

# 732A98: Visualization - Cheatbook

Anubhav Dikshit (anud287)

02 Nov 2018

- Reading Data
- Data Mugging
  - Quantile Computation
- Single Plots
  - Density Plot
    - Density Plot with Outlier Highlight using GGplot2
  - Histogram Plot
    - Histogram plot with Outlier Highlight using Plotly
  - Scatter Plot
    - Simple scatter plot with colour
    - Scatter Plot with Discretization (split a variable into classes)
    - Scatter plot size varied
    - Scatter plot angle varied
- Multiple Plots
  - Density Plots
    - Density Plot with Outlier Highlight
- Shiny

```
# Loading required R packages
library(ggplot2)
library(plotly)
library(shiny)
library(ggdistextra)
library(ggsci)
library(MASS)
library(webshot)
```

## Reading Data

## Data Mugging

## Quantile Computation

```
get_outliers <- function(x){
  quantile_values = quantile(x, probs = c(0.25, 0.75))
  q1 = quantile_values["25%"]
  q3 = quantile_values["75%"]

  return(c(which((x > (q3+1.5*(q3-q1))))), which(x < (q1-1.5*(q3-q1))))
}
```

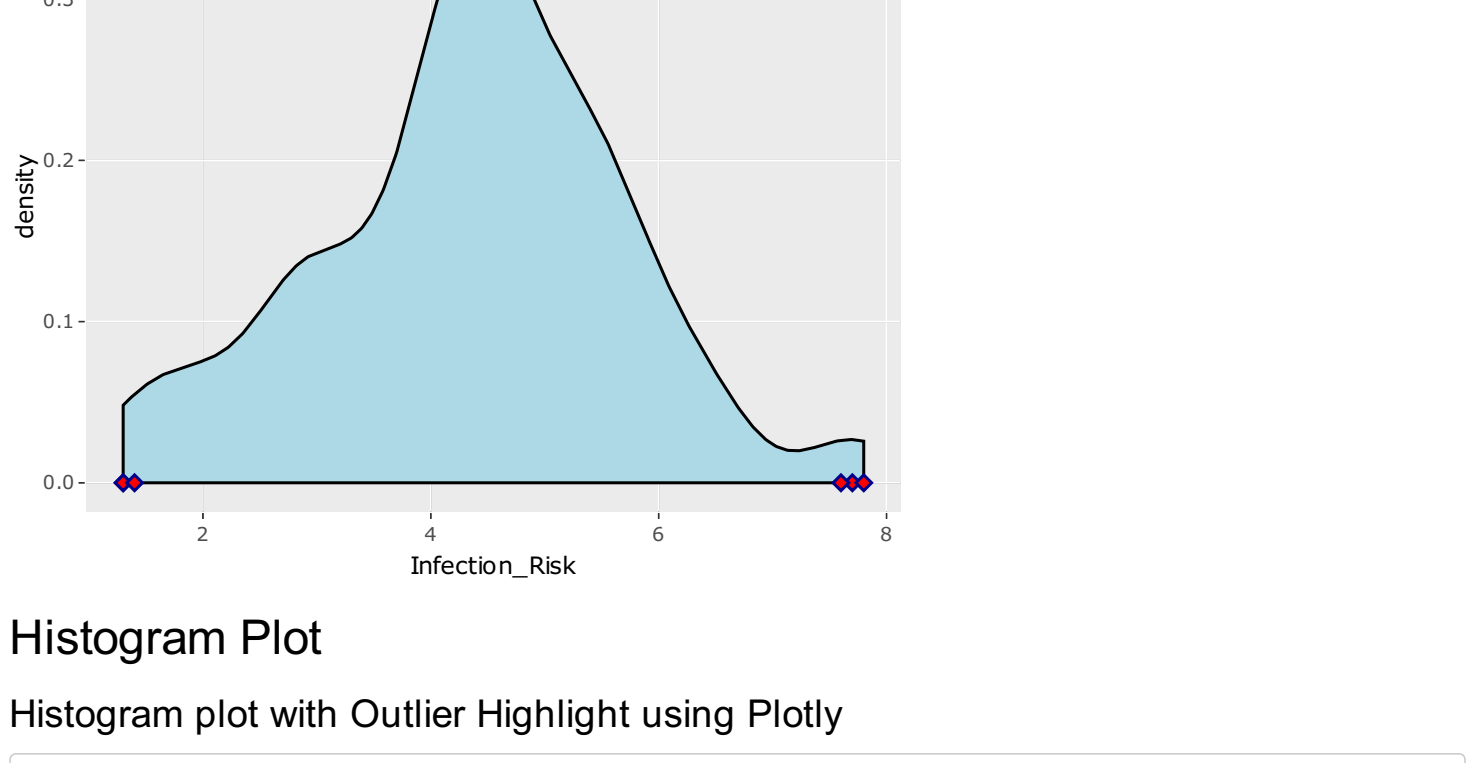
## Single Plots

### Density Plot

#### Density Plot with Outlier Highlight using GGplot2

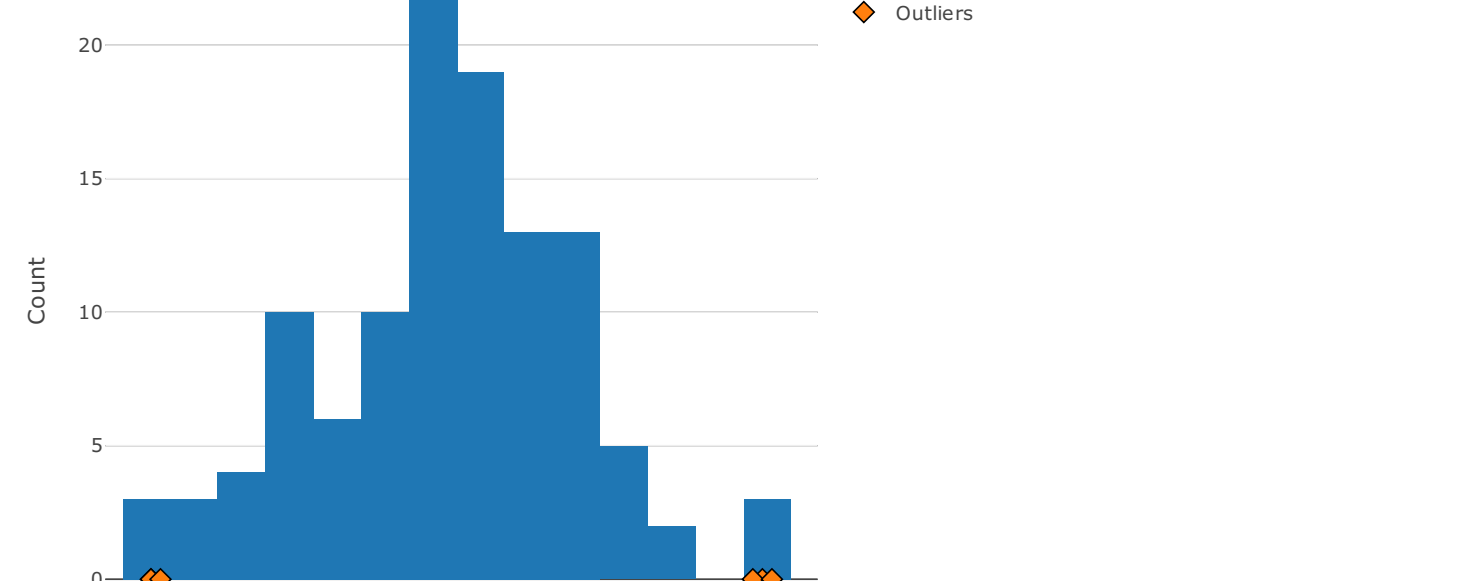
```
density_plot_infection_risk = ggplot(senic_data) +
  ggtitle("Density plot of Infection_Risk") +
  geom_density(aes(x=Infection_Risk), fill = "lightblue") +
  geom_point(data=senic_data[get_outliers(senic_data$Infection_Risk)],
    aes(x=Infection_Risk, y=0, colour="Outliers"),
    shape=23, size=2, fill="red") +
  scale_color_manual(values = c("darkblue","black")) +
  labs(colour="Legend") +
  theme(plot.title = element_text(hjust = 0.5), legend.position = "right")

density_plot_infection_risk
```



