Likelihood.R

georges

2024-10-12

Contents

```
library(latticeExtra)
```

```
## Loading required package: lattice
```

```
library(spida2)
Green <- "#008800"
```

```r
knitr::opts_chunk$set(fig.width=6.3, fig.height = 3.3)
pause <- function() invisible()
```

Samples from the exponential distribution:

e.g.

- wait times where theta is mean waiting time
- lifetimes of non-aging components with no lemons where theta is mean lifetime

$$f(x; theta) = \frac{1}{\theta} e^{-x/\theta} \qquad x, \theta > 0$$

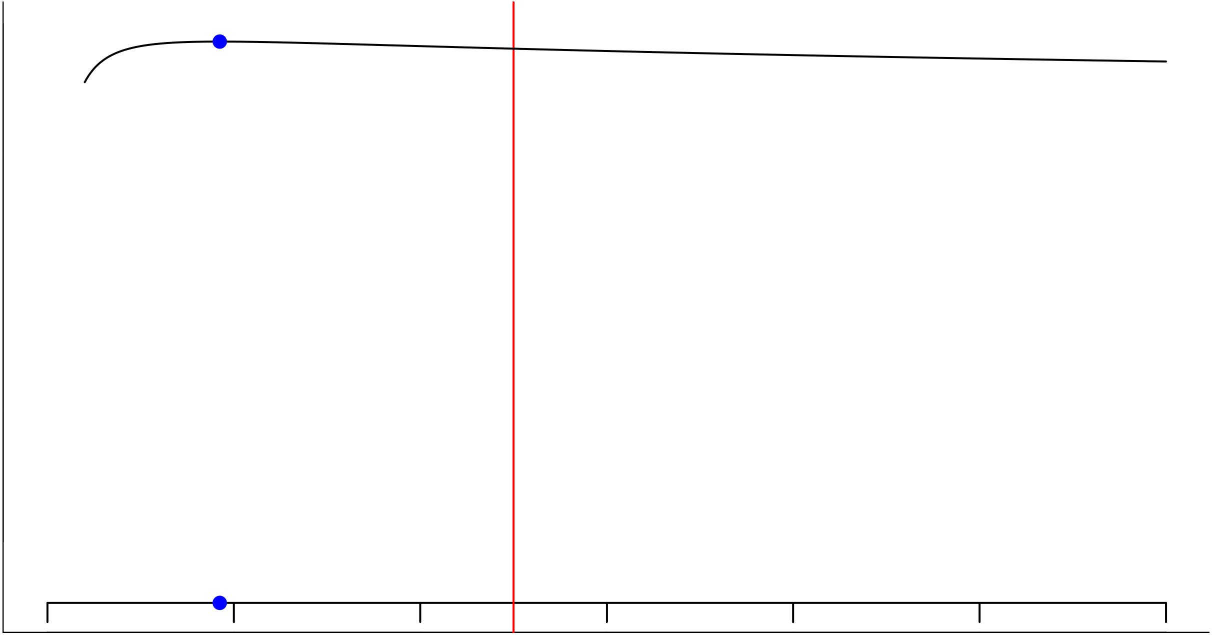
```r
#
#  Exponential distribution with mean parametrization
#

{

  # Maximum likelihood estimator
```

```r
mle <- function(x) mean(x)

# log-likelihood

lx <- function(x,thetas) {
  # log-likelihood from sample x from exponential theta
  # evaluated for a fixed sample x over thetas
  n <- length(x)
  sx <- sum(x)
  -n*log(thetas) - sx / thetas
}

lxn <- function(x, thetas) {
  # normed log-likelihood
  lx(x, thetas) - lx(x, mle(x))
}

# Score: derivative of log-likelihood: gradient
```

```
lx. <- function(x,thetas) {
  # slope (first derivative) of log-likelihood
  n <- length(x)
  sx <- sum(x)
  -n/thetas + sx / thetas^2
}

# Negative of Observed Fisher Information: second derivative of log-likelihood: Hessia

lx.. <- function(x,thetas) {
  # curvature (second derivative) of log-likelihood
  n <- length(x)
  sx <- sum(x)
  n/thetas^2 - 2 * sx / thetas^3
}

qappx <- function(x, thetas) {
  lx0 <- lx(x, mle(x))
  lxq <- lx..(x, mle(x))
```

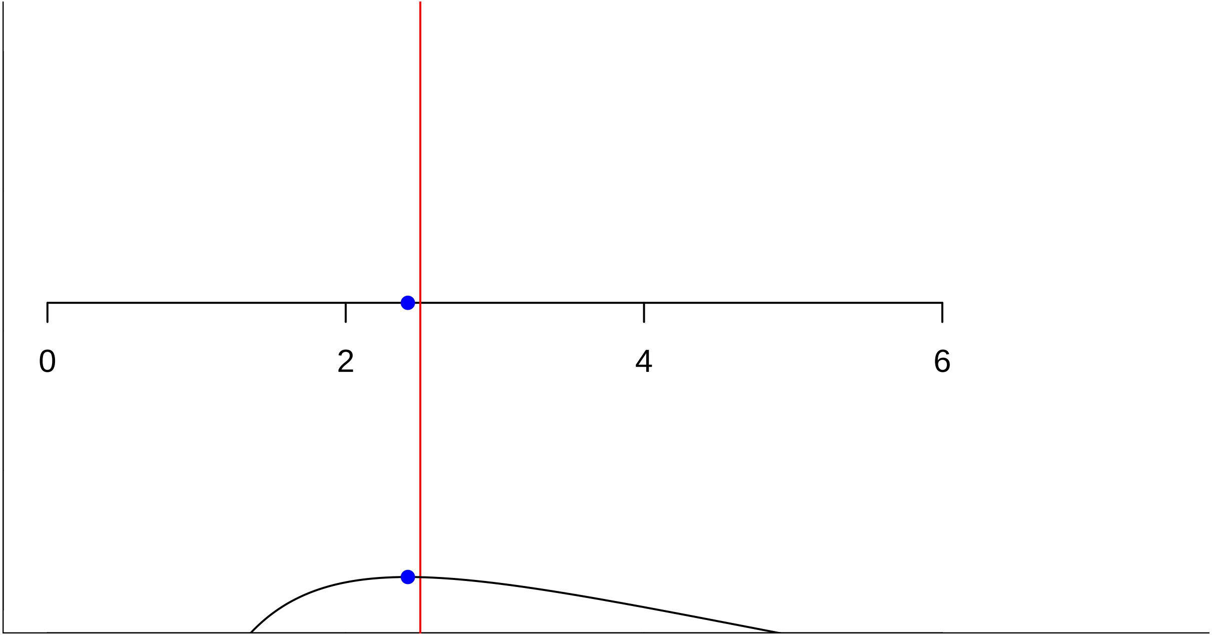
```
    lx0 + lxq * (thetas - mle(x))^2
  }

  # Generate random sample using mean parameter

  rexp2 <- function(n, theta) {
    # random sample of size n
    # using waiting time as parameter, instead of rate
    x <- rexp(n, rate = 1/theta)
    attr(x,'theta') <- theta
    class(x) <- 'sample'
    print(as.vector(x))
    x
  }
}


{
  # setup
```

```r
  n <- 30
  theta_true <- 2.5

  low <- lx(rep(3*theta_true,n), 3 * theta_true)
  par(mar=c(0,0,0,0))
  plot(c(0,6),c(low,0), type = 'n')
  axis(1, pos = low + 1)
  ths <- seq(.2, 6, .01)

  col <- 'blue'
  abline(v = theta_true, col = 'red')
}
{
  n <- 3
  # log-likelihood
  x <- rexp(n, 1/theta_true )
  print(x)
  lines(ths, lx(x, ths))
  points(mle(x), lx(x, mle(x)), pch = 16, col = col)
  points(mle(x), low + 1, pch = 16, col = col)
  # lines(ths, qappx(x, ths), col = 'red')
}
```



```
## [1] 0.07504909 1.38884595 1.30905439
```

```r
{

  # log-likelihood

  col <- 'blue'
  theta_true <- 2.5
  ths <- seq(.02, 3 * theta_true, .01)
  plot(c(0, 3* theta_true), c(-20, 1), type = 'n')
  axis(1, pos = -9)
  abline(v = theta_true, col = 'red')

  n <- 10
  {
    # Score
    x <- rexp(n, 1/2.5 )
    print(x)
    lines(ths, lx(x, ths))
    points(mle(x), lx(x, mle(x)), pch = 16, col = col)
    points(mle(x), -9, pch = 16, col = col)

  }
}
```


0
2
4
6

```
##  [1] 6.8497235 4.2753986 1.7216940 1.5404830 2.2456327 1.3270874 3.1366149
##  [8] 0.5111745 2.4142671 0.1436774
```

```r
{
  init.ex <- function(xlim = c(0,5)){
    graphics.off()
    plot(1, type = 'n',xlim = xlim, ylim = c(-10,4),
         xlab = expression(~theta), bty = 'n',
         ylab = expression(~l[x](theta)), axes = FALSE)
    lhat <- -1
    axis(1,pos = lhat)
    text(-.2,lhat, expression(~hat(theta)), xpd = TRUE)
    axis(1,pos = -10)
    inc <- .8
    lp0 <- .5
    lpp0 <- .5 + inc
    lptrue <- .5 + 2* inc
    lpptrue <- .5 + 3 * inc
    lpphat <- .5 + 4 * inc
    seg <- function(h) segments(xlim[1]+.2,h,xlim[2],h)
```

```r
text(0,lp0,expression(l*minute[x](theta[0])))
seg(h = lp0)
text(0,lpp0,expression(l*second[x](theta[0])))
seg(h = lpp0)
text(0,lptrue, expression(l*minute[x](theta[TRUE])))
seg(h=lptrue)
text(0,lpptrue, expression(l*second[x](theta[TRUE])))
seg(h=lpptrue)
text(0,lpphat, expression(l*second[x](hat(theta))))
seg(h=lpphat)

segments(mean(xlim)+c(0,1,-1),.5,mean(xlim)+c(0,1,-1),.3)
text(mean(xlim)+c(0,1,-1),.3, c(0,1,-1), pos = 1)
# text(0,.5+.2,expression(l*minute[x](theta[0])))
# seg(h = .5)
# text(0,0 +.2,expression(l*second[x](theta[0])))
# seg(h = 0)
# segments(mean(xlim)+c(0,1,-1),0,mean(xlim)+c(0,1,-1),-0.1)
# text(mean(xlim)+c(0,1,-1),-0.1, c(0,1,-1), pos = 1)
```

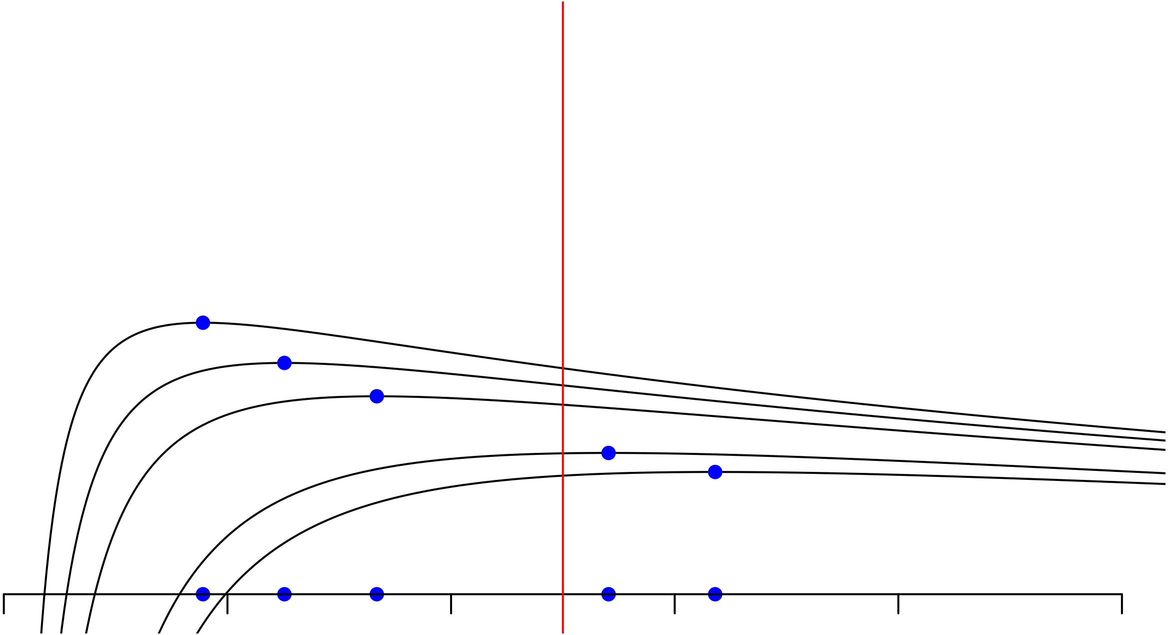
```
    #
    parms <- list(lhat = lhat,lp0 =lp0,lpp0 = lpp0)
    parms
  }



  }


{

  init <- function(
    xlim=c(0,5), ylim = c(-10,5)    ){
    # graphics.off()
    par(mar=c(0,0,0,0))
    plot(xlim,ylim, type = 'n', bty = 'n', axes = FALSE)
    # axis(1,pos=ylim[1])
    # axis(2,pos=0)
  }
}
```

```
init()

ths <- seq(.02,7,.01)
plot_lx <- function(xs, theta_true = NULL, theta_0 = NULL, ...){
  init()
  axy <- par('usr')[3]+1
  lapply(xs, function(x) {
    lines(ths, lx(x, ths))
    points(mle(x), lx(x, mle(x)), pch = 16, col = 'blue')
    points(mle(x), axy, pch = 16, col = 'blue')
  }
  )
  axis(1, pos = axy)
  if(!is.null(theta_true)) abline(v = theta_true, col = 'red')
  if(!is.null(theta_0)) abline(v = theta_0, col = Green)
}
```

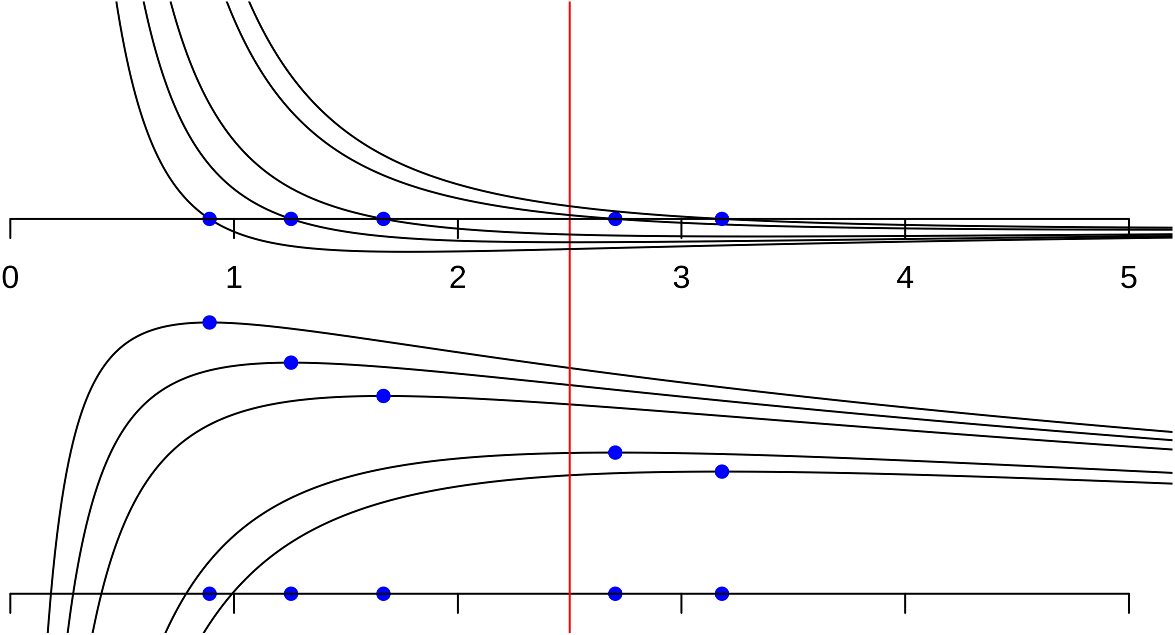
```r
set.seed(765)
theta_true <- 2.5
xs <- lapply(rep(3,5), rexp, 1/theta_true )
xs
```

```
## [[1]]
## [1] 2.7993574 1.2340332 0.9703138
##
## [[2]]
## [1] 3.271532 1.448859 3.392492
##
## [[3]]
## [1] 1.7990311 0.2985643 0.5738256
##
## [[4]]
## [1] 5.4837032 3.4688220 0.5902387
##
## [[5]]
## [1] 0.4789819 1.2406494 2.0448230
```

