Example knitr document: estimating π

Stephen J Eglen

October 26, 2015

1 Introduction

This is an example document created using the knitr system (http://yihui.name/knitr/). knitr is a tool for combining both LaTeX documentation and R code within the same file, similar to Sweave. For this document, the master file is estimatek.Rnw. This is processed by knitr in R , which runs the R code to generate textual/graphical output, and also creates a LaTeX document. The LaTeX document is then typeset to create the pdf document. On recent machines, once knitr is installed, you should be able to generate the pdf using:

```
require(knitr)
knit2pdf('estimatek.Rnw')
```

Within RStudio, there is a handy "Compile PDF" button.

knitr is newer than Sweave, and is more flexible. Both estimatek.Rnw and estimatek.pdf are available from:

https://github.com/lgatto/spr/tree/master/estimate

(you will need estimate.Rnw and diff.R)

One key difference between Sweave and knitr is that knitr has built-in support for caching. By default it is turned off in this document, but change FALSE to TRUE in the following code-chunk and see if you can work out what happens:

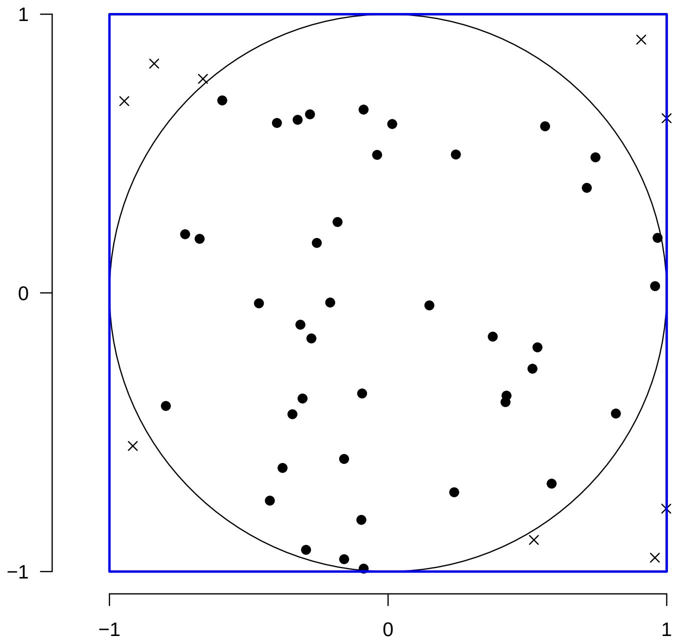
```
require(knitr)
require(xtable)

## Loading required package:  xtable

opts_chunk$set(cache=FALSE)       # £ (dollar needed by Emacs.)
```

2 Task: estimate the value of π

Our task is to estimate the value of π by simulating darts being thrown at a dartboard. Imagine that the person throwing the darts is not very good, and randomly throws each dart so that it falls uniformly within a square of side length $2r$, with the dartboard of radius r centred within that square. If the player throws n darts, and d of them hit the dartboard, then for large enough n, the ratio d/n should approximate the ratio of the area of the dartboard to the enclosing square, $\pi r^2/4r^2 \equiv \pi/4$. From this, we can estimate $\pi \approx 4d/n$.

We start with an example, using R to draw both the dartboard and the surrounding square, together with $n = 50$ darts. The radius of the dartboard here is 1 unit, although the value is not important.

```
r <- 1
n <- 50
par(las=1)
plot(NA, xlim=c(-r,r), ylim=c(-r,r), asp=1, bty='n',
     xaxt='n', yaxt='n', xlab='', ylab='')
axis(1, at=c(-r,0,r)); axis(2, at=c(-r,0,r))
symbols(x=0, y=0, circles=r, inch=F, add=T)
x <- runif(n, -r, r);  y <- runif(n, -r, r)
inside <- (x^2 + y^2) < r^2
d <- sum(inside)
points(x, y, pch=ifelse(inside, 19, 4))
rect(-r, -r, r, r, border='blue', lwd=2)
```



A dart is drawn as a filled circle if it falls within the dartboard, else it is drawn as a cross. In this case the number of darts within the circle is 41, and so the estimated value is $\pi \approx 3.28$.

The estimate of π should improve as we increase the number of darts thrown at the dartboard. To verify this, we write a short function that, given the number of darts to throw, n, returns an estimate of π.

```
estimate.pi <- function(n=1000) {
  ## Return an estimate of PI using dartboard
  ## method with N trials.
  r <- 1 ## radius of dartboard
  x <- runif(n, min=-r, max=r)
  y <- runif(n, min=-r, max=r)
  l <- sqrt(x^2 + y^2)
  d <- sum(l<r)
  4*d/n
}
```

We can then test the procedure a few times, using the default number of darts, 1000:

```
replicate(9, estimate.pi())

## [1] 3.212 3.132 3.132 3.068 3.100 3.124 3.120 3.160 3.132
```

