Quiz 2

Nick Strayer, Alex Sundermann, Sara Savage, Lingjun Fu

15 September, 2015

Question 1:

	Nick	Alex	Sara	Lingjun
Wins	5	4	5	5
Loses	5	6	5	5
Winnings	$0	-$2	$0	$0

Discussion:

Three of the four people in our group broke even after ten games of Craps with a total of $0 each. Alex had worse luck, resulting in a net winning of -$2. This small sample demonstrates that the probability of winning and losing in Craps is roughly even with a suggestion of a slightly negative expected value.

Question 2:

The pmf is the addition of the probabilities of each way to roll a number. For example, a 4 can be rolled when:

Die 1	Die 2
1	3
2	2
3	1

The probability of rolling each number per die is 1/6. Therefore, the pmf for 4 is:

$$\frac{1}{6} \cdot \frac{1}{6} + \frac{1}{6} \cdot \frac{1}{6} + \frac{1}{6} \cdot \frac{1}{6} = \frac{3}{36}$$

Score	2	3	4	5	6	7	8	9	10	11	12
Prob	1/36	2/36	3/36	4/36	5/36	6/36	5/36	4/36	3/36	2/36	1/36

Question 3:

$$P(\text{Rolling Natural}) = P(7) + P(11) = \frac{6}{36} + \frac{2}{36} = \frac{8}{36} = \boxed{\frac{2}{9}}$$

Question 4:

$$\begin{aligned} P(W|\text{roll } 6) &= ? \\ P(\text{first roll}) &= \frac{5}{36} \\ P(\text{2nd}) &= \frac{5}{36} \cdot \frac{25}{36} \\ P(\text{3rd}) &= \frac{5}{36} \cdot (\frac{25}{36})^2 \\ &\dots \\ P(\text{nth}) &= \frac{5}{36} \cdot (\frac{25}{36})^{n-1} \end{aligned}$$

So the sum of these terms gives us $P(\text{winning} \mid \text{rolling a 6 on the first roll})$.

$$\sum_{n=1}^{\infty} \frac{5}{36} \cdot (\frac{25}{36})^{n-1}$$

This is a geometric series which converges.

$$\begin{aligned} P(\text{W} \mid \text{Roll 6}) &= \lim_{n \to \infty} \frac{5}{36}[(1-(\frac{25}{36})^n)/(1-\frac{25}{36}))] \\ &= \frac{5}{36}[(1)/(\frac{11}{36}))] \\ &\boxed{= \frac{5}{11}} \end{aligned}$$

Question 5:

$$\begin{aligned} E[W|\text{roll 6 first}] &= (1) \cdot \frac{5}{11} + (-1) \cdot \frac{6}{11} = -\frac{1}{11} \\ E[W^2|\text{roll 6 first}] &= (1^2) \cdot \frac{5}{11} + (-1^2) \cdot \frac{6}{11} = 1 \\ V[W|\text{roll 6 first}] &= E[W^2|\text{roll 6 first}] - E[W|\text{roll 6 first}]^2 = 1 - (-\frac{1}{11})^2 = \frac{120}{121} \\ SD[W|\text{roll 6 first}] &= \sqrt{\frac{120}{121}} \end{aligned}$$

Question 6:

Let A = winning and B = rolling a 6. Then it follows that P(winning by rolling a 6) is:

$$
\begin{aligned}
P(A \cap B) &= P(A|B)P(B) \\
&= \frac{5}{11} \cdot \frac{5}{36} \\
&= \frac{25}{396} = 0.0631
\end{aligned}
$$

It differs from question 4 in that this is a probability of intersection while question 4 is a conditional probability.

Question 7:

The probability of winning can be calculated as the sum of its intersections with different conditions: $P(W) = \sum P(W \cap i)$. Here, i is the total number of the first roll which ranges from 2 to 12. It is easy to see that: $P(W \cap 2) = (W \cap 3) = (W \cap 12) = 0$, $P(W \cap 7) = \frac{1}{6}$, $P(W \cap 11) = \frac{2}{36}$, and $P(W \cap 6) = \frac{25}{396}$.

