Data Science in Psychology

Team Psychology

Table of contents

1 Authors

- Sashank Gangadharabhotla
- Brahmendra Charan Attanti
- Yamini Priyanka Kandikattu
- Anila Reddy Musku
- Megha Veeregowda

2 What is Psychology

The scientific study of behavior and the mind is known as psychology.In simple words, it explains how the human brain works and why It behaves in such a particular way.

3 Question of interest?

Our main aim is to find out the factors leading to psychological issues like depression (mainly focusing on the Age and Gender of that person) and it is important to figure out those factors as depression strongly effects human's life.

Source - we got the data from kaggle https://www.kaggle.com/datasets/diegobabativa/depression

4 Description of the dataset

1 - Our dataset contains 1429 rows and 23 columns which is a reasonable size to perform analysis.

2 - target label is the 'depression' column which takes on 0 and 1 values with 0 being not depressed and 1 being depressed.

3 - The independent columns in our data set which are going to affect our target label are majorly of two categories - Family related information and financial perspective.

4 - some of the columns related to one's personal(family) information are Age, Gender, Married, Number of children, total numbers in the family, educational level etc..

5 - some of the columns related to one's financial information are the incoming salary, living expenses, Gained assets, buisinesses if any and so on.

5 Analysing the dataset

5.1 Loading Library

```
library('tidyverse')
```

```
-- Attaching packages --------------------------------------- tidyverse 1.3.2 --
v ggplot2 3.4.0      v purrr   0.3.4
v tibble  3.1.8      v dplyr   1.0.10
v tidyr   1.2.0      v stringr 1.4.0
v readr   2.1.3      v forcats 0.5.1
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

```
library('ggplot2')
```

5.2 Reading Dataset

```
df <- read_csv('/Users/yaminipriyankakandikattu/Downloads/b_depressed.csv')
```

```
Rows: 1429 Columns: 23
-- Column specification ---------------------------------------------------------
Delimiter: ","
dbl (23): Survey_id, Ville_id, sex, Age, Married, Number_children, education...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(df)
```

```
# A tibble: 6 x 23
  Survey_id Ville_id   sex   Age Married Numbe~1 educa~2 total~3 gaine~4 durab~5
      <dbl>    <dbl> <dbl> <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
1       926       91     1    28       1       4      10       5  2.89e7  2.29e7
2       747       57     1    23       1       3       8       5  2.89e7  2.29e7
3      1190      115     1    22       1       3       9       5  2.89e7  2.29e7
4      1065       97     1    27       1       2      10       4  5.27e7  1.97e7
5       806       42     0    59       0       4      10       6  8.26e7  1.74e7
6       483       25     1    35       1       6      10       8  3.59e7  7.37e5
# ... with 13 more variables: save_asset <dbl>, living_expenses <dbl>,
#   other_expenses <dbl>, incoming_salary <dbl>, incoming_own_farm <dbl>,
#   incoming_business <dbl>, incoming_no_business <dbl>,
#   incoming_agricultural <dbl>, farm_expenses <dbl>, labor_primary <dbl>,
#   lasting_investment <dbl>, no_lasting_investmen <dbl>, depressed <dbl>, and
#   abbreviated variable names 1: Number_children, 2: education_level,
#   3: total_members, 4: gained_asset, 5: durable_asset
# i Use `colnames()` to see all variable names
```

```
dim(df)
```

```
[1] 1429   23
```

```
names(df)
```

```
 [1] "Survey_id"             "Ville_id"              "sex"
 [4] "Age"                   "Married"               "Number_children"
 [7] "education_level"       "total_members"         "gained_asset"
[10] "durable_asset"         "save_asset"            "living_expenses"
```

```
[13] "other_expenses"        "incoming_salary"       "incoming_own_farm"
[16] "incoming_business"     "incoming_no_business"  "incoming_agricultural"
[19] "farm_expenses"         "labor_primary"         "lasting_investment"
[22] "no_lasting_investmen"  "depressed"
```

```
df_stat <- df
colSums(is.na(df_stat))
```

```
            Survey_id              Ville_id                   sex
                    0                     0                     0
                  Age               Married       Number_children
                    0                     0                     0
      education_level         total_members          gained_asset
                    0                     0                     0
        durable_asset            save_asset       living_expenses
                    0                     0                     0
       other_expenses       incoming_salary     incoming_own_farm
                    0                     0                     0
    incoming_business  incoming_no_business incoming_agricultural
                    0                     0                     0
        farm_expenses         labor_primary    lasting_investment
                    0                     0                     0
 no_lasting_investmen             depressed
                   20                     0
```

We are having just 20 null values in ‘no_lasting_investmen’ column - dropping those

