Introduction to the Practice of Statistics using R: Chapter 4

Nicholas J. Horton* Ben Baumer

March 10, 2013

Contents

Introduction

This document is intended to help describe how to undertake analyses introduced as examples in the Sixth Edition of *Introduction to the Practice of Statistics* (2009) by David Moore, George McCabe and Bruce Craig. More information about the book can be found at `http://bcs.whfreeman.com/ips6e/`. This file as well as the associated `knitr` reproducible analysis source file can be found at `http://www.math.smith.edu/~nhorton/ips6e`.

This work leverages initiatives undertaken by Project MOSAIC (`http://www.mosaic-web.org`), an NSF-funded effort to improve the teaching of statistics, calculus, science and computing in the undergraduate curriculum. In particular, we utilize the `mosaic` package, which was written to simplify the use of R for introductory statistics courses. A short summary of the R needed to teach introductory statistics can be found in the mosaic package vignette (`http://cran.r-project.org/web/packages/mosaic/vignettes/MinimalR.pdf`).

To use a package within R, it must be installed (one time), and loaded (each session). The package can be installed using the following command:

```
> install.packages('mosaic')                # note the quotation marks
```

The # character is a comment in R, and all text after that on the current line is ignored.

Once the package is installed (one time only), it can be loaded by running the command:

*Department of Mathematics and Statistics, Smith College, nhorton@smith.edu

```
> require(mosaic)
```

This needs to be done once per session.
We also set some options to improve legibility of graphs and output.

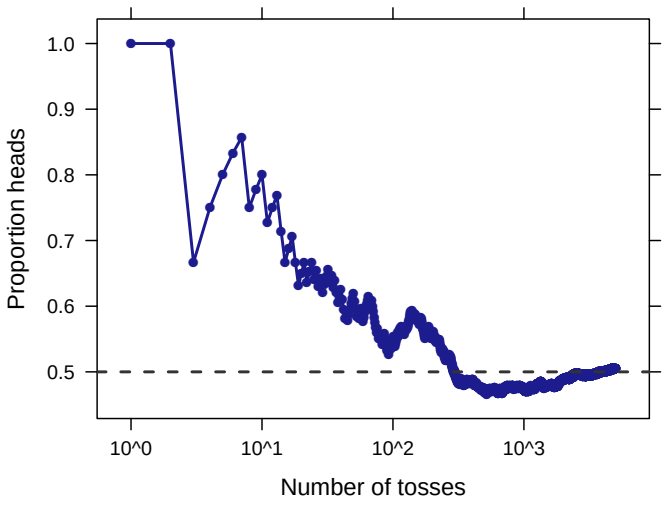
```
> trellis.par.set(theme=col.mosaic())  # get a better color scheme for lattice
> options(digits=3)
```

The specific goal of this document is to demonstrate how to replicate the analysis described in Chapter 4: Probability (The Study of Randomness).

1 Randomness

It's straightforward to replicate displays such as Figure 4.1 (page 240) using R. We begin by specifying the random number seed (this is set arbitrarily if `set.seed()` is not run), then generating a thousand coin flips (using `rbinom()`) then calculating the running average for each of the tosses. To match the Figure, we use a log scale for the x-axis.

```
> set.seed(42)
> numtosses = 5000
> runave = numeric(numtosses)
> toss = rbinom(numtosses, size=1, prob=0.50)
> for (i in 1:numtosses) {
  runave[i] = mean(toss[1:i])
 }
> xyplot(runave ~ 1:numtosses, type=c("p", "l"), scales=list(x=list(log=T)),
   ylab="Proportion heads", xlab="Number of tosses", lwd=2)
> ladd(panel.abline(h=0.50, lty=2))
```



Random digits can be sampled using the `sample()` command, as described using Table B at the bottom of page 239 (0 through 4 called *tails* or false and 5 through 9 *heads* or true:

```
> x = sample(0:9, size=10, replace=TRUE)
> x

 [1] 7 5 4 8 4 1 6 9 7 0

> x > 4

 [1]  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE
```

Alternatively, heads and tails can be generated directly.

```
> rbinom(10, size=1, prob=0.50)

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

2 Probability models

The `mosaic` package includes support for samples of cards.

```
> Cards

 [1] "2C"  "3C"  "4C"  "5C"  "6C"  "7C"  "8C"  "9C"  "10C" "JC"  "QC"
[12] "KC"  "AC"  "2D"  "3D"  "4D"  "5D"  "6D"  "7D"  "8D"  "9D"  "10D"
[23] "JD"  "QD"  "KD"  "AD"  "2H"  "3H"  "4H"  "5H"  "6H"  "7H"  "8H"
[34] "9H"  "10H" "JH"  "QH"  "KH"  "AH"  "2S"  "3S"  "4S"  "5S"  "6S"
[45] "7S"  "8S"  "9S"  "10S" "JS"  "QS"  "KS"  "AS"
```

Let's create a deck which is missing an ace (to verify the calculation on page 252):