#### Density Plot with Outlier Highlight using Plotly (converting from ggplot2)

```
x <- ggplotly(p=density_plot_infection_risk)
x
```



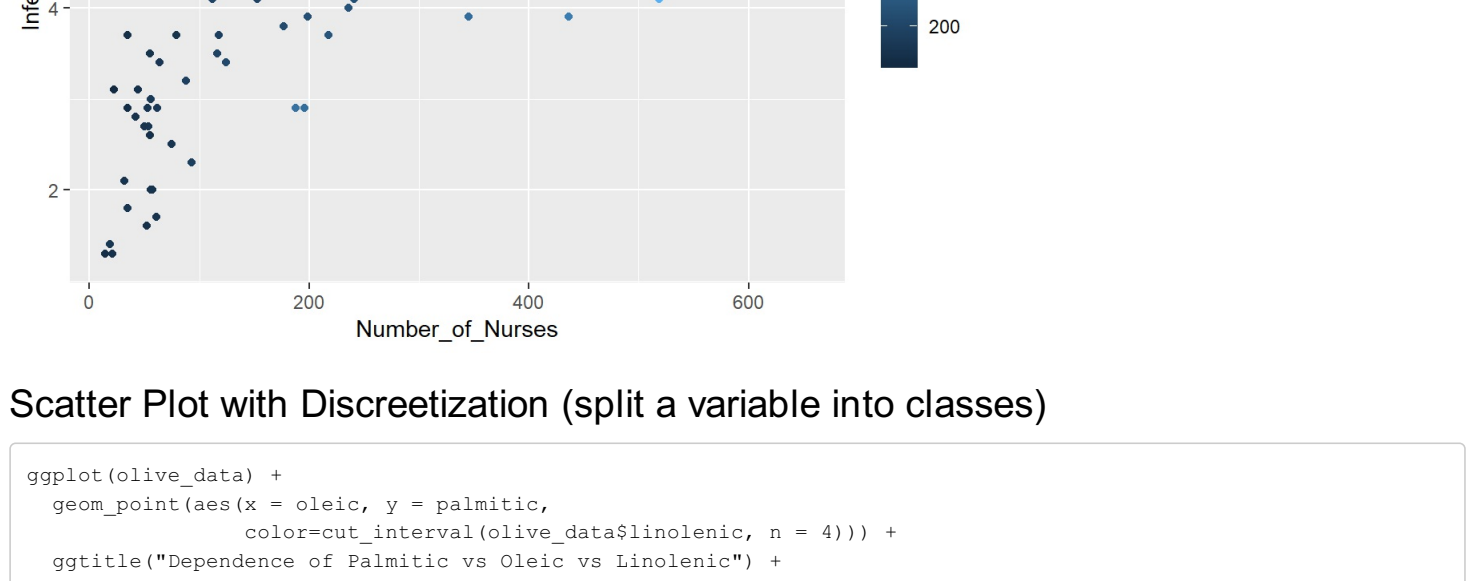
## Histogram Plot

### Histogram plot with Outlier Highlight using Plotly

```
outliers = senic_data[get_outliers(senic_data$Infection_Risk),c("Infection_Risk")]
senic_data$zero = 0