```
df <- na.omit(df)
colSums(is.na(df))
```

```
            Survey_id              Ville_id                   sex
                    0                     0                     0
                  Age               Married       Number_children
                    0                     0                     0
      education_level         total_members          gained_asset
                    0                     0                     0
        durable_asset            save_asset       living_expenses
                    0                     0                     0
       other_expenses       incoming_salary     incoming_own_farm
                    0                     0                     0
    incoming_business  incoming_no_business incoming_agricultural
                    0                     0                     0
        farm_expenses         labor_primary    lasting_investment
                    0                     0                     0
 no_lasting_investmen             depressed
                    0                     0
```

looking at our target label - ‘depressed’ column

```
unique(df)
```

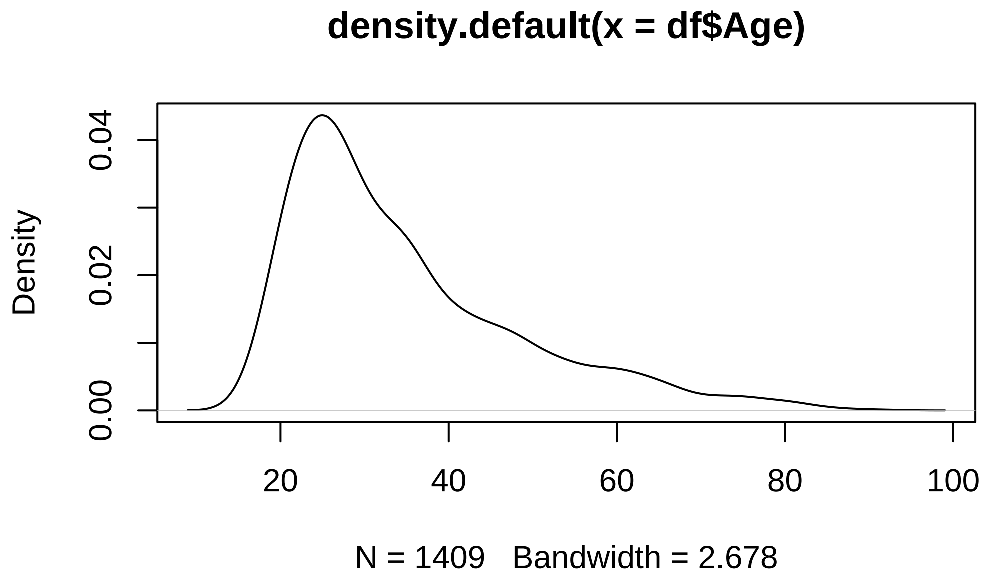
```
# A tibble: 1,409 x 23
   Survey_id Ville~1   sex   Age Married Numbe~2 educa~3 total~4 gaine~5 durab~6
       <dbl>   <dbl> <dbl> <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
 1       926      91     1    28       1       4      10       5  2.89e7  2.29e7
 2       747      57     1    23       1       3       8       5  2.89e7  2.29e7
 3      1190     115     1    22       1       3       9       5  2.89e7  2.29e7
 4      1065      97     1    27       1       2      10       4  5.27e7  1.97e7
 5       806      42     0    59       0       4      10       6  8.26e7  1.74e7
 6       483      25     1    35       1       6      10       8  3.59e7  7.37e5
 7       849     130     0    34       0       1       9       3  4.13e7  2.19e7
 8      1386      72     1    21       1       2      10       4  1.20e7  2.03e7
 9       930     195     1    32       1       7       9       9  1.11e7  2.52e7
10       390      33     1    29       1       4      10       5  2.89e7  2.29e7
# ... with 1,399 more rows, 13 more variables: save_asset <dbl>,
#   living_expenses <dbl>, other_expenses <dbl>, incoming_salary <dbl>,
#   incoming_own_farm <dbl>, incoming_business <dbl>,
#   incoming_no_business <dbl>, incoming_agricultural <dbl>,
#   farm_expenses <dbl>, labor_primary <dbl>, lasting_investment <dbl>,
#   no_lasting_investmen <dbl>, depressed <dbl>, and abbreviated variable names
#   1: Ville_id, 2: Number_children, 3: education_level, 4: total_members, ...
# i Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

