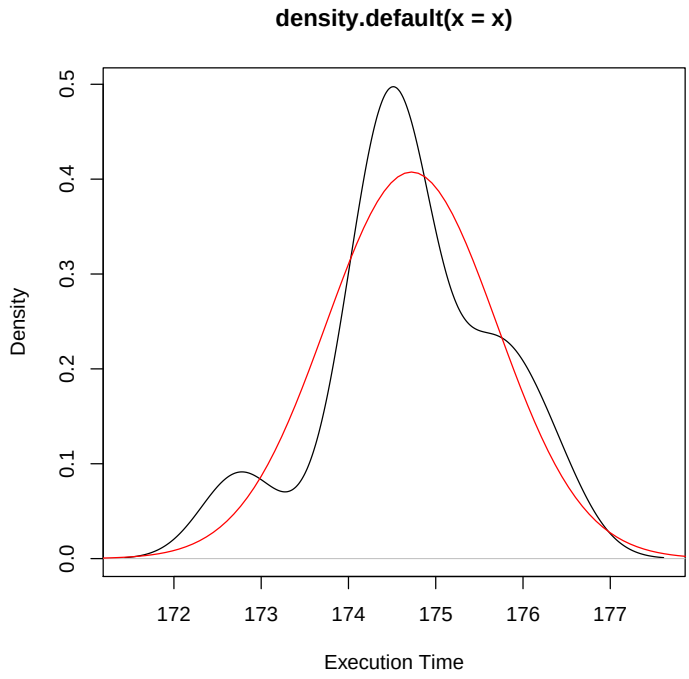
```
> boxplot(x, xlab="Execution Time")
```



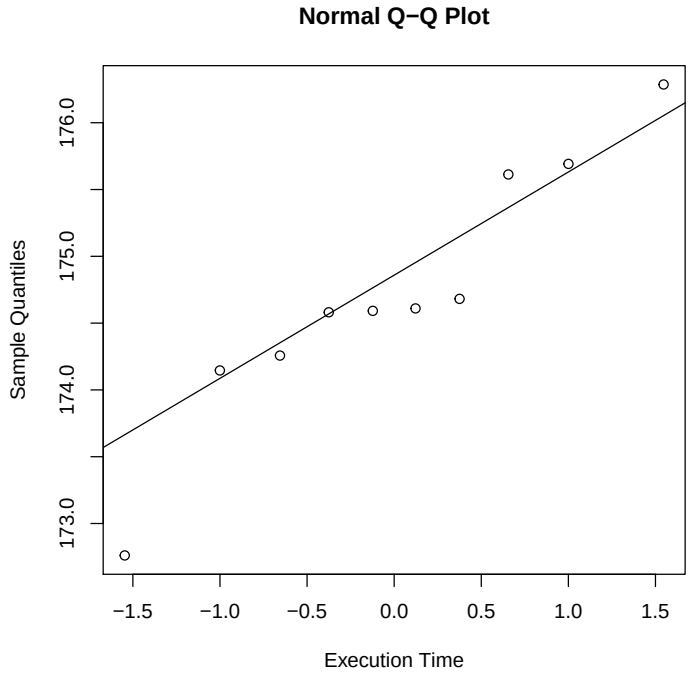

```
> hist(x, xlab="Execution Time")
```



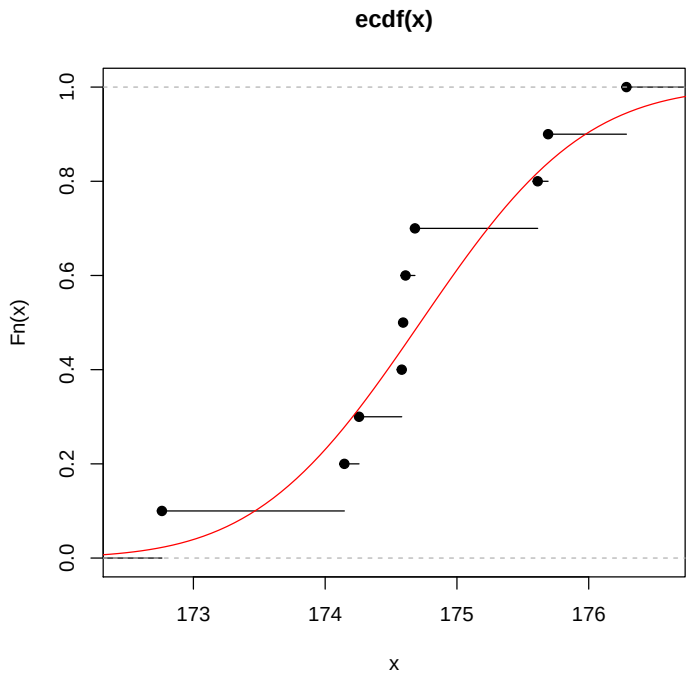

```
> plot(density(x), xlab="Execution Time")
> plot(function(z) dnorm(z, mean=mean(x), sd=sd(x)),
+      from=mean(x)-4*sd(x), to=mean(x)+4*sd(x), add=TRUE, col=2)
```




```
> qqnorm(x, xlab="Execution Time")
> qqline(x)
```



