1.1 Random Binary Expansions

QUESTION 1

The program to approximate the cumulative distribution function (CDF) by the Monte-Carlo simulation, by generating a random sample of variables X_i and then using this sample to calculate and plot the empirical cumulative distribution, is **Code 1** found on page 9, labelled as

```
Monte_Carlo(p,n)
```

I chose my N to be 10^5, because the standard deviation is proportional to $\frac{1}{\sqrt{N}}$, hence selecting N as 10^5 would reduce the standard deviation by over 100 times, making its accuracy over 90%.

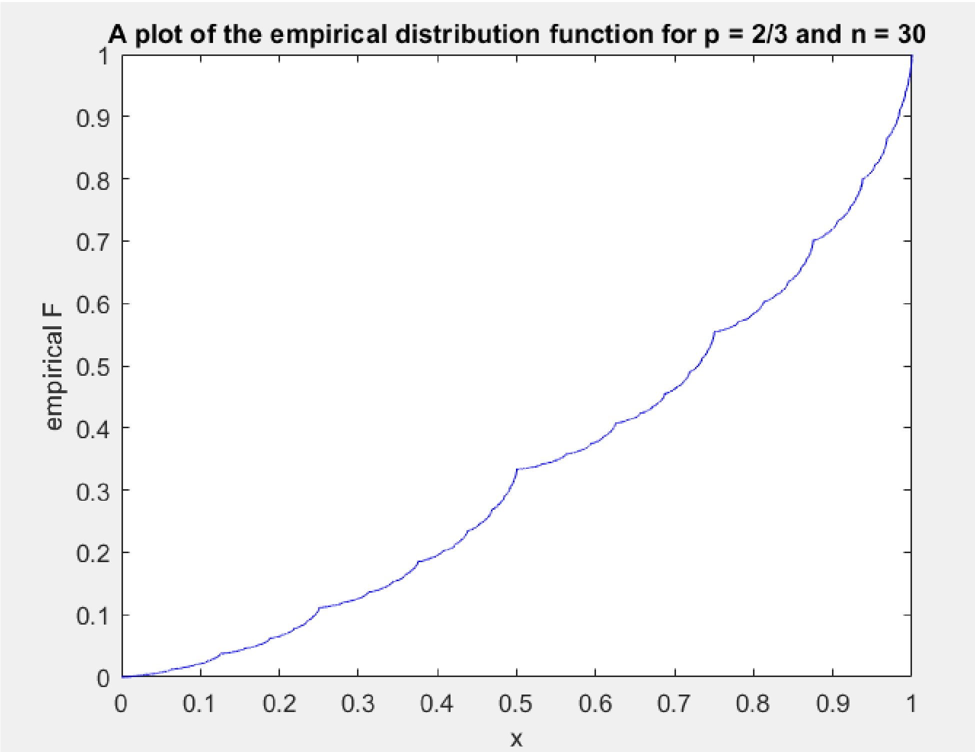

Figure 1: *A graph of the empirical distribution function for p = 2/3 and n = 30, approximated using the Monte-Carlo simulation*

QUESTION 2

Recall the definition of F(x):

$$F(x) = \mathbb{P}(X \leq x)$$

We are given that:

$$x = \sum_{i=1}^{n} \frac{x_i}{2^i}$$

, where each x_i can take one of two values: 0 or 1.

We can also think of X as

$$X = \sum_{i=1}^{n} \frac{X_i}{2^i}$$

, where each X_i can be either 1 or 0 as well.

Then we can think of X as

$$X = (0.X_1X_2X_3 \dots)_2$$

, a decimal of base 2. We can do the same for x too:

$$x = (0.x_1x_2x_3 \dots)_2$$

Then, in order for X < x, we can see that:

$$\begin{aligned}\mathbb{P}(X \leq x) = \mathbb{P}(X < x) \\ &= \mathbb{P}(X_1 < x_1) + \mathbb{P}(X_1 = x_1, X_2 < x_2) + \mathbb{P}(X_1 = x_1, X_2 = x_2, X_3 < x_3) + \cdots \\ &+ \mathbb{P}(X_1 = x_1, X_2 = x_2, \dots X_n < x_n)\end{aligned}$$

Look at $\mathbb{P}(X_1 < x_1)$ first:

- In order for this to occur, we require $X_1 = 0$, which has probability 1 – p. But we also definitely require x_1 to be equal to 1, otherwise the probability will be zero.
- So, we can think of the probability as $(1-p)I[x_1 = 1] = (1-p)x_1$.

