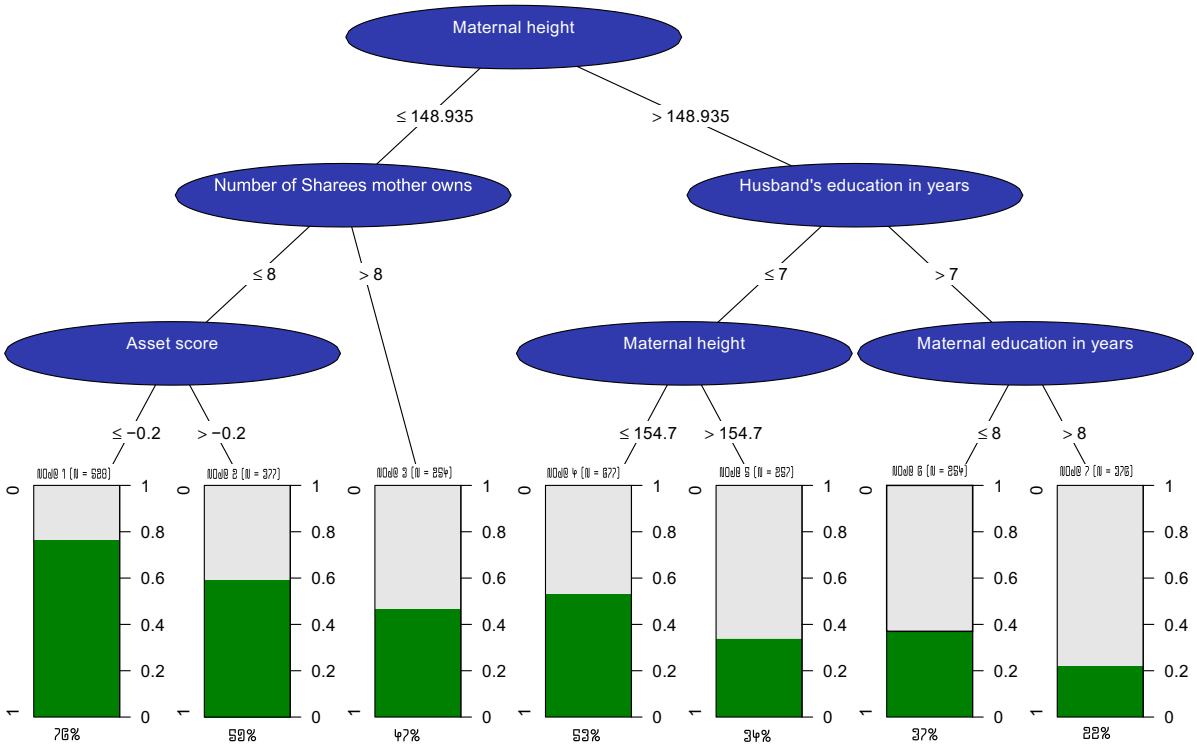
Lab1

Priya Kurian(priku577) and Andreas christopoulos(andch552)

17 September 2018

Assignment 1

Made the required changes to the tree.pdf The final one is tree_fixed.svg



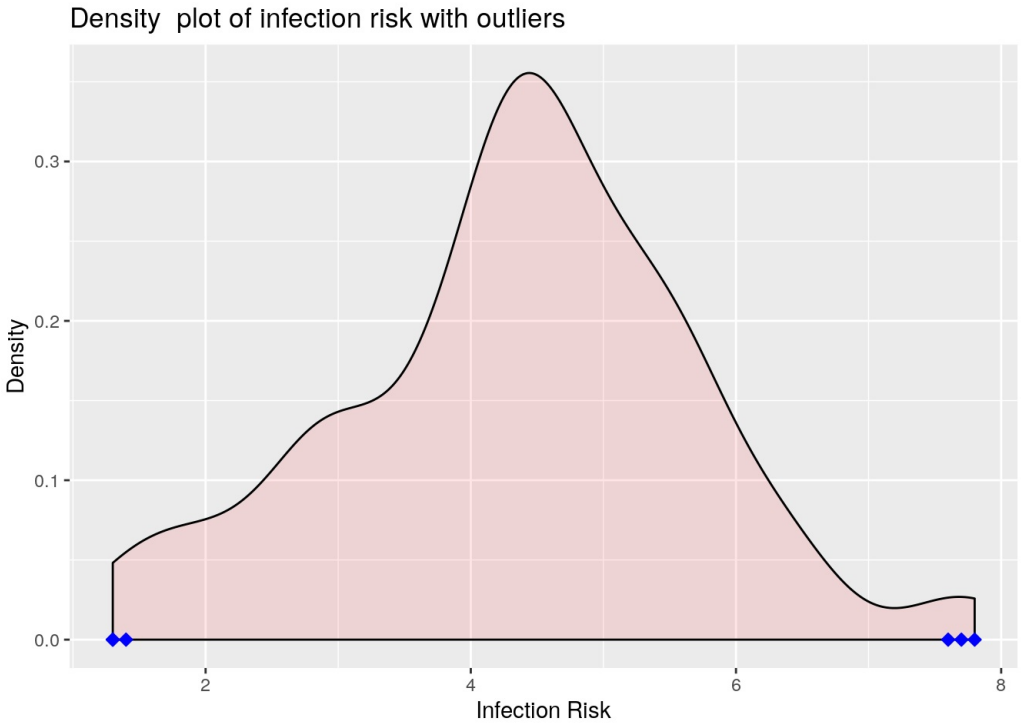
Assignment 2

Read Data from file and load libraries

creating function to return indices

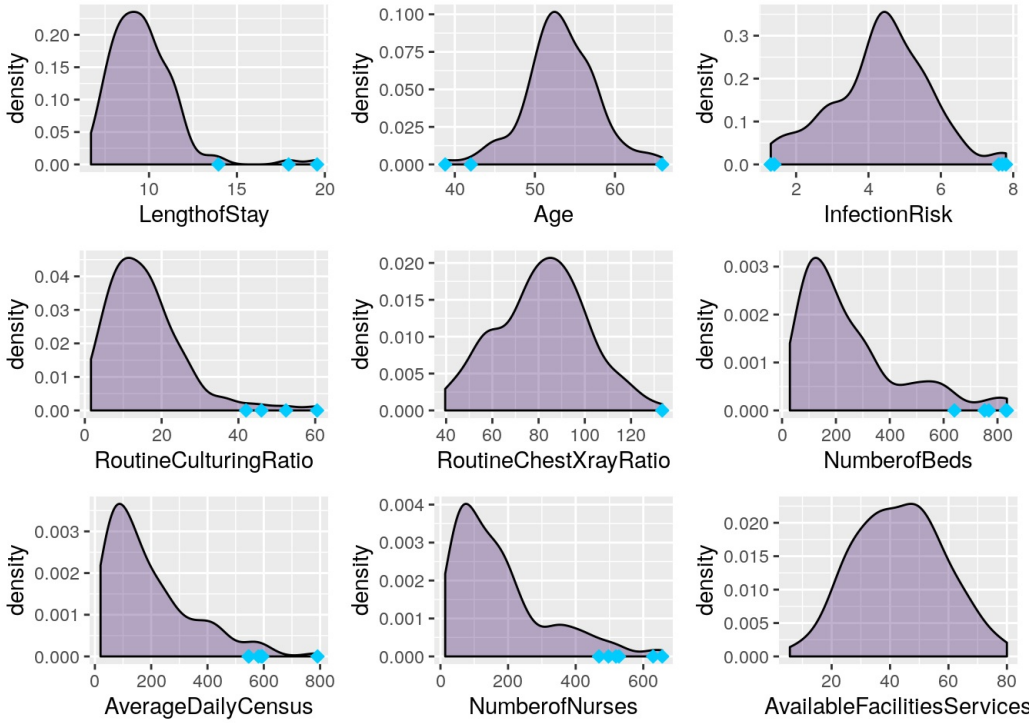
```
# Assignment 1 ------------------------------------------------------------

# 1 -------------------------------

senic=read.table("SENIC.txt")

# 2 -------------------------------

outlier_function=function(X){
  q1=as.numeric(quantile(X,0.25))
  q3=as.numeric(quantile(X,0.75))
  greater=q3+1.5*(q3-q1)
  smaller=q1-1.5*(q3-q1)
  indices=which(X>greater | X<smaller)
  return(indices)
}
```

Density Ploting of Infection Risk



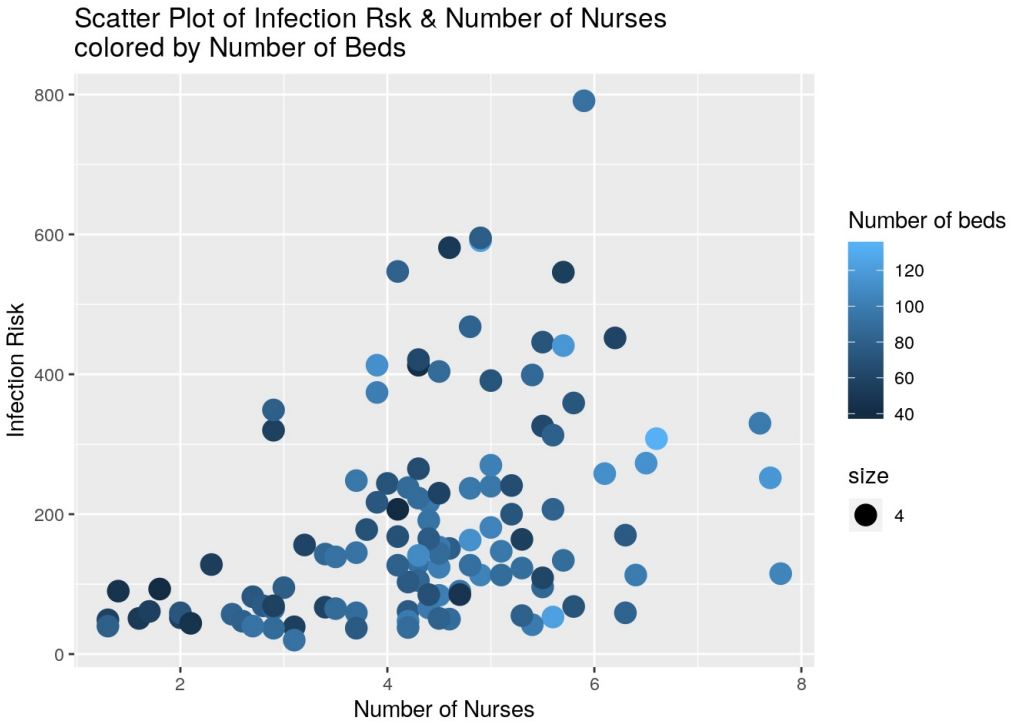
The density of infection rate has 5 outliers.the outliers are on either side of the graph. the probability of infection risk lies in a band of 2-6 and there is a very minimal probability for infection risk to cross the above mentioned limit.

Producce graphs for all other variables



We could see that all graphs exept AvailableFacilitiesServices have outliers. We feel that the graphs of Age,InfectionRisk,RoutineChestXrayRatio,AvailableFacilitiesServices follow a normal distribution. Also the LengthofStay,RoutineCulturingRatio,NumberofBeds,AverageDailyCensus,NumberofNurses follow a chi-square distribution. The outliers skews the data.

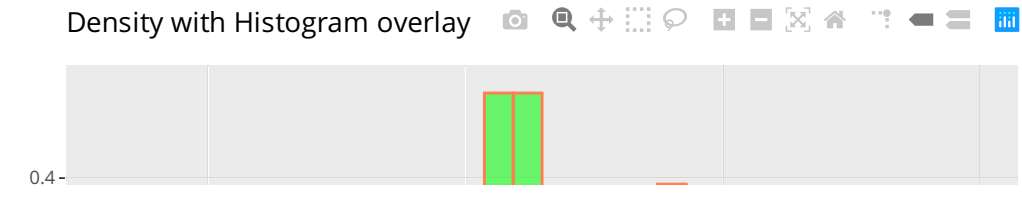
Scatter Plot

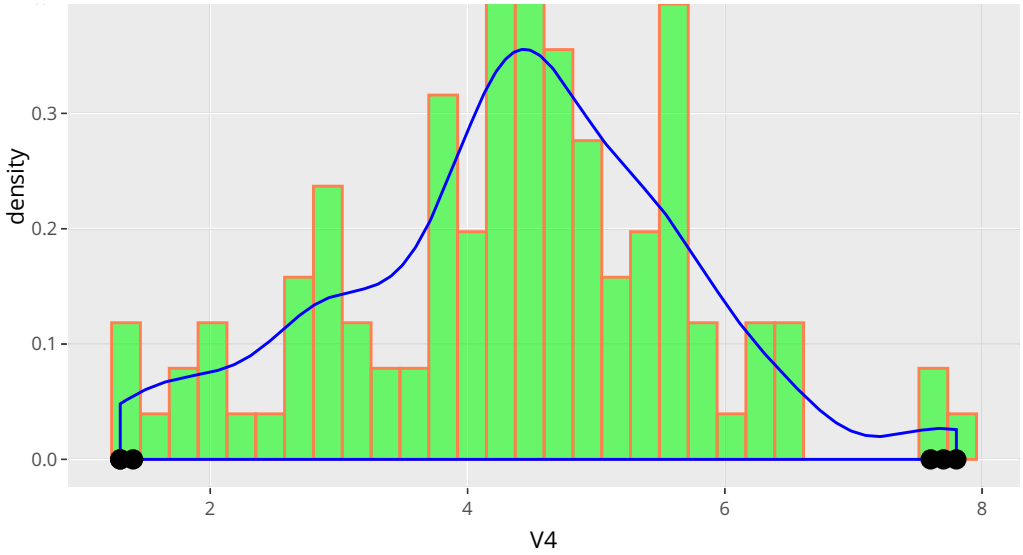


This scatter plot shows you the dependence of Infection Risk on the Number of Nurses where the points are colored by Number of Beds.Here the graph shows number of nurses and number of bed increases the infection rate.More number of beds means more number of patients can be accomadated.As the patients increases the infections also increases as the patients coming to hospital will be sick.The color scale itself is confusing as we usually think the light colours give a smaller value and dark a higher . Since the colours are very similar it is difficult to differentiate as well.

Plotly Graph

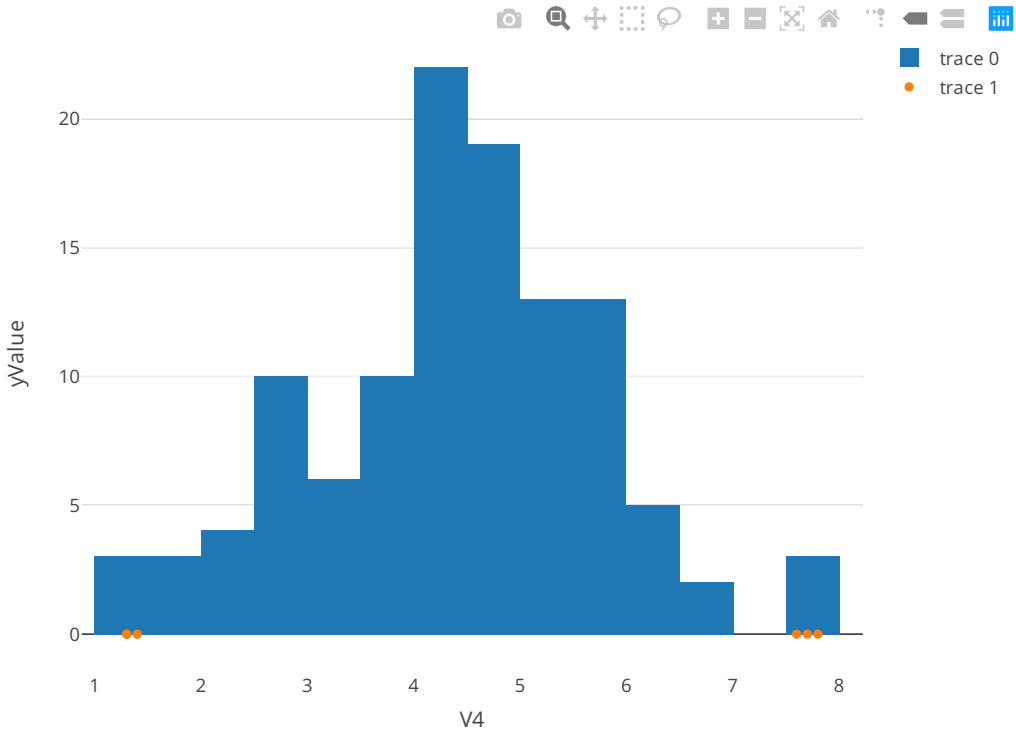
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```





THe plotly has features like zoom in,zoom out,reset the axis,autoscaling,box slection,lassso selection etc.As a whole we feel like there are lot more that we can do with plotly graphs when compared to ggplot.

Plolty Plot Made with Pipe operator



Shiny App

```
library(shiny)
library(gridExtra)
df<-read.table("SENIC.txt")
quantile_function<-function(dat)
{
  Q1<-quantile(dat,0.25)
  Q3<-quantile(dat,0.75)
  greater<-Q3+1.5*(Q3-Q1)
  smaller<-Q1-1.5*(Q3-Q1)
  result<-subset(dat,dat > greater | dat < smaller)
  indices<-which(dat %in% result)
  return(result)
}



# Define UI for application that draws a histogram

ui <- fluidPage(

  # Application title
  titlePanel("Density Plots"),

  # Sidebar with checkbox of variables
  sidebarLayout(
    sidebarPanel(
      checkboxGroupInput(inputId = "input_value",
                         label = "Select Variables:",
                         choiceValues = paste("V", c(2:7,10,11), sep = ""),
                         choiceNames = c("LengthofStay", "Age", "InfectionRisk",
                                         "RoutineCulturingRatio", "Routine Chest X-ray Ratio",
                                         "Number of Beds", "Average Daily Census",
                                         "Number of Nurses"),
                         selected = "V4"),

      sliderInput(inputId = "bw_value", label = "Bandwidth", min = 0, max = 20, value = 1, step = .1)
    ),

    # Show a plot of the generated distribution
    mainPanel(
      plotOutput("output_val")
    )
  )
)


server <- function(input, output){

  output$output_val <- renderPlot({
    # Render plot
    p <- lapply(input$input_value, function(i){ggplot() +
        geom_density(aes(df[,i]), fill = "green", bw = input$bw_value) +
        geom_point(aes(x = df[quantile_function(df[,i]),i], y = 0), shape = 23, col = "black", fill = "black") +
        labs(x = "") +
        theme_minimal()})

    if(length(p) <= 4){
      n_col <- length(p)
      n_row = 1} else{
        n_col <- 4
        n_row <- 2
      }

    marrangeGrob(p, ncol = n_col, nrow = n_row)
  })
}

# Run the application
shinyApp(ui = ui, server = server)
```

Seems like a bandwidth of 4 is a good choice.

Appendix

```
## ----echo=FALSE----------------------------------------------------------
knitr::opts_chunk$set(echo = F)
```

```
## ----message=FALSE-------------------------------------------------------

# libraries ---------------------------------------------------------------
library(ggplot2)
library(gridExtra)
library(plotly)
library(shiny)
library(gridExtra)
library(tidyverse)



## ----echo=TRUE-----------------------------------------------------------

# Assignment 1 ------------------------------------------------------------

# 1 -------------------------------

senic=read.table("SENIC.txt")

# 2 -------------------------------

outlier_function=function(X){
  q1=as.numeric(quantile(X,0.25))
  q3=as.numeric(quantile(X,0.75))
  greater=q3+1.5*(q3-q1)
  smaller=q1-1.5*(q3-q1)
  indices=which(X>greater | X<smaller)
  return(indices)
}


## ------------------------------------------------------------------------
# 3-------------------------------

outliersInf=outlier_function(senic$V4)
infRisk=data.frame("outliers"=senic$V4[outliersInf],"rep"=0)

datfrm<-data.frame(senic$V4)
names(datfrm)<-c('InfectionRisk')
ggplot_value<-ggplot(datfrm,aes(x=InfectionRisk))+geom_density(color="black",fill="red",alpha=0.1)+
  geom_point(data=infRisk,aes(x=outliers,y=rep),shape=23,color="blue",size=2,fill="blue")+
  labs(x="Infection Risk",y="Density")+ggtitle("Density  plot of infection risk with outliers")
ggplot_value



## ------------------------------------------------------------------------
# 4 -------------------------------
dt<-senic[c(-1,-8,-9)]
names(dt)<-c('LengthofStay','Age','InfectionRisk','RoutineCulturingRatio',
             'RoutineChestXrayRatio','NumberofBeds','AverageDailyCensus',
             'NumberofNurses','AvailableFacilitiesServices')

my_plots=list()

for(name in names(dt)){
  indices=outlier_function(dt[[name]])
  outliers=dt[[name]][indices]
  outliers_names=tibble("x"=outliers,"y"=0)
  my_plots[[name]]=ggplot(dt, aes_string(x =name)) + geom_density(color="black",fill="#330066",alpha=0.3)+
    geom_point(data=outliers_names,aes(x,y),size=2,color="#00CCFF",pch=23,fill="#00CCFF")
}
grid.arrange(grobs=my_plots)



## ------------------------------------------------------------------------
# 5 -------------------------------
ggplot(senic,aes(V4,V10))+geom_point(aes(size=4,color=V6))+
  ggtitle("Scatter Plot of Infection Rsk & Number of Nurses \ncolored by Number of Beds")+
  labs(x="Number of Nurses",y="Infection Risk")+
  scale_colour_continuous("Number of beds")



## ------------------------------------------------------------------------
# 6 -------------------------------
```

```
indices_infection=outlier_function(senic$V4)
outliers_infection=senic$V4[indices_infection]
outliers = tibble(x = outliers_infection, y = 0)

plot<-ggplot(senic,aes(V4))+geom_histogram(aes(y=..density..,alpha=0.7),col="coral",fill="green")+
  geom_density(col="blue")+ggtitle("Density with Histogram overlay")+
  geom_point(data = outliers, aes(x,y),size=3)

ggplotly(plot)

# using the plot from 3
#p<-ggplot_value+geom_histogram(aes(y=..density..,alpha=0.5),col="gray",fill="blue",alpha=0.5)
#ggplotly(p)



## ------------------------------------------------------------------------
# 7 -------------------------------

Outlier_indices= outlier_function(senic$V4)
Outlier_values<-senic$V4[Outlier_indices]
yValue <- rep(0,length(Outlier_values))

hisPlot <- senic %>% select(V4) %>% plot_ly(x=~V4,type="histogram")  %>%
  add_markers(x=~Outlier_values, y=~yValue)

hisPlot


## ----eval=FALSE,echo=T---------------------------------------------------
##
## library(shiny)
## library(gridExtra)
## df<-read.table("SENIC.txt")
## quantile_function<-function(dat)
## {
##   Q1<-quantile(dat,0.25)
##   Q3<-quantile(dat,0.75)
##   greater<-Q3+1.5*(Q3-Q1)
##   smaller<-Q1-1.5*(Q3-Q1)
##   result<-subset(dat,dat > greater | dat < smaller)
##   indices<-which(dat %in% result)
##   return(result)
## }
##
##
##
## # Define UI for application that draws a histogram
##
## ui <- fluidPage(
##
##   # Application title
##   titlePanel("Density Plots"),
##
##   # Sidebar with checkbox of variables
##   sidebarLayout(
##     sidebarPanel(
##       checkboxGroupInput(inputId = "input_value",
##                          label = "Select Variables:",
##                          choiceValues = paste("V", c(2:7,10,11), sep = ""),
##                          choiceNames = c("LengthofStay", "Age", "InfectionRisk",
##                                          "RoutineCulturingRatio", "Routine Chest X-ray Ratio",
##                                          "Number of Beds", "Average Daily Census",
##                                          "Number of Nurses"),
##                          selected = "V4"),
##
##       sliderInput(inputId = "bw_value", label = "Bandwidth", min = 0, max = 20, value = 1, step = .1)
##     ),
##
##     # Show a plot of the generated distribution
##     mainPanel(
##       plotOutput("output_val")
##     )
##   )
## )
##
##
## server <- function(input, output){
```

```
##
##   output$output_val <- renderPlot({
##     # Render plot
##     p <- lapply(input$input_value, function(i){ggplot() +
##         geom_density(aes(df[,i]), fill = "green", bw = input$bw_value) +
##         geom_point(aes(x = df[quantile_function(df[,i]),i], y = 0), shape = 23, col = "black", fill = "black")
+
##         labs(x = "") +
##         theme_minimal()})
##
##     if(length(p) <= 4){
##       n_col <- length(p)
##       n_row = 1} else{
##         n_col <- 4
##         n_row <- 2
##       }
##
##     marrangeGrob(p, ncol = n_col, nrow = n_row)
##   })
## }
##
## # Run the application
## shinyApp(ui = ui, server = server)
##
##
##


## ----code = readLines(knitr::purl("~/Courses/Visualization_Group_labs/Group25_Lab1/Group25_lab1.Rmd",documentati
on = 1)), echo = T, eval = F----
## NA
```