0 refers to not depressed and 1 refers to being depressed

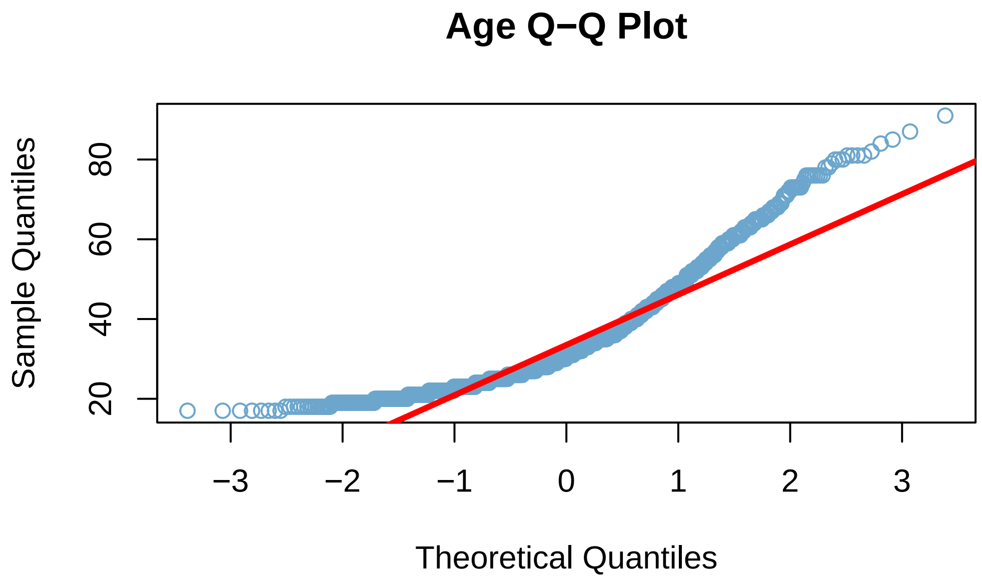
5.3 Visualizations

5.3.1 understanding of the data attributes by looking at how they are distributed

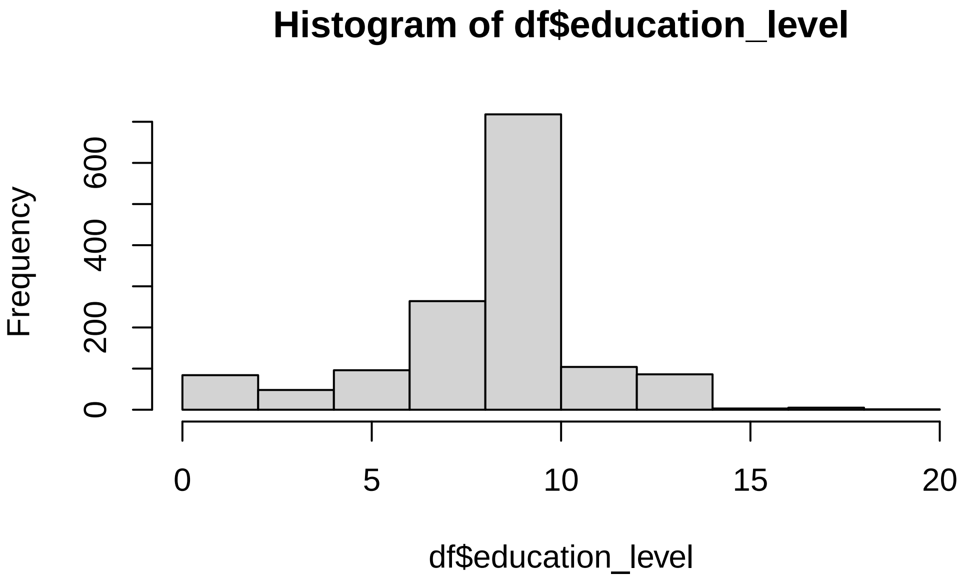
```
plot(density(df$Age))
```



```
# most of the age between 20 and 40 mark

qqnorm(df$Age, col = "skyblue3",
       main = "Age Q-Q Plot")
qqline(df$Age, col = "red", lwd = 3)
```