Now consider $\mathbb{P}(X_1 = x_1, X_2 < x_2)$ next:

- In order for this to occur, we require the same conditions on X_2 as we had before on X_1, giving us a probability of $(1-p)x_2$.
- However, we also must ensure that $X_1 = x_1$; if $x_1 = 1$, we need $X_1 = 1$, which has probability p, but if $x_1 = 0$, we need $X_1 = 0$, which has probability 1 – p.
- So, we can think of $\mathbb{P}(X_1 = x_1) = p^{I[x_1=1]}(1-p)^{I[x_1=0]} = p^{x_1}(1-p)^{1-x_1}$.
- This leads to total probability $p^{x_1}(1-p)^{1-x_1}(1-p)x_2 = p^{x_1}(1-p)^{2-x_1}x_2$.

Hence, we get subsequent terms $p^{x_1+x_2}(1-p)^{3-(x_1+x_2)}x_3, \dots, p^{x_1+\cdots x_{n-1}}(1-p)^{n-(x_1+\cdots+x_{n-1})}x_n$.

Altogether, we get $\mathbb{P}(X \leq x) = \mathbb{P}(X < x) = (1-p)x_1 + p^{x_1}(1-p)^{2-x_1}x_2 + \cdots +$
$p^{x_1+\cdots x_{n-1}}(1-p)^{n-(x_1+\cdots+x_{n-1})}x_n$

$$= \boldsymbol{\sum_{j=1}^{n} p^{\sum_{i=1}^{j-1} x_i}(1-p)^{j-\sum_{i=1}^{j-1} x_i} x_j}.$$

QUESTION 3

The program to generate and plot the cumulative distribution function (CDF) for p = 3/4 and n = 11, sampling F(x) at $x = 0, \frac{1}{2^n}, \frac{2}{2^n}, \frac{3}{2^n}, \ldots, 1$ is **Code 2** found on page 10, labelled as

```
CDF(p,n)
```

Every program used in question 3 onwards, i.e. questions 5 and 6 inclusively implements a program used to write a number between 0 and 1 as a finite binary expansion. This is also written under **Code 2,** labelled as

```
binary(fraction,n)
```

