

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
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
# find number of NAs for each column
sapply(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"
```

```
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", "More than 10 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$Financial.Stress = factor(depression$Financial.Stress, levels = c(1, 2, 3, 4, 5))
```

```
depression$Family.History.of.Mental.Illness = as.factor(depression$Family.History.of.Mental.Illness)
```

```
depression$Depression = as.factor(depression$Depression)
```

```
depression$Pressure = factor(depression$Pressure, levels = c(1, 2, 3, 4, 5))
```

```
depression$Satisfaction = factor(depression$Satisfaction, levels = c(1, 2, 3, 4, 5))
```

```
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
##          TRUE
##          0
##
## $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
##
## $Financial.Stress
##
## 1 2 3 4 5
## 517 549 488 501 501
##
## $Family.History.of.Mental.Illness
##
## No Yes
## 1311 1245
##
## $Depression
##
## No Yes
## 2101 455
##
## $Pressure
##
## 1 2 3 4 5
## 500 501 529 504 522
##
## $Satisfaction
##
## 1 2 3 4 5
## 482 531 507 508 528

```

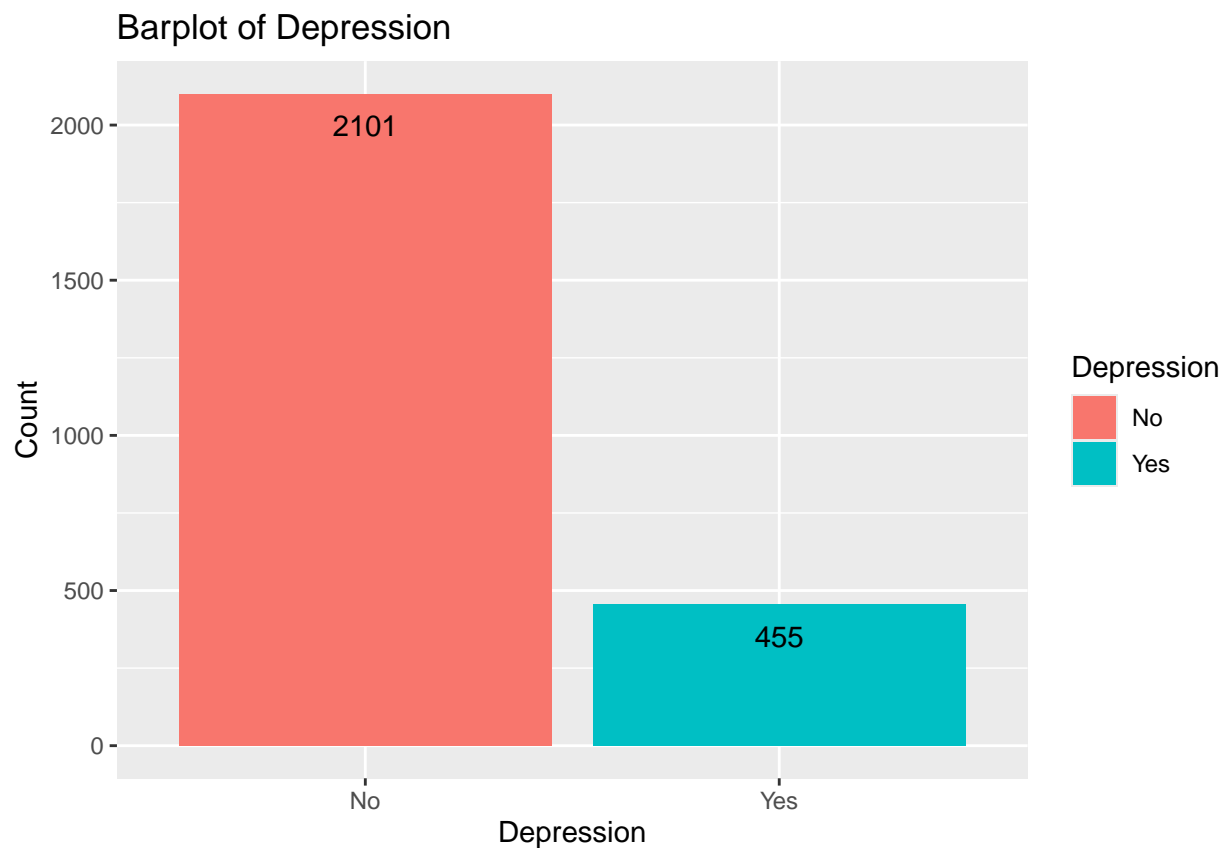
IF YOU WANT TO CHANGE THE COLOR, PLEASE USE THESE TWO LINKS:

<https://sape.inf.usi.ch/quick-reference/ggplot2/colour>

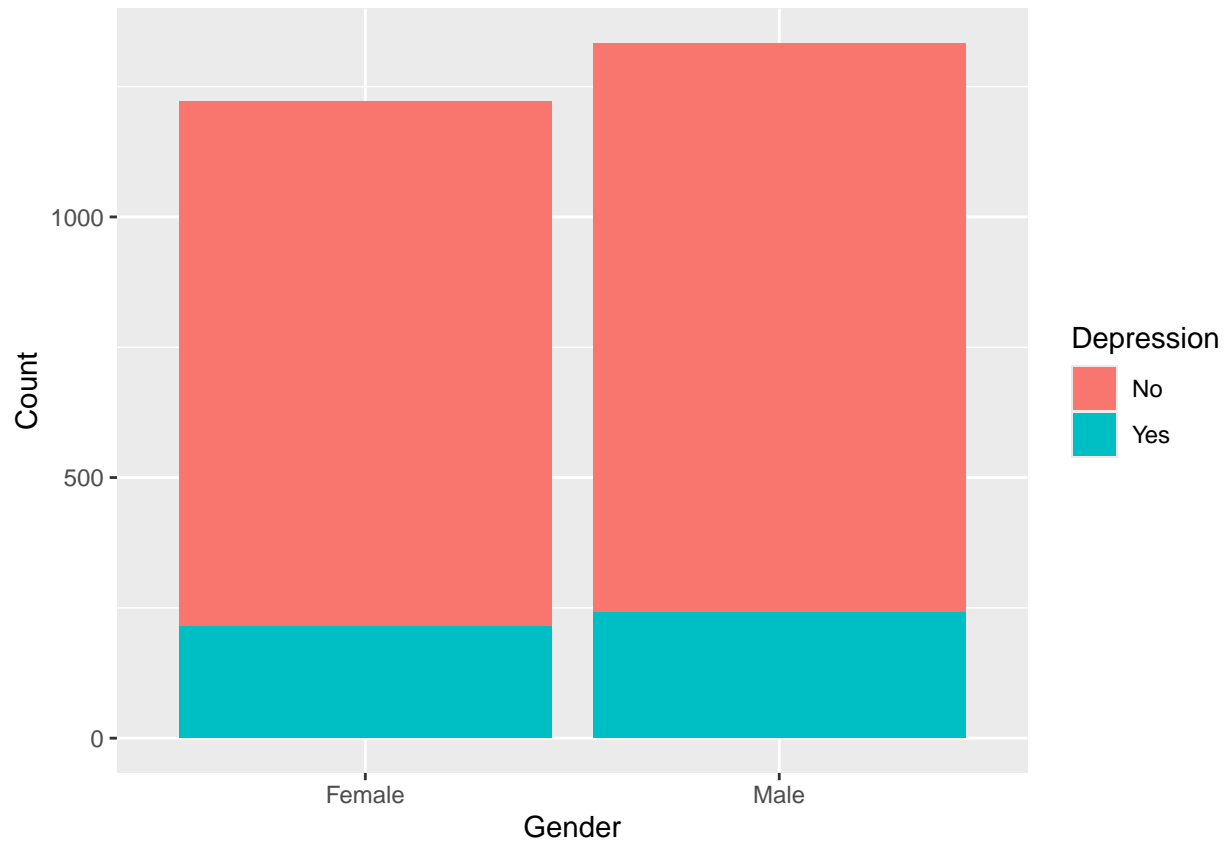
https://www.rapidtables.com/web/color/RGB_Color.html

