Analyzing Sleep Health and Lifestyle: Key Insights and Challenges

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

Introduction: The Sleep Health and Lifestyle dataset provides valuable information on sleep patterns, lifestyle habits,

and cardiovascular health across 400 individuals. This dataset includes a wide range of variables such as sleep duration,

quality, physical activity, stress levels, BMI, and the presence of sleep disorders. While the data provides insights

into how lifestyle factors influence sleep, challenges such as the subjective nature of some metrics

(e.g., stress levels, sleep quality), and missing data could impact the accuracy of any conclusions drawn from the dataset.

The goal of this project is to explore these challenges and use the dataset to uncover meaningful correlations between lifestyle factors and sleep health, as well as predict potential sleep disorders.

Reference: https://www.kaggle.com/datasets/uom190346a/sleep-health-and-lifestyle-dataset

Here are five key questions we aim to answer using the Sleep Health and Lifestyle dataset:

1. How do physical activity levels correlate with the quality of sleep?

2. Does stress level impact sleep duration?

3. Can we predict someone's BMI category based on their sleep duration and physical activity level?

4. How does age impact sleep duration and quality?

5. What is the relationship between BMI category and stress levels?

```r
# Importing libraries
library(shiny)
library(ggplot2)
library(plotly)
```

```
##
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':
##
##     last_plot

## The following object is masked from 'package:stats':
##
##     filter

## The following object is masked from 'package:graphics':
##
##     layout
```

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(shinythemes)
library(caret)
```

```
## Loading required package: lattice
```

```r
library(ROSE)
```

```
## Loaded ROSE 0.0-4
```

```r
library(rpart)
library(shinydashboard)
```

```
##
## Attaching package: 'shinydashboard'

## The following object is masked from 'package:graphics':
##
##     box
```

```r
library(reshape2)
```

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

```r
# Load the Dataset
dataset <- read.csv("C:/Users/elena/OneDrive/Documents/Semester-4/Rproj/sleepv2/sleep_dataset.csv")
```

```r
# Rename columns
colnames(dataset) <- c("PersonID", "Gender", "Age", "Occupation", "SleepDuration",
                       "QualityOfSleep", "PhysicalActivity", "StressLevel",
                       "BMICategory", "BloodPressure", "HeartRate", "DailySteps", "SleepDisorder")
```

```r
# Check for missing values and remove them if any
if (sum(is.na(dataset)) > 0) {
  dataset <- na.omit(dataset)
}
```

Data preparation

```r
# Ensure BMICategory is a factor for classification tasks
dataset$BMICategory <- as.factor(dataset$BMICategory)
```

```r
# Create a binary BMI category for binary classification
dataset$BMICategoryBinary <- ifelse(dataset$BMICategory == "Normal", "Normal", "Non-Normal")
dataset$BMICategoryBinary <- as.factor(dataset$BMICategoryBinary)
```

Demonstrate the data creation process used in the solution initiating the following a step-bystep process of the following data structures.

```r
# Create a vector (e.g., extracting sleep duration):
sleep_duration <- dataset$Sleep.Duration
```

```r
# Create a list (e.g., combining sleep data and age):
sleep_list <- list(sleep_duration = dataset$Sleep.Duration, age = dataset$Age)
```

```r
# Create an array (e.g., for stress levels and physical activity levels):
stress_array <- array(dataset$StressLevel, dim = c(10, 10))
```

```r
# Create a matrix (e.g., age and physical activity):
act_matrix <- matrix(c(dataset$Age[1:10], dataset$PhysicalActivity[1:10]), nrow = 10, ncol = 2)
```

MACHINE LEARNING

```r
# Split the data into training and testing sets based on BMICategory
set.seed(123)
trainIndex <- createDataPartition(dataset$BMICategory, p = 0.7, list = FALSE)
trainData <- dataset[trainIndex, ]
testData <- dataset[-trainIndex, ]
```

```r
# Oversample the minority classes
balanced_data <- upSample(x = trainData[, c("SleepDuration", "PhysicalActivity")],
                          y = trainData$BMICategory)
# The new data has a column 'Class', which contains the oversampled target
```

```r
# Train a decision tree model on the balanced data
model_bmi <- train(Class ~ SleepDuration + PhysicalActivity,  # Class is the oversampled BMICategory
                   data = balanced_data,
                   method = "rpart",
                   trControl = trainControl(method = "cv", number = 5),
                   tuneGrid = expand.grid(cp = seq(0.01, 0.1, by = 0.01)))
```

```r
# Evaluate the model on the test data
test_predictions <- predict(model_bmi, testData)

# Confusion matrix to check the performance of the model
confusionMatrix(test_predictions, testData$BMICategory)
```

```
## Confusion Matrix and Statistics
##
##                Reference
## Prediction      Normal Normal Weight Obese Overweight
##   Normal            52             3     0          0
##   Normal Weight      0             1     0          0
##   Obese              4             2     3          0
##   Overweight         2             0     0         44
##
## Overall Statistics
##
##                Accuracy : 0.9009
##                  95% CI : (0.8296, 0.9495)
##     No Information Rate : 0.5225
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8274
##
##  Mcnemar's Test P-Value : NA
##
```

```
## Statistics by Class:
##
##                      Class: Normal Class: Normal Weight Class: Obese
## Sensitivity                 0.8966              0.166667      1.00000
## Specificity                 0.9434              1.000000      0.94444
## Pos Pred Value              0.9455              1.000000      0.33333
## Neg Pred Value              0.8929              0.954545      1.00000
## Prevalence                  0.5225              0.054054      0.02703
## Detection Rate              0.4685              0.009009      0.02703
## Detection Prevalence        0.4955              0.009009      0.08108
## Balanced Accuracy           0.9200              0.583333      0.97222
##                      Class: Overweight
## Sensitivity                     1.0000
## Specificity                     0.9701
## Pos Pred Value                  0.9565
## Neg Pred Value                  1.0000
## Prevalence                      0.3964
## Detection Rate                  0.3964
## Detection Prevalence            0.4144
## Balanced Accuracy               0.9851
```