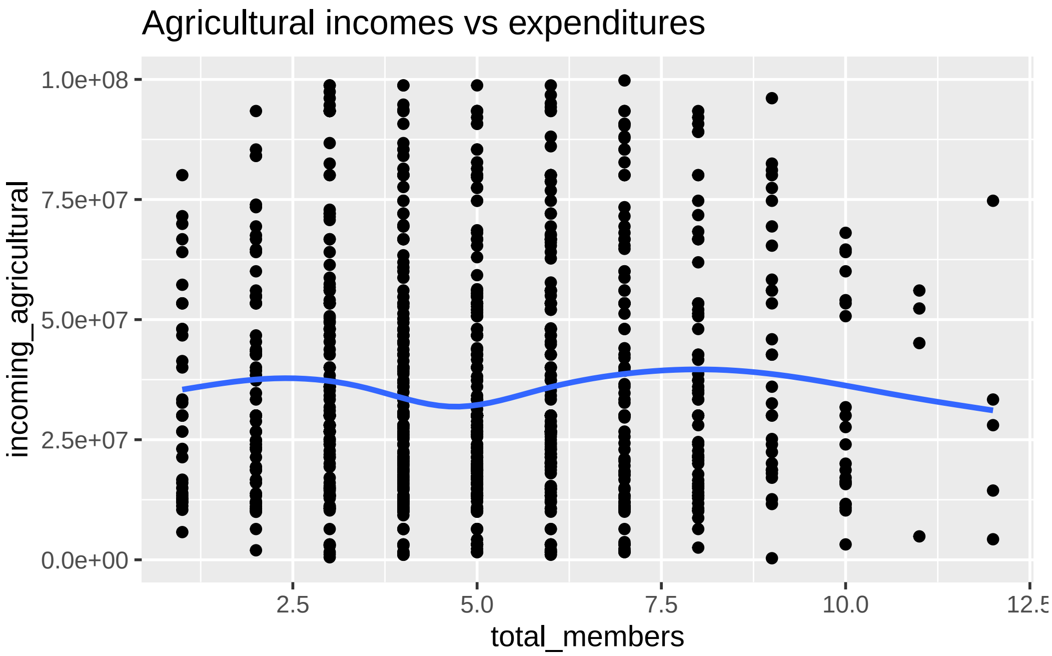
```
# by looking at the graph above and the qqplot we can say that it's slightly a bit right skewed
hist(df$education_level)
```



Most of the people have studied till their 10th standard(around 700 - 50% of the total records) and the curve is more or less normally distributed.

```
ggplot(data = df, mapping = aes(x = total_members, y = incoming_agricultural)) +
  geom_point() +
  geom_smooth(se = FALSE) +
  labs(title = 'Agricultural incomes vs expenditures')
```

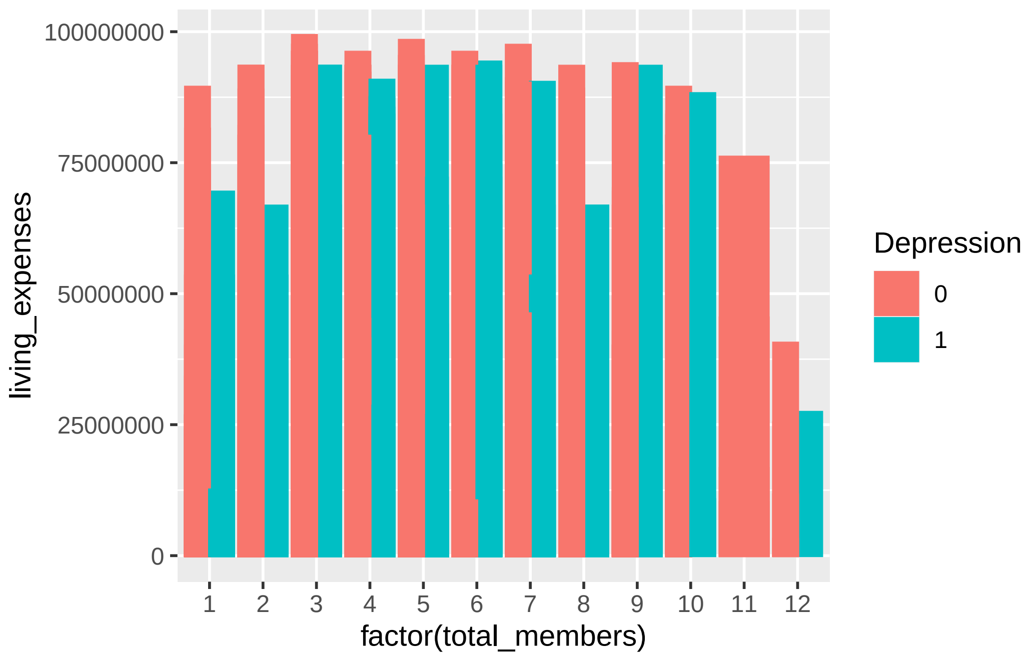
```
`geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



