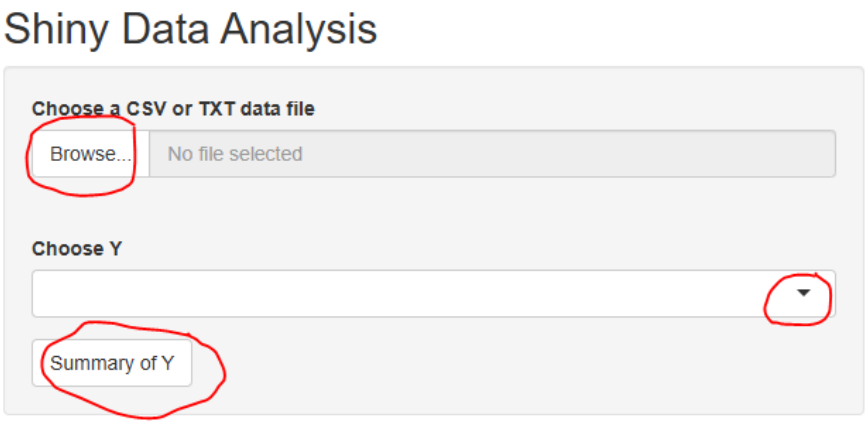
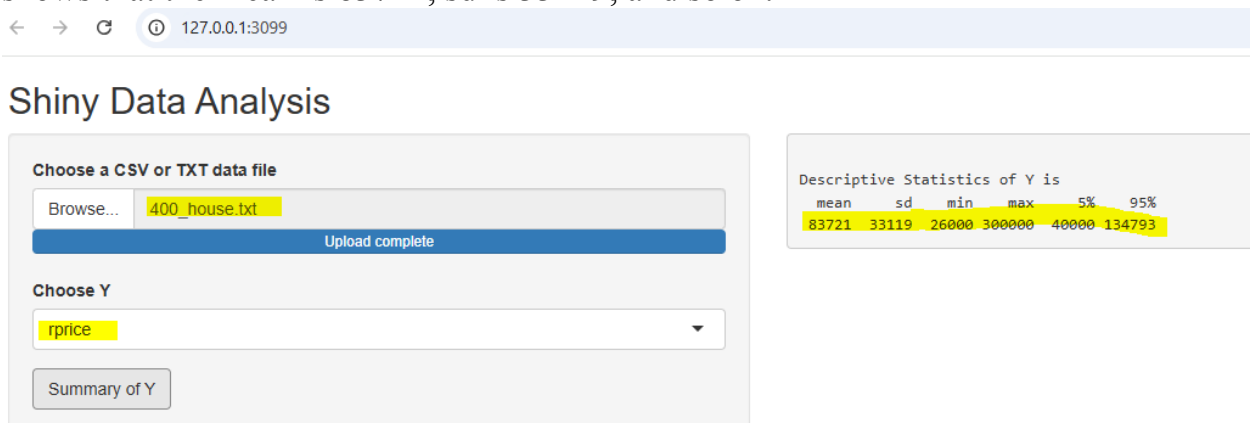
Introduction to R Shiny

(Jing Li, Miami University)