In general, when i is not equal to 2, 3, 7, 11, or 12 we have:

$$
\begin{aligned}
P(W \cap i) &= P(W|i)P(i) \\
&= P(i) \cdot \sum_{n=1}^{\infty} P(i) \cdot (1 - P(7) - P(i))^{n-1} \\
&= \frac{P(i)^2}{P(7) + P(i)}
\end{aligned}
$$

A simple check using i equals 6 shows our formula agrees with the answer we calculated in question 6. Therefore, the probability of winning overall is $P(W) = \sum P(W \cap i)$, for i ranging from 2-12:

$$
P(\mathrm{W}) = \frac{244}{495} = 0.4929
$$

We then used the probability of W to find the expected value for W:

$$
\begin{aligned}
E[W] &= 1 \cdot P(W) - 1 \cdot P(L) \\
&= 1 \cdot \frac{244}{495} \cdot +(-1)(1 - \frac{244}{495}) \\
&= \frac{244 - 251}{495} \\
&= \frac{-7}{495} = -0.0141
\end{aligned}
$$

Calculation for the variance of W:

$$
\begin{aligned}
V[W] &= E[W^2] - E[W]^2 \\
&= 1 - (\frac{-7}{495})^2 \\
&= \frac{244,976}{245,025} = 0.9998
\end{aligned}
$$

Calcuation for the standard deviation of W:

$$
\begin{aligned}
SD[W] &= \sqrt{V[W]]} \\
&= \sqrt{\frac{244,976}{245,025}} = 0.9999
\end{aligned}
$$

Question 8:

Instead of having 36 potential outcomes with two six-sided die, we now have 32 potential outcomes using one 4-sided die and one 8-sided die. The sample space of possible sum still ranges from 2 to 12, but the probability of arriving at an individual sum is different. Most notably, the probabilities of rolling a total of 5, 6, 7 8, or 9 with one 4- and one 8-sided die all equal 4/32. Consequently, the pdf of the possible events in Dragon Kraps is still symmetrical, but the center of the pdf curve is a plateau instead of a hill. (See table of the pdf below).

Score	Normal Craps	Dragon Kraps
2	1/36	1/32
3	2/36	2/32
4	3/36	3/32
5	4/36	4/32
6	5/36	4/32
7	6/36	4/32
8	5/36	4/32
9	4/36	4/32
10	3/36	3/32
11	2/36	2/32
12	1/36	1/32

Table: Pdf for single roll in Normal Craps and Dragon Kraps In Dragon Kraps, the probability of throwing a natural is 0.1875 while throwing craps is 0.125. In regular Craps, the probability of throwing a natural is 0.22 while throwing craps is 0.11.

