Understanding probability

# Understanding probability

Let’s roll a die…

sample(x = 1:6, size = 5, replace = FALSE)

## [1] 6 3 1 5 2

sample(x = 1:6, size = 20, replace = TRUE)

## [1] 2 5 5 3 1 6 6 1 5 4 2 5 6 1 2 4 3 1 5 3

sample(x = 1:6, size = 20, replace = TRUE) # you get different numbers

## [1] 3 4 4 1 3 6 3 5 1 5 6 2 1 6 2 4 3 2 1 4

sample(x = 1:6, size = 20, replace = TRUE) # again

## [1] 5 1 6 3 2 1 6 3 3 4 6 5 4 4 5 4 4 2 6 1

How often do you get a 6?

RollDie <- function(n) sample(1:6, n, replace = TRUE)  
d1 <- RollDie(n = 50)  
sum(d1 == 6)

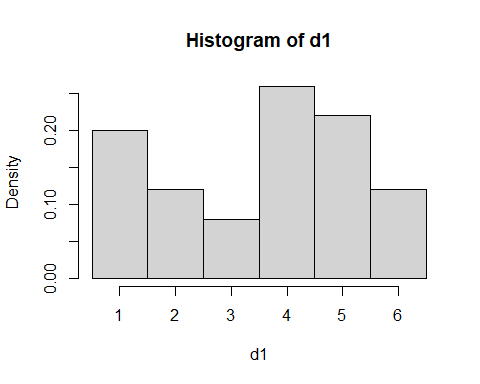
## [1] 6

sum(d1 == 6)/50

## [1] 0.12

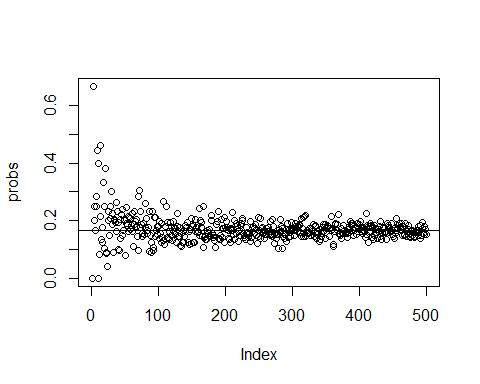
We can plot a relative histogram using:

hist(d1, probability = TRUE, breaks = seq(0.5,6.5,1))



Let’s simulate 500 rolls:

sims <- vector("list", 500)  
probs <- vector("numeric", 500)  
for (n in 1:500) {  
 sims[[n]] <- RollDie(n)  
 probs[n] <- sum(sims[[n]] == 6)/n  
}  
plot(probs)  
abline(h=1/6)



Please remember that our definition of probability is ‘long-run relative frequency’.  
If we repeat an experiment (like flipping a coin or rolling a dice) a large number of times and tabulate the outcomes, the relative frequencies will ‘converge’ to the probabilities of each outcome.  
This is the basic principles of Monte Carlo Simulations - Monte Carlo Simulation means using a computer to repeatedly carry out a random experiment and keeping track of the outcomes. That’s what we just did!! (<http://ditraglia.com/Econ103Public/Rtutorials/Rtutorial4.html#introduction>)

The probability of obtaining a 6 is a **marginal probability**. “The interesting thing about a marginal probability is that the term sounds complicated, but it’s actually the probability that we are most familiar with. Basically anytime you are in interested in a single event irrespective of any other event (i.e. “marginalizing the other event”), then it is a marginal probability. For instance, the probability of a coin flip giving a head is considered a marginal probability because we aren’t considering any other events.” (<http://tinyheero.github.io/2016/03/20/basic-prob.html>)

sample(x=c("testa","croce"), size = 5, replace = TRUE)

## [1] "testa" "croce" "testa" "testa" "testa"

E.g. what is the probability of capturing an adult snow vole?

Now we can roll two dice:

sample(1:6, size = 1, replace = TRUE)

## [1] 2

sample(1:6, size = 1, replace = TRUE)

## [1] 4

How often do we get the same number if we roll 2 dice?