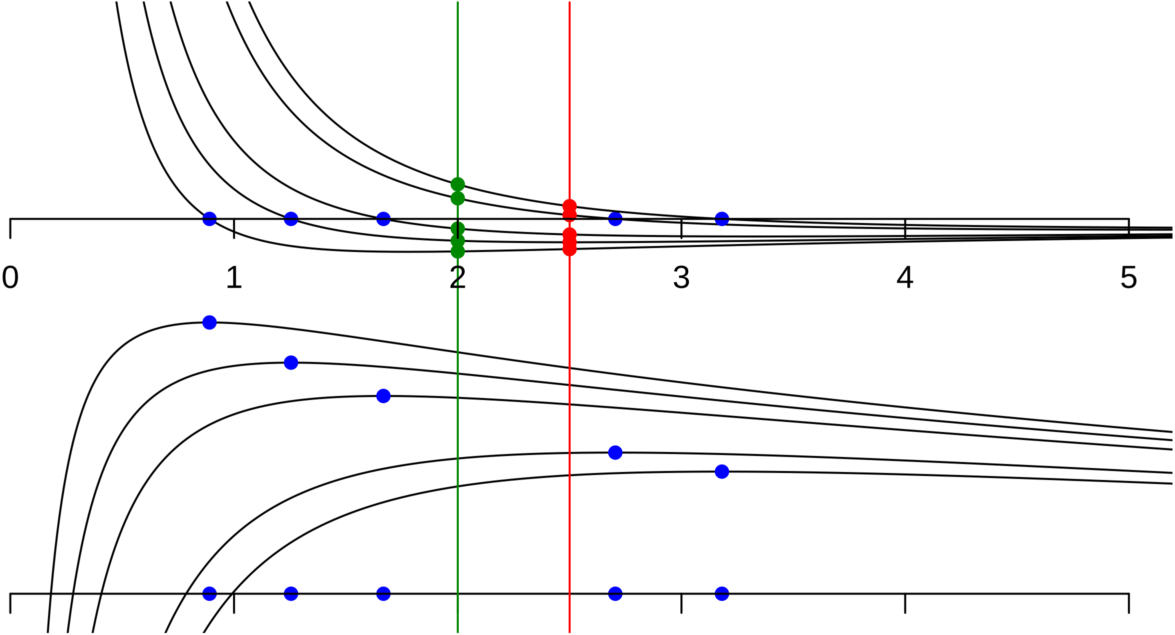
```r
plot_lx(xs, 2.5)
```



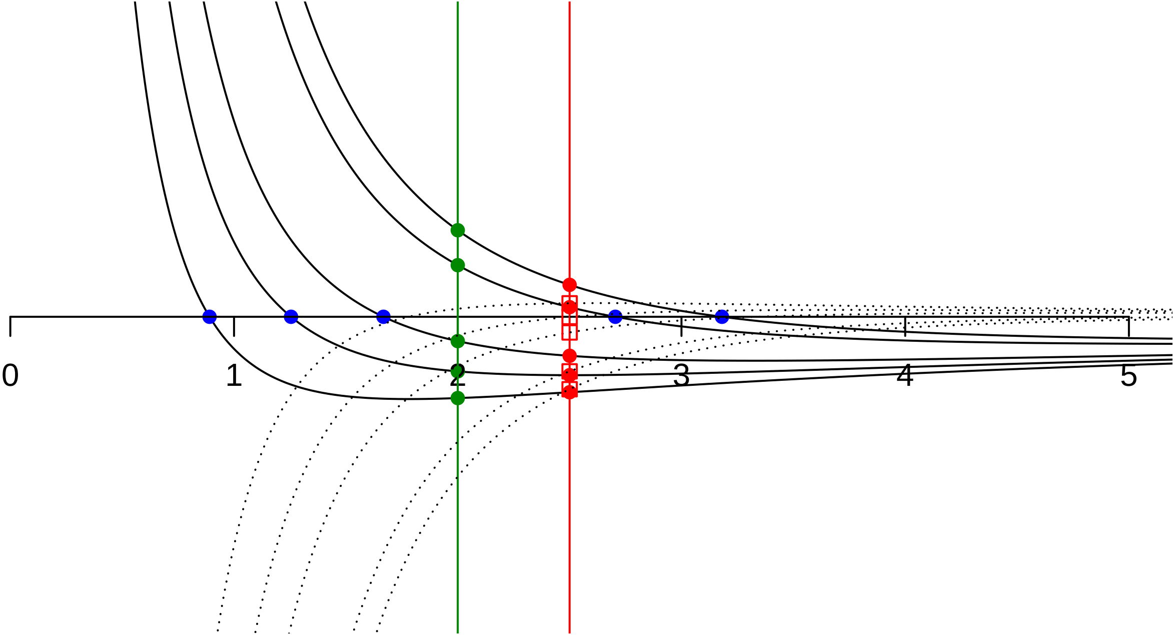
```r
plot_lx. <- function(xs, theta_true = NULL, theta_0 = NULL, ...){
  init()
  plot_lx(xs, theta_true = theta_true, theta_0 = theta_0, ...)
  lapply(xs, function(x) {
    lines(ths, lx.(x,ths))
    points(mle(x), 0, pch = 16, col = 'blue')
  })
  axis(1, pos = 0)
}

plot_lx.(xs, theta_true = 2.5)
```


0
1
2
3
4
5

```r
plot_lx.p <- function(xs, theta_true = NULL, theta_0 = NULL, ...){
  init()
  plot_lx(xs, theta_true = theta_true, theta_0 = theta_0, ...)
  lapply(xs, function(x) {
    lines(ths, lx.(x,ths))
    points(mle(x), 0, pch = 16, col = 'blue')
    if(!is.null(theta_true)) points(theta_true, lx.(x, theta_true), pch = 16, col = 'red
    if(!is.null(theta_0)) points(theta_0, lx.(x, theta_0), pch = 16, col = Green)
  })
  axis(1, pos = 0)
}
plot_lx.p(xs, theta_true = 2.5, theta_0 = 2)
```


0
1
2
3
4
5

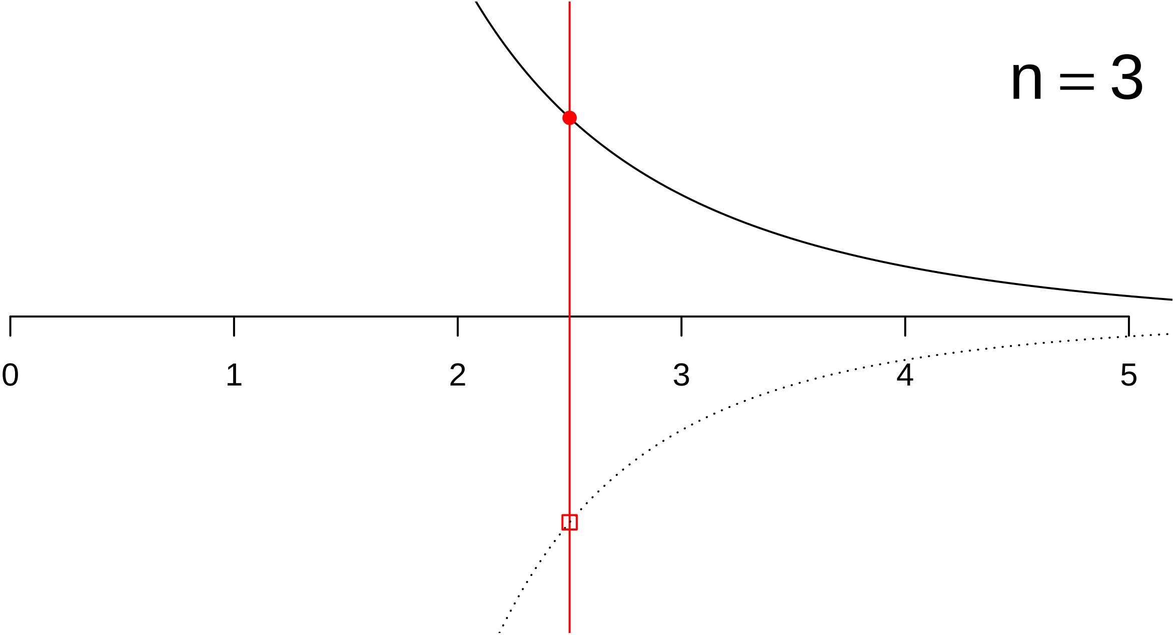
```r
plot_lx.o <- function(xs, theta_true = NULL, theta_0 = NULL, ...){
  init(ylim = c(-3,3))
  lapply(xs, function(x) {
    lines(ths, lx.(x,ths))
    points(mle(x), 0, pch = 16, col = 'blue')
    if(!is.null(theta_true)) points(theta_true, lx.(x, theta_true), pch = 16, col = 'red
    if(!is.null(theta_0)) points(theta_0, lx.(x, theta_0), pch = 16, col = Green)
  })
  axis(1, pos = 0)
  if(!is.null(theta_true)) abline(v = theta_true, col = 'red')
  if(!is.null(theta_0)) abline(v = theta_0, col = Green)
  lapply(xs, function(x) {
    lines(ths, lx..(x,ths), lty = 3)

    if(!is.null(theta_true)) points(theta_true, lx..(x, theta_true), pch = 0, col = 'red
    # if(!is.null(theta_0)) points(theta_0, lx.(x, theta_0), pch = 16, col = Green)
  })
}
plot_lx.o(xs, theta_true = 2.5, theta_0 = 2)
```

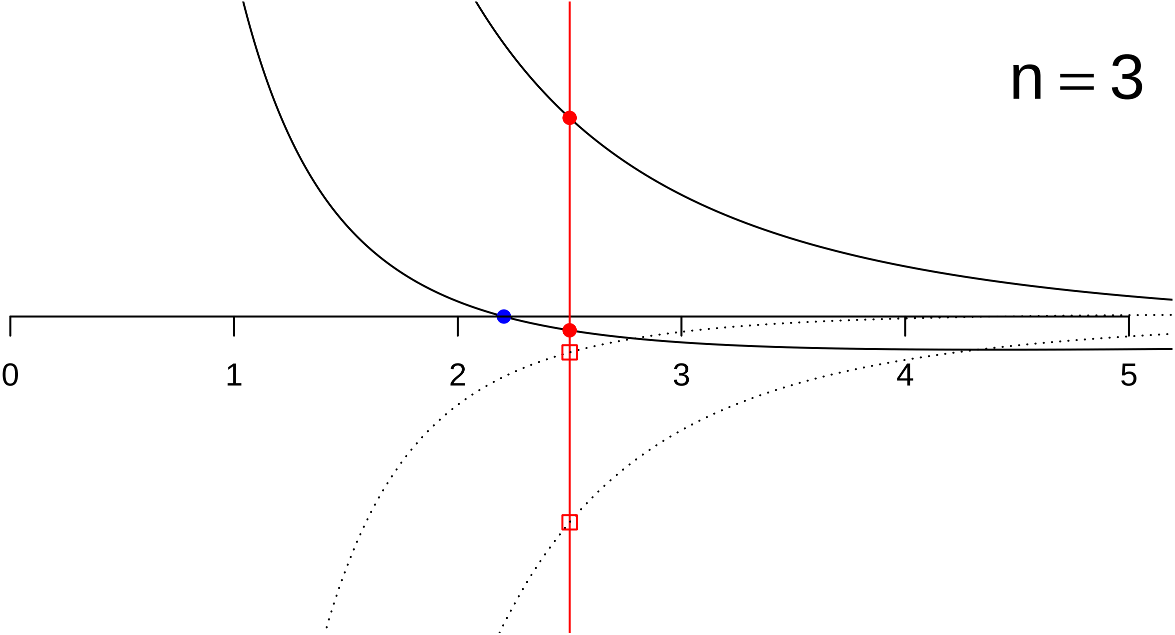

0
1
2
3
4
5

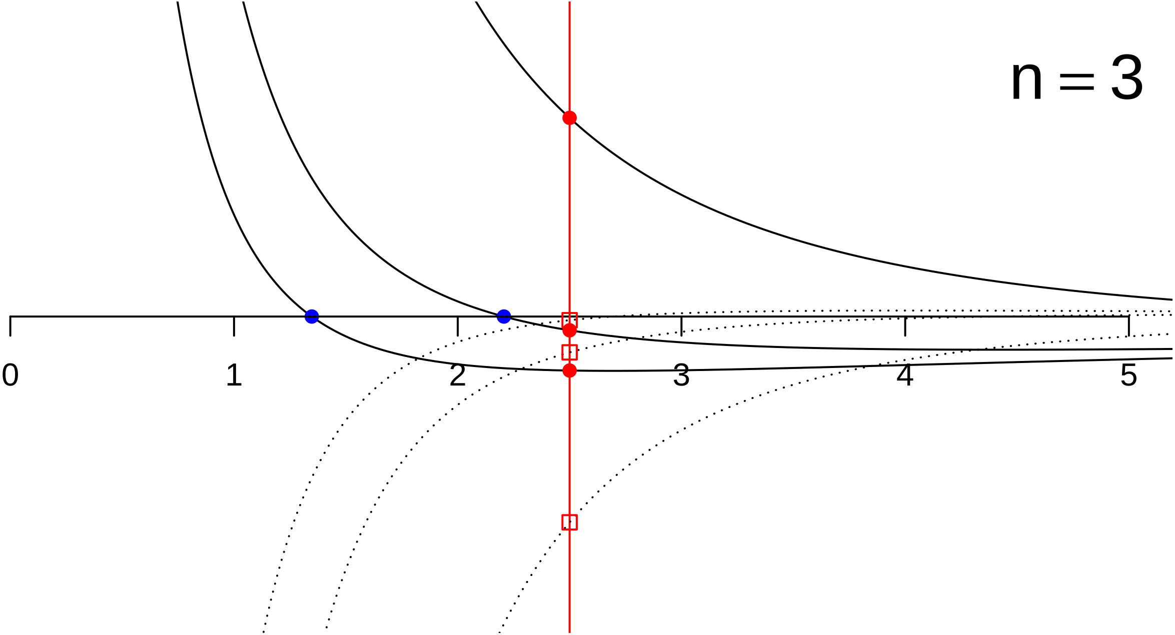
```
## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
##
## [[4]]
## NULL
##
## [[5]]
## NULL
```

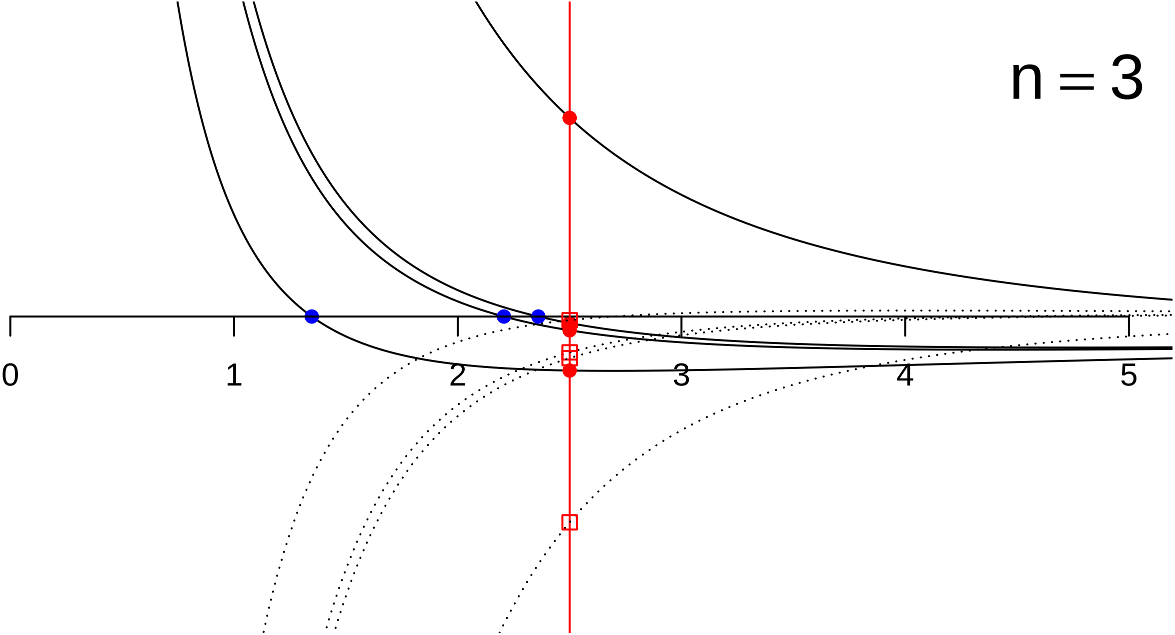
```r
# N = 3

N <- 3

xs <- lapply(rep(N,10), rexp, 1/theta_true)
```

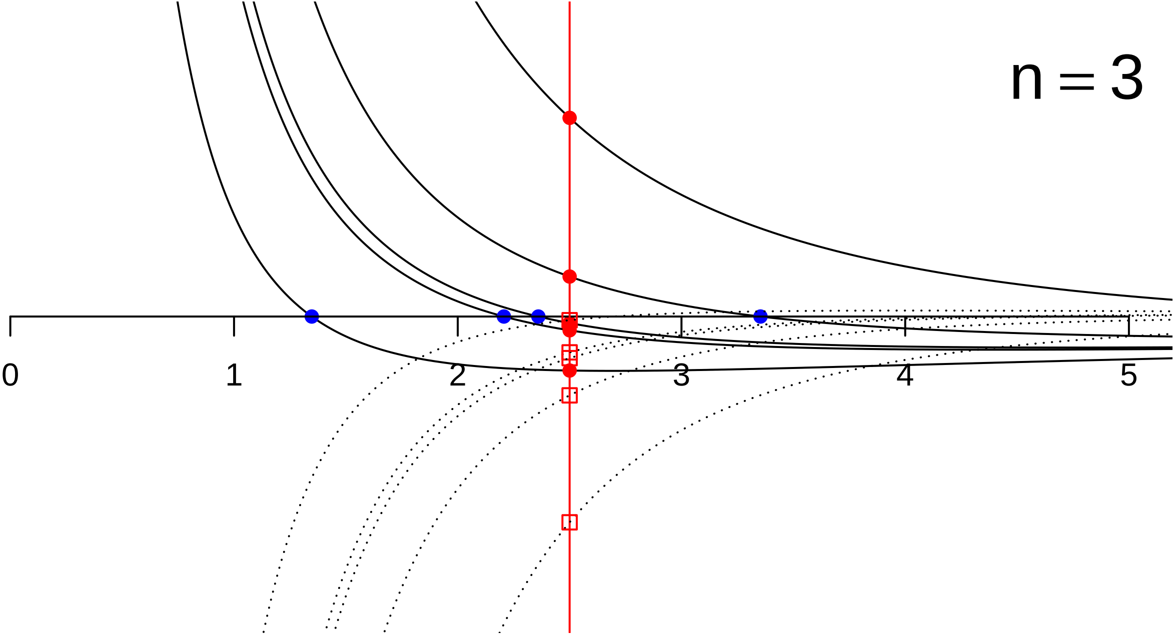
```r
for(n in 1:10){
plot_lx.o(xs[1:n], theta_true = 2.5)
  legend('topright', as.expression(bquote(n == .(N))) , cex = 2, bty = 'n')
  pause()
}
```

