Final Project Analysis

Jinhong Yu

2025-05-14

Statistical Analysis for Student Habits vs. Academic Performance

1. Introduction

This analysis investigates how various lifestyle habits influence academic performance among students, using a synthetic dataset of 1,000 records from Kaggle. The dataset includes a continuous target variable of final exam score, alongside 15+ features such as study hours, sleep duration, diet quality, social media usage, and mental health.

The goal is to apply linear regression analysis to:

1. Explore the relationships between lifestyle variables and academic success
2. Check and address regression assumptions
3. Select meaningful predictors and construct the prediction models
4. Quantify and interpret the impact of key features

2. Exploratory Data Analysis

2.1 Prepare Packages

```
library("readr")
```

```
## Warning: package 'readr' was built under R version 4.4.2
```

```
library("dplyr")
```

```
## Warning: package 'dplyr' was built under R version 4.4.2
```

```
##
## 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("ggplot2")
```

```
## Warning: package 'ggplot2' was built under R version 4.4.2
```

```r
library("corrplot")
```

```
## Warning: package 'corrplot' was built under R version 4.4.3
```

```
## corrplot 0.95 loaded
```

```r
library("lmtest")
```

```
## Warning: package 'lmtest' was built under R version 4.4.3
```

```
## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 4.4.2
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library("sandwich")
```

```
## Warning: package 'sandwich' was built under R version 4.4.2
```

```r
library("leaps")
```

```
## Warning: package 'leaps' was built under R version 4.4.3
```

```r
library("glmnet")
```

```
## Warning: package 'glmnet' was built under R version 4.4.3
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```

```r
library("boot")
```

2.2 Load Dataset

```
data <- read_csv("https://raw.githubusercontent.com/jy773Cornell/final-project-stsci6020-2025/refs/heads
```

```
## `curl` package not installed, falling back to using `url()`
## Rows: 1000 Columns: 16
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (7): student_id, gender, part_time_job, diet_quality, parental_education...
## dbl (9): age, study_hours_per_day, social_media_hours, netflix_hours, attend...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
colnames(data)
```

```
##  [1] "student_id"                    "age"
##  [3] "gender"                        "study_hours_per_day"
##  [5] "social_media_hours"            "netflix_hours"
##  [7] "part_time_job"                 "attendance_percentage"
##  [9] "sleep_hours"                   "diet_quality"
## [11] "exercise_frequency"            "parental_education_level"
## [13] "internet_quality"              "mental_health_rating"
## [15] "extracurricular_participation" "exam_score"
```

In this dataset, several variables are categorical variables. We converted those variables to factors.

```
# Convert categorical variables to factors
data <- data %>%
  mutate(
    gender = as.factor(gender),
    part_time_job = as.factor(part_time_job),
    diet_quality = as.factor(diet_quality),
    parental_education_level = as.factor(parental_education_level),
    internet_quality = as.factor(internet_quality), extracurricular_participation = as.factor(extracurri
  )

# Show the data type for each variable
data.frame(
  Column = names(data),
  Type = sapply(data, class)
)
```

```
##                                                 Column      Type
## student_id                                  student_id character
## age                                                age   numeric
## gender                                          gender    factor
## study_hours_per_day                study_hours_per_day   numeric
## social_media_hours                  social_media_hours   numeric
## netflix_hours                            netflix_hours   numeric
## part_time_job                            part_time_job    factor
## attendance_percentage            attendance_percentage   numeric
## sleep_hours                                sleep_hours   numeric
```

