Smith, Catherine BUAD 502A Probability and Statistics Module 7 Expert Problem Set

Exercise 25.2: The probability that all six faces appear exactly once in six tosses of a fair die is roughly 0.015 or 1.5%. I got to this answer by writing R code that had a vector (x) that contains integers from 1-6 being "rolled" 6 times and returning a count if the value was unique on that "roll" from the previous "roll" until I achieved each value (1, 2, 3, 4, 5, 6). I used the "unique" function first, but received an error that the length would only be one and that wouldn't work for what I was asking R to do. I then added the "length" function to return the number of elements in the vector (6, in this case). The "unique" function returns only elements different than others in the vector. I referenced chapter 4 of the textbook to better understand how to combine and use these functions.

```
> #Exercise 25.2
> nrep = 100000
> count = 0
> for (i in 1:nrep) {
+   x = sample (1:6, 6, replace = TRUE)
+   if(length(unique(x)) == 6) count = (count + 1)
+ }
>
> print(count / nrep)
[1] 0.01621
> #Exercise 25.2
> nrep = 100000
> count = 0
> for (i in 1:nrep) {
+   x = sample (1:6, 6, replace = TRUE)
+   if(length(unique(x)) == 6) count = (count + 1)
+ }
>
> print(count / nrep)
[1] 0.01504
> #Exercise 25.2
> nrep = 100000
> count = 0
> for (i in 1:nrep) {
+   x = sample (1:6, 6, replace = TRUE)
+   if(length(unique(x)) == 6) count = (count + 1)
+ }
>
> print(count / nrep)
[1] 0.01522
> #Exercise 25.2
> nrep = 100000
> count = 0
> for (i in 1:nrep) {
+   x = sample (1:6, 6, replace = TRUE)
+   if(length(unique(x)) == 6) count = (count + 1)
+ }
>
> print(count / nrep)
[1] 0.01526
```

```
> #Exercise 25.2
> nrep = 100000
> count = 0
> for (i in 1:nrep) {
+   x = sample (1:6, 6, replace = TRUE)
+   if(length(unique(x)) == 6) count = (count + 1)
+ }
>
> print(count / nrep)
[1] 0.01497
> #Exercise 25.2
> nrep = 100000
> count = 0
> for (i in 1:nrep) {
+   x = sample (1:6, 6, replace = TRUE)
+   if(length(unique(x)) == 6) count = (count + 1)
+ }
>
> print(count / nrep)
[1] 0.01467
>
```

Exercise 25.3: The probability of no two women standing next to each other in a line composed of forty men and ten women is roughly 11%. I came to this answer by creating a "for loop" in which I first made a vector "x" that includes forty men and ten women. I then made an index "y" that samples the vector "x" to show one possible line order of those fifty people. I then called each position of a woman in that line into another vector. Next, I created a command to determine if the value next to that woman's position in line was also a woman. I then added together all the instances of every repetition (10,000, in this case) in which women stood next to one another in the line and divided that by the number of reps in the "for loop" (10,000). That gave me the probability that two women were standing next to one

another. Therefore, when I "print" the answer, I need to subtract that value from 1 because I am looking for the opposite answer (the probability that no two women hold adjacent positions in the line).

```
> #Exercise 25.3
> nrep = 10000
> count = 0
> for(i in 1:nrep) {
+   x = (c(replicate(10, "W"), replicate(40, "M")))
+   y = sample(x)
+   WomenInLine = which(y=="W")
+   IfWoman = y[WomenInLine + 1]=="W"
+   SumIfWoman = as.numeric(sum(IfWoman, na.rm = TRUE))
+   if(SumIfWoman >= 1) count = (count + 1)
+ }
> print (1-(count/ nrep))
[1] 0.1096
> #Exercise 25.3 take 2
> nrep = 10000
> count = 0
> for(i in 1:nrep) {
+   x = (c(rep(1, 10), rep(0, 40)))
+   y = sample(x)
+   WomenInLine = which(y=="1")
+   IfWomanLogical = y[WomenInLine + 1]=="1"
+   IfWomanNumeric = as.numeric(sum(IfWomanLogical, na.rm = TRUE))
+   if(IfWomanNumeric >= 1) count = (count + 1)
+ }
> print (1-(count/ nrep))
[1] 0.1082
>
```

