```
library(shiny)

# Define UI for application that draws a histogram
shinyUI(fluidPage(

  # Application title
  titlePanel("Population Data"),
  # Include radio buttons  (Only one of these can be selected)
  # These radioButtons are used in the server as an input variable called
  # 'dataSource'
  radioButtons("dataSource", "",
               c("Data Source 2006" = "DataFile2006",
                 "Data Source 2011" = "DataFile2011")),
  # Sidebar with a slider input for number of bins
  sidebarLayout(
    sidebarPanel(
        # This is a slider for the number of bins
        #The value determined here will be used as an input in server.R
        #The minimum is 1
        #The max is 50
        #It starts at a value of 30
        #by default it will grow by 1 but here I specify that it will step by 2
        #
        # The format is
        #sliderInput(VariableName, Stuff2WriteInTheWebPage, MinimumValue,MaxValue,Default,StepIncrement)
        sliderInput("bins",  #This is the variable name, the value of which is determined by the slider.
            "Number of bins:",#Label on the html page
            min = 1,#minimum for the slider
            max = 50,#maximum for the slider
            value = 30,#default value
            step=2),#slider increments
    numericInput('minx', 'minimum x', value = 1000),
    numericInput('maxx', 'maximum x', value = 1000000),
    checkboxInput("densityplot", "Add density plot?")),
```

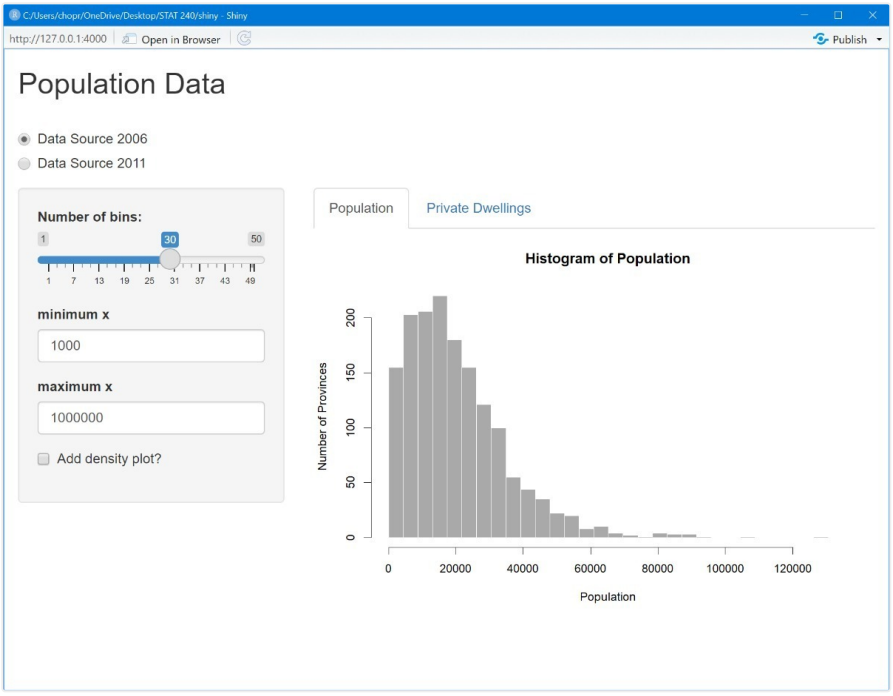
```
    sidebarPanel(
        # This is a slider for the number of bins
        #The value determined here will be used as an input in server.R
        #The minimum is 1
        #The max is 50
        #It starts at a value of 30
        #by default it will grow by 1 but here I specify that it will step by 2
        #
        # The format is
        #sliderInput(VariableName, Stuff2WriteInTheWebPage, MinimumValue,MaxValue,Default,StepIncrement)
        sliderInput("bins",  #This is the variable name, the value of which is determined by the slider.
            "Number of bins:",#Label on the html page
            min = 1,#minimum for the slider
            max = 50,#maximum for the slider
            value = 30,#default value
            step=2),#slider increments
    numericInput('minx', 'minimum x', value = 1000),
    numericInput('maxx', 'maximum x', value = 1000000),
    checkboxInput("densityplot", "Add density plot?")),


    # Show a plot of the generated distribution
    mainPanel(

      tabsetPanel(type = "tabs",
                  tabPanel("Population", plotOutput("distPlot")),
                  tabPanel("Private Dwellings", plotOutput("dist1Plot"))
      )

    )

))
)
```

```
library(shiny)

# Define server logic required to draw a histogram
shinyServer(function(input, output) {

  output$distPlot <- renderPlot({

    #Use those radioButtons to select the data to use
    #Note that the input was called "dataSource" in the ui.R file
    #Here it is an element from a list.