p <- plot_ly(senic_data, x=Infection_Risk) %>%
  add_histogram(name="Histogram count") %>%
  filter(is.element(Infection_Risk, outliers)) %>%
  add_markers(x=Infection_Risk,y=zero, name="Outliers",
    marker_size(symbol="diamond", size=0, line = list(color="black", width=1))) %>%
  layout(title="Histogram of Infection_Risk", yaxis=list(title="Count"))

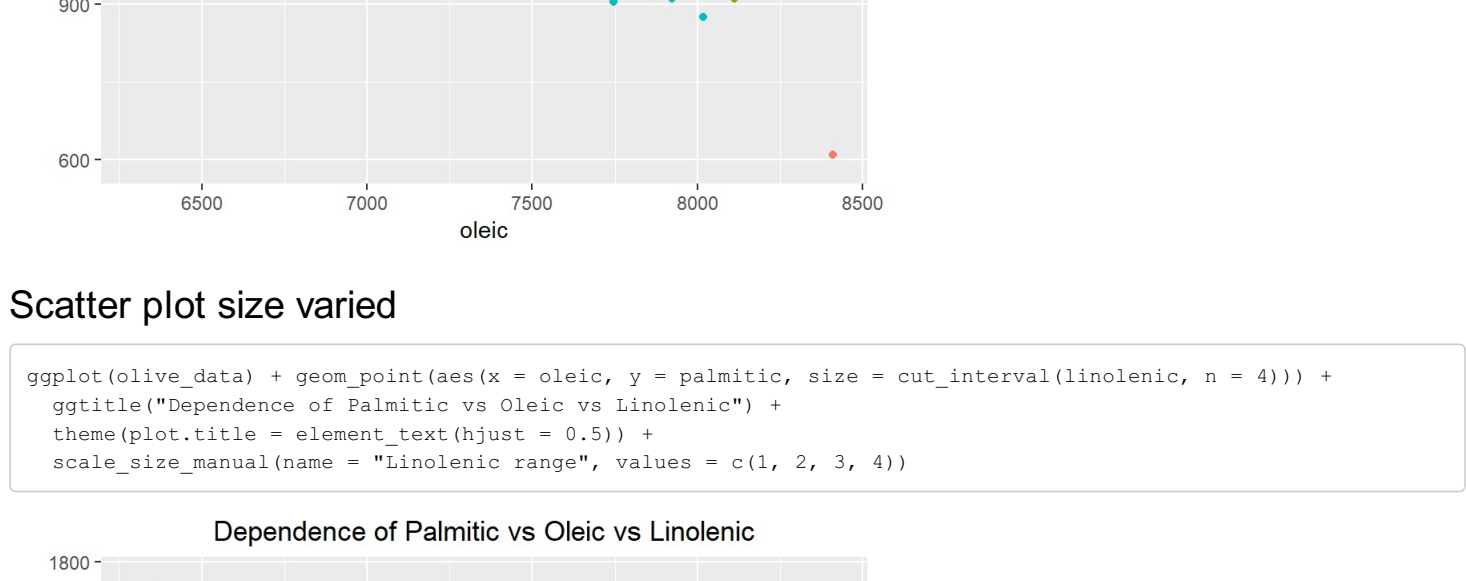
p
```



