Homework #1

Andy Hansen

Due Thursday, September 1 by 5pm on your GitHub repositories

**Instructions:** The entirety of this assignment must be submitted as an R Markdown file (.Rmd) on your GitHub repository. Use the .Rmd note handout files and the R Markdown cheat sheet as guidelines. You are encouraged to save this HW1.Rmd file and fill in the questions with your answers, then submit. **I should be able to knit your .Rmd file and compile your code myself, so make sure you do some bug checks before submitting!** (I.e., knit the document yourself a couple times and search for errors.)

Consider Example 2 in the notes. 2 dice are rolled, one red and one white. Let be the random variable that denotes the maximum value of the two rolls. We will use simulation to find the mean and variance of , and then verify that our simulated results match what we would expect theoretically.

**Theoretical section**

## Warning: package 'ggplot2' was built under R version 3.2.5

1. (3pts) Define the pmf, find , , and . Show all your work.

# Define the PMF

y <- 1:6  
py <- c(1/36,3/36,5/36,7/36,9/36,11/36)  
data.frame(HighestRoll=y,Probability=py,Value=y\*py)

## HighestRoll Probability Value  
## 1 1 0.02777778 0.02777778  
## 2 2 0.08333333 0.16666667  
## 3 3 0.13888889 0.41666667  
## 4 4 0.19444444 0.77777778  
## 5 5 0.25000000 1.25000000  
## 6 6 0.30555556 1.83333333

# Expected Value

#Expected Value (mean)  
mu <- sum(y\*py)  
mu

## [1] 4.472222

# Variance

EY2 <- sum(y^2\*py)  
EY2

## [1] 21.97222

sigma2 <- EY2-mu^2  
sigma2

## [1] 1.971451

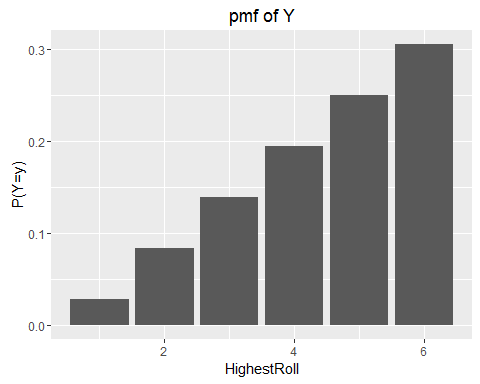
# Standard Deviation

SD <- sqrt(sigma2)  
SD

## [1] 1.404084

1. (2pts) Use ggplot() to plot the pmf; see Handout 1 notes for an example.

dd <- data.frame(HighestRoll=y,probs = py)  
ggplot(aes(x=HighestRoll,y=probs),data=dd) + geom\_bar(stat='identity') +   
 ylab('P(Y=y)') + ggtitle('pmf of Y')



1. (2pts) Consider the random variable . What is and ? Show all work.

# Expected Value

yz <- c(3,5,7,9,11,13)  
pyz <- c(1/36,3/36,5/36,7/36,9/36,11/36)  
muz <- sum(yz\*pyz)  
muz

## [1] 9.944444

# Variance

Ey2z <- sum(yz^2\*pyz)  
sigma2z <- Ey2z-muz^2  
sigma2z

## [1] 7.885802

**Simulation section**

Write a function called one.Y that simulates rolling two dice and returns the maximum roll. Try the function out a few times and include the results of these test-runs in your R Markdown output. I have written some code below to get you started; each line of "pseudo-code" should be repaced with actual code:

one.y <- function(){  
oneRollSampleSpace <- c(1,2,3,4,5,6)  
RedDie1 <- sample(oneRollSampleSpace,1,replace=TRUE)  
WhiteDie1 <- sample(oneRollSampleSpace,1,replace=TRUE)  
maxRoll <- max(RedDie1,WhiteDie1)  
return(maxRoll)  
}  
one.y()

## [1] 2

Each of the following can be answered with 1-2 lines of R code (and corresponding output, of course)

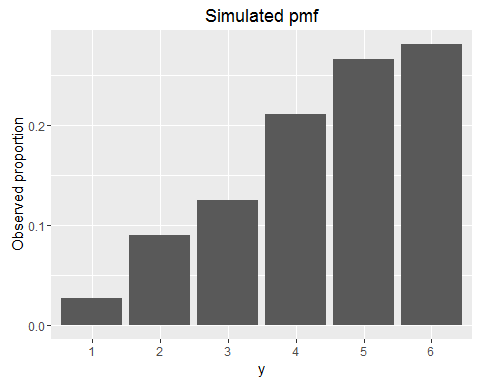
1. (2pts) Use replicate() to simulate the results of 1000 pairs of rolls. These are 1000 realizations of the random variable . Save the 1000 realizations in an object called many.Y.

# Replication

many.y <- replicate(1000,one.y())

1. (2pts) Use ggplot() to create the empirical (i.e., observed) pmf of your simulation. See Handout 1 for example R code. How does it compare with your theoretical pmf?

#put results into data frame:  
df <- data.frame(x=as.factor(many.y))  
ggplot(aes(x=as.factor(many.y)),data=df) + geom\_bar(aes(y=(..count..)/(sum(..count..)))) +  
 ylab('Observed proportion') + xlab('y') + ggtitle('Simulated pmf')

 How does it compare with with your theoretical pmf?

The results are almost identical.Replicating this 1000 times gets the variance and means close to the theoretical pmf.

1. (1pt) What is the mean of the 1000 realizations?

# Mean

MEAN <-mean(many.y)  
MEAN

## [1] 4.442

7.(1pt) What is the variance of the 1000 realizations?

# Variance

VAR <- var(many.y)  
VAR

## [1] 1.924561

1. (1pt) What is the standard deviation of the 1000 realizations?

# Standard Deviation

SD1 <- sqrt(VAR)  
SD1

## [1] 1.387285

1. (1pt) Create a new object called many.Z that creates 1000 realizations of .

# Create and Replication

one.Z <- function(){  
oneRollSampleSpaceZ <- c(1,2,3,4,5,6)  
RedDie1Z <- sample(oneRollSampleSpaceZ,1,replace=TRUE)  
WhiteDie1Z <- sample(oneRollSampleSpaceZ,1,replace=TRUE)  
maxRollZ <- max(RedDie1Z,WhiteDie1Z)  
return((2\*maxRollZ)+1)  
}  
one.Z()

## [1] 11

#Replicate it 1000 times  
many.Z <- replicate(1000,one.Z())

1. (1pt) What is the mean of ?

# Mean

MEANZ <- mean(many.Z)  
MEANZ

## [1] 9.906

1. (1pt) What is the variance of ?

# Variance

VARZ <- var(many.Z)  
VARZ

## [1] 8.031195

1. (1pt) Note that your simulated results should be similar to the theoretical quantities; if they aren't, re-check your R code! What is the reason for any differences?

The is little difference between are simulation and the actual statistics because we only replicated this 1000 times. This is not going to be exactly the same values as the theoretical values. The higher number of simulations we give closer numbers to the theoretical values.