```
## diet_quality                                diet_quality    factor
## exercise_frequency                     exercise_frequency   numeric
## parental_education_level          parental_education_level    factor
## internet_quality                          internet_quality    factor
## mental_health_rating                  mental_health_rating   numeric
## extracurricular_participation extracurricular_participation    factor
## exam_score                                      exam_score   numeric
```

2.3 Visualization of Distributions and Relationships

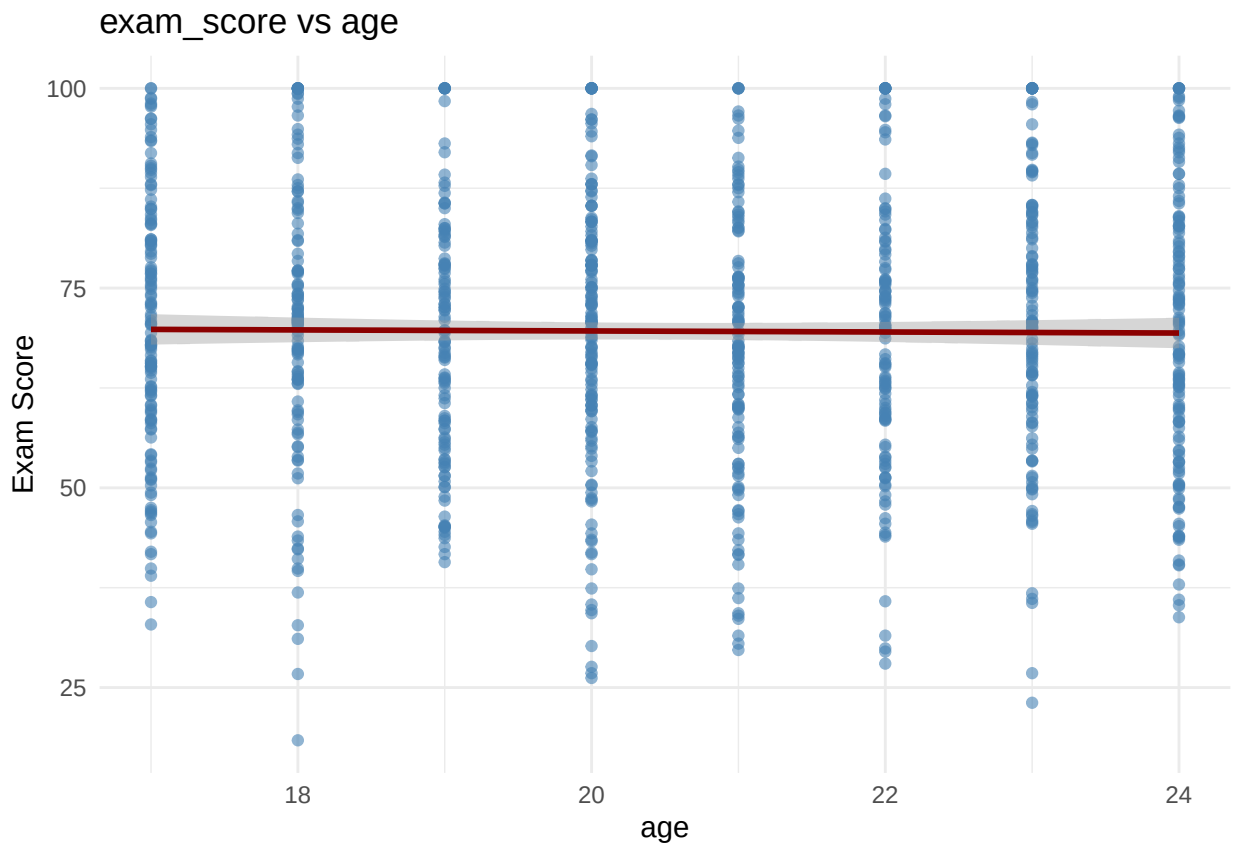
Based on the visualization plots, there are some preliminary findings:

1. study hours per day has the strongest positive relationship with exam scores.
2. Both social media usage and Netflix hours show a moderate negative correlation with exam scores.
3. Variables like mental health rating, exercise frequency, sleep hours, and attendance percentage all exhibit weak to modest positive associations with exam performance.
4. All the factor variable show very weak associations with exam scores. To build a model with parsimony, we will only consider above numerical variables as the initial predictors for later analysis.

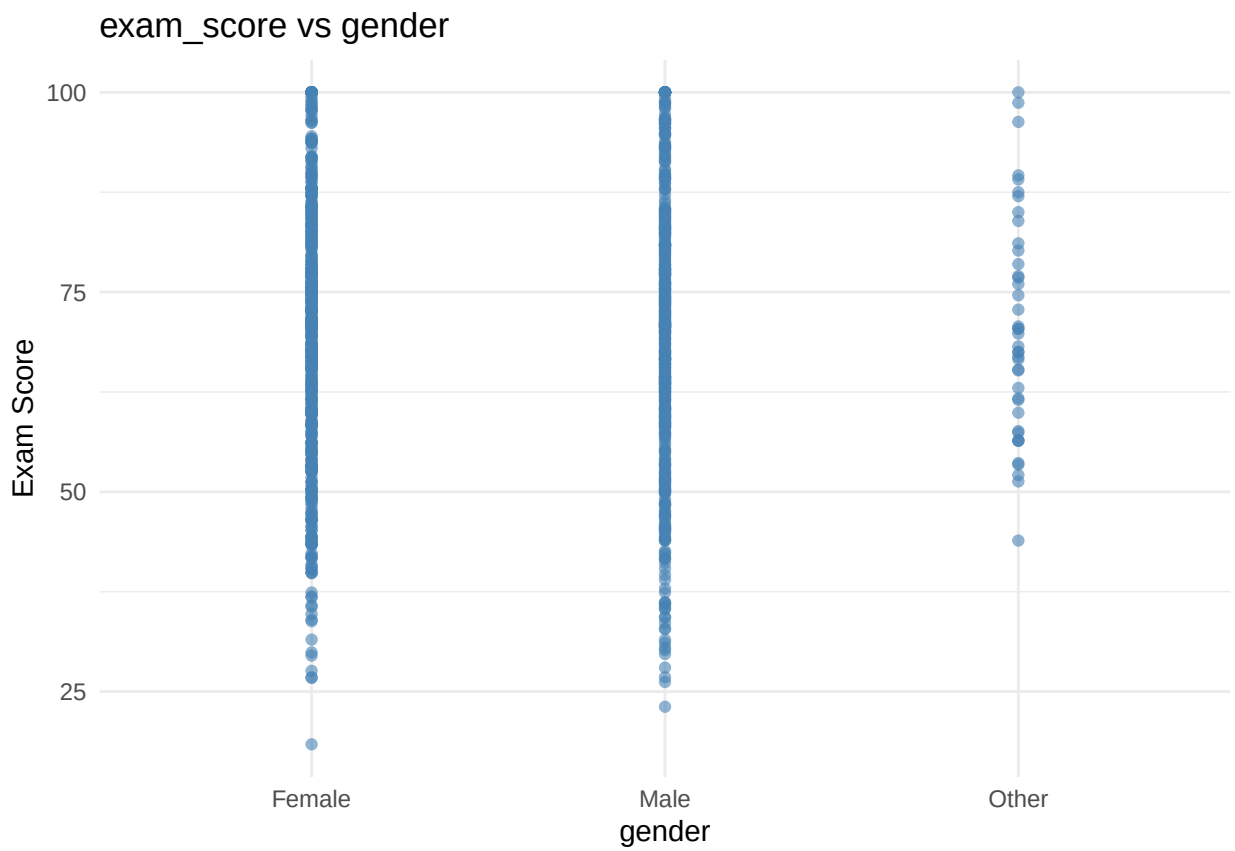
```r
# Visualize the distribution and relationships with the exam score
predictors <- data %>%
  select(-exam_score) %>%
  select(-student_id)

for (var in names(predictors)) {
  p <- ggplot(data, aes_string(x = var, y = "exam_score")) +
    geom_point(alpha = 0.6, color = "steelblue") +
    geom_smooth(method = "lm", se = TRUE, color = "darkred") +
    labs(title = paste("exam_score vs", var),
         x = var,
         y = "Exam Score") +
    theme_minimal()