```
# 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.
```



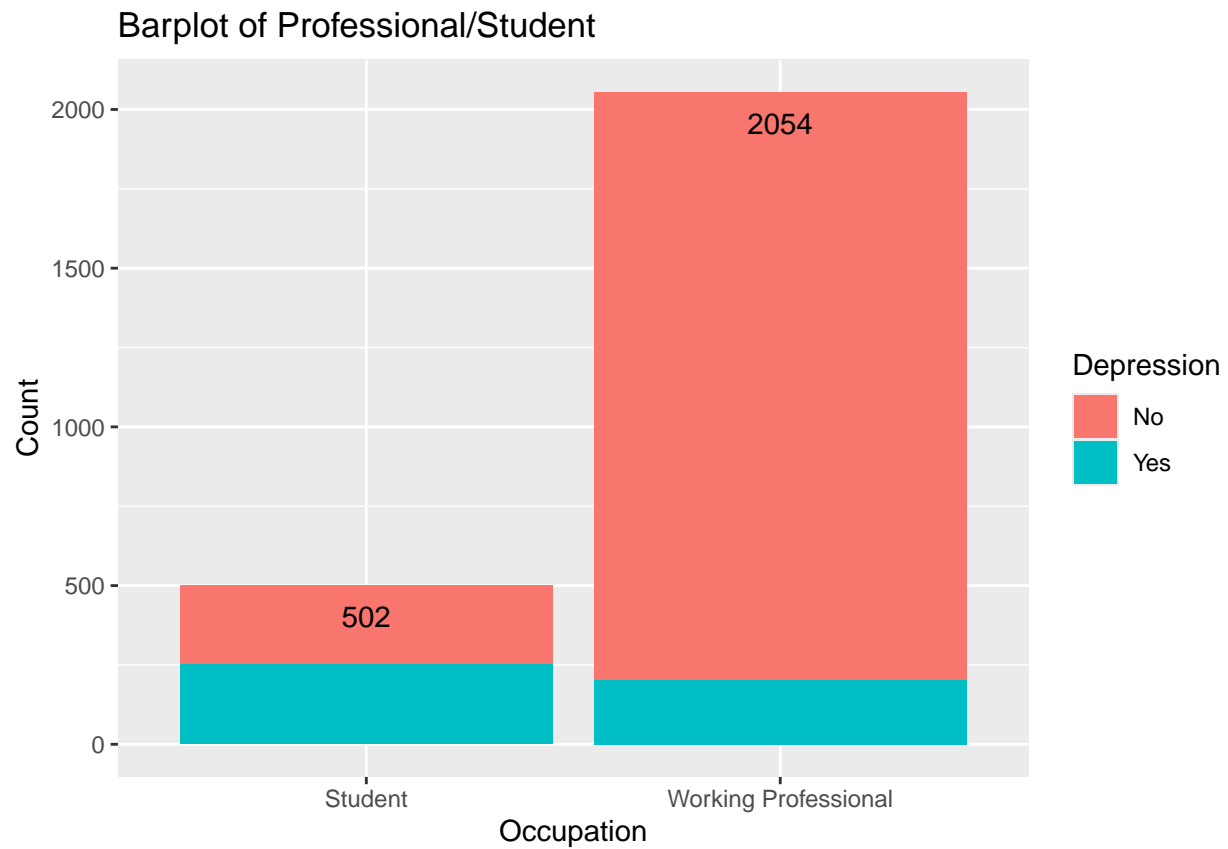
```
# plot gender
ggplot(depression, aes(x = Gender)) +
  geom_bar(aes(fill = Depression)) +
  ylab("Count")
```



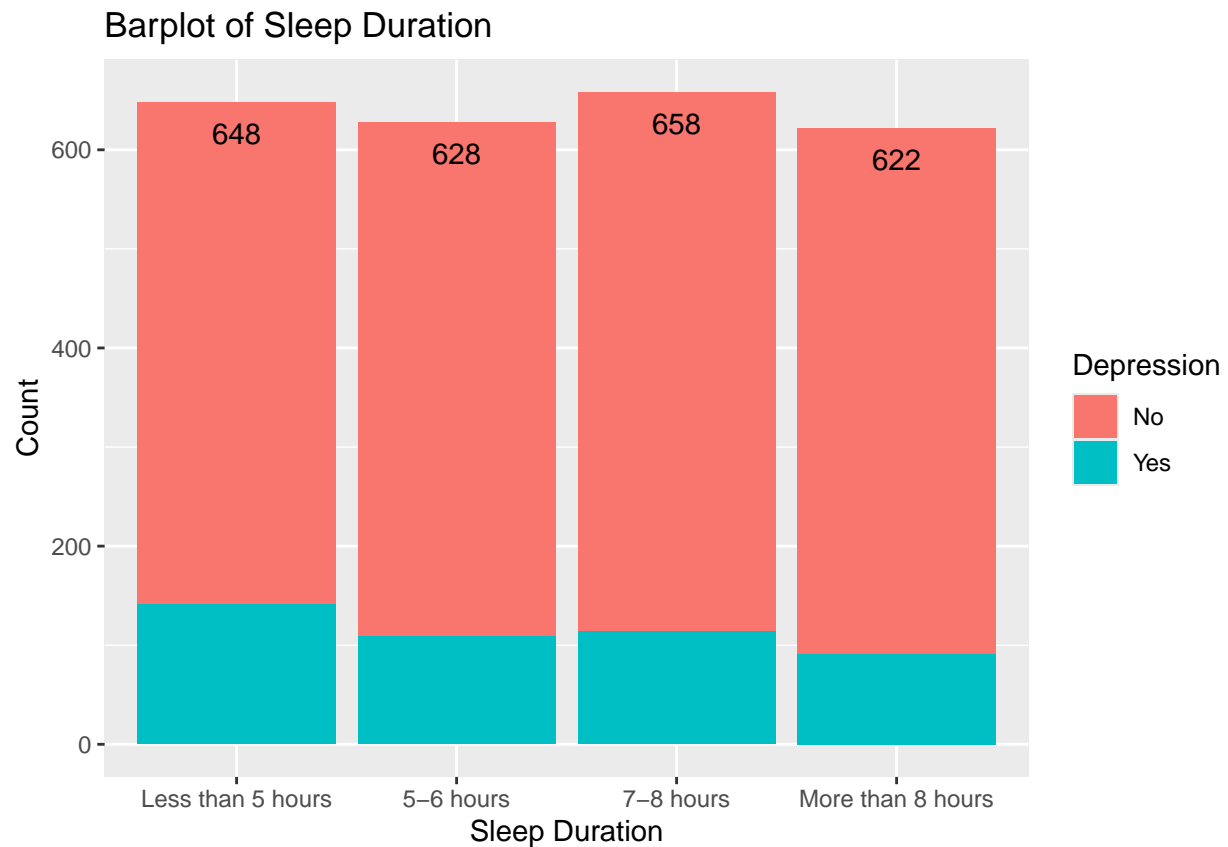
```
ggtitle("Barplot of Gender") +
geom_text(aes(label = ..count..), stat = "count", vjust = 2)
```

NULL

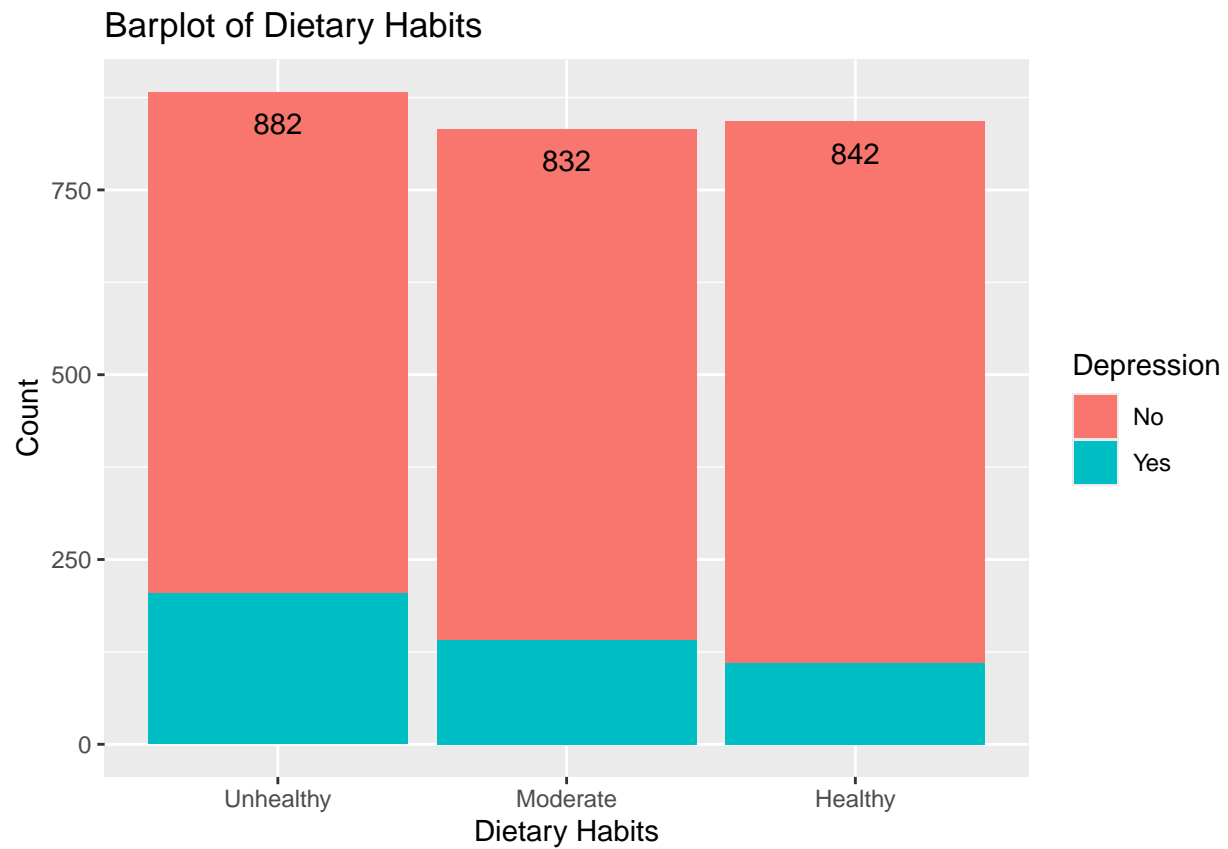
```
# plot whether or not person is a working professional or student
ggplot(depression, aes(x = Working.Professional.or.Student)) +
  geom_bar(aes(fill = Depression)) +
  xlab("Occupation") +
  ylab("Count") +
  ggtitle("Barplot of Professional/Student") +
  geom_text(aes(label = ..count..), stat =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