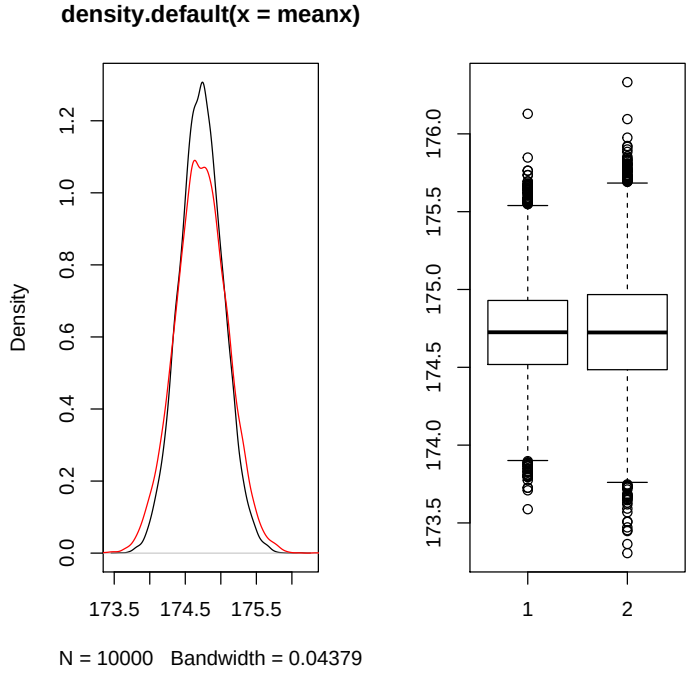
```
> mx <- mean(x)
> sx <- sd(x)
> plot(ecdf(x))
> plot(function(x) pnorm(x, mx, sx), from=mx-3*sx, to=mx+3*sx, col=2, add=TRUE)
```



3 Proprietà di media e mediana usando tecniche Monte Carlo

```
> n <- length(x)
> nrep <- 10000
> meanx <- medianx <- rep(NA, nrep)
> set.seed(1234)
> system.time(
+ for (i in 1:nrep) {
+   x <- rnorm(n=n, mean=mx, sd=sx)
+   meanx[i] <- mean(x)
+   medianx[i] <- median(x)
+ }
+ )

   user  system elapsed
  0.664   0.005   0.669

> par(mfcol=c(1,2))
> plot(density(meanx))
```

```
> lines(density(medianx), col=2)
> boxplot(meanx, medianx)
```




```
> var(meanx)

[1] 0.09499514

> var(medianx)

[1] 0.1314064

> sx^2/length(x)

[1] 0.09584556

> (4*n*dnorm(x=0, sd=sx)^2)^(-1)

[1] 0.1505539

> locationnorm <- function(size, mean, sd) {
+   x <- rnorm(n=size, mean=mean, sd=sd)
+   m1 <- mean(x)
+   m2 <- median(x)
+   return(c(m1,m2))
+ }
> system.time(
+   res <- replicate(n=nrep, locationnorm(n, mx, sx))
+ )
```

```
   user  system elapsed
  0.704   0.005   0.712
```

```
> apply(res, 1, var)
```

```
[1] 0.09323551 0.13030289
```

4 Proprietà di media e mediana usando tecniche bootstrap

```
> meanxboot <- medianxboot <- rep(NA, nrep)
> for (i in 1:nrep) {
+   xtemp <- x[sample.int(n, replace=TRUE)]
+   meanxboot[i] <- mean(xtemp)
+   medianxboot[i] <- median(xtemp)
+ }
> var(meanxboot)
```

```
[1] 0.2596356
```

```
> var(medianxboot)
```

```
[1] 0.6105402
```