## Scatter Plot

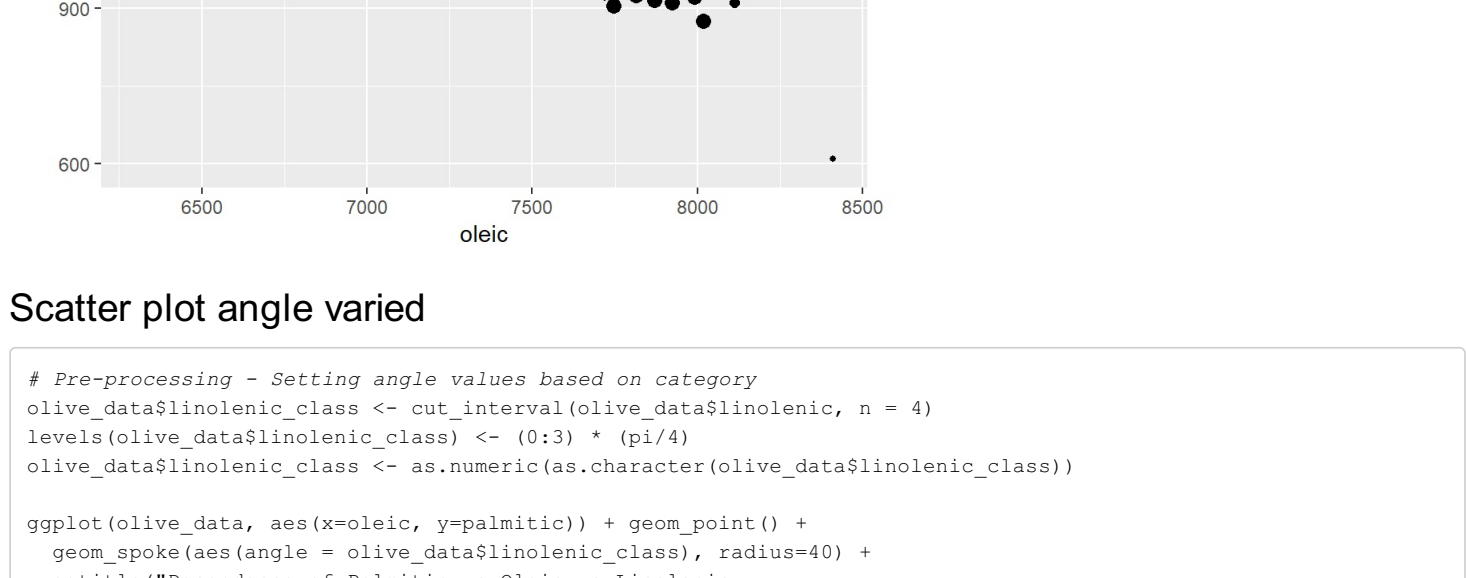
### Simple scatter plot with colour

```
ggplot(senic_data) + geom_point(aes(x=Number_of_Nurses, y=Infection_Risk, color=Number_of_Beds)) +
  ggtitle("Scatterplot of Infection_Risk vs Number_of_Nurses") +
  theme(plot.title = element_text(hjust = 0.5))
```



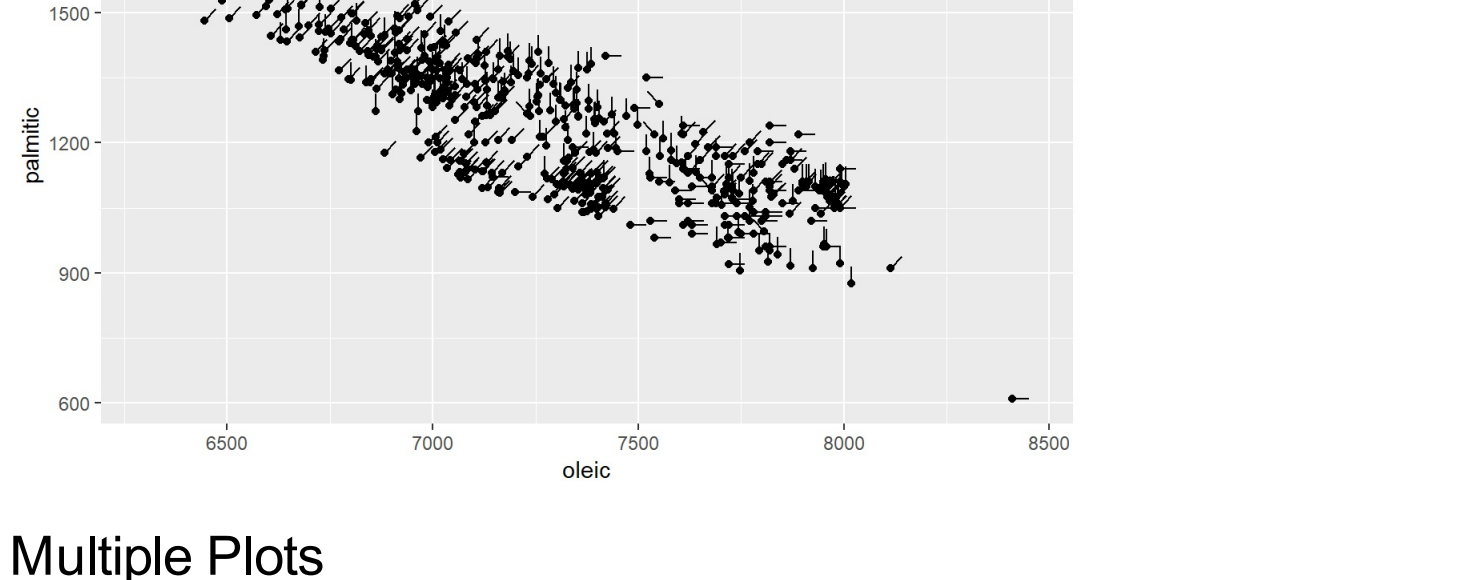
### Scatter Plot with Discretization (split a variable into classes)

```
ggplot(olive_data) +
  geom_point(aes(x = oleic, y = palmitic,
    color=cut_interval(olive_data$linolenic, n = 4))) +
  ggtitle("Dependence of Palmitic vs Oleic vs Linolenic") +
  theme(plot.title = element_text(hjust = 0.5)) +
  labs(color = "Linolenic range")
```