    #I extract it using input$nameOfThing
    if (input$dataSource == "DataFile2006") {
      x = read.csv("pop2006.csv")
    } else {
      if(input$dataSource == "DataFile2011"){
        x = read.csv("pop2011.csv")
      }else{
        #This means do not use any data.  This will break things.
        x=NULL
      }
    }
    #I didn't need to use if{}else{if{}else{}}
    # but I do so to show how to use multiple if else statements.


    # generate bins based on input$bins from ui.R
    a <- x[-1, 3]
    x <- x[-1, 3]
    for (i in 1:length(a))
    {

      if(a[i] <= input$minx || a[i] >= input$maxx)
```

```
  # generate bins based on input$bins from ui.R
  a <- x[-1, 3]
  x <- x[-1, 3]
  for (i in 1:length(a))
  {

    if(a[i] <= input$minx || a[i] >= input$maxx)
    {
      a[i] = NA
    }
  }
  a <- a[!is.na(a)]
  bins <- seq(min(x), max(x), length.out = input$bins + 1)

  # draw the histogram with the specified number of bins
  hist(a, breaks = bins, col = 'darkgray', border = 'white',
       main = "Histogram of Population", ylab = "Number of Provinces", xlab = "Population")

  if(input$densityplot)
  {
    hist(a, breaks = bins,prob = TRUE, col = 'darkgray', border = 'white',
         main = "Histogram of Population", ylab = "Frequency", xlab = "Population")
    lines(density(a),lwd = 2, col = "red")
  }

})
output$dist1Plot <- renderPlot({

  #Use those radioButtons to select the data to use
  #Note that the input was called "dataSource" in the ui.R file
  #Here it is an element from a list.
  #I extract it using input$nameOfThing
  if (input$dataSource == "DataFile2006") {
    x = read.csv("pop2006.csv")
```

```
})
output$dist1Plot <- renderPlot({

  #Use those radioButtons to select the data to use
  #Note that the input was called "dataSource" in the ui.R file
  #Here it is an element from a list.
  #I extract it using input$nameOfThing
  if (input$dataSource == "DataFile2006") {
    x = read.csv("pop2006.csv")
  } else {
    if(input$dataSource == "DataFile2011"){
      x = read.csv("pop2011.csv")
    }else{
      #This means do not use any data.  This will break things.
      x=NULL
    }
  }
  #I didn't need to use if{}else{if{}else{}}
  # but I do so to show how to use multiple if else statements.


  # generate bins based on input$bins from ui.R
  a <- x[-1, 4]
  x <- x[-1, 4]
  for (i in 1:length(a))
  {

    if(a[i] <= input$minx || a[i] >= input$maxx)
    {
      a[i] = NA
    }
  }
  a <- a[!is.na(a)]
```

```
        x=NULL
      }
    }
    #I didn't need to use if{}else{if{}else{}}
    # but I do so to show how to use multiple if else statements.


    # generate bins based on input$bins from ui.R
    a <- x[-1, 4]
    x <- x[-1, 4]
    for (i in 1:length(a))
    {

      if(a[i] <= input$minx || a[i] >= input$maxx)
      {
        a[i] = NA
      }
    }
    a <- a[!is.na(a)]
    bins <- seq(min(x), max(x), length.out = input$bins + 1)

    # draw the histogram with the specified number of bins
    hist(a, breaks = bins, col = 'darkgray', border = 'white',
         main = "Histogram of Population", ylab = "Number of Provinces", xlab = "Population")
    if(input$densityplot)
    {
      hist(a, breaks = bins,prob = TRUE, col = 'darkgray', border = 'white',
           main = "Histogram of Population", ylab = "Frequency", xlab = "Population")
      lines(density(a),lwd = 2, col = "red")
    }

  })

})
```


C:/Users/chopr/OneDrive/Desktop/STAT 240/shiny - Shiny
http://127.0.0.1:4000
Open in Browser
Publish
Population Data
Data Source 2006
Data Source 2011
Number of bins:
1
30
50
1 7 13 19 25 31 37 43 49
minimum x
1000
maximum x
1000000
Add density plot?