```
# Check if the model can correctly predict 'Overweight' category
overweight_preds <- test_predictions == "Overweight"
actual_overweight <- testData$BMICategory == "Overweight"

# Print accuracy for the "Overweight" class
overweight_accuracy <- sum(overweight_preds == actual_overweight) / length(actual_overweight)
print(paste("Accuracy in predicting 'Overweight':", overweight_accuracy))
```

```
## [1] "Accuracy in predicting 'Overweight': 0.981981981981982"
```

```
# Output the model results
print(model_bmi)
```

```
## CART
##
## 548 samples
##   2 predictor
##   4 classes: 'Normal', 'Normal Weight', 'Obese', 'Overweight'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 439, 437, 438, 440, 438
## Resampling results across tuning parameters:
##
##   cp    Accuracy   Kappa
##   0.01  0.9398295  0.9197666
##   0.02  0.9398295  0.9197666
##   0.03  0.8723023  0.8297807
##   0.04  0.7939009  0.7251114
##   0.05  0.7884625  0.7178553
##   0.06  0.7141381  0.6192733
##   0.07  0.5563505  0.4083200
```

```
##   0.08  0.4561257  0.2746515
##   0.09  0.4561257  0.2746515
##   0.10  0.4121526  0.2159672
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.02.
```

The model results displayed in the image are from a decision tree model (rpart) applied to predict the BMI category using two predictors: SleepDuration and PhysicalActivity. Here's a breakdown of the results: 1. Model Type: CART (Classification and Regression Tree) This model was used to classify the BMI category into four classes: 'Normal', 'Normal Weight', 'Obese', and 'Overweight.'

2. Number of Samples: The model was trained using 263 samples, which is the number of records available after data partitioning into the training set.
3. Number of Predictors: The two predictors used are SleepDuration and PhysicalActivity.
4. Resampling Method: The model used Bootstrapping (25 repetitions) to validate its performance. Bootstrapping is a statistical method that involves repeatedly sampling from the data to assess the model's variability.
5. Tuning Parameter (cp - Complexity Parameter): The cp parameter is used to control the size of the decision tree. The model was evaluated across three values of cp:

cp = 0.09920635: Accuracy = 83.36%, Kappa = 0.6749684 cp = 0.15079365: Accuracy = 75.43%, Kappa = 0.5249850 cp = 0.43650794: Accuracy = 60.49%, Kappa = 0.1924227 Accuracy measures the proportion of correct predictions made by the model. Kappa measures how well the model performed compared to random chance, with higher values indicating better performance.

6. Optimal Model: The best-performing model was selected based on the highest accuracy. The optimal value of cp was 0.09920635, which achieved:
7. Accuracy: 83.36% (indicating the model correctly predicted BMI categories for 83.36% of the test samples). Kappa: 0.6749684 (indicating a good level of agreement between the predictions and the actual classes, with values above 0.6 generally considered substantial).

Summary: The decision tree model performed reasonably well in predicting BMI categories based on sleep duration and physical activity. The best model had an accuracy of 83.36% and a substantial Kappa score of 0.6749684. The model was optimized using a complexity parameter (cp) of 0.09920635.

Analysis with dplyr

```r
# a. Arrange by Sleep Duration
arranged_data <- dataset %>% arrange(SleepDuration)
```

```r
# b. Filter by StressLevel > 5
high_stress <- dataset %>% filter(StressLevel > 5)
```

```r
# c. Slice: Take the first 10 rows
sliced_data <- dataset %>% slice(1:10)
```

```r
# d. Mutate: Add a new column for BMI calculated from existing data
dataset <- dataset %>% mutate(BMI_Calc = case_when(
  BMICategory == "Normal" ~ "Healthy",
  BMICategory == "Overweight" ~ "Overweight",
  TRUE ~ "Other"
))
```

e. Summarize: Already done