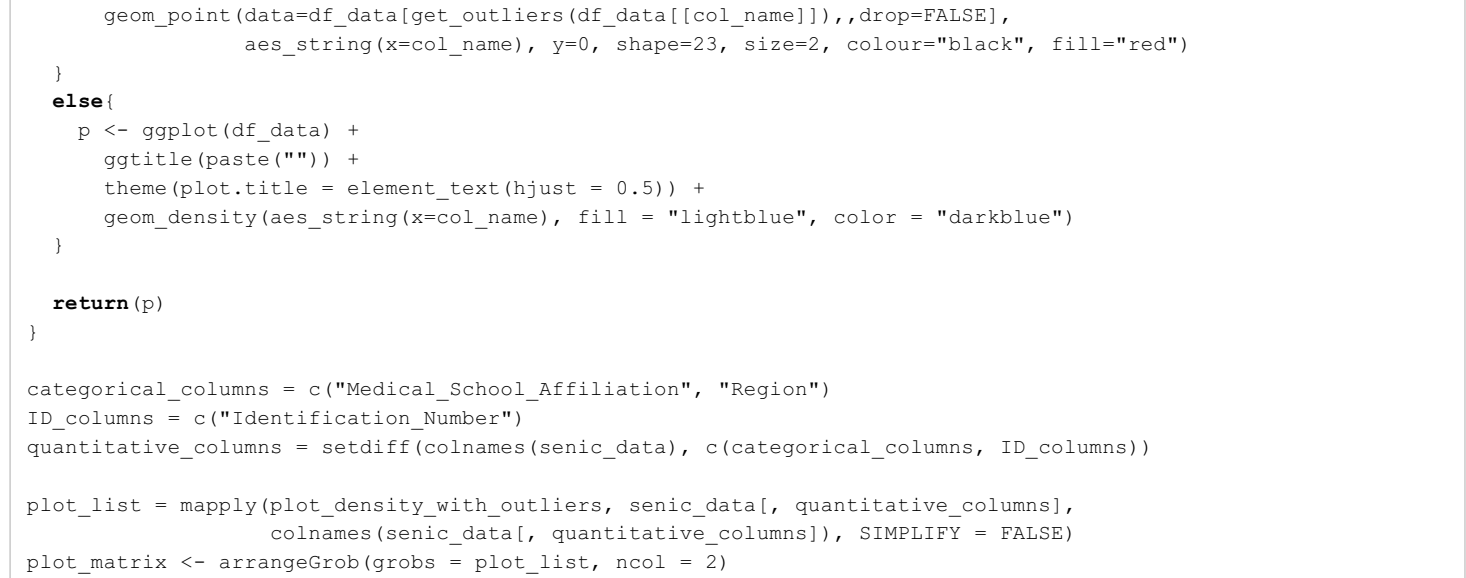
### Scatter plot size varied

```
ggplot(olive_data) + geom_point(aes(x = oleic, y = palmitic, size = cut_interval(linolenic, n = 4))) +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_size_manual(name = "Linolenic range", values = c(1, 2, 3, 4))
```



```
# Pre-processing - Setting angle values based on category
olive_data$linolenic_class <- cut_interval(olive_data$linolenic, n = 4)
linolein(olive_data$linolenic_class) <- c(0.1) + (pi/4)
olive_data$linolenic_class <- as.numeric(as.character(olive_data$linolenic_class))

ggplot(olive_data, aes(x=oleic, y=palmitic)) + geom_point() +
  geom_spoke(aes(angle = olive_data$linolenic_class, radius=40)) +
  ggtitle("Dependence of Palmitic vs Oleic vs Linolenic")
  legend
  Orientation angle of spoke : linolenic class
  0:(0,18.5], 45:(18.5,37], 90:(37,55.5], 135:(55.5,90)
  theme(plot.title = element_text(hjust = 0.5))
```