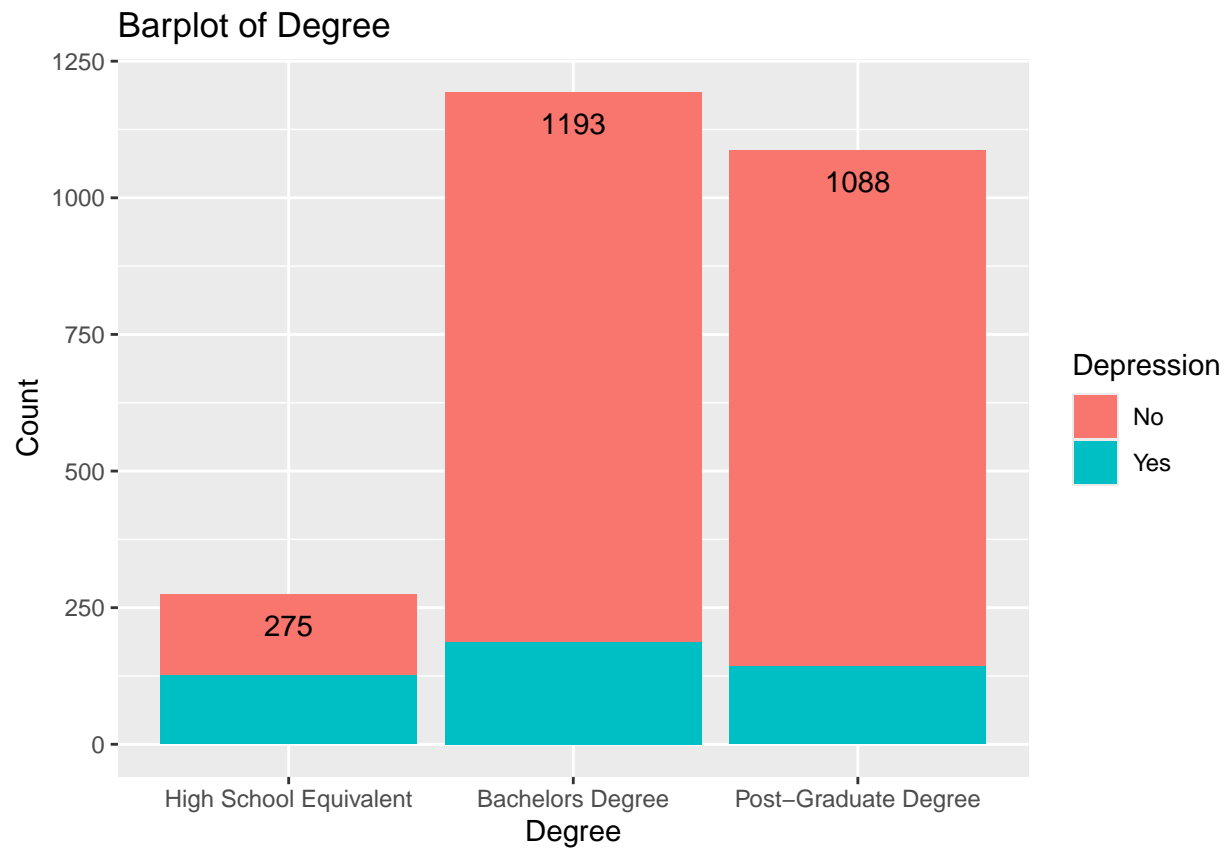
```
# plot sleep duration habits
ggplot(depression, aes(x = Sleep.Duration)) +
  geom_bar(aes(fill = Depression)) +
  xlab("Sleep Duration") +
  ylab("Count") +
  ggtitle("Barplot of Sleep Duration") +
  geom_text(aes(label = ..count..), stat = "count", vjust = 2)
```

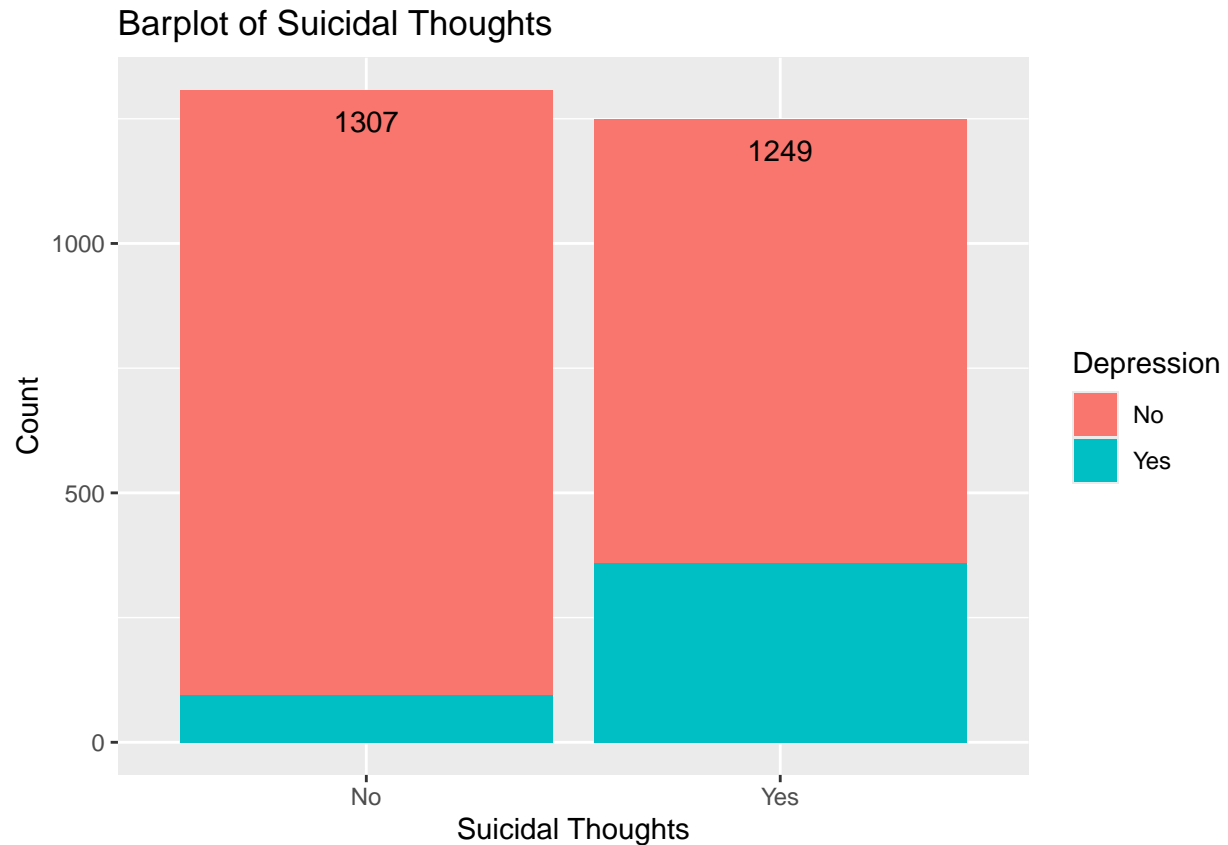
```
# plot dietary habits
ggplot(depression, aes(x = Dietary.Habits)) +
  geom_bar(aes(fill = Depression)) +
  xlab("Dietary Habits") +
  ylab("Count") +
  ggtitle("Barplot of Dietary Habits") +
  geom_text(aes(label = ..count..), stat = "count", vjust = 2)
```



```
# 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)
```



```
# 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)
```

