Link to the data set : https://www.kaggle.com/datasets/alphiree/cardiovascular-diseases-risk-prediction-dataset/data

Link to the shiny app

https://jafaralissa.shinyapps.io/Correlation/

Use Case: Specific problem descriptions of all stakeholders involved

The problem revolves around understanding the correlation between depression and specific health conditions, such as (heart diseases, skin cancer, diabetes, and arthritis), across various age groups. The primary stakeholders in this scenario are healthcare providers and patients.

Healthcare Providers

-Problem: Difficulty in identifying and addressing the psychological aspects of patients dealing with both physical health conditions and depression

-Challenge: Lack of tailored treatment plans that include psychological support based on age groups and specific health conditions.

-Impact: Risk of overlooking or underestimating the mental health needs of patients, leading to potential complications in overall treatment efficacy.

Patients

-Problem: Limited acknowledgment and integration of mental health considerations in the treatment plans for those with coexisting physical health conditions.

-Challenge: Managing the emotional toll of dealing with specific diseases, potentially affecting adherence to medical treatments.

-Impact: Reduced quality of life, potential exacerbation of depression symptoms, and inadequate support for coping with both physical and mental health challenges.

Solution: pathway to finding a solution to the problem

To address the identified problems and challenges, a comprehensive solution involves developing an integrated model that considers both physical and mental health aspects for patients dealing with specified health conditions and depression.

using the data set to gather information on depression incidence rates across age groups and specific health conditions, and then compare it with the rates of depression in the absence of the diseases.

The solution will help in :

Collaborative Care: Encourage interdisciplinary collaboration between medical and psychological healthcare providers to ensure holistic patient care

Medication Considerations: Implement protocols to consider potential drug interactions between depression medications and medications for specific health conditions

Patient Education: Provide patients with comprehensive education on the potential mental health impacts of their physical conditions, fostering awareness and proactive management

Stakeholder Involvement

Healthcare Providers : Engage in continuous education and training on the integrated model, fostering a collaborative care approach.

Patients: Actively involve patients in treatment decisions, emphasizing the importance of addressing both physical and mental health aspects.

implementation: description of the detailed steps and settings in the tools used to show how the interactions and data generation:

Due to the need of a clear data set , but also to avoid making a project that was made before, I found on Kaggle website a data set used for Cardiovascular Diseases Risk Prediction Dataset, , the data set was **already clean** , and highly used by different users.

cleaning phase was not needed.

On the other hand, I converted two columns (Age Category) and (Depression) into factors.

Link to the data set : https://www.kaggle.com/datasets/alphiree/cardiovascular-diseases-risk-prediction-dataset/data

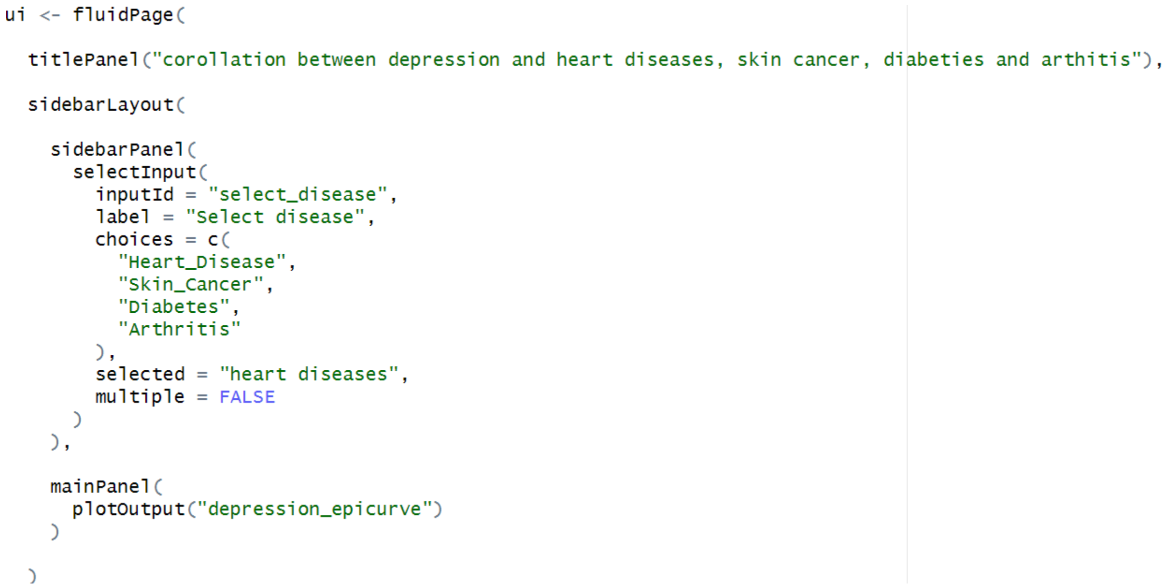
The data related to depression was formed in Yes, No . so necessary modifications were made.