## Multiple Plots

## Density Plots

### Density Plot with Outlier Highlight

```
plot_density_with_outliers <- function(var_data, col_name){
  p <- NULL
  df_data = setNames(data.frame(var_data,col_name)
    if(length(get_outliers(df_data[[col_name]])) > 0){
      p <- ggplot(df_data) +
        geom_density(aes_string(x=col_name, fill = "lightblue", color = "darkblue") +
          geom_point(data=df_data[get_outliers(df_data[[col_name]]),drop=FALSE],
            aes_string(x=col_name, y=0, shape=23, size=2, colour="black", fill="red")
        )
    }
    else{
      p <- ggplot(df_data) +
        ggtitle(paste("Density plot of ", col_name)) +
        theme(plot.title = element_text(hjust = 0.5)) +
        geom_density(aes_string(x=col_name, fill = "lightblue", color = "darkblue", bw=bw) +
          geom_point(data=df_data[get_outliers(df_data[[col_name]]),
            aes_string(x=col_name, y=0, shape=23, size=2, colour="black", fill="red")
        )
    }
  }
  return(p)
}
```

```
categorical_columns = c("Medical_School_Affiliation", "Region")
ID_columns = c("Identification_Number")
quantitative_columns = setdiff(colnames(senic_data), c(categorical_columns, ID_columns))

plot_list = map2(plot_density_with_outliers, senic_data, quantitive_columns,
  colnames(senic_data), quantitative_columns), SIMPLIFY = FALSE)
plot_matrix <- arrangeGrob(grobs = plot_list, ncol = 2)
grid.arrange(plot_matrix, sepact="FUS", top="Density plots of SENIC data variables")
```



## Shiny

```
##UI component
ui <- fluidPage(
  sliderInput(inputId="bw_value", label="Choose bandwidth size", value=4.5, min=0.1, max=80),
  checkboxGroupInput("selected_variables", "Variables to show:", quantitative_columns, inline=TRUE),
  plotOutput("densaPlot", height = "650px")
)

plot_density_with_outliers_shiny <- function(df_data, col_name, bw){
  p <- NULL
  if(length(get_outliers(senic_data[[col_name]])) > 0){
    p <- ggplot(df_data) +
      ggtitle(paste("Density plot of ", col_name)) +
      theme(plot.title = element_text(hjust = 0.5)) +
      geom_density(aes_string(x=col_name, fill = "lightblue", color = "darkblue", bw=bw) +
        geom_point(data=df_data[get_outliers(df_data[[col_name]]),
          aes_string(x=col_name, y=0, shape=23, size=2, colour="black", fill="red")
        )
    }
    else{
      p <- ggplot(df_data) +
        ggtitle(paste("Density plot of ", col_name)) +
        theme(plot.title = element_text(hjust = 0.5)) +
        geom_density(aes_string(x=col_name, fill = "lightblue", color = "darkblue", bw=bw)
    }
  }
  return(p)
}
```

```
server <- function(input, output) {
  output$densaPlot <- renderPlot({
    selected_columns = input$selected_variables
    plot_list = vector("list", length(selected_columns))

    for(i in 1:length(selected_columns)){
      plot_list[[i]] = plot_density_with_outliers_shiny(senic_data, selected_columns[i],
        bw = input$bw_value)
    }
    plot_matrix <- arrangeGrob(grobs = plot_list, ncol = 2)
    grid.arrange(plot_matrix)
  })
}
```

```
shinyApp(ui = ui, server = server, options = list(width="800px", height="900px"))
```