# PyData Shiny Demo

# C. Ashton Drew

# KDV Decision Analysis LLC

# ashton@kdv-decisions.com

library(ggplot2)

library(plotly)

library(leaflet)

qDat <- quakes

ui <- fluidPage(

titlePanel("pyData Shiny Demo"),

sidebarLayout(

sidebarPanel(

h3("Fiji Earthquake Data"),

selectInput("select01", "Select earthquakes based on:",

choices=c("Magnitude"="mag",

"Depth"="depth"),

selected="mag"),

conditionalPanel(condition="input.select01=='mag'",

sliderInput("sld01\_mag",

label="Show earthquakes of magnitude:",

min=min(qDat$mag), max=max(qDat$mag),

value=c(min(qDat$mag),max(qDat$mag)), step=0.1)

),

conditionalPanel(condition="input.select01=='depth'",

sliderInput("sld02\_depth",

label="Show earthquakes of depth:",

min=min(qDat$depth), max=max(qDat$depth),

value=c(min(qDat$depth),max(qDat$depth)), step=5)

),

plotlyOutput("hist01")

),

mainPanel(

leafletOutput("map01"),

dataTableOutput("table01")

)

)

)

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

qSub <- reactive({

if (input$select01=="mag"){

subset <- qDat[qDat$mag>=input$sld01\_mag[1] & qDat$mag<=input$sld01\_mag[2],]

}else{

subset <- qDat[qDat$depth>=input$sld02\_depth[1] & qDat$depth<=input$sld02\_depth[2],]

}

subset

})

output$hist01 <- renderPlotly({

ggplot(data=qSub(), aes(x=stations))+

geom\_histogram(binwidth=5)+

xlab("Number of Reporting Stations")+

xlim(min(qDat$stations), max(qDat$stations))+

ylab("Count")+

ggtitle("Earthquakes near Fiji")

})

output$table01 <- renderDataTable({

qSub()

})

output$map01 <- renderLeaflet({

pal <- colorNumeric("YlOrRd", domain=c(min(quakes$mag), max(quakes$mag)))

qMap <- leaflet(data = qSub()) %>%

addTiles() %>%

addCircleMarkers(

radius = 2,

color = ~pal(mag),

stroke = FALSE, fillOpacity = 1, popup=~as.character(mag)) %>%

addLegend("bottomright", pal = pal, values = ~mag,

title = "Earthquake Magnitude",

opacity = 1)

qMap

})

})

shinyApp(ui = ui, server = server)