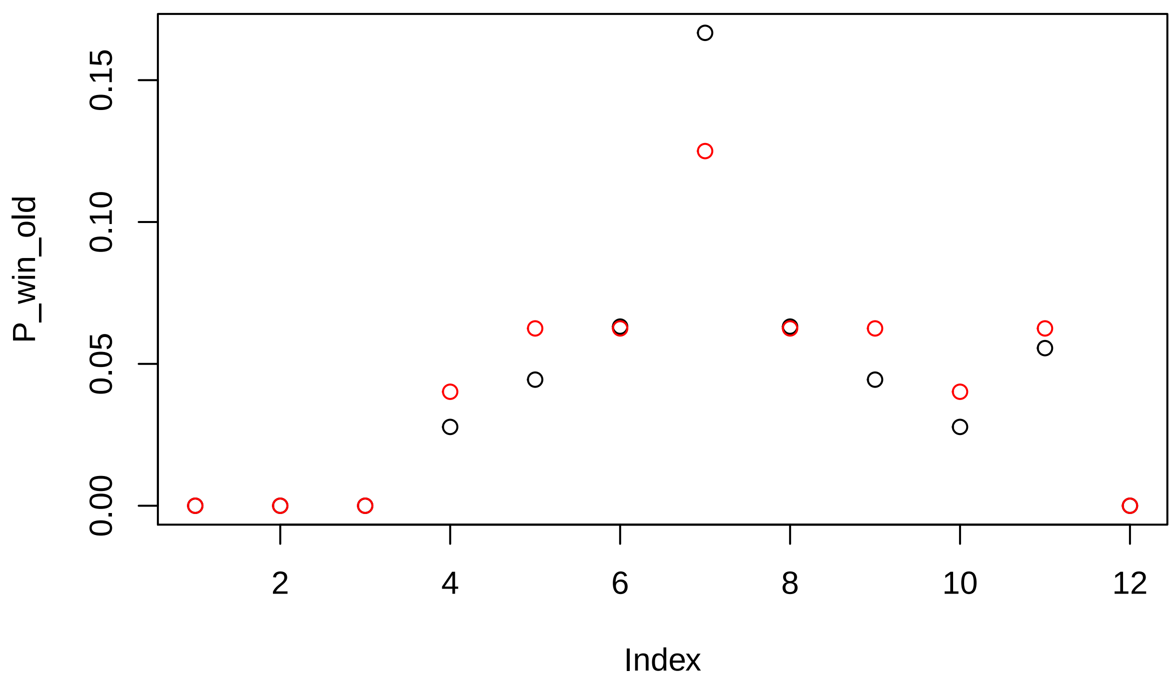
As aforementioned, the pdf of Dragon Kraps plateaus for rolls summing to 5, 6, 7, 8, and 9. This flattening of the distribution is reflected in the probability of winning the game given that the value i is rolled on the first turn. To compare this difference, the probability of winning given a first roll of i generated through calculations and through simulation is shown below for both games:

i	Normal Craps Calculated	Normal Craps Simulated	Dragon Kraps Calculated	Dragon Kraps Simulated
2	0.0000	0.0000	0.0000	0.0000
3	0.0000	0.0000	0.0000	0.0000
4	0.0278	0.0282	0.0402	0.0398
5	0.0444	0.0449	0.0625	0.0632
6	0.0631	0.0624	0.0625	0.0626
7	0.1667	0.1680	0.1250	0.1236
8	0.0631	0.0630	0.0625	0.0625
9	0.0444	0.0435	0.0625	0.0623
10	0.0278	0.0273	0.0402	0.0399
11	0.0556	0.0556	0.0625	0.0623

i	Normal Craps Calculated	Normal Craps Simulated	Dragon Kraps Calculated	Dragon Kraps Simulated
12	0.0000	0.0000	0.0000	0.0000
Overall	0.4929	0.4930	0.5179	0.5162

Table: P(winning with X) Below is a graphical representation of the probability of winning given first roll equals i. Probabilities for Normal Craps shown in black, probabilities for Dragon Kraps shown in red.

```r
p_old<-c(0, 1/36, 2/36, 3/36, 4/36, 5/36, 6/36, 5/36, 4/36, 3/36, 2/36, 1/36)
p_dragon<-c(0, 1/32, 2/32, 3/32, 4/32, 4/32, 4/32, 4/32, 4/32, 3/32, 2/32, 1/32)
#p_old is the pmf for old game rolling a total of (1 to 12)
#p_dragon is the pmf for dragon game rolling a total of (1 to 12)
P_win_old<-rep(0, 12)
P_win_old[7]<-1/6
P_win_old[11]<-2/36
for(i in 1:11){
    if(i != 2 & i != 3 & i != 7 & i != 11 & i != 12){
        P_win_old[i] = p_old[i]^2/(p_old[i] + p_old[7])
    }
}

P_win_dragon<-rep(0, 12)
P_win_dragon[7]<-4/32
P_win_dragon[11]<-2/32
for(i in 1:11){
    if(i != 2 & i != 3 & i != 7 & i != 11 & i != 12){
        P_win_dragon[i] = p_dragon[i]^2/(p_dragon[i] + p_dragon[7])
    }
}
#P_win_old is the probability of winning old game by rolling a total of (1 to 12)
#P_win_dragon is the probability of winning dragon game rolling a total of (1 to 12)


plot(P_win_old)
points(P_win_dragon, col = "red")
```