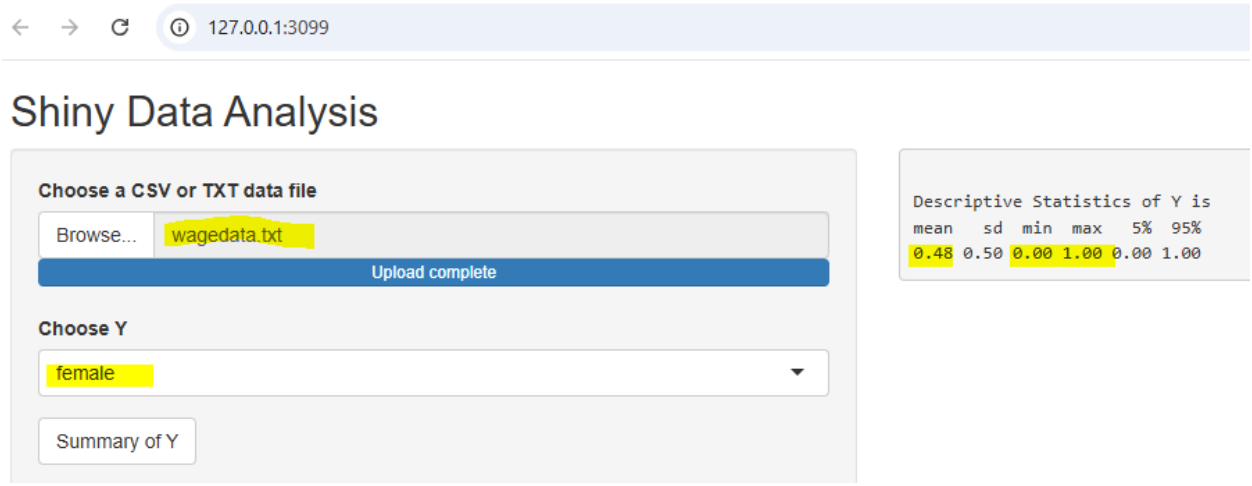
This note introduces R shiny, which aims to build interactive web applications directly from R. Shiny is designed to be easy for R users. If you're familiar with R, you can quickly start building applications without needing to learn a lot of additional web development skills (like HTML, CSS, or JavaScript). For instance, as a starting point, the webpage below asks you to click the browse button to choose a data file in either csv or txt format. Then pick a Y variable in the dataset, and summarize Y.



First, I choose 400_house.txt data, and select rprice as the Y variable, see graph below. Then R shows that the mean is 83721, sd is 33119, and so on.



R shiny is interactive in the sense that I get different results immediately if I make a change. Next I re-click the browse button to choose wagedata.txt file and re-select female as Y. The min and max values suggest this is a dummy variable. Therefore, the mean 0.48 is the proportion of female in data.



Below are R codes with the shiny library. You need to run install.packages("shiny") first.

The ui (user-interface) block defines the layout and the elements that the user interacts with. It contains all the visual components such as inputs (like text boxes, sliders, etc.), outputs (like tables, plots, or text), and layout elements (like rows, columns, and tabs). In Shiny, you typically define the UI using the fluidPage(), fluidRow(), column(), tabPanel(), or sidebarLayout()functions.

Here, I use fluidPage that contains fileInput, selectInput and actionButton. The only output is verbatimTextOutput.

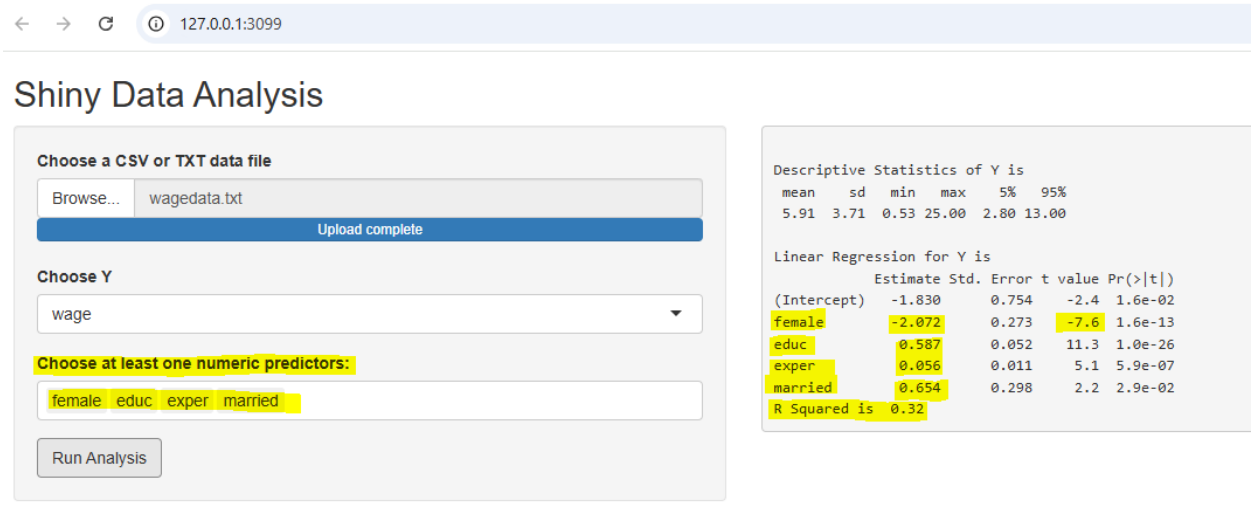
I found chatgpt extremely useful because it can give me a basic template to start with. To be honest, at the beginning I just copy codes from chatgpt and try them. The question I ask in chatgpt is "please give me R shiny codes that allow me to choose a data file, choose a variable in the file, and report mean value of that variable"

```
#####################################
# step I: read data and summarize Y #
#####################################
options(digits = 2)
library(tools)
library(shiny)

ui <- fluidPage(
  titlePanel("Shiny Data Analysis"),
  sidebarLayout(
    sidebarPanel(
      fileInput("file", "Choose a CSV or TXT data file"),
      selectInput("y", "Choose Y", ""),
      actionButton("input_sta", "Summary of Y")
    ),
    mainPanel(
      verbatimTextOutput("sta")
    )
  )
)
```