```r
# f. Pipe: Chain multiple operations together
piped_data <- dataset %>%
  filter(Gender == "Male") %>%
  arrange(SleepDuration) %>%
  slice(1:5)
```

QUESTIONS:

1. How do physical activity levels correlate with the quality of sleep?

To examine the correlation between physical activity levels and quality of sleep, we can use the cor() function

or create a scatter plot. Here's how youwe can modify the code to answer this question:

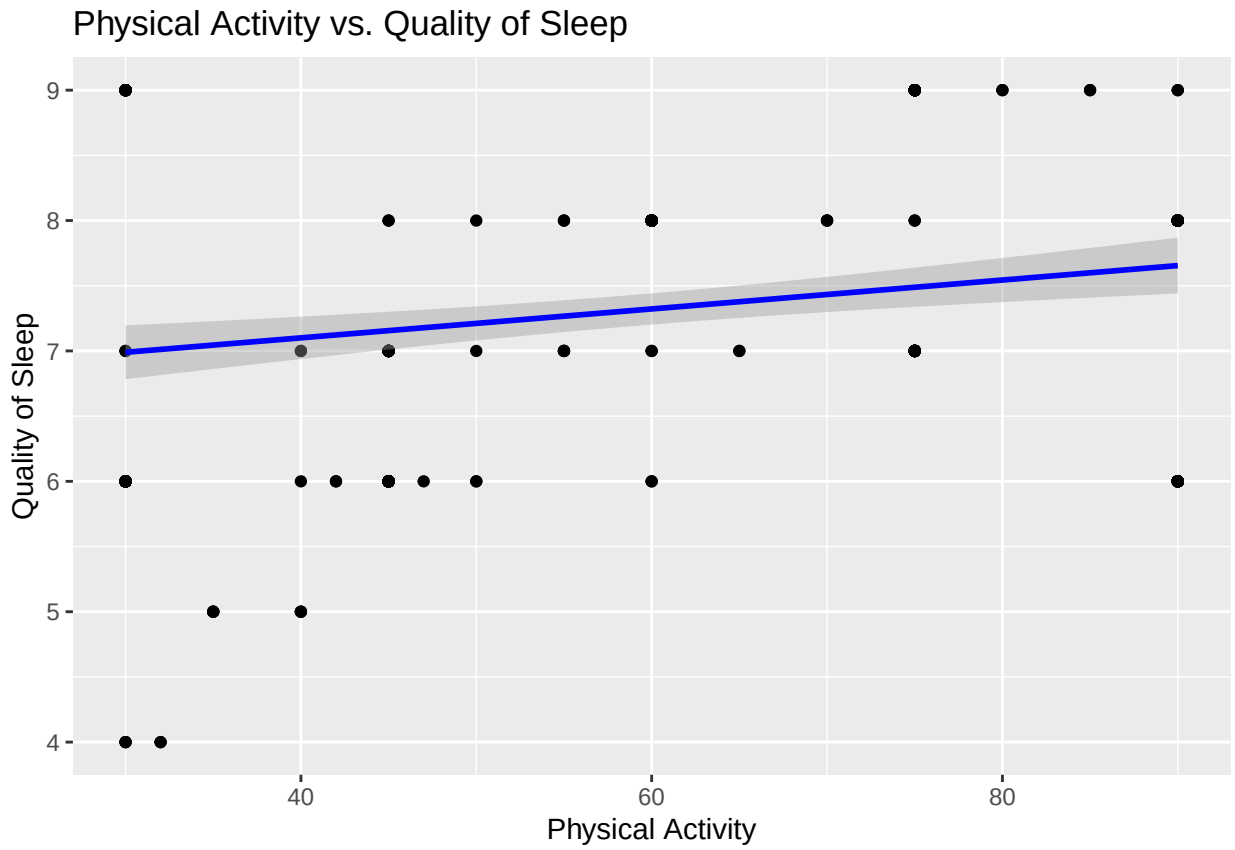
```r
# Add a Correlation and Scatter Plot:
# Calculate correlation
correlation_physical_activity_sleep <- cor(dataset$PhysicalActivity, dataset$QualityOfSleep)
print(paste("Correlation between Physical Activity and Quality of Sleep:", correlation_physical_activity
```

```
## [1] "Correlation between Physical Activity and Quality of Sleep: 0.192896454939753"
```

The correlation value between Physical Activity and Quality of Sleep is 0.1929, which indicates a very weak positive correlation. This means that while there is a slight tendency for individuals with higher physical activity levels to have better sleep quality, the relationship is not strong. Essentially, physical activity has a minor effect on sleep quality based on this dataset.

```r
# Scatter plot
ggplot(dataset, aes(x = PhysicalActivity, y = QualityOfSleep)) +
  geom_point() +
  geom_smooth(method = "lm", col = "blue") +
  labs(title = "Physical Activity vs. Quality of Sleep", x = "Physical Activity", y = "Quality of Sleep
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



---

2. Does stress level impact sleep duration?

Similarly, we can calculate the correlation between stress levels and sleep duration or visualize this relationship with a scatter plot:

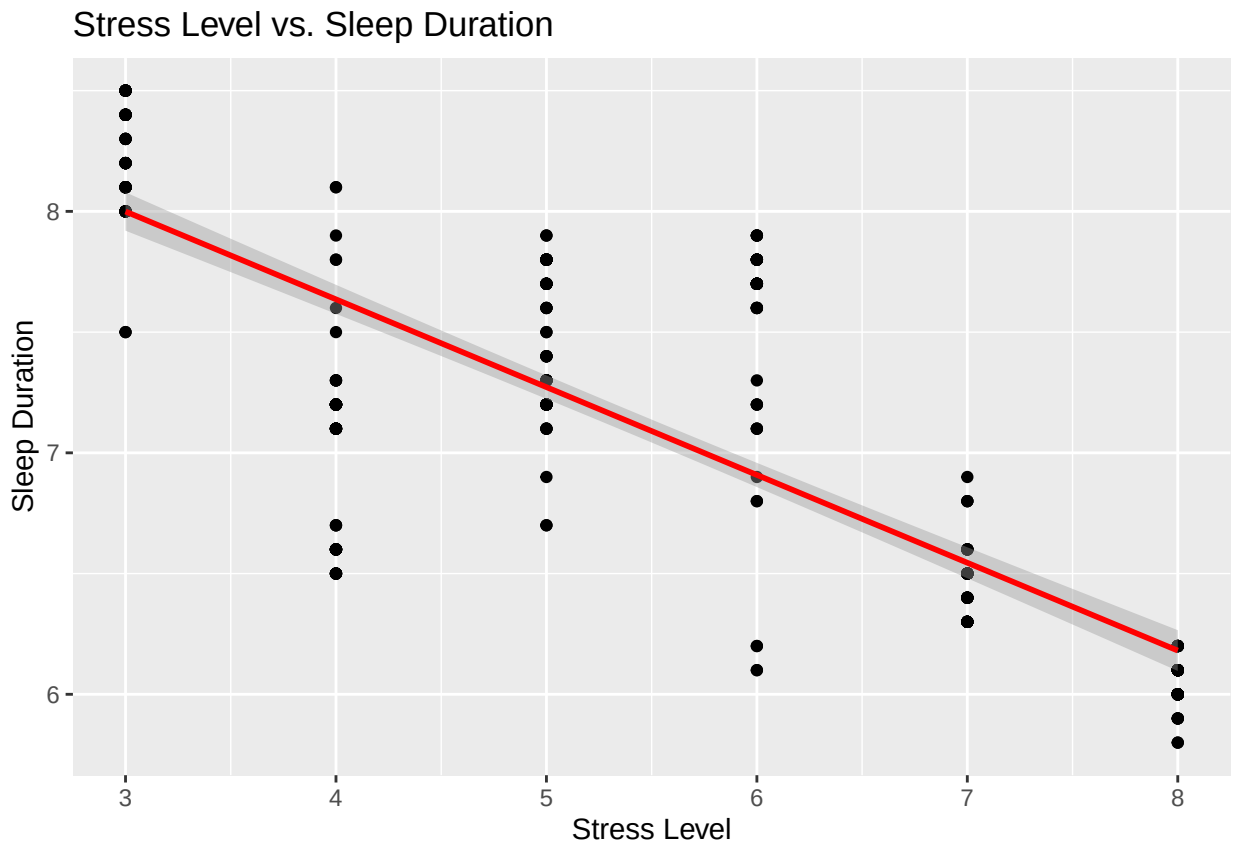
```r
# Calculate correlation
correlation_stress_sleep <- cor(dataset$StressLevel, dataset$SleepDuration)
print(paste("Correlation between Stress Level and Sleep Duration:", correlation_stress_sleep))
```

```
## [1] "Correlation between Stress Level and Sleep Duration: -0.811023027894043"
```