The table demenstrates that the probability of winning in Dragon craps is 0.5179, compared to the probability of winning in Normal craps 0.4929. Therefore, if we were gambling, the expected value of winning would be positive 0.0358 compared to the a negative expected value in Normal Craps of -0.0141. This means that when playing Dragon Kraps, one can expect to make a profit on average of $0.0358 per game. In summary, play Dragon craps if the casino is offering.

Question 9

Setup the craps code

```
die = function(numSides = 6){ return(sample(1:numSides, 1)) }

# Let's play some craps
playCraps = function(dragonStyle = F, roll1 = NA){
  lastRoll = NULL #Initiate a holder for the last roll variable. This is needed for problem 6

  #Set up some functions first --------------------------------------
  roll = function(){
    if(dragonStyle){
      d1 = die(numSides = 4)
      d2 = die(numSides = 8)
```

```r
    } else {
      d1 = die()
      d2 = die()
    }
    lastRoll <<- d1+d2 #using the double less thans allows this to write to a global variable

    return(list(one = d1, two = d2, sum = lastRoll))
  }

  pointsRoll = function(valToMatch){
    newRoll = roll()$sum #roll again

    if(newRoll == valToMatch){
      return("w")
    } else if(newRoll == 7){
      return("l")
    } else {
      pointsRoll(valToMatch)
    }
  }

## ------------------------------------------------------------------------------------
  # first roll, if we didn't win then we record the sum of the dice and use it for the later rolls
  if(is.na(roll1)){ #If we didn't specify our first value then generate one
    roll1 = roll()$sum
  }

  # If it is a 7 or 11 then we win,
  if(roll1 %in% c(7,11) ){
    return(list(res = "w", last = lastRoll))

  # if we roll 2,3,12: lose
  } else if (roll1 %in% c(2,3,12)){
    return(list(res = "l", last = lastRoll))

  # if we didn't win or lose we need to go into the points rolls
  } else {
    return(list(res = pointsRoll(roll1), last = lastRoll))
  }
}
```

Let's test it to make sure it works…

```r
playCraps()
```

```
## $res
## [1] "w"
##
## $last
## [1] 6
```

```r
playCraps(dragonStyle = T)
```

```
## $res
## [1] "w"
##
## $last
## [1] 6
```

```r
playCraps(roll1 = 4)
```

```
## $res
## [1] "w"
##
## $last
## [1] 4
```

Question 1:

Four players play ten rounds of craps.

```r
# Input the players
players = c("Lingjun", "Sara", "Alex", "Nick")

for(player in players){

  #Start at 0 dollars in winnings
  dollars = 0

  #10 rounds
  for(i in 1:10){
    #play the game
    game = playCraps()$res

    #divy reward
    if(game == 'w'){
      dollars = dollars + 1
    } else {
      dollars = dollars - 1
    }
  }

  #record the winnings
  print(paste0(player, " won $", dollars, "."))
}
```

