Stat 450

Le Tang (R Assignment 1)

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 Y be the random variable that denotes the maximum value of the two rolls. We will use simulation to find the mean and variance of Y, and then verify that our simulated results match what we would expect theoretically.

Theoretical section

1. (3pts) Define the pmf, find $\mu = E(Y)$, $\sigma^2 = Var(Y)$, and $\sigma = SD(Y)$. Show all your work.

```
y <-1:6
y
```

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

```
py <-c(1/36,3/36,5/36,7/36,9/36,11/36)
py
```

```
## [1] 0.02777778 0.08333333 0.13888889 0.19444444 0.25000000 0.30555556
```

Below are the μ, σ^2, and σ, respectively:

```
mu<-sum(y*py)
mu
```

```
## [1] 4.472222
```

```
EY2<-sum(y^2*py)
sigma2<-EY2-mu^2
sigma2
```

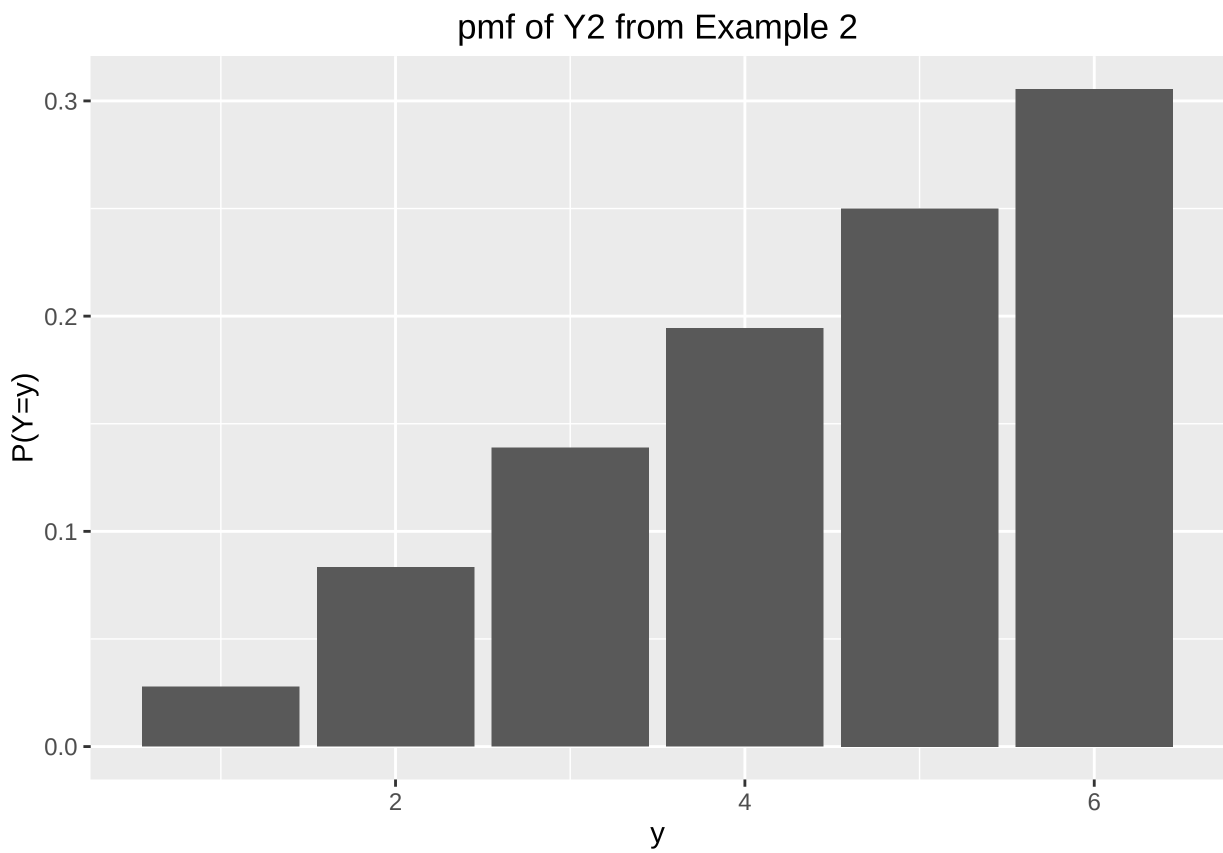
```
## [1] 1.971451
```

```
sigma<-sqrt(sigma2)
sigma
```

```
## [1] 1.404084
```

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

```
library(ggplot2)
dd <-data.frame(y=y, probs=py)
ggplot(aes(x=y, y=probs), data=dd) + geom_bar(stat='identity') + ylab('P(Y=y)') +
ggtitle('pmf of Y2 from Example 2')
```



3. (2pts) Consider the random variable $Z = 2Y + 1$. What is $E(Z)$ and $Var(Z)$? Show all work.

```
muZ<-2*mu+1
muZ
```

```
## [1] 9.944444
```

```
varZ<-2^2*sigma2
varZ
```

```
## [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() {
  sample.space <- c(1,2,3,4,5,6)
```

```
  red.roll <- sample(sample.space,1)
  white.roll <- sample(sample.space,1)
  num.rolled <- c(red.roll,white.roll)
  max.final <- max(num.rolled)
  return(max.final)
  }
one.Y()
```

```
## [1] 2
```