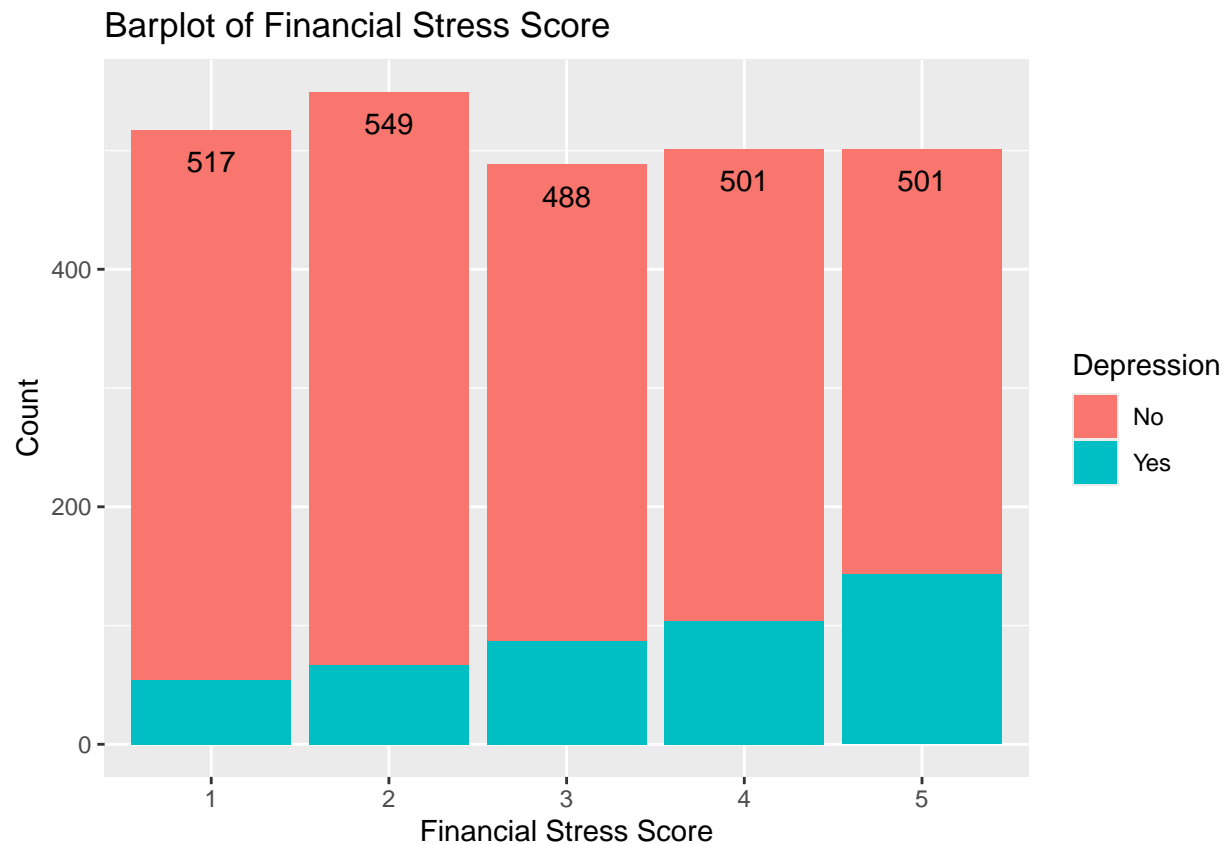


variables look highly correlated, especially when we plot a table of depression and suicidal thoughts
`table(depression$Have.you.ever.had.suicidal.thoughts.., depression$Depression)`

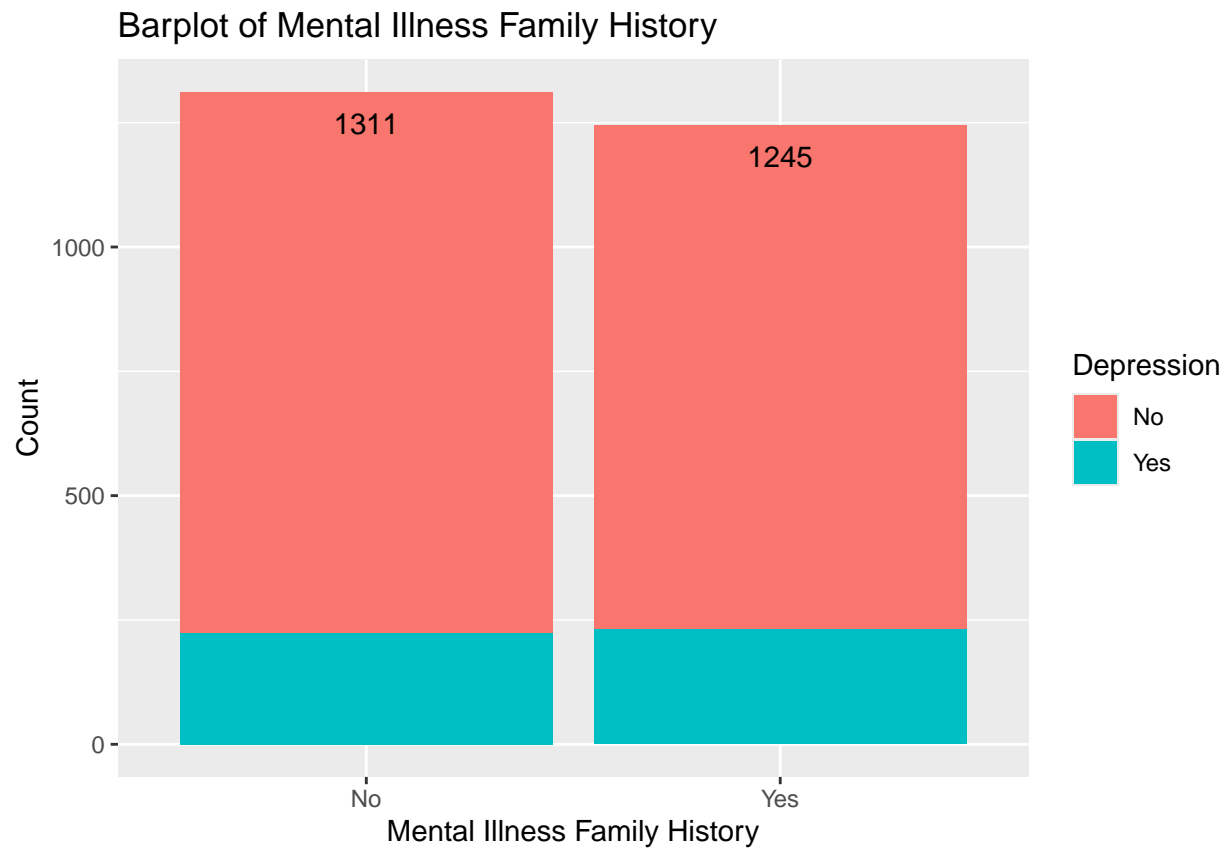
```
##
##           No  Yes
##  No  1212  95
##  Yes   889 360
```

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

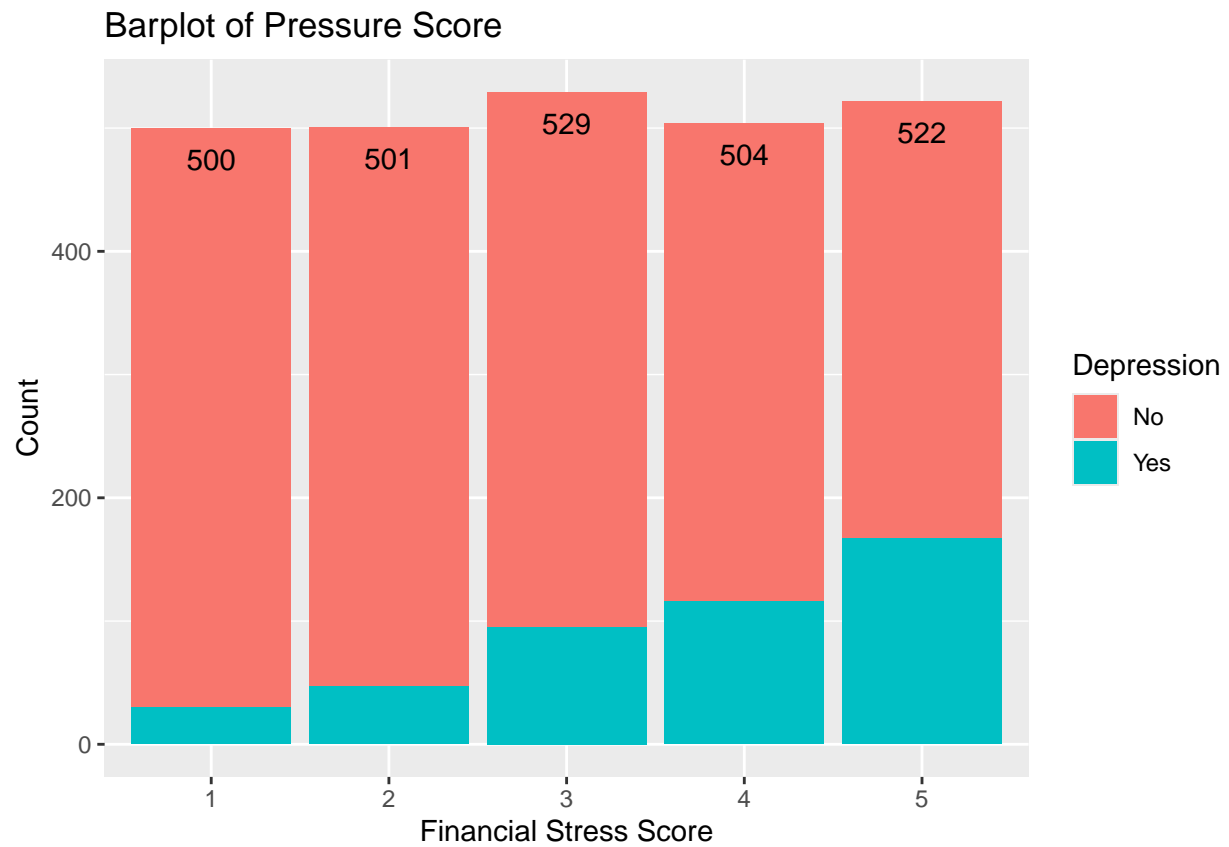
plot financial stress count
`ggplot(depression, aes(x = Financial.Stress)) +
 geom_bar(aes(fill = Depression)) +
 xlab("Financial Stress Score") +
 ylab("Count") +
 ggtitle("Barplot of Financial Stress Score") +
 geom_text(aes(label = ..count..), stat = "count", vjust = 2)`



```
# 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)
```

