

Depression Draft 1

Christy Hui

2024-11-30

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.4.1
## Warning: package 'ggplot2' was built under R version 4.4.1
## Warning: package 'tidyr' was built under R version 4.4.1
## Warning: package 'readr' was built under R version 4.4.1
## Warning: package 'purrr' was built under R version 4.4.1
## Warning: package 'stringr' was built under R version 4.4.1
## Warning: package 'forcats' was built under R version 4.4.1
## Warning: package 'lubridate' was built under R version 4.4.1

## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr    1.5.1
## v ggplot2    3.5.1      v tibble     3.2.1
## v lubridate  1.9.3      v tidyr      1.3.1
## v purrr      1.0.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(cowplot)
```

```
## Warning: package 'cowplot' was built under R version 4.4.2
##
## Attaching package: 'cowplot'
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.4.1
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
```

```
##
## lift
library(ROCR)

## Warning: package 'ROCR' was built under R version 4.4.2
library(sjPlot)

## Warning: package 'sjPlot' was built under R version 4.4.2
##
## Attaching package: 'sjPlot'
##
## The following objects are masked from 'package:cowplot':
##
## plot_grid, save_plot
library(visdat)

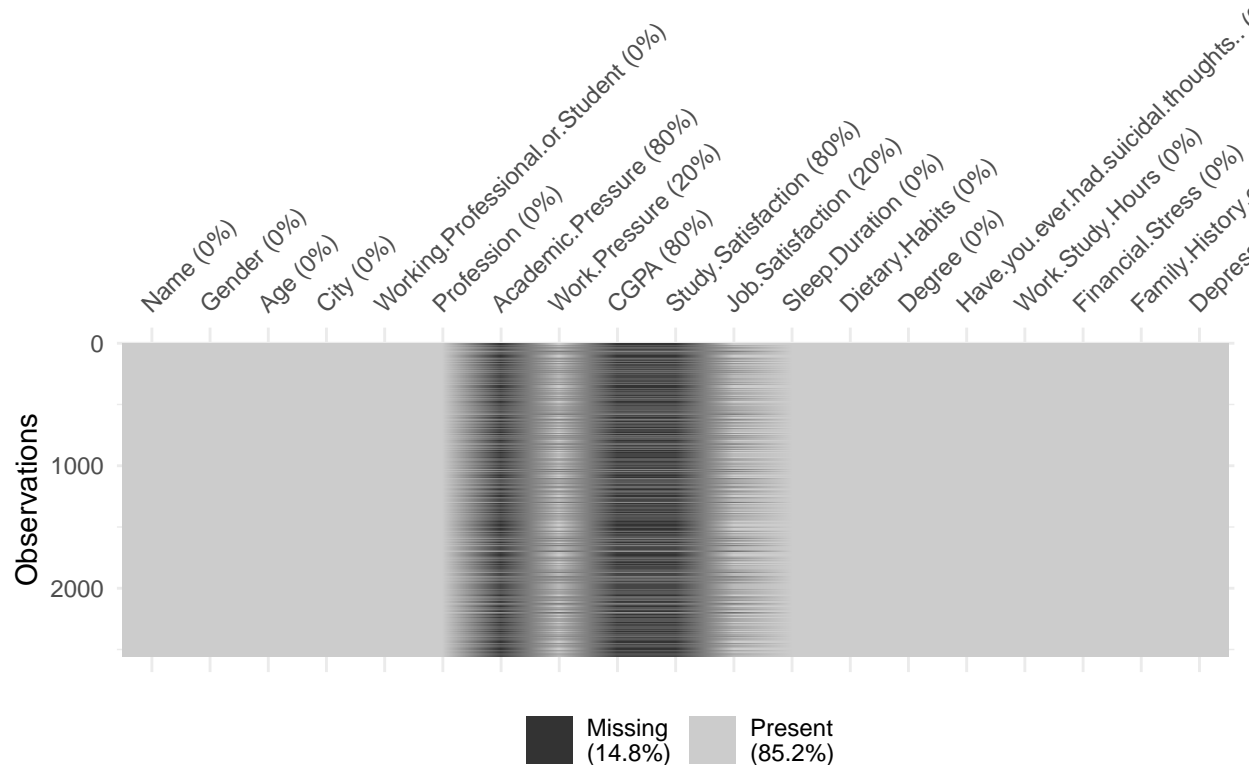
## Warning: package 'visdat' was built under R version 4.4.2
# read data
depression = read.csv("final_depression_dataset_1.csv")

# find the dimension of depression
dim(depression)

## [1] 2556 19

# find if there exist duplicates
sum(duplicated(depression))

## [1] 0
vis_miss(depression)
```



```
# find number of NAs for each column
apply(depression, function(x) {sum(is.na(x))})
```

```
##           Name           Gender
##           0           0
##           Age           City
##           0           0
## Working.Professional.or.Student           Profession
##           0           0
##           Academic.Pressure           Work.Pressure
##           2054           502
##           CGPA           Study.Satisfaction
##           2054           2054
##           Job.Satisfaction           Sleep.Duration
##           502           0
##           Dietary.Habits           Degree
##           0           0
## Have.you.ever.had.suicidal.thoughts..           Work.Study.Hours
##           0           0
##           Financial.Stress           Family.History.of.Mental.Illness
##           0           0
##           Depression
##           0
```

```
# combine pressure columns into one
helper1 = ifelse(is.na(depression$Academic.Pressure), 0, depression$Academic.Pressure)
```

```

helper2 = ifelse(is.na(depression$Work.Pressure), 0, depression$Work.Pressure)

depression$Pressure = helper1 + helper2

# combine satisfaction into one column
helper3 = ifelse(is.na(depression$Study.Satisfaction), 0, depression$Study.Satisfaction)

helper4 = ifelse(is.na(depression$Job.Satisfaction), 0, depression$Job.Satisfaction)

depression$Satisfaction = helper3 + helper4

# delete columns with NAs
depression = depression[, -c(7:11)]
sapply(depression, function(x) {sum(is.na(x))})

```

```

##              Name              Gender
##              0              0
##              Age              City
##              0              0
##      Working.Professional.or.Student      Profession
##              0              0
##              Sleep.Duration      Dietary.Habits
##              0              0
##              Degree Have.you.ever.had.suicidal.thoughts..
##              0              0
##              Work.Study.Hours      Financial.Stress
##              0              0
##      Family.History.of.Mental.Illness      Depression
##              0              0
##              Pressure      Satisfaction
##              0              0

```

```

# due to a large amount of varied answers for "City" and "Profession," we delete the variables
# we also delete name because we don't care about that variable
unique(depression$City)

```

```

## [1] "Ghaziabad"      "Kalyan"      "Bhopal"      "Thane"
## [5] "Indore"         "Pune"        "Bangalore"   "Hyderabad"
## [9] "Srinagar"       "Nashik"      "Kolkata"     "Ahmedabad"
## [13] "Varanasi"       "Chennai"     "Jaipur"      "Surat"
## [17] "Vasai-Virar"   "Rajkot"      "Patna"       "Mumbai"
## [21] "Vadodara"      "Lucknow"     "Faridabad"   "Meerut"
## [25] "Kanpur"        "Visakhapatnam" "Ludhiana"    "Nagpur"
## [29] "Delhi"         "Agra"

```

```

unique(depression$Profession)

```

```

## [1] "Teacher"          "Financial Analyst"      "UX/UI Designer"
## [4] "Civil Engineer"   "Accountant"            "Lawyer"
## [7] "Content Writer"    ""                      "Pilot"
## [10] "Customer Support"  "Judge"                 "Architect"
## [13] "HR Manager"        "Digital Marketer"       "Sales Executive"
## [16] "Business Analyst"  "Mechanical Engineer"    "Consultant"
## [19] "Data Scientist"    "Pharmacist"            "Software Engineer"
## [22] "Travel Consultant" "Manager"               "Entrepreneur"
## [25] "Doctor"           "Researcher"            "Plumber"

```

```
## [28] "Finanancial Analyst"      "Marketing Manager"      "Educational Consultant"
## [31] "Chemist"                  "Research Analyst"       "Chef"
## [34] "Electrician"              "Graphic Designer"       "Investment Banker"
```

```
depression = subset(depression, select = -c(Name, City, Profession))
```

```
# degree has many varied answers as well; however, they can be recoded into three main categories: high
unique(depression$Degree)
```

```
## [1] "MA"      "B.Com"    "M.Com"    "MD"      "BE"      "MCA"
## [7] "BA"      "LLM"      "BCA"      "Class 12" "B.Ed"    "M.Tech"
## [13] "LLB"     "B.Arch"   "ME"       "MBA"     "M.Pharm" "MBBS"
## [19] "PhD"     "BSc"     "MSc"     "MHM"     "BBA"     "BHM"
## [25] "B.Tech"  "M.Ed"    "B.Pharm"
```

```
# recode degree into three categories
```

```
depression$Degree = case_when(depression$Degree == "Class 12" ~ "High School Equivalent",
                              grepl("[BL]", depression$Degree) ~ "Bachelors Degree",
                              grepl("[MP]", depression$Degree) ~ "Post-Graduate Degree")
```

```
table(depression$Degree)
```

```
##
##      Bachelors Degree High School Equivalent      Post-Graduate Degree
##              1193              275              1088
```

```
# find type of each variable so we can change each type
sapply(depression, function(x) {class(x)})
```

```
##                      Gender                      Age
##                      "character"                  "integer"
##      Working.Professional.or.Student              Sleep.Duration
##                      "character"                  "character"
##                      Dietary.Habits                Degree
##                      "character"                  "character"
## Have.you.ever.had.suicidal.thoughts..            Work.Study.Hours
##                      "character"                  "integer"
##                      Financial.Stress              Family.History.of.Mental.Illness
##                      "integer"                    "character"
##                      Depression                    Pressure
##                      "character"                  "numeric"
##                      Satisfaction
##                      "numeric"
```

```
# change each categorical into a factor, changing the base/ordering them if needed
```

```
depression$Gender = as.factor(depression$Gender)
depression$Working.Professional.or.Student = as.factor(depression$Working.Professional.or.Student)
depression$Sleep.Duration = factor(depression$Sleep.Duration, levels = c("Less than 5 hours", "5-6 hours", "7-8 hours", "9-10 hours", "11-12 hours", "13-14 hours", "15-16 hours", "17-18 hours", "19-20 hours", "21-22 hours", "23-24 hours"))
depression$Dietary.Habits = factor(depression$Dietary.Habits, levels = c("Unhealthy", "Moderate", "Healthy"))
depression$Degree = factor(depression$Degree, levels = c("High School Equivalent", "Bachelors Degree", "Post-Graduate Degree"))
depression$Have.you.ever.had.suicidal.thoughts.. = as.factor(depression$Have.you.ever.had.suicidal.thoughts..)
depression$Family.History.of.Mental.Illness = as.factor(depression$Family.History.of.Mental.Illness)
depression$Depression = as.factor(depression$Depression)
```

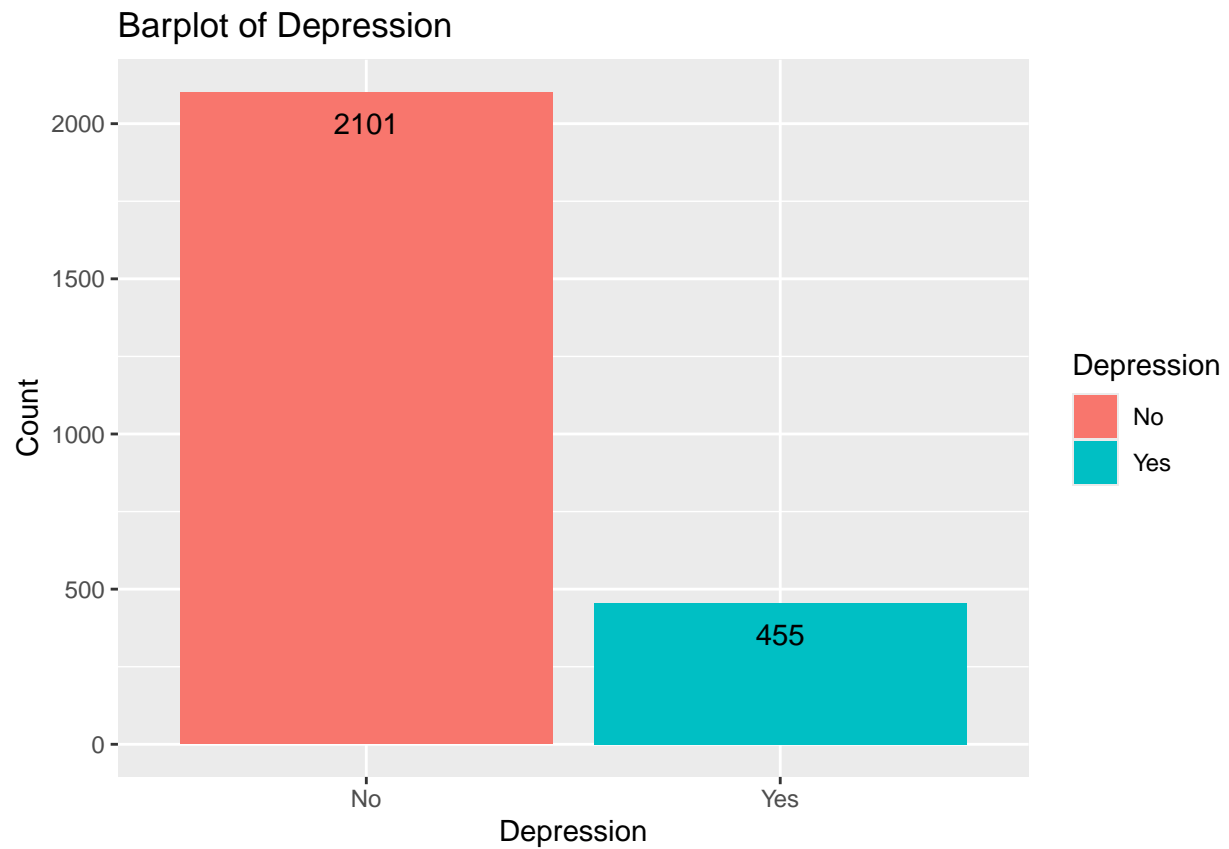
```
# find if any variables are unbalanced
```

