Problem 5

(a)

For a Monte Carlo Approximation, we can simulate a sample of 20 8-sided die and counting the number of times we get i rolls greater than or equal to 6 and j rolls greater than or equal to 4. By counting these we can estimate the probability.

```
# Run number of simulations to estimate probability
  number_sim <- 10000

  # Add conditions to satisfy calculating probability for events
  condition_1 = condition_2 <- c(0)

  # Simulating n= number_sim rolls
  for(k in 1:number_sim){
    sample_rolls <- sample(1:8, size = 20, replace = TRUE)

    # Counting the number of dice rolls greater than or equal to 6 in the sample
    i_check <- sum(sample_rolls[]>=6)
    # Counting the number of dice rolls greater than or equal to 4 in the sample
    j_check <- sum(sample_rolls[]>=4)

    # Counting the samples which satisfy conditions 1, that is there are at least i rolls greater than 
    if(i_check >= i){
      condition_1 = condition_1 + 1
    }
    # Counting the samples which satisfy conditions 1, that is there are at least j rolls greater than 
    if(j_check >= j){
      condition_2 = condition_2 + 1
    }
  }
  # Estimating conditional probability
  prob <- condition_1/condition_2
```

(b) and (c)

We can create a function with inputs i and j to output the probabilities for the given i and j. Then we can run a loop to create a matrix for the probabilities for each i= [12,14] and j=[8,12].

```
sim_dice <- function(i,j){
  number_sim <- 10000
  condition_1 = condition_2 <- c(0)
  for(k in 1:number_sim){
    sample_rolls <- sample(1:8, size = 20, replace = TRUE)
    if(sum(sample_rolls[]>=6) >= i){
      condition_1 = condition_1 + 1
    }
    if(sum(sample_rolls[]>=4) >= j){
      condition_2 = condition_2 + 1
    }
  }
```

```r
  prob <- condition_1/condition_2

  return(prob)
}

output <- data.frame(row.names = c("j=8","j=9","j=10","j=11","j=12"))
for(m in 1:3){
  for(o in 1:5){
    output[o,m] <- sim_dice(m+11,o+7)
  }
}
colnames(output) <- c("i=12","i=13","i=14")

output
```

```
##             i=12       i=13        i=14
## j=8  0.03395654 0.01354630 0.003644094
## j=9  0.03656894 0.01108923 0.003408739
## j=10 0.03432171 0.01144788 0.004779492
## j=11 0.04113267 0.01381249 0.004506150
## j=12 0.05020313 0.01763928 0.004199855
```

The above are the estimated probabilities for every combination of is and js.