```
# The number of people in the family doesn't relate that much to the agricultural income of their
```

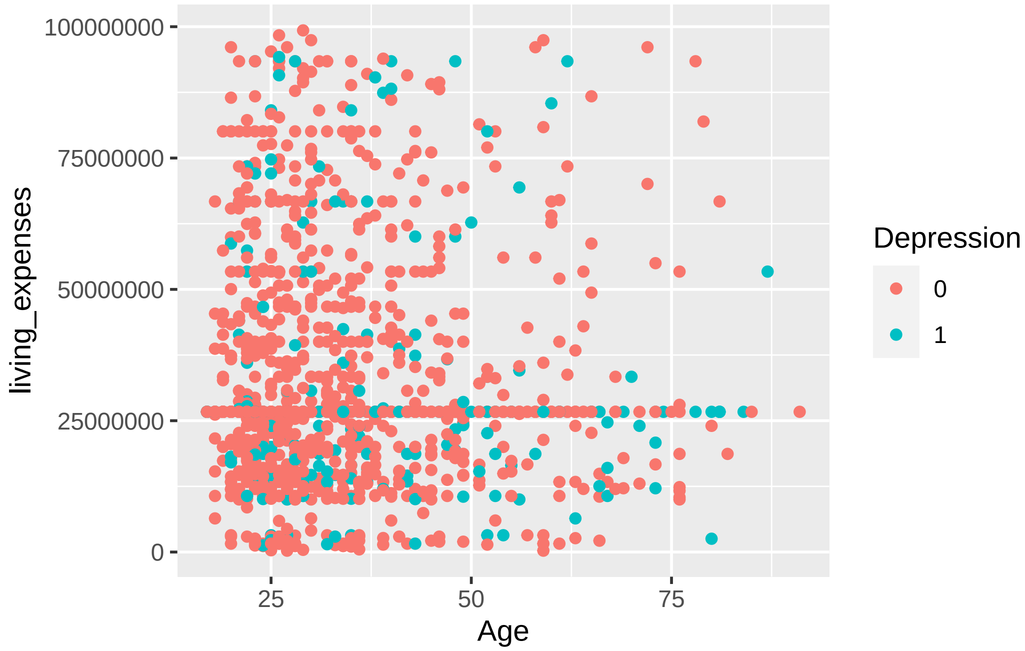
The number of people in the family doesn't relate that much to the agricultural income of their family.

```
Depression <- as.character(df$depressed)
options(scipen=999)

ggplot(df, aes(x = factor(total_members), y = living_expenses, fill =Depression, color = Depressi
  geom_bar(stat = "identity",position = "dodge")
```



We can kind of say that as we have more number of records related to non depressed people, we can see that living expenses are being higher in their case (as its sums up more numbers). By intuiting we can say that in most of the cases depressed people are spending lavishly (per person calculation) as they don't plan proper spending on expenditures.

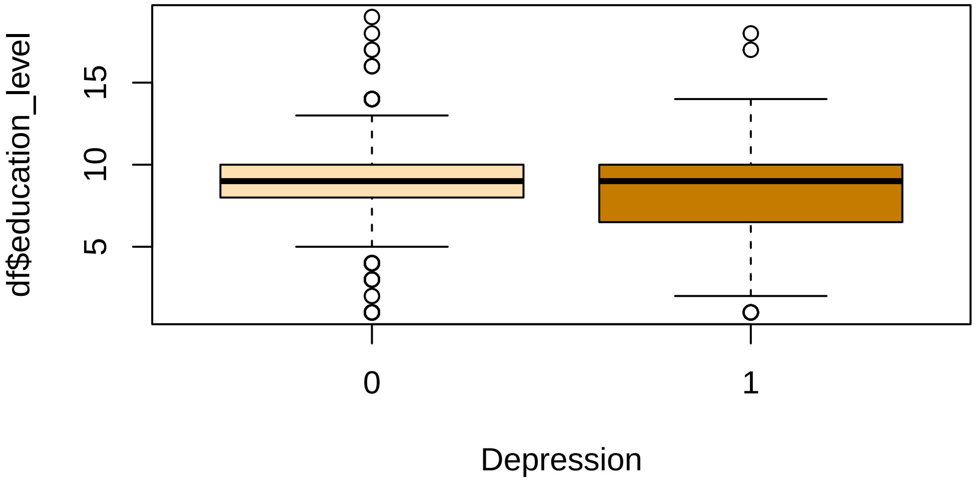
```
ggplot(df,aes(x=Age,y=living_expenses,col=Depression))+geom_point()
```



```
options(scipen=999)
```

One thing we can analyse from the graph is that for lower age values most of them are lesser living expenses and as age increases - most people of that age group are having higher living expenses.