  print(p)
}
```

```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

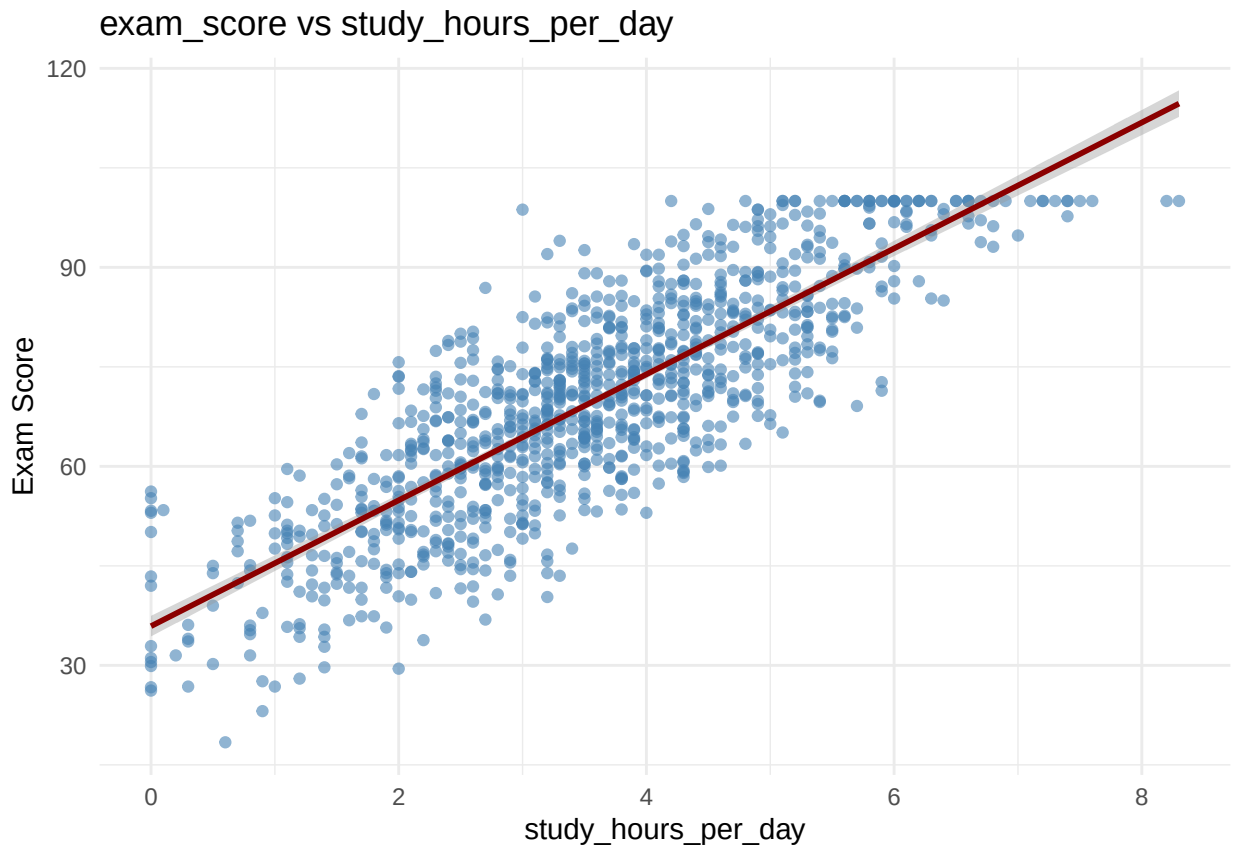
```
## `geom_smooth()` using formula = 'y ~ x'
```



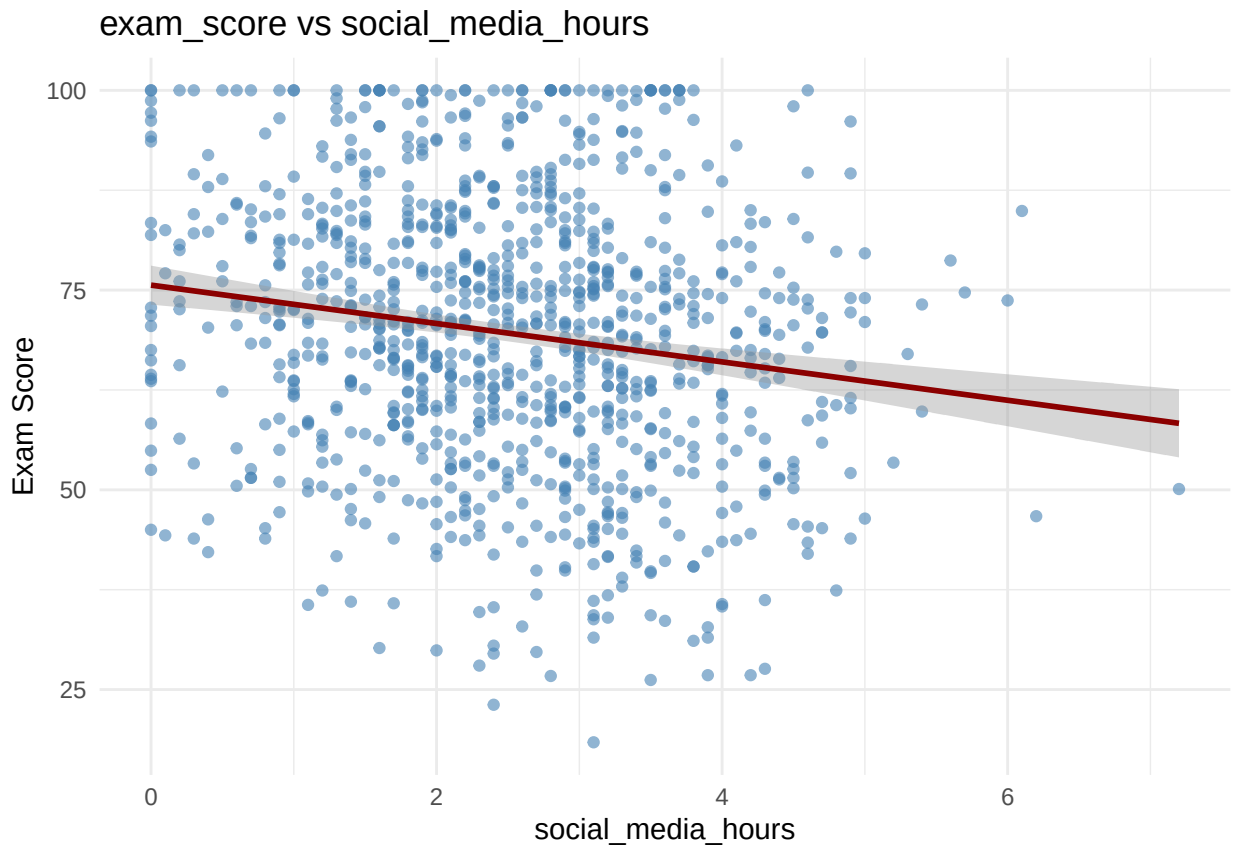

```
## `geom_smooth()` using formula = 'y ~ x'
```