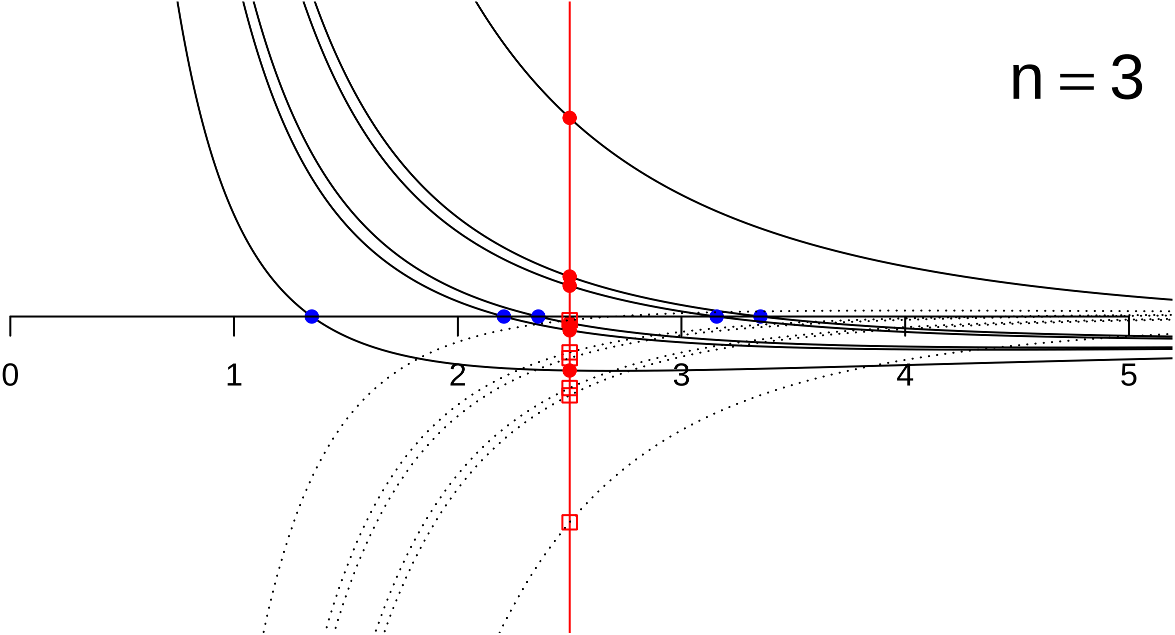

n = 3
0
1
2
3
4
5

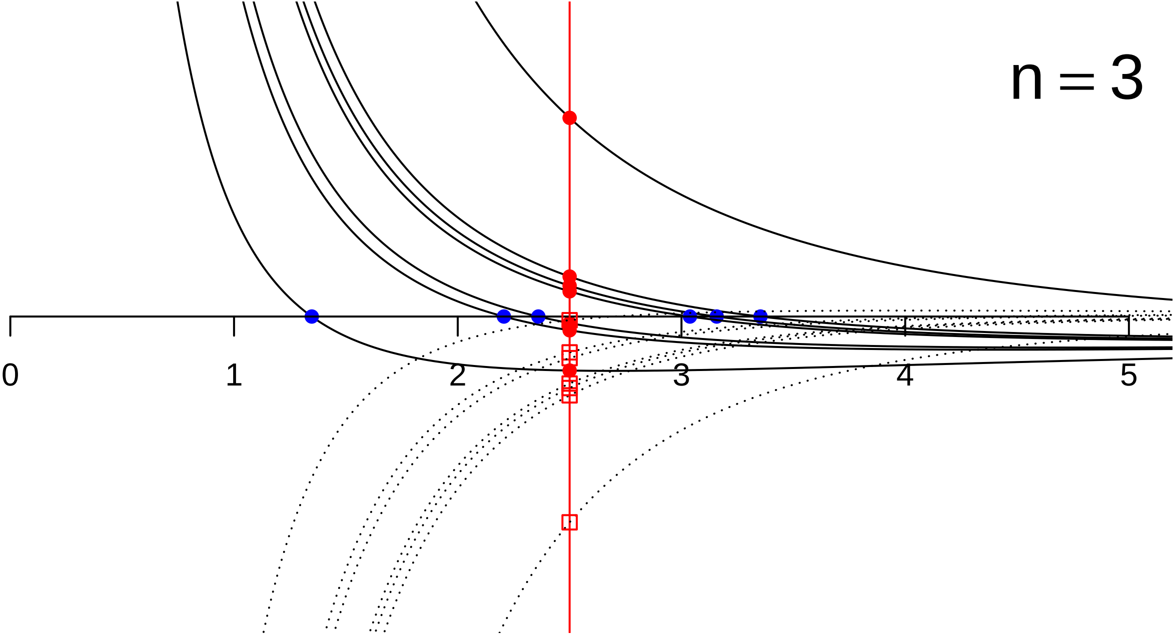

n = 3
0
1
2
3
4
5

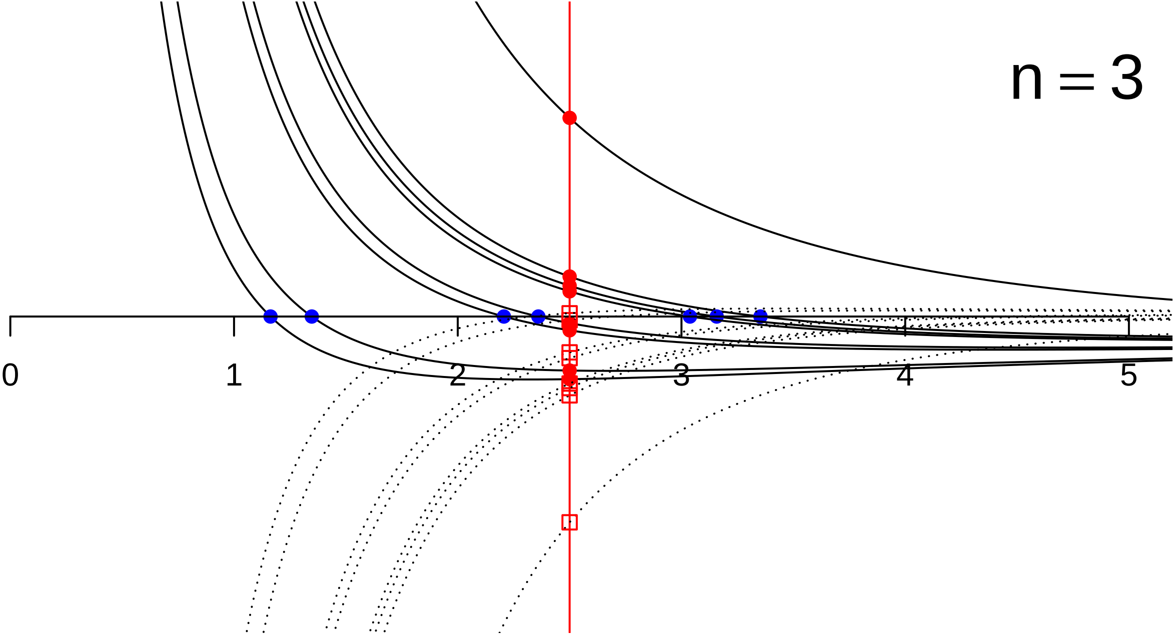

n = 3
0
1
2
3
4
5

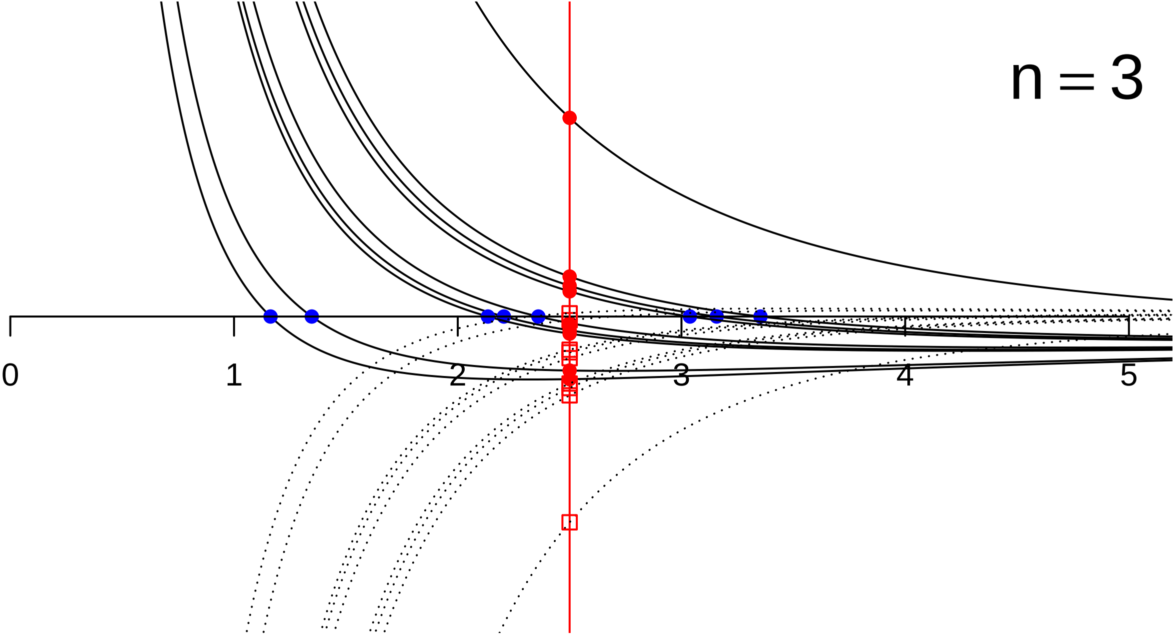

n = 3
0
1
2
3
4
5

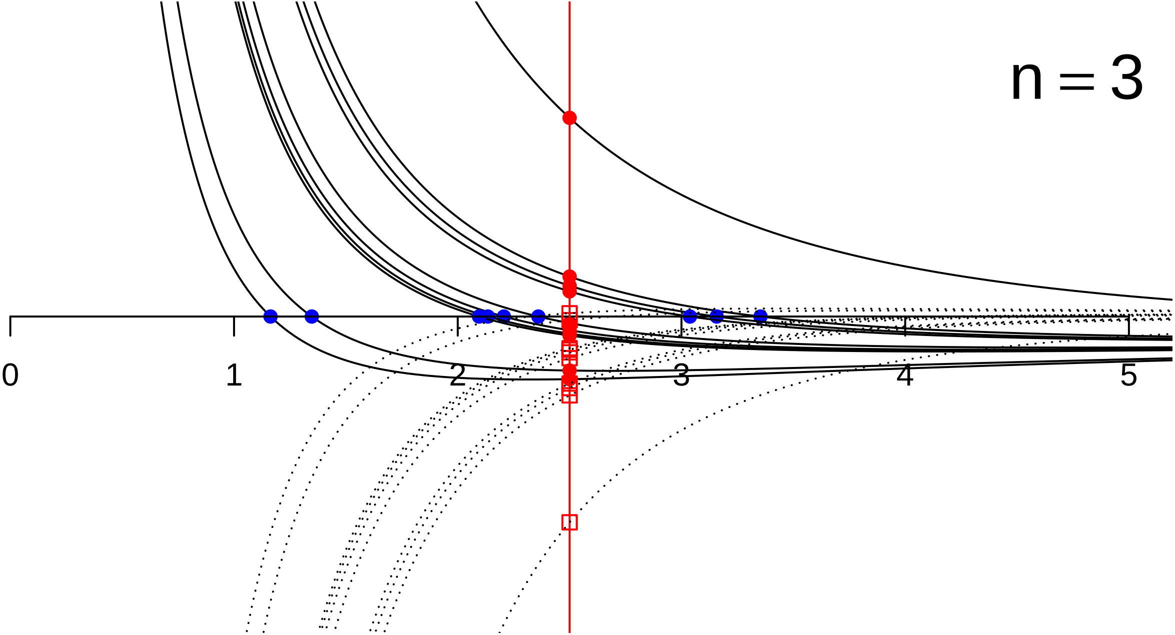

n = 3
0
1
2
3
4
5


n = 3
0
1
2
3
4
5

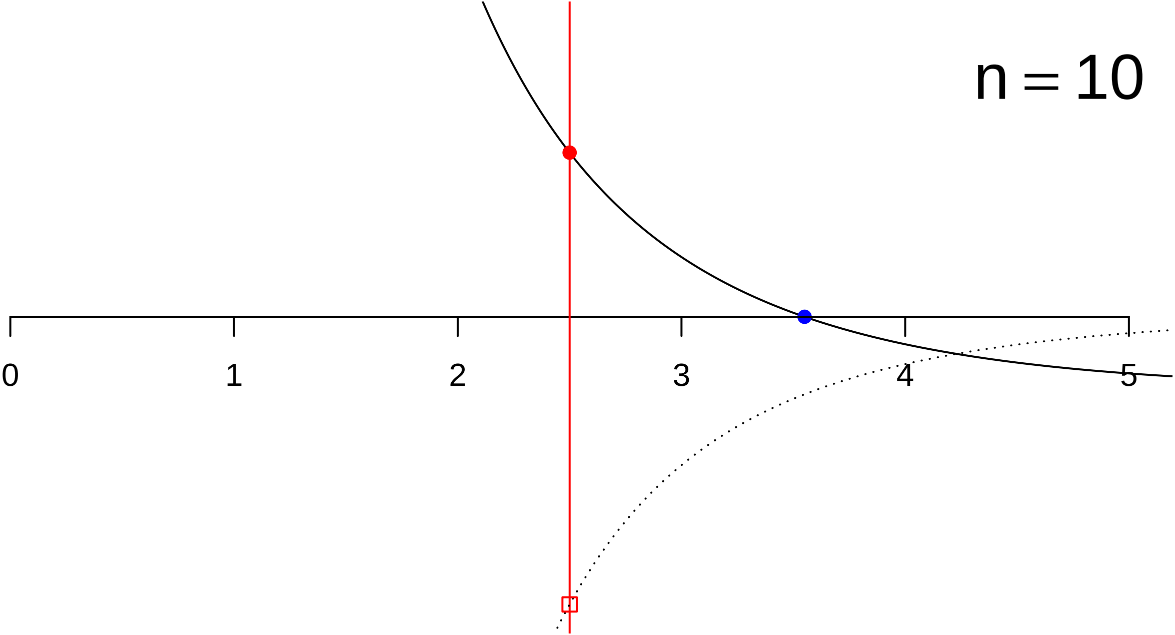

n = 3
0
1
2
3
4
5

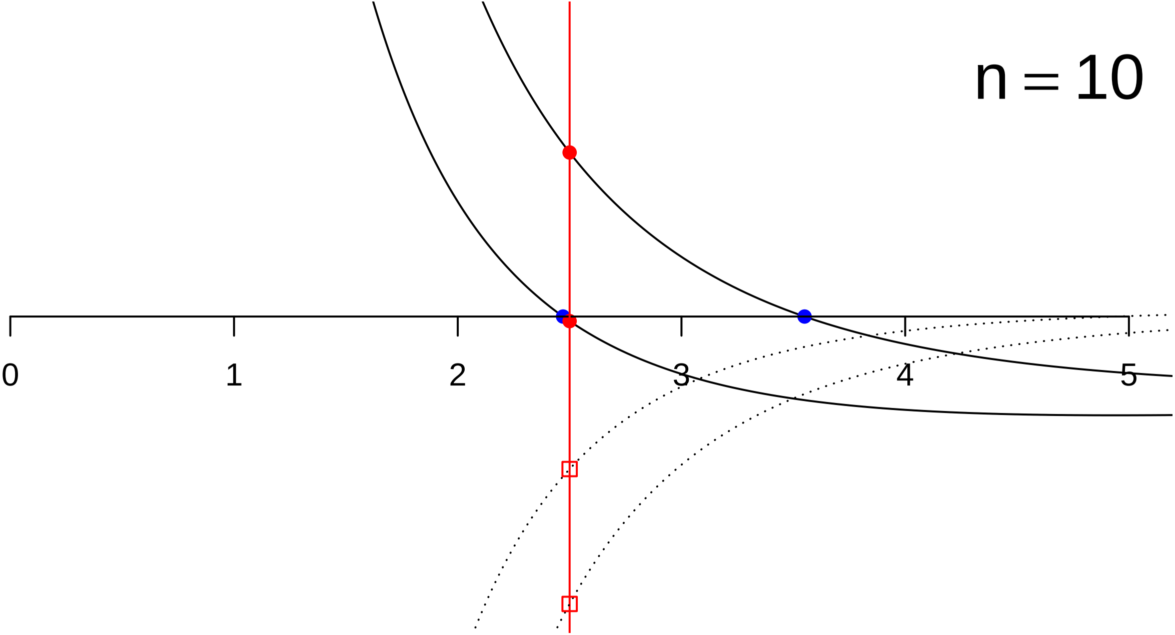

n = 3
0
1
2
3
4
5

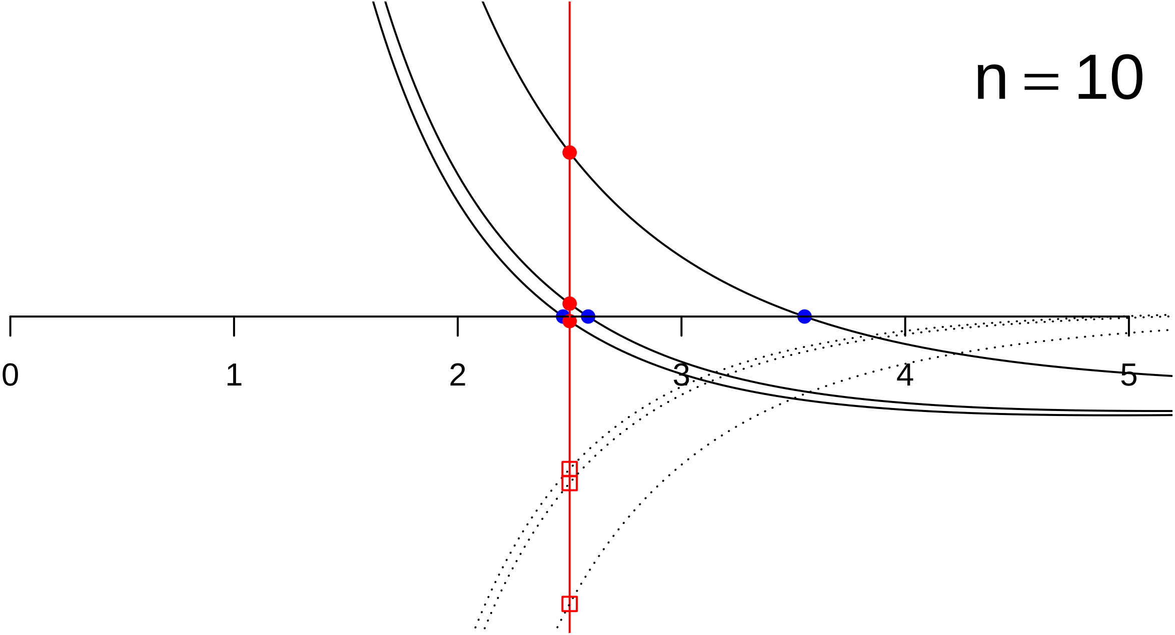

n = 3
0
1
2
3
4
5

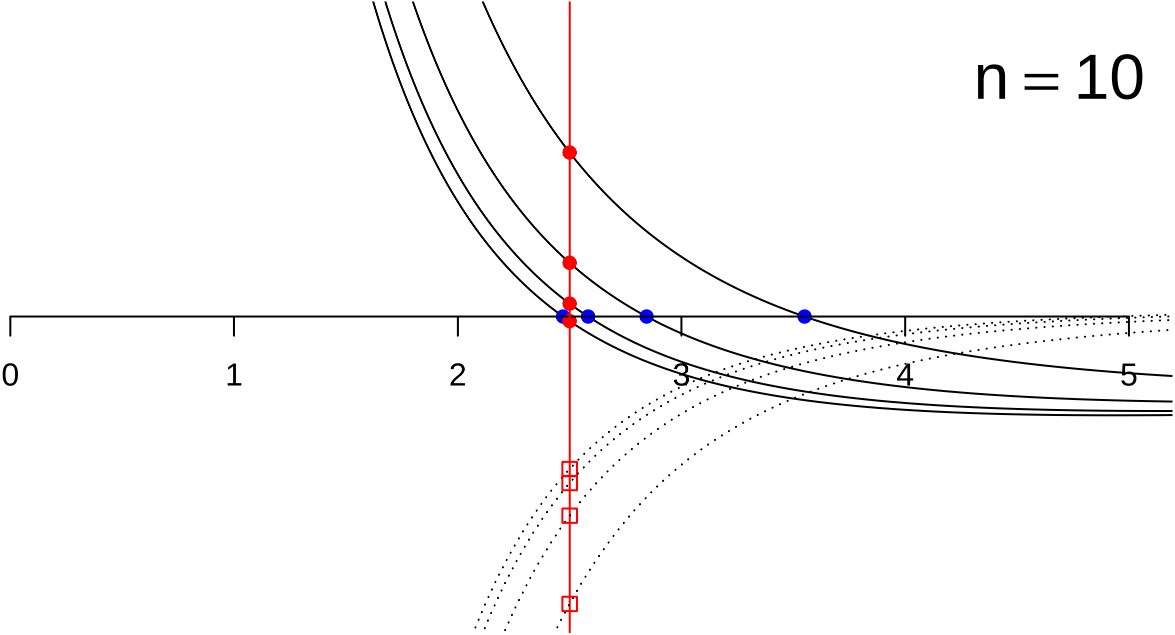

n = 3
0
1
2
3
4
5

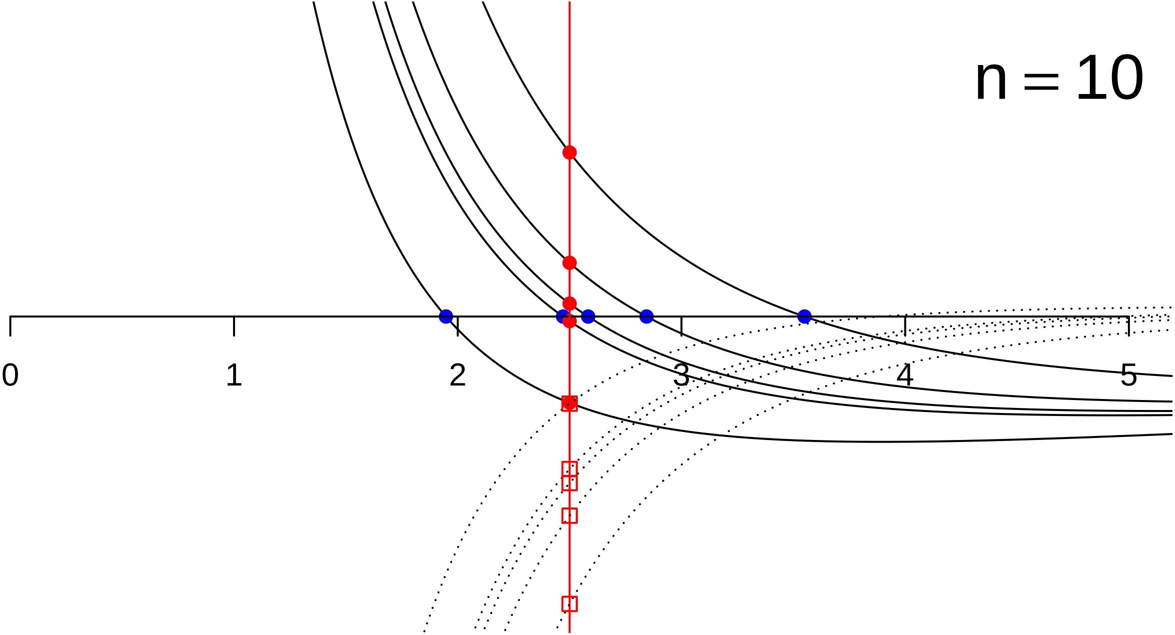
```
# N = 10

N <- 10

xs <- lapply(rep(N,10), rexp, 1/theta_true)

for(n in 1:10){
  plot_lx.o(xs[1:n], theta_true = 2.5)
  legend('topright', as.expression(bquote(n == .(N))) , cex = 2, bty = 'n')
  pause()
}
```