Finally, for a given value of n, we can show 99 estimates of π, as clearly the estimate will vary from run to run. In Figure 1, we compare the estimates of π for three different values of n.

```
ns <- 10^c(2,3,4)
res <- lapply(ns, function(n) replicate(99, estimate.pi(n)))
par(las=1, bty='n')
stripchart(res, method="jitter", group.names=ns,
             xlab="number of darts",
             ylab=expression(paste('estimate of ', pi)),
             vert=TRUE, pch=20, cex=0.5)
abline(h=pi, col='red')
```

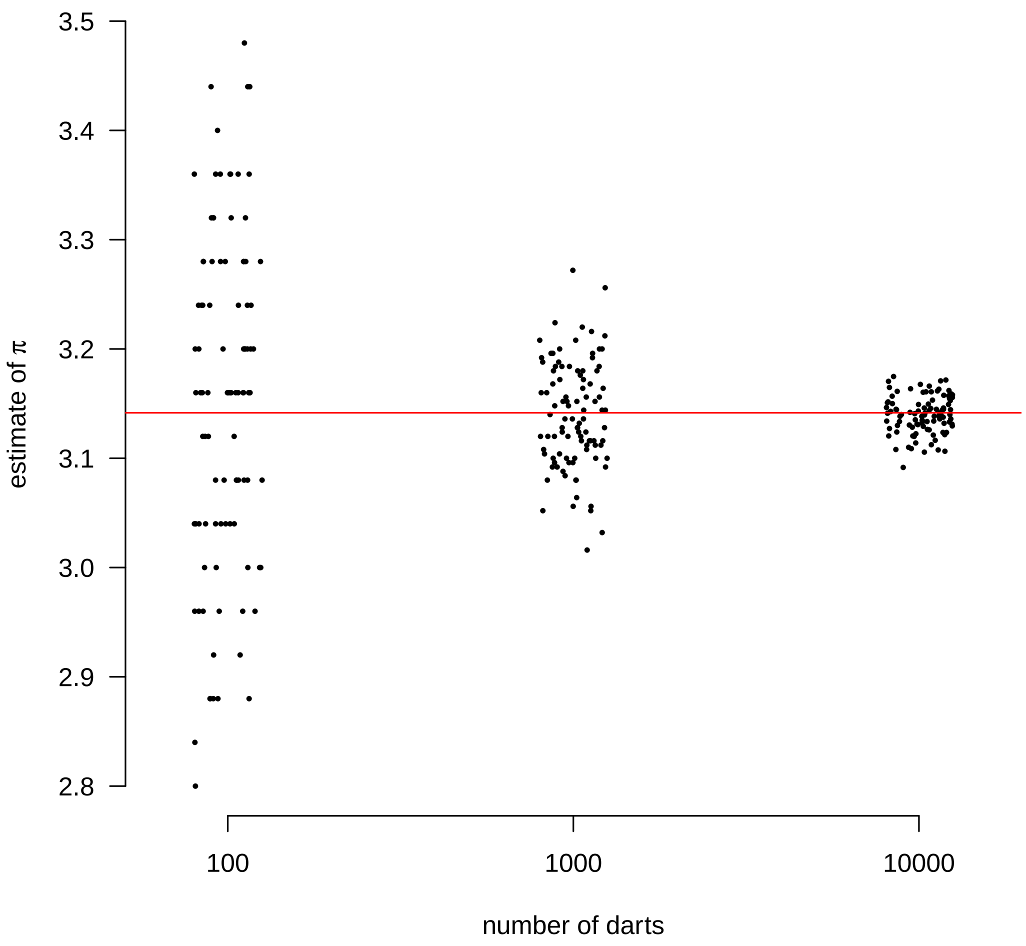


Figure 1: Mone-Carlo estimates of π improve as n increases. Red line denotes the true value of π.

xtable

xtable provides a convenient interface for making tables. Here's a simple example.

```
df = data.frame(name=c("joe", "ann", "bob"),
  age=c(19, 24, 27),
  height=c(1.8, 1.75, 1.7))
xtable(df)
```

	name	age	height
1	joe	19.00	1.80
2	ann	24.00	1.75
3	bob	27.00	1.70

Or, see how you can wrap it into a table environment, e.g. see Table 1.

```
xtable(df, caption="Example output from xtable.",label="tab:example")
```

	name	age	height
1	joe	19.00	1.80
2	ann	24.00	1.75
3	bob	27.00	1.70

Table 1: Example output from xtable.

Listing

Finally, an example listing. We read in all the code from a given file (here diff.R) and then assign it to the "all" label which we then include without evaluation:

```
diff1 <- function(e) {
  ## Explicit loop
  n <- length(e)
  interval <- rep(0, n-1) ## pre-allocate result
  for (i in 1:(n-1)) {
    interval[i] <- e[i+1] - e[i]
  }
  interval
}

diff2 <- function(e) {
  ## Vectorized solution
  n <- length(e)
  e[-1] - e[-n]
}

x <- c(5.9, 10.2, 12.4, 18.8)
all.equal(diff1(x), diff2(x))
```

See http://yihui.name/knitr/demo/externalization/ for further information.