Lecture2\_BD

Sabrina Streipert

27 February 2019

knitr::opts\_chunk$set(echo = TRUE)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

### Chapter 2: The Birthday Probability

If is the number of students in a class, then the probability that at least 2 students share the same birthday is

So, let us calculate the probability that none of the students share a birthday and then we just substract that from 1.

Frist, we create a class:

Classsize <- 2:70  
BDex <- Classsize %>% data.frame()   
head(BDex)

## .  
## 1 2  
## 2 3  
## 3 4  
## 4 5  
## 5 6  
## 6 7

Let us change the column name to the Number:

BDex <- BDex %>% rename(Number = '.')  
head(BDex)

## Number  
## 1 2  
## 2 3  
## 3 4  
## 4 5  
## 5 6  
## 6 7

Now, let us calculate the probability that none of the students share a birthday (= complement) by adding a column:

BDex1 <- BDex %>% mutate(Prob\_compl = factorial(365)/(factorial(365-Number)\*(365\*\*Number)))   
head(BDex1)

## Number Prob\_compl  
## 1 2 NaN  
## 2 3 NaN  
## 3 4 NaN  
## 4 5 NaN  
## 5 6 NaN  
## 6 7 NaN

Oh NO, the values are too large. What now? -> One option: seek online help -> get the inforamtion of lfactorial, which is the logarithm of the factorial value…. If you want some more information about it, use the command: ?lfactorial

Can I do that?

BDex2 <- BDex %>% mutate(lprob\_compl = lfactorial(365)/(lfactorial(365-Number)\*log(365\*\*Number)))

No!!!! recall that

So, we need to be careful with these operations!

We recall some logarithm operations:

And even one more step to get

Let’s try that

BDex3 <- BDex %>% mutate(lprob\_compl = lfactorial(365)-lfactorial(365-Number)- Number\*log(365))  
head(BDex3)

## Number lprob\_compl  
## 1 2 -0.002743486  
## 2 3 -0.008238005  
## 3 4 -0.016491147  
## 4 5 -0.027510542  
## 5 6 -0.041303864  
## 6 7 -0.057878829

Oh no, why are there negative values???

Don’t forget to transform the probability back via the inverse of the logarithm function:

BDex3 <- BDex3 %>% mutate(prob\_compl = exp(lprob\_compl))  
head(BDex3)

## Number lprob\_compl prob\_compl  
## 1 2 -0.002743486 0.9972603  
## 2 3 -0.008238005 0.9917958  
## 3 4 -0.016491147 0.9836441  
## 4 5 -0.027510542 0.9728644  
## 5 6 -0.041303864 0.9595375  
## 6 7 -0.057878829 0.9437643

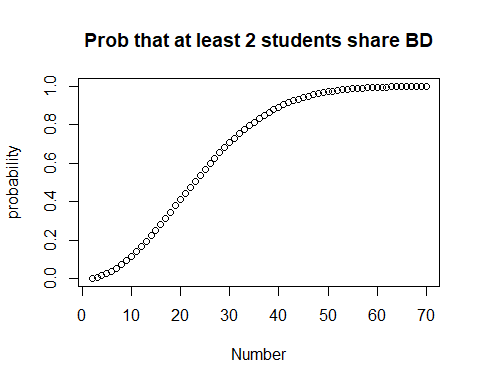
Since we are interested in the probability that at least two students share the same birthday, we have to subtract the complement from 1, that is:

BDex4 <- BDex3 %>% mutate(prob = 1-prob\_compl)  
head(BDex4)

## Number lprob\_compl prob\_compl prob  
## 1 2 -0.002743486 0.9972603 0.002739726  
## 2 3 -0.008238005 0.9917958 0.008204166  
## 3 4 -0.016491147 0.9836441 0.016355912  
## 4 5 -0.027510542 0.9728644 0.027135574  
## 5 6 -0.041303864 0.9595375 0.040462484  
## 6 7 -0.057878829 0.9437643 0.056235703

Now, let us plot that

plot(BDex4$Number, BDex4$prob, xlab = 'Number', ylab = 'probability', main = 'Prob that at least 2 students share BD')



## Obtain an approximation of this probability through sampling (Monte Carlo Method):

We create several classes, say 10000, and count how many of these classes have students that share the same birthday. For that let us fix a class room size, say of 30 students.

# pick the year 1998 (or any other year with 365 days)  
# make a list of all dates:  
Possible\_Dates <- seq( as.Date("1998-01-01"), as.Date("1998-12-31"), by="+1 day")  
#check  
head(Possible\_Dates)

## [1] "1998-01-01" "1998-01-02" "1998-01-03" "1998-01-04" "1998-01-05"  
## [6] "1998-01-06"

tail(Possible\_Dates)

## [1] "1998-12-26" "1998-12-27" "1998-12-28" "1998-12-29" "1998-12-30"  
## [6] "1998-12-31"

Now, we take samples from this list

Nr\_class <- 10000  
Nr\_students <- 30  
  
  
counter <- 0  
for (i in 1:Nr\_class){  
   
 Class <- 2:Nr\_students %>% data.frame() %>% rename(StudentID = '.' )   
 Class$BD <- sample(Possible\_Dates, (Nr\_students-1),replace=TRUE)  
 BD\_counter <- Class %>% group\_by(BD) %>% summarize(Countpergroup = n())  
   
 counter <- counter + max(BD\_counter$Countpergroup>1)  
}  
  
prob <- counter/Nr\_class  
  
print(prob)

## [1] 0.6819

Compared to the theoretical value:

Theor <- filter(BDex4, Number == Nr\_students)  
print(Theor$prob)

## [1] 0.7063162