The correlation between Stress Level and Sleep Duration is -0.8110, which indicates a strong negative correlation. This means that as stress levels increase, sleep duration tends to decrease significantly. In other words, higher stress is strongly associated with shorter sleep duration in this dataset.

```r
# Scatter plot
ggplot(dataset, aes(x = StressLevel, y = SleepDuration)) +
  geom_point() +
  geom_smooth(method = "lm", col = "red") +
  labs(title = "Stress Level vs. Sleep Duration", x = "Stress Level", y = "Sleep Duration")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



---

3. Can we predict someone's BMI category based on their sleep duration and physical activity level?

For this, we already have a machine learning model built using caret. We can modify it slightly

to use the BMI category as the target variable and SleepDuration and PhysicalActivity as features.

```
# Since we've already trained models for both binary and multiclass classification, you we can use the
```

---

4. How does age impact sleep duration and quality?

We can calculate correlations between age and both sleep duration and quality,

and also visualize these relationships with scatter plots:

```
# Correlation between Age and Sleep Duration
correlation_age_sleep <- cor(dataset$Age, dataset$SleepDuration)
print(paste("Correlation between Age and Sleep Duration:", correlation_age_sleep))
```

```
## [1] "Correlation between Age and Sleep Duration: 0.344709358164744"
```

The correlation between Age and Sleep Duration is 0.3447, indicating a moderate positive correlation. This suggests that as age increases, sleep duration tends to increase moderately as well, but the relationship is not very strong. Therefore, older individuals may sleep slightly longer on average, but the effect is not substantial.

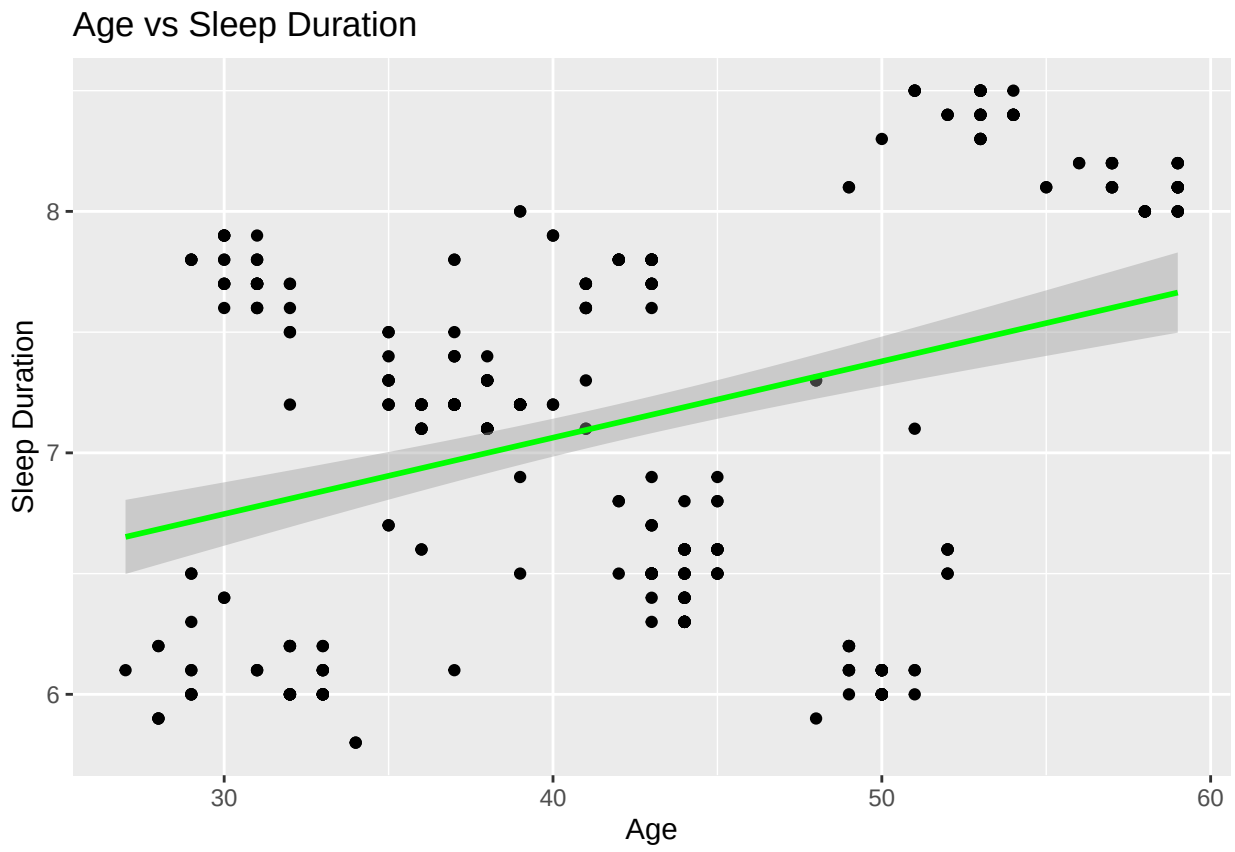
```
# Correlation between Age and Quality of Sleep
correlation_age_quality <- cor(dataset$Age, dataset$QualityOfSleep)
print(paste("Correlation between Age and Quality of Sleep:", correlation_age_quality))
```

