VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

An Analysis of Student Mental Health

Winter Semester 2022-2023 (Semester 4)

Submitted To: Prof. Dr. Revathi G K

Submitted By:

21BCE1592 Radhika Arvind Sardeshpande

21BCE1986 Sanjana Boligorla

Introduction

Mental health refers to a state of emotional, cognitive and behavioural well being. Unfortunately, it doesn't get the attention that it requires. This negligence has a significant impact on health, social, economic factors and human rights.

Surveys by the Indian Council of Medical Research (ICMR) revealed that 12-13 per cent of students in India suffer from psychological, emotional and behavioural conditions.

Through this analysis, we aim to understand the relation between depression among college students and their CGPA, Age, Gender and some other factors.

Dataset

	Timestamp	Gender	Age	Course	Year	CGPA	Married	Depression	Anxiety	Panic_Attack	Treatment
1	08-07-2020 12:02	Female	18	Engineering	year 1	3.00 - 3.49	No	Yes	No	Yes	No
2	08-07-2020 12:04	Male	21	Islamic education	year 2	3.00 - 3.49	No	No	Yes	No	No
3	08-07-2020 12:05	Male	19	BIT	Year 1	3.00 - 3.49	No	Yes	Yes	Yes	No
4	08-07-2020 12:06	Female	22	Laws	year 3	3.00 - 3.49	Yes	Yes	No	No	No
5	08-07-2020 12:13	Male	23	Mathemathics	year 4	3.00 - 3.49	No	No	No	No	No
6	08-07-2020 12:31	Male	19	Engineering	Year 2	3.50 - 4.00	No	No	No	Yes	No
7	08-07-2020 12:32	Female	23	Pendidikan islam	year 2	3.50 - 4.00	Yes	Yes	No	Yes	No
8	08-07-2020 12:33	Female	18	BCS	year 1	3.50 - 4.00	No	No	Yes	No	No
9	08-07-2020 12:35	Female	19	Human Resources	Year 2	2.50 - 2.99	No	No	No	No	No
10	08-07-2020 12:39	Male	18	Irkhs	year 1	3.50 - 4.00	No	No	Yes	Yes	No
11	08-07-2020 12:39	Female	20	Psychology	year 1	3.50 - 4.00	No	No	No	No	No
12	08-07-2020 12:39	Female	24	Engineering	Year 3	3.50 - 4.00	Yes	Yes	No	No	No
13	08-07-2020 12:40	Female	18	BCS	year 1	3.00 - 3.49	No	Yes	No	No	No
14	08-07-2020 12:41	Male	19	Engineering	year 1	3.00 - 3.49	No	No	No	No	No
15	08-07-2020 12:43	Female	18	KENMS	Year 2	3.50 - 4.00	No	No	Yes	No	No
16	08-07-2020 12:43	Male	24	BCS	Year 3	3.50 - 4.00	No	No	No	No	No
17	08-07-2020 12:46	Female	24	Accounting	year 3	3.00 - 3.49	No	No	No	No	No
18	08-07-2020 12:52	Female	24	ENM	year 4	3.00 - 3.49	Yes	Yes	Yes	Yes	No
19	08-07-2020 13:05	Female	20	BIT	Year 2	3.50 - 4.00	No	No	Yes	No	No
20	08-07-2020 13:07	Female	18	Marine science	year 2	3.50 - 4.00	Yes	Yes	Yes	Yes	No