The server block defines the backend logic of the application. It processes user inputs, performs calculations, updates outputs, and manages interactions between the UI and data. For example, I create a read_data function to import the txt or csv data. Another function f is created to combine the mean, sd, min, max, and two percentiles together. Again, ask specific questions in chatgpt when issues arise. Designing shiny app involves a lot of trial and error.

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

  read_data = function(f) {
    if (tools::file_ext(f)=="csv") {
      read.csv(f)
    }
    else {
      read.table(f, sep = "\t", header=T)
    }
  }

  data <- reactive({
    req(input$file)
    file_extension <- tools::file_ext(input$file$name)
    read_data(input$file$datapath)
  })

  observe({
    updateSelectInput(session, "y", choices = names(data()))
  })

  output$sta <- renderPrint({
    req(input$input_sta)

    f = function(x) {
      return(c(mean=mean(x,na.rm=T),sd=sd(x,na.rm=T),min=min(x,na.rm=T),max=max(x,na.rm=T),quantile(x, 0.05, na.rm = T),quantile(x, 0.95, na.rm = T)))
    }

    ndata <- data()
    m = f(ndata[[input$y]])
    cat("      ","","\n")
    cat("Descriptive Statistics of Y is","","\n")
    print(m)
  })
}

shinyApp(ui, server)
```

Next I add a new component in ui so that I can choose predictor X variables. The regression results are shown on the right

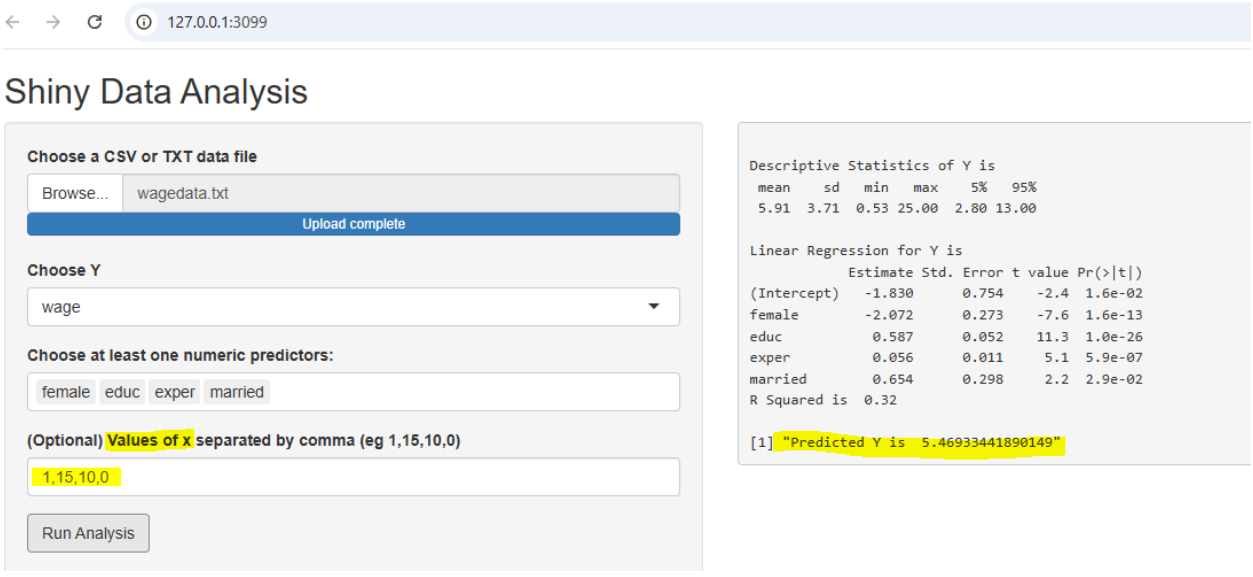


I highlight the changes in Shiny codes in yellow