Population
Private Dwellings
Histogram of Population
Number of Provinces
0 50 100 150 200
0 20000 40000 60000 80000 100000 120000
Population

```
library(shiny)

# Define UI for application that draws a histogram
shinyUI(fluidPage(

  # Application title
  titlePanel("Population Data"),
  # Include radio buttons  (Only one of these can be selected)
  # These radioButtons are used in the server as an input variable called
  # 'dataSource'
  radioButtons("dataSource", "",
               c("Data Source 2006" = "DataFile2006",
                 "Data Source 2011" = "DataFile2011")),
  # Sidebar with a slider input for number of bins
  sidebarLayout(
    sidebarPanel(
        # This is a slider for the number of bins
        #The value determined here will be used as an input in server.R
        #The minimum is 1
        #The max is 50
        #It starts at a value of 30
        #by default it will grow by 1 but here I specify that it will step by 2
        #
        # The format is
        #sliderInput(VariableName, Stuff2WriteInTheWebPage, MinimumValue,MaxValue,Default,StepIncrement)
        sliderInput("bins",  #This is the variable name, the value of which is determined by the slider.
            "Number of bins:",#Label on the html page
            min = 1,#minimum for the slider
            max = 50,#maximum for the slider
            value = 30,#default value
            step=2),#slider increments
    numericInput('minx', 'minimum x', value = 1000),
    numericInput('maxx', 'maximum x', value = 1000000),
    checkboxInput("densityplot", "Add density plot?")),
```

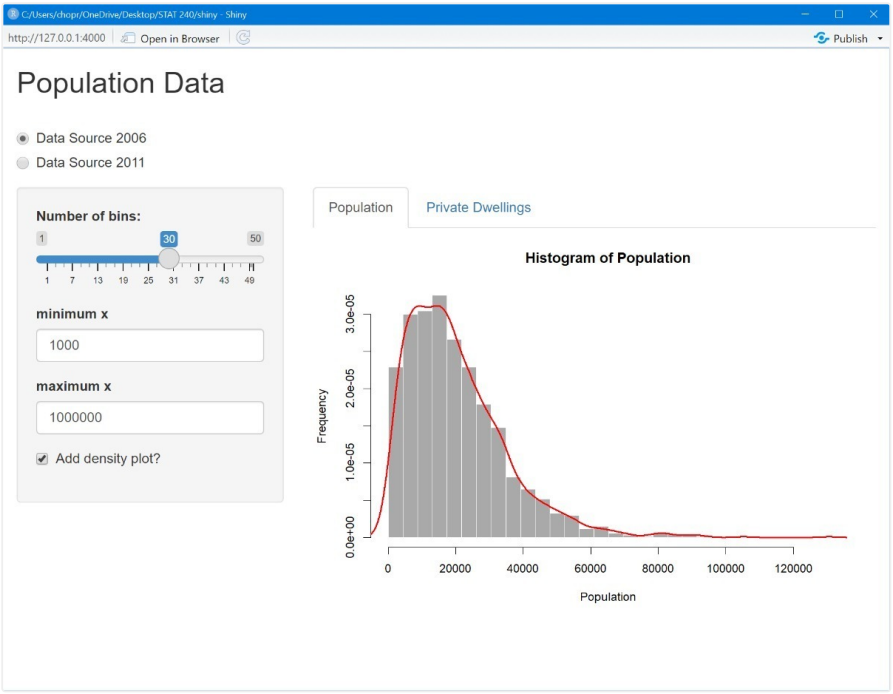
```
    sidebarPanel(
        # This is a slider for the number of bins
        #The value determined here will be used as an input in server.R
        #The minimum is 1
        #The max is 50
        #It starts at a value of 30
        #by default it will grow by 1 but here I specify that it will step by 2
        #
        # The format is
        #sliderInput(VariableName, Stuff2WriteInTheWebPage, MinimumValue,MaxValue,Default,StepIncrement)
        sliderInput("bins",  #This is the variable name, the value of which is determined by the slider.
            "Number of bins:",#Label on the html page
            min = 1,#minimum for the slider
            max = 50,#maximum for the slider
            value = 30,#default value
            step=2),#slider increments
    numericInput('minx', 'minimum x', value = 1000),
    numericInput('maxx', 'maximum x', value = 1000000),
    checkboxInput("densityplot", "Add density plot?")),


    # Show a plot of the generated distribution
    mainPanel(

      tabsetPanel(type = "tabs",
                  tabPanel("Population", plotOutput("distPlot")),
                  tabPanel("Private Dwellings", plotOutput("dist1Plot"))
      )

    )

))
)
```

```
library(shiny)

# Define server logic required to draw a histogram
shinyServer(function(input, output) {

  output$distPlot <- renderPlot({

    #Use those radioButtons to select the data to use
    #Note that the input was called "dataSource" in the ui.R file
    #Here it is an element from a list.
