HW4shining.R

Pantazis

Sun Nov 03 17:43:56 2019

```
library(shiny)
```

```
## Warning: package 'shiny' was built under R version 3.5.3
```

```
ui <- fluidPage(titlePanel("HOC-HW#4 STAT705"),
  fileInput(inputId="signal",label="Choose txt/csv file :",
            multiple = FALSE, accept = c("text/csv"),
            width = NULL, buttonLabel = "Select a file..."),
            placeholder = "",
  plotOutput("graph"),
  textOutput("vector")
)

server <- function(input, output, session) {

  output$graph<-renderPlot({
    if(is.null(input$signal)) {

      mean<-matrix(0,9,1)
      standev<-matrix(0,9,1)

      N=1000
      samples=1000
      d<-matrix(0,samples,9)

      for(j in 1:samples){
        ds<-matrix(0,N,9)
        deltaz<-matrix(0,N,9)
        x<-matrix(0,N,9)
        z<-rnorm(N)
        deltaz[1:N,1]=z[1:N]
        for (y in (1:N)[deltaz[,1]>=0]){
          x[y,1]=1
        }
        ds[1:(N-1),1]<-abs(x[2:N,1]-x[1:(N-1),1])^2
        d[j,1]<-sum(ds[,1])
        for(i in 2:9){
          deltaz[1:(N-1),i]=deltaz[2:N,i-1]-deltaz[1:(N-1),i-1]
          for (y in (1:N)[deltaz[,i]>=0]){
            x[y,i]=1

          }
          ds[1:(N-1),i]<-abs(x[2:N,i]-x[1:(N-1),i])^2
          d[j,i]<-sum(ds[,i])
        }



      }
      for(i in 1:9){
        mean[i]<-mean(d[,i])
        standev[i]<-sd(d[,i])
      }
      plot(mean,main = "Average values of white noise over k with the confidence intervals mean+-stdev",xlab
= k, type="b")
      points(d_signal,col="blue",pch="o")

      lines(mean+standev,lty = 2,col="red")
      lines(mean-standev,lty = 2,col="red")
      lines(mean+2*standev,lty = 3,col="red")
      lines(mean-2*standev,lty = 3,col="red")
      lines(mean+3*standev,col="red")
      lines(mean-3*standev,col="red")

    }else{
```

```
signal<-matrix(scan(input$signal$datapath),ncol=1)

mean<-matrix(0,9,1)
standev<-matrix(0,9,1)

N=length(signal)
samples=1000
d<-matrix(0,samples,9)

for(j in 1:samples){
  ds<-matrix(0,N,9)
  deltaz<-matrix(0,N,9)
  x<-matrix(0,N,9)
  z<-rnorm(N)
  deltaz[1:N,1]=z[1:N]
  for (y in (1:N)[deltaz[,1]>=0]){
    x[y,1]=1
  }
  ds[1:(N-1),1]<-abs(x[2:N,1]-x[1:(N-1),1])^2
  d[j,1]<-sum(ds[,1])
  for(i in 2:9){
    deltaz[1:(N-1),i]=deltaz[2:N,i-1]-deltaz[1:(N-1),i-1]
    for (y in (1:N)[deltaz[,i]>=0]){
      x[y,i]=1

    }
    ds[1:(N-1),i]<-abs(x[2:N,i]-x[1:(N-1),i])^2
    d[j,i]<-sum(ds[,i])
  }



}
for(i in 1:9){
  mean[i]<-mean(d[,i])
  standev[i]<-sd(d[,i])
}



N=length(signal)
d_signal<-rep(c(0),9)
delta_signal<-matrix(0,9,N)
x<-matrix(0,9,N)

delta_signal[1,]<-signal

for(i in 2:9){
  delta_signal[i,1:(N-1)]<-delta_signal[i-1,2:N]-delta_signal[i-1,1:(N-1)]
  for (y in (1:N)[delta_signal[i,]>=0]){
    x[i,y]=1
  }
  d_signal[i]<-sum(abs(x[i, 2:N]-x[i,1:(N-1)]))
}
mk<-replicate(c(0),9)
deltak<-replicate(c(0),9)
for(k in 1:9){
  if(k==1){
    mk=mean[k]
    deltak=d_signal[k]
  }else if(k<9){
    mk=mean[k]-mean[k-1]
    deltak=d_signal[k]-d_signal[k-1]
  }else{
    mk=N-mean[k-1]
    deltak=N-d_signal[k-1]
  }
}
vector=sum((deltak-mk)^2/mk)
plot(mean,main = "Average values of white noise over k with the corresponding confidence intervals mea
n+-stdev",xlab = k, type="b")
points(d_signal,col="blue",pch="o")
```

```
    points(d_signal,col="blue",pch="o")

    lines(mean+standev,lty = 2,col="red")
    lines(mean-standev,lty = 2,col="red")
    lines(mean+2*standev,lty = 3,col="red")
    lines(mean-2*standev,lty = 3,col="red")
    lines(mean+3*standev,col="red")
    lines(mean-3*standev,col="red")
  }

})
output$vector<-renderText({
  if(is.null(input$signal)) return(NULL)

  signal<-matrix(scan(input$signal$datapath),ncol=1)

  mean<-matrix(0,9,1)
  standev<-matrix(0,9,1)

  N=length(signal)
  samples=1000
  d<-matrix(0,samples,9)

  for(j in 1:samples){
    ds<-matrix(0,N,9)
    deltaz<-matrix(0,N,9)
    x<-matrix(0,N,9)
    z<-rnorm(N)
    deltaz[1:N,1]=z[1:N]
    for (y in (1:N)[deltaz[,1]>=0]){
      x[y,1]=1
    }
    ds[1:(N-1),1]<-abs(x[2:N,1]-x[1:(N-1),1])^2
    d[j,1]<-sum(ds[,1])
    for(i in 2:9){
      deltaz[1:(N-1),i]=deltaz[2:N,i-1]-deltaz[1:(N-1),i-1]
      for (y in (1:N)[deltaz[,i]>=0]){
        x[y,i]=1

      }
      ds[1:(N-1),i]<-abs(x[2:N,i]-x[1:(N-1),i])^2
      d[j,i]<-sum(ds[,i])
    }



  }
  for(i in 1:9){
    mean[i]<-mean(d[,i])
    standev[i]<-sd(d[,i])
  }



  N=length(signal)
  d_signal<-rep(c(0),9)
  delta_signal<-matrix(0,9,N)
  x<-matrix(0,9,N)

  delta_signal[1,]<-signal

  for(i in 2:9){
    delta_signal[i,1:(N-1)]<-delta_signal[i-1,2:N]-delta_signal[i-1,1:(N-1)]
    for (y in (1:N)[delta_signal[i,]>=0]){
      x[i,y]=1
    }
    d_signal[i]<-sum(abs(x[i, 2:N]-x[i,1:(N-1)]))
  }
  mk<-replicate(c(0),9)
  deltak<-replicate(c(0),9)
  for(k in 1:9){
    if(k==1){
      mk=mean[k]
```

```
        deltak=d_signal[k]
      }else if(k<9){
        mk=mean[k]-mean[k-1]
        deltak=d_signal[k]-d_signal[k-1]
      }else{
        mk=N-mean[k-1]
        deltak=N-d_signal[k-1]
      }
    }
    vector=sum((deltak-mk)^2/mk)
  })
}

shinyApp(ui, server)
```

```
## PhantomJS not found. You can install it with webshot::install_phantomjs(). If it is installed, please mak
e sure the phantomjs executable can be found via the PATH variable.
```

Shiny applications not supported in static R Markdown documents