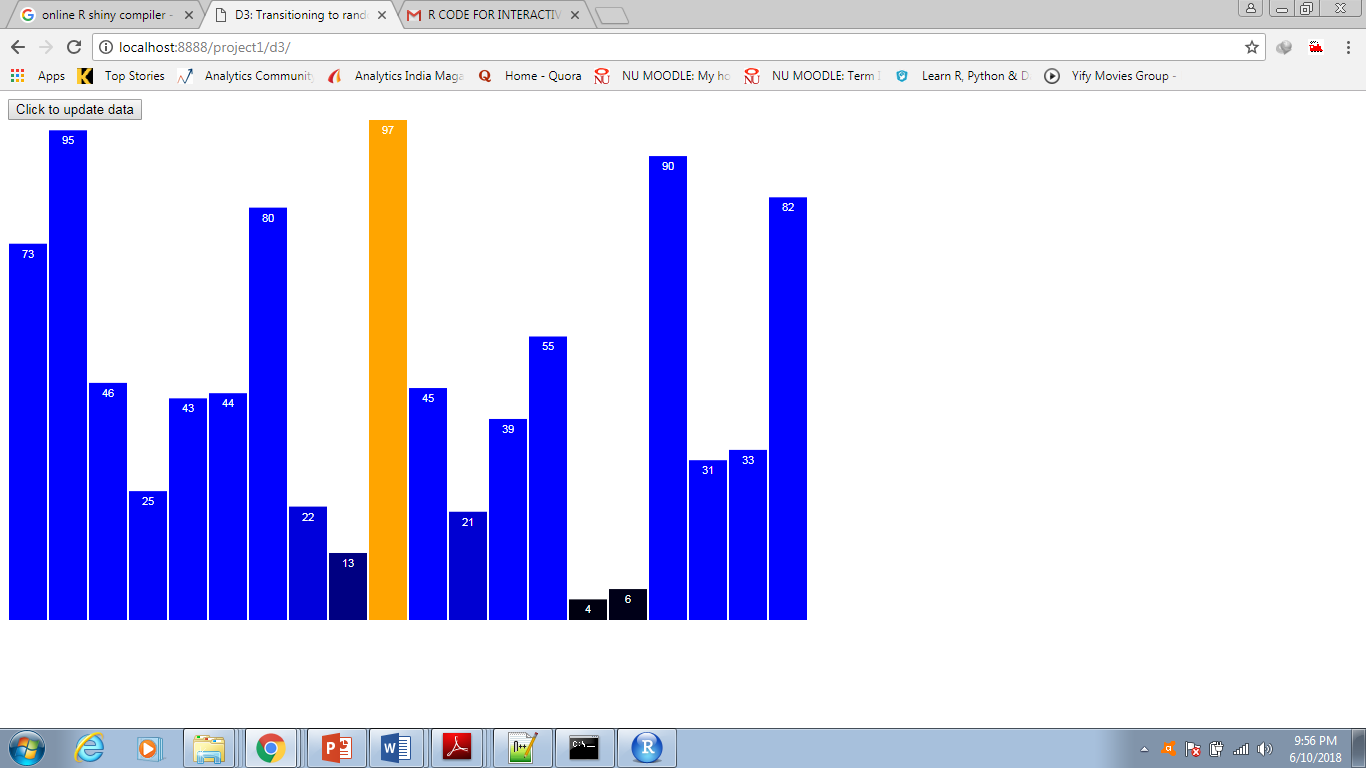
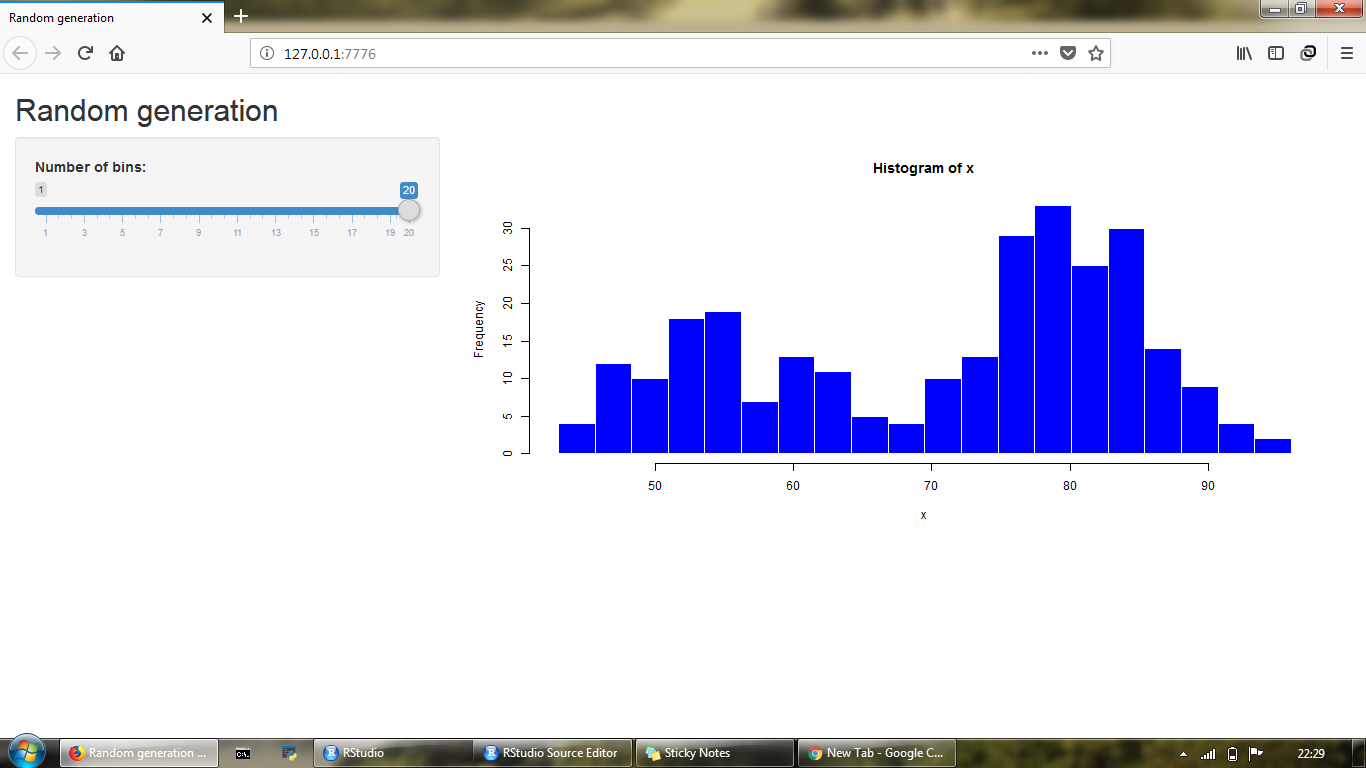
**Project on *Data Driven Documents- UI***