```
depressionFactored = select(depression, where(is.factor))
sapply(depressionFactored, table)
```

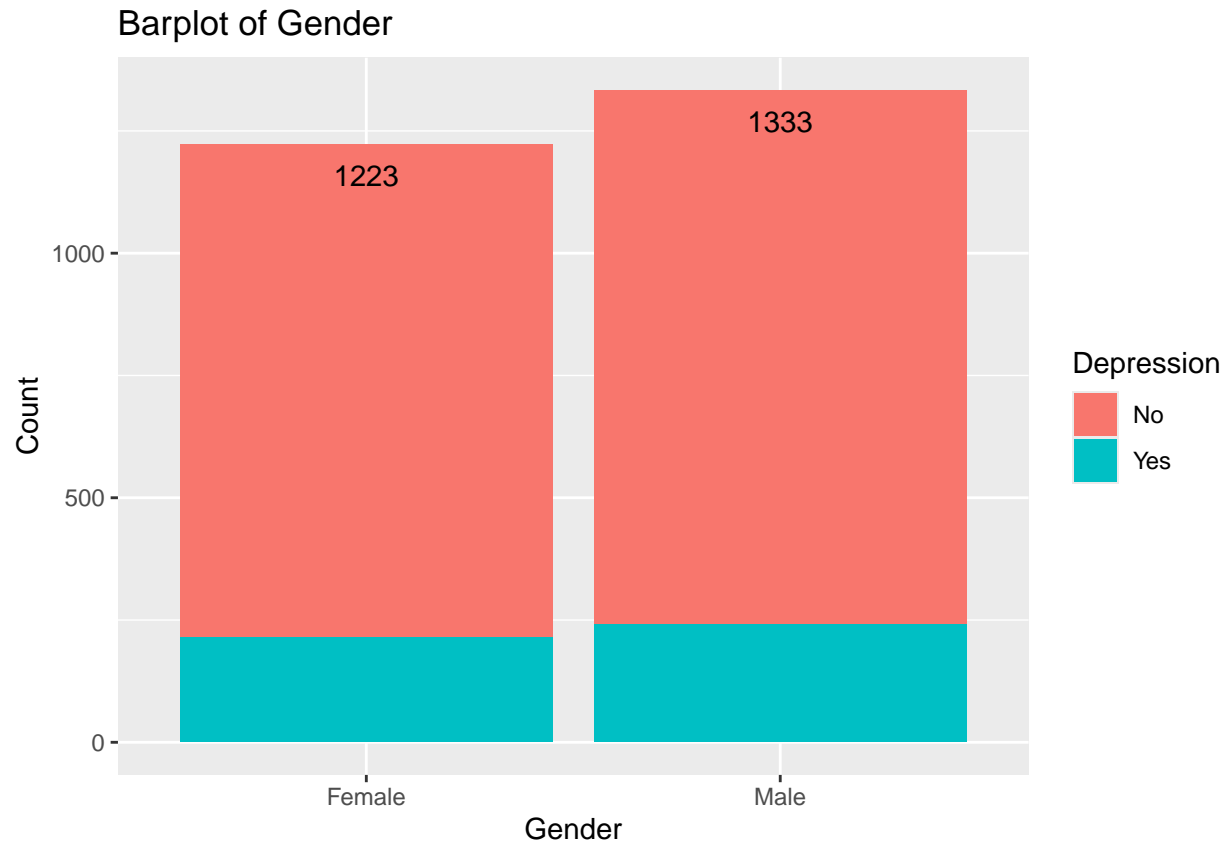
```
## $Gender
##
## Female    Male
##   1223    1333
##
## $Working.Professional.or.Student
##
##           Student Working Professional
##           502                2054
##
## $Sleep.Duration
##
## Less than 5 hours      5-6 hours      7-8 hours More than 8 hours
##           648                628                658                622
##
## $Dietary.Habits
##
## Unhealthy Moderate   Healthy
##     882       832     842
##
## $Degree
##
## High School Equivalent      Bachelors Degree      Post-Graduate Degree
##           275                1193                1088
##
## $Have.you.ever.had.suicidal.thoughts..
##
##   No   Yes
## 1307 1249
##
## $Family.History.of.Mental.Illness
##
##   No   Yes
## 1311 1245
##
## $Depression
##
##   No   Yes
## 2101  455
```

```
# plot depression count
ggplot(depression, aes(x = Depression)) +
  geom_bar(aes(fill = Depression)) +
  xlab("Depression") +
  ylab("Count") +
  ggtitle("Barplot of Depression") +
  geom_text(aes(label = ..count..), stat = "count", vjust = 2)
```

```
## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(count)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



```
# plot gender  
ggplot(depression, aes(x = Gender)) +  
  geom_bar(aes(fill = Depression)) +  
  xlab("Gender") +  
  ylab("Count") +  
  ggtitle("Barplot of Gender") +  
  geom_text(aes(label = ..count..), stat = "count", vjust = 2)
```



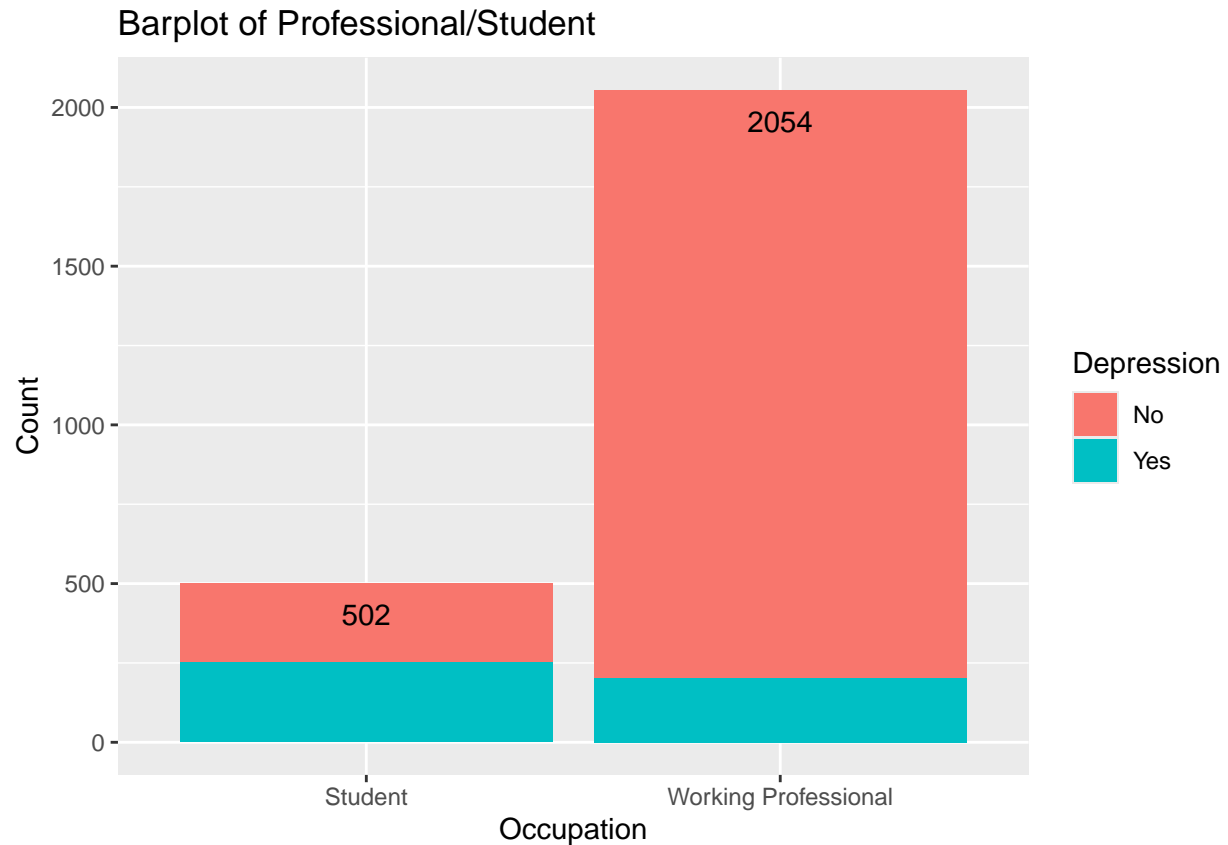
```
table(depression$Depression, depression$Gender)
```

```
##
##      Female Male
##   No    1009 1092
##   Yes     214  241
```

```
prop.table(table(depression$Depression, depression$Gender), margin = 1)
```

```
##
##      Female      Male
##   No 0.4802475 0.5197525
##   Yes 0.4703297 0.5296703
```

```
# plot whether or not person is a working professional or student
ggplot(depression, aes(x = Working.Professional.or.Student)) +
  geom_bar(aes(fill = Depression)) +
  xlab("Occupation") +
  ylab("Count") +
  ggtitle("Barplot of Professional/Student") +
  geom_text(aes(label = ..count..), stat = "count", vjust = 2)
```

```
table(depression$Depression, depression$Working.Professional.or.Student)
```

```
##
##      Student Working Professional
##   No      250             1851
##   Yes      252             203
```

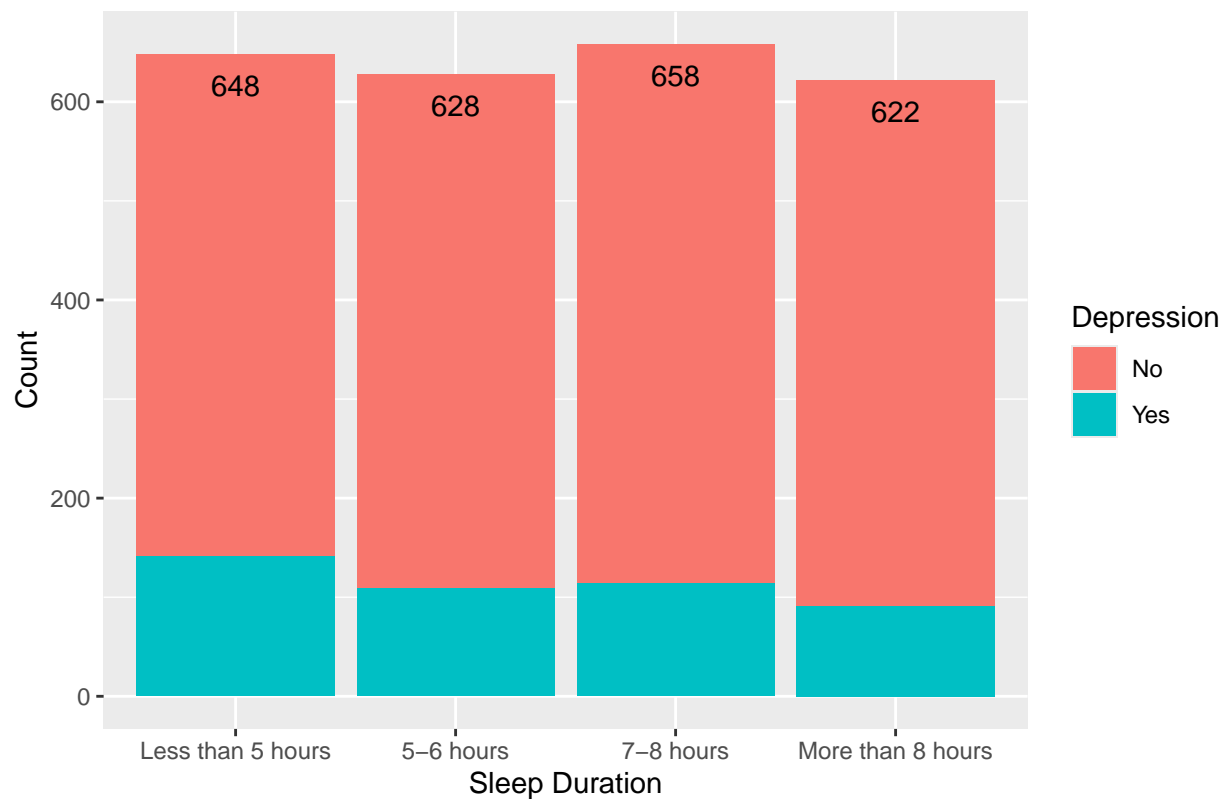
```
prop.table(table(depression$Depression, depression$Working.Professional.or.Student), margin = 1)
```

```
##
##      Student Working Professional
##   No 0.1189910      0.8810090
##   Yes 0.5538462      0.4461538
```

```
# plot sleep duration habits
```

```
ggplot(depression, aes(x = Sleep.Duration)) +
  geom_bar(aes(fill = Depression)) +
  xlab("Sleep Duration") +
  ylab("Count") +
  ggtitle("Barplot of Sleep Duration") +
  geom_text(aes(label = ..count..), stat = "count", vjust = 2)
```

