Problem 4

(a)

The probability of a bit being decoded erroneously is p. Since the occurs independently from bit to bit, we can say that the probability for j bits to be decoded erroneously when n bits are transmitted is given by:

$$P(j\ bits\ erroneous\ |\ n\ bits\ transmitted) = \binom{n}{j} \cdot p^j \cdot (1-p)^{(n-j)}$$

(b)

To calculate the probability that the error-correcting code can correct the errors given n-digits are transmitted and j bits are incorrectly decoded, we can find the number of ways codes can be constructed which leads to the error-correcting code not working. Thus,

$$P(Code\ cannot\ be\ corrected) = \frac{\binom{j}{2} \cdot 2! \cdot (n-1)!}{n!}$$

We choose 2 out of the j incorrect bits, which can be arranged themselves in 2! ways, and arrange the rest $(n-1)$ bits in $(n-1)!$ ways. We divide this by the total number of ways the bits can be arranged, which is $n!$.

Now, the probability that the error-correcting code can correct the errors is given by,

$$P(Code\ can\ be\ corrected) = 1 - P(Code\ cannot\ be\ corrected)$$
$$= 1 - \frac{\binom{j}{2} \cdot 2! \cdot (n-1)!}{n!}$$

(c)

```
decoder_prob <- function( no_of_bits = 8, prob_incorrect = 0.01, runs = 1000){
  count <- c(0)
  for(i in 1:runs){
    decoder_sample <- sample( c(0,1), size = no_of_bits, replace = TRUE,
                              prob= c(prob_incorrect, (1-prob_incorrect)))
    flag <- c(0)
    for(i in 1:(no_of_bits-1)){
      if(decoder_sample[i]==0 && decoder_sample[i+1]==0){
       flag = 1
      }
    }
    if(flag==1){
      count <- count + 1
    }
  }
  (runs-count)/runs
}

prob_1 <- data.frame()
for(i in 1:4){
```

```
  for(j in 1:4){
    if(j==1){
      prob_1[i,j] <- decoder_prob(i*8, 0.01, 10000)
    }
    if(j>1){
      prob_1[i,j] <- decoder_prob(i*8, (j-1)*0.05, 10000)
    }
  }
}
row.names(prob_1) <- c("n=8","n=16", "n=24", "n=32")
colnames(prob_1) <- c("p=0.01", "p=0.05","p=0.10","p=0.15")
prob_1
```

```
##      p=0.01 p=0.05 p=0.10 p=0.15
## n=8  0.9988 0.9833 0.9347 0.8689
## n=16 0.9990 0.9628 0.8715 0.7385
## n=24 0.9981 0.9474 0.8077 0.6260
## n=32 0.9982 0.9287 0.7577 0.5350
```

(d)

```
improved_decoder_prob <- function(no_of_bits = 8, prob_incorrect = 0.01, runs = 1000){
  count <- c(0)
  for(i in 1:runs){
    decoder_sample <- sample( c(0,1), size = no_of_bits, replace = TRUE,
                              prob= c(prob_incorrect, (1-prob_incorrect)))
    flag <- c(0)
    for(i in 1:(no_of_bits-2)){
      if(decoder_sample[i]==0 && decoder_sample[i+1]==0 && decoder_sample[i+2]==0){
       flag = 1
      }
    }
    if(flag==1){
      count <- count + 1
    }
  }
  (runs-count)/runs
}

prob_2 <- data.frame()
for(i in 1:4){
  for(j in 1:4){
    if(j==1){
      prob_2[i,j] <- improved_decoder_prob(i*8, 0.01, 10000)
    }
    if(j>1){
      prob_2[i,j] <- improved_decoder_prob(i*8, (j-1)*0.05, 10000)
    }
  }
}
row.names(prob_2) <- c("n=8","n=16", "n=24", "n=32")
```

```
colnames(prob_2) <- c("p=0.01", "p=0.05","p=0.10","p=0.15")
prob_2
```

```
##      p=0.01 p=0.05 p=0.10 p=0.15
## n=8       1 0.9993 0.9936 0.9814
## n=16      1 0.9980 0.9878 0.9597
## n=24      1 0.9972 0.9827 0.9357
## n=32      1 0.9955 0.9729 0.9158
```

Problem 5

(a)