```
boxplot(df$education_level ~ Depression, col = c("#FFE0B2", "#c57B00"))
```



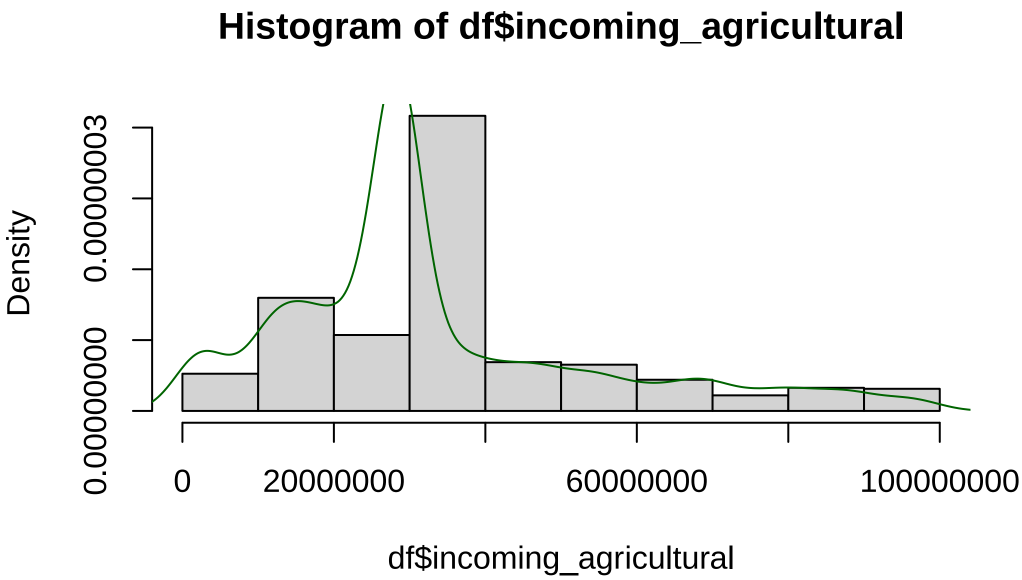
Just seeing how education level values are distributes across two categories of our ‘depressed’ column.

Interestingly, People who are depressed and those who are not are having the same mean(nearly).

People of non depressed category have varying educational levels from 0 till 15+ that’s why you can see some outliers on either sides and also almost most of them are having education levels from 7-10

People of depressed category are mostly having education levels around 6-10 with mean close to 9 and only very few outliers on the right.

```
hist(df$incoming_agricultural, prob = TRUE)
lines(density(df$lasting_investment), col = "#006400")
```



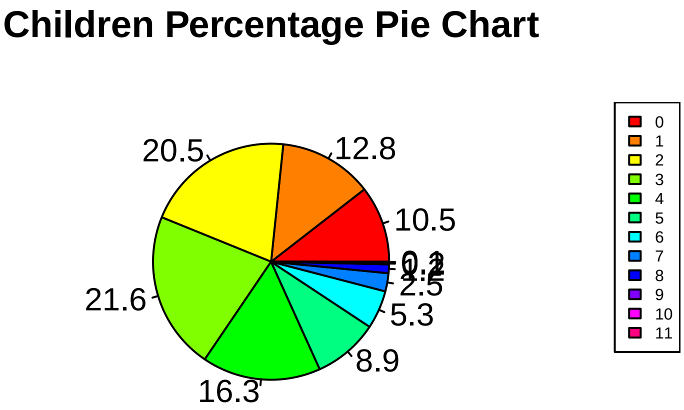
Agricultural incomes are a bit right skewed - some of the families are getting high profits and in general, most of the families are getting around 30 - 40,000k rupees.

```
library(scales)
```

```

Attaching package: 'scales'

The following object is masked from 'package:purrr':

    discard
```

```
The following object is masked from 'package:readr':

    col_factor
```

```
new <-df %>%
  group_by(Number_children) %>%
  summarise(n = n()) %>%
  mutate(Freq = n/sum(n))
new$Number_children <- as.character(new$Number_children)

geeks <- new$n
labels <- new$Number_children

piepercent<- round(100 * geeks / sum(geeks), 1)