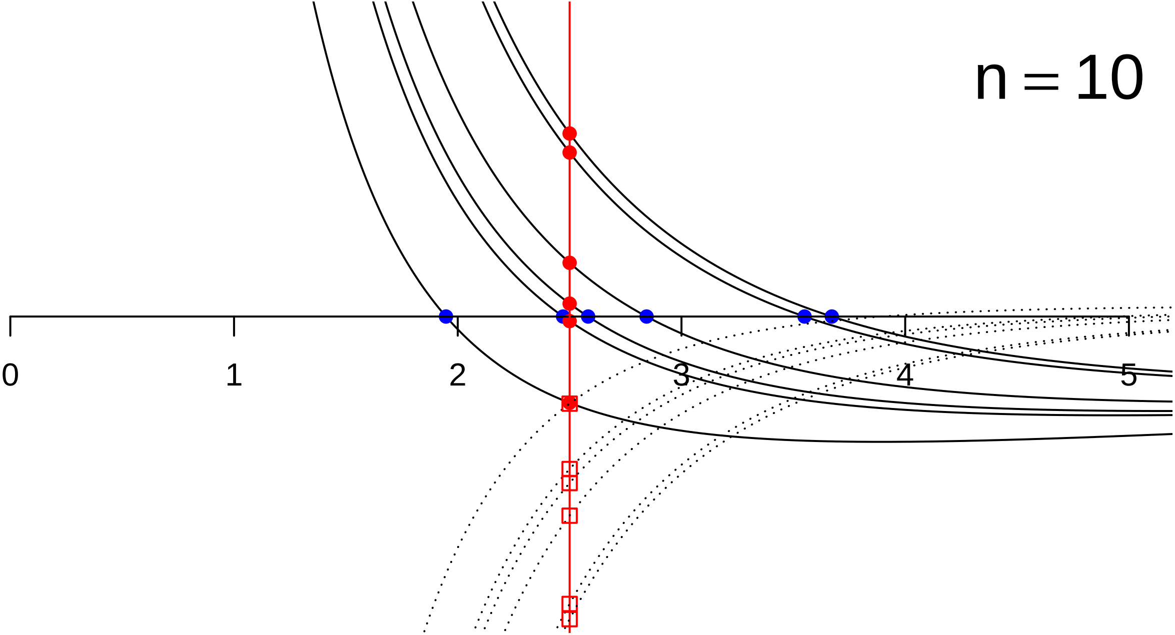

n = 10
0
1
2
3
4
5

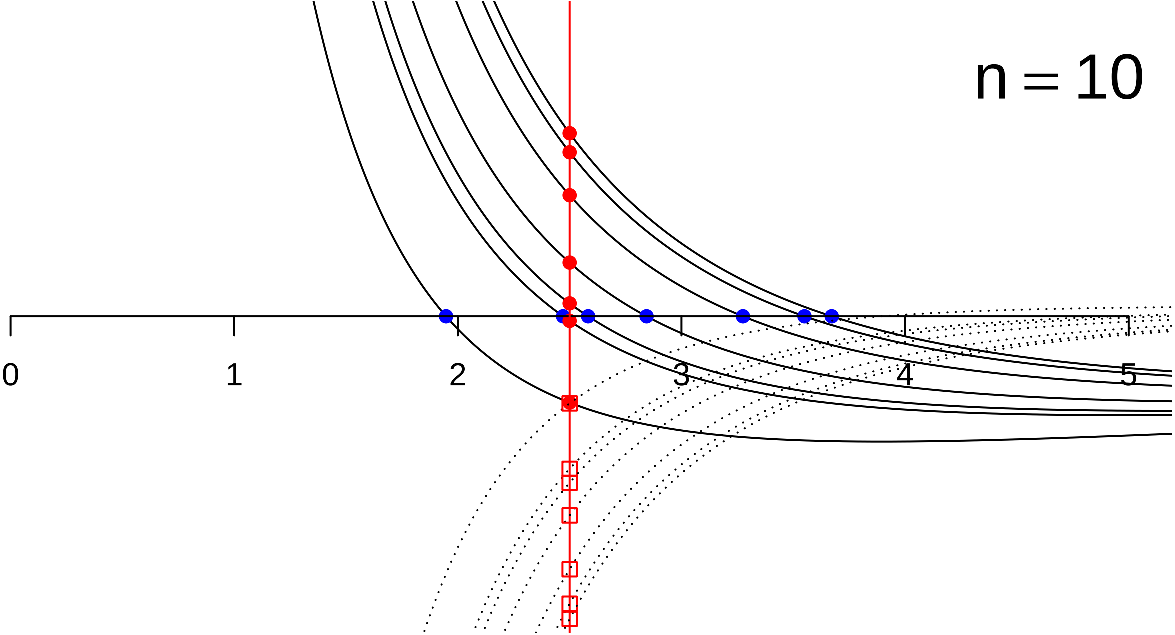

n = 10
0
1
2
3
4
5

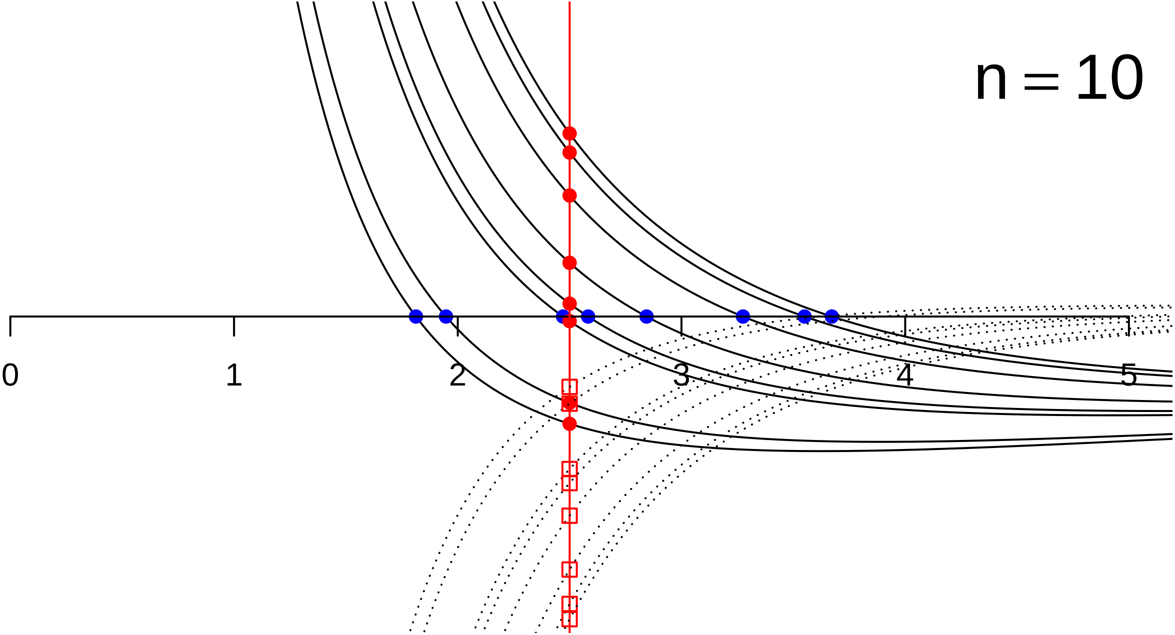

n = 10
0
1
2
3
4
5

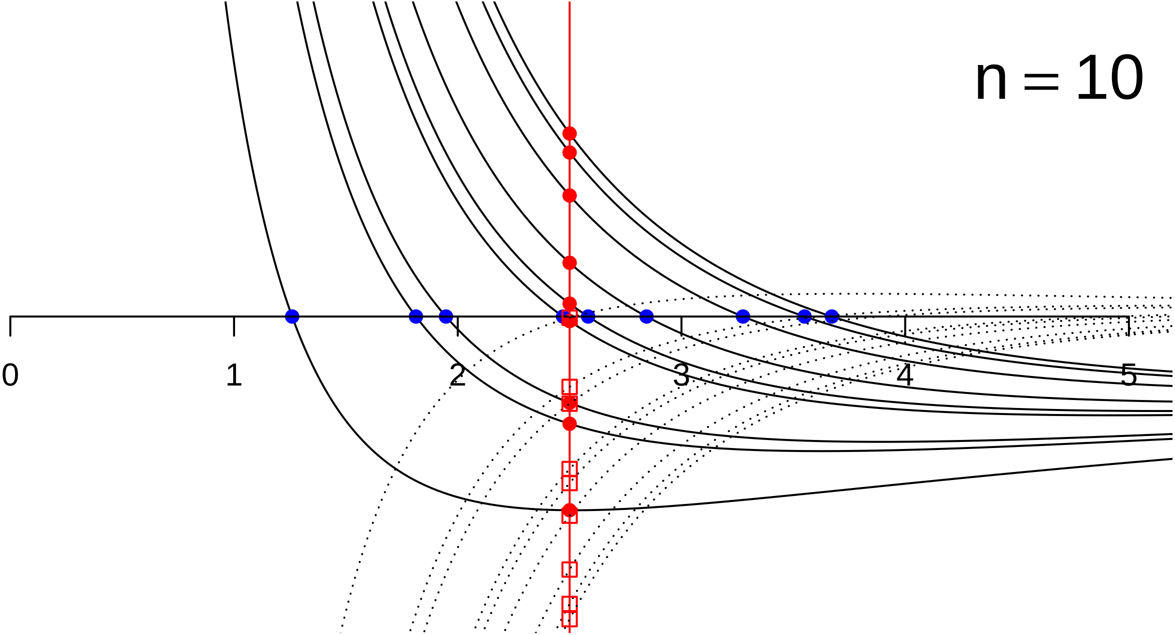

n = 10
0
1
2
3
4
5

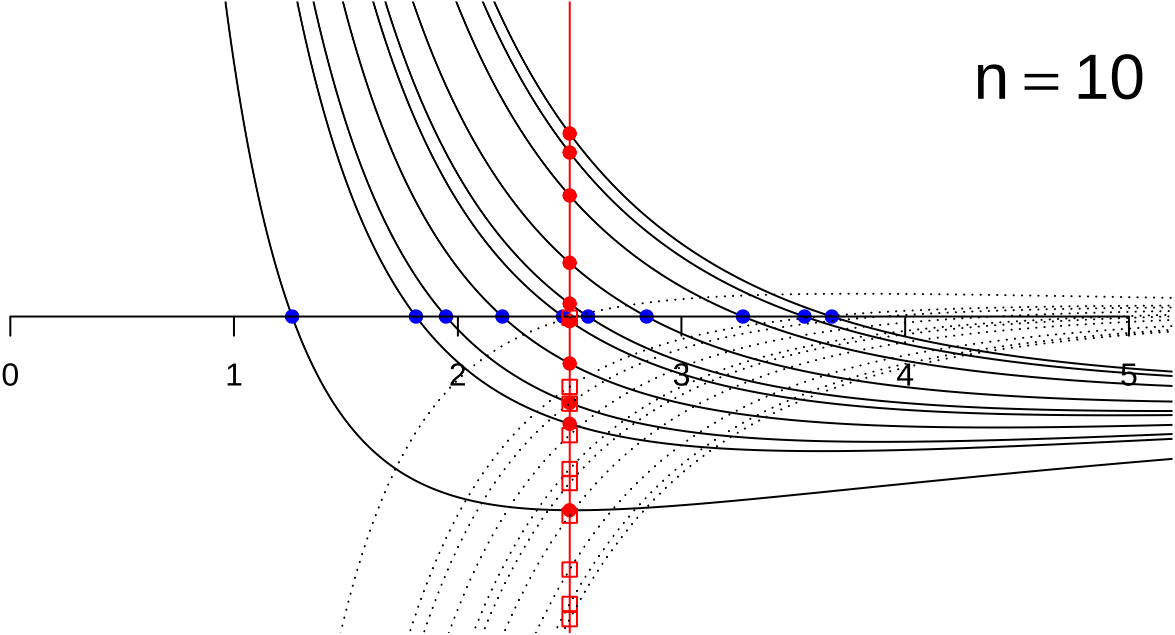

n = 10
0
1
2
3
4
5


n = 10
0
1
2
3
4
5

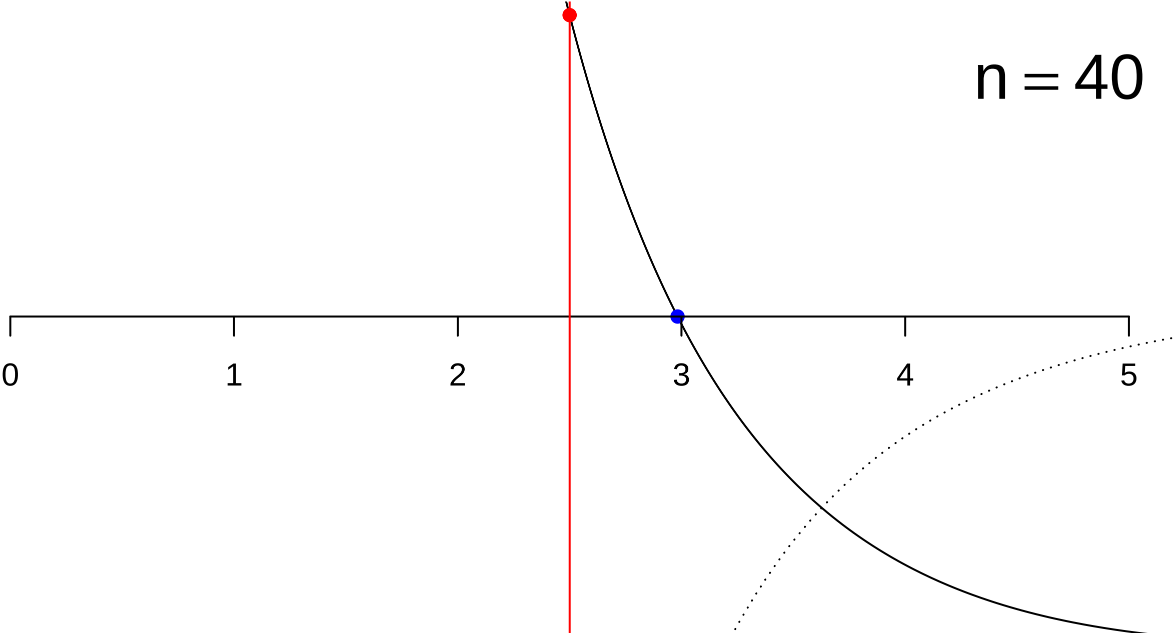

n = 10
0
1
2
3
4
5

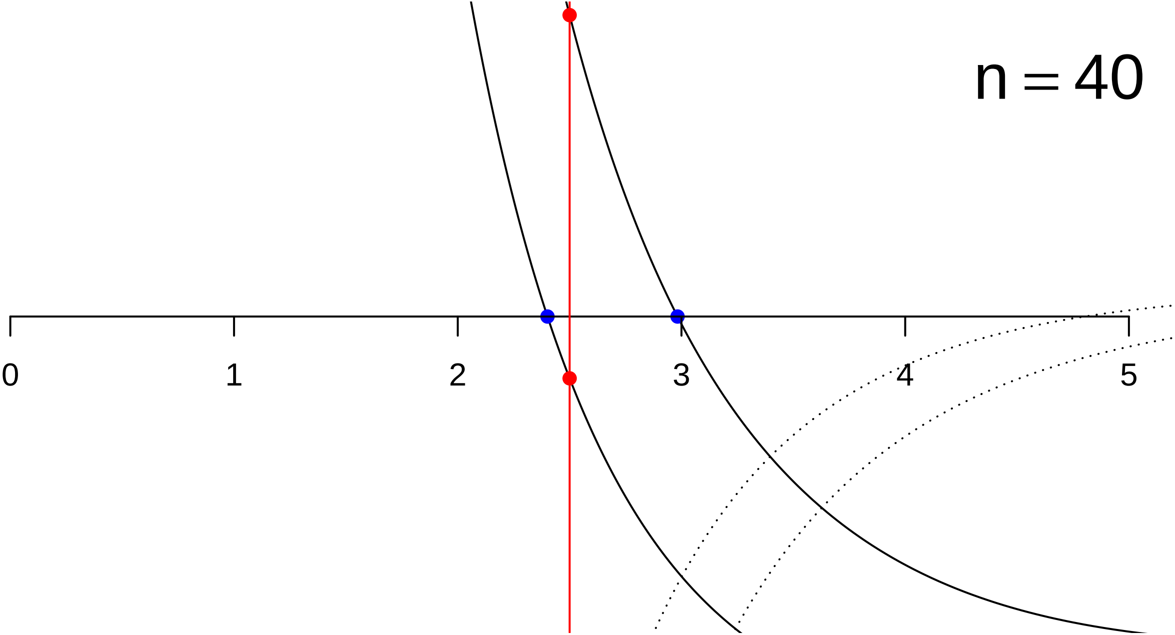