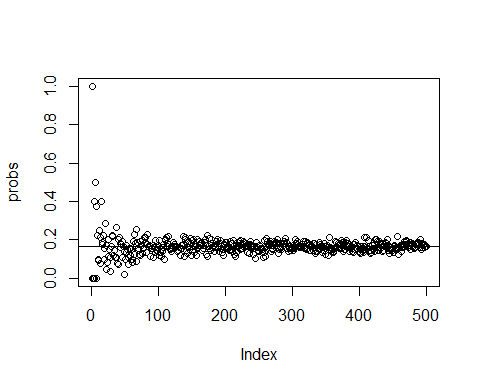
roll1 = NULL #This initializes our variable - i.e. it creates a spot in memory for it. We need to do this for any vector, table, matrix, dataframe, but not for single numbers  
roll2 = NULL  
for (i in 1:100) {  
 roll1[i] = RollDie(1)  
 roll2[i] = RollDie(1)  
}  
# We can ask how many doubles we came up with:  
sum(roll1 == roll2)

## [1] 18

# relative frequency  
sum(roll1 == roll2)/100

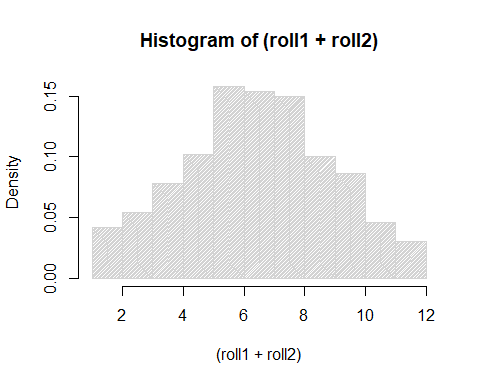
## [1] 0.18

roll1 <- vector("list", 500)  
roll2 <- vector("list", 500)  
probs <- vector("numeric", 500)  
for (n in 1:500) {  
 roll1[[n]] <- RollDie(n)  
 roll2[[n]] <- RollDie(n)  
 probs[n] <- sum(roll1[[n]] == roll2[[n]])/n  
}  
plot(probs)  
abline(h=1/6)

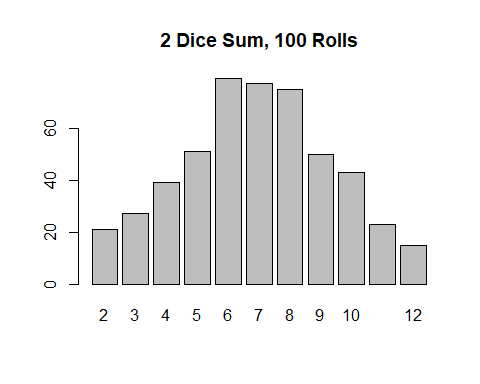


Now we can ask for the sum of the two rolls:

n = 500  
roll1 = NULL #This initializes our variable - i.e. it creates a spot in memory for it. We need to do this for any vector, table, matrix, dataframe, but not for single numbers  
roll2 = NULL  
for (i in 1:n) {  
 roll1[i] = RollDie(1)  
 roll2[i] = RollDie(1)  
}  
hist((roll1 + roll2), density = 100, breaks = 1:12, prob = T)



barplot(table(roll1 + roll2), main = "2 Dice Sum, 100 Rolls") #this works better for this case



rolls <- roll1 + roll2  
sum(rolls == 7)

## [1] 77

sum(rolls == 7)/n

## [1] 0.154

# Joint probability

**Joint probability**: probability of the joint occurrence of two independent events. *i.e.*, the probability of two different events occurring at the same time. If two events (A and B) are **independent**, then P(A and B) = P(A) \* P(B).

Again, we roll 2 dice.  
How often do we get a 6 in both rolls?

n = 1500  
roll1 = NULL #This initializes our variable - i.e. it creates a spot in memory for it. We need to do this for any vector, table, matrix, dataframe, but not for single numbers  
roll2 = NULL  
for (i in 1:n) {  
 roll1[i] = RollDie(1)  
 roll2[i] = RollDie(1)  
}  
# We can ask how many times this happen:  
sum(roll1 == 6 & roll2 == 6)

## [1] 35

# the relative frequency is:  
sum(roll1 == 6 & roll2 == 6)/n

## [1] 0.02333333

Repeat the previous code, changing n (50, 100, 200, 500). Find the relative frequencies.

Compare with:

(1/6)\*(1/6)

## [1] 0.02777778

E.g. you caught a snow vole: what is the probability that it is an adult and that it is female? The joint probability can be calculated by taking the proportion of times a specific combination of age and sex occurs, divided by the total number of age-sex classes combinations (but see below for this specific example - joint distribution of conditionally dependent variables).

# Mutually exclusive events

If two events (A and B) are **mutually exclusive**, then P(A and B) = P(A) + P(B).