The ball has a 50-50 chance of going to the left or to the right. Thus, for the ball to go into cell 0, it has to take 5 left turns at each peg encountered. So,

$$P(Ball\ goes\ into\ cell\ 0) = .5^5 = 0.03125$$

For cell 1, the ball has to take 4 left turns, and 1 right turn. However, the right turn can take place at different pegs. To take into account the different paths that the ball can take, we can use combination.

$$P(Ball\ goes\ into\ cell\ 1) = \binom{5}{1}(0.5)^4(0.5)^1 = 5 \cdot (.5)^4 \cdot (.5)^1 = 0.15625$$

Similarly, we can calculate other probabilities as well:

$$P(Ball\ goes\ into\ cell\ 2) = \binom{5}{2}(0.5)^3(0.5)^2 = 0.3125$$
$$P(Ball\ goes\ into\ cell\ 3) = \binom{5}{3}(0.5)^2(0.5)^3 = 0.3125$$
$$P(Ball\ goes\ into\ cell\ 4) = \binom{5}{4}(0.5)^1(0.5)^4 = 0.15625$$
$$P(Ball\ goes\ into\ cell\ 5) = \binom{5}{5}(0.5)^0(0.5)^5 = 0.03125$$

(b)

We can make a function that simulates the Galton Board or quincunx for a given number of runs and number of rows of pegs.

```
galton_sim <- function(no_row_of_pegs = 5, no_runs = 100){
  cells <- as.data.frame(matrix(0, nrow = (no_row_of_pegs+1), ncol = 1))
  for(i in 0:(no_row_of_pegs)){
    row.names(cells)[i+1] <- paste("Cell",i)
  }

  for(i in 1:(no_runs)){
      marble_run <- sample((0:1), size = no_row_of_pegs, replace = TRUE)
      cells[(sum(marble_run)+1), 1] <- cells[(sum(marble_run)+1), 1] + 1
```

```
  }
  colnames(cells) <- paste("Number of Balls")
  cells
}
```

We can use this function to simulate a quincunx with 5 rows of pegs, and drop a ball 1000 times, recording the values.

```
sim_1 <- galton_sim(5, 1000)
```

To compare with theoretical values, we can make another row which shows theoretical values and another with expected values for a 1000 runs. We can see that the frequencies calculated through running the experiment quite closely resemble the expected number of balls given by theoretical probabilites.

```
for (i in 1:6) {
  sim_1[i, 2] <- choose(5,(i-1))*((0.5)^(i))*((0.5)^(5-i))
  sim_1[i, 3] <- (sim_1[i,2])*1000
}
colnames(sim_1) <- c("Number of Balls","Theoretical Probabilities",
                     "Expected Number of Balls")
sim_1
```

```
##        Number of Balls Theoretical Probabilities Expected Number of Balls
## Cell 0              25                   0.03125                    31.25
## Cell 1             136                   0.15625                   156.25
## Cell 2             341                   0.31250                   312.50
## Cell 3             313                   0.31250                   312.50
## Cell 4             143                   0.15625                   156.25
## Cell 5              42                   0.03125                    31.25
```

(c)

For each cell, we can generalize the probability given the number of rows of pegs is 100, and there is a 50-50 chance of the ball going left or right. For a given cell $k-1$, where $k \in N$ is given by:

$$P(Ball\ Goes\ Into\ Cell\ k-1) = \binom{100}{k}(0.5)^k(0.5)^{(100-k)}$$

(d)

We can use the same function again with the number of rows set as 100, and do a 1000 runs on this, recording the values.

```
sim_2 <- galton_sim(100, 1000)
```