Barplot of Sleep Duration



```
table(depression$Depression, depression$Sleep.Duration)
```

```
##
##      Less than 5 hours 5-6 hours 7-8 hours More than 8 hours
## No           507      519      544           531
## Yes          141      109      114           91
```

```
prop.table(table(depression$Depression, depression$Sleep.Duration), margin = 1)
```

```
##
##      Less than 5 hours 5-6 hours 7-8 hours More than 8 hours
## No           0.2413137 0.2470252 0.2589243           0.2527368
## Yes          0.3098901 0.2395604 0.2505495           0.2000000
```

```
# plot dietary habits
ggplot(depression, aes(x = Dietary.Habits)) +
  geom_bar(aes(fill = Depression)) +
  xlab("Dietary Habits") +
  ylab("Count") +
  ggtitle("Barplot of Dietary Habits") +
  geom_text(aes(label = ..count..), stat = "count", vjust = 2)
```

Barplot of Dietary Habits



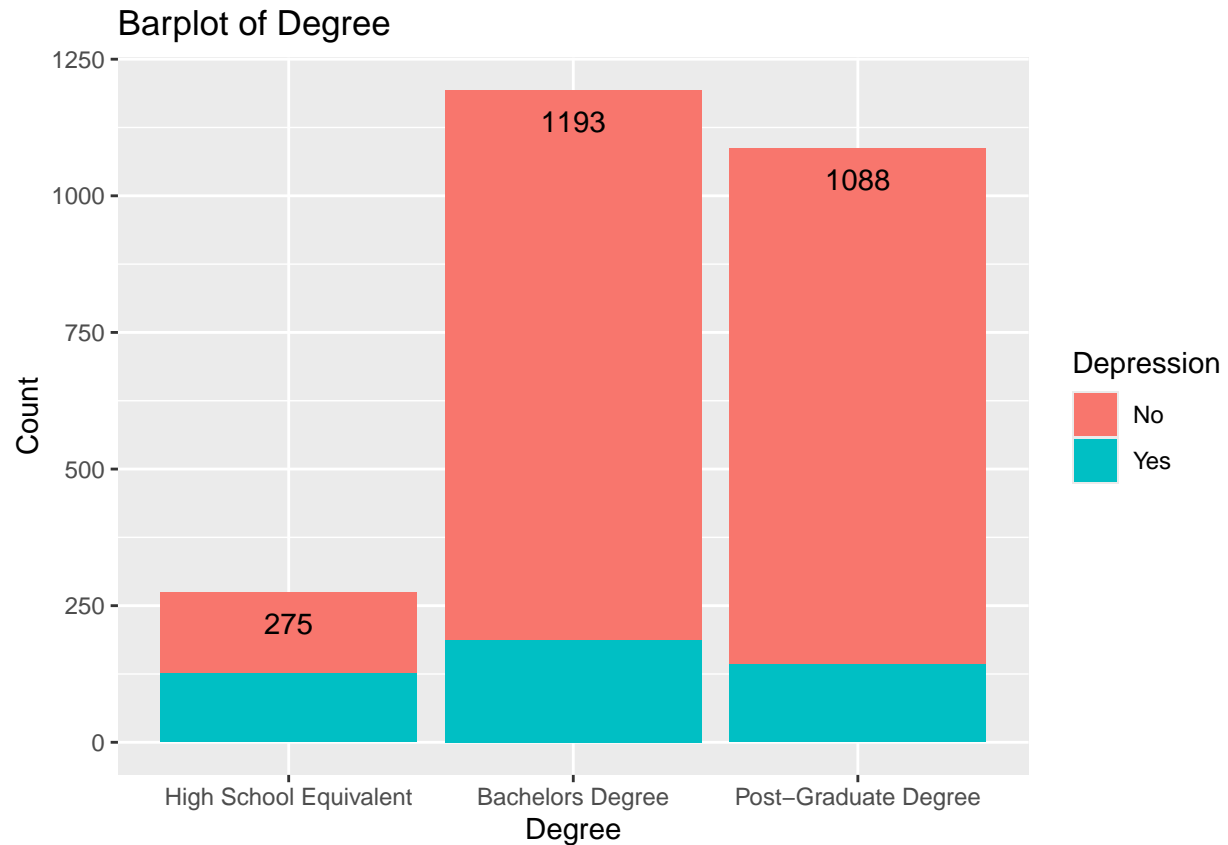
```
table(depression$Depression, depression$Dietary.Habits)
```

```
##
##      Unhealthy Moderate Healthy
## No      678      691      732
## Yes     204      141      110
```

```
prop.table(table(depression$Depression, depression$Dietary.Habits), margin = 1)
```

```
##
##      Unhealthy Moderate Healthy
## No  0.3227035 0.3288910 0.3484055
## Yes 0.4483516 0.3098901 0.2417582
```

```
# plot degree count
ggplot(depression, aes(x = Degree)) +
  geom_bar(aes(fill = Depression)) +
  xlab("Degree") +
  ylab("Count") +
  ggtitle("Barplot of Degree") +
  geom_text(aes(label = ..count..), stat = "count", vjust = 2)
```



```
table(depression$Depression, depression$Degree)
```

```
##
##      High School Equivalent Bachelors Degree Post-Graduate Degree
##   No                149                1006                946
##   Yes                126                187                142
```

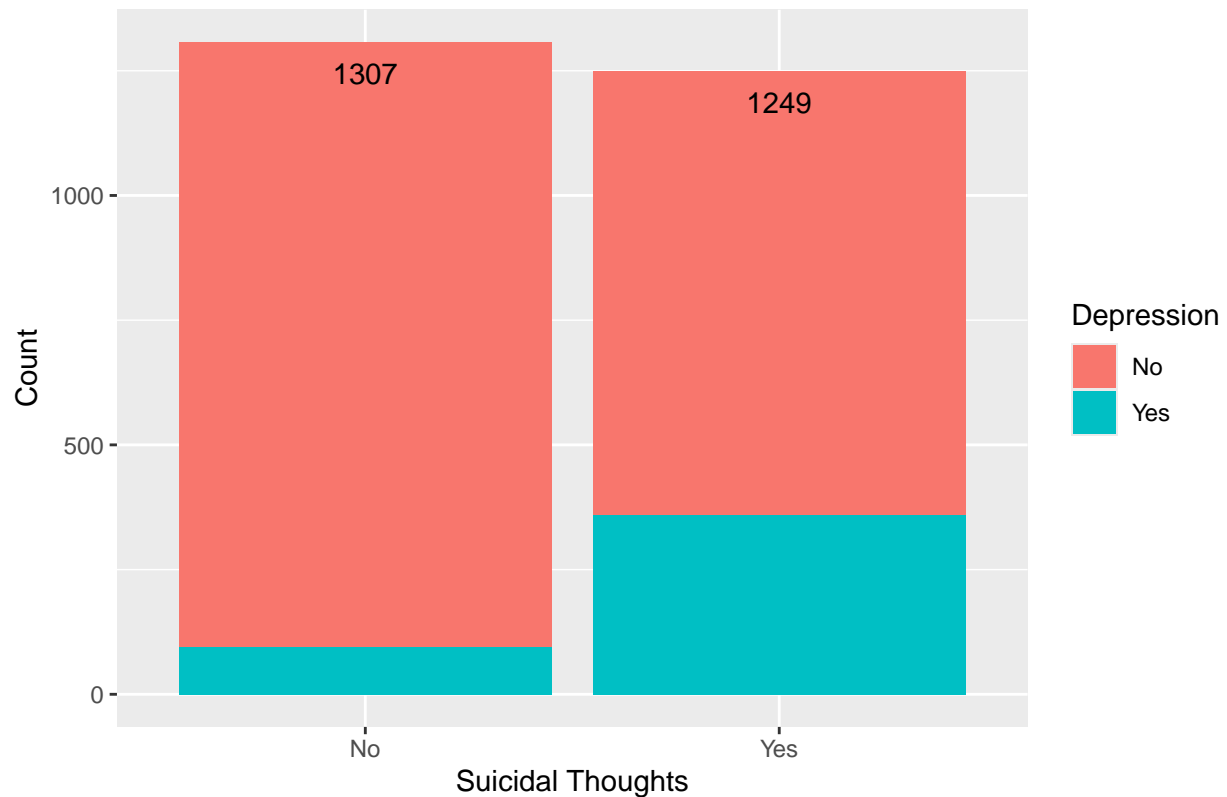
```
prop.table(table(depression$Depression, depression$Degree), margin = 1)
```

```
##
##      High School Equivalent Bachelors Degree Post-Graduate Degree
##   No                0.07091861                0.47881961                0.45026178
##   Yes                0.27692308                0.41098901                0.31208791
```

```
# plot degree count
```

```
ggplot(depression, aes(x = Have.you.ever.had.suicidal.thoughts..)) +
  geom_bar(aes(fill = Depression)) +
  xlab("Suicidal Thoughts") +
  ylab("Count") +
  ggtitle("Barplot of Suicidal Thoughts") +
  geom_text(aes(label = ..count..), stat = "count", vjust = 2)
```

Barplot of Suicidal Thoughts



```
table(depression$Depression, depression$Have.you.ever.had.suicidal.thoughts..)
```

```
##
##      No  Yes
## No 1212 889
## Yes  95 360
```

```
prop.table(table(depression$Depression, depression$Have.you.ever.had.suicidal.thoughts..), margin = 1)
```

```
##
##      No      Yes
## No 0.5768682 0.4231318
## Yes 0.2087912 0.7912088
```