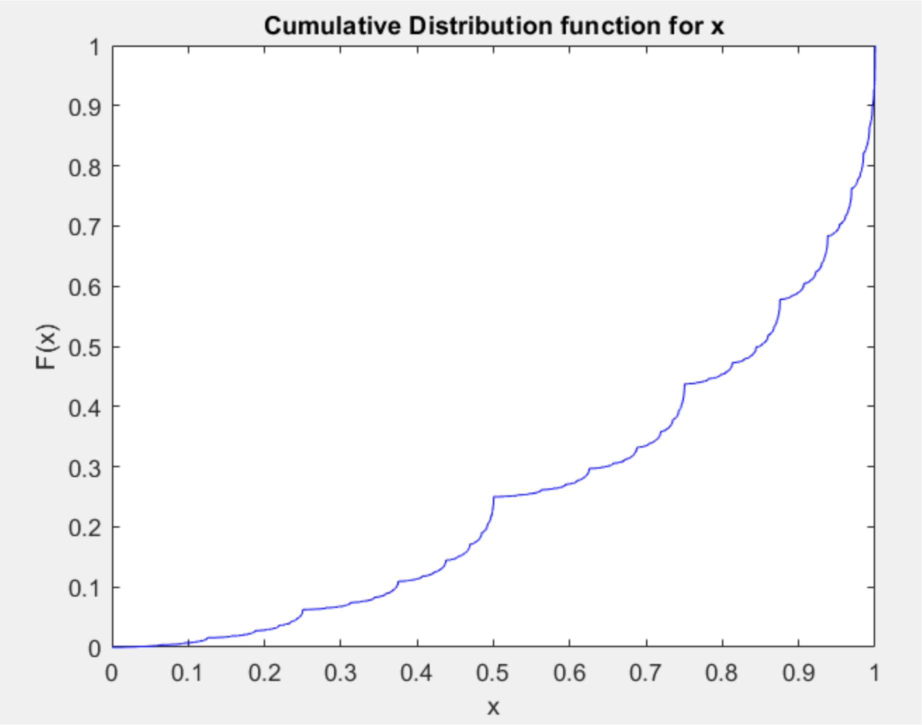

Figure 2: *A graph of the cumulative distribution function for p = 3/4*

The general shape of this graph is similar to the graph we obtained in Question 1, except that it dips lower, due to the probability being 3/4 here (higher than the 2/3 in Question 1). As for complexity, the complexity of my Monte-Carlo program is O(n), whereas the second program that calculates F(x) explicitly for finite binary expansions is $O(2^n)$.

QUESTION 4

To test for continuity at c, where c can be written as a finite binary expansion, we need to see the behaviour of $F(c + \delta) - F(c)$; take $\delta = \frac{1}{2^n}$. When adding $\frac{1}{2^n}$ to c, there are four cases of what could happen to the final two binary values x_{n-1} and x_n:

- 01 → 10
- 10 → 11

- 11 → 00
- 00 → 01

In every single case, we end up getting $F(c+\delta) - F(c)$ proportional to $(1-p)^n$, but I will illustrate it with one case, 10 → 11. As the difference only lies in the last two binary digits, I will choose to only look at the j = n – 1 and j = n terms in the sum definition of $F(x)$:

$$p^{\sum_{i=1}^{n-2} x_i}(1-p)^{n-1-\sum_{i=1}^{n-2} x_i}x_{n-1} + p^{\sum_{i=1}^{n-2} x_i}(1-p)^{n-\sum_{i=1}^{n-2} x_i}x_n - p^{\sum_{i=1}^{n-1} c_i}(1-p)^{n-1-\sum_{i=1}^{n-1} c_i}c_{n-1} - p^{\sum_{i=1}^{n-1} c_i}(1-p)^{n-\sum_{i=1}^{n-1} c_i}c_n$$

, where the x_i's are the binary coefficients of $c+\delta$, and the c_i's are the binary coefficients of c. In this case, we can see that:

- $x_{n-1} = c_{n-1} = 1$
- $x_n = 1, c_n = 0$

So, the expression above simplifies to:

$$\left(\left(\frac{p}{1-p}\right)^{\sum_{i=1}^{n-2} x_i} - \left(\frac{p}{1-p}\right)^{\sum_{i=1}^{n-2} c_i}\right)(1-p)^{n-1} + \left(\left(\frac{p}{1-p}\right)^{\sum_{i=1}^{n-1} x_i}\right)(1-p)^n$$
$$= \left(\left(\frac{p}{1-p}\right)^{\sum_{i=1}^{n-1} x_i}\right)(1-p)^n$$

$\left(\frac{p}{1-p}\right)^{\sum_{i=1}^{n} x_i}$ is unaffected as n →∞, as we are only dealing with finite binary expansions, so it is only necessary to focus on $(1-p)^n$. As $1-p<1$, $(1-p)^n \to 0$ as $n \to \infty$, so we get $F(c+\delta) - F(c) \to 0, n \to \infty$ too, which is exactly what is wanted! With the same method applied to the other three cases for the last two binary digits, we get that $F(c+\delta) - F(c) \to 0, n \to \infty$ for those cases too. As this is a cumulative distribution and hence strictly increasing, we need to only deal with right-continuity, which we have proven. Therefore, $F(x)$ is continuous at c.

However, as the plot suggests, this proof fails for c that's not a finite binary expansion and is therefore not continuous at those points. In the proof above, I chose to not focus on $\left(\frac{p}{1-p}\right)^{\sum_{i=1}^{n-1} x_i}$, as the proof only dealt with finite binary expansions then, and hence this factor was unaffected as n →∞. However, in this case, we cannot ignore that factor in the same way, as it will be affected as n →∞, leading to that proof failing.

