

4.4 Appendix: Computing Probabilities in R

by EV Nordheim, MK Clayton & BS Yandell, September 23, 2003

R can be used to compute probabilities of interest associated with numerous probability distributions. In this section we describe its use for calculating probabilities associated with the binomial, Poisson, and normal distributions.

First, we discuss computing the probability of a particular outcome for discrete distributions. (This can also be used with continuous distributions for computing probability densities but we will not be concerned with that application here.) R begins these commands with the letter **d** for distribution or density, namely **dbinom** and **dpois**.

Consider again the calf example with abnormal clotting discussed near the beginning of the section on the binomial distribution. The random variable is $X \sim B(4, p)$. In the text we indicated that if $p = 0.3$, we can calculate $P(X = 4) = 0.0081$. We can use the **dbinom** command to perform this calculation.

```
> dbinom(4, 4, 0.3)
```

```
[1] 0.0081
```

The number 4 after the **dbinom** command indicates that $X = 4$ is the value for which the probability is required. The second and third arguments specify the n and p parameters of the binomial distribution, 4 and 0.3 respectively in this case. The resultant probability is 0.0081 as expected.

In the text we also considered the case $X \sim B(8, p)$ and computed $P(X = 6)$ with $p = 0.3$. We do the same with R.

```
> dbinom(6, 8, 0.3)
```

```
[1] 0.01000188
```

Here is an example of the use of the **dpois** command with a Poisson random variable. Recall the cake example. Here the random variable is $X \sim Po(2)$. Calculating by hand, we found $P(X = 3)$ to be 0.1804. Here is the result using R:

```
> dpois(3, 2)
```

```
[1] 0.1804470
```

As noted, since the Poisson is defined by only a single parameter, we only need that parameter (λ) in the second argument.

R starts distribution commands with the letter **p**, for probability, to compute $P(X \leq x)$ for any value x . We illustrate this command with the binomial and normal distribution. For the binomial situation, consider again the calf example. In Section 4.1 we computed $P(X \leq 1)$ for $X \sim B(8, 0.3)$. Using the **pbinom** command we proceed as follows.

```
> pbinom(1, 8, 0.3)
```

```
[1] 0.2552983
```

The input is very similar except that `pbinom` replaces `dbinom`. Note that since the binomial distribution is discrete, there is a difference between $P(X < 1)$ and $P(X \leq 1)$. The `pbinom` command *always* includes the upper limit.

When we consider normal random variables, we need to point out a distinction between the `pnorm` command in R and the tables in this book. The `pnorm` command finds the probability to the *left* of a particular value, while the normal tables in this book give probabilities to the *right* of a particular value. (Probabilities to the right of a given value are of direct utility in testing as we will see in Chapter 6.) You can use the `lower.tail=FALSE` argument to get the upper tails with the `pnorm`, as shown below.

Recall the topsoil example in which X represents the pH of a random sample of topsoil in the vicinity of Oxford. In this example $X \sim N(7.1, 0.6^2)$. As always, the parameters in this notation are the mean, μ and the variance σ^2 . We wish to find $P(X \leq 6.2)$. In R we enter:

```
> pnorm(6.2, 7.1, 0.6)
```

```
[1] 0.0668072
```

```
> pnorm(6.2, 7.1, 0.6, lower.tail = FALSE)
```

```
[1] 0.9331928
```

The second and third arguments are μ and σ , respectively. Thus, we must be careful to type in the standard deviation and not the variance. Note that with the normal distribution (as with any continuous distribution) $P(X \leq 6.2) = P(X < 6.2)$. When the argument `lower.tail` is set to `FALSE`, we get $P(X > 6.2)$ in the second command line.

Finally we can make the *inverse* calculation in which we calculate the point x so that $P(X \leq x)$ is equal to some given probability. The R commands for this begin with the letter `q` for quantile. This command is awkward to use with a discrete distribution because of the discrete jumps in probability. However, it is easy and useful to use the `qnorm` command with the continuous normal distribution. For example, consider again the GRE example with $X \sim N(485, 123^2)$. We wish to find x so that $P(X \leq x) = 0.25$. We proceed as follows.

```
> qnorm(0.25, 485, 123)
```

```
[1] 402.0378
```

The answer from R is not rounded to the nearest integer as was done in our example. This recalls our caution that a normal model may not be totally correct for the GRE situation even though it gives results that are useful in practice.