n = 10
0
1
2
3
4
5

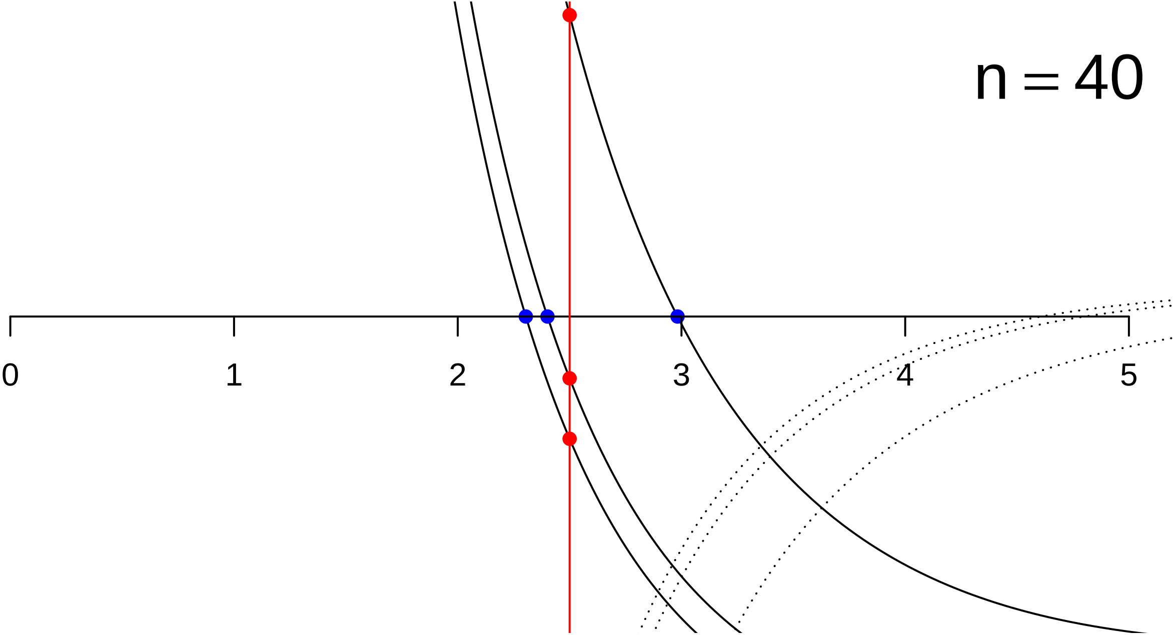

n = 10
0
1
2
3
4
5

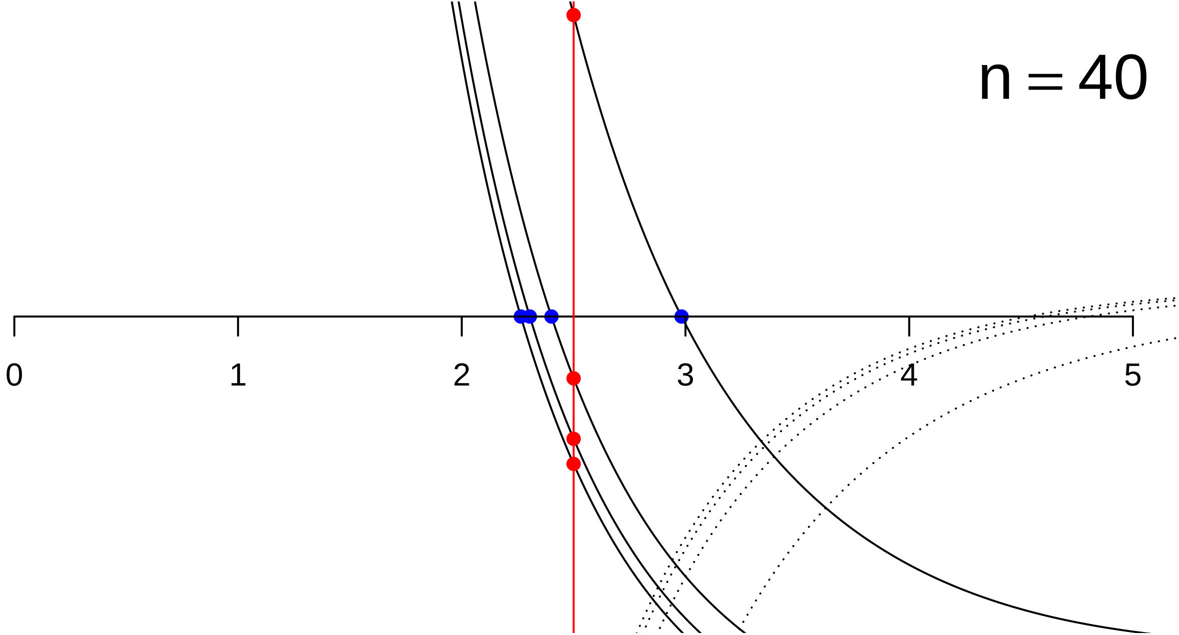

n = 10
0
1
2
3
4
5

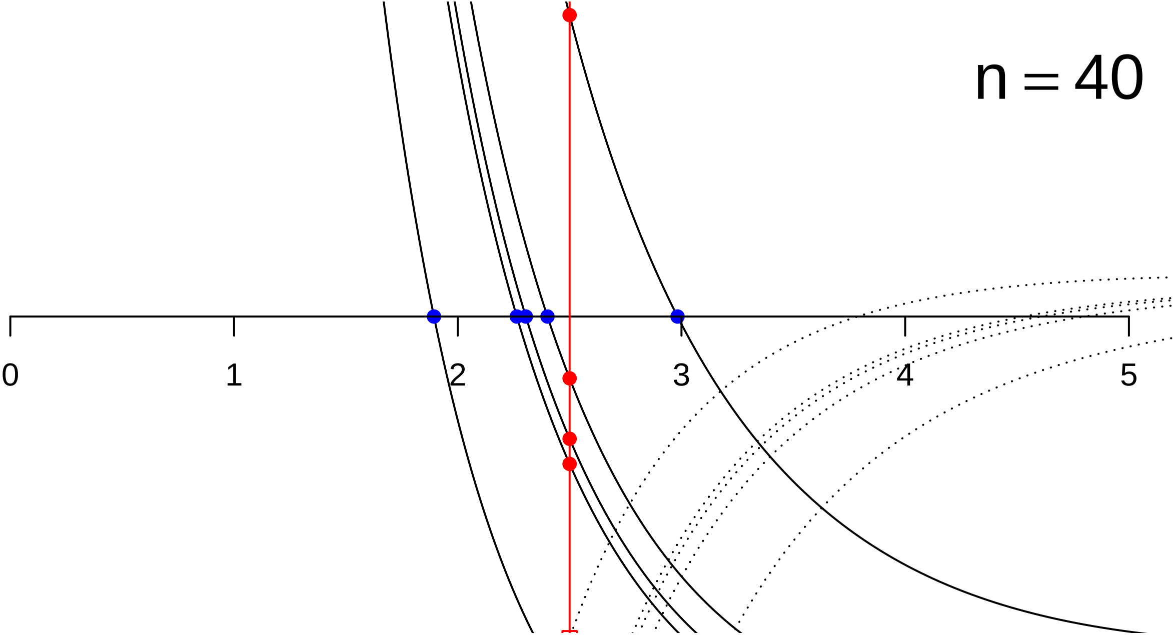
```r
# N = 40

N <- 40

xs <- lapply(rep(N,10), rexp, 1/theta_true)

for(n in 1:10){
  plot_lx.o(xs[1:n], theta_true = 2.5)
  legend('topright', as.expression(bquote(n == .(N))) , cex = 2, bty = 'n')
  pause()
}
```