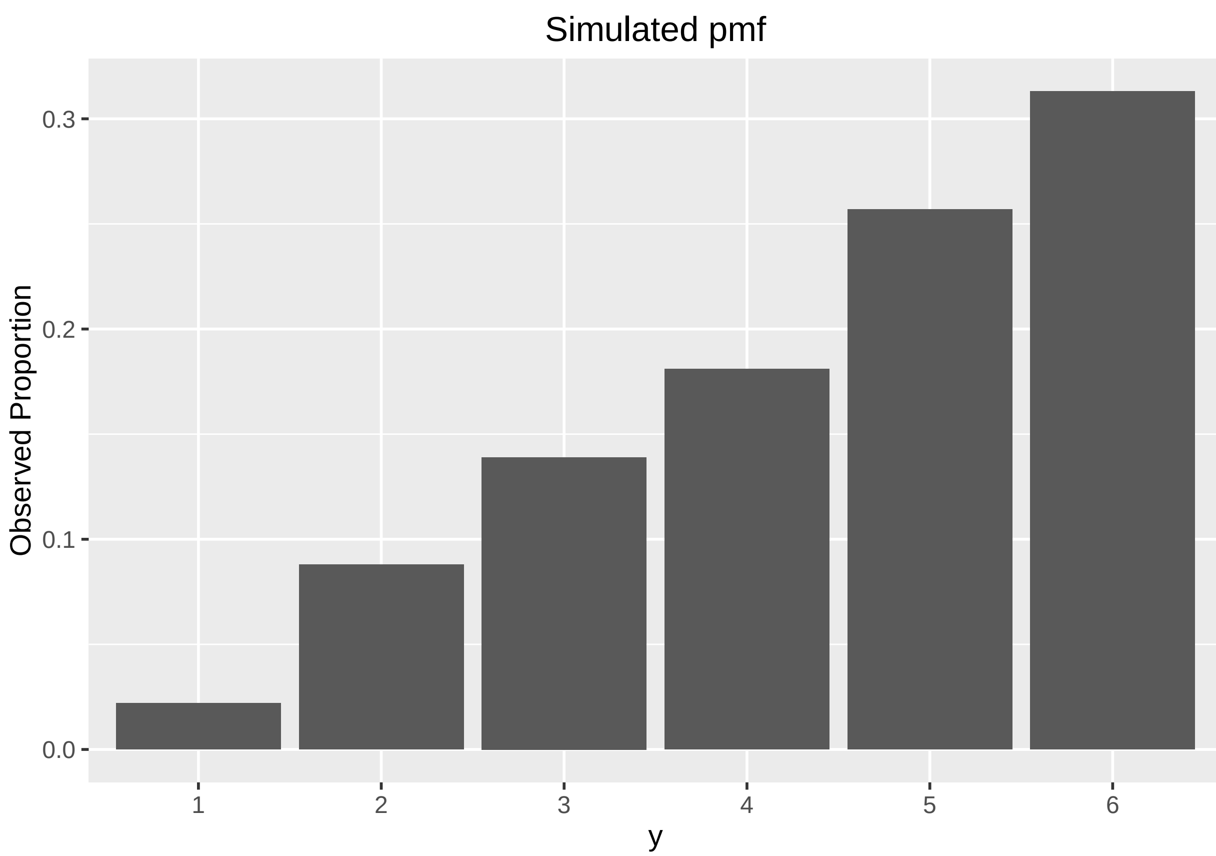
Each of the following can be answered with 1-2 lines of R code (and corresponding output, of course)

4. (2pts) Use `replicate()` to simulate the results of 1000 pairs of rolls. These are 1000 realizations of the random variable Y. Save the 1000 realizations in an object called `many.Y`.

```
set.seed(2222)
many.Y <-replicate(1000,one.Y())
```

5. (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?

```
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')
```



6. (1pt) What is the mean of the 1000 realizations?

```r
mean(many.Y)
```

```
## [1] 4.502
```

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

```r
var(many.Y)
```

```
## [1] 1.947944
```

8. (1pt) What is the standard deviation of the 1000 realizations?

```r
sd(many.Y)
```

```
## [1] 1.395688
```

9. (1pt) Create a new object called `many.Z` that creates 1000 realizations of Z.

```r
many.Z <- 2*many.Y+1
```

10. (1pt) What is the mean of Z?

```r
mean(many.Z)
```

```
## [1] 10.004
```

11. (1pt) What is the variance of Z?

```r
var(many.Z)
```

```
## [1] 7.791776
```

12. (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?

Besides that I am not quite sure how "set.seed" influences our results, the differences exist because of relatively small sample size (we only simulated 1000 pair of rolls). If we let the sample size, n, goes to infinity, our observed data will be closer enough to what we initially expected.