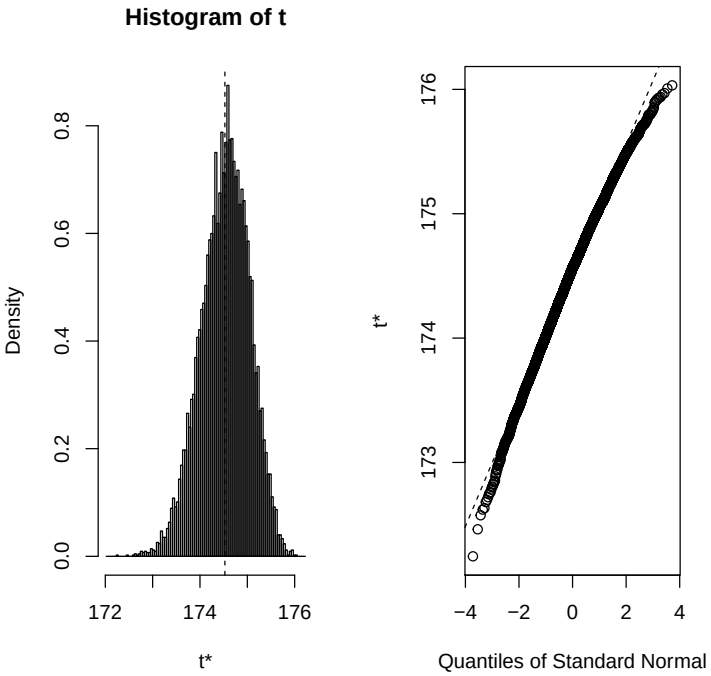
```
> library(boot)
> res1 <- boot(x, function(z, inds) mean(z[inds]), R=10000)
> res2 <- boot(x, function(z, inds) median(z[inds]), R=10000)
> res1
```

```
ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = x, statistic = function(z, inds) mean(z[inds]), R = 10000)


Bootstrap Statistics :
    original      bias    std. error
t1* 174.5261 0.006832416   0.5117979
```

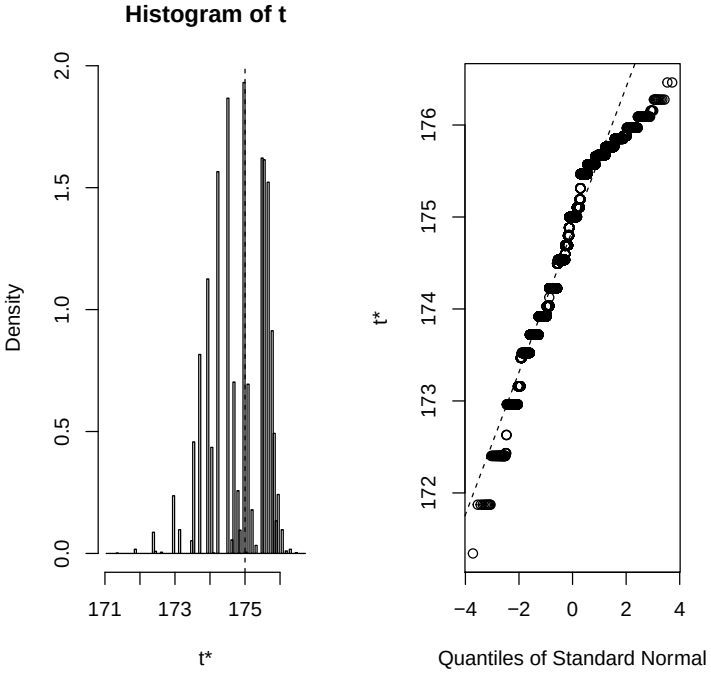
```
> res2
```

```
ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = x, statistic = function(z, inds) median(z[inds]),
    R = 10000)


Bootstrap Statistics :
    original     bias    std. error
t1* 175.0003 -0.1377014    0.774746
```