QUESTION 5

The programs used to plot $\frac{F(c+\delta)-F(c)}{\delta}$ against δ for both positive and negative δ is on page 11 under **Code 3,** labelled as

```
CDF_differentiability_right(p,c)
& CDF_differentiability_left(p,c)
```

This program inclusively implements a program to calculate F(x) at an inputted point x, with probability p. This is also written under **Code 3**, labelled as

```
CDF_2(p,x)
```

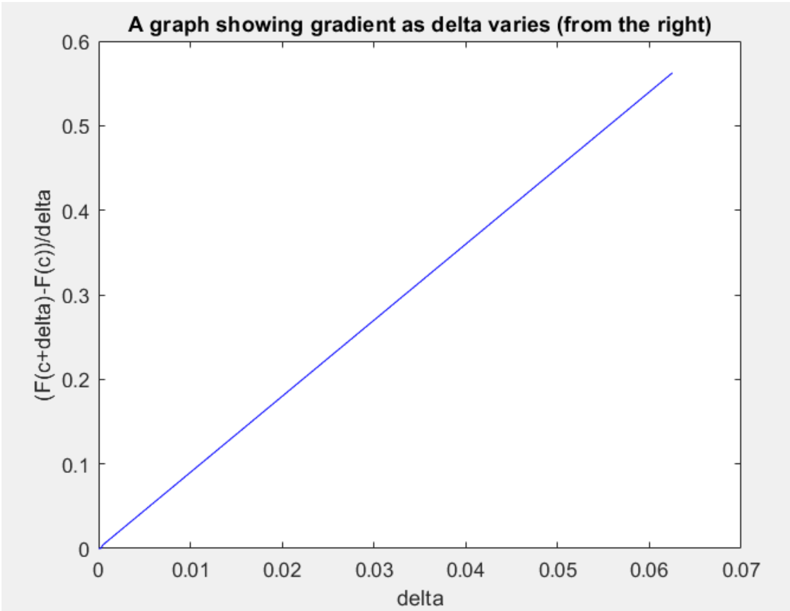

Figure 3: *A graph showing* $\frac{F(c+\delta)-F(c)}{\delta}$ *against* δ *for* δ *positive, showing right differentiability*

Looking at this graph, we see that as $\delta \to 0$, $\frac{F(c+\delta)-F(c)}{\delta} \to 0$ too, meaning that from the right, the gradient seems to be 0.

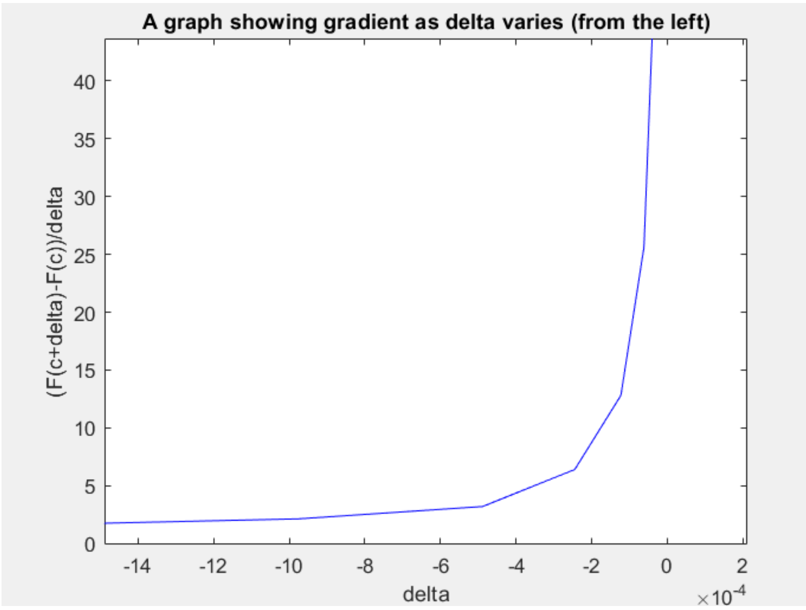

Figure 4: *A graph showing* $\frac{F(c+\delta)-F(c)}{\delta}$ *against* δ *for* δ *negative, showing left differentiability*

Looking at this graph, we see that as $\delta \to 0$, $\frac{F(c+\delta)-F(c)}{\delta} \to \infty$, meaning that from the left, the gradient tends to ∞.