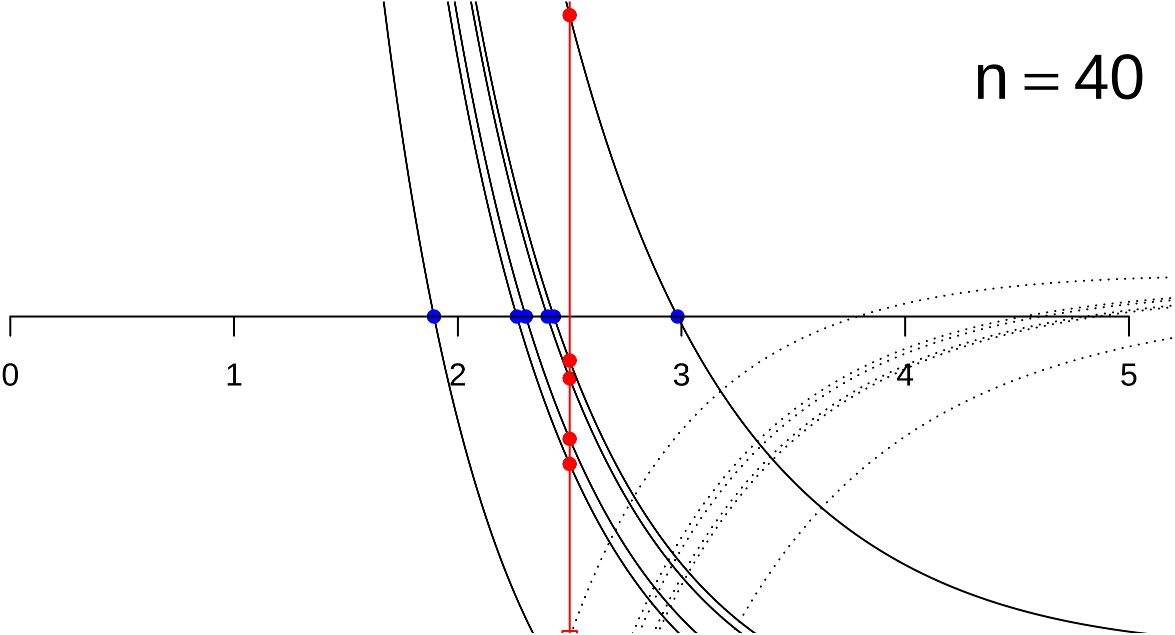

n = 40
0
1
2
3
4
5

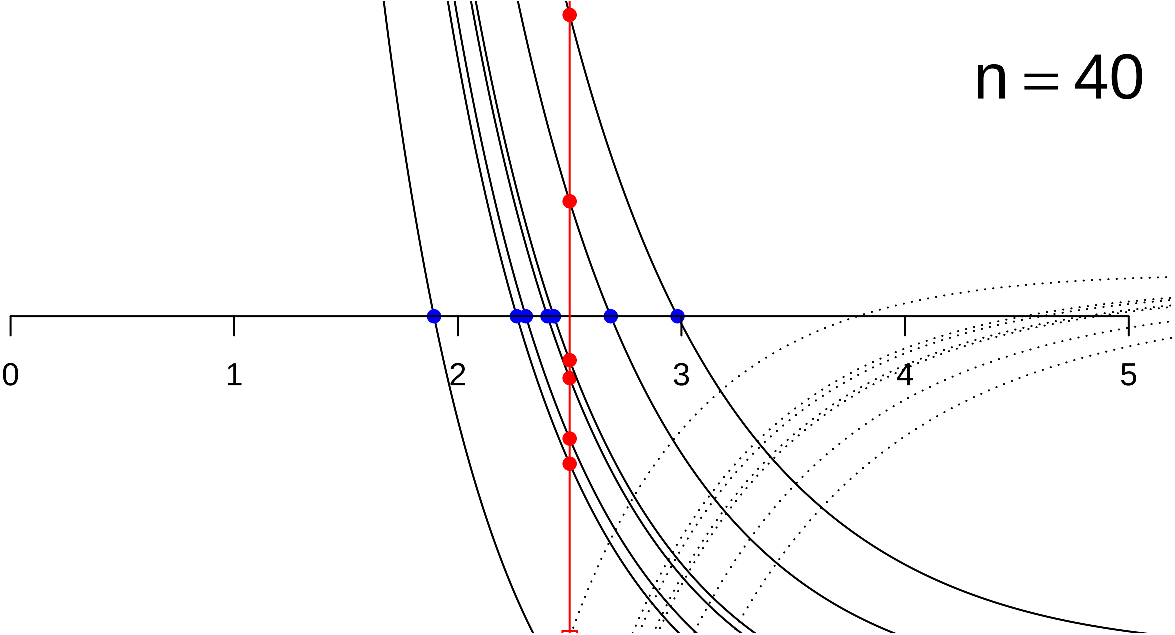

n = 40
0
1
2
3
4
5

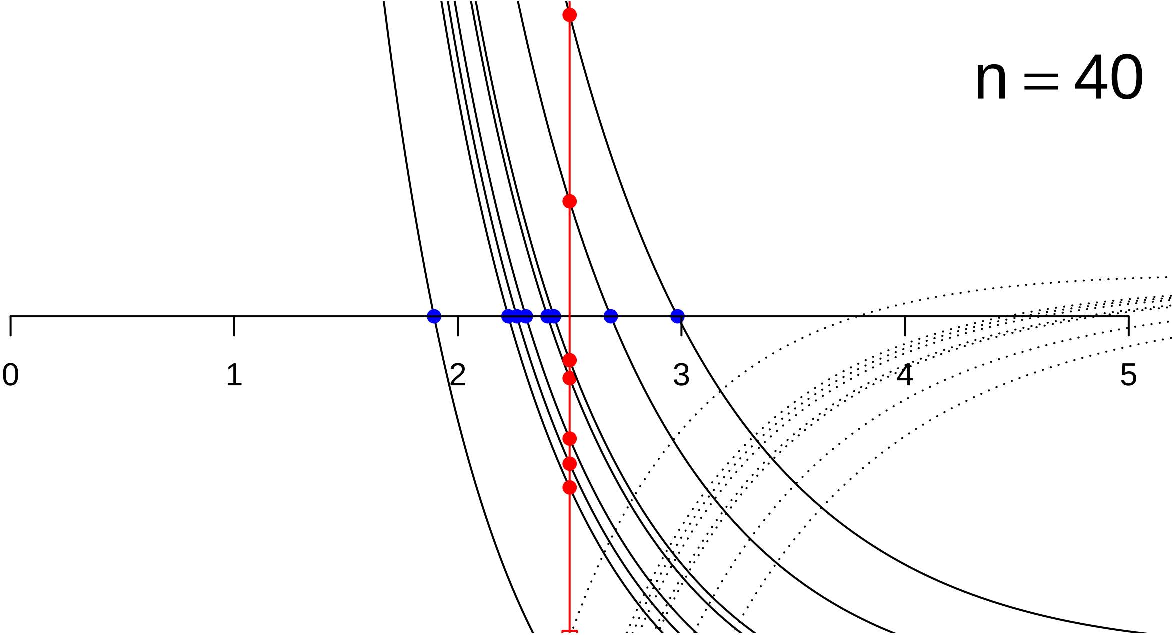

n = 40
0
1
2
3
4
5

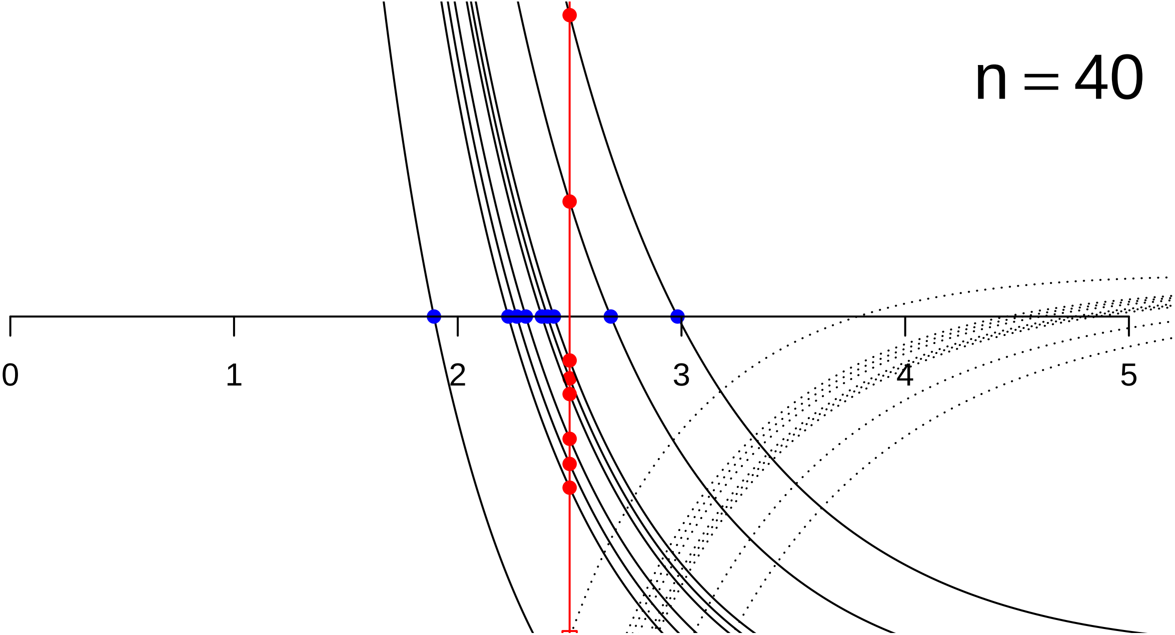

n = 40
0
1
2
3
4
5


n = 40
0
1
2
3
4
5

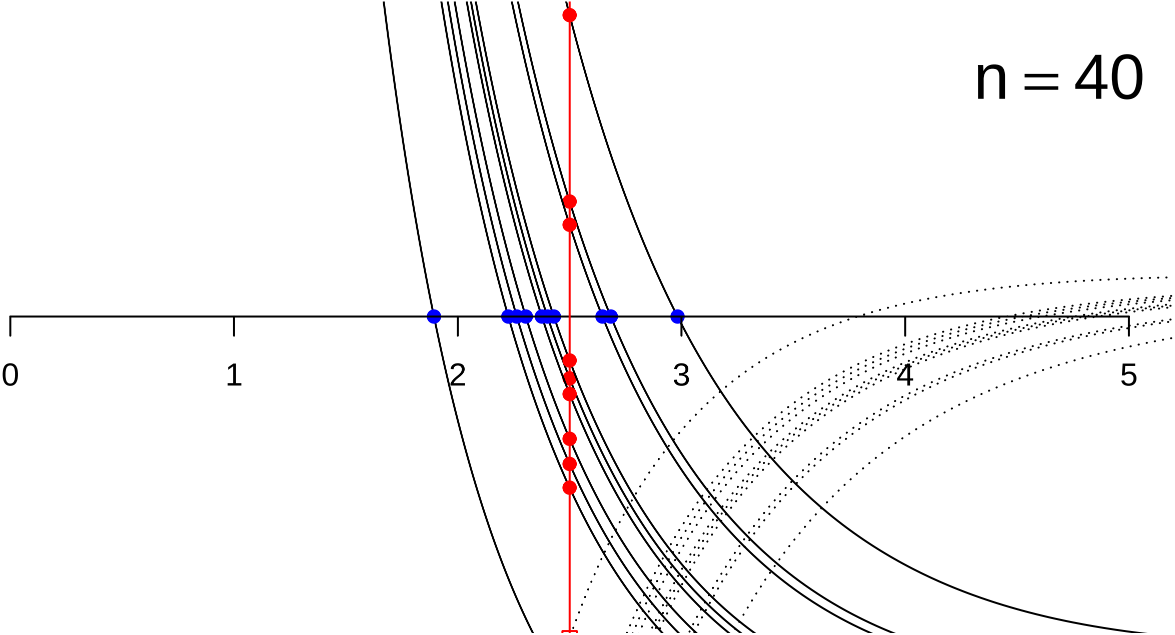

n = 40
0
1
2
3
4
5


n = 40
0
1
2
3
4
5

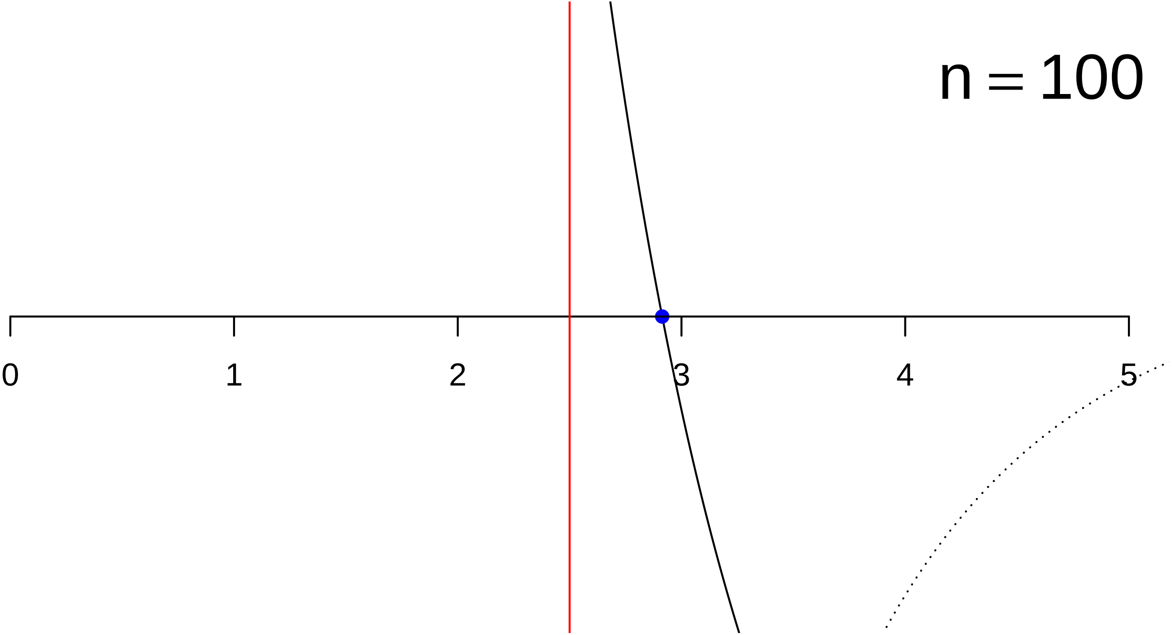

n = 40
0
1
2
3
4
5

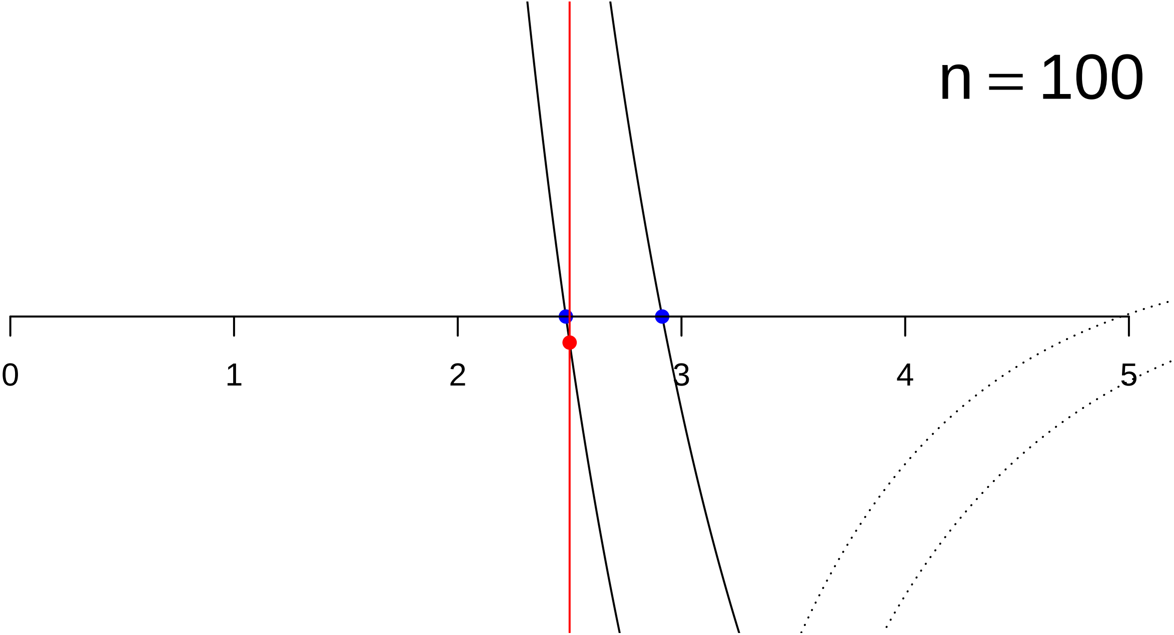


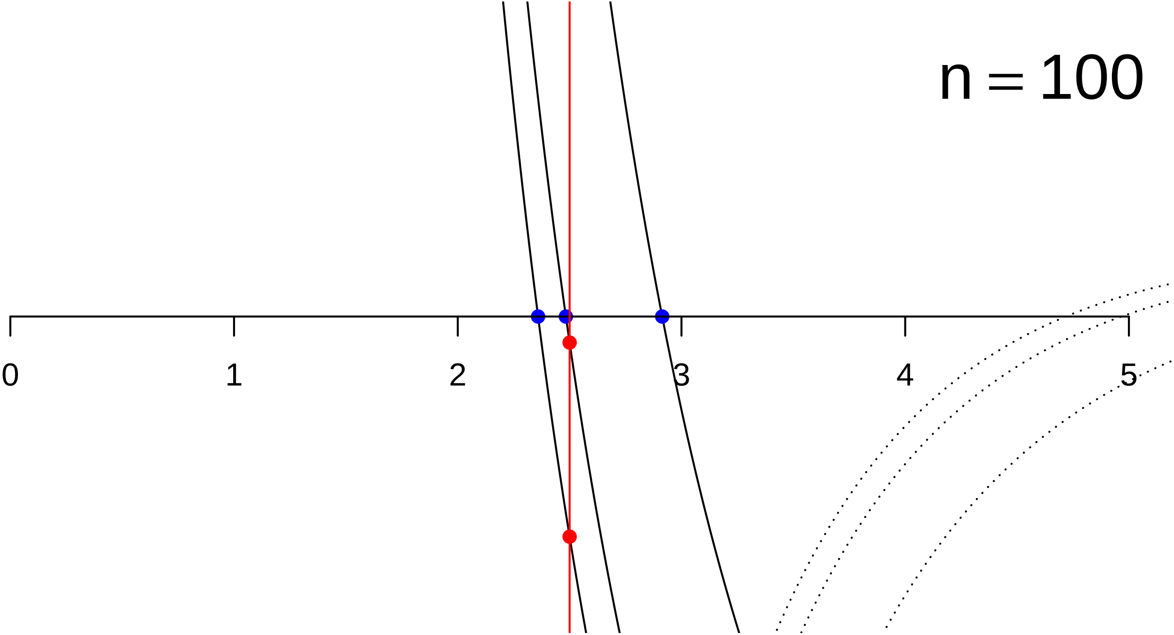
n = 40
0
1
2
3
4
5

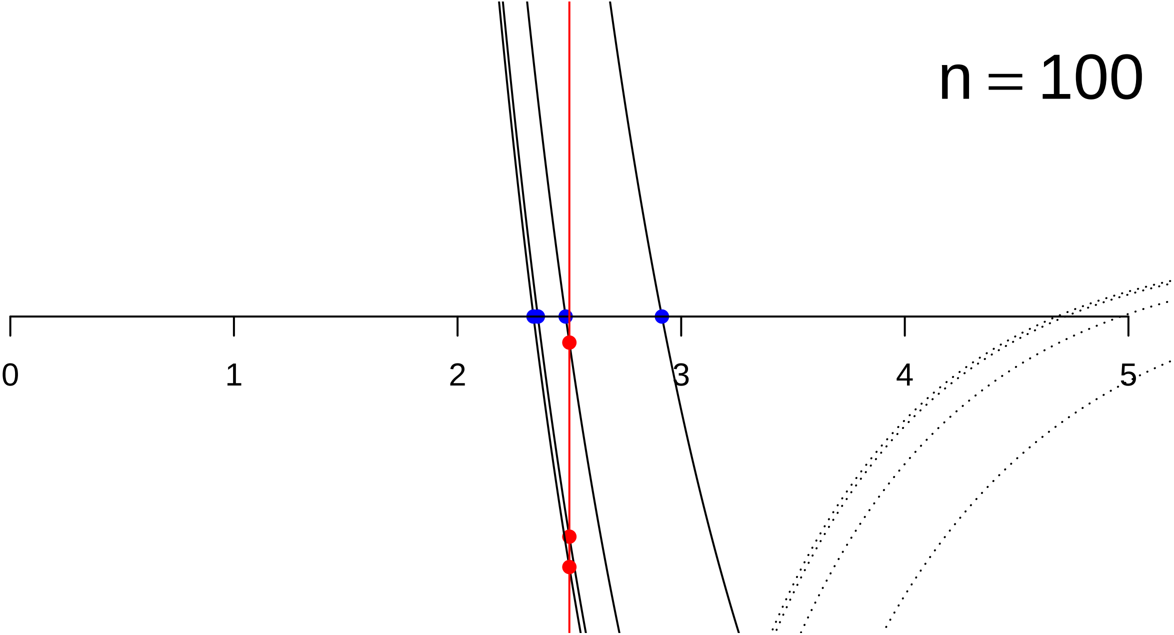

n = 40
0
1
2
3
4
5

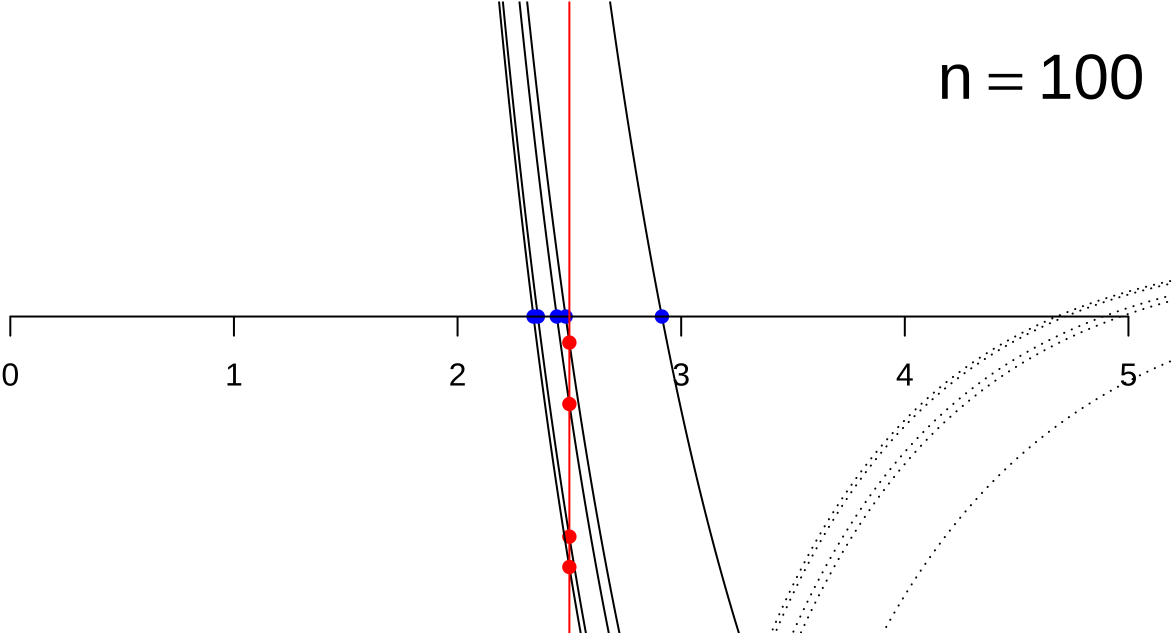
```r
# N = 100

N <- 100

xs <- lapply(rep(N,10), rexp, 1/theta_true)

for(n in 1:10){
  plot_lx.o(xs[1:n], theta_true = 2.5)
  legend('topright', as.expression(bquote(n == .(N))) , cex = 2, bty = 'n')
  pause()
}
```

