ETON graphing

Piotr Chlebicki

2022-06-05

Function that computes $T_\beta x = \beta x (mod 1)$

```
Tb <- function(beta) {
  function(x, ign = NULL) {
    (beta * x) %% 1
  }
}
```

Function that computes sum:

$$\sum_{x < T_\beta^n 1} \frac{1}{\beta^n}$$

indecies for which sum will be made are computed as a part of function factory, condition $T_\beta^n 1 \leq (2 \cdot \varepsilon)^{\frac{1}{4}}$ is here because of computational zeros. Since for large n $\frac{1}{\beta^n} \approx 0$ and we only have finite computational power I'll only check first 2147484 n's

```
Sigma1b <- function(beta) {
  TbI <- Tb(beta)
  n <- 1
  Vcomp <- NULL

  while(n < .Machine$integer.max / 1000) {
    if (Reduce(f = TbI, x = 1:n, init = 1) <= .Machine$double.eps ** .25) {
      n <- n - 1
      break
    }
    n <- n + 1
  }

  Vcomp <- Reduce(f = TbI, x = 1:n, init = 1, accumulate = TRUE)

  function(x) {
    if ((x > 1) || (x < 0)) {return(0)}
    TOSUM <- (c(which(x < Vcomp)) - 1)
    sum((1 / beta) ** TOSUM)
  }
}
```

Finally a function that normalises above sum to make it a probability density function:

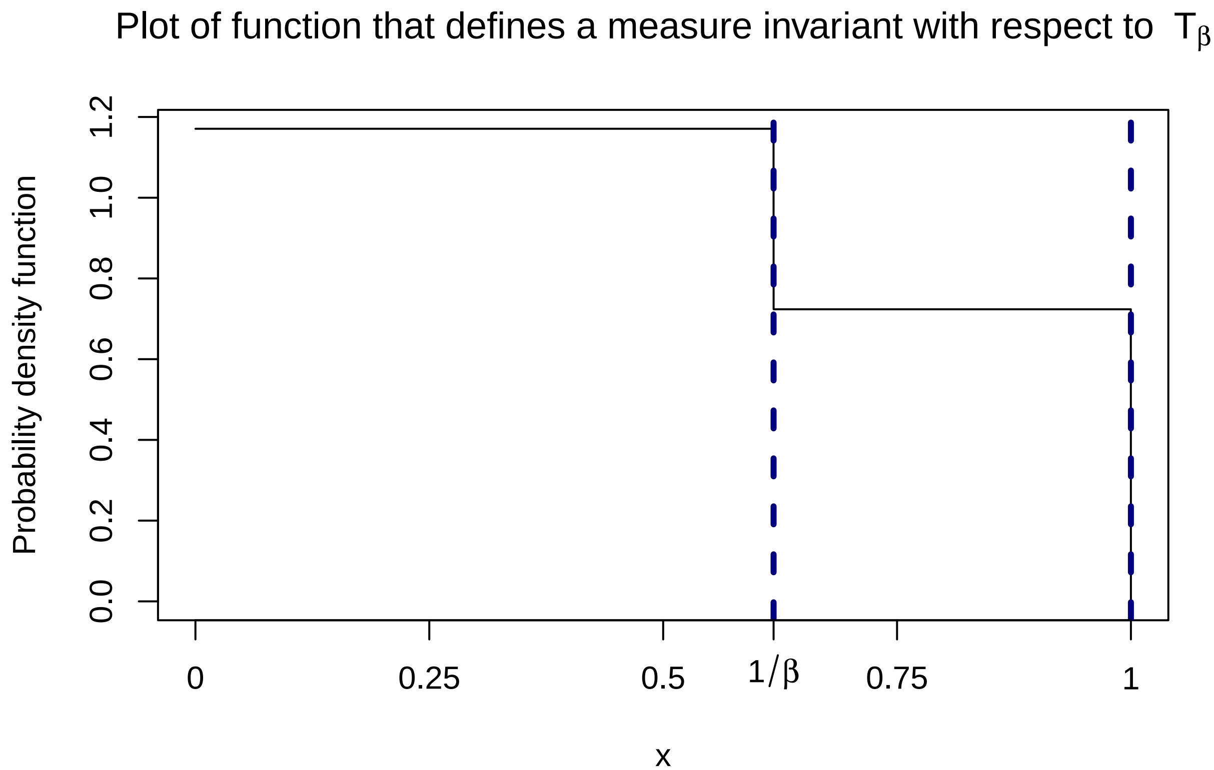
```
Normalise <- function (f) {
  g <- function(x) {sapply(X = x, FUN = f)}

  Constant <- integrate(f = g,
                        lower = 0,
                        upper = 1,
                        rel.tol = .Machine$double.eps ** .5,
                        subdivisions = 100000)$value

  function(x) {g(x) / Constant}
}
```