```
ui <- fluidPage(
  titlePanel("Shiny Data Analysis"),
  sidebarLayout(
    sidebarPanel(
      fileInput("file", "Choose a CSV or TXT data file"),
      selectInput("y", "Choose Y", ""),
      selectInput("x", "Choose at least one numeric predictors:", "", multiple = T),
      actionButton("input_sta", "Run Analysis")
    ),
    mainPanel(
      verbatimTextOutput("sta")
    )
  )
)
```

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

  read_data = function(f) {
    if (tools::file_ext(f)=="csv") {
      read.csv(f)
    }
    else {
      read.table(f, sep = "\t", header=T)
    }
  }

  data <- reactive({
    req(input$file)
    file_extension <- tools::file_ext(input$file$name)
    read_data(input$file$datapath)
  })

  observe({
    updateSelectInput(session, "y", choices = names(data()))
    updateSelectInput(session, "x", choices = names(data()))
  })

  output$sta <- renderPrint({
    req(input$input_sta)

    f = function(x) {
      return(c(mean=mean(x,na.rm=T),sd=sd(x,na.rm=T),min=min(x,na.rm=T),max=max(x,na.rm=T),quantile(x, 0.05, na.rm = T),quantile(x, 0.95, na.rm = T)))
    }

    ndata <- data()
    m = f(ndata[[input$y]])
    cat("     ","","\n")
    cat("Descriptive Statistics of Y is","","\n")
    print(m)

    formula_lr <- as.formula(paste(input$y,"~",paste(input$x,collapse="+")))
    lr = lm(formula_lr,ndata)
    cat("     ","","\n")
    cat("Linear Regression for Y is","","\n")
    print(summary(lr)$coef)
    cat("R Squared is ",summary(lr)$r.squared,"\n")
  })
}

shinyApp(ui, server)
```

Finally, I provide the values of predictors and Shiny reports the predicted Y value



```
ui <- fluidPage(
  titlePanel("Shiny Data Analysis"),
  sidebarLayout(
    sidebarPanel(
      fileInput("file", "Choose a CSV or TXT data file"),
      selectInput("y", "Choose Y", ""),
      selectInput("x", "Choose at least one numeric predictors:", "", multiple = T),
      textInput("input_x", "(Optional) Values of x separated by comma (eg 1,15,10,0)", value=""),
      actionButton("input_sta", "Run Analysis")
    ),
    mainPanel(
      verbatimTextOutput("sta")
    )
  )
)
```

```
formula_lr <- as.formula(paste(input$y,"~",paste(input$x,collapse="+")))
lr = lm(formula_lr,ndata)
cat("      ","","\n")
cat("Linear Regression for Y is","","\n")
print(summary(lr)$coef)
cat("R Squared is ",summary(lr)$r.squared,"\n")

coe = coef(lr)
inputx = c(1,as.numeric(unlist(strsplit(input$input_x,split=","))))
cat("      ","","\n")
paste("Predicted Y is ", sum(coe*inputx))
```

You may watch this youtube video

https://www.youtube.com/watch?v=tfN10IUX9Lo&list=PLtqF5YXg7GLkxx_GGXDI_EiAvkhY9olbe

In short, **R Shiny** offers a user-friendly yet powerful platform for building interactive data-driven applications directly from R, making it an excellent tool for data scientists, statisticians, and analysts who want to present their work in an interactive format.