# Plot the chart.
pie(geeks, labels = piepercent,
    main = "Children Percentage Pie Chart", col = rainbow(length(geeks)))
legend("topright", labels,
                    cex = 0.5, fill = rainbow(length(geeks)))
```



Just seeing how the number of childrens column is distributed - most families (around 71%) are having from 0-3 children with 1 and 2 children being occurred in most of the cases.

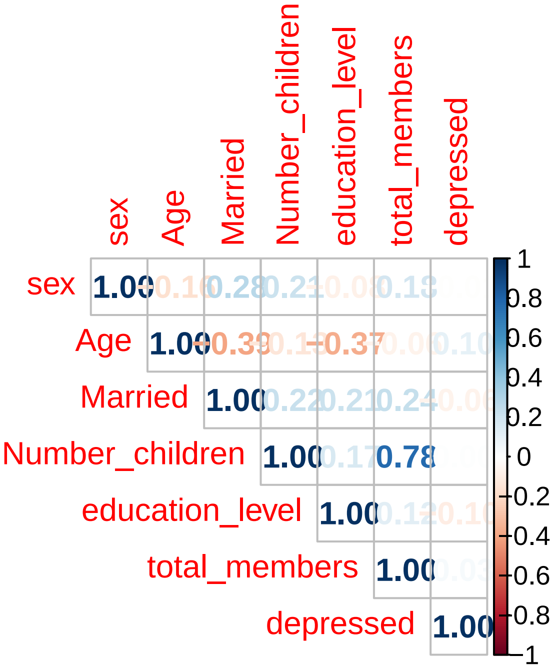
5.4 Analysis

Checking the correlations between the data attributes and with the target label

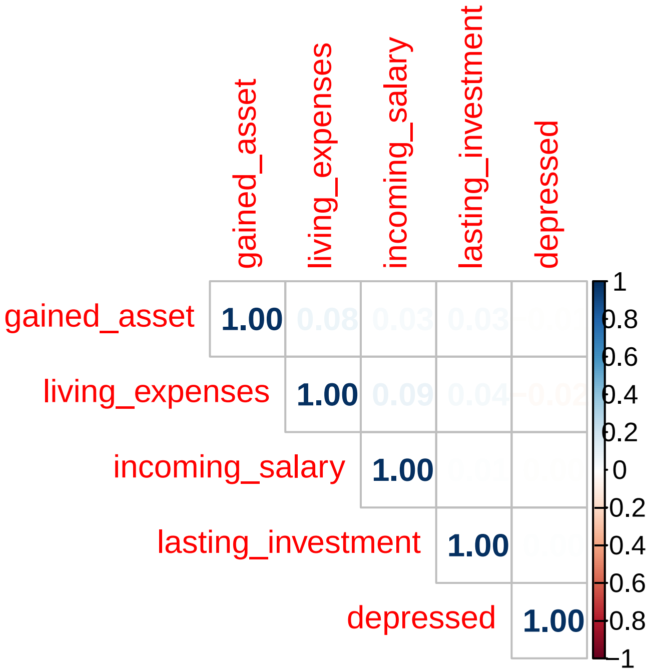
```
library(corrplot)
```

```
corrplot 0.92 loaded
```

```
df_pesronal <- df[c(3,4,5,6,7,8,23)]
df_financial <- df[c(9,12,14,21,23)]
corrplot(cor(df_pesronal), method = "number", type = "upper")
```