Presentation for $\beta = \varphi = \frac{1+\sqrt{5}}{2}$

```
FF <- Normalise(Sigma1b(beta = (1 + sqrt(5)) / 2))
curve(FF, from = 0, to = 1, n = 10000,
      ylab = "Probability density function",
      main = expression("Plot of function that defines a measure invariant with respect to " ~ T[beta]),
      xaxt='n')
abline(v = (sqrt(5) - 1) / 2,
       lty = 2,
       col = "navy",
       lwd = 3)
abline(v = 1,
       lty = 2,
       col = "navy",
       lwd = 3)
axis(side = 1,
     at = c(0, .25, .5, (sqrt(5) - 1) / 2, .75, 1),
     labels = c(0, .25, .5, expression(1 / beta), .75, 1))
```



A confirmation that FF is a PDF with support on $[0, 1]$

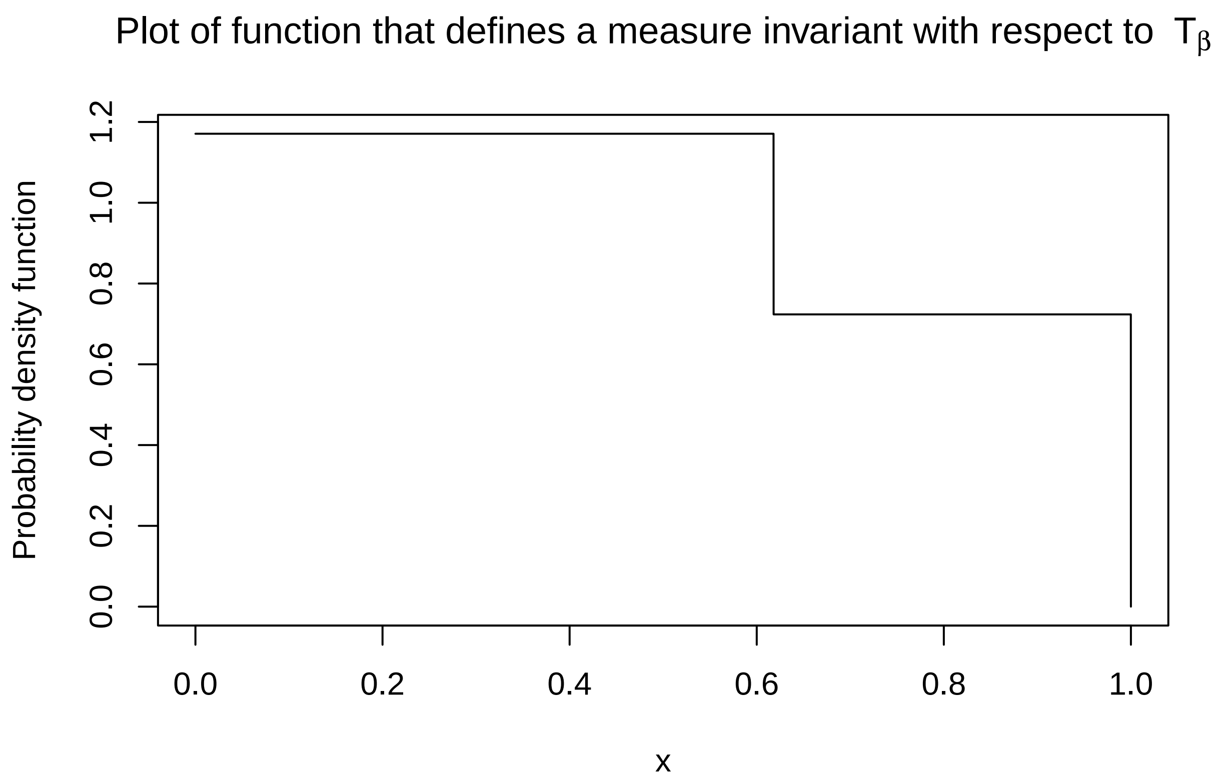
```
integrate(FF, lower = 0, upper = 1, rel.tol = .Machine$double.eps ** .5)
```

```
## 1 with absolute error < 1.2e-08
```