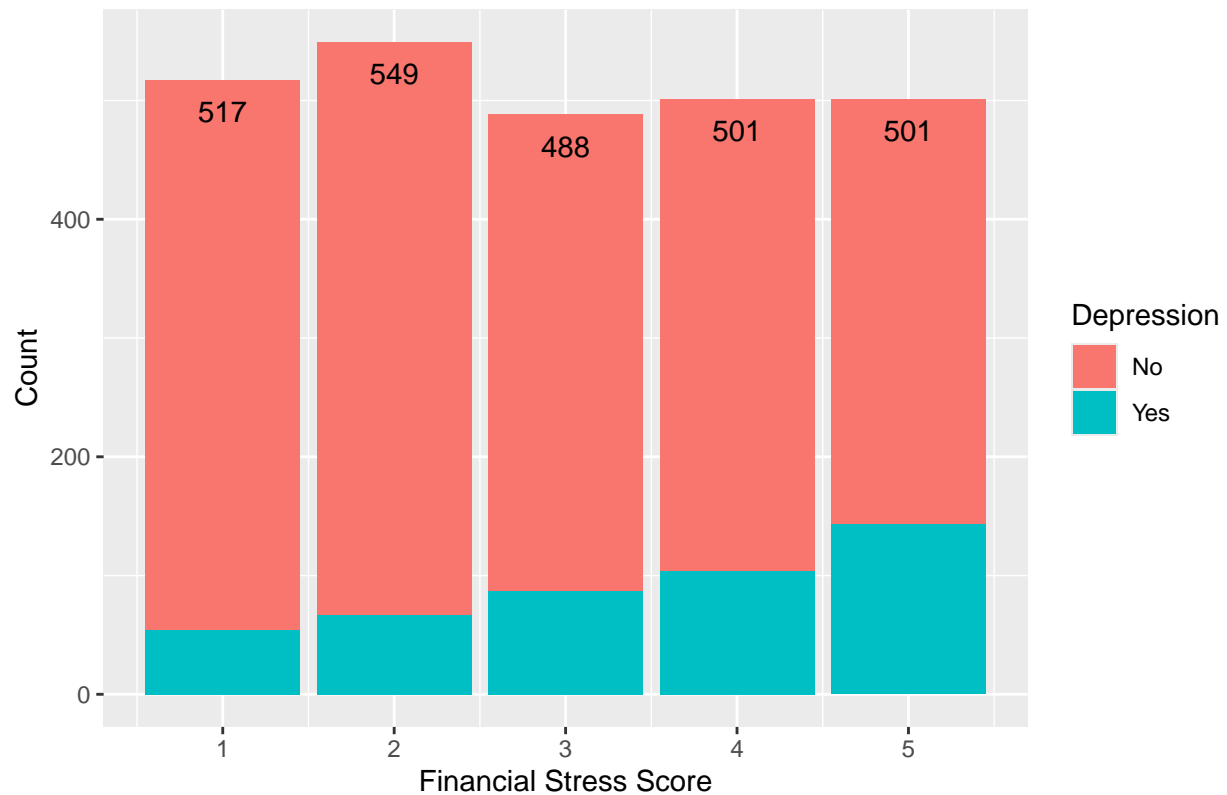
```
# delete suicidal thoughts variable
```

```
depression = subset(depression, select = -c(Have.you.ever.had.suicidal.thoughts..))
```

```
# plot financial stress count
```

```
ggplot(depression, aes(x = Financial.Stress)) +
  geom_bar(aes(fill = Depression)) +
  xlab("Financial Stress Score") +
  ylab("Count") +
  ggtitle("Barplot of Financial Stress Score") +
  geom_text(aes(label = ..count..), stat = "count", vjust = 2)
```

Barplot of Financial Stress Score



```
table(depression$Depression, depression$Financial.Stress)
```

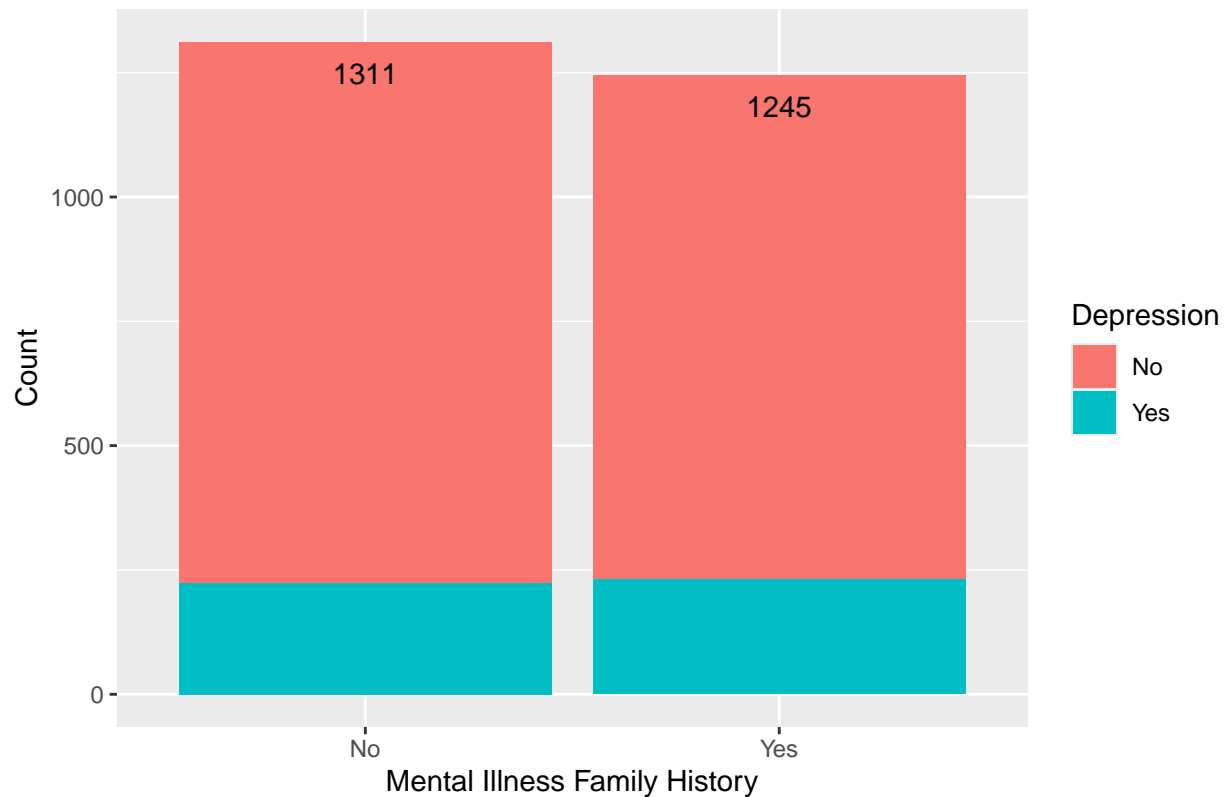
```
##
##      1  2  3  4  5
## No  463 482 401 397 358
## Yes  54  67  87 104 143
```

```
prop.table(table(depression$Depression, depression$Financial.Stress), margin = 1)
```

```
##
##      1      2      3      4      5
## No  0.2203713 0.2294146 0.1908615 0.1889576 0.1703950
## Yes 0.1186813 0.1472527 0.1912088 0.2285714 0.3142857
```

```
# plot family history of mental illness count
ggplot(depression, aes(x = Family.History.of.Mental.Illness)) +
  geom_bar(aes(fill = Depression)) +
  xlab("Mental Illness Family History") +
  ylab("Count") +
  ggtitle("Barplot of Mental Illness Family History") +
  geom_text(aes(label = ..count..), stat = "count", vjust = 2)
```

Barplot of Mental Illness Family History



```
table(depression$Depression, depression$Family.History.of.Mental.Illness)
```

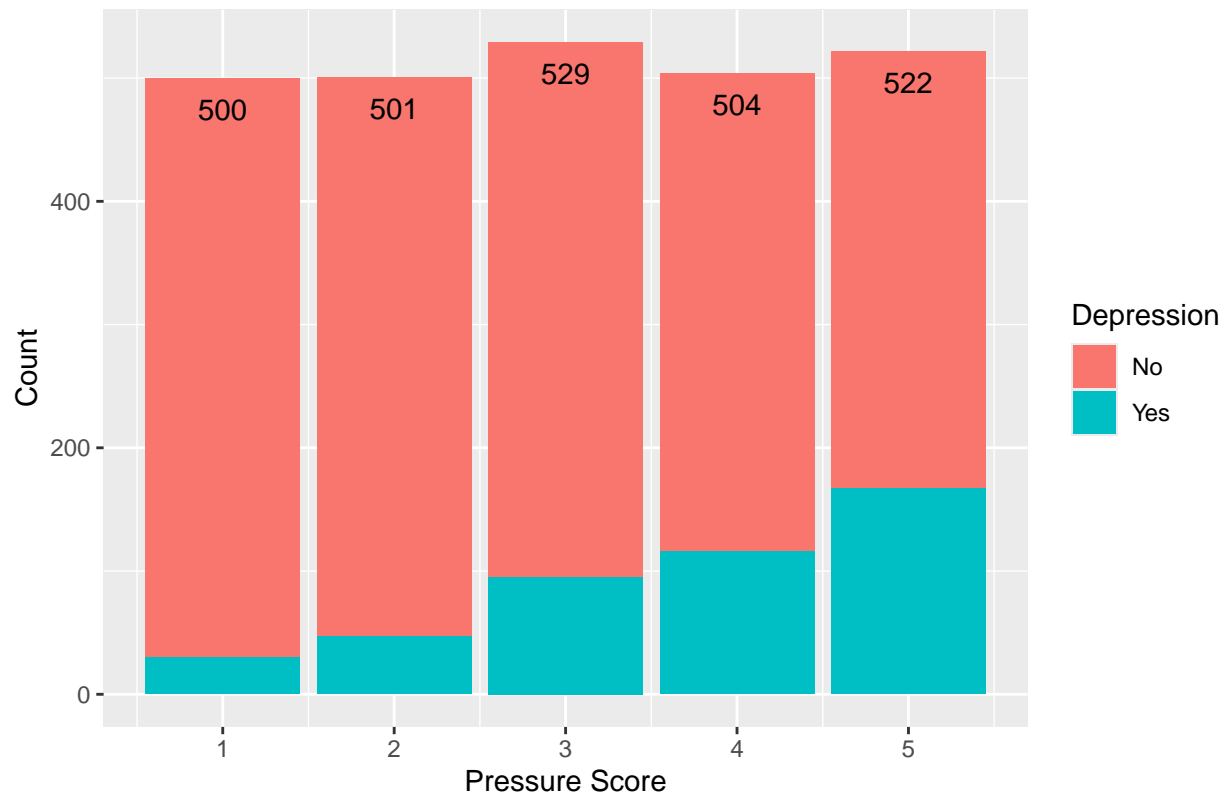
```
##
##      No  Yes
## No 1087 1014
## Yes  224  231
```

```
prop.table(table(depression$Depression, depression$Family.History.of.Mental.Illness), margin = 1)
```

```
##
##      No      Yes
## No 0.5173727 0.4826273
## Yes 0.4923077 0.5076923
```

```
# plot financial stress count
ggplot(depression, aes(x = Pressure)) +
  geom_bar(aes(fill = Depression)) +
  xlab("Pressure Score") +
  ylab("Count") +
  ggtitle("Barplot of Pressure Score") +
  geom_text(aes(label = ..count..), stat = "count", vjust = 2)
```

Barplot of Pressure Score



```
table(depression$Depression, depression$Pressure)
```

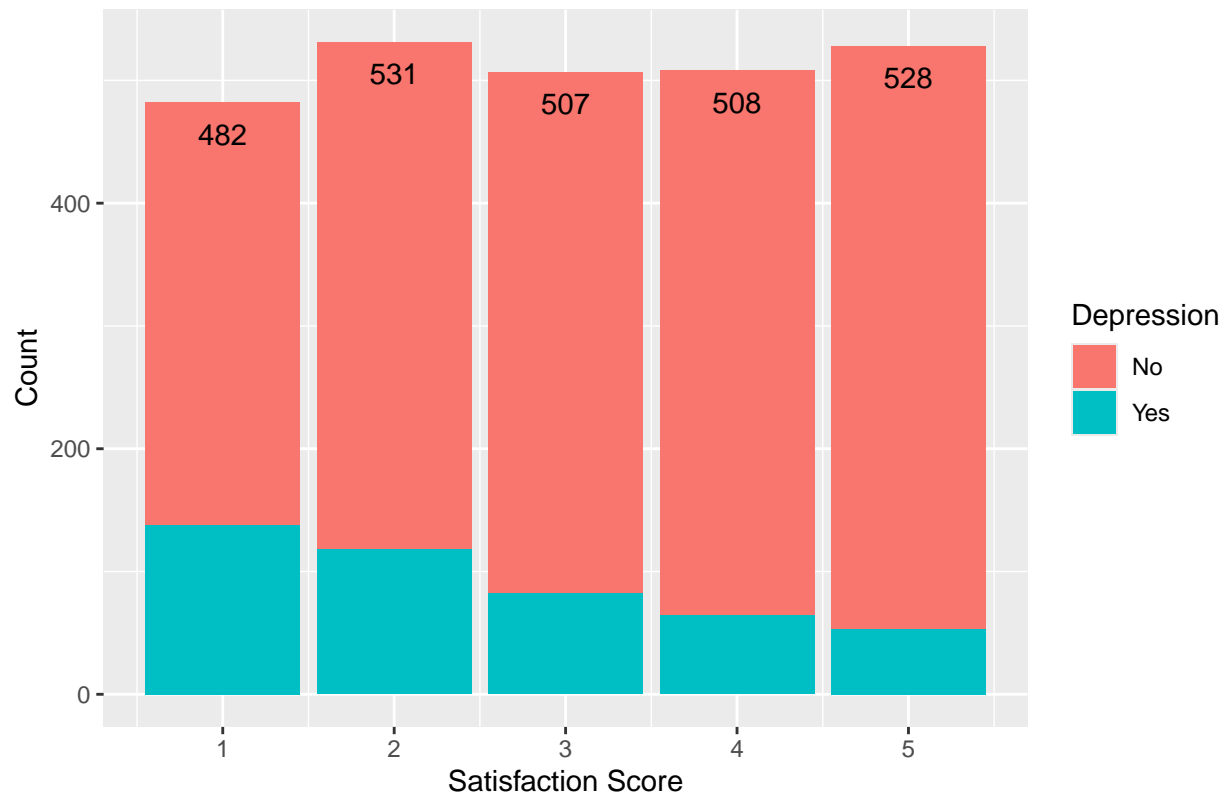
```
##
##      1  2  3  4  5
## No  470 454 434 388 355
## Yes   30  47  95 116 167
```

```
prop.table(table(depression$Depression, depression$Pressure), margin = 1)
```

```
##
##      1      2      3      4      5
## No 0.22370300 0.21608758 0.20656830 0.18467396 0.16896716
## Yes 0.06593407 0.10329670 0.20879121 0.25494505 0.36703297
```

```
ggplot(depression, aes(x = Satisfaction)) +
  geom_bar(aes(fill = Depression)) +
  xlab("Satisfaction Score") +
  ylab("Count") +
  ggtitle("Barplot of Pressure Score") +
  geom_text(aes(label = ..count..), stat = "count", vjust = 2)
```