```
corrplot(cor(df_financial), method = "number", type = "upper")
```



we can say that Age has the highest correlation with 'depressed' column but as our target label is categorical its better to see the relationship in an other way than just looking at the correlation values.looking at how different variables (Age and Gender) are having an effect on the target label

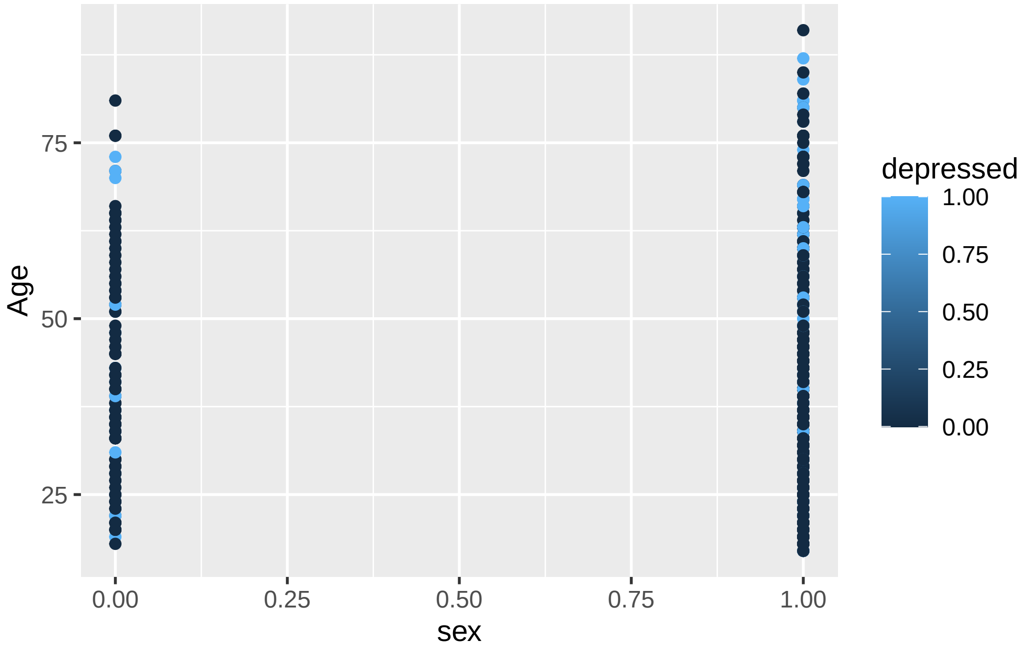
```
t <- df %>% select(depressed, Age) %>% group_by(Age) %>% summarise(count = n(),
                                                                      avg_depression = mean(depresse
#t$avg_age == 0.25
t[order(t$avg_depression, decreasing = TRUE),]
```

```
# A tibble: 69 x 3
     Age count avg_depression
   <dbl> <int>          <dbl>
 1    50     2          1
 2    70     1          1
 3    74     1          1
 4    84     1          1
 5    87     1          1
 6    67     4          0.75
 7    81     4          0.75
 8    80     3          0.667
 9    78     2          0.5
10    73     7          0.429
# ... with 59 more rows
# i Use `print(n = ...)` to see more rows
```

epressed people are slightly of more age - of all the age values which have an average depression greater than 0.25 (24 different age groups) the second least age group is 39 (i.e., 23 out of the top 24 cases have age greater than 39)

Interesting thing - the remaining 1 age group left above in the top 24 groups is that of Age 17 which is the minimum age value in this data set.

```
df %>% ggplot(aes(x = sex, y = Age, color = depressed)) +
  geom_point()
```



In male(category value 0) we can see that depressed people are spread across the whole age groups and coming to the females, you can mostly see that after an age of 60 they are being more depressed comparitively

```
df %>% select(sex, depressed) %>% group_by(sex,depressed) %>% summarise(count = n())
```

```
`summarise()` has grouped output by 'sex'. You can override using the `.groups`
argument.
```

```
# A tibble: 4 x 3
# Groups:   sex [2]
    sex depressed count
  <dbl>     <dbl> <int>
1     0         0    95
2     0         1    20
```

```
3      1          0  1079
4      1          1   215
```

out of the 115 males in the dataset 20 of them are depressed (around 17%) out of the 1294 females in the dataset 215 of them are depressed (around 16%)

More records of female but from the data we are having we can say that Gender doesn't have that much effect on whether the person is depressed or not.

6 Conclusion

After performing some analysis on the How the Age and Gender columns are going to affect the depression, we can conclude that -

1) More Age groups tend to have more chances of being depressed(escpecially for females) - positive correlation although people of very young age also have quite a fair amount of chance of getting more into the depression kind of activities.
2) Gender doesn't have a strong impact on the depression.

6.1 Any possible Sources of bias in the data set?

```
df %>% count(sex)
```

```
# A tibble: 2 x 2
    sex     n
  <dbl> <int>
1     0   115
2     1  1294
```

```
df %>% count(Married)
```

```
# A tibble: 2 x 2
  Married     n
    <dbl> <int>
1       0   318
2       1  1091
```

```
df %>% count(incoming_salary)
```

```
# A tibble: 2 x 2
  incoming_salary     n
            <dbl> <int>
1               0  1161
2               1   248
```

```
df %>% count(depressed)
```

```
# A tibble: 2 x 2
  depressed     n
      <dbl> <int>
1         0  1174
2         1   235
```

We can see that in all the 4 columns listed above one category is completely dominating the other in terms of the number of times it has occurred. Most of the data is that of females, married, those who don't have any fixed income(salary) and mainly in the dependent column 'depressed' - it has only 15% of records which give us information about depressed people.

So, we can say that this dataset is not a true sample i.e., not a true represntation of the whole population and it does have bias in it.

7 GitHub Link

https://github.com/Charan-AB/Data-science-in-psychology