To compare easily with theoretical values, we can make another row which shows theoretical values and another with expected values for a 1000 runs. We can see again that the actual frequencies obtained through the experiment closely resemble the theoretical values or the expected number of balls in each cell.

```r
for (i in 1:101) {
  sim_2[i, 2] <- round(choose(100,(i-1))*((0.5)^(i))*((0.5)^(100-i)), digits = 5)
  sim_2[i, 3] <- round((sim_2[i,2])*1000, digits = 2)
}
colnames(sim_2) <- c("Number of Balls","Theoretical Probabilities",
                     "Expected Number of Balls")
sim_2
```

```
##         Number of Balls Theoretical Probabilities Expected Number of Balls
## Cell 0                0                   0.00000                     0.00
## Cell 1                0                   0.00000                     0.00
## Cell 2                0                   0.00000                     0.00
## Cell 3                0                   0.00000                     0.00
## Cell 4                0                   0.00000                     0.00
## Cell 5                0                   0.00000                     0.00
## Cell 6                0                   0.00000                     0.00
## Cell 7                0                   0.00000                     0.00
## Cell 8                0                   0.00000                     0.00
## Cell 9                0                   0.00000                     0.00
## Cell 10               0                   0.00000                     0.00
## Cell 11               0                   0.00000                     0.00
## Cell 12               0                   0.00000                     0.00
## Cell 13               0                   0.00000                     0.00
## Cell 14               0                   0.00000                     0.00
## Cell 15               0                   0.00000                     0.00
## Cell 16               0                   0.00000                     0.00
## Cell 17               0                   0.00000                     0.00
## Cell 18               0                   0.00000                     0.00
## Cell 19               0                   0.00000                     0.00
## Cell 20               0                   0.00000                     0.00
## Cell 21               0                   0.00000                     0.00
## Cell 22               0                   0.00000                     0.00
## Cell 23               0                   0.00000                     0.00
## Cell 24               0                   0.00000                     0.00
## Cell 25               0                   0.00000                     0.00
## Cell 26               0                   0.00000                     0.00
## Cell 27               0                   0.00000                     0.00
## Cell 28               0                   0.00000                     0.00
## Cell 29               0                   0.00001                     0.01
## Cell 30               0                   0.00002                     0.02
## Cell 31               0                   0.00005                     0.05
## Cell 32               0                   0.00011                     0.11
## Cell 33               1                   0.00023                     0.23
## Cell 34               1                   0.00046                     0.46
## Cell 35               2                   0.00086                     0.86
## Cell 36               1                   0.00156                     1.56
## Cell 37               2                   0.00270                     2.70
## Cell 38               5                   0.00447                     4.47
## Cell 39              11                   0.00711                     7.11
## Cell 40              11                   0.01084                    10.84
## Cell 41              17                   0.01587                    15.87
## Cell 42              18                   0.02229                    22.29
## Cell 43              28                   0.03007                    30.07
```

```
## Cell 44           39           0.03895           38.95
## Cell 45           38           0.04847           48.47
## Cell 46           71           0.05796           57.96
## Cell 47           64           0.06659           66.59
## Cell 48           77           0.07353           73.53
## Cell 49           77           0.07803           78.03
## Cell 50           91           0.07959           79.59
## Cell 51           70           0.07803           78.03
## Cell 52           57           0.07353           73.53
## Cell 53           63           0.06659           66.59
## Cell 54           73           0.05796           57.96
## Cell 55           46           0.04847           48.47
## Cell 56           37           0.03895           38.95
## Cell 57           28           0.03007           30.07
## Cell 58           27           0.02229           22.29
## Cell 59           12           0.01587           15.87
## Cell 60            9           0.01084           10.84
## Cell 61            9           0.00711            7.11
## Cell 62            4           0.00447            4.47
## Cell 63            5           0.00270            2.70
## Cell 64            3           0.00156            1.56
## Cell 65            0           0.00086            0.86
## Cell 66            3           0.00046            0.46
## Cell 67            0           0.00023            0.23
## Cell 68            0           0.00011            0.11
## Cell 69            0           0.00005            0.05
## Cell 70            0           0.00002            0.02
## Cell 71            0           0.00001            0.01
## Cell 72            0           0.00000            0.00
## Cell 73            0           0.00000            0.00
## Cell 74            0           0.00000            0.00
## Cell 75            0           0.00000            0.00
## Cell 76            0           0.00000            0.00
## Cell 77            0           0.00000            0.00
## Cell 78            0           0.00000            0.00
## Cell 79            0           0.00000            0.00
## Cell 80            0           0.00000            0.00
## Cell 81            0           0.00000            0.00
## Cell 82            0           0.00000            0.00
## Cell 83            0           0.00000            0.00
## Cell 84            0           0.00000            0.00
## Cell 85            0           0.00000            0.00
## Cell 86            0           0.00000            0.00
## Cell 87            0           0.00000            0.00
## Cell 88            0           0.00000            0.00
## Cell 89            0           0.00000            0.00
## Cell 90            0           0.00000            0.00
## Cell 91            0           0.00000            0.00
## Cell 92            0           0.00000            0.00
## Cell 93            0           0.00000            0.00
## Cell 94            0           0.00000            0.00
## Cell 95            0           0.00000            0.00
## Cell 96            0           0.00000            0.00
## Cell 97            0           0.00000            0.00
```

```
## Cell 98               0               0.00000                0.00
## Cell 99               0               0.00000                0.00
## Cell 100              0               0.00000                0.00
```