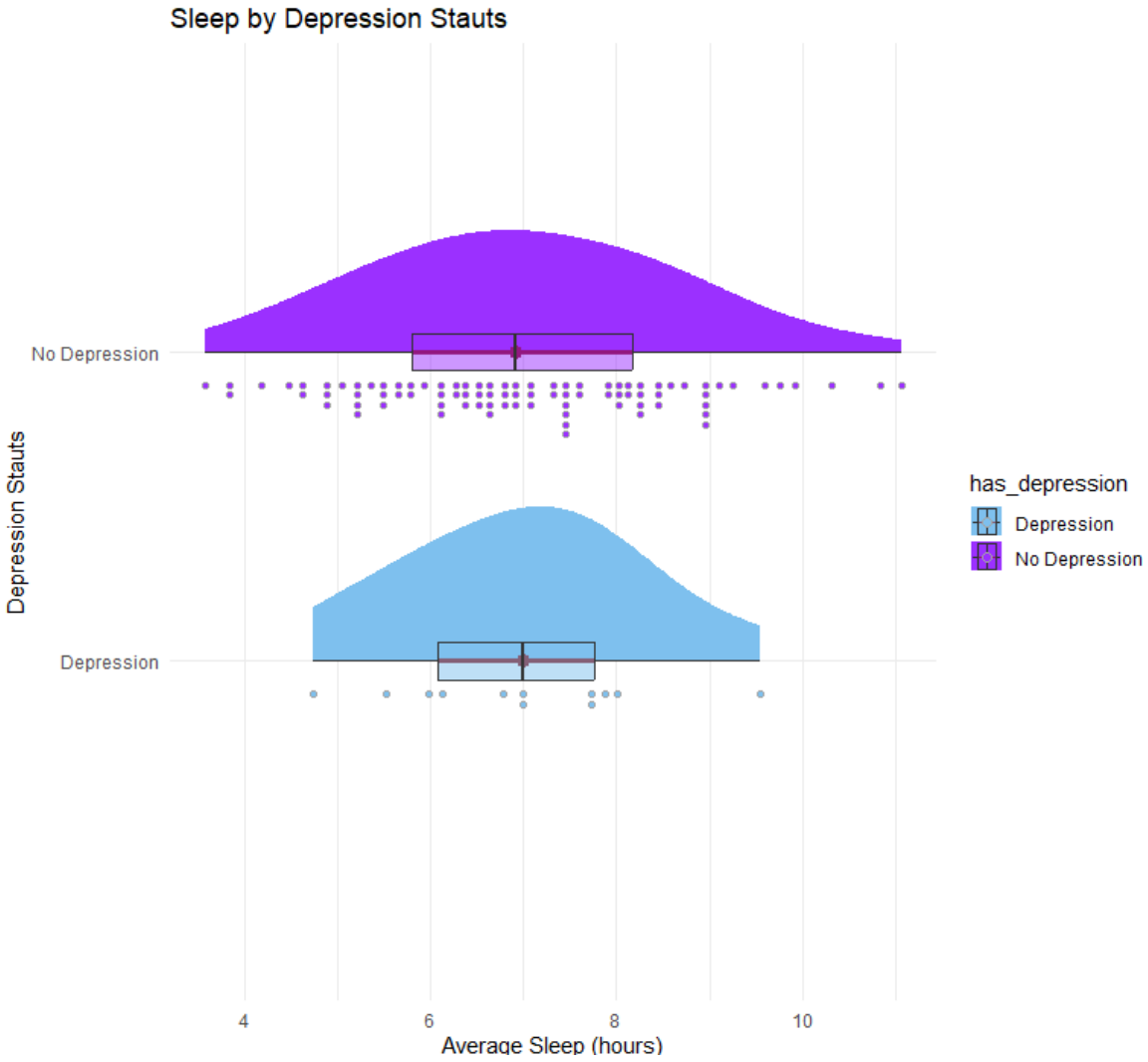
```r
# R course for beginners
# Week 4 part 1
# assignment by Amir Mano, id 205779788

#### prepare workspace ----
rm(list = ls()) #or Ctrl + Shift + F10 & Ctrl + L

# import packages
library(tidyverse)

#### create data ----

# creating variables
N = 100
sub_id <- seq(1,N)
sex <- factor(
  sample(c('m','f'), N, replace=T, prob = c(0.5, 0.5)),
  levels = c('m', 'f'),
  labels = c("Male", "Female")
)
age <- runif(N, 15, 40)
iq <- rnorm(N, 100, 15)
has_depression <- factor(
  rbinom(N,1,0.083),
  levels = c(0, 1),
  labels = c("No Depression", "Depression")
  )
sleep <- rnorm(N, 7, 1.5)
df <- data.frame(sub_id, sex, age, iq, has_depression, sleep)

# adding IQ to data frame
write.csv(df, './HW_4.csv', row.names = FALSE)
```

```r
# R course for beginners
# week 4 part 2
# assignment by Amir Mano, id 205779788

#### prepare workspace ----
rm(list = ls()) #or Ctrl + Shift + F10 & Ctrl + L

# import packages
library(tidyverse)
library(ggplot2)
library(ggdist)

# load variables
df <- read.csv('HW_4.csv')
df <- df|> mutate(sex=factor(sex))
df <- df|> mutate(has_depression=factor(has_depression))

#### descriptive statistics ----
# extracting counts
gender_counts <- sum(df |> pull(sex) |> as.numeric() == 1)
depression_count <- sum(df |> pull(has_depression) |> as.numeric() == 1)

# printing stats
cat('Number of subjects:', nrow(df), '\n',
    'Number by gender:', levels(df$sex)[1], gender_counts,
    ',', levels(df$sex)[2], nrow(df)-gender_counts,'\n',
    'Range of ages:', df |> pull(age) |> min() |> floor(),
    '-', df |> pull(age) |> max() |> ceiling(), '\n',
    'Average IQ:', df |> pull(iq) |> mean(), ' (', df |> pull(iq) |> sd(), ')\n',
    'Average sleep:', df |> pull(sleep) |> mean(), ' (', df |> pull(sleep) |> sd(), ')\n',
    'Median percentage of depression:', depression_count/nrow(df)*100, '%', '\n')

#### plotting ----
ggplot(df, aes(x=sleep, y=has_depression, fill=has_depression)) +
  stat_halfeye(.width=c(0.9, 0.5), color='darkred',scale=0.5) +
  geom_boxplot(
    width = 0.12,
    outlier.color = NA,
    alpha = 0.5
  ) +
  stat_dots(
    side = "left",
    justification = 1.1,