These plots suggest that F is right-differentiable at point c.

QUESTION 6

This question requires exactly the same code as in Question 5, but with different input parameters used. These input parameters are specified below.

(p, c) = (1/4, 9/16):

From the right:

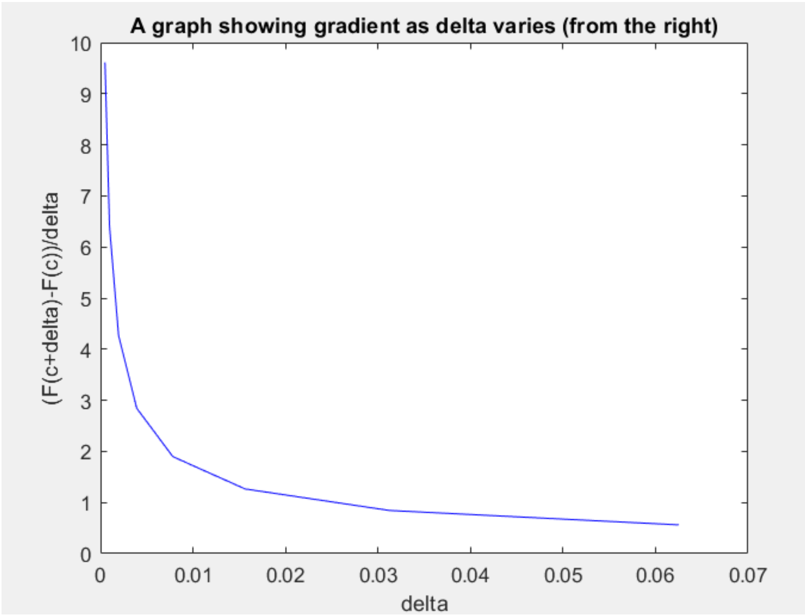

Figure 5: *A graph showing* $\frac{F(c+\delta)-F(c)}{\delta}$ *against δ for δ positive and p<1/2, showing right differentiability*

From the left:

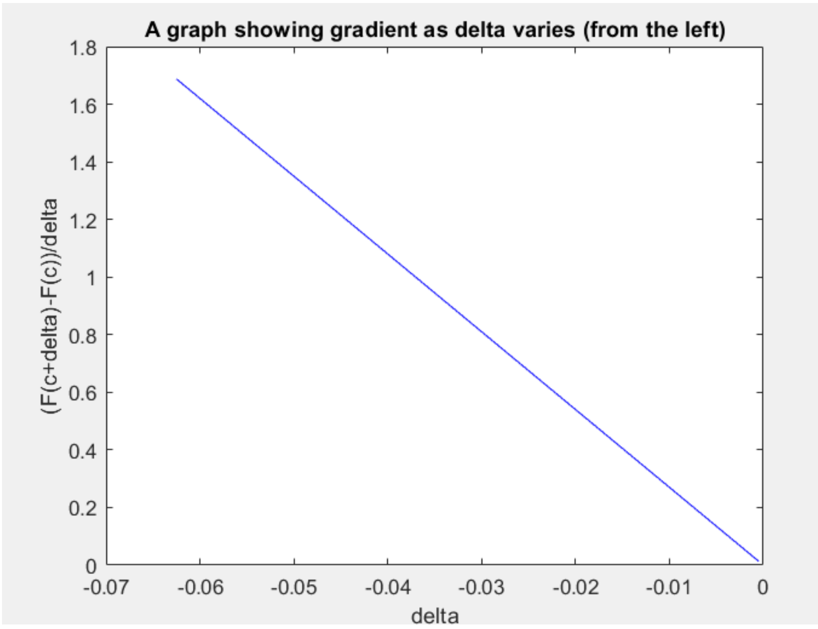

Figure 6: *A graph showing* $\frac{F(c+\delta)-F(c)}{\delta}$ *against δ for δ negative and p<1/2, showing left differentiability*