    #I extract it using input$nameOfThing
    if (input$dataSource == "DataFile2006") {
      x = read.csv("pop2006.csv")
    } else {
      if(input$dataSource == "DataFile2011"){
        x = read.csv("pop2011.csv")
      }else{
        #This means do not use any data.  This will break things.
        x=NULL
      }
    }
    #I didn't need to use if{}else{if{}else{}}
    # but I do so to show how to use multiple if else statements.


    # generate bins based on input$bins from ui.R
    a <- x[-1, 3]
    x <- x[-1, 3]
    for (i in 1:length(a))
    {

      if(a[i] <= input$minx || a[i] >= input$maxx)
```

```
  # generate bins based on input$bins from ui.R
  a <- x[-1, 3]
  x <- x[-1, 3]
  for (i in 1:length(a))
  {

    if(a[i] <= input$minx || a[i] >= input$maxx)
    {
      a[i] = NA
    }
  }
  a <- a[!is.na(a)]
  bins <- seq(min(x), max(x), length.out = input$bins + 1)

  # draw the histogram with the specified number of bins
  hist(a, breaks = bins, col = 'darkgray', border = 'white',
       main = "Histogram of Population", ylab = "Number of Provinces", xlab = "Population")

  if(input$densityplot)
  {
    hist(a, breaks = bins,prob = TRUE, col = 'darkgray', border = 'white',
         main = "Histogram of Population", ylab = "Frequency", xlab = "Population")
    lines(density(a),lwd = 2, col = "red")
  }

})
output$dist1Plot <- renderPlot({

  #Use those radioButtons to select the data to use
  #Note that the input was called "dataSource" in the ui.R file
  #Here it is an element from a list.
  #I extract it using input$nameOfThing
  if (input$dataSource == "DataFile2006") {
    x = read.csv("pop2006.csv")
```

```
})
output$dist1Plot <- renderPlot({

  #Use those radioButtons to select the data to use
  #Note that the input was called "dataSource" in the ui.R file
  #Here it is an element from a list.
  #I extract it using input$nameOfThing
  if (input$dataSource == "DataFile2006") {
    x = read.csv("pop2006.csv")
  } else {
    if(input$dataSource == "DataFile2011"){
      x = read.csv("pop2011.csv")
    }else{
      #This means do not use any data.  This will break things.
      x=NULL
    }
  }
  #I didn't need to use if{}else{if{}else{}}
  # but I do so to show how to use multiple if else statements.


  # generate bins based on input$bins from ui.R
  a <- x[-1, 4]
  x <- x[-1, 4]
  for (i in 1:length(a))
  {

    if(a[i] <= input$minx || a[i] >= input$maxx)
    {
      a[i] = NA
    }
  }
  a <- a[!is.na(a)]
```

```
        x=NULL
      }
    }
    #I didn't need to use if{}else{if{}else{}}
    # but I do so to show how to use multiple if else statements.


    # generate bins based on input$bins from ui.R
    a <- x[-1, 4]
    x <- x[-1, 4]
    for (i in 1:length(a))
    {

      if(a[i] <= input$minx || a[i] >= input$maxx)
      {
        a[i] = NA
      }
    }
    a <- a[!is.na(a)]
    bins <- seq(min(x), max(x), length.out = input$bins + 1)

    # draw the histogram with the specified number of bins
    hist(a, breaks = bins, col = 'darkgray', border = 'white',
         main = "Histogram of Population", ylab = "Number of Provinces", xlab = "Population")
    if(input$densityplot)
    {
      hist(a, breaks = bins,prob = TRUE, col = 'darkgray', border = 'white',
           main = "Histogram of Population", ylab = "Frequency", xlab = "Population")
      lines(density(a),lwd = 2, col = "red")
    }

  })

})
```


C:/Users/chopr/OneDrive/Desktop/STAT 240/shiny - Shiny
http://127.0.0.1:4000
Open in Browser
Publish
Population Data
Data Source 2006
Data Source 2011
Number of bins:
1
30
50
1 7 13 19 25 31 37 43 49
minimum x
1000
maximum x
1000000
Add density plot?