Barplot of Pressure Score



```
table(depression$Depression, depression$Satisfaction)
```

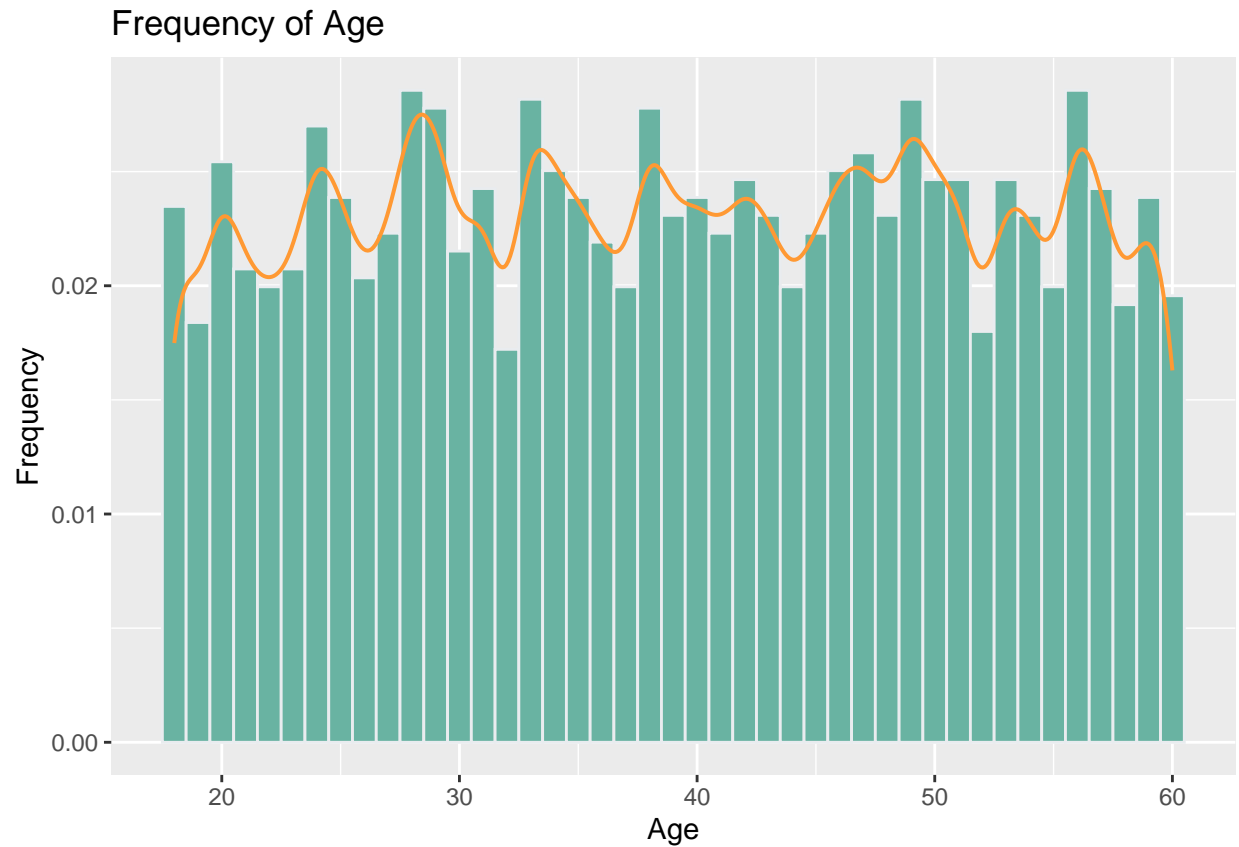
```
##
##      1  2  3  4  5
## No  344 413 425 444 475
## Yes 138 118  82  64  53
```

```
prop.table(table(depression$Depression, depression$Satisfaction), margin = 1)
```

```
##
##      1      2      3      4      5
## No 0.1637316 0.1965731 0.2022846 0.2113279 0.2260828
## Yes 0.3032967 0.2593407 0.1802198 0.1406593 0.1164835
```

```
# create specific data frames to separate those with and without risk of depression
depressionYes = depression[depression$Depression == "Yes", ]
depressionNo = depression[depression$Depression == "No", ]
```

```
ggplot(depression, aes(x = Age, y = after_stat(density))) +
  geom_histogram(binwidth = 1, fill="#69B3A2", color = "#E9ECEF") +
  geom_density(color = "#FF9933", linewidth = 0.7, adjust = 0.3) +
  ggtitle("Frequency of Age") +
  ylab("Frequency")
```



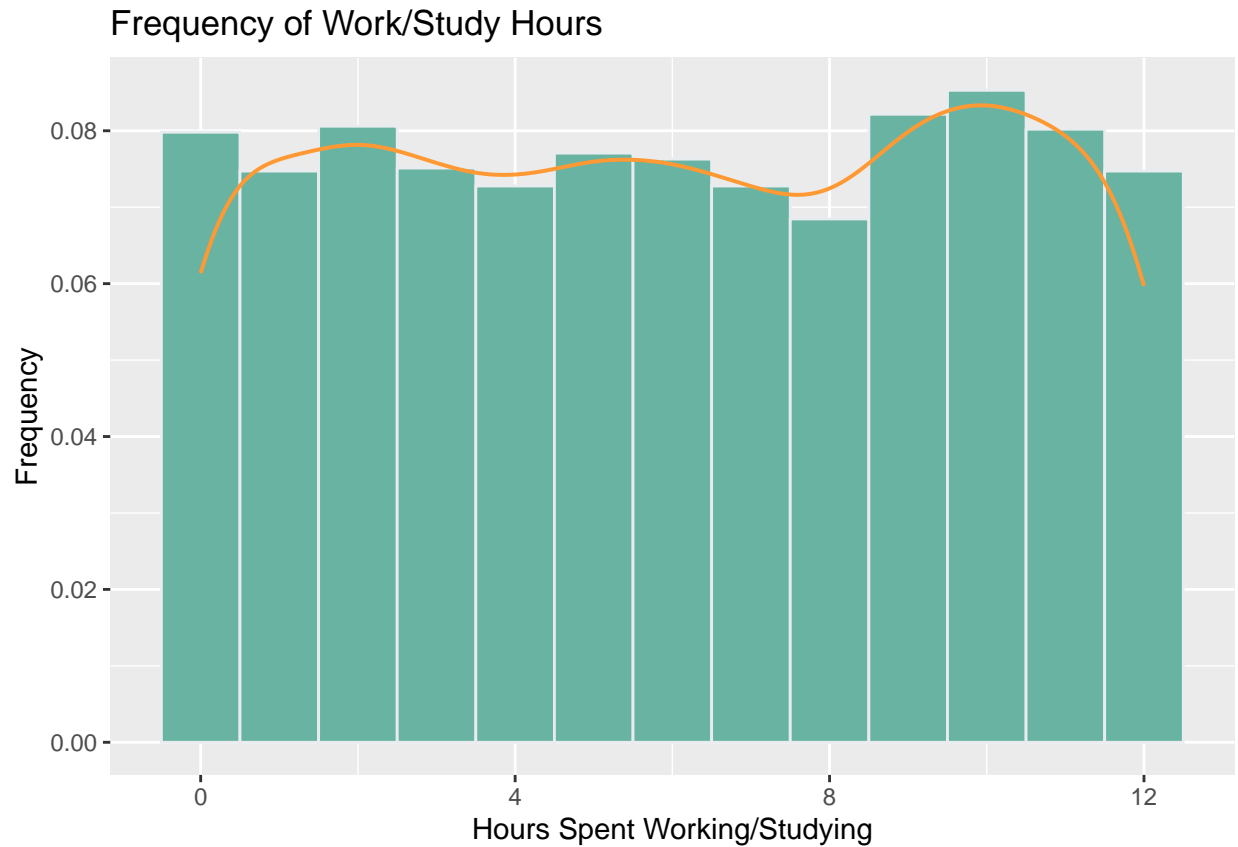
```
p1 = ggplot(depressionYes, aes(x = Age, y = after_stat(density))) +
  geom_histogram(binwidth = 1, fill="#69B3A2", color = "#E9ECEF") +
  geom_density(color = "#FF9933", linewidth = 0.7, adjust = 0.3) +
  ggtitle("Age of Depressed Yes") +
  ylab("Frequency") +
  ylim(0, 0.10)

p2 = ggplot(depressionNo, aes(x = Age, y = after_stat(density))) +
  geom_histogram(binwidth = 1, fill="#69B3A2", color = "#E9ECEF") +
  geom_density(color = "#FF9933", linewidth = 0.7, adjust = 0.3) +
  ggtitle("Age of Depressed No") +
  ylab("Frequency") +
  ylim(0, 0.10)

cowplot::plot_grid(p1, p2)
```



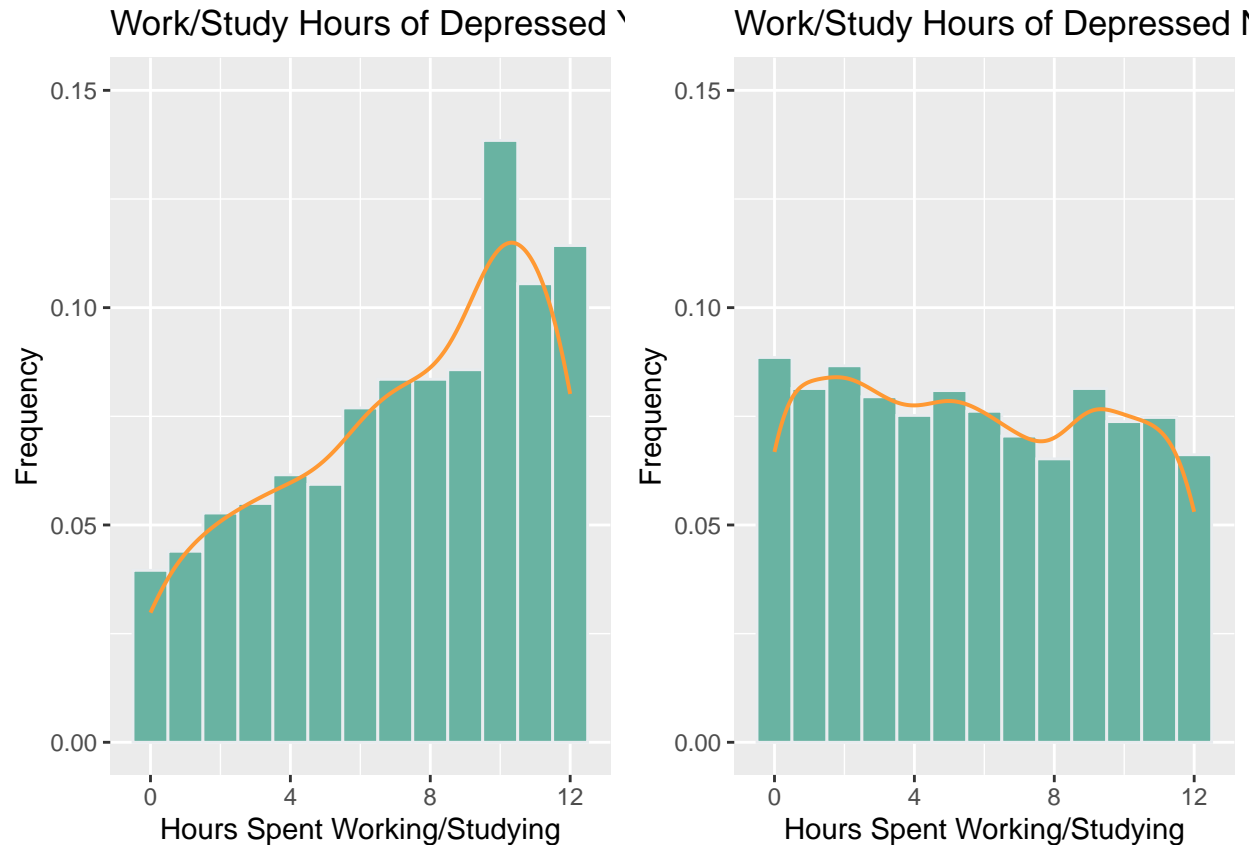
```
ggplot(depression, aes(x = Work.Study.Hours, y = after_stat(density))) +
  geom_histogram(binwidth = 1, fill="#69B3A2", color = "#E9ECEF") +
  geom_density(color = "#FF9933", linewidth = 0.7, adjust = 1) +
  ggtitle("Frequency of Work/Study Hours") +
  xlab("Hours Spent Working/Studying") +
  ylab("Frequency")
```



```
p3 = ggplot(depressionYes, aes(x = Work.Study.Hours, y = after_stat(density))) +
  geom_histogram(binwidth = 1, fill="#69B3A2", color = "#E9ECEF") +
  geom_density(color = "#FF9933", linewidth = 0.7, adjust = 1) +
  ggtitle("Work/Study Hours of Depressed Yes") +
  xlab("Hours Spent Working/Studying") +
  ylab("Frequency") +
  ylim(0, 0.15)
```

```
p4 = ggplot(depressionNo, aes(x = Work.Study.Hours, y = after_stat(density))) +
  geom_histogram(binwidth = 1, fill="#69B3A2", color = "#E9ECEF") +
  geom_density(color = "#FF9933", linewidth = 0.7, adjust = 1) +
  ggtitle("Work/Study Hours of Depressed No") +
  xlab("Hours Spent Working/Studying") +
  ylab("Frequency") +
  ylim(0, 0.15)
```

```
cowplot::plot_grid(p3, p4, ncol = 2)
```



```
# create train and test set
set.seed(213)
index = createDataPartition(depression$Depression, p = 0.80, list = FALSE, times = 1)
depression_train = depression[index,]
depression_test = depression[-index,]
```

```
# create model with all predictors (no interaction effects)
depression_glm = glm(Depression ~ ., data = depression_train, family = "binomial")
summary(depression_glm)
```

