```
## This R code reads a dataset, constructs, tabulates and plots empirical pmf
## also constructs and plots empirical CDF

dataset = as.matrix(read.table("folderpath/filename.extension", header = T or
 F))

n = length(dataset) ## size of dataset

print("sample size")
print(n)

## empirical pmf construction and plotting

## only for values that occur in the data
## probability mass is no of occurrences divided by sample size n

empirical_pmf = function(x){length(which(dataset==x))/n}

## identify the unique values in the dataset
## arrange them from smallest to largest
empirical_pmf_points = sort(unique(dataset))

## how many unique values
no_of_unique_values = length(empirical_pmf_points)

## now determine the probability mass at those unique values
empirical_pmf_values = sapply(empirical_pmf_points,empirical_pmf)

## Now display it in a tabular form as we did in class
print("################################################")
epmf_table = data.frame(points = empirical_pmf_points, probability_mass =
 empirical_pmf_values )
print(epmf_table, row.names=F)
print("################################################")

## Now plot the empirical pmf
x11()

plot(empirical_pmf_points, empirical_pmf_values, "p", pch = 20 )
## drawing the vertical lines at observed points
segments(x0 = empirical_pmf_points,y0 = rep(0,no_of_unique_values),x1 =
 empirical_pmf_points, y1 = empirical_pmf_values)

## ecdf construction and plotting

empirical_CDF = ecdf(dataset)

## plot the ecdf over the range of values in the dataset

## create a very fine grid of points spanning the range of data
plot_at_points = seq(min(dataset) - 0.1, max(dataset) + 0.1,by = 0.0005)
```

```
## now calculate empirical CDF at chosen points
ecdf_values = sapply(plot_at_points,empirical_CDF)

## Now create the plot
x11()

plot(plot_at_points,ecdf_values,"l")
```