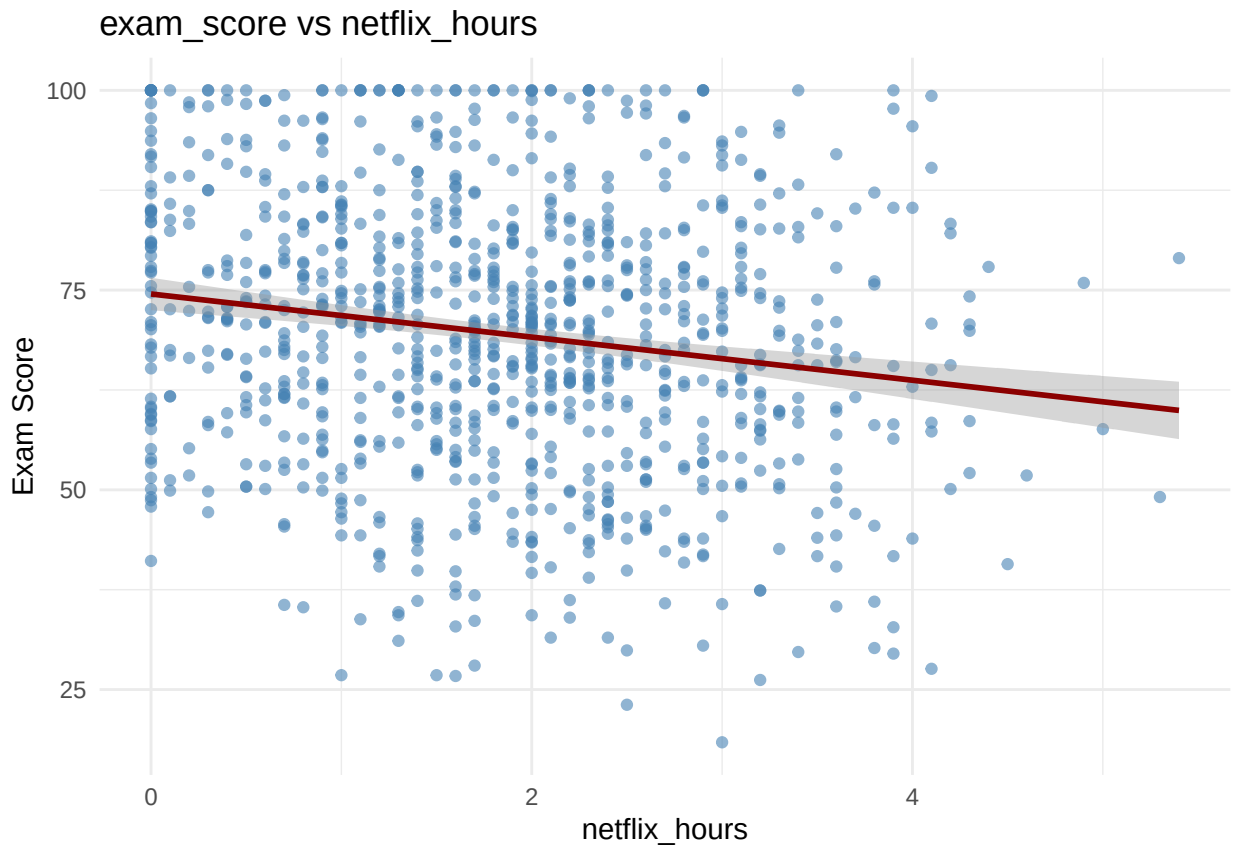

```
## `geom_smooth()` using formula = 'y ~ x'
```



