HW2.1

```r
set.seed(1984)
```

In the posted homework solutions, there is a 20% chance that there is an error in the posted solutions. If an error is present, then the number of students who send emails about the error follows a poisson(3) distrubution.

If no emails were recieved, what is the probability that there was no error

We must first define our information

The probability of an error in the posted solutions is 20%
$\therefore prob.of.error.hw = P(E) = 0.20$

Also,
$\therefore prob.of.no.error.hw = P(E^c) = 0.80$

```r
set.seed(1984)
hw_errors <- function(n){
  num_of_emails.df <- data.frame() #Creation of Empty Data Frame
for(i in 1:n){x_ind <- 0 #Indicator Variable
  x_ind <- sample(c(0,1),1,replace = TRUE, prob = c(0.8,0.2)) #Probability of Error in HW
if(x_ind == 1){num_of_emails.df <- rbind(num_of_emails.df, rpois(1,3))}
} #If there is an error what is the number of emails received with
#respect to the poisson distrubution
no_emails_with_error <- sum(num_of_emails.df == 0) #Sum the number of 0 email occurances
return(no_emails_with_error/n) #Sum of the number of 0 emails weighted by the total
##number of HW Solutions posted
    }

#Therefore, the probability of an error being in the solutions, yet no email
#received about it, is about
hw_errors(50000)
```

```
## [1] 0.00958
```

HW2.2

Calculate P(X is odd) in each setting below.
(a) X ~ Geometric(0.7)
(b) X ~ Binomial(101, 0.5)
(c) First let Y ~ N(0,1), and then let X be the answer you get when you round Y to the nearest integer.
###

a) X ~ Geometric(0.7)

```r
set.seed(1984)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
```

```
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
prob_geom_odd <- function(n, p){
  geom.df <- data.frame((rgeom(n, p))/2)
  colnames(geom.df) <- c("Binary")
    geom.binary.df <- (1 *(geom.df%%1==0))
return(1 - (sum(geom.binary.df)/n))
}

prob_geom_odd(500000, 0.7)
```

```
## [1] 0.231108
```

b) Binomial(101, 0.5)

```
prob.binom.odd <- function(n, size, prob){
  binom.df <- data.frame((rbinom(n, size, prob))/2)
  colnames(binom.df) <- c("Binary")
    binom.binary.df <- (1 *(binom.df%%1==0))
return(1 - (sum(binom.binary.df)/n))
}

prob.binom.odd(101, 500000, 0.5)
```

```
## [1] 0.4950495
```

(c) First let Y ~ N(0,1), and then let X be the answer you get when you round Y to the nearest integer.

```
prob_norm_odd <- function(n){
  norm.df <- data.frame(rnorm(n))
    norm.round.df <- round(norm.df)
      colnames(norm.round.df) <- c("Binary")
        norm.round.df <- (norm.round.df/2)
          norm.round.binary.df <- (1 *(norm.round.df%%1==0))
return(1 - (sum(norm.round.binary.df)/n))
  }

prob_norm_odd(50000)
```

```
## [1] 0.48746
```