Population
Private Dwellings
Histogram of Population
Frequency
0.0e+00
1.0e-05
2.0e-05
3.0e-05
0
20000
40000
60000
80000
100000
120000
Population

```
library(shiny)

# Define UI for application that draws a histogram
shinyUI(fluidPage(

  # Application title
  titlePanel("Population Data"),
  # Include radio buttons  (Only one of these can be selected)
  # These radioButtons are used in the server as an input variable called
  # 'dataSource'
  radioButtons("dataSource", "",
               c("Data Source 2006" = "DataFile2006",
                 "Data Source 2011" = "DataFile2011")),
  # Sidebar with a slider input for number of bins
  sidebarLayout(
    sidebarPanel(
        # This is a slider for the number of bins
        #The value determined here will be used as an input in server.R
        #The minimum is 1
        #The max is 50
        #It starts at a value of 30
        #by default it will grow by 1 but here I specify that it will step by 2
        #
        # The format is
        #sliderInput(VariableName, Stuff2WriteInTheWebPage, MinimumValue,MaxValue,Default,StepIncrement)
        sliderInput("bins",  #This is the variable name, the value of which is determined by the slider.
            "Number of bins:",#Label on the html page
            min = 1,#minimum for the slider
            max = 50,#maximum for the slider
            value = 30,#default value
            step=2),#slider increments
    numericInput('minx', 'minimum x', value = 1000),
    numericInput('maxx', 'maximum x', value = 1000000),
    checkboxInput("densityplot", "Add density plot?")),
```

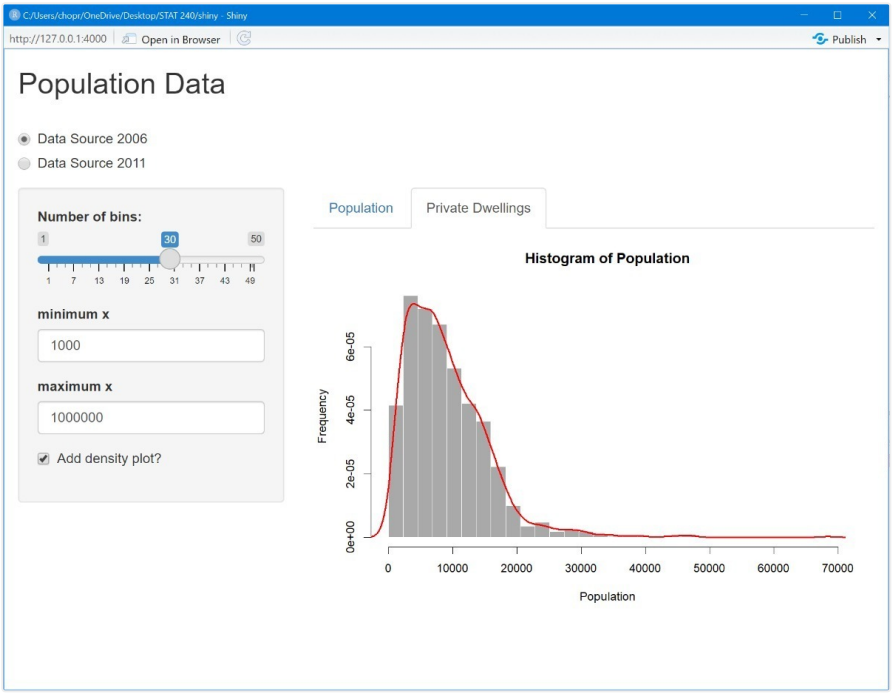
```
    sidebarPanel(
        # This is a slider for the number of bins
        #The value determined here will be used as an input in server.R
        #The minimum is 1
        #The max is 50
        #It starts at a value of 30
        #by default it will grow by 1 but here I specify that it will step by 2
        #
        # The format is
        #sliderInput(VariableName, Stuff2WriteInTheWebPage, MinimumValue,MaxValue,Default,StepIncrement)
        sliderInput("bins",  #This is the variable name, the value of which is determined by the slider.
            "Number of bins:",#Label on the html page
            min = 1,#minimum for the slider
            max = 50,#maximum for the slider
            value = 30,#default value
            step=2),#slider increments
    numericInput('minx', 'minimum x', value = 1000),
    numericInput('maxx', 'maximum x', value = 1000000),
    checkboxInput("densityplot", "Add density plot?")),


    # Show a plot of the generated distribution
    mainPanel(

      tabsetPanel(type = "tabs",
                  tabPanel("Population", plotOutput("distPlot")),
                  tabPanel("Private Dwellings", plotOutput("dist1Plot"))
      )

    )

))
)
```

```
library(shiny)

# Define server logic required to draw a histogram
shinyServer(function(input, output) {

  output$distPlot <- renderPlot({

    #Use those radioButtons to select the data to use
    #Note that the input was called "dataSource" in the ui.R file
    #Here it is an element from a list.