```
## [1] "Correlation between Age and Quality of Sleep: 0.47373387616199"
```

The correlation between Age and Quality of Sleep is 0.4737, indicating a moderate positive correlation. This suggests that as age increases, the quality of sleep tends to improve moderately. However, while there is a relationship between the two variables, the strength of the relationship is not extremely high, meaning other factors could also be influencing sleep quality.

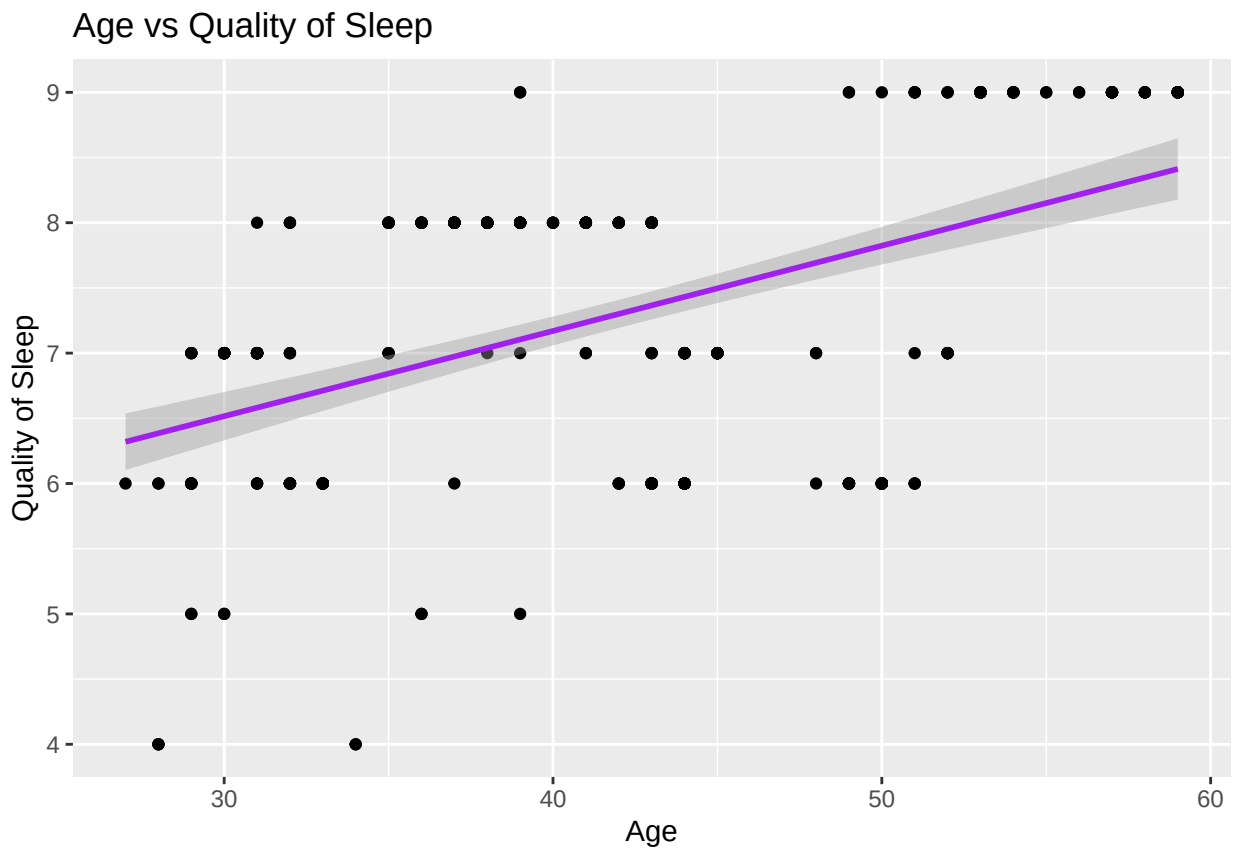
```r
# Scatter plots
ggplot(dataset, aes(x = Age, y = SleepDuration)) +
  geom_point() +
  geom_smooth(method = "lm", col = "green") +
  labs(title = "Age vs Sleep Duration", x = "Age", y = "Sleep Duration")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