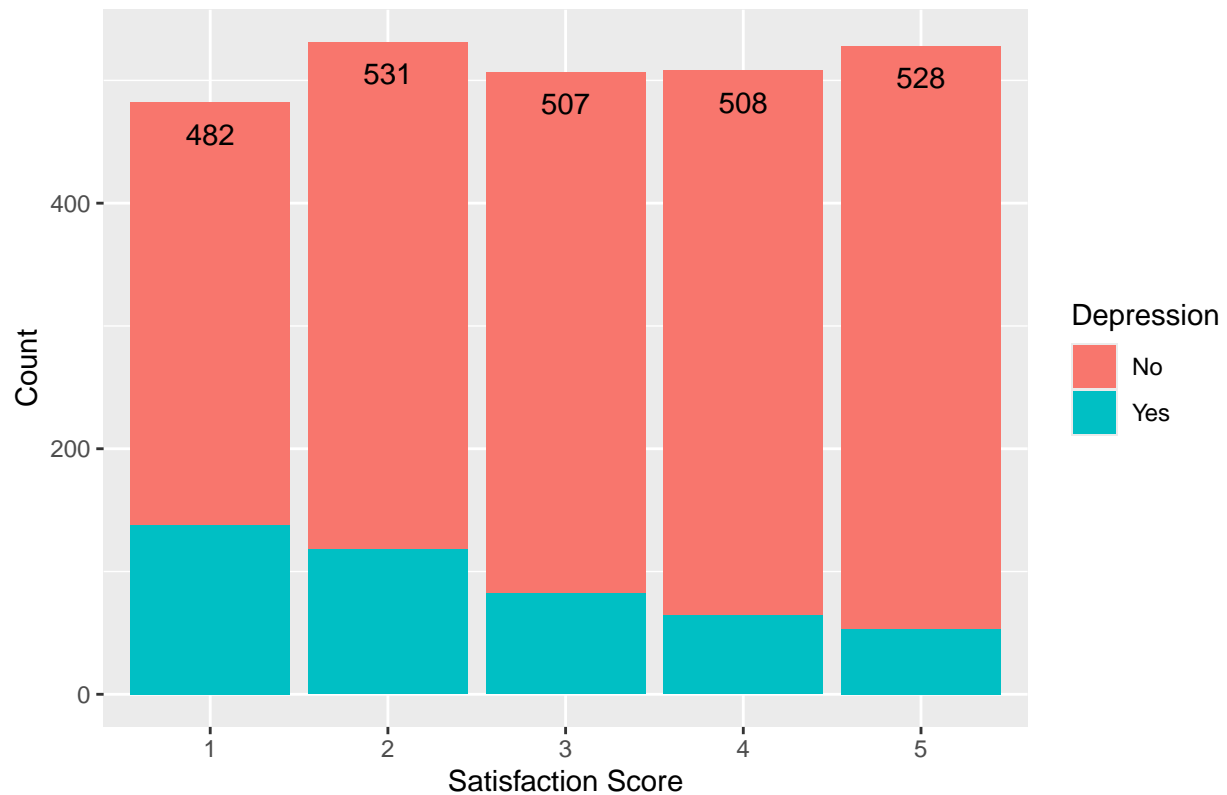


```
ggplot(depression, aes(x = Pressure)) +  
  geom_bar(aes(fill = Depression)) +  
  xlab("Financial Stress Score") +  
  ylab("Count") +  
  ggtitle("Barplot of Pressure Score") +  
  geom_text(aes(label = ..count..), stat = "count", vjust = 2)
```



```
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

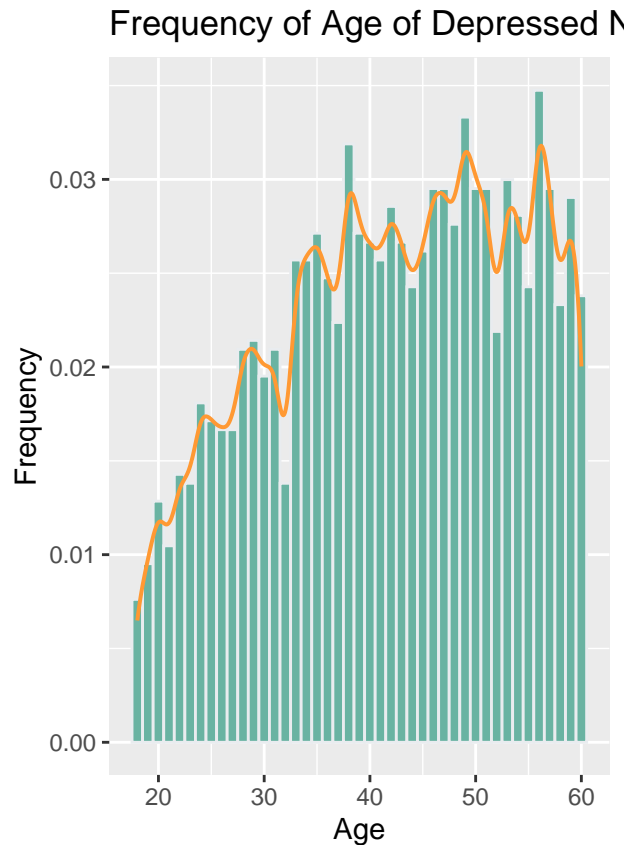
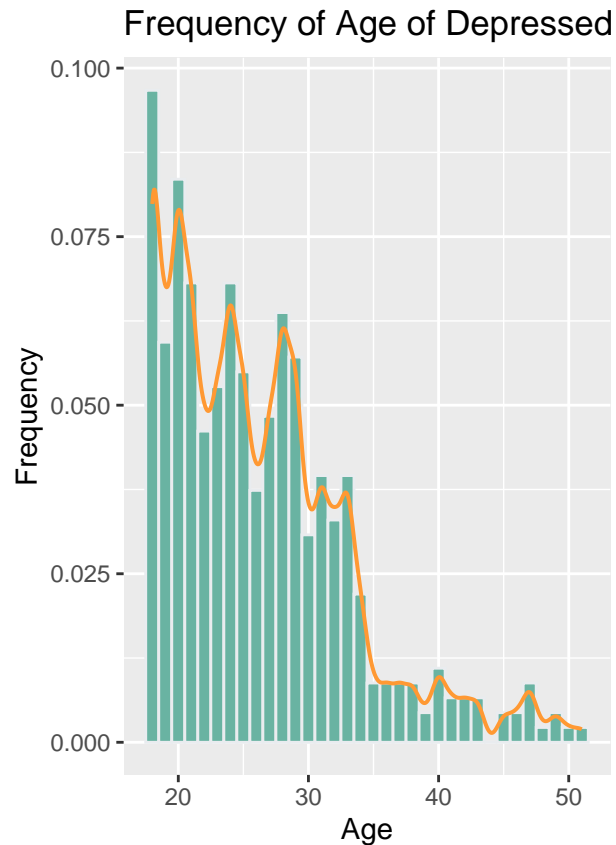


```
depressionYes = depression[depression$Depression == "Yes", ]
depressionNo = depression[depression$Depression == "No", ]

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("Frequency of Age of Depressed Yes") +
  ylab("Frequency")

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("Frequency of Age of Depressed No") +
  ylab("Frequency")