```
> noacespades = subset(Cards, Cards != "AS")
> noacespades

 [1] "2C"  "3C"  "4C"  "5C"  "6C"  "7C"  "8C"  "9C"  "10C" "JC"  "QC"
[12] "KC"  "AC"  "2D"  "3D"  "4D"  "5D"  "6D"  "7D"  "8D"  "9D"  "10D"
[23] "JD"  "QD"  "KD"  "AD"  "2H"  "3H"  "4H"  "5H"  "6H"  "7H"  "8H"
[34] "9H"  "10H" "JH"  "QH"  "KH"  "AH"  "2S"  "3S"  "4S"  "5S"  "6S"
[45] "7S"  "8S"  "9S"  "10S" "JS"  "QS"  "KS"
```

How often is the next card also an ace? We know that the true answer is 3/51 (or 0.059), and we can estimate this through sampling.

```
> res = do(10000) * sample(noacespades, size=1, replace=TRUE)
> head(res)

  result
1    10C
2     8H
3     KS
4     4H
5     9D
6     KC

> tally(~ (result %in% c("AD", "AC", "AH")), format="percent", data=res)


  TRUE  FALSE  Total
  5.42  94.58 100.00
```

3 Random variables

Example 4.23 (page 261) derives the Binomial distribution when $n = 4$ and $p = 0.50$.

```
> dbinom(0:4, size=4, prob=0.50)   # probability mass function

[1] 0.0625 0.2500 0.3750 0.2500 0.0625

> pbinom(0:4, size=4, prob=0.50)   # cumulative probability

[1] 0.0625 0.3125 0.6875 0.9375 1.0000
```

Example 4.24 (page 262) asks about the probability of at least two heads, which is equivalent to one minus the probability of no more than one head, or $P(X = 2) + P(X = 3) + P(X = 4)$.

```
> dbinom(2:4, size=4, prob=0.50)

[1] 0.3750 0.2500 0.0625

> sum(dbinom(2:4, size=4, prob=0.50))

[1] 0.688

> 1 - pbinom(1, size=4, prob=0.50)

[1] 0.688
```

Calculations for Uniform random variables can be undertaken as easily (as seen in Example 4.25, page 263):

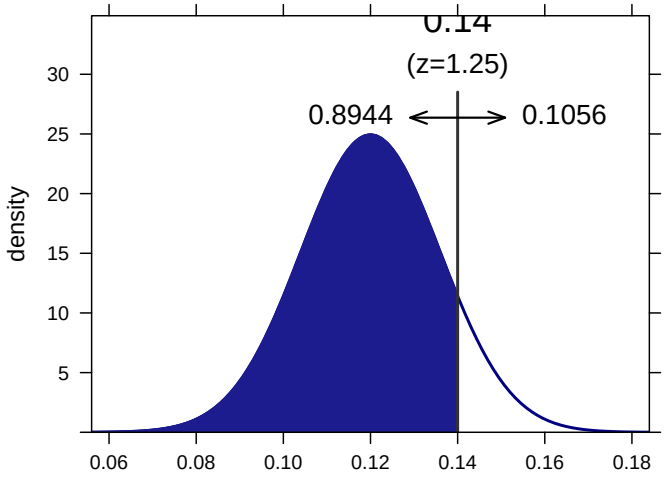
```
> punif(0.7, min=0, max=1)

[1] 0.7

> punif(0.3, min=0, max=1)

[1] 0.3

> punif(0.7, min=0, max=1) - punif(0.3, min=0, max=1)

[1] 0.4
```

Simulation studies are also easy to carry out:

```
> randnums = runif(10000, min=0, max=1)
> head(randnums)

[1] 0.0416 0.0387 0.3144 0.5852 0.4764 0.9085

> tally(~ (randnums > 0.3 & randnums < 0.7), format="percent")


 TRUE FALSE Total
 40.1  59.9 100.0
```

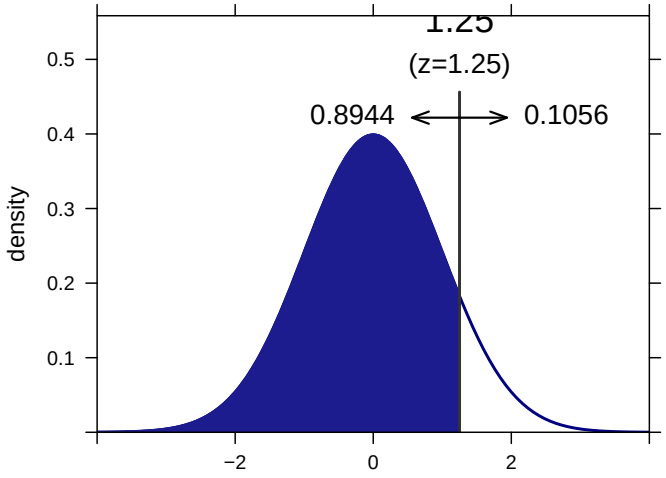
Example 4.26 (pages 265–266) displays the same type of calculation for a normal random variable:

```
> xpnorm(0.14, mean=0.12, sd=0.016)