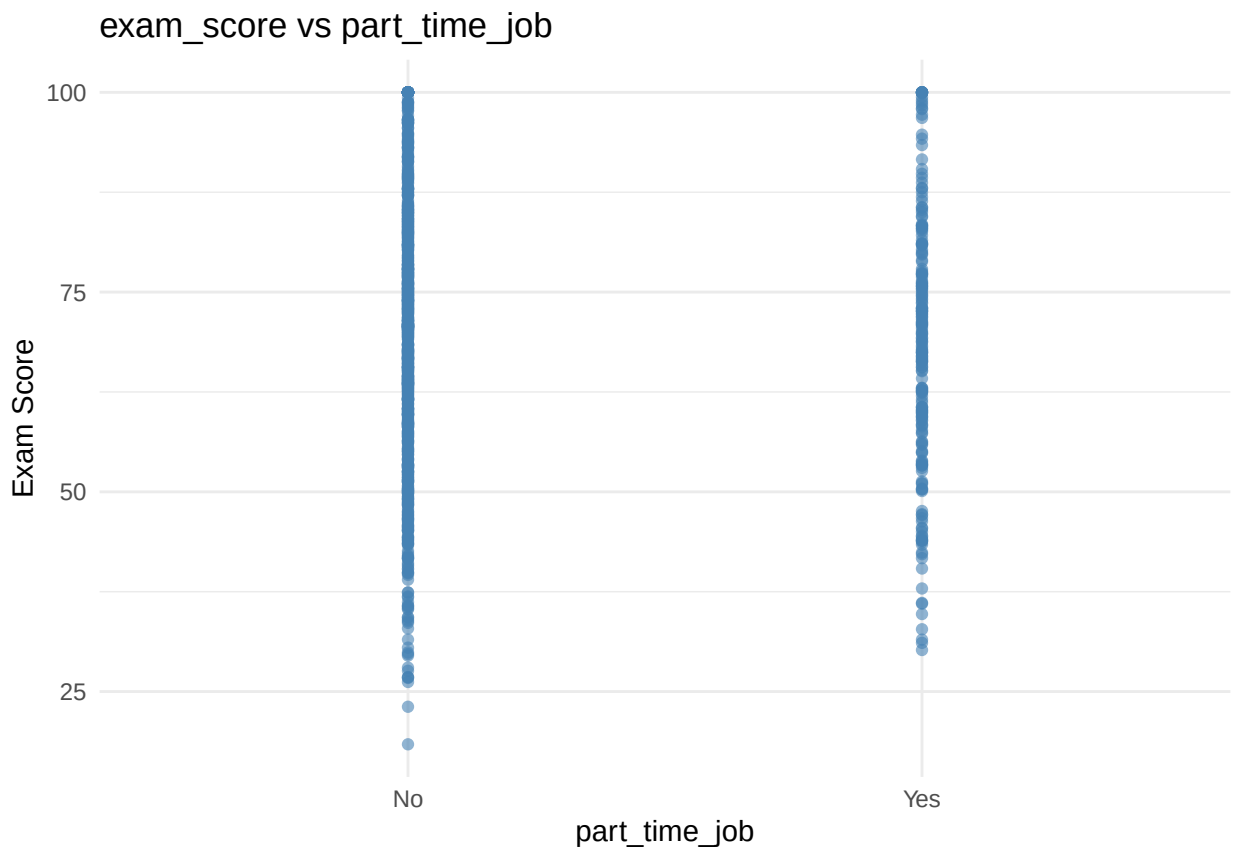

```
## `geom_smooth()` using formula = 'y ~ x'
```



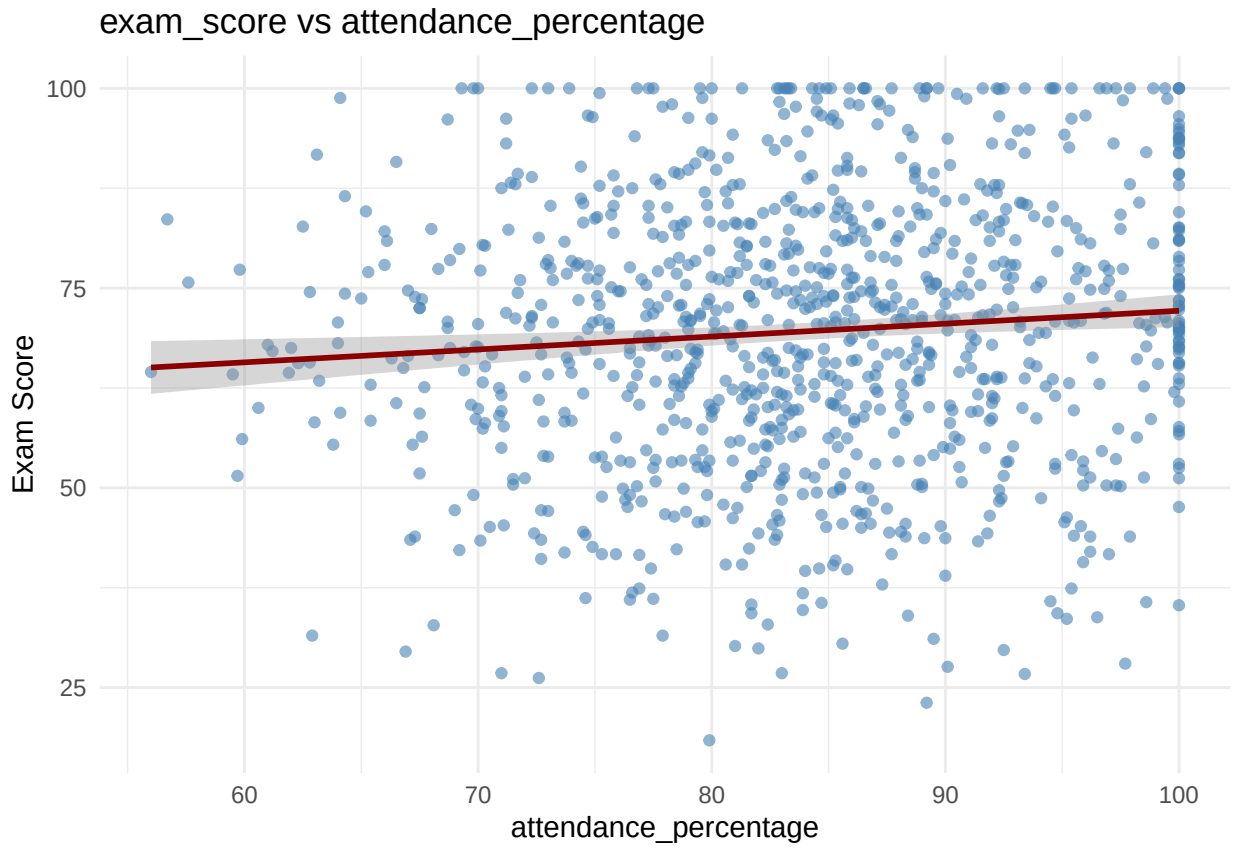

```
## `geom_smooth()` using formula = 'y ~ x'
```