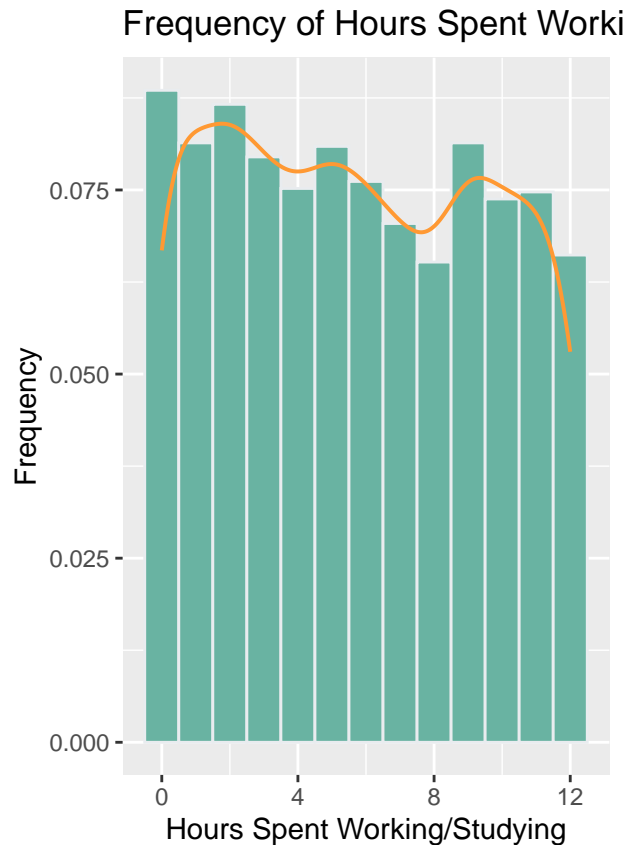
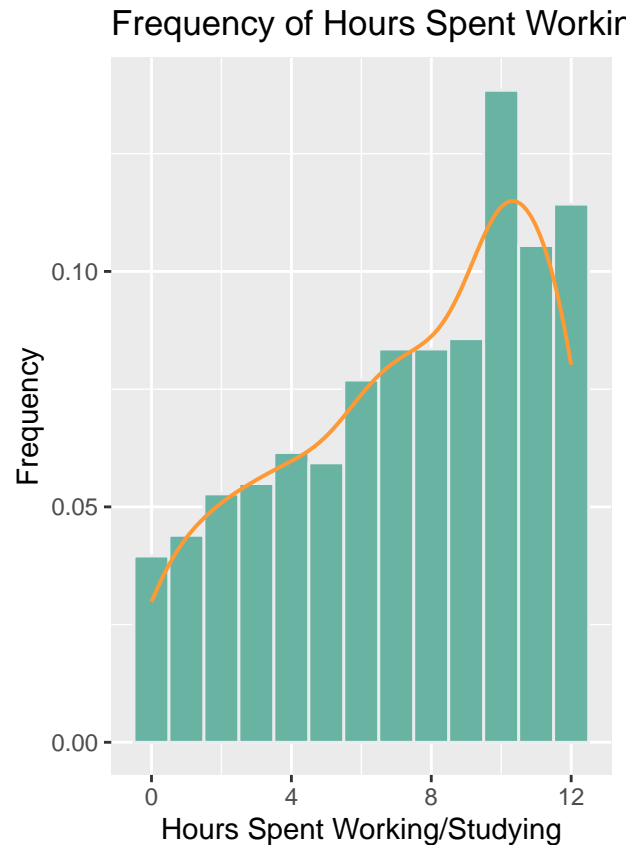
plot_grid(p1, p2)
```

```
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("Frequency of Hours Spent Working/Studying of Depressed Yes") +
  xlab("Hours Spent Working/Studying") +
  ylab("Frequency")

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("Frequency of Hours Spent Working/Studying of Depressed No") +
  xlab("Hours Spent Working/Studying") +
  ylab("Frequency")

plot_grid(p3, p4)
```



```
# 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,]
```

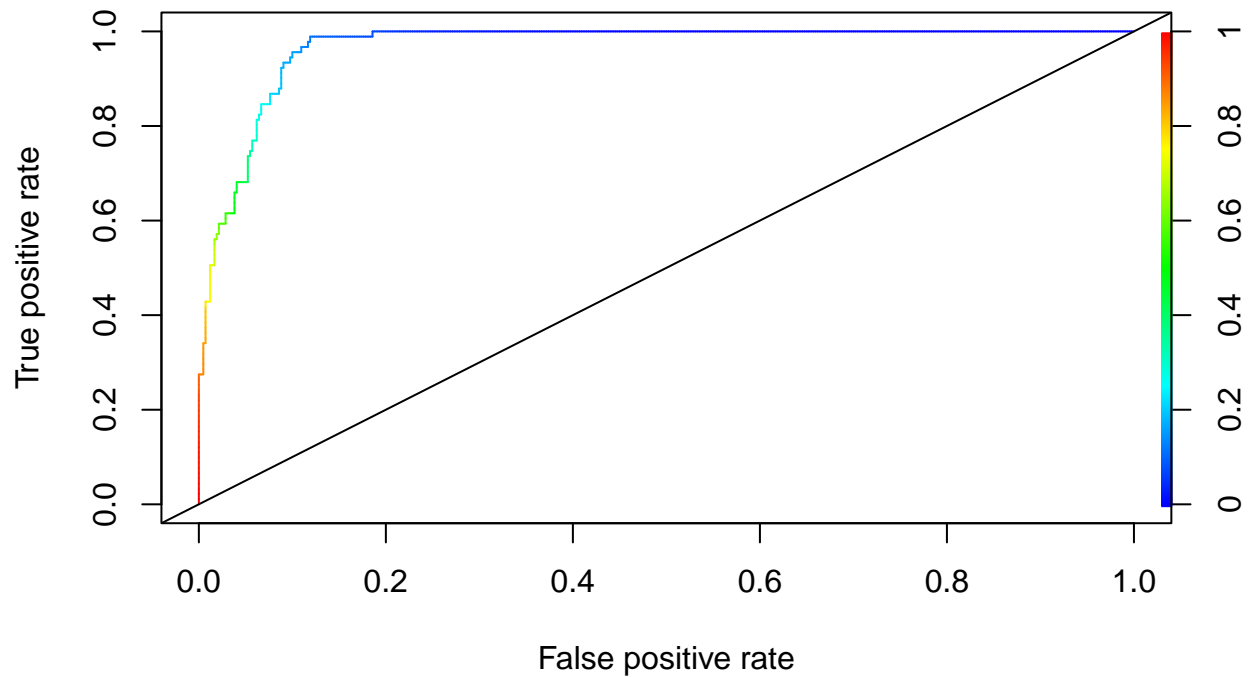
```
# create model with all predictors (no interaction effects)
depression_glm = glm(Depression ~ ., data = depression_train, family = "binomial")
summary(depression_glm)
```

```
##
## Call:
## glm(formula = Depression ~ ., family = "binomial", data = depression_train)
##
## Coefficients:
##
## (Intercept)          3.78330      0.61857    6.116
## GenderMale        -0.13997      0.19537   -0.716
## Age              -0.22579      0.01685  -13.403
## Working.Professional.or.StudentWorking Professional -1.71232      0.22493   -7.613
## Sleep.Duration5-6 hours -0.42781      0.26647   -1.605
## Sleep.Duration7-8 hours -0.92515      0.26493   -3.492
## Sleep.DurationMore than 8 hours -1.32329      0.28303   -4.675
## Dietary.HabitsModerate -0.65864      0.23724   -2.776
## Dietary.HabitsHealthy -1.38810      0.24582   -5.647
## DegreeBachelors Degree -0.35659      0.28808   -1.238
## DegreePost-Graduate Degree -0.39307      0.30470   -1.290
```

```
## Work.Study.Hours          0.24047    0.02836    8.479
## Financial.Stress2         0.49314    0.34183    1.443
## Financial.Stress3         1.35803    0.33647    4.036
## Financial.Stress4         2.02442    0.33801    5.989
## Financial.Stress5         2.69829    0.34286    7.870
## Family.History.of.Mental.IllnessYes 0.71387    0.19667    3.630
## Pressure2                 1.40523    0.39798    3.531
## Pressure3                 2.68474    0.37228    7.212
## Pressure4                 3.65120    0.38559    9.469
## Pressure5                 4.70617    0.40602   11.591
## Satisfaction2            -0.93343    0.28160   -3.315
## Satisfaction3            -1.62497    0.30082   -5.402
## Satisfaction4            -2.94344    0.34103   -8.631
## Satisfaction5            -3.51745    0.36623   -9.604
##                           Pr(>|z|)
## (Intercept)              9.58e-10 ***
## GenderMale                0.473705
## Age                       < 2e-16 ***
## Working.Professional.or.StudentWorking Professional 2.68e-14 ***
## Sleep.Duration5-6 hours   0.108390
## Sleep.Duration7-8 hours   0.000479 ***
## Sleep.DurationMore than 8 hours 2.93e-06 ***
## Dietary.HabitsModerate     0.005498 **
## Dietary.HabitsHealthy      1.64e-08 ***
## DegreeBachelors Degree     0.215785
## DegreePost-Graduate Degree 0.197042
## Work.Study.Hours          < 2e-16 ***
## Financial.Stress2         0.149121
## Financial.Stress3         5.43e-05 ***
## Financial.Stress4         2.11e-09 ***
## Financial.Stress5         3.55e-15 ***
## Family.History.of.Mental.IllnessYes 0.000284 ***
## Pressure2                 0.000414 ***
## Pressure3                 5.53e-13 ***
## Pressure4                 < 2e-16 ***
## Pressure5                 < 2e-16 ***
## Satisfaction2            0.000917 ***
## Satisfaction3            6.60e-08 ***
## Satisfaction4             < 2e-16 ***
## Satisfaction5            < 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:  717.15  on 2020  degrees of freedom
## AIC: 767.15
##
## Number of Fisher Scoring iterations: 8
```