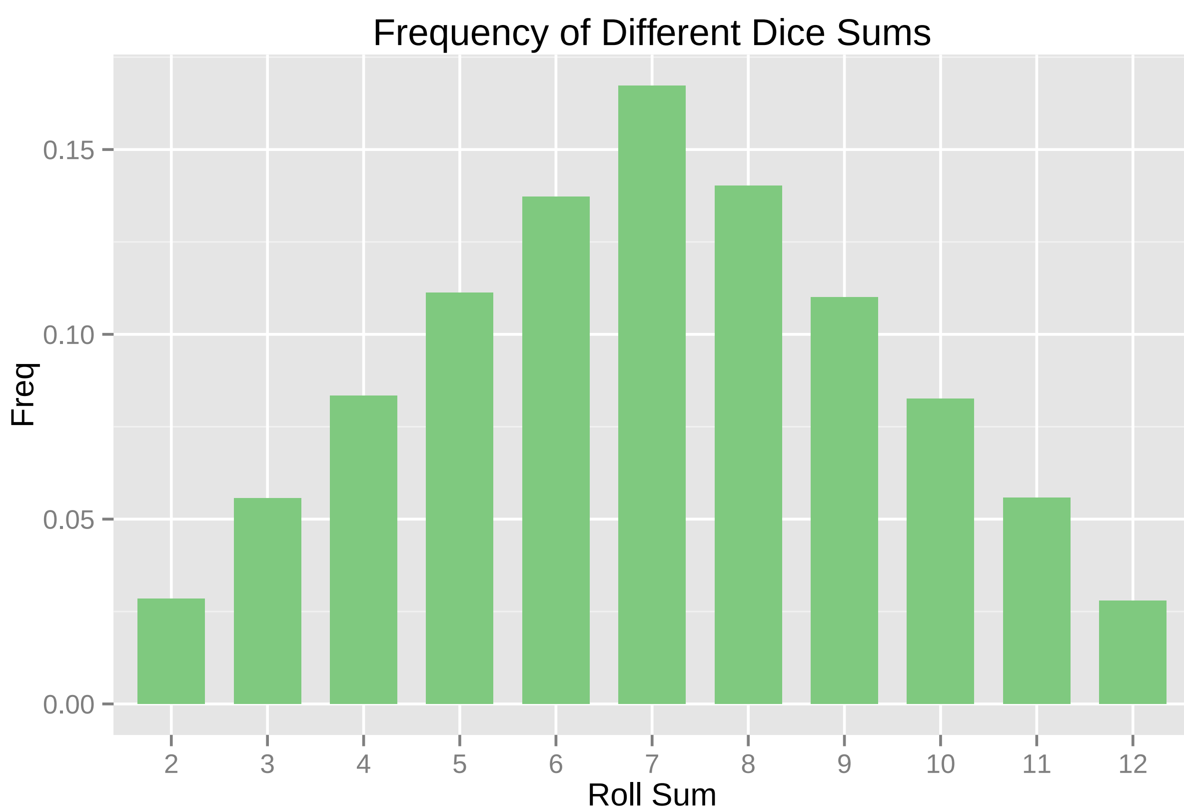
```
## [1] "Lingjun won $-2."
## [1] "Sara won $0."
## [1] "Alex won $0."
## [1] "Nick won $-8."
```

For these the `n` is too small to do much.

Question 2 ‘

I’m going to pull out my roll function from above for this.

```
roll = function(){
  d1 = die()
  d2 = die()
  return(list(one = d1, two = d2, sum = d1+d2))
}

#Let's simulate 50000 rolls
sums = NULL
for(i in 1:50000)  sums = c(sums, roll()$sum)

#and we plot... in ggplot.
d = as.data.frame(table(sums)/length(sums)) #Set up data in ggplot format
p = ggplot(d, aes(x = factor(sums), y = Freq,width=0.7)) +
  geom_bar(stat="identity", fill="#7fc97f") +
  labs(title = "Frequency of Different Dice Sums", x = "Roll Sum", y = "Freq")
p #draw
```



This looks good!