Showing 1 to 20 of 101 entries, 11 total columns

Source - https://www.kaggle.com/datasets/shariful07/student-mental-health

Project GitHub Link: https://github.com/Ra-Sp/Student-Mental-Health-Analysis

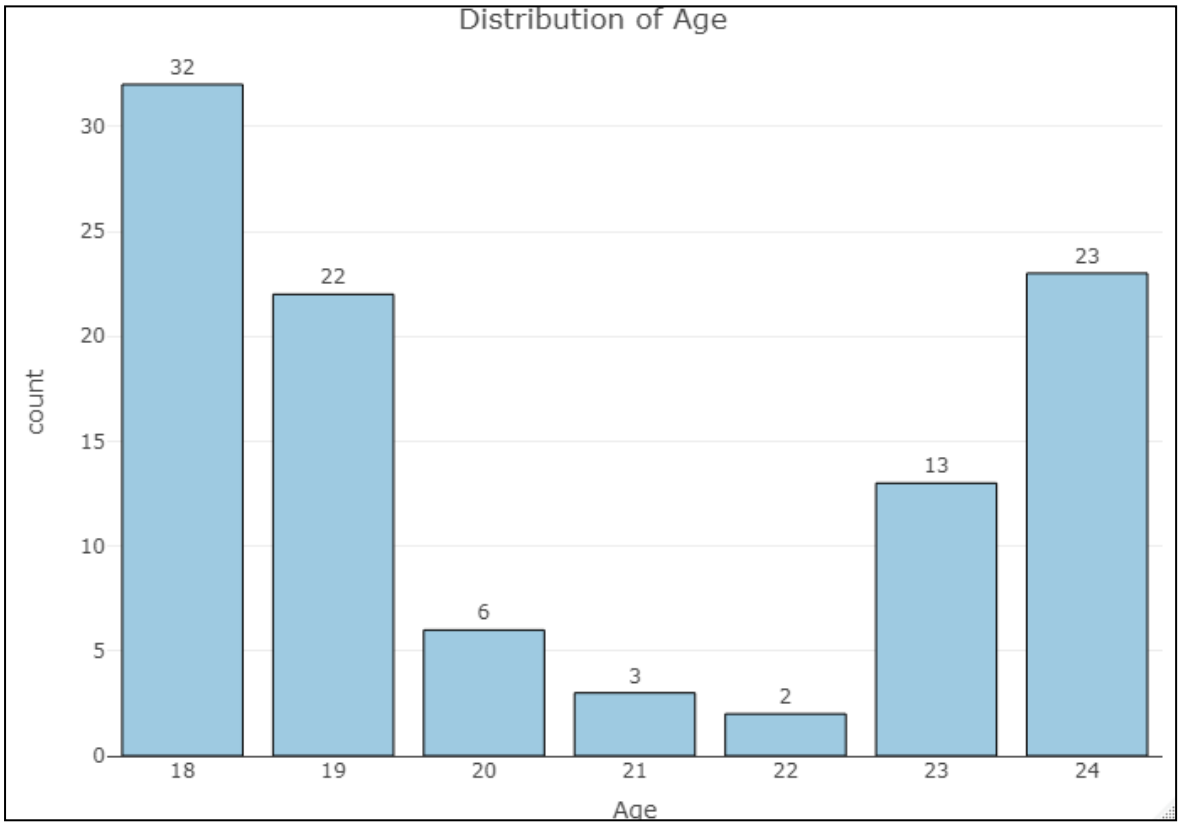
Exploratory Data Analysis (EDA)

Distribution of Age:

Code:

```
# Age Distribution
health %>%
  group_by(Age) %>%
  summarize(count = n()) %>%
  plot_ly(x = ~Age, y = ~count, type = 'bar',
          text = ~count,
          textposition = 'outside',
          marker = list(color = 'rgb(158,202,225)',
                        line = list(color = 'black',
                                    width = 1.0))) %>%
  layout(title = 'Distribution of Age')
```

Output:

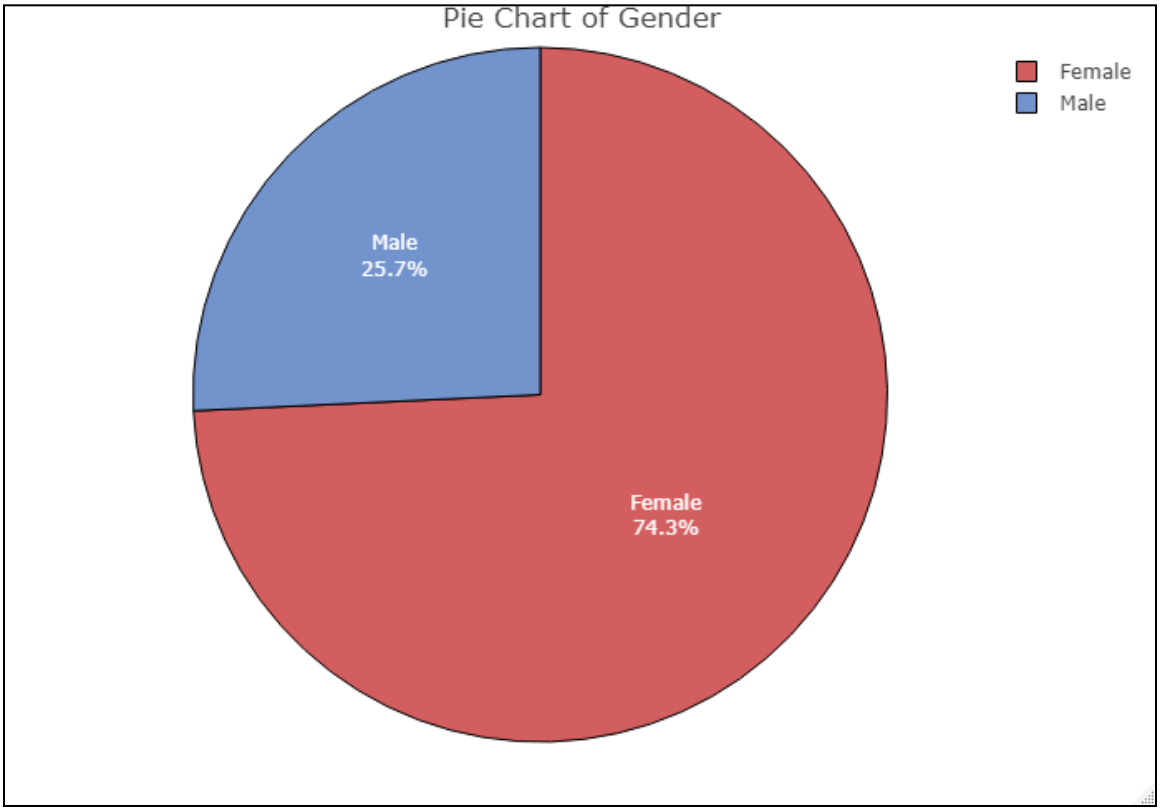


Gender Distribution:

Code:

```
# Gender Distribution
Health_SummaryStat <- health %>%
  group_by(Gender) %>%
  summarise(count = n(),
            percentage = round((n()/ nrow(health)), digits = 4))
colors <- c('rgb(211,94,96)','rgb(114,147,203)')
Gender_PieChart <- plot_ly(data = Health_SummaryStat, labels = ~Gender, values = ~percentage,
                           type = 'pie', sort = FALSE,
                           textposition = 'inside',
                           textinfo = 'label+percent',
                           insidetextfont = list(color = 'White'),
                           hoverinfo = 'text',
                           text = ~count,
                           marker = list(colors = colors,
                                         line = list(color = 'Black', width = 1)),
                           showlegend = TRUE)
Gender_PieChart <- Gender_PieChart %>% layout(title = 'Pie Chart of Gender')
Gender_PieChart
```

Output:

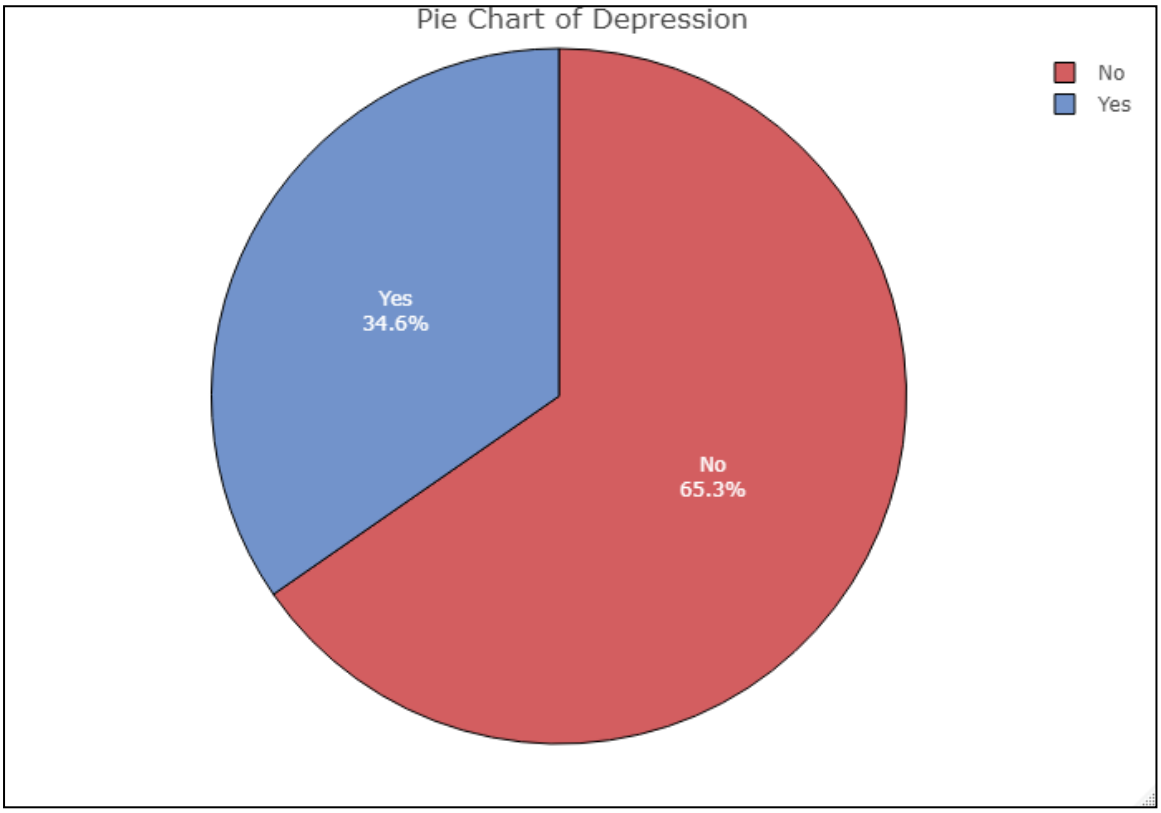


Depression Distribution:

Code:

```
# Depression
Health_SummaryStat2 <- health %>%
  group_by(Depression) %>%
  summarise(count = n(),
            percentage = round((n()/ nrow(health)), digits = 4))
Depression_PieChart <- plot_ly(data = Health_SummaryStat2, labels = ~Depression, values = ~percentage,
                               type = 'pie', sort = FALSE,
                               textposition = 'inside',
                               textinfo = 'label+percent',
                               insidetextfont = list(color = 'White'),
                               hoverinfo = 'text',
                               text = ~count,
                               marker = list(colors = colors,
                                             line = list(color = 'Black', width = 1)),
                               showlegend = TRUE)
Depression_PieChart %>% layout(title = 'Pie Chart of Depression')
```

Output:

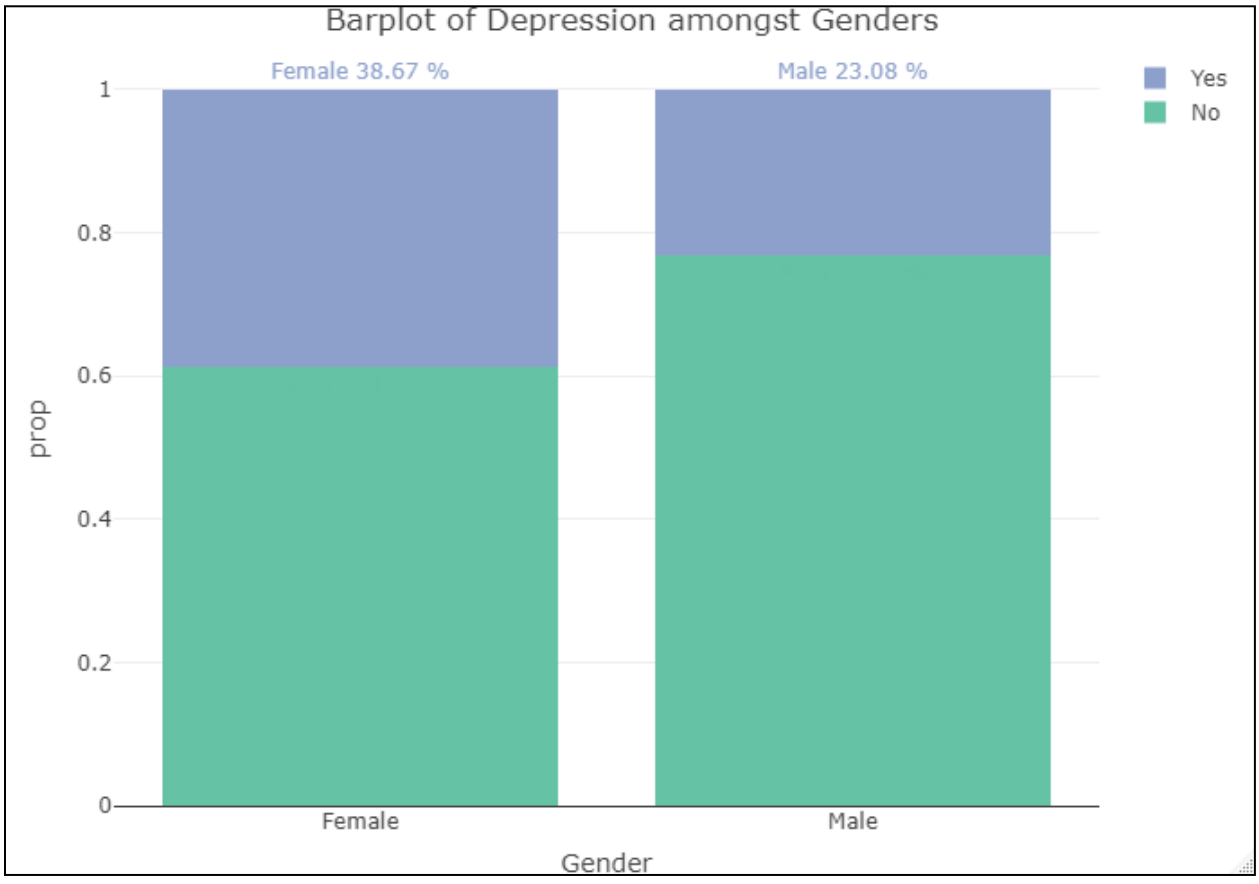


Depression amongst Genders:

Code:

```
# Depression vs Gender
health %>%
  count(Gender, Depression, sort = FALSE) %>%
  group_by(Gender) %>%
  mutate(prop = round((n / sum(n)),digits = 4)) %>%
  plot_ly(x = ~Gender, y = ~prop, color = ~Depression, type = "bar",
          text = ~paste(Gender, prop*100 ,'%'),
          textposition = 'outside') %>%
  layout(barmode = 'stack',
         title = 'Barplot of Depression amongst Genders')
```

Output:

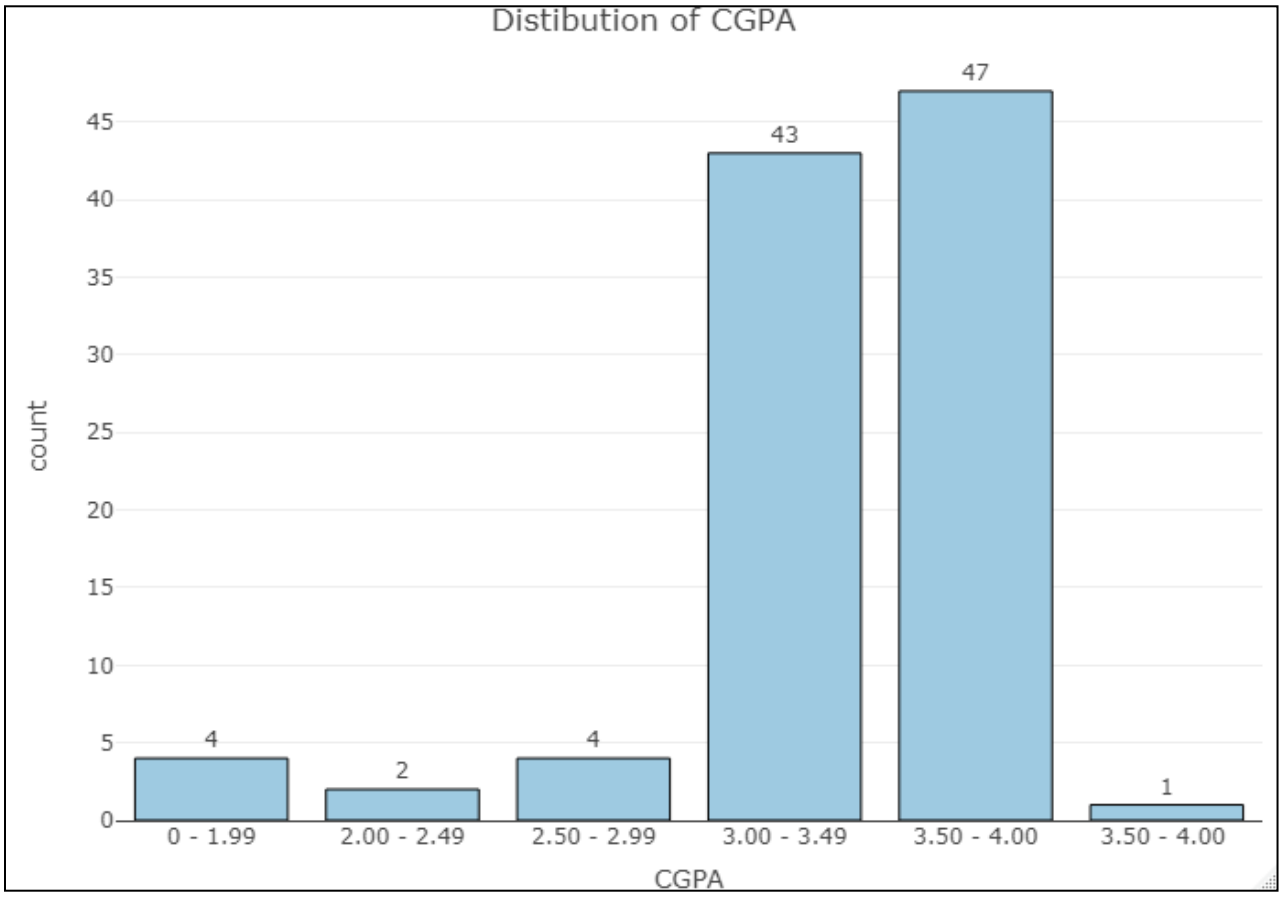


Distribution of CGPA:

Code:

```
# CGPA
health$CGPA <- as.factor(health$CGPA)
health %>%
  group_by(CGPA)%>%
  summarize(count = n()) %>%
  plot_ly(x =~CGPA, y=~count, type = 'bar',
          text = ~count,
          textposition = 'outside',
          marker = list(color = 'rgb(158,202,225)',
                        line = list(color = 'black',
                                    width = 1.0))) %>%
  layout(title = 'Distibution of CGPA')
```

Output:

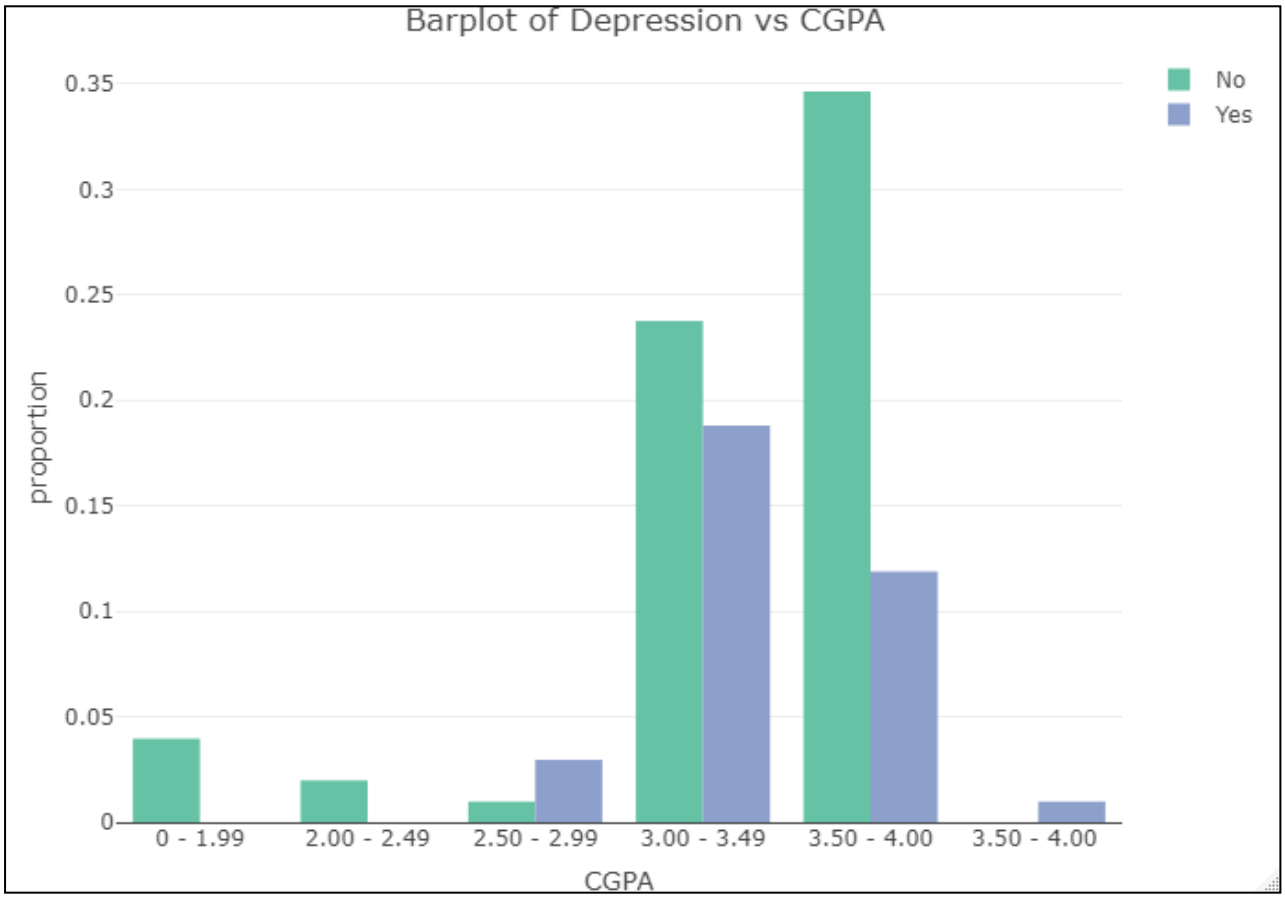


Depression vs CGPA:

Code:

```
health %>%
  count(CGPA, Depression, sort = F) %>%
  mutate(proportion = round((n/sum(n)),digits=4)) %>%
  plot_ly(x =~CGPA, y=~proportion, color = ~Depression, type = 'bar') %>%
  layout(barmode = 'Group',
         title = 'Barplot of Depression vs CGPA')
```

Output:

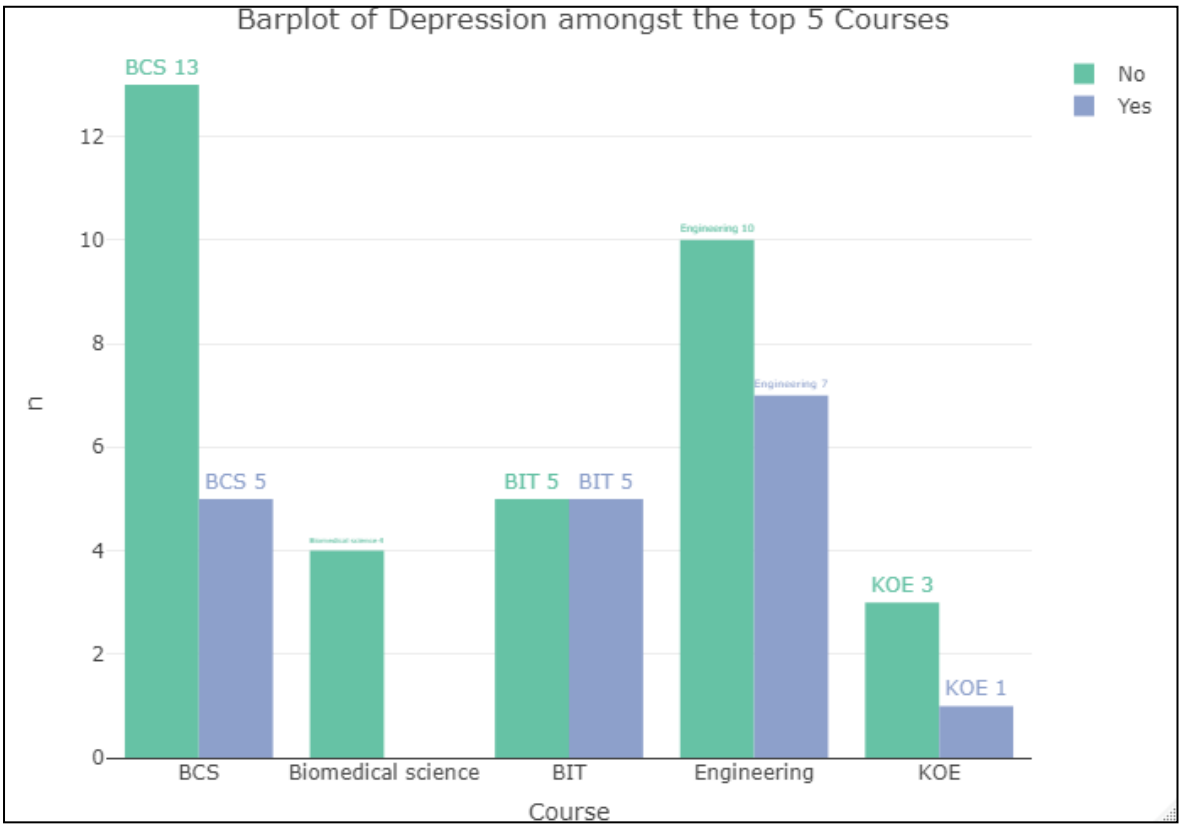


Depression amongst top 5 courses:

Code:

```
# Courses
health %>%
  group_by(Course) %>%
  summarise(count = n()) %>%
  arrange(desc(count)) %>%
  filter(count >2)
health %>%
  filter(grepl('BIT|KOE|BCS|Engineering|Biomedical science', Course)) %>%
  count(Course, Depression, sort = T) %>%
  group_by(Course) %>%
  mutate(prop = round((n / sum(n)),digits = 4)) %>%
  plot_ly(x = ~Course, y=~n, color = ~Depression, type = "bar",
          text = ~paste(Course, n),
          textposition = 'outside') %>%
  layout(barmode = 'Stacked',
         title = 'Barplot of Depression amongst the top 5 Courses')
```

Output:



Why Logistic Regression and not Linear Regression?

Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable. For example, one might want to relate the weights of individuals to their heights using a linear regression model. (ref: http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm)

But, for our dataset, where most questions have only classification based “Yes/ No” responses, linear regression is not suitable, because it is unbounded in nature.

Consider a scenario where we need to classify whether an email is spam or not. If we use linear regression for this problem, there is a need for setting up a threshold based on which classification can be done. Say if the actual class is malignant, predicted continuous value 0.4 and the threshold value is 0.5, the data point will be classified as not malignant which can lead to serious consequence in real time. (ref: https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc)

Logistic Regression

Logistic regression is used to calculate the probability of a binary event occurring, and to deal with issues of classification. For example, predicting if an incoming email is spam or not spam, or predicting if a credit card transaction is fraudulent or not fraudulent. In a medical context, logistic regression may be used to predict whether a tumor is benign or malignant. (ref: https://careerfoundry.com/en/blog/data-analytics/what-is-logistic-regression/)

In this project, we have focused only on Binary logistic regression, which is the statistical technique used to predict the relationship between the dependent variable (Y) and the independent variable (X), where the dependent variable is binary in nature. For example, the output can be Success/Failure, 0/1, True/False, or Yes/No.

Logistic Regression model:

Using CGPA as the predictor and Depression as the response variable

```
> CGPA = health$CGPA
> Depressed = health$Depression
> data <- data.frame(CGPA, Depressed)
> data$Depressed = factor(data$Depressed, labels = c(0, 1))
> # fitting the logistic regression model
> model <- glm(Depressed ~ CGPA, data = data, family = binomial())
> summary(model)

Call:
glm(formula = Depressed ~ CGPA, family = binomial(), data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6651  -0.7679  -0.7679   1.2781   1.6524

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)     -1.757e+01  1.978e+03  -0.009    0.993
CGPA2.00 - 2.49 -4.622e-09  3.426e+03   0.000    1.000
CGPA2.50 - 2.99  1.866e+01  1.978e+03   0.009    0.992
CGPA3.00 - 3.49  1.733e+01  1.978e+03   0.009    0.993
CGPA3.50 - 4.00  1.650e+01  1.978e+03   0.008    0.993
CGPA3.50 - 4.00  3.513e+01  4.423e+03   0.008    0.994

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 130.35  on 100  degrees of freedom
Residual deviance: 116.93  on  95  degrees of freedom
AIC: 128.93

Number of Fisher Scoring iterations: 16
```

Prediction

```
> newdata <- data.frame(CGPA = "3.00 - 3.49")
> predict(model, newdata, type = "response")
        1
0.4418605
```

Plotting the Regression Model:

Code:

```
#-------------------------------------------------------------------------------
#Plotting the regression model

Age = health$Age
Depressed = ifelse(health$Depression == "Yes", 1, 0)
data <- data.frame(Age, Depressed)

age_seq <- seq(min(Age), max(Age), length.out = 7)

# predict the probability of depression vs age
probs <- predict(model, newdata = data.frame(Age = age_seq), type = "response")

# plot the logistic regression curve
ggplot(health, aes(x = Age, y = Depressed)) +
  geom_point() +
  stat_smooth(method="glm", color="green", se=FALSE, method.args = list(family=binomial)) +
  xlab("Age") +
  ylab("Probability of Depression") +
  ggtitle("Logistic Regression Model")
```

Plot: