NRES 798 — Lab 3

Probability — Pitcher plants

```
> # Use RStudio
> # Try to guess the output before hitting Enter
> # Experiment!  Use the Help
> # If after that you do not understand, ask
>
>
> #####  Pitcher plants textbook example ####
>
> visit <- c('capture', 'escape')  # sample space
> # Actually, this is a vector, not a "proper" set. What is the difference?
> # Never mind, let's use vectors to represent sets
> # Random variables are functions mapping the sample space into numbers:
> VisitRV <- function(outcome) {
+ # Visit random variables
+     if (outcome == 'capture') return(1)
+     if(outcome == 'escape') return(0)
+     return(NA)
+ }
# (Let's capitalize function names to distinguish them from variables)
> VisitRV
 ...   (I'll use ... to represent whatever your output was)
> VisitRV('capture')
 ...
> VisitRV(visit[2])
 ...
> VisitRV('huh?')
 ...
> VisitRV(visit)
 ...
> # Let's re-write it so that it works for vectors, not just single numbers
```

```
> fix(VisitRV)  # or click on the function name in the Workspace listing,
> # upper-right. Edit to:
> VisitRV
function(outcome) {
# Visit random variables
    c(1, 0)[match(outcome, visit)]
}
> ?match
> # Get it? Think about it. Overkill here, but works for larger sample spaces
> # If return() is omitted, a function returns the last computed value
> VisitRV(visit)
 ...
> VisitRV('escape')
 ...
> VisitRV('huh?')
 ...
# We can assign probabilities, for a known theta:
> theta <- 0.2
> pvisit <- c(theta, 1 - theta)
> pvisit
 ...
# For more flexibility, use a function
> Pvisit <- function(outcomes, theta) {}  # edit so that:
> Pvisit
function(outcomes, theta) {
# Probabilities for visit
    c(theta, 1 - theta)[match(outcomes, visit)]
}
> Pvisit(visit, 0.2)
 ...
> This is a density, the prob. would be sum(Pvisit(visit, 0.2))
> Pvisit('escape', 0.2)
 ...
> Pvisit('escape', 0.01)
 ...
>
> # Assume theta = 0.2, and take a sample of 30 visits :
x <- sample(visit, 30, replace=T, prob=Pvisit(visit, 0.2))
> # (enter sample( and press Tab, and then F1, to get help)
x
```

```
 ...
> # How many captures?
> sum(x == 'capture')  # how does this work?  Hint: "coercion"
 ...
> # Frequency (estimate for the probability theta):
> sum(x == 'capture') / length(x)
 ...
> mean(x == 'capture')
 ...
> # Try a few more times, with different samples (and sample sizes,
> #  and theta's)
 ...
> # Same thing, with the random variables:
> x <- sample(VisitRV(visit), 30, replace=T, prob=Pvisit(visit, 0.2))
> x
 ...
> mean(x)  # why does this work?
 ...
> # Try samples of size 3000. Might be useful to use
> head(x)  # displays only the start of a vector
> ...
> # What is the effect of sample size? One way of seeing is to compute
> #  frequencies for the first 1, 2, ..., 3000 values of x:
> freqs <- cumsum(x) / 1:3000
> head(freqs, 33)
 ...
> tail(freqs)
 ...
> plot(freqs)
> abline(h=0.2)
> # Better is to generate 3000 independent samples of sizes 1, ..., 3000
> #   A bit more complicated, do not worry if you do not follow this:
> # first, allow sample() to use a vector for the size argument
> vsample <- Vectorize(sample, c('size'))  # a functional (fun. of functions)
> # then, use this as before, but with a vector of sizes
> x <- vsample(VisitRV(visit), 1:3000, replace=T, prob=Pvisit(visit, 0.2))
> head(x)
 ...
> # the result is a list of samples. Apply the mean to each one
> freqs <- sapply(x, mean)
```

```
> head(freqs)
 ...
> tail(freqs)
 ...
> plot(freqs)
> abline(h=0.2)
> # phew!
> # Perhaps simpler with a for loop, as below, but in R loops are uncool
>
> # Now, simulate the textbook sample, captures per 1000 visits over
> #  52 weeks. Assume theta = 0.01.
> # We could use vsample above, but let's be more pedestrian this time:
> capdata <- data.frame(week=1:52, ncaps=0)
> head(capdata)
 ...
> for(wk in 1:52) capdata$ncaps[wk] <- sum(sample(VisitRV(visit), 1000,
+  replace=T, prob=Pvisit(visit, 0.01)))
> head(capdata)
 ...
> summary(capdata)
 ...
> hist(capdata$ncaps)
> table(capdata$ncaps)
 ...
> plot(table(capdata$ncaps), xlab='Captures / 1000 visits',
+ ylab='Number of observations')
> # The sampling distribution of this statistic (number of
> #  captures per 1000 visits) is a Binomial(1000, theta):
> points(52 * dbinom(1:20, 1000, 0.01))  # theoretical
> # Is the model wrong? Are these differences likely to
> #  happen by pure chance?  To be continued...
>
> ?Distributions  # what other distributions are available?
> ?Random  # in case you are curious about random numbers
>
> # Go back and experiment.
```