function to make graphs automatically:

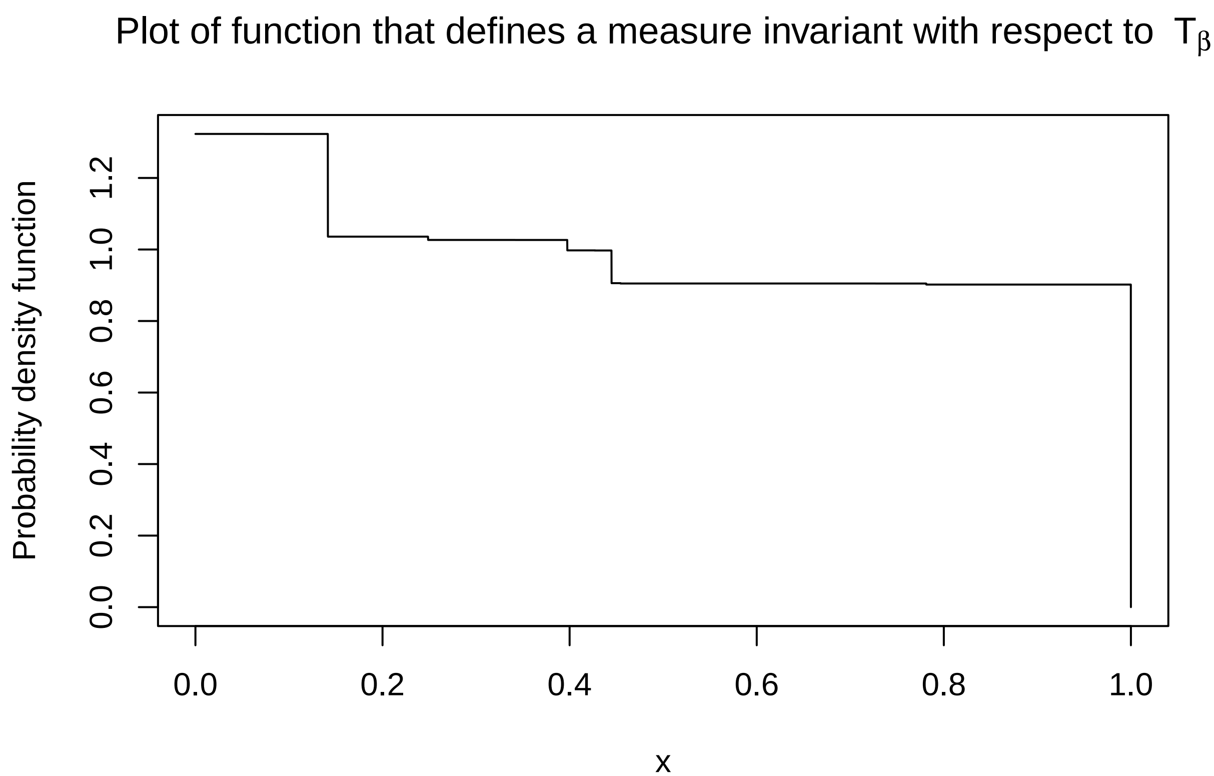
```
graphpdf <- function(FF, beta, n = 10000, ...) {
  curve(FF, from = 0, to = 1, n = 10000,
        ylab = "Probability density function",
        main = expression("Plot of function that defines a measure invariant with respect to " ~ T[beta]
        ...)
  vectposition <- NULL
  #vectexpression <- NULL
  #k <- 1
  #while(k / beta < 1) {
  #  vectposition <- c(vectposition, k / beta)
  #  vectexpression <- c(vectexpression,
  #                      eval(substitute(expr = expression(r / beta),
  #                                      env = list(r = k))))
  #  k <- k + 1
  #}
  #for (v in c(1:k/beta,1)) {
  #  abline(v = v,
```

```r
  #        lty = 2,
  #        col = "navy",
  #        lwd = 3,
  #        ...)
  #}
  # I tried to add "discontinuity points but I didn't work :(
}
graphpdf(FF = FF, beta = (1 + sqrt(5)) / 2)
```