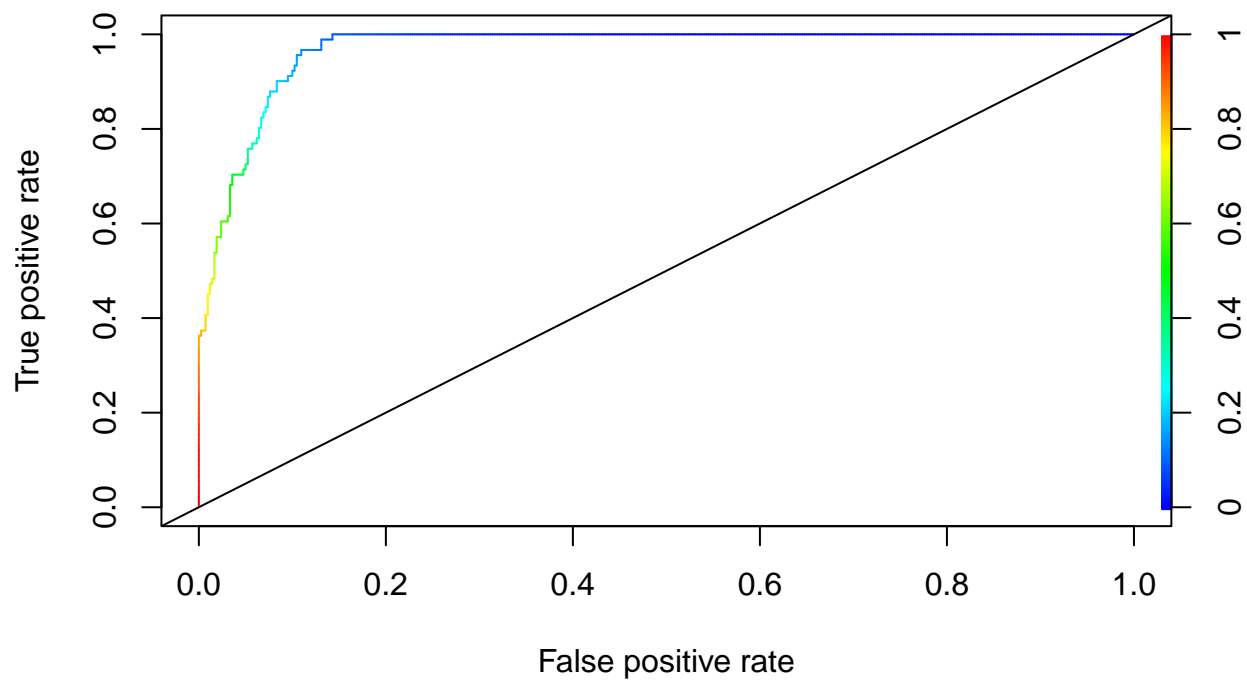
```
##
## Call:
## glm(formula = Depression ~ ., family = "binomial", data = depression_train)
##
## Coefficients:
##
##              Estimate Std. Error z value
## (Intercept)      2.96631    0.59651   4.973
## GenderMale      -0.12770    0.19393  -0.658
## Age             -0.22468    0.01667 -13.478
## Working.Professional.or.StudentWorking Professional -1.70515    0.22278  -7.654
## Sleep.Duration5-6 hours -0.44304    0.26513  -1.671
## Sleep.Duration7-8 hours -0.92674    0.26201  -3.537
## Sleep.DurationMore than 8 hours -1.31202    0.28084  -4.672
## Dietary.HabitsModerate -0.64680    0.23387  -2.766
## Dietary.HabitsHealthy -1.37469    0.24309  -5.655
## DegreeBachelors Degree -0.31858    0.28583  -1.115
## DegreePost-Graduate Degree -0.36122    0.30184  -1.197
```

```

## Work.Study.Hours          0.23994    0.02808    8.545
## Financial.Stress          0.68694    0.07639    8.993
## Family.History.of.Mental.IllnessYes 0.71444    0.19588    3.647
## Pressure                  1.12794    0.08818   12.791
## Satisfaction             -0.89314    0.08269  -10.801
##                           Pr(>|z|)
## (Intercept)              6.60e-07 ***
## GenderMale               0.510240
## Age                      < 2e-16 ***
## Working.Professional.or.StudentWorking Professional 1.95e-14 ***
## Sleep.Duration5-6 hours   0.094711 .
## Sleep.Duration7-8 hours   0.000405 ***
## Sleep.DurationMore than 8 hours 2.99e-06 ***
## Dietary.HabitsModerate     0.005680 **
## Dietary.HabitsHealthy      1.56e-08 ***
## DegreeBachelors Degree     0.265042
## DegreePost-Graduate Degree 0.231408
## Work.Study.Hours          < 2e-16 ***
## Financial.Stress          < 2e-16 ***
## Family.History.of.Mental.IllnessYes 0.000265 ***
## Pressure                  < 2e-16 ***
## Satisfaction              < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##    Null deviance: 1915.51  on 2044  degrees of freedom
## Residual deviance:  721.32  on 2029  degrees of freedom
## AIC: 753.32
##
## Number of Fisher Scoring iterations: 7
# draw a roc curve for true positive rate and true negative rate to find the optimal cutoff
glm_predictions = predict(depression_glm, newdata = depression_test, type = "response")
prob_predictions = prediction(glm_predictions, depression_test$Depression)
roc_curve = performance(prob_predictions, "tpr", "fpr")
plot(roc_curve, colorize = TRUE, main = "Model 1 (Only Main Effects) ROC Curve - TPR/FPR")
abline(0, 1)

```

Model 1 (Only Main Effects) ROC Curve – TPR/FPR

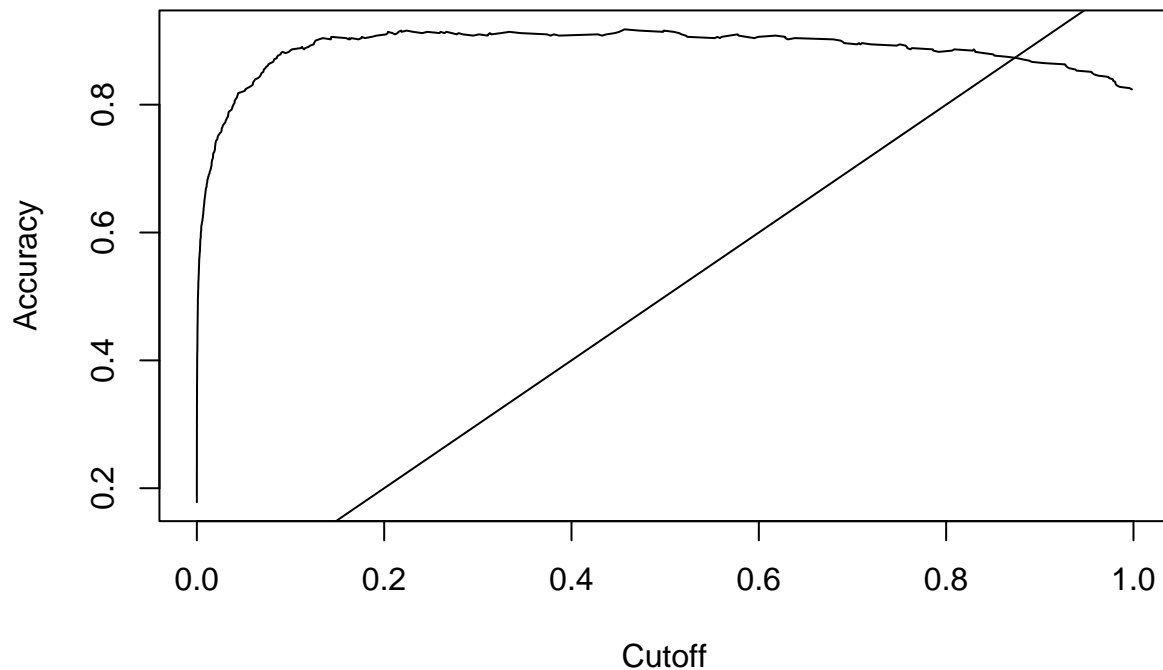


```
# auc value  
unlist(slot(performance(prob_predictions, "auc"), "y.values"))
```

```
## [1] 0.9692046
```

```
acc = performance(prob_predictions, "acc")  
plot(acc, main = "Model 1 (Only Main Effects) ROC Curve - Accuracy")  
abline(0, 1)
```

Model 1 (Only Main Effects) ROC Curve – Accuracy



```
glm_predictions2 = predict(depression_glm, newdata = depression_test)
glm_predictions2 = ifelse(glm_predictions2 > 0.35, "Yes", "No")
glm_predictions2 = as.factor(glm_predictions2)
confusionMatrix(glm_predictions2, depression_test$Depression)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No  Yes
##      No  410  39
##      Yes   10  52
##
##           Accuracy : 0.9041
##           95% CI : (0.8752, 0.9282)
##      No Information Rate : 0.8219
##      P-Value [Acc > NIR] : 1.288e-07
##
##           Kappa : 0.6257
##
##      Mcnemar's Test P-Value : 6.334e-05
##
##           Sensitivity : 0.9762
##           Specificity : 0.5714
##      Pos Pred Value : 0.9131
##      Neg Pred Value : 0.8387
##           Prevalence : 0.8219
```



```
##          Detection Rate : 0.8023
##    Detection Prevalence : 0.8787
##          Balanced Accuracy : 0.7738
##
##          'Positive' Class : No
##
# create models for interaction effects of each categorical variable and see if there are any significant
# summary(glm(Depression ~ Gender*., data = depression_train, family = "binomial"))
# summary(glm(Depression ~ Working.Professional.or.Student*., data = depression_train, family = "binomial"))
# summary(glm(Depression ~ Sleep.Duration*., data = depression_train, family = "binomial"))
# summary(glm(Depression ~ Dietary.Habits*., data = depression_train, family = "binomial"))
# summary(glm(Depression ~ Degree*., data = depression_train, family = "binomial"))
# summary(glm(Depression ~ Work.Study.Hours*., data = depression_train, family = "binomial"))
# summary(glm(Depression ~ Financial.Stress*., data = depression_train, family = "binomial"))
# summary(glm(Depression ~ Family.History.of.Mental.Illness*., data = depression_train, family = "binomial"))
# summary(glm(Depression ~ Pressure*., data = depression_train, family = "binomial"))
# summary(glm(Depression ~ Satisfaction*., data = depression_train, family = "binomial"))
```