(p, c) = (4/5, 9/32):

From the right:

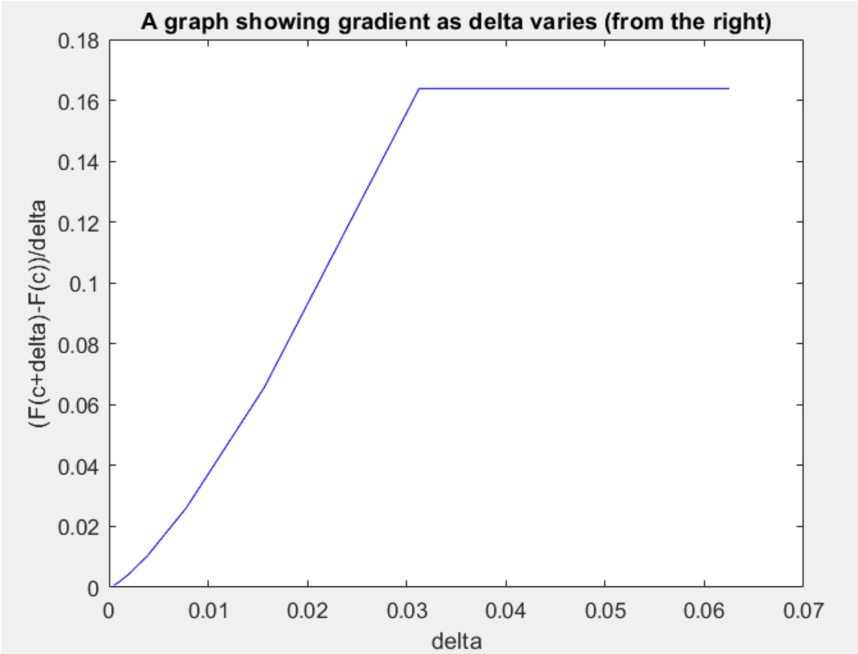

Figure 7: *A graph showing* $\frac{F(c+\delta)-F(c)}{\delta}$ *against δ for δ positive and p>1/2, showing right differentiability*

Based on these plots, my prediction is that F is only right differentiable at an arbitrary point c with a finite binary expansion for p > ½, but only left differentiable for p < ½, for arbitrary p ∈ (0, 1).

Let's begin by looking at the right-side limit of $\frac{F(c+\delta)-F(c)}{\delta}$, taking δ as $\frac{1}{2^n}$. Then we have that

$$\lim_{\delta\to 0}\frac{F(c+\delta)-F(c)}{\delta} \propto \lim_{n\to\infty}\left(2^n p^{\sum_{i=1}^{n-1}x_i}(1-p)^{n-\sum_{i=1}^{n-1}x_i}\right)$$

Once again, like in question 4, you get the following four cases for the last two binary digits:

- 01 → 10
- 10 → 11
- 11 → 00
- 00 → 01

Like in question 4, because it is only the last two digits that change, we need only look at j = n – 1, or j = n.

However, like in question 4, it is not necessary to individually consider each case, as they all generalise to what is written above.

Rewrite the right-hand side as such:

$$\lim_{n\to\infty}\left(2^n\left(\frac{p}{1-p}\right)^{\sum_{i=1}^{n-1}x_i}(1-p)^n\right)$$

The only part of this that determines what the limit is, is $(2(1-p))^n$, as $\left(\frac{p}{1-p}\right)^{\sum_{i=1}^{n-1} x_i}$ is a positive constant regardless of what value p takes. So, we get two cases depending on what p is:

- p < ½: limit's ∞
- p > ½: limit's 0

Now look at the left-side limit of $\frac{F(c+\delta)-F(c)}{\delta}$, taking δ as $-\frac{1}{2^n}$ instead. We can actually see, by chain rule, that $\frac{dF_p}{dx} = \frac{dF_{1-p}}{d(1-x)}$, so now simply looking at the previous proof for right-hand differentiability and replacing 1 – p with p, we get that the limits for various p are:

- p < ½: limit's 0
- p > ½: limit's ∞

Hence whether it's right-differentiable or left differentiable depends on the value of p:

- p < ½: left-differentiable
- p > ½: right-differentiable

Programs

CODE 1

```
function [vector_f] = Monte_Carlo(p,n)
vector_U=zeros(1,n);
vector_x=zeros(1,n);
vector_X=zeros(1,100000);
vector_f=zeros(1,10000);
for k=1:100000
    for i=1:n
        random_number=rand;
        if random_number<=p
            result=1;
        else
            result=0;
        end
    vector_U(i)=result;
    vector_x(i)=result/(2^i);
    end
    X_n=sum(vector_x);
    vector_X(k)=X_n;
end
x_interval=linspace(0,1,10000);
for j=1:10000
    total_result=0;
    for a=1:100000
        if vector_X(a)<=x_interval(j)
            result_2=1;
        else
            result_2=0;
        end
        total_result=total_result + result_2;
    end
    vector_f(j)=(total_result)/100000;
end
plot(x_interval,vector_f,'b-')
xlabel('x')
ylabel('empirical F')
title('A plot of the empirical distribution function for p = 2/3 and n = 30')
```

CODE 2

<u>Function to write each x as a binary expansion:</u>

```
function [binary_expansion] = binary(fraction,n)
binary_expansion = zeros(1,n);
for i = 1:n
    fraction = fraction * 2;
    binary_expansion(i) = floor(fraction);
    fraction = fraction - floor(fraction);
    if fraction == 0
        break;
    end
end
```

<u>Full code to plot the Cumulative Distribution Function:</u>

```
function [Cumulative_DF] = CDF(p,n)
x_interval=linspace(0,1,2^n);
Cumulative_DF=zeros(1,2^n);
for i=1:2^n
    binary_expansion=binary(x_interval(i),n);
    Cumulative_DF(i)=(1-p)*binary_expansion(1);
    for j=1:n-1
        total=0;
        for k=1:j
            total=total+binary_expansion(k);
        end
        Cumulative_DF(i)=Cumulative_DF(i)+(p)^(total)*(1-p)^(j+1-
total)*(binary_expansion(j+1));
    end
end
plot(x_interval,Cumulative_DF,'b-')
xlabel('x')
ylabel('F(x)')
title('Cumulative Distribution function for x as a binary expansion')
```

CODE 3

Function to calculate F(x) at an inputted value of x, for a certain probability p:

```
function [Cumulative_DF] = CDF_2(p,x)
binary_expansion=binary(x,11);
Cumulative_DF=(1-p)*binary_expansion(1);
for j=1:10
    total=0;
    for k=1:j
        total=total+binary_expansion(k);
    end
    Cumulative_DF=Cumulative_DF+(p)^(total)*(1-p)^(j+1-
total)*(binary_expansion(j+1));
end
```

Function to test for right-differentiability:

```
function [Gradient_CDF] = CDF_differentiability_right(p,c)
delta_interval=zeros(1,8);
for j=4:11
    delta_interval(j-3)=1/2^j;
end
Gradient_CDF=zeros(1,8);
for i=1:8
    Gradient_CDF(i)=(CDF_2(p,c+delta_interval(i))-CDF_2(p,c))/delta_interval(i);
end
plot(delta_interval,Gradient_CDF,'b-')
xlabel('delta')
ylabel('(F(c+delta)-F(c))/delta')
title('A graph showing gradient as delta varies (from the right)')
```

Function to test for left-differentiability:

```
function [Gradient_CDF_left] = CDF_differentiability_left(p,c)
delta_interval=zeros(1,8);
for j=4:11
    delta_interval(j-3)=-1/2^j;
end
Gradient_CDF_left=zeros(1,8);
for i=1:8
    Gradient_CDF_left(i)=(CDF_2(p,c+delta_interval(i))-
CDF_2(p,c))/delta_interval(i);
end
plot(delta_interval,Gradient_CDF_left,'b-')
xlabel('delta')
ylabel('(F(c+delta)-F(c))/delta')
title('A graph showing gradient as delta varies (from the left)')
```