Example for π expansion

```r
FF <- Normalise(Sigma1b(beta = pi))
graphpdf(FF = FF, beta = pi)
```

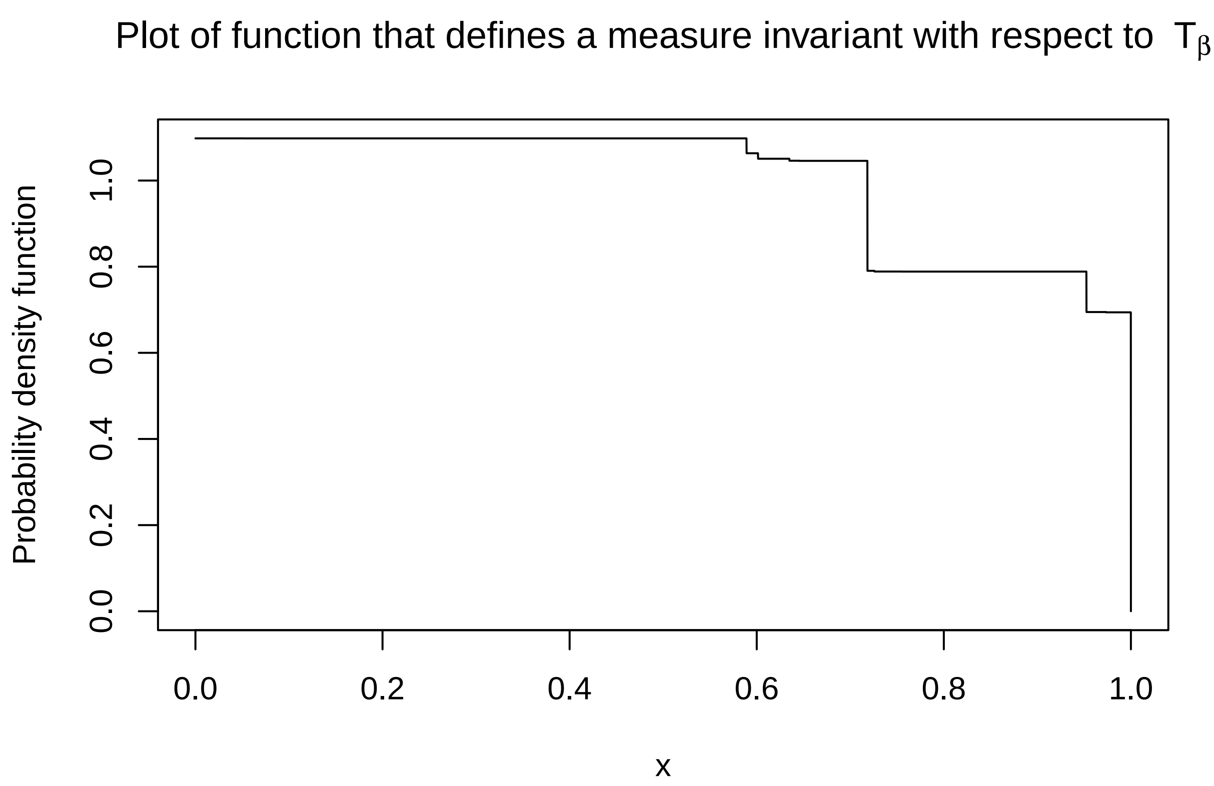



```
integrate(FF, lower = 0, upper = 1,
          rel.tol = .Machine$double.eps ** .5,
          subdivisions = 100000)
```

```
## 1 with absolute error < 1.4e-08
```