Question 3:

Probability of rolling a natural? To do this we will (1)roll a bunch of dice, (2)check to see if they are 7 or 11 and if they are (3)record that, (4)find proportion.

```
naturals = 0
numOfRolls = 50000
#1
for(i in 1:numOfRolls){
  #2
  if( roll()$sum %in% c(7,11) ){
    #3
    naturals = naturals + 1
  }
}

#4
print(paste("The probability of rolling a natural is", naturals/numOfRolls))
```

```
## [1] "The probability of rolling a natural is 0.22046"
```

Problem 4

First we write a function that finds the probability of winning based upon an initial role:

```
roll_x = function(x){
  #Initialize our rolls vector
  rolls = NULL

  #Itterate and fill in that vector
  for(i in 1:50000){
    rolls = c(rolls, playCraps(roll1 = x, dragonStyle = F)$res)
  }

  #Tally the wins and loses
  wins  = sum(rolls == 'w')
  loses = sum(rolls == 'l')

  #Return the proportion of wins
  return(wins/(wins + loses))
}
```

Now that we have that function let's just set `x = 6`.

```r
print(paste("Probability of winning after rolling a 6 first is", roll_x(6)))
```

```
## [1] "Probability of winning after rolling a 6 first is 0.45438"
```

Question 5

```r
probWin = roll_x(6)
probLose = 1 - probWin
EX = probWin*1 + probLose*(-1)
EX
```

Calculate E[X]

```
## [1] -0.09088
```

```r
#Compare to what we found
abs(-1/11 - EX)/(-1/11)
```

```
## [1] -0.00032
```

This matches what we calculated by hand.

```r
EX2 = probWin*(1^2) + probLose*((-1)^2)
VarX = EX2 - EX^2
VarX
```

Calculate Var[X]

```
## [1] 0.9917408
```

```r
#Compare
abs(120/121 - VarX)/(120/121)
```

```
## [1] 5.33248e-06
```

Again this matches.

```r
SDX = sqrt(VarX)
SDX #no need to compare as it is directly derived from V[X]
```

Calculate SD[X]

```
## [1] 0.9958619
```

Problem 6

What is the probability of winning on a 6?

```
winOnX = function(x, n = 50000, dragon = F){

  wins = 0 #wins counter

  for(i in 1:n){ #Let's play craps!
    game = playCraps(dragonStyle = dragon)

    #Did we roll an x last and win? If so, increment
    if(game$res == 'w' & game$last == x ) wins <- wins + 1
  }
  return(wins/n) #what's our prob?
}

print(paste("The probability of winning on a 6 is", winOnX(6) ) )
```

```
## [1] "The probability of winning on a 6 is 0.06306"
```

Problem 7

```
n = 50000 #itterations

wins = 0 #wins counter

for(i in 1:n){ #Let's play craps!
  game = playCraps()
  #Did we roll an x last and win? If so, increment
  if(game$res == 'w') wins <- wins + 1
}

pWin = wins/n
pLose = 1 - pWin

EW = 1*pWin - 1*pLose
print(paste("E[W] =", EW))
```

```
## [1] "E[W] = -0.00956000000000001"
```

```
EW_2 = 1*pWin + 1*pLose
VW = EW_2 - EW^2
print(paste("V[W] =", VW))
```

```
## [1] "V[W] = 0.9999086064"
```

```r
SDW = sqrt(VW)
print(paste("SD[W] =", SDW))
```

```
## [1] "SD[W] = 0.999954302155854"
```

