Manipulation of Discrete Random Variables in R with discreteRV

Eric Hare, Andreas Buja, Heike Hofmann

July 2, 2015

The Problem

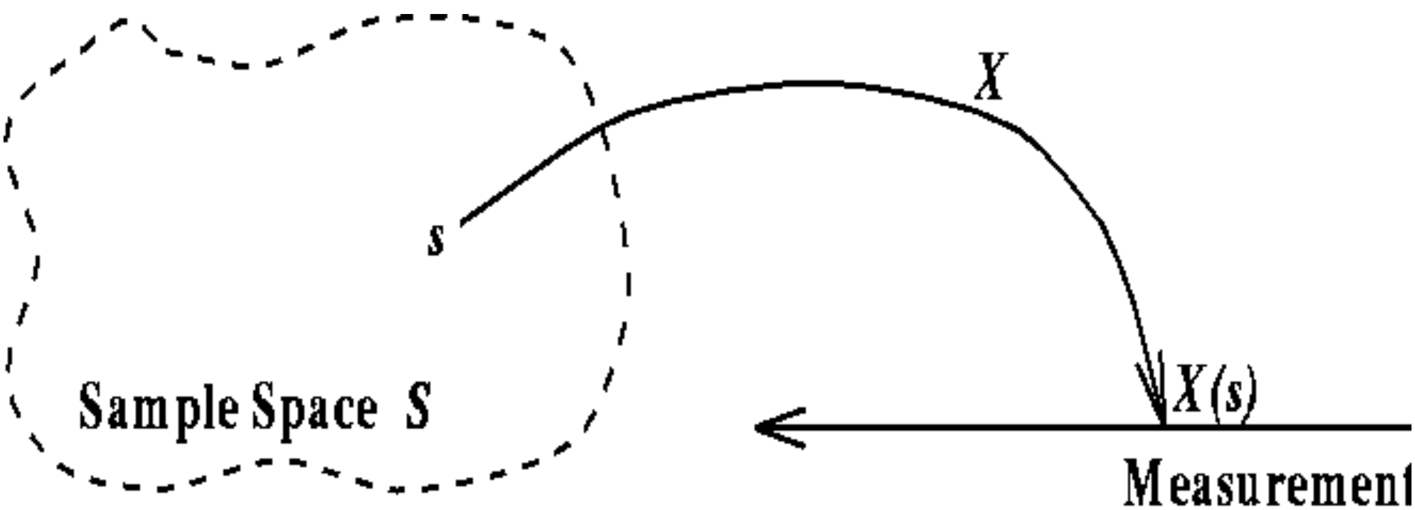
In statistics education, there is often a large disparity between the **theoretical/mathematical** coursework, and the **computational/programming** coursework.

- Different notations
- Different points of emphasis
- Little coding in theory, and little theory in coding

This can cause difficulty for an introductory statistics student with little experience in one of the two areas.

Example: The Notion of a Random Variable

- A **Random Variable** is a function that maps the set of possible outcomes of an experiment to the real line.
- Random Variables are associated with a probability density function (in the continuous case) and a probability mass function (in the discrete case)
- A probability density/mass function provides the probabilities of the possible values of the random variable.



Example (continued)

In theory textbooks:

$X \sim N(4, 2)$

$$\begin{aligned} P(X > 5) &= 1 - P(X < 5) \\ &= 1 - P(Z < 0.707) \\ &= .2398 \end{aligned}$$

In R:

```
1 - pnorm(5, 4, sqrt(2))
```

```
## [1] 0.2397501
```

discreteRV

Manipulation of Discrete Random Variables

- Goal: Thorough, powerful interface for working with discrete random variables in R
- Use a syntax consistent with introductory statistics texts
- Based on idea and code by Andreas Buja (Wharton School of Business)
- My contribution: additional functionality, packaging, documentation, and write-up

Follow along at: http://erichare.me/talks/useR/2015

Installing discreteRV

discreteRV is available on CRAN (stable) and GitHub (development)

1. Install from CRAN or GitHub

```
install.packages("discreteRV")
# or...
library(devtools)
install_github("erichare/discreteRV")
```

2. Load the package

```
library(discreteRV)
```

6/27

Creating discrete random variables

Let X be a random variable representing a single toss of a fair die. X takes on the values 1 to 6 with probability $\frac{1}{6}$

$$P(X = x) = \frac{1}{6} \quad x \in \{1, 2, 3, 4, 5, 6\}$$

In discreteRV,

```
X <- RV(1:6)
X <- RV(1:6, probs = rep(1/6, 6)) # Equivalent

X
```

```
## Random variable with 6 outcomes
##
## Outcomes   1   2   3   4   5   6
## Probs    1/6 1/6 1/6 1/6 1/6 1/6
```