None of the interaction effects were meaningfully significant; we will not be adding interaction effects to our model.

```
# create a table to easily see top important predictors and their odds for the first model
vI = cbind(varImp(depression_glm), Odds = exp(summary(depression_glm)$coefficients[-1, 1]), PValue = summary(depression_glm)$p.value[-1, 1])
vI = vI[order(-vI$Overall), , drop = FALSE]
vI
```

	Overall	Odds
## Age	13.4776802	0.7987731
## Pressure	12.7907714	3.0892894
## Satisfaction	10.8006607	0.4093688
## Financial.Stress	8.9929453	1.9876318
## Work.Study.Hours	8.5447237	1.2711695
## Working.Professional.or.StudentWorking Professional	7.6538603	0.1817443
## Dietary.HabitsHealthy	5.6551330	0.2529184
## Sleep.DurationMore than 8 hours	4.6717971	0.2692760
## Family.History.of.Mental.IllnessYes	3.6473649	2.0430348
## Sleep.Duration7-8 hours	3.5369801	0.3958427
## Dietary.HabitsModerate	2.7657051	0.5237177
## Sleep.Duration5-6 hours	1.6710548	0.6420819
## DegreePost-Graduate Degree	1.1967390	0.6968255
## DegreeBachelors Degree	1.1145532	0.7271838
## GenderMale	0.6584642	0.8801194
##	PValue	
## Age	2.116574e-41	
## Pressure	1.846321e-37	
## Satisfaction	3.417355e-27	
## Financial.Stress	2.406920e-19	
## Work.Study.Hours	1.288442e-17	

```

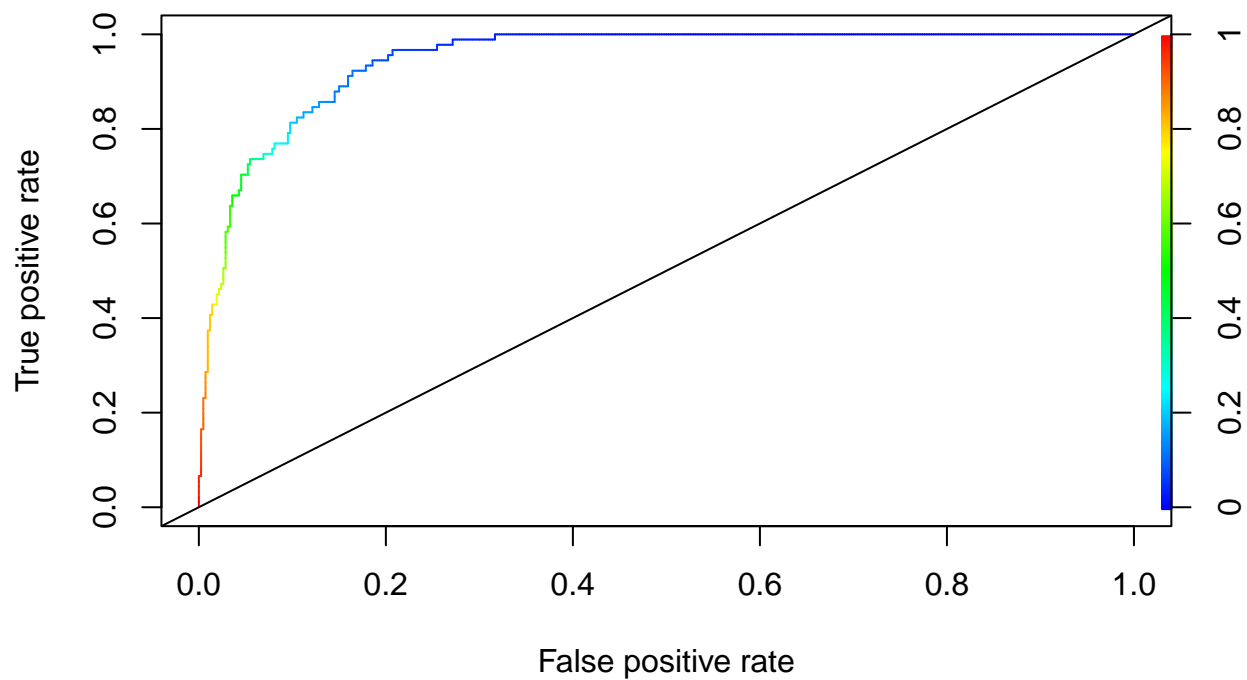
## Working.Professional.or.StudentWorking Professional 1.950341e-14
## Dietary.HabitsHealthy 1.557257e-08
## Sleep.DurationMore than 8 hours 2.985759e-06
## Family.History.of.Mental.IllnessYes 2.649435e-04
## Sleep.Duration7-8 hours 4.047301e-04
## Dietary.HabitsModerate 5.679988e-03
## Sleep.Duration5-6 hours 9.471084e-02
## DegreePost-Graduate Degree 2.314083e-01
## DegreeBachelors Degree 2.650419e-01
## GenderMale 5.102399e-01

depression_glm2 = glm(Depression ~ Age + Pressure + Satisfaction + Work.Study.Hours + Financial.Stress,
summary(depression_glm2)

##
## Call:
## glm(formula = Depression ~ Age + Pressure + Satisfaction + Work.Study.Hours +
##      Financial.Stress, family = "binomial", data = depression_train)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.53847    0.48141   3.196  0.00139 **
## Age           -0.23489    0.01378 -17.052 < 2e-16 ***
## Pressure       0.97663    0.07620  12.817 < 2e-16 ***
## Satisfaction  -0.73730    0.07058 -10.446 < 2e-16 ***
## Work.Study.Hours 0.21818    0.02540   8.590 < 2e-16 ***
## Financial.Stress 0.59007    0.06725   8.774 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 1915.5  on 2044  degrees of freedom
## Residual deviance:  850.5  on 2039  degrees of freedom
## AIC: 862.5
##
## Number of Fisher Scoring iterations: 7
# draw a roc curve for true positive rate and true negative rate to find the optimal cutoff
glm_predictions3 = predict(depression_glm2, newdata = depression_test, type = "response")
prob_predictions2 = prediction(glm_predictions3, depression_test$Depression)
roc_curve2 = performance(prob_predictions2, "tpr", "fpr")
plot(roc_curve2, colorize = TRUE, main = "Model 13 (Only Main Effects) ROC Curve - TPR/FPR")
abline(0, 1)

```

Model 13 (Only Main Effects) ROC Curve – TPR/FPR

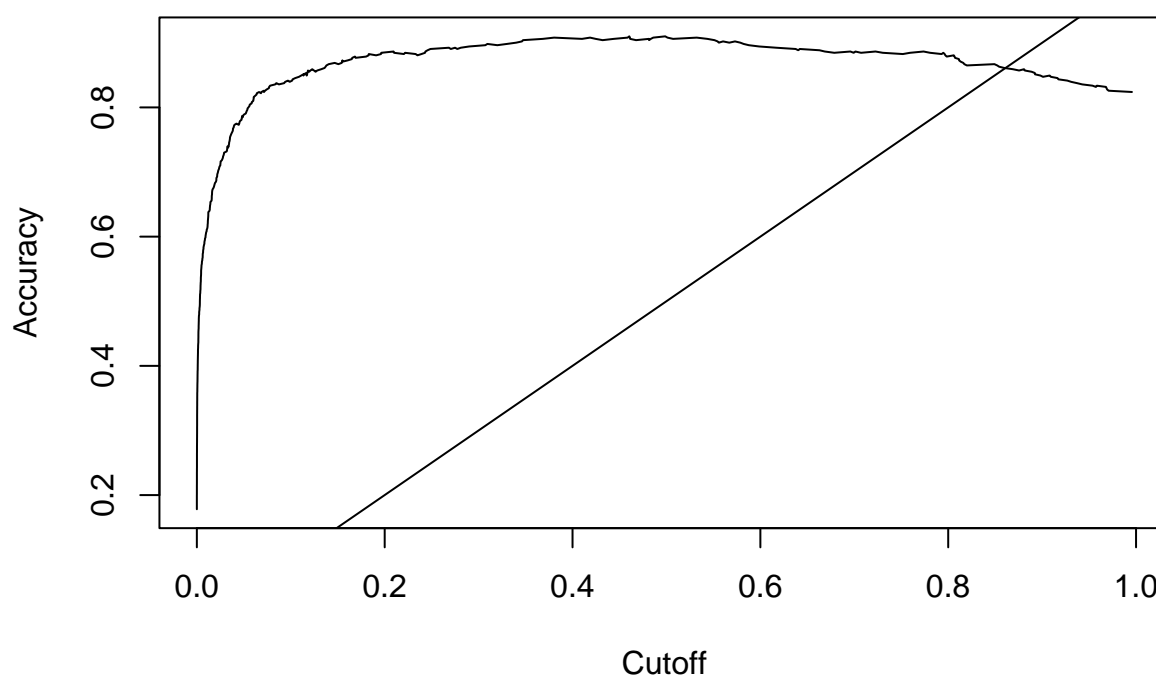


```
# auc value
unlist(slot(performance(prob_predictions2, "auc"), "y.values"))

## [1] 0.9474359

acc2 = performance(prob_predictions2, "acc")
plot(acc2, main = "Model 13 (Only Main Effects) ROC Curve - Accuracy")
abline(0, 1)
```

Model 13 (Only Main Effects) ROC Curve – Accuracy



```
glm_predictions4 = predict(depression_glm2, newdata = depression_test)
glm_predictions4 = ifelse(glm_predictions4 > 0.35, "Yes", "No")
glm_predictions4 = as.factor(glm_predictions4)
confusionMatrix(glm_predictions4, depression_test$Depression)
```

```
## Confusion Matrix and Statistics
```

```
##
```

```
##           Reference
```

```
## Prediction No Yes
```

```
##           No  408  41
```

```
##           Yes   12  50
```

```
##
```

```
##           Accuracy : 0.8963
```

```
##           95% CI : (0.8665, 0.9213)
```

```
##           No Information Rate : 0.8219
```

```
##           P-Value [Acc > NIR] : 1.978e-06
```

```
##
```

```
##           Kappa : 0.5952
```

```
##
```

```
##           McNemar's Test P-Value : 0.00012
```

```
##
```

```
##           Sensitivity : 0.9714
```

```
##           Specificity : 0.5495
```

```
##           Pos Pred Value : 0.9087
```

```
##           Neg Pred Value : 0.8065
```

```
##           Prevalence : 0.8219
```

```

##          Detection Rate : 0.7984
##    Detection Prevalence : 0.8787
##          Balanced Accuracy : 0.7604
##
##          'Positive' Class : No
##

# create a table to easily see top important predictors and their odds for the second model
vI2 = cbind(varImp(depression_glm2), Odds = exp(summary(depression_glm2)$coefficients[-1, 1]), PValue =
vI2 = vI2[order(-vI2$Overall), , drop = FALSE]
vI2

##              Overall      Odds      PValue
## Age              17.051838 0.7906572 3.387123e-65
## Pressure         12.816884 2.6555037 1.318950e-37
## Satisfaction     10.446477 0.4784051 1.520692e-25
## Financial.Stress  8.773779 1.8041063 1.727681e-18
## Work.Study.Hours  8.590176 1.2438171 8.683634e-18

paste("First Model Residual Deviance: ", depression_glm$deviance)

## [1] "First Model Residual Deviance: 721.315547208141"

paste("Second Model Residual Deviance: ", depression_glm2$deviance)

## [1] "Second Model Residual Deviance: 850.504771136837"

train_control = trainControl(method = "repeatedcv", number = 10, repeats = 3, classProbs = TRUE)
depression_cvglm = train(Depression ~ .,
                        data = depression_train,
                        method = "glm",
                        family = binomial,
                        trControl = train_control)

depression_cvglm$results

## parameter Accuracy      Kappa AccuracySD      KappaSD
## 1      none 0.8964966 0.6350167 0.02113243 0.07054203

cvglm_predictions = predict(depression_cvglm, depression_test)
confusionMatrix(cvglm_predictions, depression_test$Depression)

## Confusion Matrix and Statistics
##
##              Reference
## Prediction  No Yes
##      No    406  30
##      Yes    14  61
##
##              Accuracy : 0.9139
##              95% CI : (0.8861, 0.9367)
##      No Information Rate : 0.8219
##      P-Value [Acc > NIR] : 2.579e-09
##
##              Kappa : 0.6841
##
##      McNemar's Test P-Value : 0.02374
##