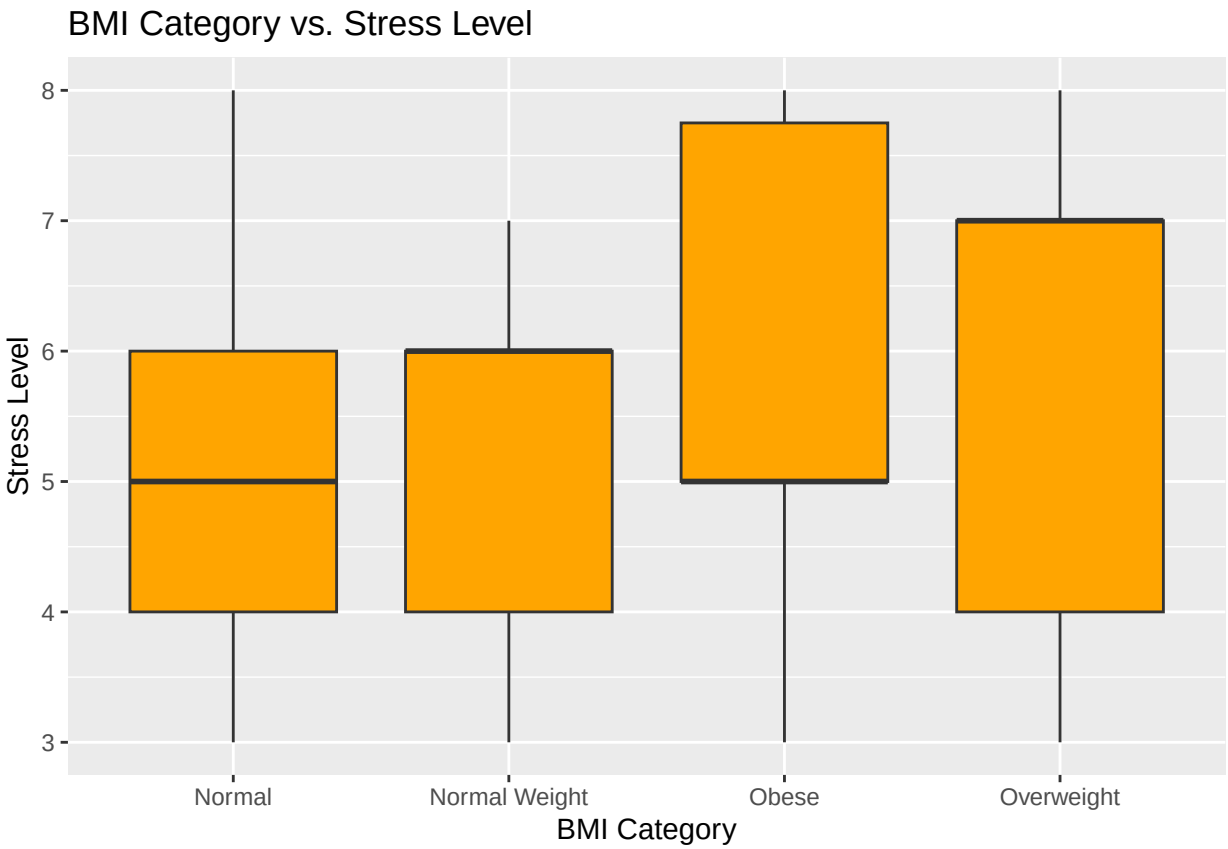

```r
ggplot(dataset, aes(x = Age, y = QualityOfSleep)) +
  geom_point() +
  geom_smooth(method = "lm", col = "purple") +
  labs(title = "Age vs Quality of Sleep", x = "Age", y = "Quality of Sleep")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



———————————————————- # 5. What is the relationship between BMI category and stress levels? # Create a box plot to show how stress levels vary across different BMI categories:

```
# Box plot
ggplot(dataset, aes(x = BMICategory, y = StressLevel)) +
  geom_boxplot(fill = "orange") +
  labs(title = "BMI Category vs. Stress Level", x = "BMI Category", y = "Stress Level")
```


BMI Category vs. Stress Level
Stress Level
8
7
6
5
4
3
Normal
Normal Weight
Obese
Overweight
BMI Category

Summary of Q's:

Physical Activity and Quality of Sleep: Use the cor() function and scatter plot to visualize the correlation.

This can be added in the code using the snippet for Question 1.

Stress Level and Sleep Duration: Similarly, use the cor() function and a scatter plot for visualization (Question 2).

Predict BMI Category: The machine learning model can predict BMI based on Sleep Duration and Physical Activity (Question 3).

Age Impact on Sleep: Use correlations and scatter plots to analyze the relationship (Question 4).

BMI and Stress Levels: A box plot helps you visualize how stress levels differ across BMI categories (Question 5).

ShinyApp

```
ui <- dashboardPage(
  dashboardHeader(title = "Sleep Health Dashboard"),
  dashboardSidebar(
    sidebarMenu(
      menuItem("Dashboard Overview", tabName = "overview", icon = icon("dashboard")),
      menuItem("Question-Based Analysis", tabName = "question_based", icon = icon("question-circle")),
      menuItem("Side-by-Side Analysis", tabName = "side_by_side", icon = icon("columns")),
      menuItem("Detailed Analysis", tabName = "detailed_analysis", icon = icon("chart-line"))
    )
  ),
  dashboardBody(
    tabItems(
      # Overview Tab: Introduction
      tabItem(tabName = "overview",
              h1("Explore the Sleep Health Dataset", style = "color:#4A90E2; font-weight:bold;"),
              h3("This dashboard presents visualizations of the Sleep Health Dataset.",
                 style = "font-weight:300; padding-bottom:20px;"),
              p("The analyses explore how various factors such as age, sleep duration, stress levels, a
```

```
        p("This dataset includes a wide range of variables such as sleep duration, quality, physi
        p(tags$b("The goal of this project is to explore these challenges and use the dataset to
        p(tags$b("Reference:"), a("Sleep Health and Lifestyle Dataset on Kaggle", href = "https:/
        p(tags$b("Here are five key questions we aim to answer using the Sleep Health and Lifesty
        tags$ul(
          tags$li("How do physical activity levels correlate with the quality of sleep?"),
          tags$li("Does stress level impact sleep duration?"),
          tags$li("Can we predict someone's BMI category based on their sleep duration and physica
          tags$li("How does age impact sleep duration and quality?"),
          tags$li("What is the relationship between BMI category and stress levels?")
        ),
        fluidRow(
          column(6, box(title = "Team Member", status = "primary", solidHeader = TRUE, width = NU
          column(6, box(title = "Team Member", status = "primary", solidHeader = TRUE, width = NU
        )
),