7/27

Creating other discrete RVs

Let Y be distributed according to a poisson random variable with mean parameter 5.

$$P(Y = y) = \frac{\lambda^y e^{-\lambda}}{y!} \quad y \in \{0, 1, \dots\}$$

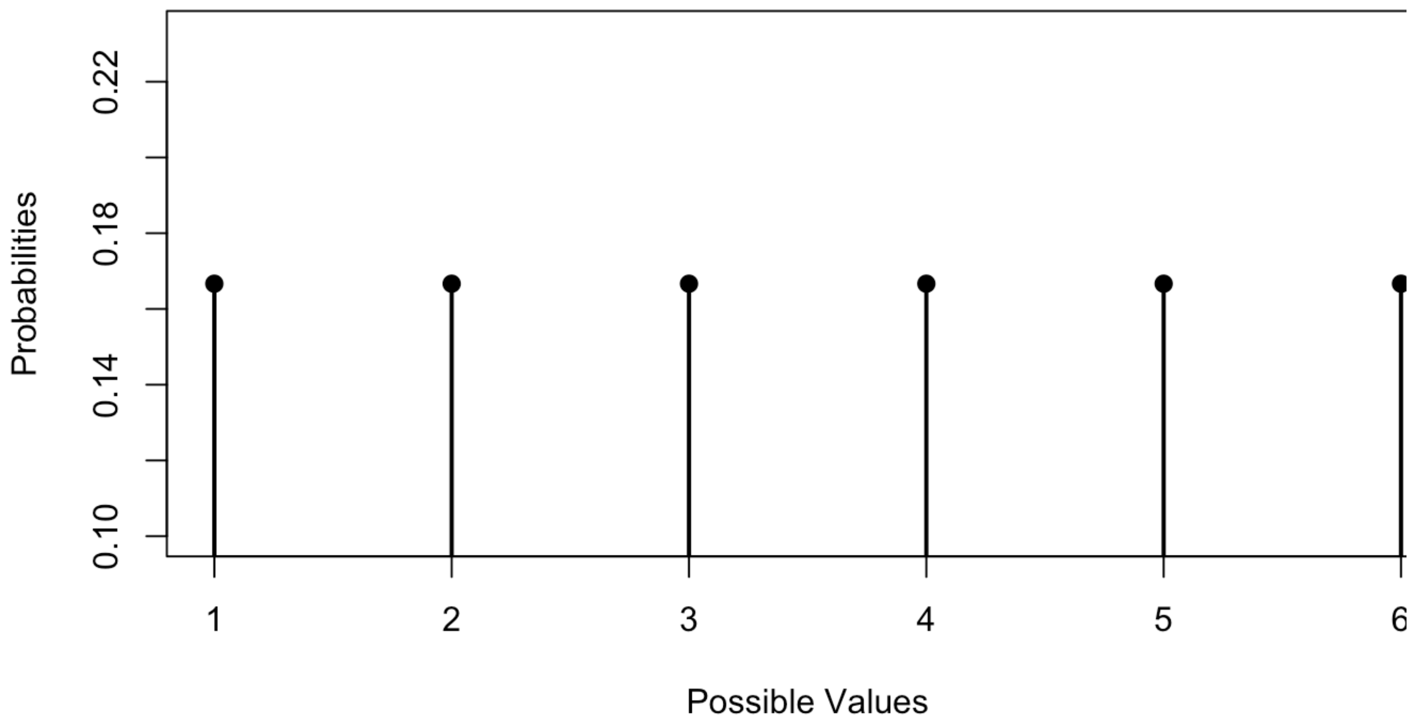
```
pois.func <- function(y, lambda) { lambda^y * exp(-lambda)
Y <- RV(c(0, Inf), pois.func, lambda = 5)

Y <- RV("poisson", lambda = 5, fractions = FALSE)
Y
```

```
## Random variable with outcomes from 0 to Inf
##
## Outcomes      0     1     2     3     4     5     6     7
## Probs     0.007 0.034 0.084 0.140 0.175 0.175 0.146 0.104
##
## Displaying first 12 outcomes
```