Exercise 25.4: The probability of getting three 1's in a row when tossing a fair die five times is 0.014038 or 1.4%. To get this answer, I ran a simulation wherein the sum of the die rolls in each combination of consecutive positions of the value 1 is 3 (1+1+1 = 3). I then divided those instances by the number of repetitions (500,000) to get the probability.

```
> #Exercise 25.4
> one_roll = sample(1:6, size = 1, replace = TRUE)
> one_roll
[1] 2
> many_rolls = sample(1:6, size = 5, replace = TRUE)
> many_rolls
[1] 1 3 3 6 2
> nrep = 500000
> count = 0
> for(i in 1:nrep) {
+   x = sample(1:6, size = 5, replace = TRUE)
+   if (x[1]+ x[2]+ x[3]==3) count = count + 1
+   if(x[2]+x[3]+x[4]==3) count = count + 1
+   if(x[3]+x[4]+x[5]==3) count = count + 1
+ }
> print (count / nrep)
[1] 0.014038
```

Smith, Catherine BUAD 502A Probability and Statistics Module 7 Expert Problem Set

Exercise 25.10: The code returns the probability that a series of 5 whole numbers, values 1 through 6, contains only numbers with a minimum over 1 (i.e.: greater than or equal to 2); examples might be 2,4,5,6,3 or 3,4,5,6,4.

```
> #Exercise 25.10
> nrep = 100000
> count = 0
> for(i in 1:nrep) {
+   x = sample(6, 5, replace = TRUE)
+   if (min(x) >= 2) count = count + 1
+ }
> print(count / nrep)
[1] 0.40443
>
```

Exercise 26.2: The lower and upper limits of the 90% confidence interval for the mean of the "precip" data set are 32.15435 and 37.61708, respectively.

```
> #Exercise 26.2
> n = length(precip)
> xbar = mean(precip)
> SDPrecip = sd(precip)
> alpha = (1-0.90)
> critical_value_precip = qt(1-alpha / 2, n - 1)
> half = critical_value_precip * SDPrecip / sqrt(n)
> xbar + c(-1, 1) * half
[1] 32.15435 37.61708
>
```

Exercise 27.3: To solve this problem, I first made the matrix "A" as described in the exercise; I then transposed this from a 2 x 3 matrix into a commensurate 3 x 2 matrix (A^T). After that, I used the "dot multiplication" method to get the values for A * A^T, wherein I multiplied the first row of A by the first column of A^T, then the first row by the second column to get the values 11, 14. I then multiplied the second row of A by the first column of A^T, then the second row of A by the second column of A^T to get the values 14, 20. I used R to check my work; the answers matched.

Smith, Catherine BUAD 502A Probability and Statistics Module 7 Expert Problem Set

```r
#Exercise 27
#By Hand
A = matrix(c(-1, 1, 3, 0, 2, 4), nrow = 2, ncol = 3, byrow = TRUE)
A

A_T = matrix(c(-1, 0, 1, 2, 3, 4), nrow = 3, ncol = 2, byrow = TRUE)
A_T

#Now I do "dot multiplication" of -1, 1, 3 x -1, 1, 3 as (-1 * -1) + (1 * 1) + (3 * 3) = 11.
# -1, 1, 3 x 0, 2, 4 as (-1 * 0) + (1 * 2) + (3 * 4) = 14
#0, 2, 4 x -1, 1, 3 as (0 * -1) + (2 * 1) + (4 * 3) = 14
#0, 2, 4 x 0, 2, 4 as (0 * 0) + (2 * 2) + (4 * 4) = 20.

#Therefore, row 1's values are 11, 14 and row 2's values are 14, 20.

#In R
t(A)

A %*% t(A)

#Check
A %*% A_T
```

```
> #Exercise 27
> #By Hand
> A = matrix(c(-1, 1, 3, 0, 2, 4), nrow = 2, ncol = 3, byrow = TRUE)
> A
     [,1] [,2] [,3]
[1,]   -1    1    3
[2,]    0    2    4
> A_T = matrix(c(-1, 0, 1, 2, 3, 4), nrow = 3, ncol = 2, byrow = TRUE)
> A_T
     [,1] [,2]
[1,]   -1    0
[2,]    1    2
[3,]    3    4
> #In R
> t(A)
     [,1] [,2]
[1,]   -1    0
[2,]    1    2
[3,]    3    4
> A %*% t(A)
     [,1] [,2]
[1,]   11   14
[2,]   14   20
> #Check
> A %*% A_T
     [,1] [,2]
[1,]   11   14
[2,]   14   20
>
```