When you roll a die, you can roll a 5 OR a 6 (the odds are 1 out of 6 for each event) and the sum of either event happening is the sum of both probabilities:  
P(rolling a 5 or rolling a 6) = P(rolling a 5) + P(rolling a 6)  
P(rolling a 5 or rolling a 6) = 1/6 + 1/6 = 2/6 = 1/3.

Think about the previous example of rolling 2 dice. We asked how many times we got the same number from them:  
P(rolling a 1 and rolling a 1) = 1/6 \* 1/6 = 0.02777778  
P(rolling a 2 and rolling a 2) = 1/6 \* 1/6 = 0.02777778  
P(rolling a 3 and rolling a 3) = 1/6 \* 1/6 = 0.02777778  
P(rolling a 4 and rolling a 4) = 1/6 \* 1/6 = 0.02777778  
P(rolling a 5 and rolling a 5) = 1/6 \* 1/6 = 0.02777778  
P(rolling a 6 and rolling a 6) = 1/6 \* 1/6 = 0.02777778

These events are mutually exclusive, we can not get 1 in both rolls and 2 in both rolls. Try to sum up these probabilities:

0.02777778+0.02777778+0.02777778+0.02777778+0.02777778+0.02777778

## [1] 0.1666667

# or  
0.02777778\*6

## [1] 0.1666667

# Conditional probability

A conditional probability is the probability of an event X occurring when a secondary event Y is true. Mathematically, it is represented as P(X | Y). This is read as “probability of X given/conditioned on Y”.

# Think about biological data…

<http://tinyheero.github.io/2016/03/20/basic-prob.html>

What is the probability of capturing an adult snow vole? (marginal)  
What is the probability of capturing a female snow vole? (marginal)  
What is the probability of capturing a female adult snow vole? (joint)  
What is the probability of capturing a female, given that it is an adult? (conditional)

y <- read.csv("data/captures.csv",sep=";")   
y <- na.omit(y[,c("age", "sex")])

The marginal probability of capturing an adult snow vole is:

library(dplyr)

## Warning: il pacchetto 'dplyr' è stato creato con R versione 4.2.2

##   
## Caricamento pacchetto: 'dplyr'

## I seguenti oggetti sono mascherati da 'package:stats':  
##   
## filter, lag

## I seguenti oggetti sono mascherati da 'package:base':  
##   
## intersect, setdiff, setequal, union

age.marginal.df <-   
 y %>%   
 group\_by(age) %>%  
 summarise(n = n()) %>%  
 ungroup() %>%  
 mutate(prop = n/sum(n))  
age.marginal.df

## # A tibble: 2 × 3  
## age n prop  
## <chr> <int> <dbl>  
## 1 A 79 0.849  
## 2 G 14 0.151

And the probability of capturing a female snow vole is:

sex.marginal.df <-   
 y %>%   
 group\_by(sex) %>%  
 summarise(n = n()) %>%  
 ungroup() %>%   
 mutate(prop = n/sum(n))  
sex.marginal.df

## # A tibble: 2 × 3  
## sex n prop  
## <chr> <int> <dbl>  
## 1 F 33 0.355  
## 2 M 60 0.645

Joint probabilities can be calculated by taking the proportion of times a specific sex-age combination occurs, divided by total number of all sex-age combinations (i.e. frequency):

joint.df <-   
 y %>%   
 group\_by(age, sex) %>%  
 summarise(n = n()) %>%  
 ungroup() %>%   
 mutate(prop = n/sum(n))

## `summarise()` has grouped output by 'age'. You can override using the `.groups`  
## argument.

joint.df

## # A tibble: 4 × 4  
## age sex n prop  
## <chr> <chr> <int> <dbl>  
## 1 A F 32 0.344   
## 2 A M 47 0.505   
## 3 G F 1 0.0108  
## 4 G M 13 0.140

What is the probability of capturing a female, given that it is an adult? (conditional)

joint.prob <-   
 joint.df %>%   
 filter(age == "A", sex == "F") %>%   
 .$prop  
joint.prob

## [1] 0.344086

marg.prob <-   
 age.marginal.df %>%   
 filter(age == "A") %>%   
 .$prop  
marg.prob

## [1] 0.8494624

cond.prob <- joint.prob/marg.prob  
cond.prob

## [1] 0.4050633

i.e. 32 adult females out of a total of 79 adults.

Please note that if variables are not conditionally independent, their joint distribution is not equal to P(A)*P(B), but rather by P(B)*P(A | B). Check this by multiplying the conditional probability obtained here by the marginal probability of being an adult (B). Compare with the joint.df table.