    #I extract it using input$nameOfThing
    if (input$dataSource == "DataFile2006") {
      x = read.csv("pop2006.csv")
    } else {
      if(input$dataSource == "DataFile2011"){
        x = read.csv("pop2011.csv")
      }else{
        #This means do not use any data.  This will break things.
        x=NULL
      }
    }
    #I didn't need to use if{}else{if{}else{}}
    # but I do so to show how to use multiple if else statements.


    # generate bins based on input$bins from ui.R
    a <- x[-1, 3]
    x <- x[-1, 3]
    for (i in 1:length(a))
    {

      if(a[i] <= input$minx || a[i] >= input$maxx)
```

```
  # generate bins based on input$bins from ui.R
  a <- x[-1, 3]
  x <- x[-1, 3]
  for (i in 1:length(a))
  {

    if(a[i] <= input$minx || a[i] >= input$maxx)
    {
      a[i] = NA
    }
  }
  a <- a[!is.na(a)]
  bins <- seq(min(x), max(x), length.out = input$bins + 1)

  # draw the histogram with the specified number of bins
  hist(a, breaks = bins, col = 'darkgray', border = 'white',
       main = "Histogram of Population", ylab = "Number of Provinces", xlab = "Population")

  if(input$densityplot)
  {
    hist(a, breaks = bins,prob = TRUE, col = 'darkgray', border = 'white',
         main = "Histogram of Population", ylab = "Frequency", xlab = "Population")
    lines(density(a),lwd = 2, col = "red")
  }

})
output$dist1Plot <- renderPlot({

  #Use those radioButtons to select the data to use
  #Note that the input was called "dataSource" in the ui.R file
  #Here it is an element from a list.
  #I extract it using input$nameOfThing
  if (input$dataSource == "DataFile2006") {
    x = read.csv("pop2006.csv")
```

```
})
output$dist1Plot <- renderPlot({

  #Use those radioButtons to select the data to use
  #Note that the input was called "dataSource" in the ui.R file
  #Here it is an element from a list.
  #I extract it using input$nameOfThing
  if (input$dataSource == "DataFile2006") {
    x = read.csv("pop2006.csv")
  } else {
    if(input$dataSource == "DataFile2011"){
      x = read.csv("pop2011.csv")
    }else{
      #This means do not use any data.  This will break things.
      x=NULL
    }
  }
  #I didn't need to use if{}else{if{}else{}}
  # but I do so to show how to use multiple if else statements.


  # generate bins based on input$bins from ui.R
  a <- x[-1, 4]
  x <- x[-1, 4]
  for (i in 1:length(a))
  {

    if(a[i] <= input$minx || a[i] >= input$maxx)
    {
      a[i] = NA
    }
  }
  a <- a[!is.na(a)]
```

```
        x=NULL
      }
    }
    #I didn't need to use if{}else{if{}else{}}
    # but I do so to show how to use multiple if else statements.


    # generate bins based on input$bins from ui.R
    a <- x[-1, 4]
    x <- x[-1, 4]
    for (i in 1:length(a))
    {

      if(a[i] <= input$minx || a[i] >= input$maxx)
      {
        a[i] = NA
      }
    }
    a <- a[!is.na(a)]
    bins <- seq(min(x), max(x), length.out = input$bins + 1)

    # draw the histogram with the specified number of bins
    hist(a, breaks = bins, col = 'darkgray', border = 'white',
         main = "Histogram of Population", ylab = "Number of Provinces", xlab = "Population")
    if(input$densityplot)
    {
      hist(a, breaks = bins,prob = TRUE, col = 'darkgray', border = 'white',
           main = "Histogram of Population", ylab = "Frequency", xlab = "Population")
      lines(density(a),lwd = 2, col = "red")
    }

  })

})
```


C:/Users/chopr/OneDrive/Desktop/STAT 240/shiny - Shiny
http://127.0.0.1:4000
Open in Browser
Publish
Population Data
Data Source 2006
Data Source 2011
Number of bins:
1
30
50
1 7 13 19 25 31 37 43 49
minimum x
1000
maximum x
1000000
Add density plot?
Population
Private Dwellings
Histogram of Population
Frequency
0e+00
2e-05
4e-05
6e-05
0
10000
20000
30000
40000
50000
60000
70000
Population

This question wasn't answered