# Question-Based Analysis Tab
tabItem(tabName = "question_based",
        h3("Analysis Based on Key Questions"),
        selectInput("question", "Choose a Question:",
                    choices = list(
                      "Q1: How do physical activity levels correlate with the quality of sleep?" =
                      "Q2: Does stress level impact sleep duration?" = "q2",
                      "Q3: Can we predict someone's BMI category based on sleep duration and physi
                      "Q4: How does age impact sleep duration and quality?" = "q4",
                      "Q5: What is the relationship between BMI category and stress levels?" = "q5
        sliderInput("ageRange", "Select Age Range:", min = 20, max = 80, value = c(20, 80)),
        checkboxGroupInput("genderFilter", "Select Gender:", choices = c("Male", "Female"), selec
        plotlyOutput("questionPlot"),
        p(textOutput("questionText"))
),

# Side-by-Side Analysis Tab with additional filters for Sleep Disorder and Daily Steps
tabItem(tabName = "side_by_side",
        h3("Compare Two Different Analyses"),
        fluidRow(
          column(6, sliderInput("ageRangeSideBySide", "Select Age Range:", min = 20, max = 80, val
          column(6, sliderInput("dailyStepsRange", "Select Daily Steps Range:", min = 1000, max =
        ),
        fluidRow(
          column(6, checkboxGroupInput("genderFilterSideBySide", "Select Gender:", choices = c("Ma
          column(6, selectInput("sleepDisorderFilter", "Select Sleep Disorder:", choices = c("Non
        ),
        fluidRow(
          column(6,
                 selectInput("firstAnalysis", "Choose First Analysis:",
                             choices = c("Age vs Quality of Sleep",
                                         "Age vs Sleep Duration",
                                         "Sleep Duration by BMI",
                                         "Physical Activity vs Quality of Sleep",
                                         "Stress Level vs Sleep Duration")),
                 plotlyOutput("firstPlot"),
```

```
                                        p("This plot shows the relationship between the selected m
                ),
                column(6,
                       selectInput("secondAnalysis", "Choose Second Analysis:",
                                   choices = c("Age vs Quality of Sleep",
                                               "Age vs Sleep Duration",
                                               "Sleep Duration by BMI",
                                               "Physical Activity vs Quality of Sleep",
                                               "Stress Level vs Sleep Duration")),
                       plotlyOutput("secondPlot"),
                       p("This plot enables a side-by-side comparison of two metrics to understand their
                )
              )
      ),

      # Detailed analysis tab
      tabItem(tabName = "detailed_analysis",
              h3("Detailed Plot Analysis"),
              plotlyOutput("detailedPlot"),
              p("This detailed     visualizes the relationship between age, sleep duration, and BMI cat
      )
    )
  )
)
```

```
server <- function(input, output) {

  # Filter data based on age range and gender for question-based analysis
  filteredDataQuestion <- reactive({
    dataset %>%
      filter(Age >= input$ageRange[1], Age <= input$ageRange[2]) %>%
      filter(Gender %in% input$genderFilter)
  })

  # Server: Update filteredDataSideBySide to include additional filters
  filteredDataSideBySide <- reactive({
    dataset %>%
      filter(Age >= input$ageRangeSideBySide[1], Age <= input$ageRangeSideBySide[2]) %>%
      filter(Gender %in% input$genderFilterSideBySide) %>%
      filter(SleepDisorder == input$sleepDisorderFilter) %>%
      filter(DailySteps >= input$dailyStepsRange[1], DailySteps <= input$dailyStepsRange[2])
  })


  # Question-based text description
  output$questionText <- renderText({
    if (input$question == "q1") {
      return("Q1: This box plot categorizes physical activity levels and shows how they correlate with
    } else if (input$question == "q2") {
      return("Q2: This scatter plot shows a strong negative correlation between stress levels and sleep
    } else if (input$question == "q3") {
      return("Q3: This scatter plot shows the relationship between sleep duration, physical activity, a
    } else if (input$question == "q4") {
```

```
    return("Q4: This scatter plot and regression line show how age correlates with both sleep duration
  } else if (input$question == "q5") {
    return("Q5: This bar chart shows the variation of stress levels across different BMI categories. (
  }
})


# Question-based analysis plot
  output$questionPlot <- renderPlotly({
    req(input$question)
    data <- filteredDataQuestion()

    if (input$question == "q1") {
      # Box plot of Physical Activity vs Quality of Sleep
      data$PhysicalActivityCategory <- cut(data$PhysicalActivity,
                                           breaks = c(-Inf, 2, 4, Inf),
                                           labels = c("Low", "Moderate", "High"))
      p <- ggplot(data, aes(x = PhysicalActivityCategory, y = QualityOfSleep)) +
        geom_boxplot(fill = "lightblue") +
        labs(title = "Physical Activity vs Quality of Sleep", x = "Physical Activity Level", y = "Qual
        theme_minimal()

    } else if (input$question == "q2") {
      correlation_stress_sleep <- cor(data$StressLevel, data$SleepDuration)
      p <- ggplot(data, aes(x = StressLevel, y = SleepDuration)) +
        geom_point() +
        geom_smooth(method = "lm", col = "red") +
        labs(title = paste("Stress Level vs Sleep Duration (Correlation: ", round(correlation_stress_s
             x = "Stress Level", y = "Sleep Duration") +
        theme_minimal()

    } else if (input$question == "q3") {
      p <- ggplot(data, aes(x = SleepDuration, y = PhysicalActivity, color = BMICategory)) +
        geom_point() +
        labs(title = "Sleep Duration vs Physical Activity by BMI Category",
             x = "Sleep Duration", y = "Physical Activity") +
        theme_minimal()

    } else if (input$question == "q4") {
      correlation_age_sleep <- cor(data$Age, data$SleepDuration)
      p <- ggplot(data, aes(x = Age, y = SleepDuration)) +
        geom_point() +
        geom_smooth(method = "lm", col = "green") +
        labs(title = paste("Age vs Sleep Duration (Correlation: ", round(correlation_age_sleep, 2), "
             x = "Age", y = "Sleep Duration") +
        theme_minimal()