```

```
##          Sensitivity : 0.9667
##          Specificity : 0.6703
##          Pos Pred Value : 0.9312
##          Neg Pred Value : 0.8133
##          Prevalence : 0.8219
##          Detection Rate : 0.7945
##          Detection Prevalence : 0.8532
##          Balanced Accuracy : 0.8185
##
##          'Positive' Class : No
##
```

```
varImp(depression_cvglm)
```

```
## glm variable importance
##
##                                     Overall
## Age                               100.000
## Pressure                           94.642
## Satisfaction                        79.117
## Financial.Stress                    65.016
## Work.Study.Hours                    61.519
## `Working.Professional.or.StudentWorking Professional` 54.570
## Dietary.HabitsHealthy                38.978
## `Sleep.DurationMore than 8 hours`    31.307
## Family.History.of.Mental.IllnessYes  23.316
## `Sleep.Duration7-8 hours`            22.455
## Dietary.HabitsModerate                16.438
## `Sleep.Duration5-6 hours`             7.899
## `DegreePost-Graduate Degree`          4.199
## `DegreeBachelors Degree`              3.558
## GenderMale                           0.000
```

```
train_control2 = trainControl(method = "repeatedcv", number = 10, repeats = 3, classProbs = TRUE)
depression_cvglm2 = train(Depression ~ Age + Pressure + Satisfaction + Work.Study.Hours + Financial.Str
                          data = depression_train,
                          method = "glm",
                          family = binomial,
                          trControl = train_control2)
```

```
depression_cvglm2$results
```

```
##   parameter Accuracy      Kappa AccuracySD      KappaSD
## 1      none 0.8960062 0.6242043 0.02380412 0.08394406
```

```
cvglm_predictions2 = predict(depression_cvglm2, depression_test)
confusionMatrix(cvglm_predictions2, depression_test$Depression)
```

```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction  No  Yes
##          No  405  32
##          Yes  15  59
##
##          Accuracy : 0.908
```

```

##          95% CI : (0.8796, 0.9316)
##    No Information Rate : 0.8219
##    P-Value [Acc > NIR] : 2.887e-08
##
##          Kappa : 0.661
##
##    McNemar's Test P-Value : 0.0196
##
##          Sensitivity : 0.9643
##          Specificity : 0.6484
##          Pos Pred Value : 0.9268
##          Neg Pred Value : 0.7973
##          Prevalence : 0.8219
##          Detection Rate : 0.7926
##          Detection Prevalence : 0.8552
##          Balanced Accuracy : 0.8063
##
##          'Positive' Class : No
##

```

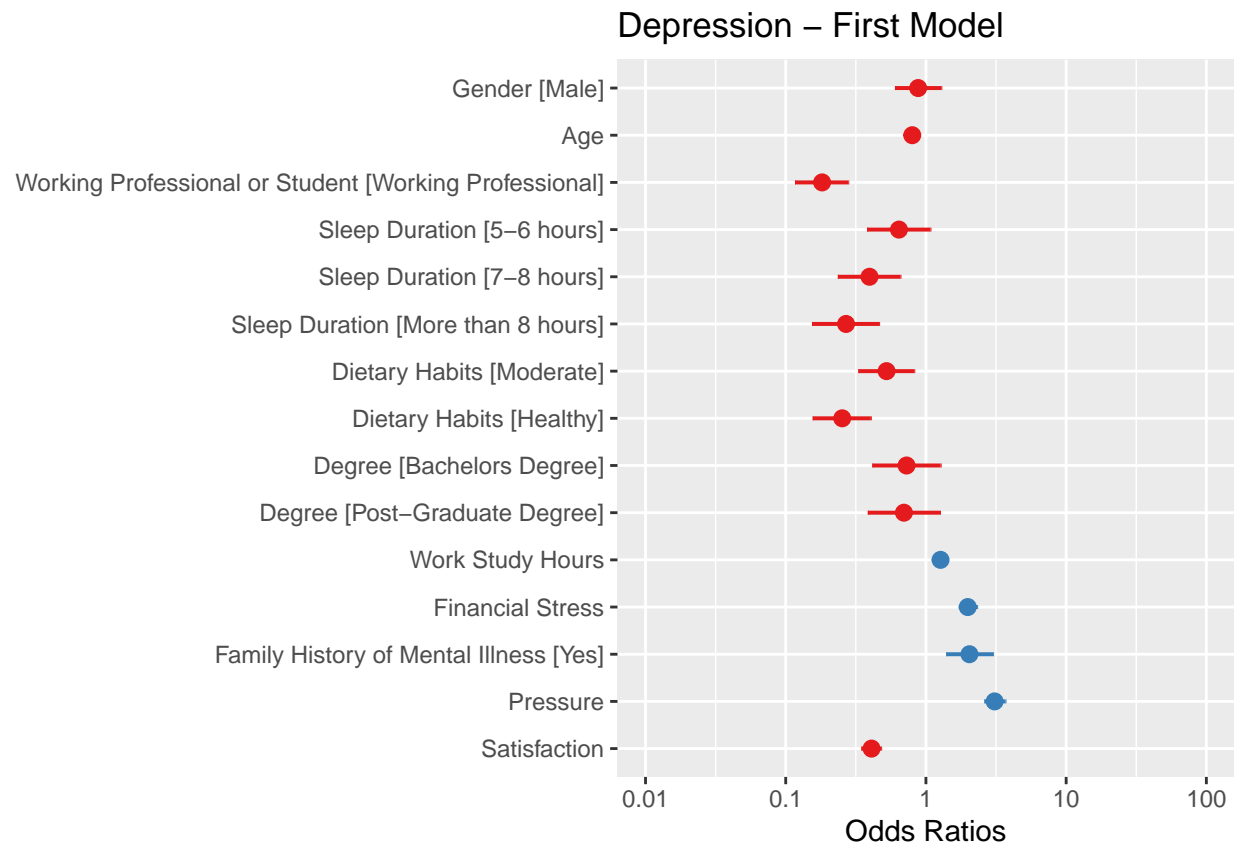
```
varImp(depression_cvglm2)
```

```

## glm variable importance
##
##          Overall
## Age          100.00
## Pressure      49.95
## Satisfaction  21.94
## Financial.Stress  2.17
## Work.Study.Hours  0.00

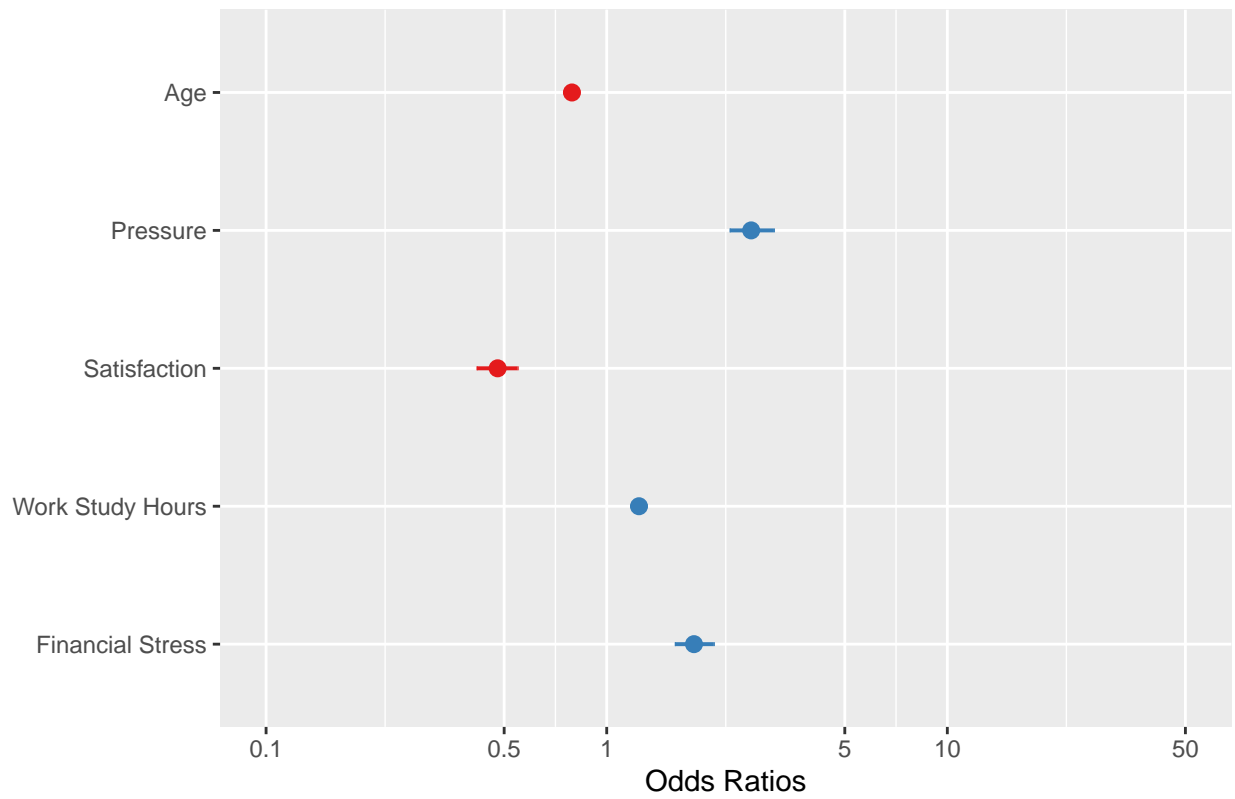
```

```
plot_model(depression_glm, title = "Depression - First Model")
```

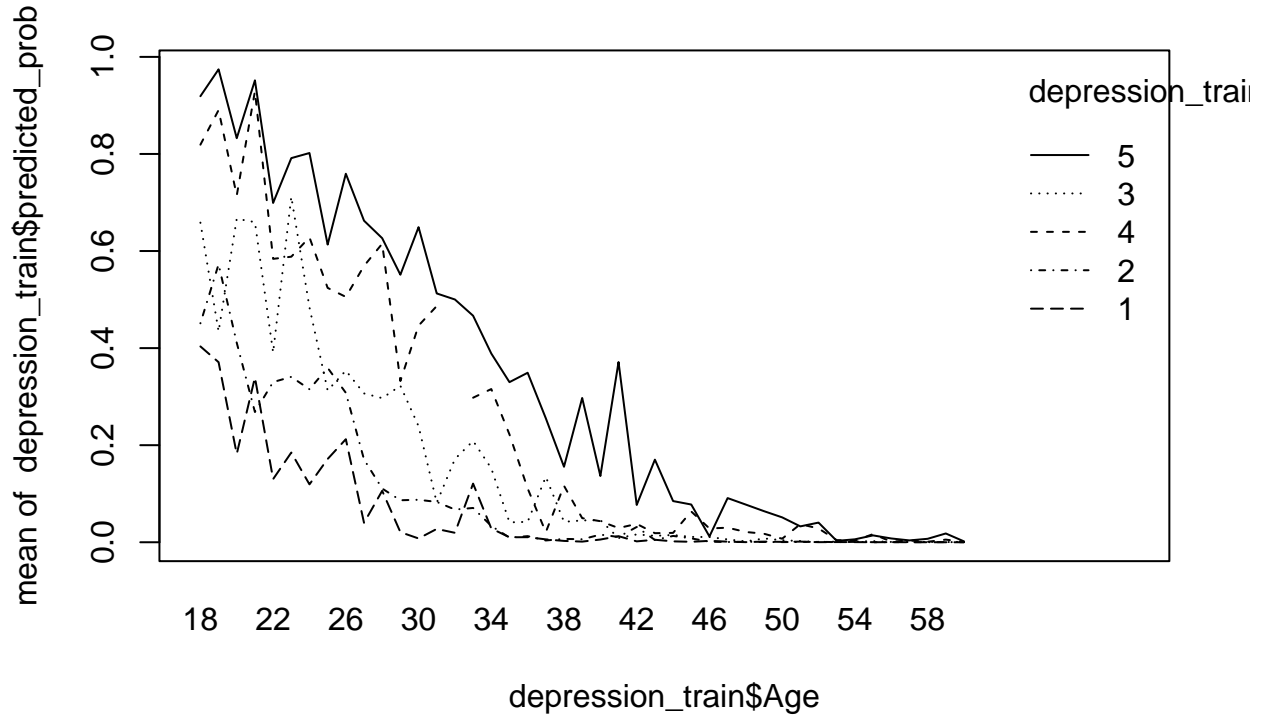


```
plot_model(depression_glm2, title = "Depression - Second Model")
```


Depression – Second Model



```
depression_train$predicted_prob <- predict(depression_glm, type = "response")  
interaction.plot(depression_train$Age, depression_train$Pressure, depression_train$predicted_prob)
```



$$\text{logit}(p) = 1.538 - 0.235 * \text{Age} + 0.977 * \text{Pressure} - 0.737 * \text{Satisfaction} + 0.218 * \text{Work.Study.Hours} + 0.590 * \text{Financial.Stress}$$

$$\text{where } \text{logit}(p) = \ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n +$$