For e

```
FF <- Normalise(Sigma1b(beta = exp(1)))
graphpdf(FF = FF, beta = exp(1))
```

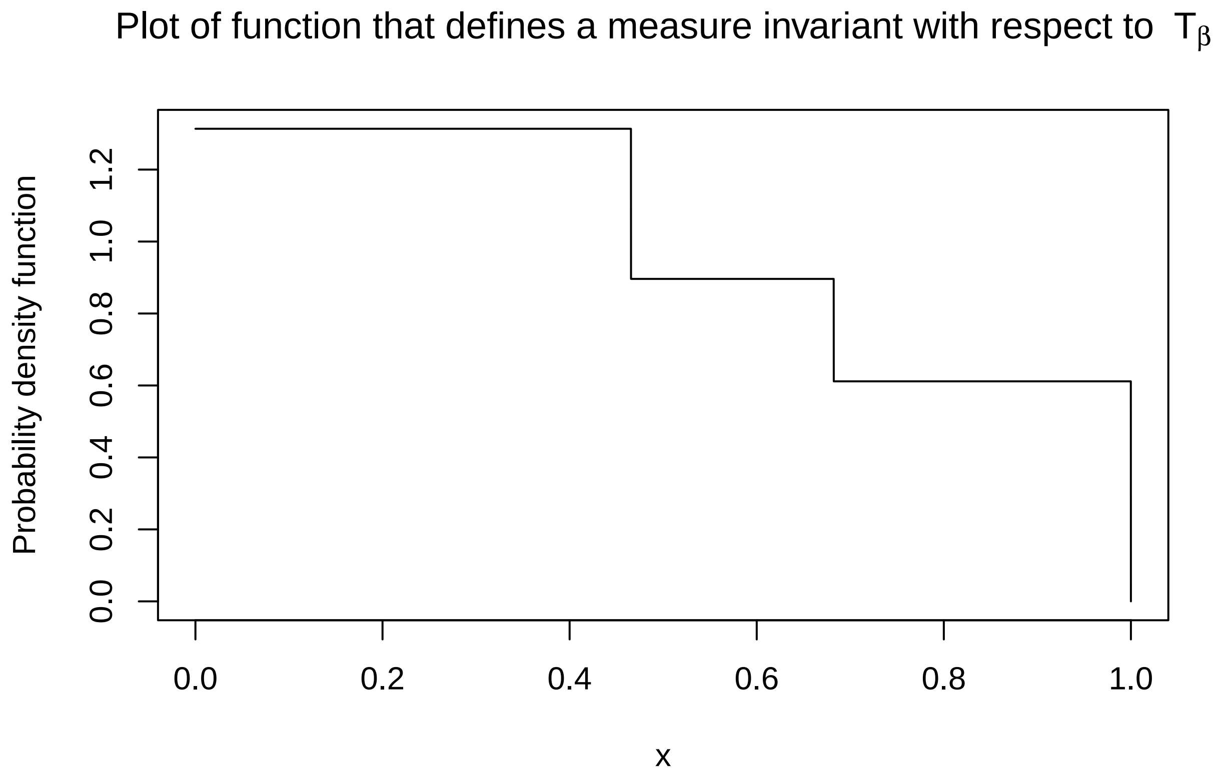



```
integrate(FF, lower = 0, upper = 1,
          rel.tol = .Machine$double.eps ** .5,
          subdivisions = 100000)
```

```
## 1 with absolute error < 1.1e-08
```