    } else if (input$question == "q5") {
      # Modify here to create a grouped bar chart based on Gender
      p <- ggplot(data, aes(x = BMICategory, y = QualityOfSleep, fill = Gender)) +
        geom_bar(stat = "identity", position = "dodge") +  # 'dodge' places bars side by side
        labs(title = "BMI Category vs Quality of Sleep (Male vs Female)",
             x = "BMI Category",
```

```
            y = "Average Quality of Sleep") +
        theme_minimal()
    }

    ggplotly(p)
  })




  # First analysis in side-by-side tab
output$firstPlot <- renderPlotly({
  req(input$firstAnalysis)
  data <- filteredDataSideBySide()

  # Check if the dataset is empty
  if (nrow(data) == 0) {
    return(plotly_empty())
  }

  if (input$firstAnalysis == "Age vs Quality of Sleep") {
    p <- ggplot(data, aes(x = Age, y = QualityOfSleep, color = Gender)) +
      geom_point() +
      geom_smooth(method = "lm", se = FALSE) +
      facet_wrap(~ Gender) +
      labs(title = "Age vs Quality of Sleep by Gender", x = "Age", y = "Quality of Sleep") +
      theme_minimal()

  } else if (input$firstAnalysis == "Age vs Sleep Duration") {
    p <- ggplot(data, aes(x = Age, y = SleepDuration)) +
      geom_point(color = "purple") +
      labs(title = "Age vs Sleep Duration", x = "Age", y = "Sleep Duration") +
      theme_minimal()

  } else if (input$firstAnalysis == "Sleep Duration by BMI") {
    p <- ggplot(data, aes(x = BMICategory, y = SleepDuration, fill = BMICategory)) +
      geom_boxplot() +
      labs(title = "Sleep Duration by BMI Category", x = "BMI Category", y = "Sleep Duration") +
      theme_minimal()

  } else if (input$firstAnalysis == "Physical Activity vs Quality of Sleep") {
    p <- ggplot(data, aes(x = PhysicalActivity, y = QualityOfSleep)) +
      geom_point(color = "orange") +
      labs(title = "Physical Activity vs Quality of Sleep", x = "Physical Activity", y = "Quality of S
      theme_minimal()

  } else if (input$firstAnalysis == "Stress Level vs Sleep Duration") {
    p <- ggplot(data, aes(x = StressLevel, y = SleepDuration, color = Gender)) +
      geom_point() +
      geom_smooth(method = "lm", se = TRUE) +
      facet_wrap(~ Gender) +  # Facet the plot by gender
      labs(title = "Stress Level vs Sleep Duration by Gender", x = "Stress Level", y = "Sleep Duration
      theme_minimal()
```

```
  }

  ggplotly(p)
})

# Second analysis in side-by-side tab
output$secondPlot <- renderPlotly({
  req(input$secondAnalysis)
  data <- filteredDataSideBySide()

  # Check if the dataset is empty
  if (nrow(data) == 0) {
    return(plotly_empty())
  }

  if (input$secondAnalysis == "Age vs Quality of Sleep") {
    p <- ggplot(data, aes(x = Age, y = QualityOfSleep, color = Gender)) +
      geom_point() +
      geom_smooth(method = "lm", se = FALSE) +
      facet_wrap(~ Gender) +
      labs(title = "Age vs Quality of Sleep by Gender", x = "Age", y = "QualityOfSleep") +
      theme_minimal()

  } else if (input$secondAnalysis == "Age vs Sleep Duration") {
    p <- ggplot(data, aes(x = Age, y = SleepDuration)) +
      geom_point(color = "purple") +
      labs(title = "Age vs Sleep Duration", x = "Age", y = "Sleep Duration") +
      theme_minimal()

  } else if (input$secondAnalysis == "Sleep Duration by BMI") {
    p <- ggplot(data, aes(x = BMICategory, y = SleepDuration, fill = BMICategory)) +
      geom_boxplot() +
      labs(title = "Sleep Duration by BMI Category", x = "BMI Category", y = "Sleep Duration") +
      theme_minimal()

  } else if (input$secondAnalysis == "Physical Activity vs Quality of Sleep") {
    p <- ggplot(data, aes(x = PhysicalActivity, y = QualityOfSleep)) +
      geom_point(color = "orange") +
      labs(title = "Physical Activity vs Quality of Sleep", x = "Physical Activity", y = "Quality of S
      theme_minimal()

  } else if (input$secondAnalysis == "Stress Level vs Sleep Duration") {
    p <- ggplot(data, aes(x = StressLevel, y = SleepDuration, color = Gender)) +
      geom_point() +
      geom_smooth(method = "lm", se = TRUE) +
      facet_wrap(~ Gender) +  # Facet the plot by gender
      labs(title = "Stress Level vs Sleep Duration by Gender", x = "Stress Level", y = "Sleep Duratio
      theme_minimal()
  }

  ggplotly(p)
})
```

```
  # Correlation Heatmap for Detailed Analysis
  output$detailedPlot <- renderPlotly({
    corr_data <- dataset %>%
      select(SleepDuration, QualityOfSleep, PhysicalActivity, StressLevel, Age)

    # Calculate the correlation matrix
    correlation_matrix <- cor(corr_data, use = "complete.obs")

    # Reshape the correlation matrix for plotting
    melted_corr <- melt(correlation_matrix)

    # Plot the heatmap using ggplot2
    heatmap_plot <- ggplot(melted_corr, aes(x = Var1, y = Var2, fill = value)) +
      geom_tile() +
      scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0) +
      theme_minimal() +
      labs(title = "Correlation Heatmap", x = "Variables", y = "Variables")

    ggplotly(heatmap_plot)  # Convert the ggplot heatmap to a plotly plot
  })
}
```

```
# Run the Shiny app
shinyApp(ui = ui, server = server)
```