```r
## This R code is for Chi-square gof for Discrete Uniform or Poisson distributions only
## does not work for any continuous parametric distribution

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

n = length(dataset) ## size of dataset

print("sample size")
print(n)

## specify cells in the table

## must cover all values allowed under proposed parametric model

## specify starting integer of each cell, separated by commas
I_start = c()
## specify ending integer of each cell, separated by commas
I_end = c()

k = length(I_start) ## how many cells there are

## calculate the number of observations in each cell
n_freq = array(0,k)
for (i in 1:k) n_freq[i] = length(which( dataset >= I_start[i] & dataset <= I_end[i] ))

## Now, need to calculate probability of each cell
## under the parametric model
p_cell = array(0,k)

####################################################################

## run the following four lines only for Discrete Uniform distribution, skip them otherwise

a =  I_start[1] ## the first cell stats at left integer boundary of parametric model
b =  I_end[k]   ## the final cell ends at right integer boundary of parametric model

print(paste("Fitting a DU(",a,",",b,") parametric model",sep=""))

for (i in 1:k) p_cell[i] = (I_end[i] - I_start[i] + 1)/(b-a+1)

## run the following three lines only for Poisson distribution, skip them otherwise

lambda_hat = mean(dataset) ## estimate the lambda parameter from data
print(paste("Fitting a Poi(",round(lambda_hat,3),") parametric model",sep=""))
```

```
for (i in 1:k) p_cell[i] = ppois(I_end[i], lambda = lambda_hat) -
 ppois(I_start[i]-1, lambda = lambda_hat)

####################################################################

## calculate the discrepancy column

discrepancy = n_freq - n*p_cell

## now, create the table

## create the first column where the cells are specified
cells = array(0,k)
for (i in 1:k)
{
 cells[i] = paste("{",I_start[i], " to ", I_end[i],"}",sep="")
}

out =  data.frame( Cells = noquote(cells), data_count = n_freq,
 parametric_probability = round(p_cell,3), discrepancy = round(discrepancy,3) )

## print it

print("------------------------------------------------------")
print(out,row.names=F)
print("------------------------------------------------------")

## check required conditions
print("check required conditions:")

print(sum(out$data_count)) ## 2nd required condition
print(sum(p_cell)) ## 3rd required condition
print(sum(discrepancy)) ## 4th required condition

## check recommended condition
print("Check recommended condition")

## reconstruct the cells if any of the following entires is below 5
print(n*p_cell)

## if the table satisfies all required and recommended conditions,
## compute the test statistic
test_stat = sum((out$discrepancy^2)/(n*out$parametric_probability))

print("test statistics")
print(test_stat)

## now determine the degrees of freedom
## it depends on the parametric model that we are testing
## then use the chi-square table to find a range for the p-value
```