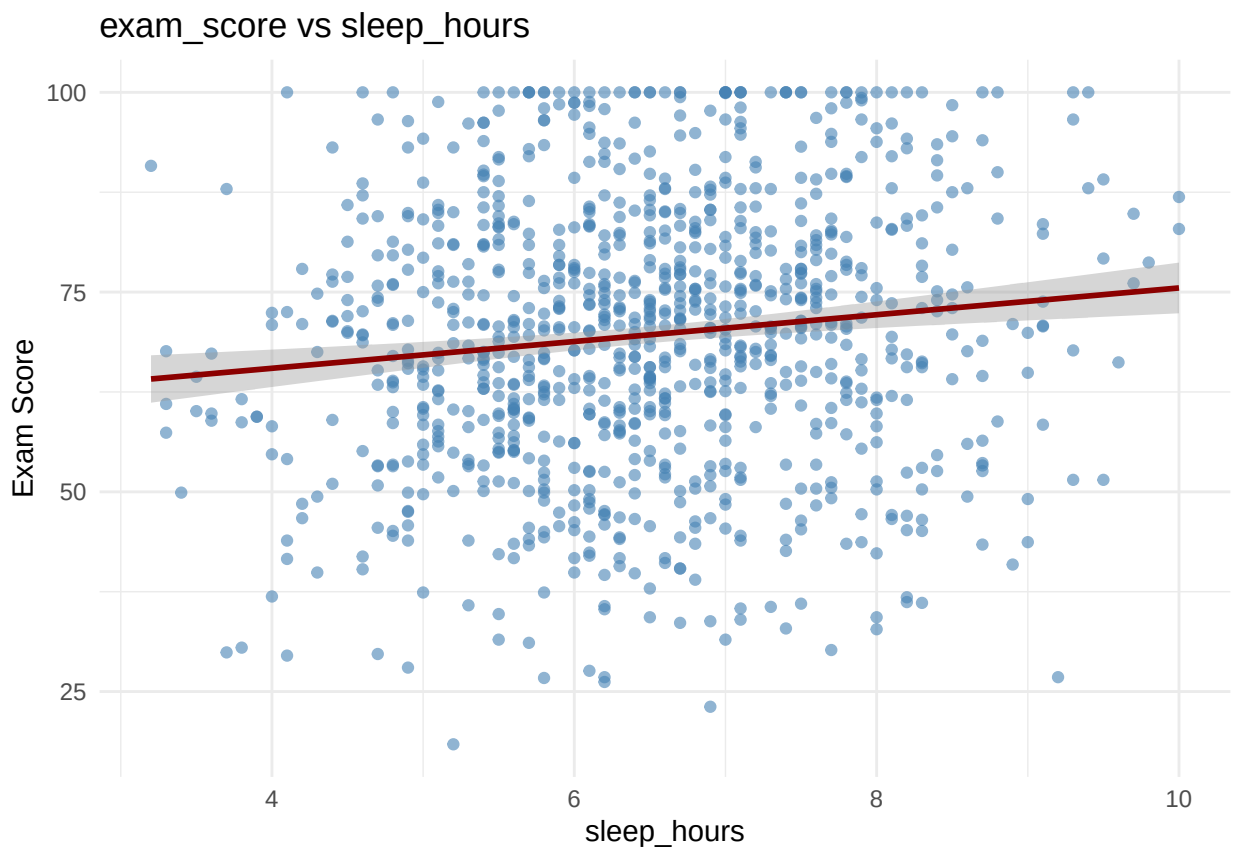

```
## `geom_smooth()` using formula = 'y ~ x'
```



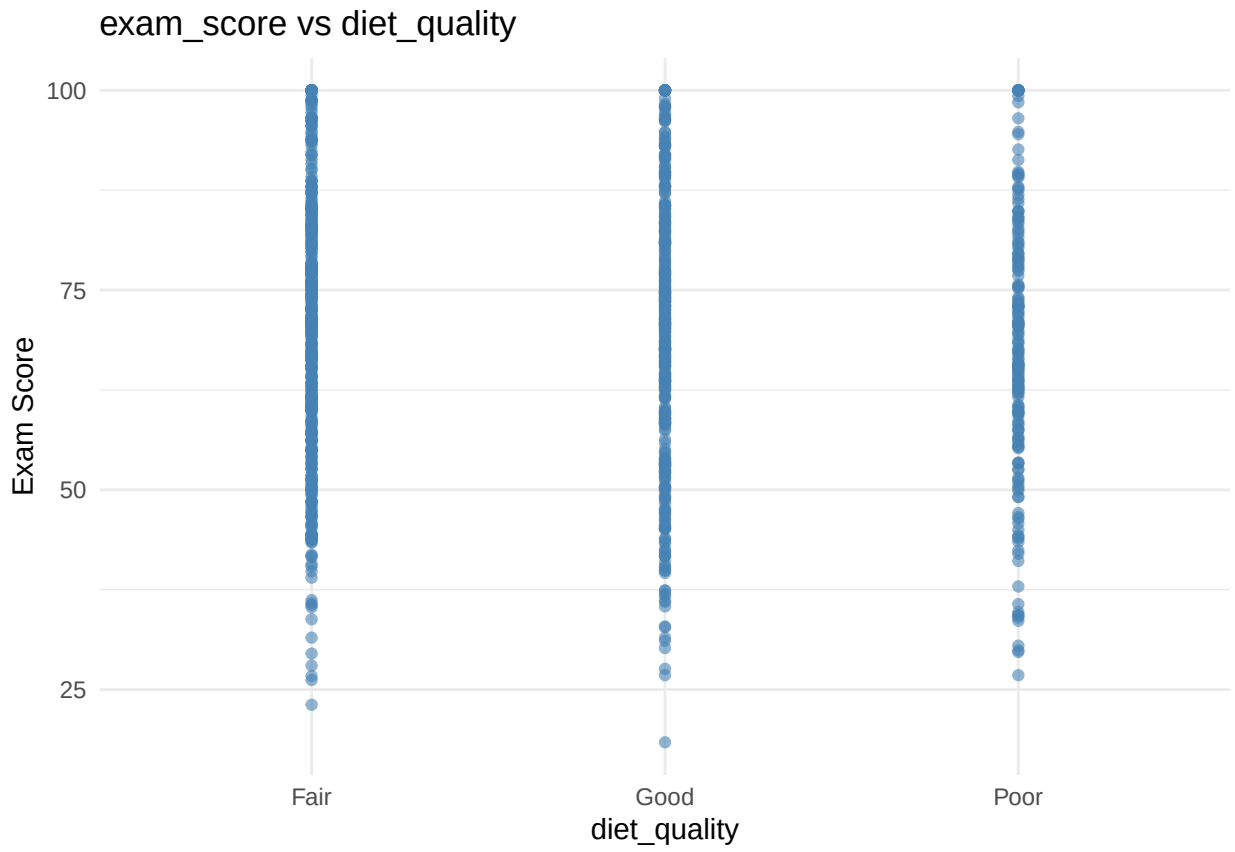

```
## `geom_smooth()` using formula = 'y ~ x'
```