```
# 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)
```

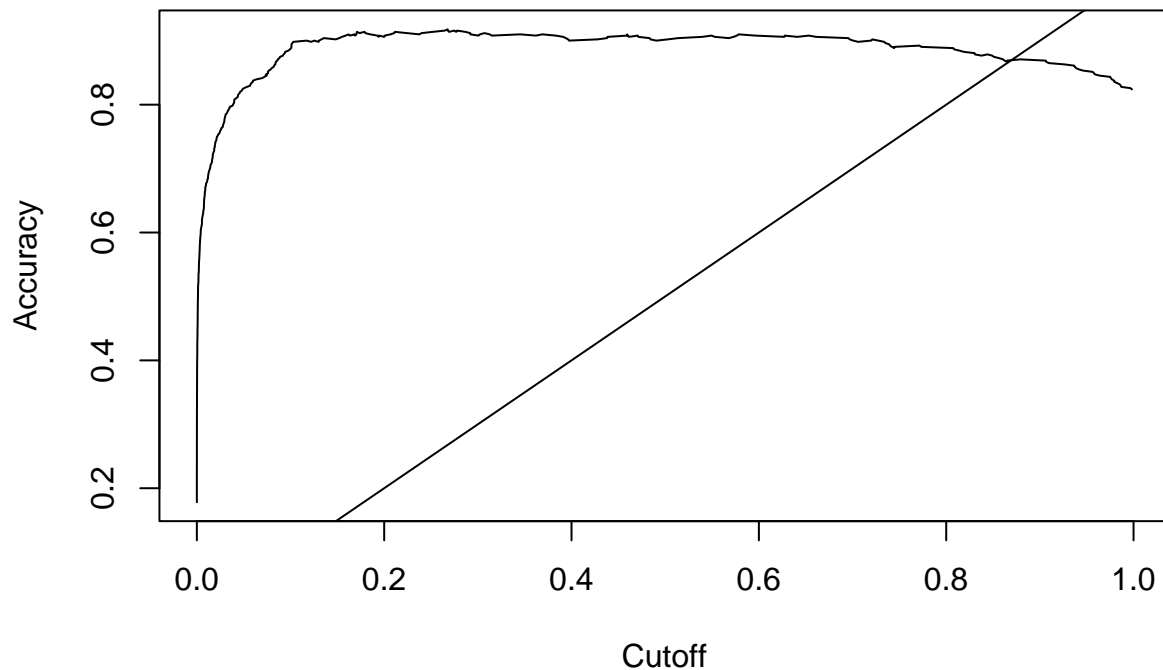
```
roc_curve = performance(prob_predictions, "tpr", "fpr")
plot(roc_curve, colorize = TRUE)
abline(0, 1)
```



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

```
## [1] 0.9682365
```

```
acc = performance(prob_predictions, "acc")
plot(acc)
abline(0, 1)
```



```
glm_predictions2 = predict(depression_glm, newdata = depression_test)
glm_predictions2 = ifelse(glm_predictions2 > 0.30, "Yes", "No")
glm_predictions2 = as.factor(glm_predictions2)
confusionMatrix(glm_predictions2, depression_test$Depression)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No  Yes
##      No  410  37
##      Yes   10  54
##
##           Accuracy : 0.908
##           95% CI : (0.8796, 0.9316)
##      No Information Rate : 0.8219
##      P-Value [Acc > NIR] : 2.887e-08
##
##           Kappa : 0.6445
##
##  Mcnemar's Test P-Value : 0.0001491
##
##           Sensitivity : 0.9762
##           Specificity : 0.5934
##      Pos Pred Value : 0.9172
##      Neg Pred Value : 0.8437
##           Prevalence : 0.8219
```

```
##          Detection Rate : 0.8023
##    Detection Prevalence : 0.8748
##          Balanced Accuracy : 0.7848
##
##          '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.

```
vI = cbind(varImp(depression_glm), summary(depression_glm)$coefficients[-1, c(1, 4)])
vI[order(-vI$Overall), , drop = FALSE]
```

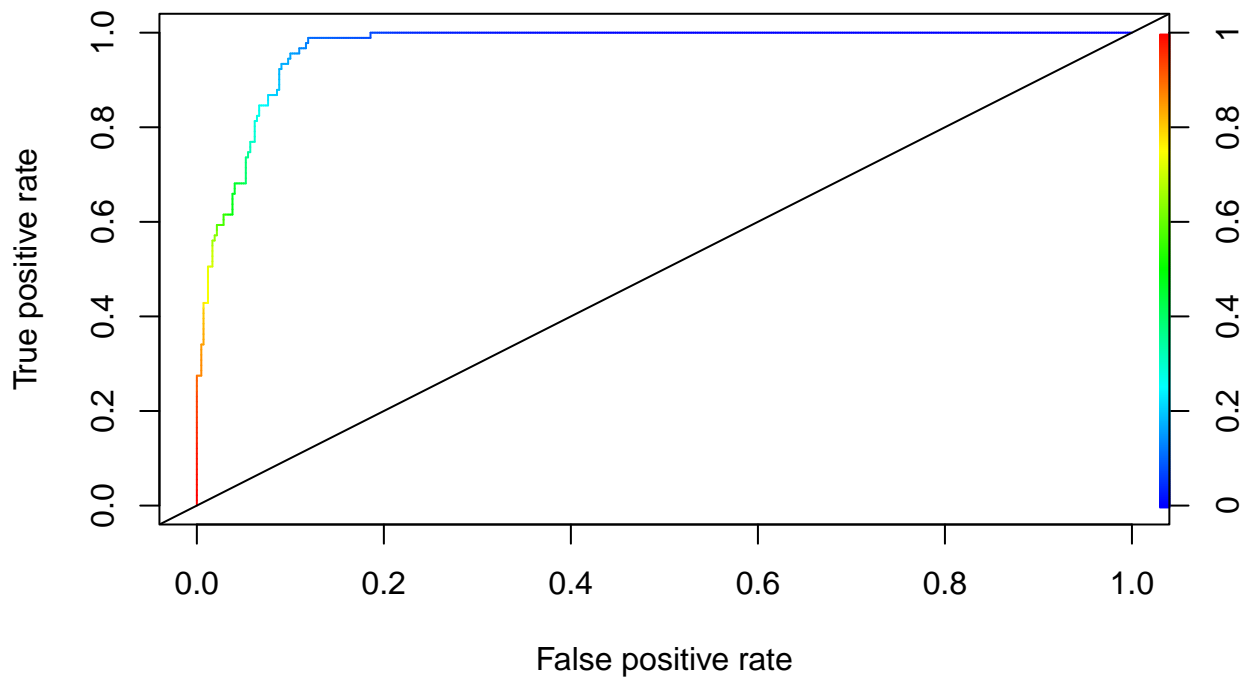
	Overall	Estimate
## Age	13.4032252	-0.2257893
## Pressure5	11.5908627	4.7061676
## Satisfaction5	9.6044377	-3.5174483
## Pressure4	9.4691662	3.6512049
## Satisfaction4	8.6310127	-2.9434364
## Work.Study.Hours	8.4793681	0.2404746
## Financial.Stress5	7.8699917	2.6982869
## Working.Professional.or.StudentWorking Professional	7.6127723	-1.7123213
## Pressure3	7.2116333	2.6847351
## Financial.Stress4	5.9892963	2.0244248
## Dietary.HabitsHealthy	5.6467516	-1.3880973
## Satisfaction3	5.4017959	-1.6249677
## Sleep.DurationMore than 8 hours	4.6754764	-1.3232888
## Financial.Stress3	4.0361694	1.3580299
## Family.History.of.Mental.IllnessYes	3.6297969	0.7138718
## Pressure2	3.5309243	1.4052300
## Sleep.Duration7-8 hours	3.4920284	-0.9251458
## Satisfaction2	3.3147020	-0.9334275
## Dietary.HabitsModerate	2.7763090	-0.6586400
## Sleep.Duration5-6 hours	1.6054713	-0.4278123
## Financial.Stress2	1.4426423	0.4931382
## DegreePost-Graduate Degree	1.2900262	-0.3930683
## DegreeBachelors Degree	1.2378154	-0.3565854

```
## GenderMale 0.7164644 -0.1399742
## Pr(>|z|)
## Age 5.789167e-41
## Pressure5 4.584912e-31
## Satisfaction5 7.657472e-22
## Pressure4 2.820862e-21
## Satisfaction4 6.081113e-18
## Work.Study.Hours 2.264203e-17
## Financial.Stress5 3.546647e-15
## Working.Professional.or.StudentWorking Professional 2.682782e-14
## Pressure3 5.528469e-13
## Financial.Stress4 2.107509e-09
## Dietary.HabitsHealthy 1.635078e-08
## Satisfaction3 6.597701e-08
## Sleep.DurationMore than 8 hours 2.932720e-06
## Financial.Stress3 5.433099e-05
## Family.History.of.Mental.IllnessYes 2.836443e-04
## Pressure2 4.141102e-04
## Sleep.Duration7-8 hours 4.793672e-04
## Satisfaction2 9.174088e-04
## Dietary.HabitsModerate 5.497992e-03
## Sleep.Duration5-6 hours 1.083901e-01
## Financial.Stress2 1.491213e-01
## DegreePost-Graduate Degree 1.970416e-01
## DegreeBachelors Degree 2.157845e-01
## GenderMale 4.737046e-01
```