Plot Method

```
plot(X)
```



Probability Functions

discreteRV features probability functions which have a syntax very similar to Casella & Berger:

```
P(X > 4)
```

```
## [1] 0.3333333
```

```
P(X == 1 | X <= 2)
```

```
## [1] 0.5
```

```
P((X == 2) %OR% (X == 3))
```

```
## [1] 0.3333333
```

Probability Functions (continued)

```
E(X)
```

```
## [1] 3.5
```

```
V(X)
```

```
## [1] 2.916667
```

```
SD(X)
```

```
## [1] 1.707825
```

Joint Distributions

discreteRV allows easy computation of joint distributions:

```
iid(X, n = 2, fractions = TRUE)
```

```
## Random variable with 36 outcomes
##
## Outcomes  1,1  1,2  1,3  1,4  1,5  1,6  2,1  2,2  2,3  2
## Probs    1/36 1/36 1/36 1/36 1/36 1/36 1/36 1/36 1/36 1/
##
## Displaying first 12 outcomes
```

Joint Distributions (continued)

Joint distributions need not be iid:

```
AB <- jointRV(list(1:3, 0:2), probs = 1:9 / sum(1:9))
AB
```

```
## Random variable with 9 outcomes
##
## Outcomes  1,0  1,1  1,2  2,0  2,1  2,2  3,0  3,1  3,2
## Probs    1/45 2/45 1/15 4/45  1/9 2/15 7/45 8/45  1/5
```

Manipulating Joint Distributions

```
(A <- marginal(AB, 1))
```

```
## Random variable with 3 outcomes
##
## Outcomes     1     2     3
## Probs     2/15   1/3  8/15
```

```
(B <- marginal(AB, 2))
```

```
## Random variable with 3 outcomes
##
## Outcomes     0     1     2
## Probs     4/15   1/3   2/5
```

Conditional Distributions

```
independent(A, B)
```

```
## [1] FALSE
```

```
P(A == 1 | B == 1)
```

```
## [1] 0.1333333
```

```
P(A == 1 | B == 2)
```

```
## [1] 0.1666667
```

More Conditionals

```
P(A < B)
```

```
## [1] 0.06666667
```

```
E(A | B == 1)
```

```
## [1] 2.4
```

```
A | B == 1
```

```
## Random variable with 3 outcomes
##
## Outcomes     1     2     3
## Probs     2/15   1/3  8/15
```

Sum of Random Variables

We can obtain a new random variable by summing independent realizations of a random variable:

```
SofIID(X, n = 2, fractions = TRUE)
```

```
## Random variable with 11 outcomes
##
## Outcomes     2    3    4    5    6    7    8    9   10
## Probs     1/36 1/18 1/12  1/9 5/36  1/6 5/36  1/9 1/12 1/
```

More Convenient Notation

The + and * operators have been overloaded for convenience:

```
A * B
```

```
## Random variable with 9 outcomes
##
## Outcomes     1,0     1,1     1,2     2,0     2,1     2,2     3,
## Probs      8/225    2/45    4/75    4/45     1/9    2/15 32/22
```