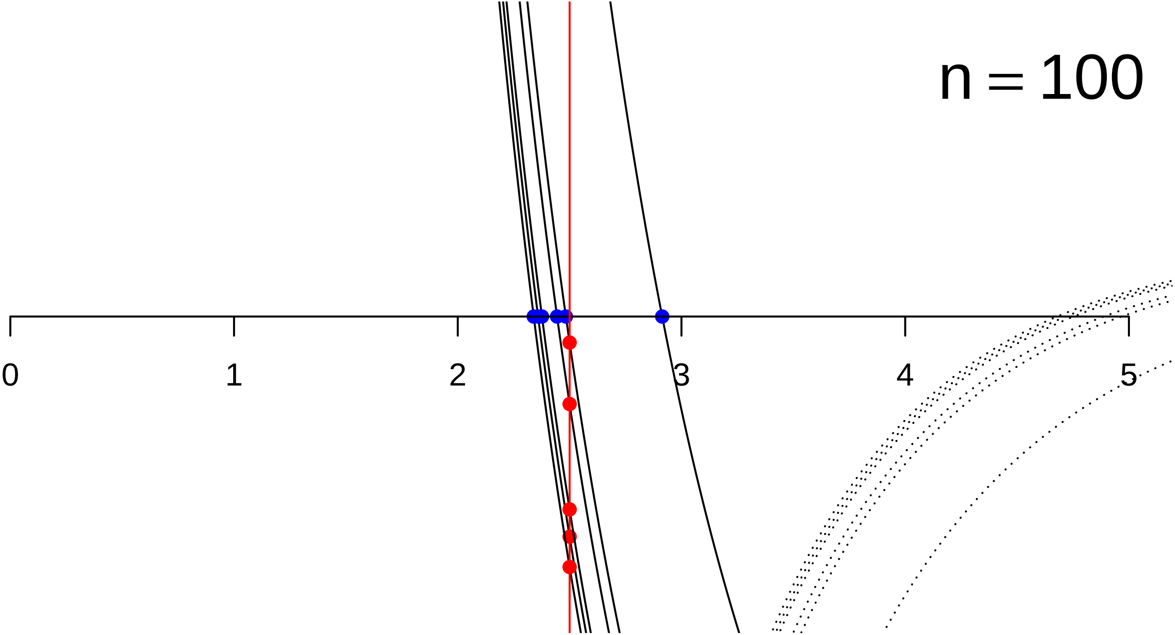

n = 100
0
1
2
3
4
5

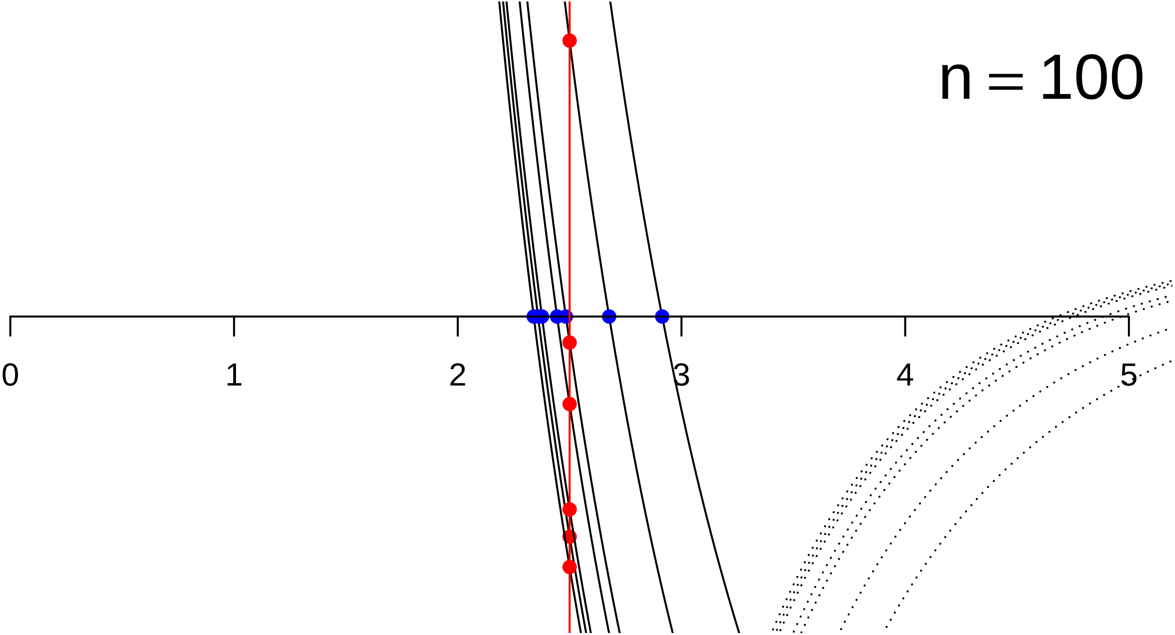

n = 100
0
1
2
3
4
5

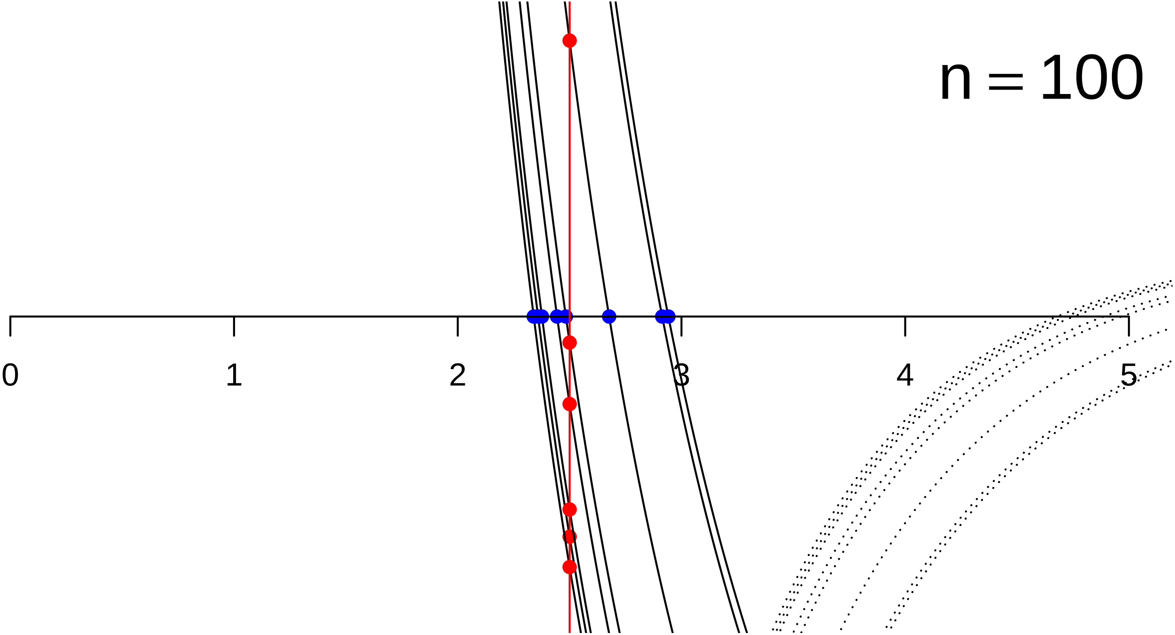

n = 100
0
1
2
3
4
5

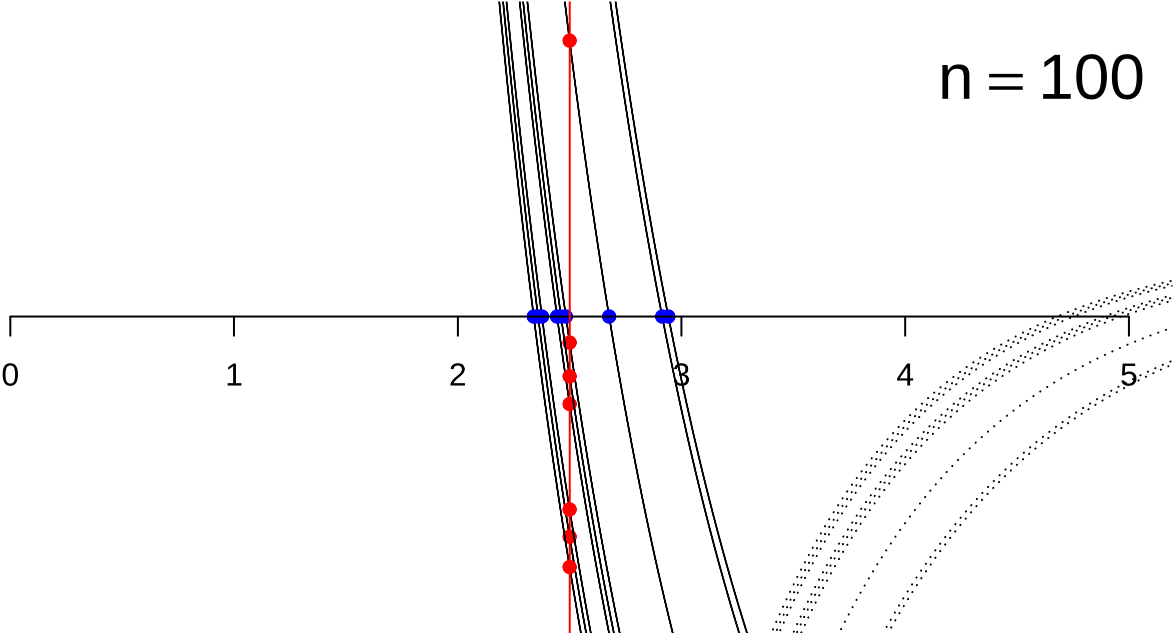

n = 100
0
1
2
3
4
5

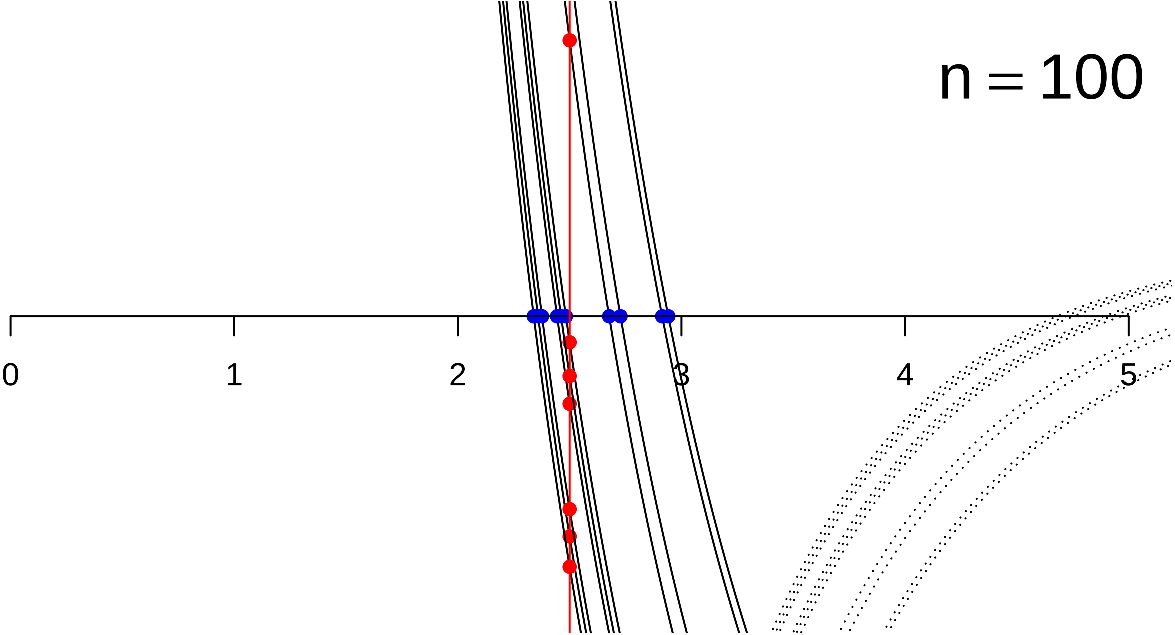

n = 100
0
1
2
3
4
5


n = 100
0
1
2
3
4
5

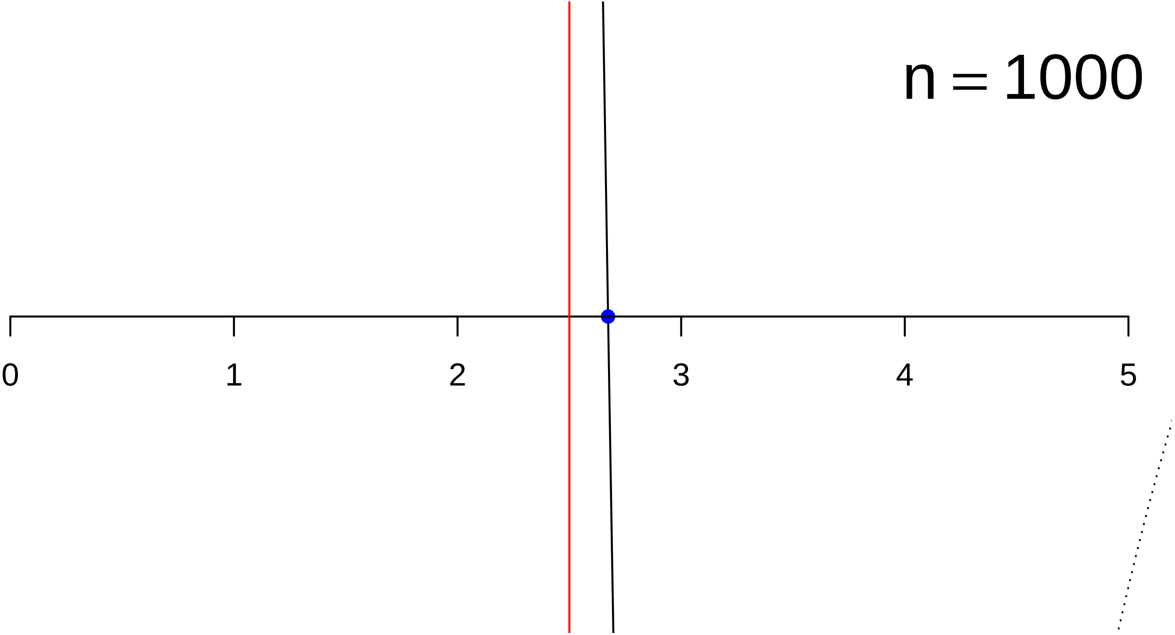

n = 100
0
1
2
3
4
5

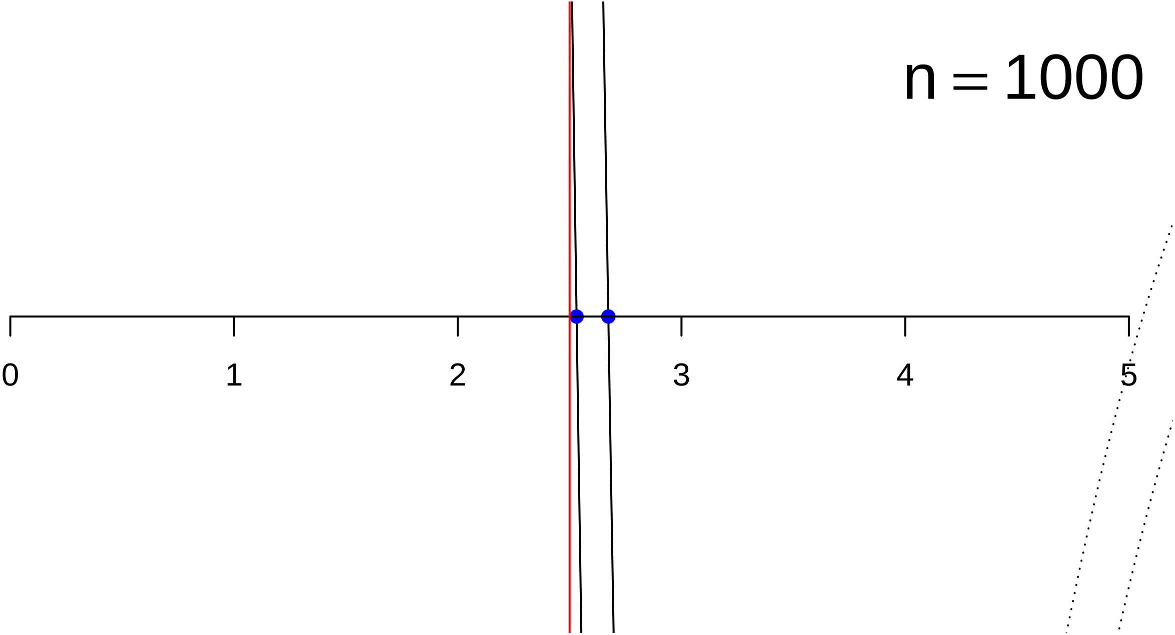

n = 100
0
1
2
3
4
5

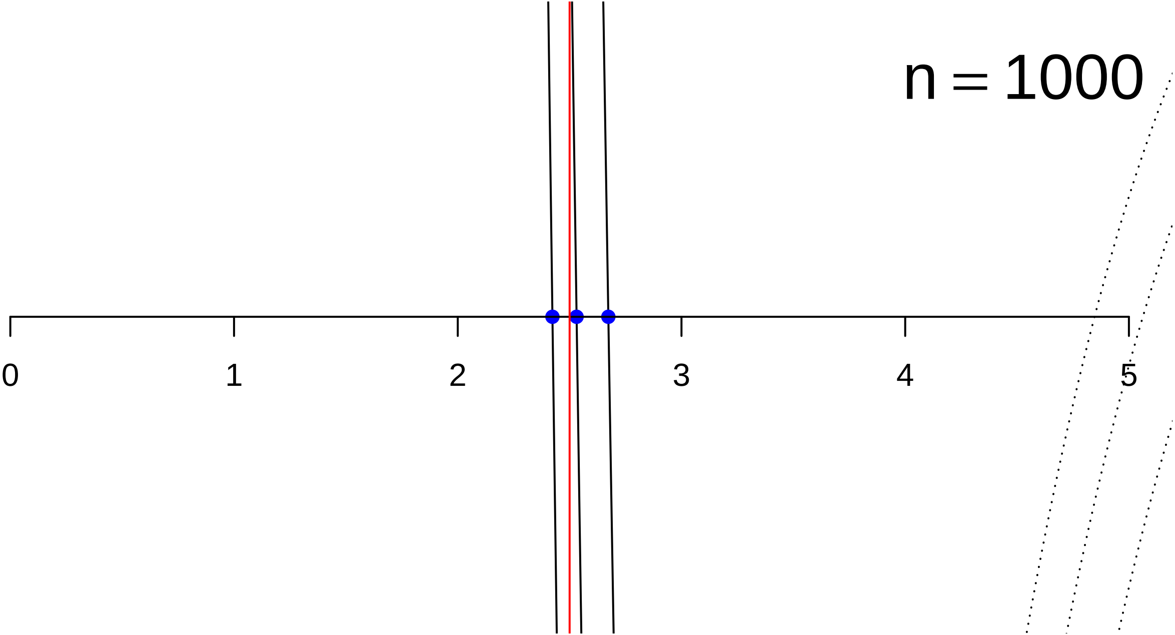

n = 100
0
1
2
3
4
5


n = 100
0
1
2
3
4
5

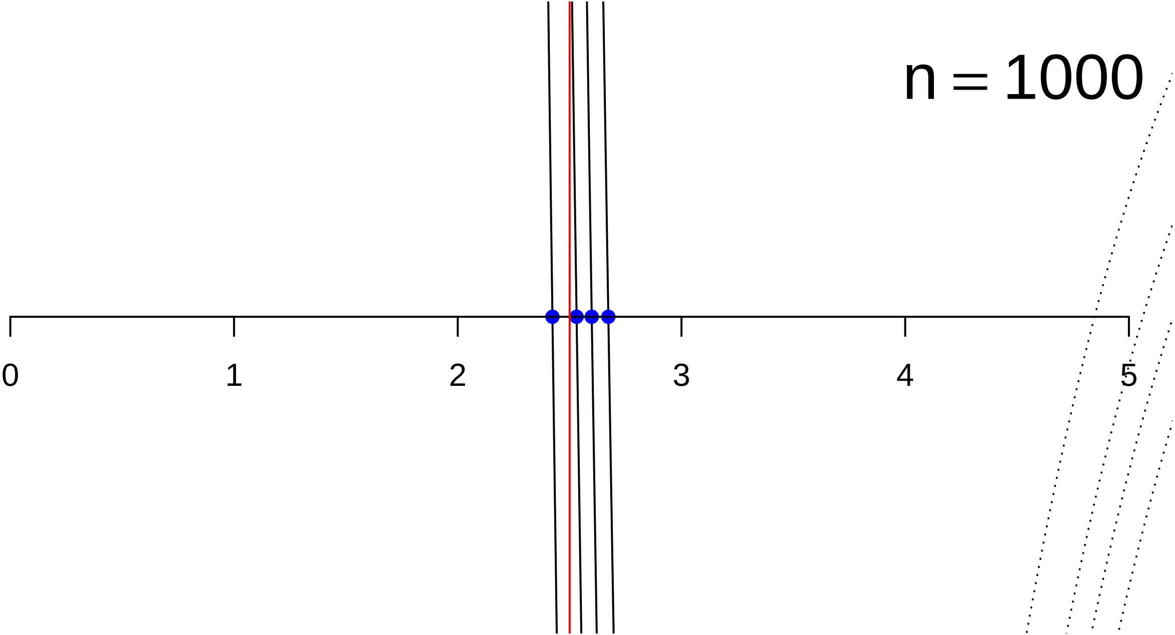
```r
# N = 1000

N <- 1000

xs <- lapply(rep(1000,10), rexp, 1/theta_true)