(glimpse) was used repeatedly to print a concise summary of a data frame, providing a quick overview of its structure.

The plan is use Rstudio to clean the data , and isolate the needed data for my purpose as following based on what was mentioned during the lectures and using a guide example I found online.

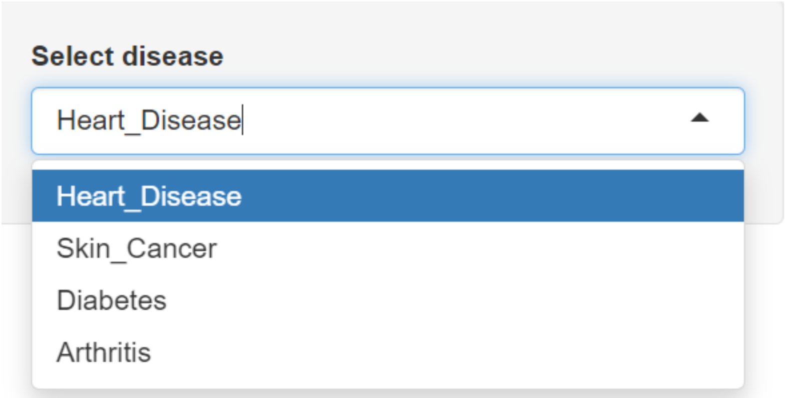
Link to guide example : https://epirhandbook.com/en/dashboards-with-shiny.html

Creating a name for the project, and then categorize each disease in the side bar, in order to make navigating on the user interface clear and easy to navigate. The following code was used.



```
ui <- fluidPage(

  titlePanel("corollation between depression and heart diseases, skin cancer, diabeties and arthitis"),

  sidebarLayout(

    sidebarPanel(
      selectInput(
        inputId = "select_disease",
        label = "Select disease",
        choices = c(
          "Heart_Disease",
          "Skin_Cancer",
          "Diabetes",
          "Arthritis"
        ),
        selected = "heart diseases",
        multiple = FALSE
      )
    ),

    mainPanel(
      plotOutput("depression_epicurve")
    )

  )
```

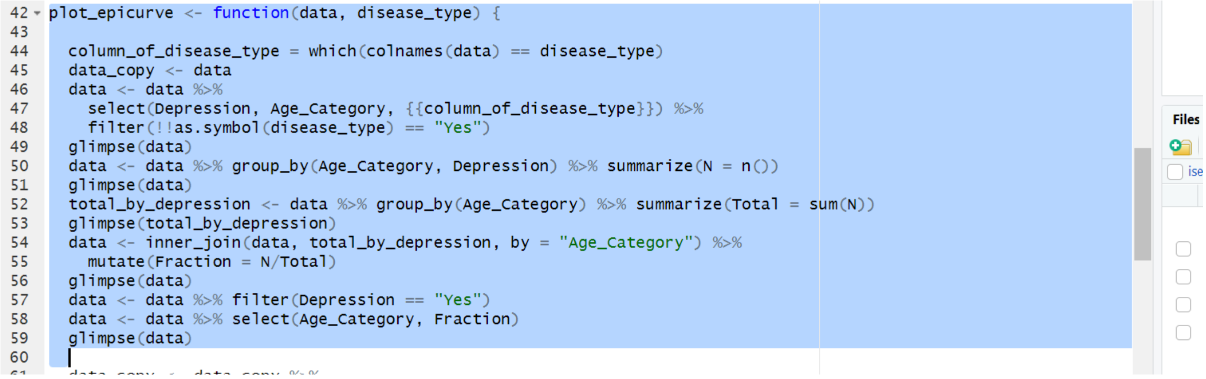
The result is :



