Problem1

Yufei Yin

Question 1a, (4 points):

Create a function in R named counts. This function should take as parameters a numeric vector x and also a number indicating a number of bins n. The function will consider the range [min(x), max(x)], and then consider a partition of this interval into n non-overlapping equally sized half open intervals: $I_1 = [min(x), b_1); I_2 = [b_1, b_2), ..., I_n = [b_{n-1}, max(x))$. Note that since the intervals are equally sized, the value of bi is constrained. The function will then return a vector of length n such that the i-th element of this vector is the number of coordinates of x that lie in the interval I_i. Provide the code for this function: It should be of the following form:

```
counts = function(...) {
  ...
  return(...)
}
```

```
counts = function (x, n) {
  lowvalue = floor(min(x))
  highvalue = ceiling(max(x))
  width2use = (highvalue-lowvalue)/n
  BinEdges = seq(from = lowvalue, to = highvalue, by = width2use)
  a = BinEdges[-length(BinEdges)]
  b = BinEdges[-1]
  out = rep(NA, length(a))
  for(index in seq_along(a)){
    out[index]=sum(x>=a[index] & x<b[index])
  }
  return(out)
}
```

Question 1b, (3 points):

Create a function in R called histo. This function should take the same x and n parameters as the count function you wrote in the previous question, and it should plot a histogram of the vector x with n bins. The body of this function may make use of the count function you wrote in the previous question. The only plotting functions you may make use of are the plot function and the lines function ('just add an egg'). You may not make use of the hist function. Provide your code for this function. Hint: There are several ways to do this, one way would be to create a new and empty plot with:

```
plot(1,
    type="n",
    xlab="x",
    ylab="Counts",
    xlim=c(...),
    ylim=c(...))
```

Then make a for loop through the bins and call lines in the body of the for loop so that three lines (delinating the left, top, and right of the box) are drawn. As before (although this time without a return statement), your code should be in the following form:

```
histo = function( ... ){
  ...
}
```

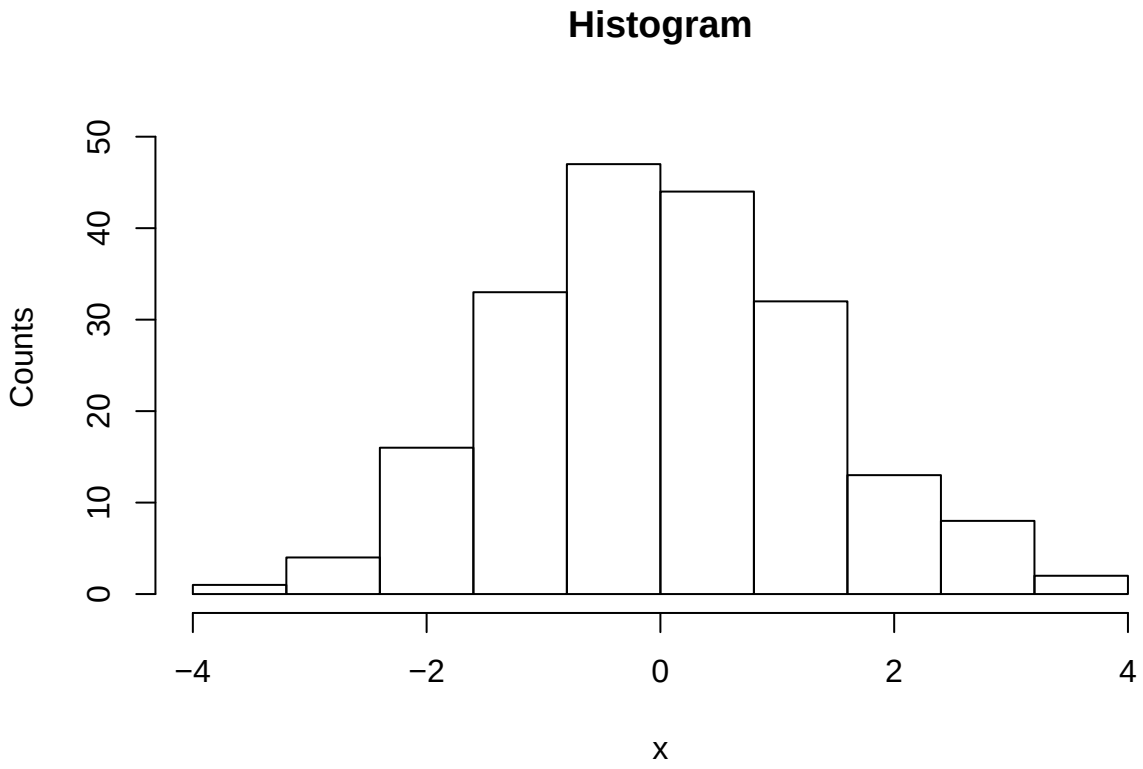
```
histo = function(x, n){
  lowvalue = floor(min(x))
  highvalue = ceiling(max(x))
  width2use = (highvalue-lowvalue)/n
  BinEdges = seq(from = lowvalue, to = highvalue, by = width2use)
  a = BinEdges[-length(BinEdges)]
  b = BinEdges[-1]

  # counts function from question 1a
  out = counts(x,n)

  # corner case
  if (highvalue == max(x)){
    out[length(out)] = sum(x>=a[length(out)] & x<=b[length(out)])
  }

  # plot
  plot(1,
       type="n",
       xlab="x",
       ylab="Counts",
       main="Histogram",
       bty="n",
       xlim = c(lowvalue, highvalue),
       ylim = c(0,max(out)*1.1))
  for (i in seq_along(a)){
    lines(c(a[i],a[i],b[i],b[i],a[i]),c(0,out[i],out[i],0,0))
  }
}
```

Question 1c, (2 points):

We will now test the histo function. Create a vector of length 200 such that the first 100 elements are independent draws from a normal distribution with mean -1 and variance 1 and the second 100 elements are independent draws from a normal distribution with mean 1 and variance 1. Call histo on this vector with 10 bins and provide a graphic including the resulting plot.

```
set.seed(1)
v1 = rnorm(n = 100, mean = -1, sd = 1)
v2 = rnorm(n = 100, mean = 1, sd = 1)
x = c(v1, v2)
```

```
histo(x = x, n = 10)
```



Question 1d, (2 points):

We will now test a corner case of the histo function. Call histo on the vector x = (0, 0, 0, 1, 1, 2) with 3 bins, and provide a graphic including the resulting plot. Note that this is a corner case because the middle values lie at the boundary between two bins, and so their contribution to the histogram's height requires that the half-open nature of the intervals be respected.

```
x = c(0, 0, 0, 1, 1, 2)
n = 3
histo(x = x, n = n)
```