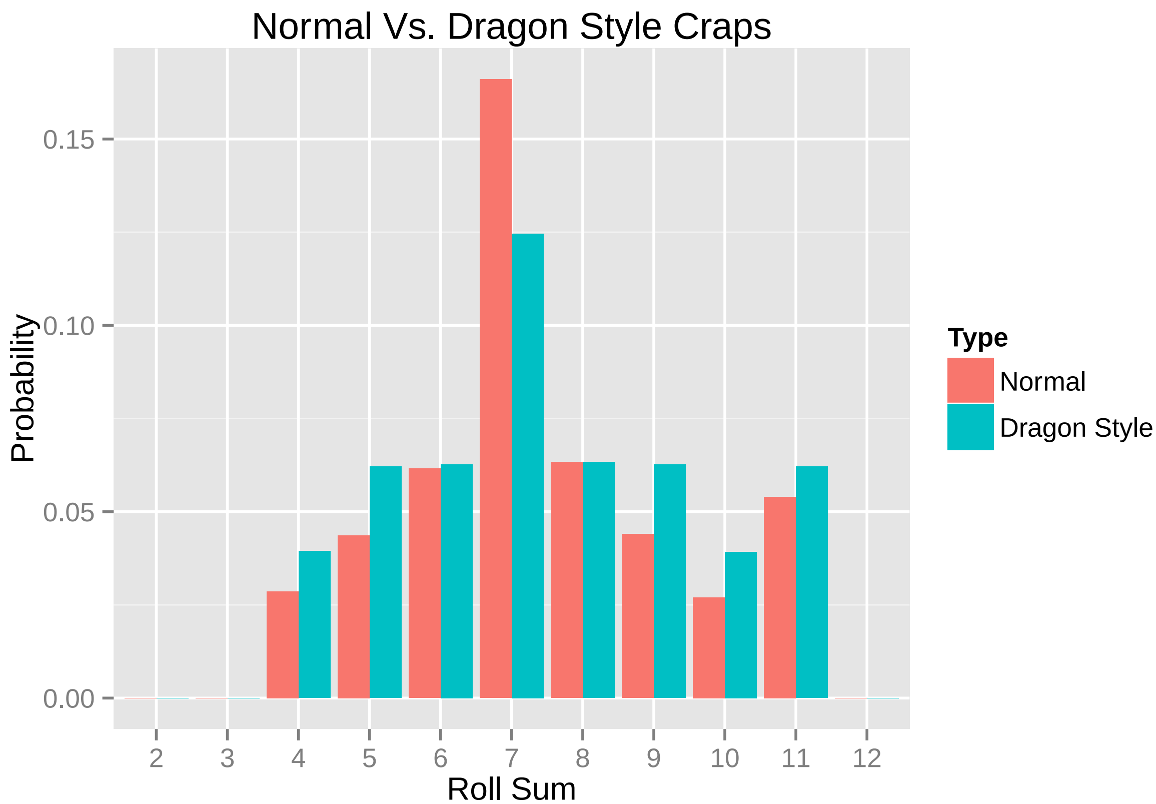
Problem 8

```r
normal = NULL
dragon = NULL
for(x in 2:12){
  normal = c(normal, winOnX(x))
  dragon = c(dragon, winOnX(x, dragon = T))
}

resultsDf = data.frame(val = 2:12, normalStyle = normal, dragonStyle = dragon)
resultsDf
```

```
##    val normalStyle dragonStyle
## 1    2     0.00000     0.00000
## 2    3     0.00000     0.00000
## 3    4     0.02866     0.03946
## 4    5     0.04364     0.06214
## 5    6     0.06156     0.06276
## 6    7     0.16600     0.12462
## 7    8     0.06334     0.06334
## 8    9     0.04398     0.06266
## 9   10     0.02702     0.03924
## 10  11     0.05402     0.06212
## 11  12     0.00000     0.00000
```

Let's plot to compare the two games visually.

```r
#We need to restructure the data to fit with ggplot's desired format. We use the reshape library's "mel
plotData <- melt(resultsDf, id=c("val"))
#Rename the columns so they make more sense.
names(plotData) <- c("val", "type", "prob")

#Construct the ggplot.
probsPlot = ggplot(plotData, aes(x = factor(val), y = prob, fill = type)) +
  geom_bar(stat="identity", position=position_dodge()) +
  labs(title = "Normal Vs. Dragon Style Craps", x = "Roll Sum", y = "Probability") +
  scale_fill_discrete(name="Type", breaks=c("normalStyle", "dragonStyle"),
                                   labels=c("Normal", "Dragon Style"))
probsPlot
```



Normal Vs. Dragon Style Craps
Probability
0.15
0.10
0.05
0.00
2
3
4
5
6
7
8
9
10
11
12
Roll Sum
Type
Normal
Dragon Style