Select disease

Heart_Disease

<u>Creating Plot epicurve :</u> it takes two parameters: `data` and `disease_type`. The purpose of this function to analyze and visualize the relationship between depression, age categories, and a specific disease type in the provided dataset.



```
plot_epicurve <- function(data, disease_type) {

  column_of_disease_type = which(colnames(data) == disease_type)
  data_copy <- data
  data <- data %>%
    select(Depression, Age_Category, {{column_of_disease_type}}) %>%
    filter(!!as.symbol(disease_type) == "Yes")
  glimpse(data)
  data <- data %>% group_by(Age_Category, Depression) %>% summarize(N = n())
  glimpse(data)
  total_by_depression <- data %>% group_by(Age_Category) %>% summarize(Total = sum(N))
  glimpse(total_by_depression)
  data <- inner_join(data, total_by_depression, by = "Age_Category") %>%
    mutate(Fraction = N/Total)
  glimpse(data)
  data <- data %>% filter(Depression == "Yes")
  data <- data %>% select(Age_Category, Fraction)
  glimpse(data)

  data_copy <- data_copy %>%
```

Explaining each code :

1. column_of_disease_type = which(colnames(data) == disease_type)`

This line finds the column index of the specified `disease_type` in the dataset.

2. `data_copy <- data`
This line creates a copy of the original dataset (`data_copy`) to preserve the original data.

3. `data <- data %>% select(Depression, Age_Category, {{column_of_disease_type}}) %>% filter(!!as.symbol(disease_type) == "Yes")`:

   This line selects only the columns relevant for the analysis (Depression, Age_Category, and the specified disease type) and filters the rows where the disease type is marked as "Yes".

4. `glimpse(data)`
This line prints a concise summary of the selected and filtered data, providing information on the structure and content of the dataset.

5. `data <- data %>% group_by(Age_Category, Depression) %>% summarize(N = n())`
This line groups the data by Age_Category and Depression, then calculates the count of occurrences (`N`) for each group.

6. `glimpse(data)`
Another summary of the grouped and summarized data.
7. `total_by_depression <- data %>% group_by(Age_Category) %>% summarize(Total = sum(N))`:

   This line calculates the total count (`Total`) for each Age_Category.

8. `glimpse(total_by_depression)`:
   A summary of the total count data.

9. `data <- inner_join(data, total_by_depression, by = "Age_Category") %>% mutate(Fraction = N/Total)`:
   This line joins the grouped data with the total count data, calculates the fraction of each group, and adds a new column (`Fraction`) representing the proportion of cases for each Age_Category.