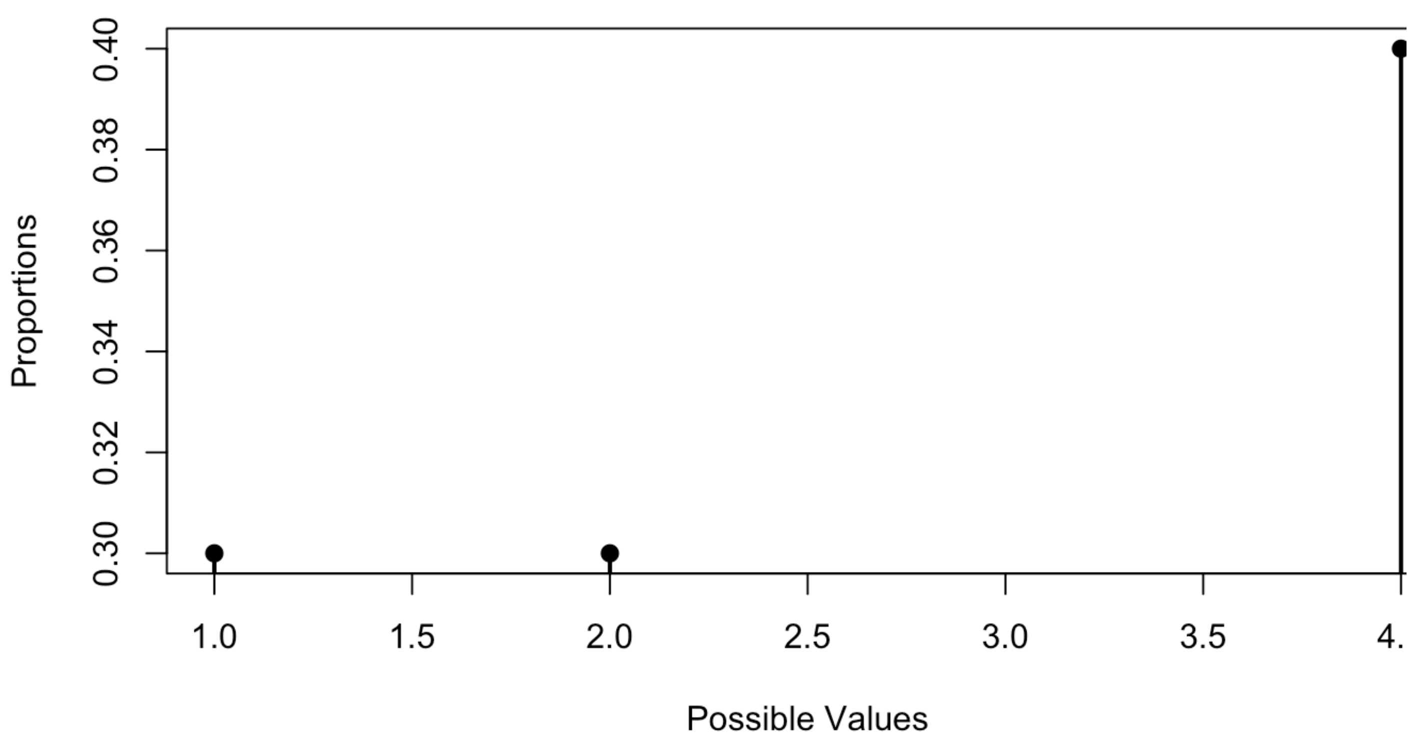
```
A + B
```

```
## Random variable with 5 outcomes
##
## Outcomes       1      2      3      4      5
## Probs      8/225   2/15 23/75 14/45 16/75
```

Simulations

We can simulate trials from any defined random variable:

```
X.sim <- rsim(X, 10)
plot(X.sim)
```



Extended Example: Playing Craps

Craps is a common dice game played in casinos. The game begins with what is called the "Come Out" roll, in which two fair dice are rolled.

- If a sum of seven or eleven is obtained, the player wins.
- If a sum of two, three, or twelve is obtained, the player loses.
- In all other cases, the roll obtained is declared the "Point" and the player rolls again in an attempt to obtain this same point value. If the player rolls the Point, they win, but if they roll a seven, they lose. Rolls continue until one of these two outcomes is achieved.

Playing Craps: The Roll

We begin by creating a random variable representing the result of a single roll (the sum of two fair dice):

```
(Roll <- RV(1:6) + RV(1:6))
```

```
## Random variable with 11 outcomes
##
## Outcomes     2    3    4    5    6    7    8    9   10
## Probs     1/36 1/18 1/12  1/9 5/36  1/6 5/36  1/9 1/12 1/
```

Results Conditioned on Game Ending After First Roll

```
P(Roll %in% c(7, 11) | Roll %in% c(7, 11, 2, 3, 12))
```

```
## [1] 0.6666667
```

```
P(Roll %in% c(2, 3, 12) | Roll %in% c(7, 11, 2, 3, 12))
```

```
## [1] 0.3333333
```

Probability of Winning Game Within Two Rolls

```
TwoRolls <- iid(Roll, 2)

First <- marginal(TwoRolls, 1)
Second <- marginal(TwoRolls, 2)

P(First %in% c(7, 11) %OR% (First %in% 4:10 %AND% (First ==
```

```
## [1] 0.2993827
```

Winning Craps

```
craps_game <- function(RV) {

    my.roll <- rsim(RV, 1)

    if (my.roll %in% c(7, 11)) { return(1) }
    else if (my.roll %in% c(2, 3, 12)) { return(0) }
    else {
        new.roll <- 0
        while (new.roll != my.roll & new.roll != 7) {
            new.roll <- rsim(RV, 1)
        }

        return(as.numeric(new.roll == my.roll))
    }
}
```

Winning Craps (continued)

```
sim.results <- replicate(100000, craps_game(Roll))
mean(sim.results)
```

```
## [1] 0.49195
```

Special Thanks

Dr. Andreas Buja, Wharton School, University of Pennsylvania

Dr. Heike Hofmann, Department of Statistics, Iowa State University

Thank You

Any questions?