DSSH 6301 - HW 03 Solutions

Here are suggested solutions to the homework assignment. Note that many problems have multiple solutions, so we present here merely one example. If you did not have points taken off for a question, we considered your solution correct.

Problem 1

Part a

There are five ways to get some number followed by its succcessor:

```
5 * 1/6 * 1/6
```

```
## [1] 0.1388889
```

Alternatively, you can just count up the total number of ordered pairs, and divide by the total sample space size: {1,2},{2,3},{3,4},{4,5},{5,6} = 5 out of 36 possible combinations.

Part b

If B = the inner circle and A = the outer circle, then $P(B) = P(A\&B) = 0.05$ and $P(A) = 2/3$. What we want is $P(A\&B|A)$ – the probability of getting the bullseye conditional on knowing you're already inside A. Informally, if the outer circle has area 2/3, and the inner circle has area 0.05, then the percentage of A that is B (assuming B is entirely within A) is 0.05 / (2/3), or 0.075. More formally, since we know that $P(A\&B|A)P(A) = P(A\&B)$ and therefore $P(A\&B|A) = P(A\&B)/P(A)$, thus $P(A\&B|A) =$

```
5/100 / (2/3)
```

```
## [1] 0.075
```

This is of course larger than 0.05 because the bullseye is a larger percentage of the inner circle, so your chance of getting the bullseye condition on being in the inner circle is higher.

Part c

For efficiency, define a function using these relationships. This function can be reused later.

$$\Pr(d|+) = \frac{\Pr(+|d) * \Pr(d)}{\Pr(+)}$$

where

$$\Pr(+) = \Pr(+|d)\Pr(d) + \Pr(+|nd)(1 - \Pr(d))$$

```r
p_disease_given_pos <- function(p_disease, sensitivity, fp_rate) {
  sensitivity*p_disease / (sensitivity*p_disease + fp_rate*(1-p_disease))
}

p_disease   <- 1/1000
sensitivity <- 95/100
fp_rate     <- 5/100

p_disease_given_pos(p_disease, sensitivity, fp_rate)
```

```
## [1] 0.01866405
```

Part d

```r
p_disease_given_pos(1/10000, sensitivity, fp_rate)
```

```
## [1] 0.001896586
```

Part e

The probability that a person has a disease after testing positive for that disease is proportional to the probability of the disease in the general population. So very rare diseases can be difficult to test for. You would need to get the false positive rate down below the probability of the disease, close to 0, to have a good test. This is extremely difficult. From this relationship between a rare classification and the false positive rate arises the false positive paradox.

```r
probs <- c(p_disease_given_pos(1/10000, sensitivity, fp_rate=10/100),
           p_disease_given_pos(1/10000, sensitivity, fp_rate=1/100),
           p_disease_given_pos(1/10000, sensitivity, fp_rate=1/1000),
           p_disease_given_pos(1/10000, sensitivity, fp_rate=1/10000),
           p_disease_given_pos(1/10000, sensitivity, fp_rate=1/100000))
cat(paste(probs, collapse="\n"))
```

```
## 0.000949193185792077
## 0.00941153160293244
## 0.086765914695406
## 0.487204472024206
## 0.904770521624016
```

Problem 2

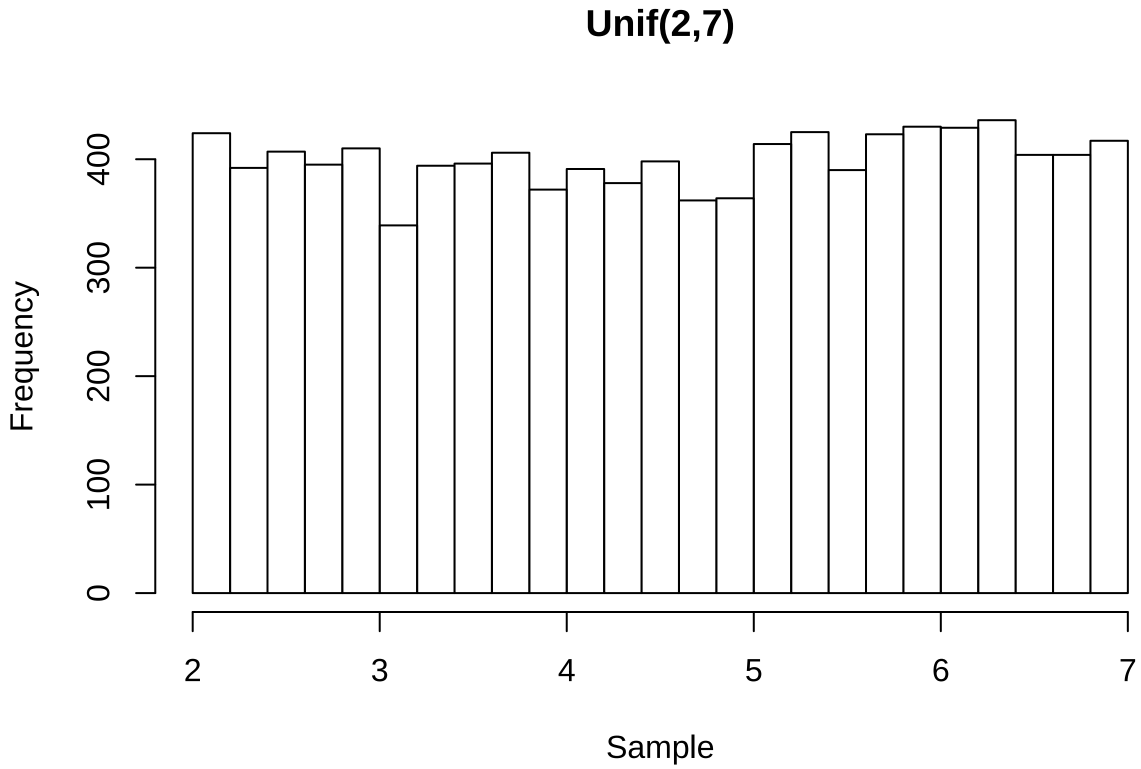
Part a

```r
dice_rolls <- sample(1:20, size=1000, replace=T)
sum(dice_rolls < 11)
```

```
## [1] 484
```