If X ~ N(0.12,0.016), then

P(X <= 0.14) = P(Z <= 1.25) = 0.8944
P(X >  0.14) = P(Z >  1.25) = 0.1056
[1] 0.894

> pnorm(0.10, mean=0.12, sd=0.016)

[1] 0.106

> pnorm(0.14, mean=0.12, sd=0.016) - pnorm(0.10, mean=0.12, sd=0.016)

[1] 0.789
```



or on the normalized scale:

```
> xpnorm(1.25, mean=0, sd=1)


If X ~ N(0,1), then

P(X <= 1.25) = P(Z <= 1.25) = 0.8944
P(X >  1.25) = P(Z >  1.25) = 0.1056
[1] 0.894

> pnorm(-1.25, mean=0, sd=1)

[1] 0.106

> pnorm(1.25, mean=0, sd=1) - pnorm(-1.25, mean=0, sd=1)

[1] 0.789
```



4 Means and variances of random variables

Example 4.29 (page 272) calculates the mean of the first digits following Benford's law:

```
> V = 1:9
> probV = c(0.301, 0.176, 0.125, 0.097, 0.079, 0.067, 0.058, 0.051, 0.046)
> sum(probV)

[1] 1

> xyplot(probV ~ V, xlab="Outcomes", ylab="Probability")
> V*probV

[1] 0.301 0.352 0.375 0.388 0.395 0.402 0.406 0.408 0.414

> benfordmean = sum(V*probV)
> benfordmean

[1] 3.44
```

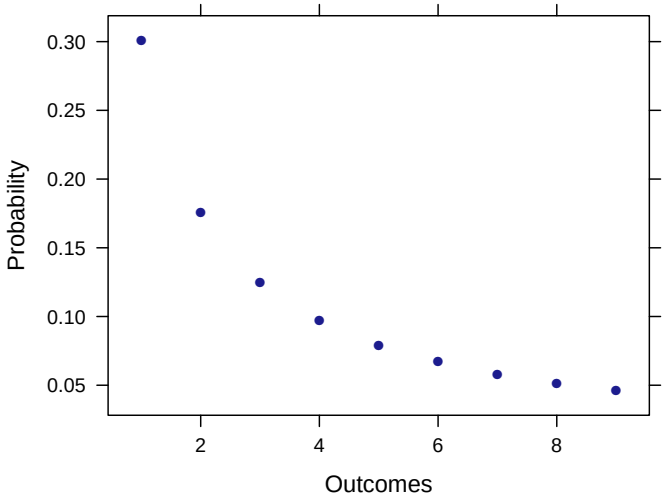
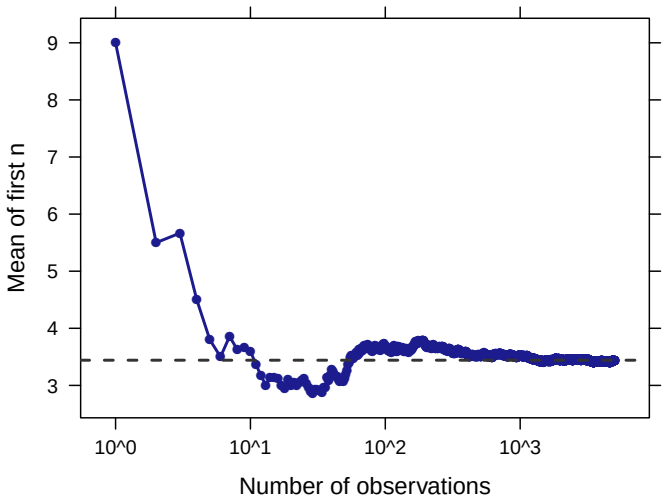


Figure 4.14 (page 275) describes the law of large numbers in action. We can display this for samples from the Benford distribution:

```
> runave = numeric(numtosses)
> benford = sample(V, size=numtosses, prob=probV, replace=TRUE)
> for (i in 1:numtosses) {
  runave[i] = mean(benford[1:i])
 }
> xyplot(runave ~ 1:numtosses, type=c("p", "l"), scales=list(x=list(log=T)),
   ylab="Mean of first n", xlab="Number of observations", lwd=2)
> ladd(panel.abline(h=3.441, lty=2))
```



The variance (introduced on page 280) can be carried out in a similar fashion:

```
> sum((V - benfordmean)^2 * probV)

[1] 6.06
```

Note that we can estimate this value from the variance of the simulated samples above:

```
> var(benford)

[1] 6.11
```

Similar calculations can be undertaken on either the original or linearly transformed scale for the Tri-State pick 3 lottery example (4.34) on page 282:

```
> X = c(0, 500)
> probX = c(0.999, 0.001)
> xmean = sum(X*probX)
> xmean

[1] 0.5

> sum((X - xmean)^2 * probX)

[1] 250
```

For Example 4.35 (page 283), since $W = X - 1$ we know that $\mu_w = \mu_x - 1$:

```
> W = X - 1
> wmean = sum(W*probX)
> wmean

[1] -0.5

> sum((W - wmean)^2 * probX)

[1] 250
```