    binwidth = 0.1
  ) +
  labs(
    title = 'Sleep by Depression Stauts',
    y = 'Depression Stauts',
    x = 'Average Sleep (hours)'
  ) +
  scale_y_discrete(expand = expansion(mult = c(0.5, 0.5))) +
  scale_fill_manual(values = c('skyblue2', 'purple1')) +
  theme_minimal()
```


Sleep by Depression Stauts
No Depression
Depression
Depression Stauts
4
6
8
10
Average Sleep (hours)
has_depression
Depression
No Depression

```r
# R course for beginners
# Week 4 part 3
# assignment by Amir Mano, id 205779788

#### prepare workspace ----
rm(list = ls()) #or Ctrl + Shift + F10 & Ctrl + L

# import packages
library(tidyverse)
library(ggplot2)
library(ggdist)
library(effectsize)

# load variables
df <- read.csv('HW_4.csv')
df <- df|> mutate(sex=factor(sex))
df <- df|> mutate(has_depression=factor(has_depression))


#### test to check difference from 3 ----
t_test_result <- t.test(
  df$sleep,
  alternative = "two.sided",
  mu = 3,
  conf.level = 0.95
)
print(
  paste(
    'p-value:', t_test_result$p.value,
    ', with CI:', round(t_test_result$conf.int[1], 3),
    '-', round(t_test_result$conf.int[2], 3)
  )
)

#### test to check difference by sleeping time ----
t_test_result <- t.test(data = df,
  sleep ~ has_depression,
  alternative = "two.sided",
  conf.level = 0.95
)
print(
  paste(
    'p-value:', t_test_result$p.value,
    ', with CI:', round(t_test_result$conf.int[1], 3),
    '-', round(t_test_result$conf.int[2], 3)
  )
)

# calculating Cohen's d
d_automated <- cohens_d(
  df$sleep[df$has_depression=='Depression'],
  df$sleep[df$has_depression!='Depression']
)
mask <- df$has_depression==levels(df$has_depression)[1]
n <- c(
  length(df$has_depression[mask]),
  length(df$has_depression[!mask])
  )
avg <-c(
  mean(df$sleep[mask]),
  mean(df$sleep[!mask])
)
s <-c(
  sd(df$sleep[mask]),
  sd(df$sleep[!mask])
)
sd_pooled <- sqrt(((n[1]-1)*s[1]^2+(n[2]-1)*s[2]^2)/(n[1]+n[2]-2))
d_manual <- (avg[1]-avg[2])/sd_pooled

cat(
    "cohen's d automated:", d_automated$Cohens_d,
    "\ncohen's d manual:", d_manual)

#### fit to linear model ----
model <- lm(data = df,
            sleep ~ has_depression)
print(model$coef)
```