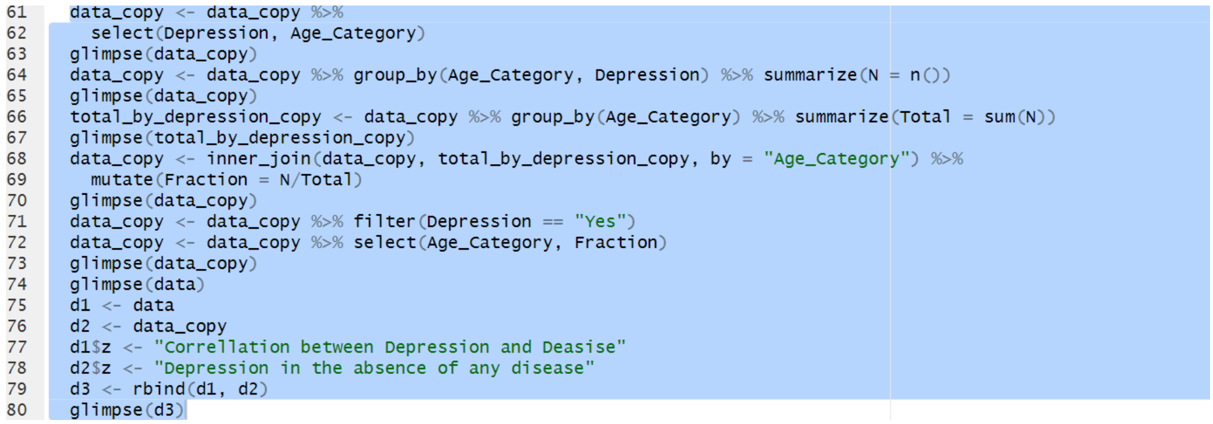
` .10glimpse(data)`: A summary of the joined and mutated data.

10. `data <- data %>% filter(Depression == "Yes")`:
    This line further filters the data to include only rows where Depression is marked as "Yes".

11. `data <- data %>% select(Age_Category, Fraction)`:
    This line selects only the Age_Category and Fraction columns.

` .13glimpse(data)`: A final summary of the filtered and selected data.

Analyzing Depression and Disease Correlation Across Ages



```
data_copy <- data_copy %>%
  select(Depression, Age_Category)
glimpse(data_copy)
data_copy <- data_copy %>% group_by(Age_Category, Depression) %>% summarize(N = n())
glimpse(data_copy)
total_by_depression_copy <- data_copy %>% group_by(Age_Category) %>% summarize(Total = sum(N))
glimpse(total_by_depression_copy)
data_copy <- inner_join(data_copy, total_by_depression_copy, by = "Age_Category") %>%
  mutate(Fraction = N/Total)
glimpse(data_copy)
data_copy <- data_copy %>% filter(Depression == "Yes")
data_copy <- data_copy %>% select(Age_Category, Fraction)
glimpse(data_copy)
glimpse(data)
d1 <- data
d2 <- data_copy
d1$z <- "Correllation between Depression and Deasise"
d2$z <- "Depression in the absence of any disease"
d3 <- rbind(d1, d2)
glimpse(d3)
```

```
data_copy <- data_copy %>%
  select(Depression, Age_Category)
glimpse(data_copy)
```

to select only the 'Depression' and 'Age_Category' columns from the copied dataset (`data_copy`) and print a summary using `glimpse`.

```
data_copy <- data_copy %>%
  group_by(Age_Category, Depression) %>%
  summarize(N = n())
glimpse(data_copy)
```

Group the copied data by 'Age_Category' and 'Depression', then calculate the count of occurrences (`N`) for each group and print a summary.

```
total_by_depression_copy <- data_copy %>%
  group_by(Age_Category) %>%
  summarize(Total = sum(N))
glimpse(total_by_depression_copy)
```

to calculate the total count (`Total`) for each 'Age_Category' in the copied dataset and print a summary.

```
data_copy <- inner_join(data_copy, total_by_depression_copy, by = "Age_Category") %>%
  mutate(Fraction = N/Total)
glimpse(data_copy)
```

to perform an inner join between the grouped data and the total count data for the copied dataset, calculate the fraction of each group, and print a summary.

```
data_copy <- data_copy %>%
  filter(Depression == "Yes")
data_copy <- data_copy %>%
  select(Age_Category, Fraction)
glimpse(data_copy)
```

