Math 133: Statistical Learning Methods - Module 1 Assessment

Pranav Jayakumar

February 19, 2025

Abstract

In this module assessment, we will draw from survey data and fit simple linear models to understand the effects of various predictors on students' stress.

1 Data Analysis

For the following analysis, we will fit simple linear models of the form

$$Y = \beta_0 + \beta_1 X$$

where $Y =$ `stress_score` and X is one of the following numeric features: `maximum_alcohol_consumed`, `gpa`, or `screen_time`.

1.1 Data Visualization

We will begin by creating scatterplots of students' stress scores vs. each of the three aforementioned predictors. We will use a 70-30 training-testing split for our models.

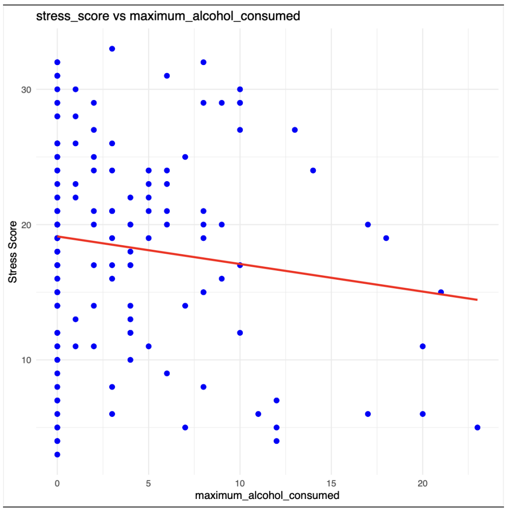

Figure 1: Scatterplot of Stress Score vs. Maximum Alcohol Consumed

The linear model for stress score vs. maximum alcohol consumed is represented by:

$$\hat{y} = 18.8543 - 0.2999x$$

We observe an R^2 score of 0.0417.

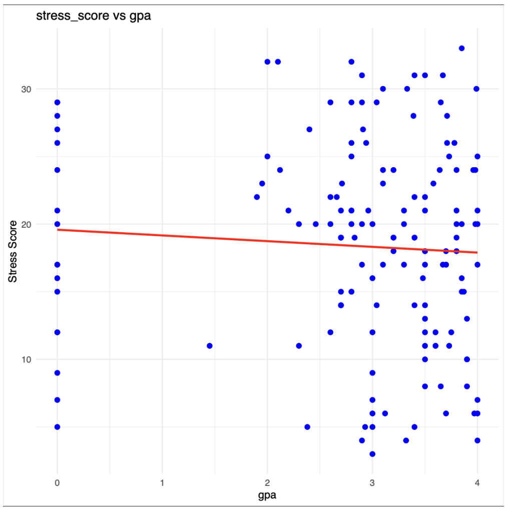

Figure 2: Scatterplot of Stress Score vs. GPA

The linear model for stress score vs. GPA is represented by:

$$\hat{y} = 19.8029 - 0.3343x$$

We observe an R^2 score of 0.0027.

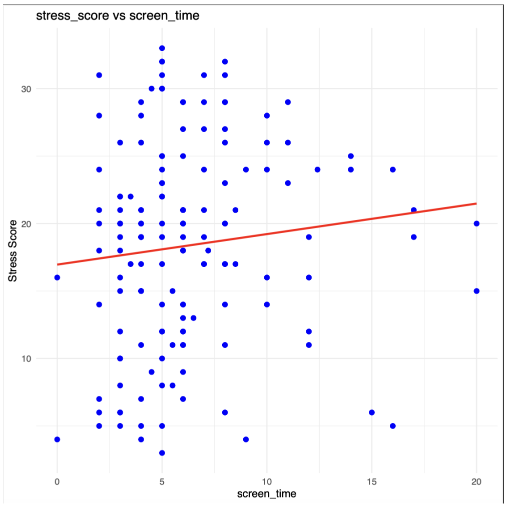

Figure 3: Scatterplot of Stress Score vs. Screen Time

The linear model for stress score vs. screen time is represented by:

$$\hat{y} = 16.7664 + 0.3193x$$

We observe an R^2 score of 0.0222.