For super golden ratio $\psi = \frac{1+\sqrt[3]{\frac{29+3\sqrt{93}}{2}}+\sqrt[3]{\frac{29-3\sqrt{93}}{2}}}{3}$

```
FF <- Normalise(Sigma1b(beta =
      (1 + ((29 + 3 * sqrt(93)) / 2) ** (1 / 3) +
      ((29 - 3 * sqrt(93)) / 2) ** (1 / 3)) / 3))
graphpdf(FF = FF, beta =
      (1 + ((29 + 3 * sqrt(93)) / 2) ** (1 / 3) +
      ((29 - 3 * sqrt(93)) / 2) ** (1 / 3)) / 3)
```

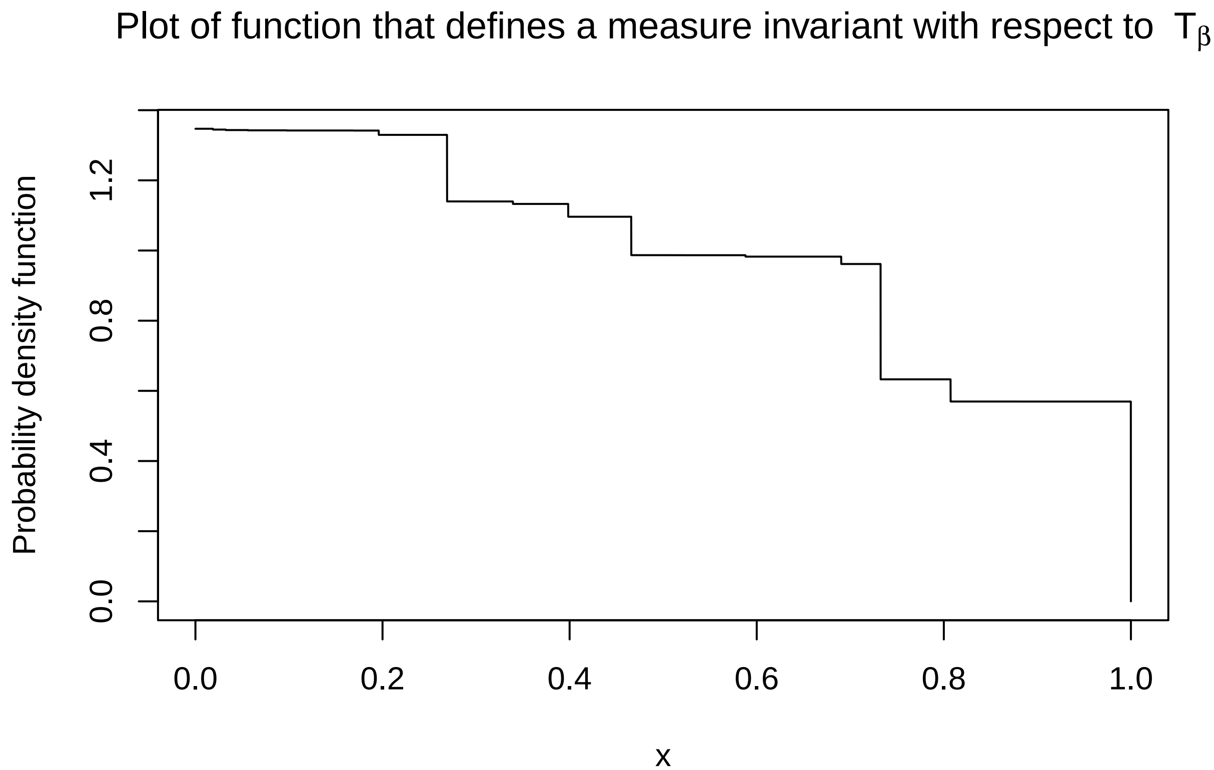



```r
integrate(FF, lower = 0, upper = 1,
          rel.tol = .Machine$double.eps ** .5,
          subdivisions = 100000)
```

```
## 1 with absolute error < 7.4e-09
```

For $\frac{1}{\gamma} \approx 1.732455$

```r
FF <- Normalise(Sigma1b(beta = 1 / -digamma(1)))
graphpdf(FF = FF, beta = 1 / -digamma(1))
```




```
integrate(FF, lower = 0, upper = 1,
          rel.tol = .Machine$double.eps ** .5,
          subdivisions = 100000)
```

```
## 1 with absolute error < 6.8e-09
```