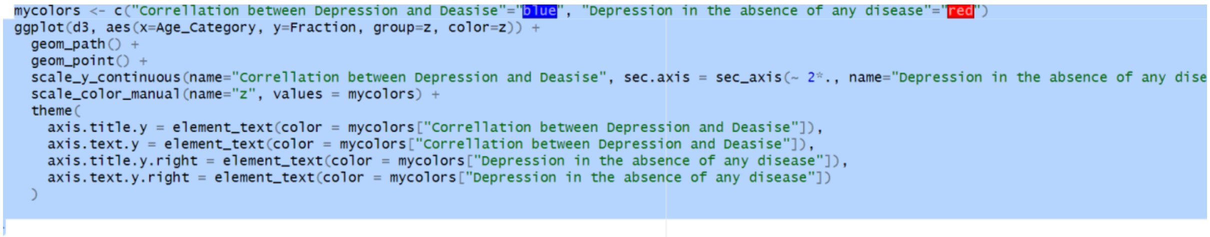
further filter the copied data to include only rows where 'Depression' is marked as "Yes," select only 'Age_Category' and 'Fraction' columns, and print a summary.

```
glimpse(data)
d1 <- data
d2 <- data_copy
d1$z <- "Correlation between Depression and Disease"
d2$z <- "Depression in the absence of any disease"
d3 <- rbind(d1, d2)
glimpse(d3)
```

Combine the original dataset (`data`) with the modified copy (`data_copy`) and assign labels ('z') to each dataset. The resulting combined dataset is stored in 'd3', and glimpses are printed.

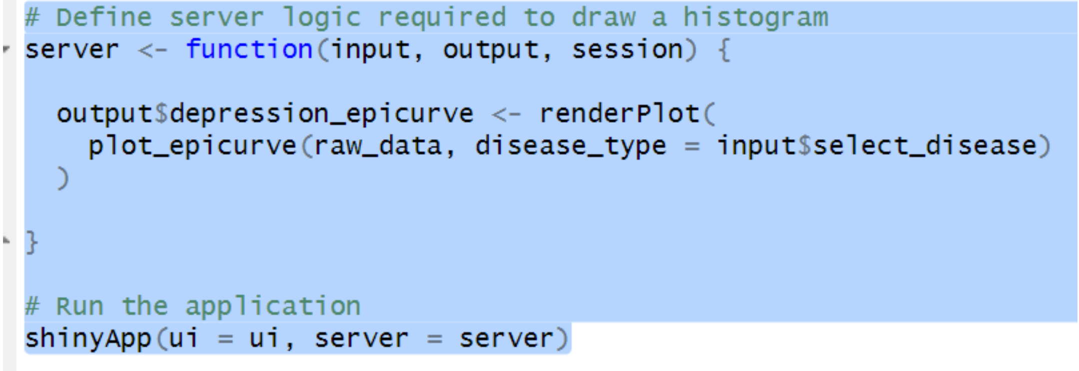
This modified code appears to perform a comprehensive analysis of the correlation between depression and a specific disease in different age categories, visualizing the results using a combined dataset (`d3`). The labels for each dataset have been adjusted to reflect the presence or absence of any disease related to depression.

Creating the plot , and define color mapping:



```
mycolors <- c("Correllation between Depression and Deasise"="blue", "Depression in the absence of any disease"="red")
ggplot(d3, aes(x=Age_Category, y=Fraction, group=z, color=z)) +
  geom_path() +
  geom_point() +
  scale_y_continuous(name="Correllation between Depression and Deasise", sec.axis = sec_axis(~ 2*., name="Depression in the absence of any dise
  scale_color_manual(name="z", values = mycolors) +
  theme(
    axis.title.y = element_text(color = mycolors["Correllation between Depression and Deasise"]),
    axis.text.y = element_text(color = mycolors["Correllation between Depression and Deasise"]),
    axis.title.y.right = element_text(color = mycolors["Depression in the absence of any disease"]),
    axis.text.y.right = element_text(color = mycolors["Depression in the absence of any disease"])
  )
```

Define server logic required to draw an epicurve plot



```
# Define server logic required to draw a histogram
server <- function(input, output, session) {

  output$depression_epicurve <- renderPlot(
    plot_epicurve(raw_data, disease_type = input$select_disease)
  )

}

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