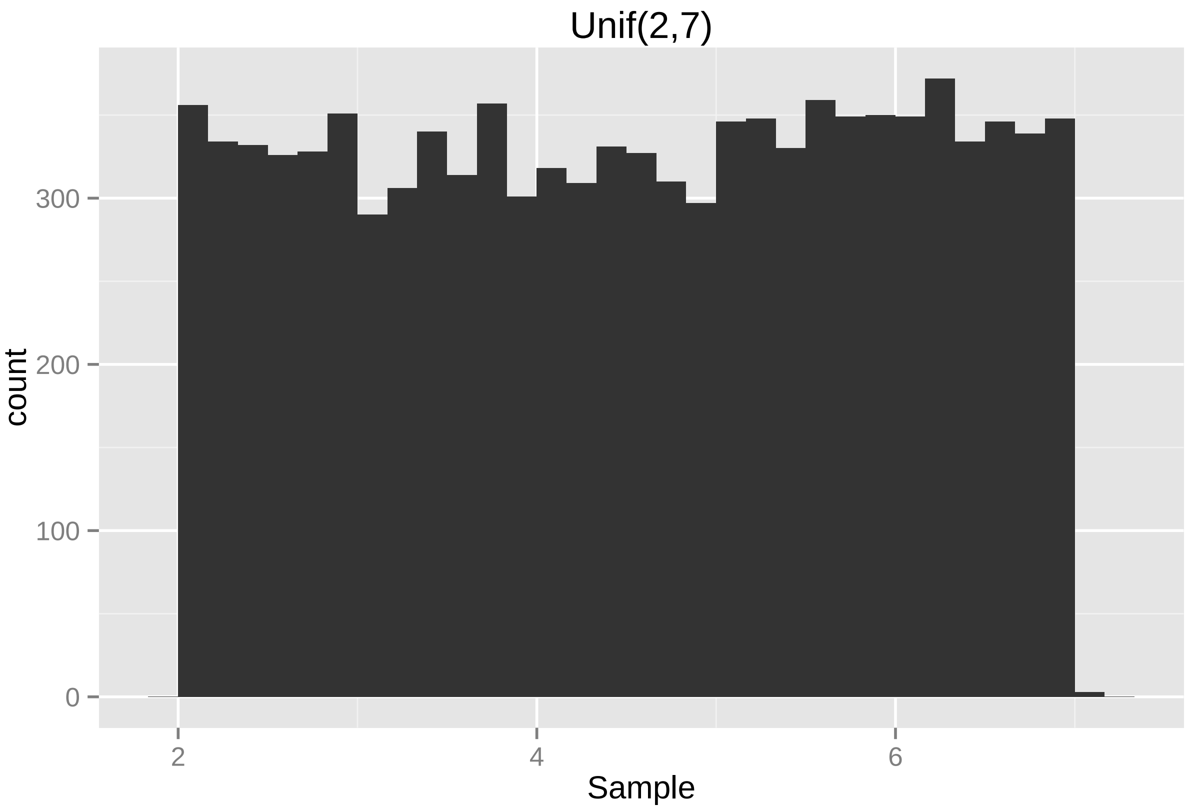
Part b

```
unif_draws <- runif(10000, min=2, max=7)

xlab <- "Sample"
main <- "Unif(2,7)"

library(ggplot2)
hist(unif_draws, main=main, xlab=xlab, breaks=30)
```



```
ggplot(data=data.frame(x=unif_draws), aes(x)) + geom_histogram() +
  ggtitle(main) + xlab(xlab)
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



Part c

Latex:

```
$$ p(x) = \left\{\begin{array}{cl} \frac{1}{7-2} & \mbox{for } x \in [2,7] \\ 0 & \mbox{elsewhere}
\end{array}\right.  $$
```

$$ p(x) = \left\{\begin{array}{cl} \frac{1}{7-2} & \mbox{for } x \in [2,7] \\ 0 & \mbox{elsewhere} \end{array}\right. $$

Part d

```
punif(3.2, min=2, max=7) - punif(1.5, min=2, max=7)
```

```
## [1] 0.24
```

Problem 3

Part a

```
sum(dbinom(0:500, 10000, prob = 1/20))
```

```
## [1] 0.511895
```

```r
# This can also be calculated directly with the CDF:
pbinom(500, 10000, prob = 1/20)
```

```
## [1] 0.511895
```

```r
sum(dice_rolls==20) / length(dice_rolls)
```

```
## [1] 0.05
```

Part b

```r
rbinom(1, 100, 1/100)
```

```
## [1] 0
```

Part c

```r
avg <- 1
1 - ppois(1, lambda = avg)
```

```
## [1] 0.2642411
```

```r
# lower.tail=F is another way to do the upper half of the distribution
ppois(1, lambda = avg, lower.tail=F)
```

```
## [1] 0.2642411
```

Part d

```r
mean <- 70
sd   <- 10

1- pnorm(85, mean=mean, sd=sd)
```

```
## [1] 0.0668072
```

```r
# or equivalently:
pnorm(85, mean=mean, sd=sd, lower.tail=F)
```

```
## [1] 0.0668072
```

```r
pnorm(60, mean=mean, sd=sd) - pnorm(50, mean=mean, sd=sd)
```

```
## [1] 0.1359051
```