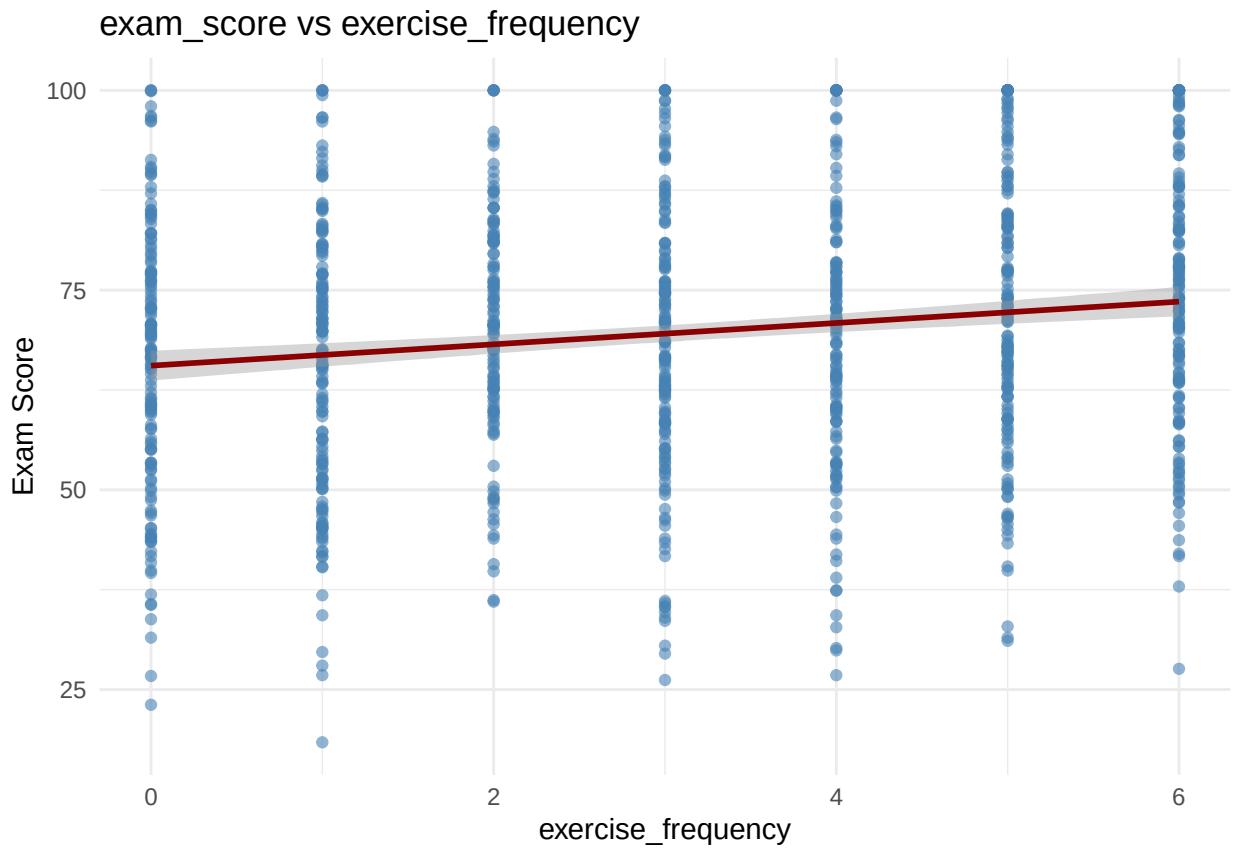

```
## `geom_smooth()` using formula = 'y ~ x'
```