Lohit Khanna MB17GBA203

**Randomized histogram using html and d3.js running on localhost**



**Randomized histogram using shinyApp in R**



**Appendix**

**Code using html and d3.js run on local host**:

*<!DOCTYPE html>*

*<html lang="en">*

*<head>*

*<meta charset="utf-8">*

*<title>D3: Transitioning to randomized data values</title>*

*<script type="text/javascript" src="d3.js"></script>*

*<style type="text/css">*

*/\* No style rules here yet \*/*

*</style>*

*</head>*

*<body>*

*<button type="button">Click to update data</button>*

*<br>*

*<script type="text/javascript">*

*//Width and height*

*var w = 800;*

*var h = 500;*

*var dataset = [ 5, 10, 13, 19, 21, 25, 22, 18, 15, 13,*

*11, 12, 15, 20, 18, 17, 16, 18, 23, 25 ];*

*var xScale = d3.scaleBand()*

*.domain(d3.range(dataset.length))*

*.rangeRound([0, w])*

*.paddingInner(0.05);*

*var yScale = d3.scaleLinear()*

*.domain([0, d3.max(dataset)])*

*.range([0, h]);*

*//Create SVG element*

*var svg = d3.select("body")*

*.append("svg")*

*.attr("width", w)*

*.attr("height", h);*

*//Create bars*

*svg.selectAll("rect")*

*.data(dataset)*

*.enter()*

*.append("rect")*

*.attr("x", function(d, i) {*

*return xScale(i);*

*})*

*.attr("y", function(d) {*

*return h - yScale(d);*

*})*

*.attr("width", xScale.bandwidth())*

*.attr("height", function(d) {*

*return yScale(d);*

*})*

*.attr("fill", function(d) {*

*return "rgb(0, 0, " + Math.round(d \* 4) + ")";*

*})*

*.on("mouseover", function() {*

*d3.select(this)*

*.attr("fill", "orange");*

*})*

*.on("mouseout", function(d) {*

*d3.select(this)*

*.transition()*

*.duration(250)*

*.attr("fill", "rgb(0, 0, " + (d \* 10) + ")");*

*});*

*//Create labels*

*svg.selectAll("text")*

*.data(dataset)*

*.enter()*

*.append("text")*

*.text(function(d) {*

*return d;*

*})*

*.attr("text-anchor", "middle")*

*.attr("x", function(d, i) {*

*return xScale(i) + xScale.bandwidth() / 2;*

*})*

*.attr("y", function(d) {*

*return h - yScale(d) + 14;*

*})*

*.attr("font-family", "sans-serif")*

*.attr("font-size", "11px")*

*.attr("fill", "white");*

*//On click, update with new data*

*d3.select("button")*

*.on("click", function() {*

*//New values for dataset*

*var numValues = dataset.length; //Count original length of dataset*

*var maxValue = 100; //Highest possible new value*

*dataset = []; //Initialize empty array*

*for (var i = 0; i < numValues; i++) { //Loop numValues times*

*var newNumber = Math.ceil(Math.random() \* maxValue); //New random integer (0-100)*

*dataset.push(newNumber); //Add new number to array*

*}*

*//Update scale domain*

*//Recalibrate the scale domain, given the new max value in dataset*

*yScale.domain([0, d3.max(dataset)]);*

*//Update all rects*

*svg.selectAll("rect")*

*.data(dataset)*

*.transition()*

*.delay(function(d, i) {*

*return i / dataset.length \* 1000;*

*})*

*.duration(500)*

*.attr("y", function(d) {*

*return h - yScale(d);*

*})*

*.attr("height", function(d) {*

*return yScale(d);*

*})*

*.attr("fill", function(d) {*

*return "rgb(0, 0, " + Math.round(d \* 4) + ")";*

*});*

*//Update all labels*

*svg.selectAll("text")*

*.data(dataset)*

*.transition()*

*.delay(function(d, i) {*

*return i / dataset.length \* 1000;*

*})*

*.duration(500)*

*.text(function(d) {*

*return d;*

*})*

*.attr("x", function(d, i) {*

*return xScale(i) + xScale.bandwidth() / 2;*

*})*

*.attr("y", function(d) {*

*return h - yScale(d) + 14;*

*});*

*});*

*</script>*

*</body>*

*</html>*

**Secondary Code using ShinyApp in R**:

*library(shiny)*

*# Define UI for application that draws a histogram*

*ui <- fluidPage(*

*# Application title*

*titlePanel("Random generation"),*

*# Sidebar with a slider input for number of bins*

*sidebarLayout(*

*sidebarPanel(*

*sliderInput("bins",*

*"Number of bins:",*

*min = 1,*

*max = 20,*

*value = 30)*

*),*

*# Show a plot of the generated distribution*

*mainPanel(*

*plotOutput("distPlot")*

*)*

*)*

*)*

*# Define server logic required to draw a histogram*

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

*output$distPlot <- renderPlot({*

*# generate bins based on input$bins from ui.R*

*x <- runif(20, 1.0, 100.0)*

*bins <- seq(min(x), max(x), length.out = input$bins + 1)*

*# draw the histogram with the specified number of bins*

*hist(x, breaks = bins, col = 'blue', border = 'white')*

*})*

*}*

*# Run the application*

*shinyApp(ui = ui, server = server)*