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

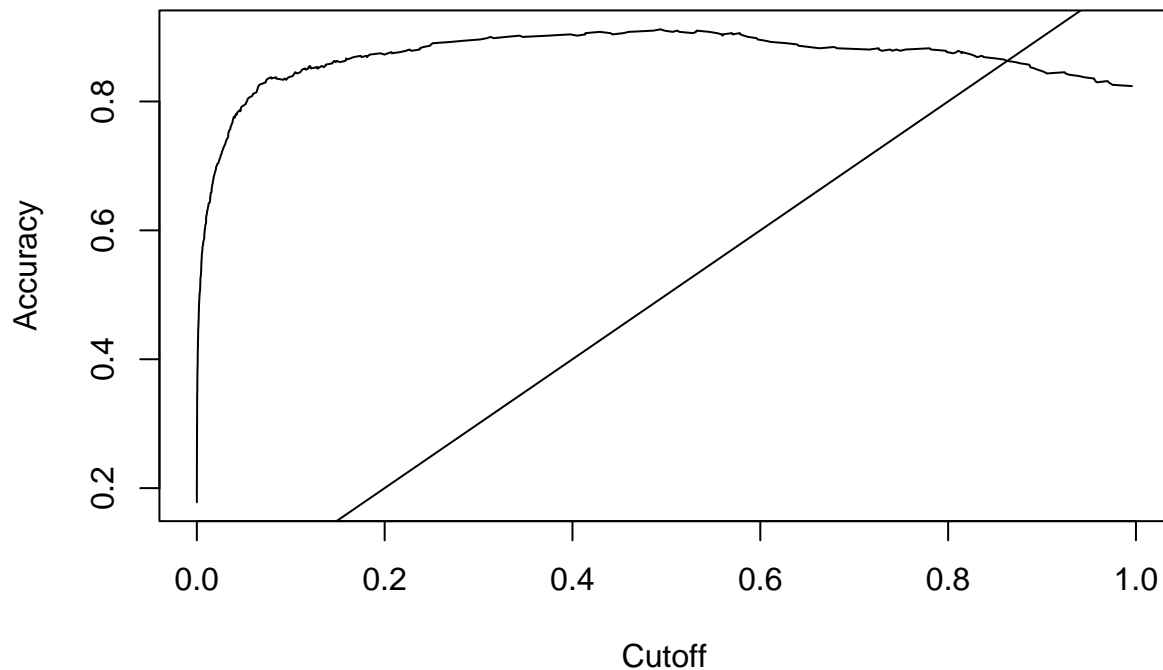
```
##
## Call:
## glm(formula = Depression ~ Age + Pressure + Satisfaction + Work.Study.Hours +
## Financial.Stress, family = "binomial", data = depression_train)
##
## Coefficients:
## Estimate Std. Error z value Pr(>|z|)
## (Intercept) 2.39596 0.50789 4.717 2.39e-06 ***
## Age -0.23613 0.01392 -16.966 < 2e-16 ***
## Pressure2 1.17217 0.36389 3.221 0.001277 **
## Pressure3 2.29104 0.33644 6.810 9.78e-12 ***
## Pressure4 3.18648 0.34283 9.295 < 2e-16 ***
## Pressure5 4.02149 0.35325 11.384 < 2e-16 ***
## Satisfaction2 -0.83502 0.25705 -3.248 0.001160 **
## Satisfaction3 -1.42610 0.27544 -5.177 2.25e-07 ***
## Satisfaction4 -2.35072 0.29667 -7.924 2.31e-15 ***
## Satisfaction5 -2.94424 0.31694 -9.289 < 2e-16 ***
## Work.Study.Hours 0.21885 0.02569 8.520 < 2e-16 ***
## Financial.Stress2 0.21229 0.30617 0.693 0.488086
## Financial.Stress3 1.09724 0.29704 3.694 0.000221 ***
## Financial.Stress4 1.56518 0.29591 5.289 1.23e-07 ***
## Financial.Stress5 2.26508 0.29631 7.644 2.10e-14 ***
## ---
## 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: 846.31 on 2030 degrees of freedom
## AIC: 876.31
##
## 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_predictions, "tpr", "fpr")
plot(roc_curve2, colorize = TRUE)
abline(0, 1)
```



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

## [1] 0.946494
acc2 = performance(prob_predictions2, "acc")
plot(acc2)
abline(0, 1)
```

```
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  39
##      Yes   12  52
##
##           Accuracy : 0.9002
##           95% CI : (0.8709, 0.9248)
##      No Information Rate : 0.8219
##      P-Value [Acc > NIR] : 5.264e-07
##
##           Kappa : 0.6142
##
##      Mcnemar's Test P-Value : 0.0002719
##
##           Sensitivity : 0.9714
##           Specificity : 0.5714
##      Pos Pred Value : 0.9128
##      Neg Pred Value : 0.8125
##           Prevalence : 0.8219
```

```

##          Detection Rate : 0.7984
##    Detection Prevalence : 0.8748
##      Balanced Accuracy : 0.7714
##
##      'Positive' Class : No
##

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

## [1] "First Model Residual Deviance: 717.146403740301"

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

## [1] "Second Model Residual Deviance: 846.31369830212"

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 0.8968226 0.6369108 0.02223823 0.07530717

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

## Confusion Matrix and Statistics
##
##              Reference
## Prediction  No Yes
##      No  405  35
##      Yes   15  56
##
##              Accuracy : 0.9022
##              95% CI : (0.873, 0.9265)
##      No Information Rate : 0.8219
##      P-Value [Acc > NIR] : 2.632e-07
##
##              Kappa : 0.6343
##
##      McNemar's Test P-Value : 0.00721
##
##              Sensitivity : 0.9643
##              Specificity : 0.6154
##      Pos Pred Value : 0.9205
##      Neg Pred Value : 0.7887
##      Prevalence : 0.8219
##      Detection Rate : 0.7926
##      Detection Prevalence : 0.8611
##      Balanced Accuracy : 0.7898
##
##      'Positive' Class : No
##

```

```
varImp(depression_cvglm)
```

```
## glm variable importance
##
##    only 20 most important variables shown (out of 24)
##
##                                     Overall
## Age                               100.000
## Pressure5                         85.715
## Satisfaction5                     70.057
## Pressure4                         68.991
## Satisfaction4                     62.384
## Work.Study.Hours                  61.189
## Financial.Stress5                 56.386
## `Working.Professional.or.StudentWorking Professional` 54.358
## Pressure3                         51.196
## Financial.Stress4                 41.562
## Dietary.HabitsHealthy              38.862
## Satisfaction3                     36.931
## `Sleep.DurationMore than 8 hours` 31.206
## Financial.Stress3                 26.167
## Family.History.of.Mental.IllnessYes 22.964
## Pressure2                         22.184
## `Sleep.Duration7-8 hours`         21.878
## Satisfaction2                     20.480
## Dietary.HabitsModerate             16.236
## `Sleep.Duration5-6 hours`         7.007
```

```
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 0.895516 0.6248041 0.02400873 0.086268
```

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

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

```
##
##           Reference
## Prediction  No  Yes
##           No 402  29
##           Yes  18  62
##
##           Accuracy : 0.908
##           95% CI : (0.8796, 0.9316)
##           No Information Rate : 0.8219
##           P-Value [Acc > NIR] : 2.887e-08
##
```

```
##           Kappa : 0.6702
##
## McNemar's Test P-Value : 0.1447
##
##           Sensitivity : 0.9571
##           Specificity : 0.6813
##           Pos Pred Value : 0.9327
##           Neg Pred Value : 0.7750
##           Prevalence : 0.8219
##           Detection Rate : 0.7867
##           Detection Prevalence : 0.8434
##           Balanced Accuracy : 0.8192
##
##           'Positive' Class : No
##
```

```
varImp(depression_cvglm2)
```

```
## glm variable importance
##
##           Overall
## Age           100.00
## Pressure5      65.70
## Pressure4      52.86
## Satisfaction5  52.83
## Work.Study.Hours 48.10
## Satisfaction4  44.43
## Financial.Stress5 42.72
## Pressure3      37.59
## Financial.Stress4 28.24
## Satisfaction3  27.56
## Financial.Stress3 18.44
## Satisfaction2  15.70
## Pressure2      15.53
## Financial.Stress2 0.00
```