1.2 Insights

We observe that the linear model `stress_score~screen_time` has the highest R^2 score. Therefore, there is evidence to suggest that screen time is the best out of the three predictors of stress.

Plotting screen time data in a histogram shows that the data is approximately normally distributed.

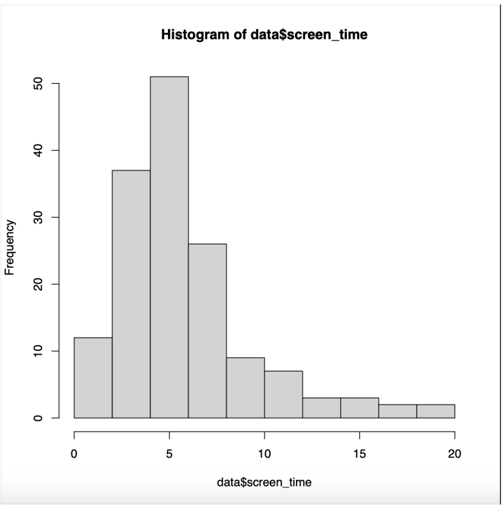

Figure 4: Histogram of Screen Time

2 R Coding

```
    #!/usr/bin/env Rscript
library(ggplot2)

set.seed(123)

fit_linear_model <- function(y, x, raw_data) {
  # Train-test split
  n <- nrow(raw_data)
  trainIndex <- sample(n, round(0.7 * n, 0))
  train <- raw_data[trainIndex, ]
  test <- raw_data[-trainIndex, ]

  # Construct formula
  formula <- as.formula(paste(y, "~", x))

  # Fit model
  model <- lm(formula, data = train)

  overview = summary(model)
  print(overview)

  # Predict on testing data
  y_test <- test[[y]]
  y_hat <- predict(model, newdata = test)

  # Analyze accuracy
  SSE <- sum((y_test - y_hat)^2)
  MSE <- SSE / nrow(test)
  RMSE <- sqrt(MSE)
  SST <- sum((y_test - mean(y_test))^2)
  R2 <- 1 - SSE / SST

  return(list(SSE = SSE, MSE = MSE, RMSE = RMSE, SST = SST, R2 = R2))
}

main <- function() {
  # Load data
  data <- read.csv("../../data/survey_fall2023.csv")

  # Define predictors
  predictors <- c("maximum_alcohol_consumed", "gpa", "screen_time")
  results <- list()

  # Open PDF device for saving all plots
  pdf("Rplots.pdf")  # This will save all plots sequentially to Rplots.pdf

  # Iterate over predictors
  for (predictor in predictors) {
    if (!predictor %in% colnames(data)) {
      cat("\nWarning:␣Predictor", predictor, "not␣found␣in␣dataset.␣Skipping...\n")
      next
    }

    cat("\nLinear␣model␣for␣stress␣score␣~", predictor, "\n")
    result <- fit_linear_model("stress_score", predictor, data)

    # Store results for later comparison
    results[[predictor]] <- result

    # Format and print results
    formatted_results <- lapply(result[1:5], function(x) format(round(x, 4), nsmall = 4))
    print(formatted_results)

    # Create scatterplot for the predictor
    ggplot(data, aes_string(x = predictor, y = "stress_score")) +
      geom_point(color = "blue", size = 2) +
      geom_smooth(method = "lm", color = "red", se = FALSE) +
      ggtitle(paste("stress_score␣vs", predictor)) +
      xlab(predictor) +
      ylab("Stress␣Score") +
      theme_minimal() -> plot  # Assign ggplot to a variable

    print(plot)  # Ensure the plot is sent to the PDF file
  }

  hist(data$screen_time, breaks = "Freedman-Diaconis") -> plot1
  print(plot1)

  # Close PDF device
  dev.off()

  cat("\nAll␣plots␣saved␣to␣Rplots.pdf.\n")
}

# Run the script if executed directly
if (interactive() || identical(Sys.getenv("R_SCRIPT"), "")) {
  main()
}
```