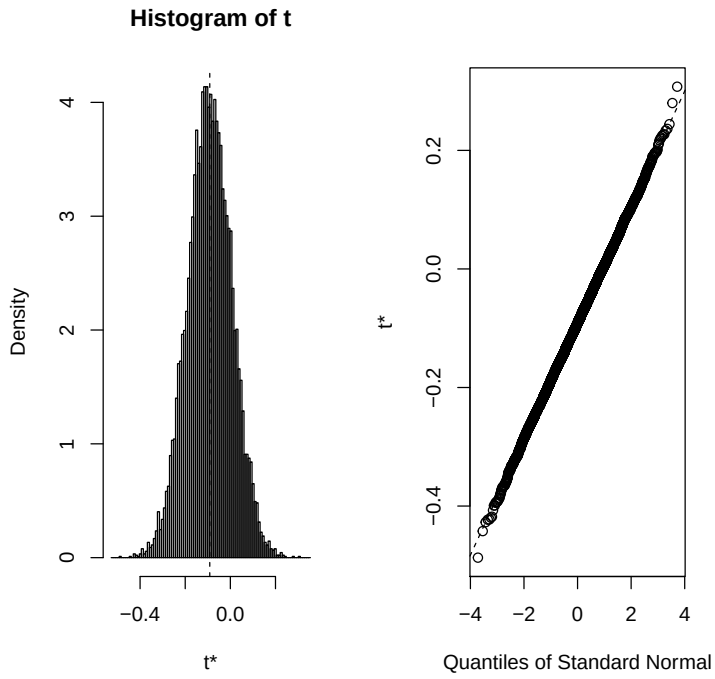
```
> plot(res1)
```



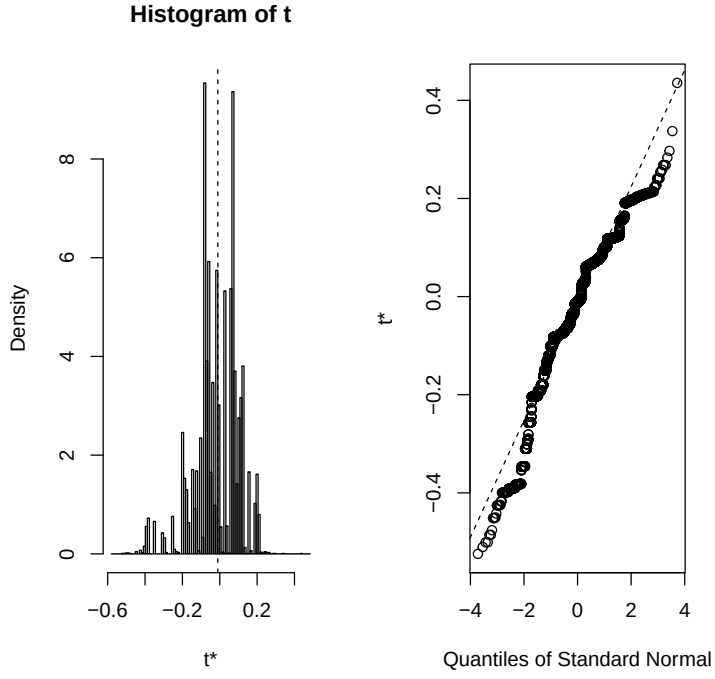

```
> plot(res2)
```



```
> y <- rnorm(100)
> res3 <- boot(y, function(z, inds) mean(z[inds]), R=10000)
> res4 <- boot(y, function(z, inds) median(z[inds]), R=10000)
> res3

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = y, statistic = function(z, inds) mean(z[inds]), R = 10000)


Bootstrap Statistics :
       original        bias    std. error
t1* -0.09121368 -0.0001303863  0.09783965

> res4

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = y, statistic = function(z, inds) median(z[inds]),
    R = 10000)


Bootstrap Statistics :
       original       bias    std. error
t1* -0.01041024 -0.003760915   0.1190498

> plot(res3)
```


Histogram of t
Density
t*
t*
Quantiles of Standard Normal

```
> plot(res4)
```


Histogram of t
Density
t*
t*
Quantiles of Standard Normal

```
> x.fun <- function(data) {
+   mx <- mean(data)
+   mx
+ }
> x.rg <- function(data, mle) {
+   rnorm(length(data), mle, 2.019148)
+ }
> res5 <- boot(x, x.fun, R = 1000, sim = "parametric", ran.gen = x.rg, mle = mean(x))
> res5

PARAMETRIC BOOTSTRAP


Call:
boot(data = x, statistic = x.fun, R = 1000, sim = "parametric",
    ran.gen = x.rg, mle = mean(x))


Bootstrap Statistics :
    original     bias    std. error
t1* 174.5261 0.01367735   0.6441316

> plot(res5)
```