for(n in 1:10){
  plot_lx.o(xs[1:n], theta_true = 2.5)
  legend('topright', as.expression(bquote( n == .(N))) , cex = 2, bty = 'n')
  pause()
}
```


n = 1000
0
1
2
3
4
5

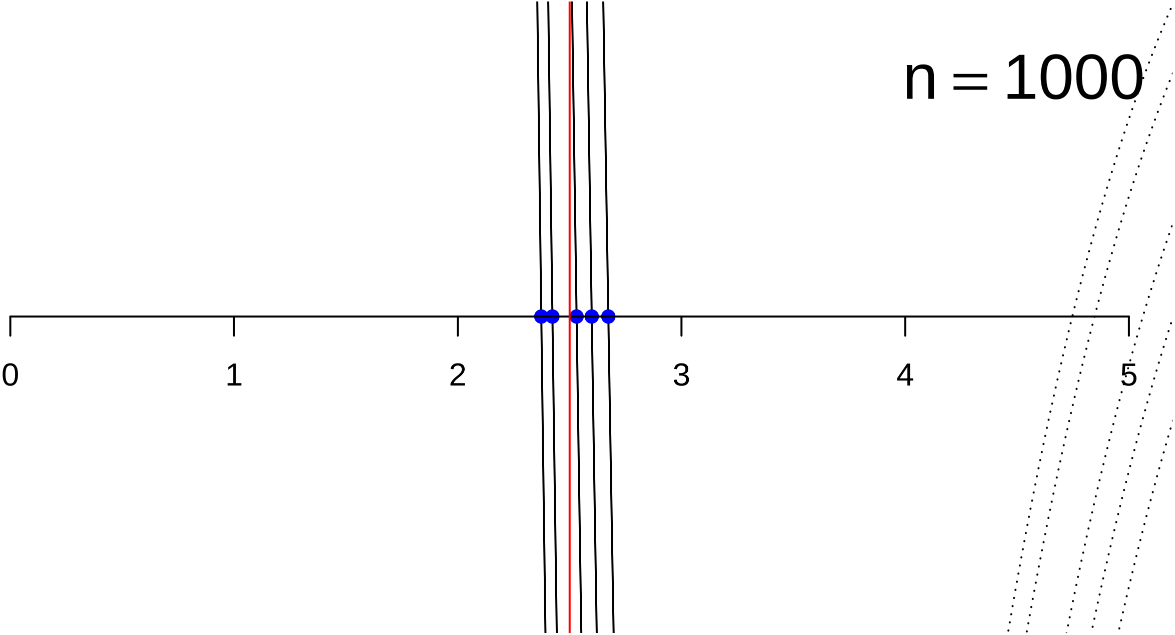

n = 1000
0
1
2
3
4
5

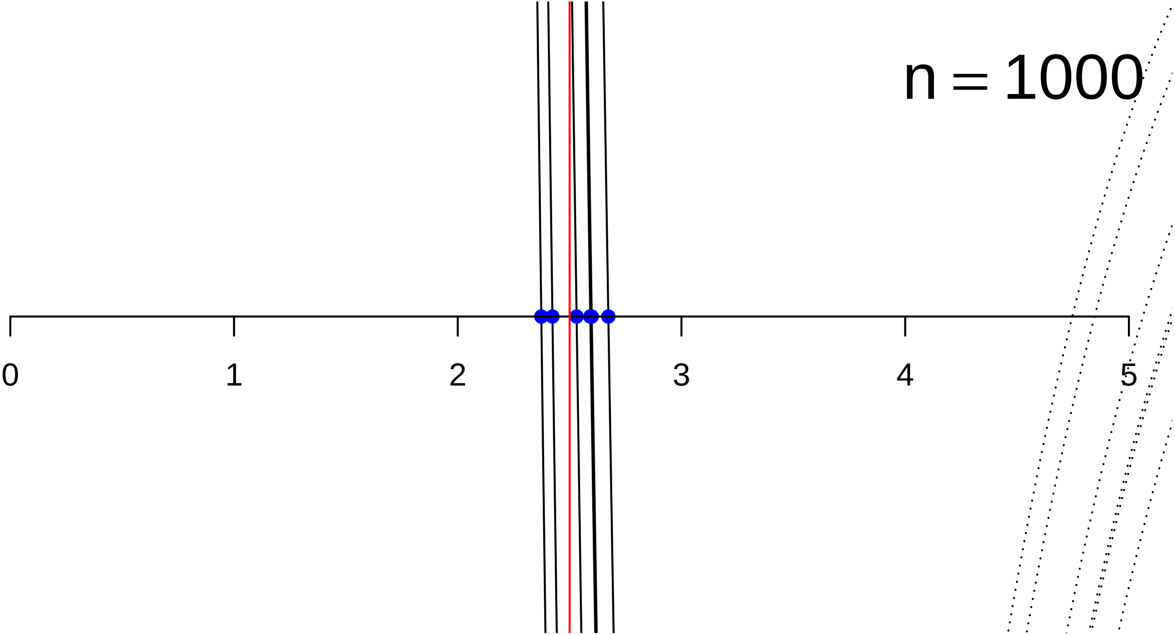


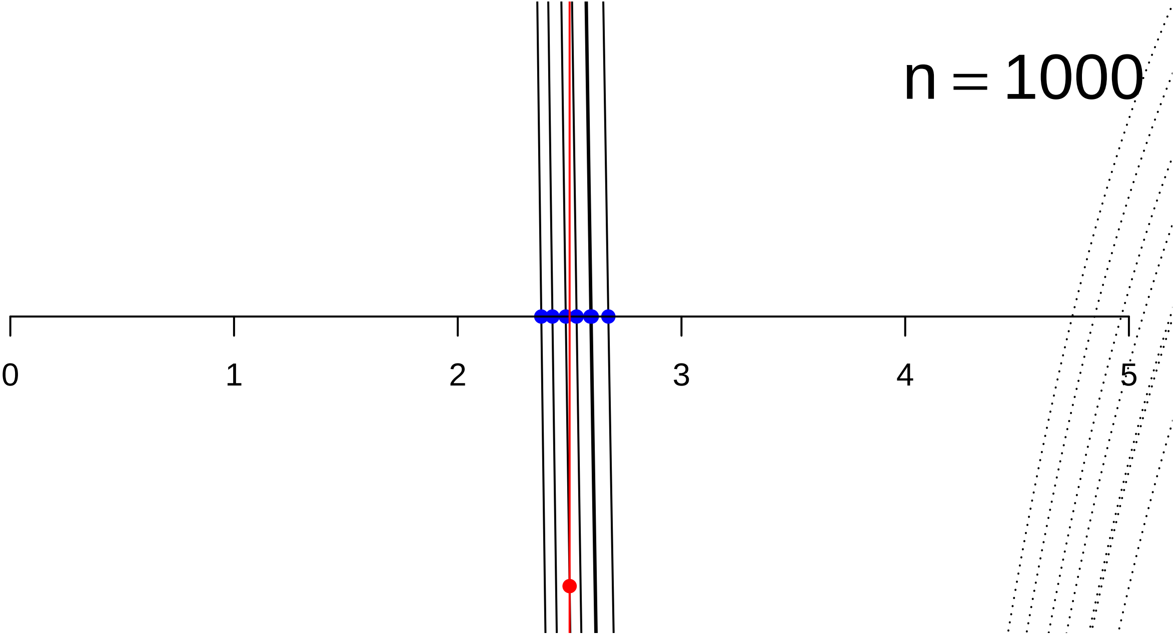
n = 1000
0
1
2
3
4
5

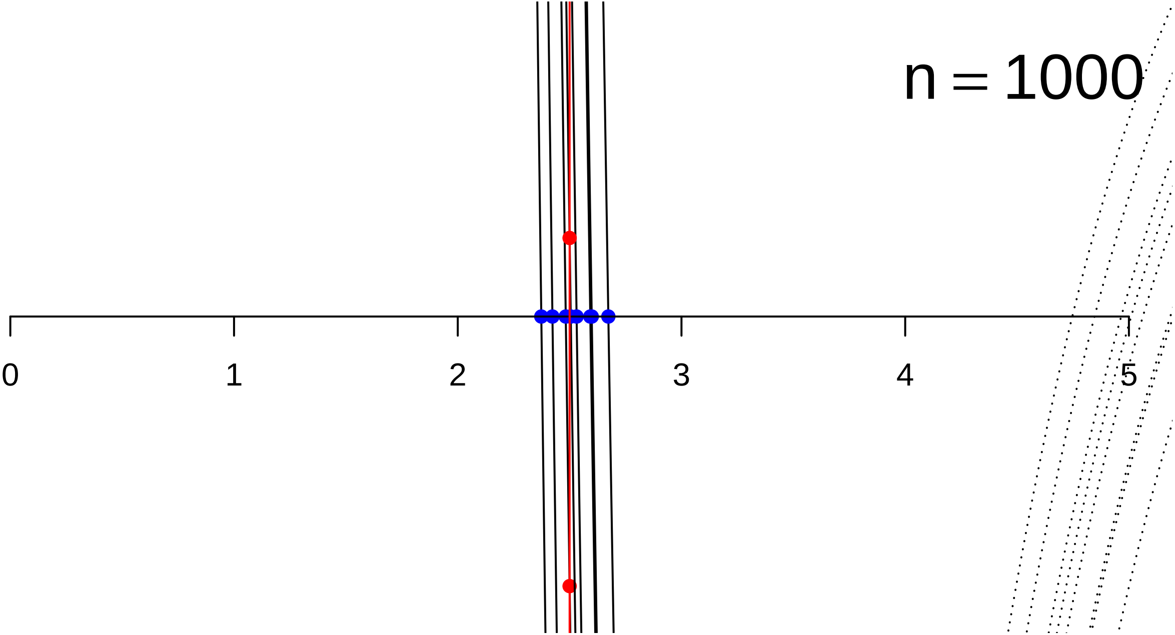


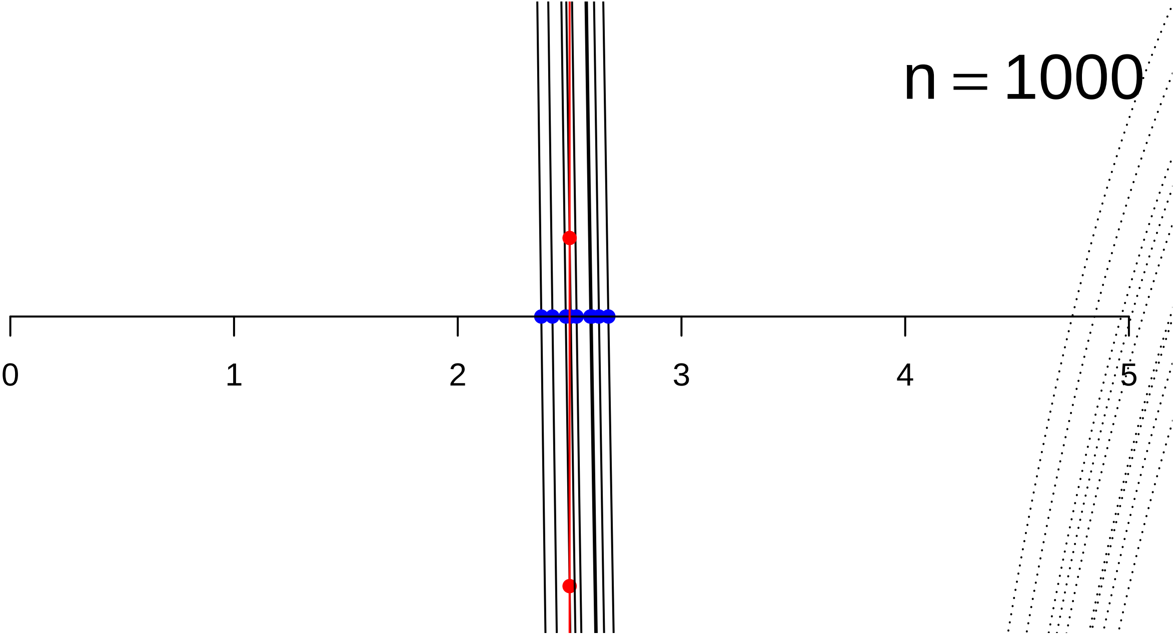
n = 1000
0
1
2
3
4
5

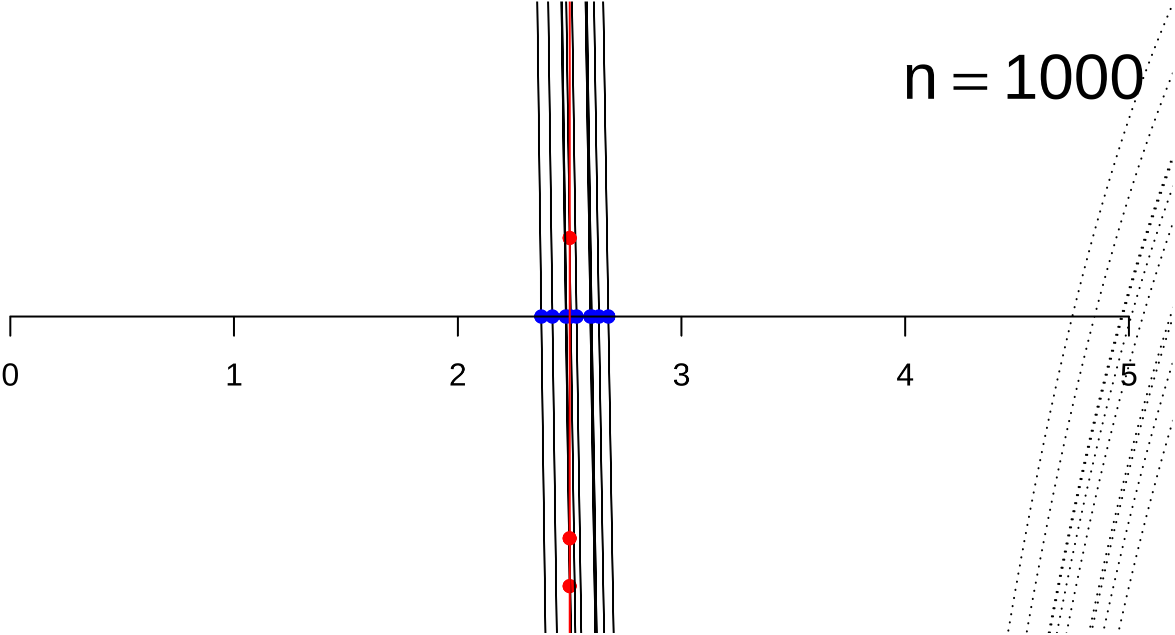

n = 1000
0
1
2
3
4
5


n = 1000
0
1
2
3
4
5


n = 1000
0
1
2
3
4
5


n = 1000
0
1
2
3
4
5


n = 1000
0
1
2
3
4
5


n = 1000
0
1
2
3
4
5