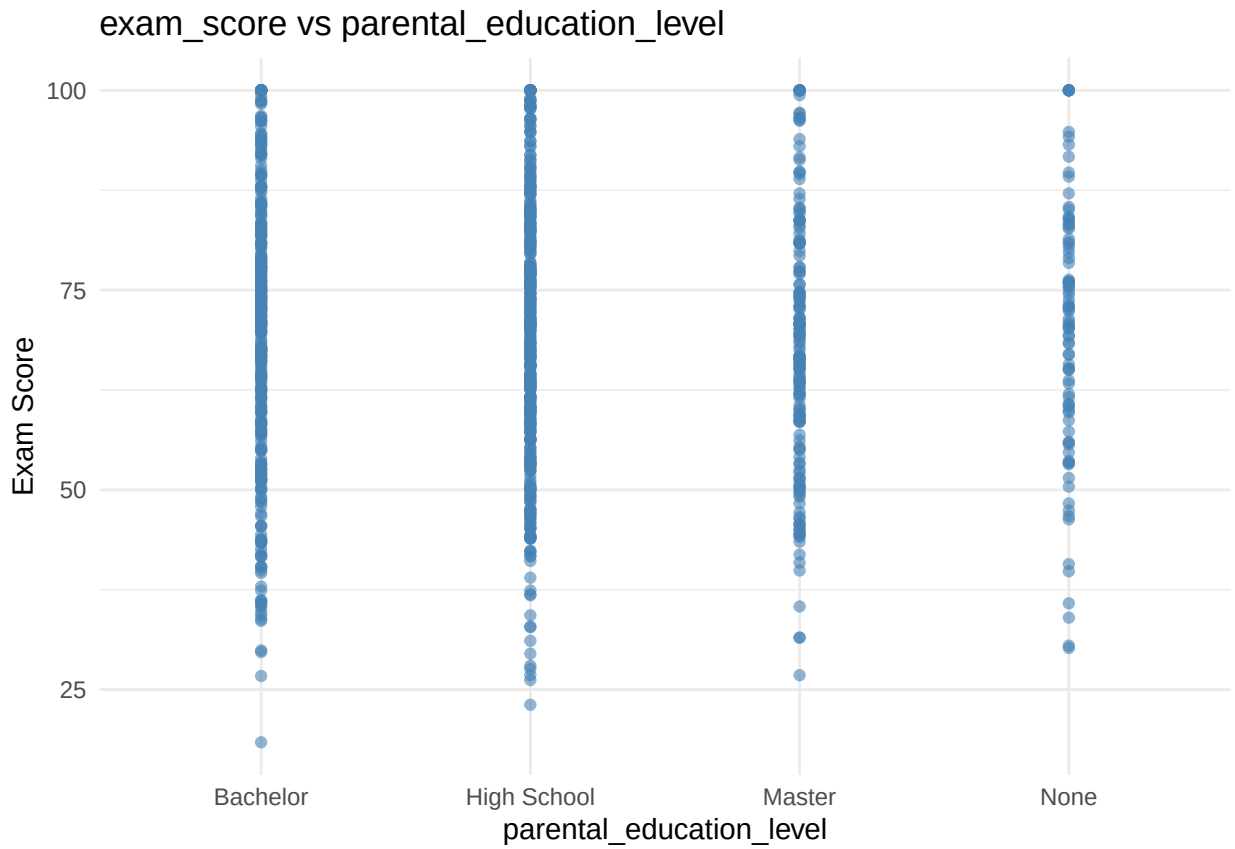

```
## `geom_smooth()` using formula = 'y ~ x'
```



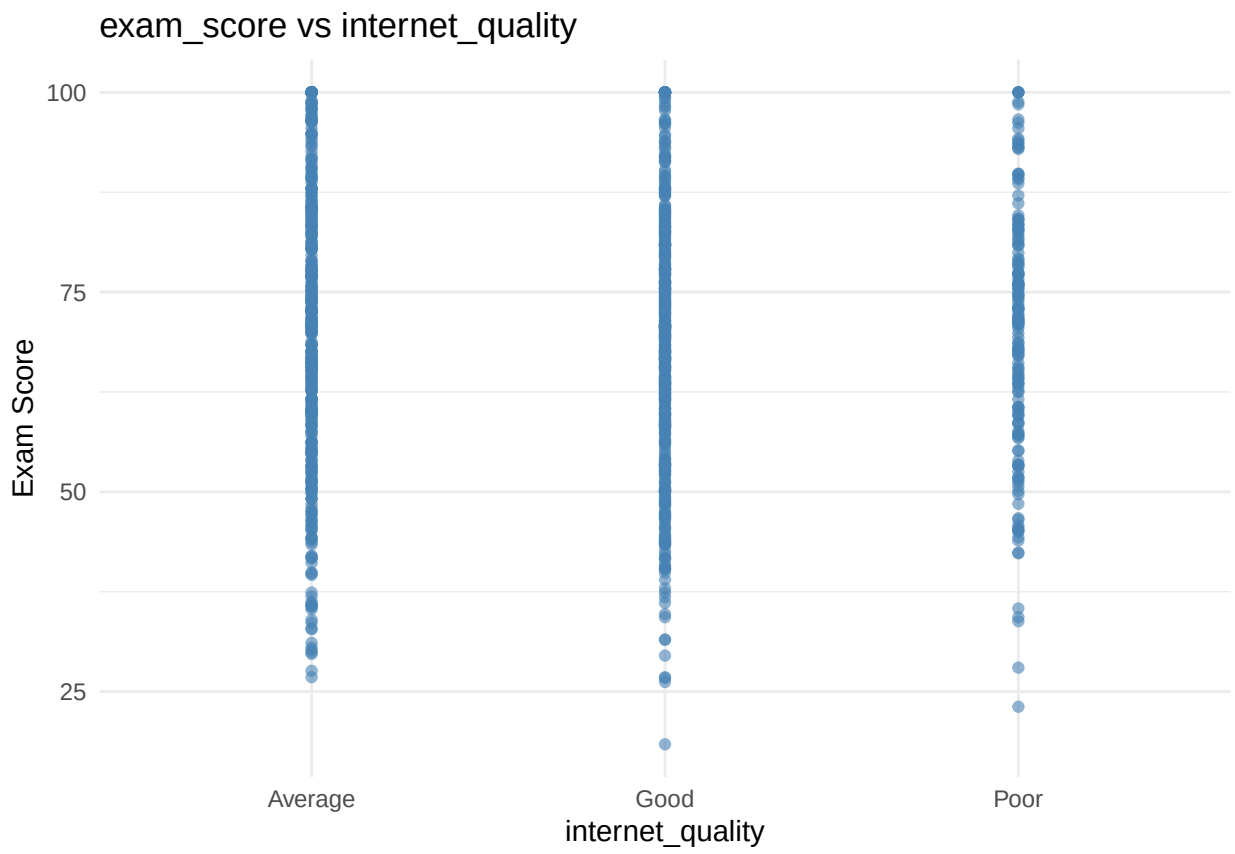

```
## `geom_smooth()` using formula = 'y ~ x'
```



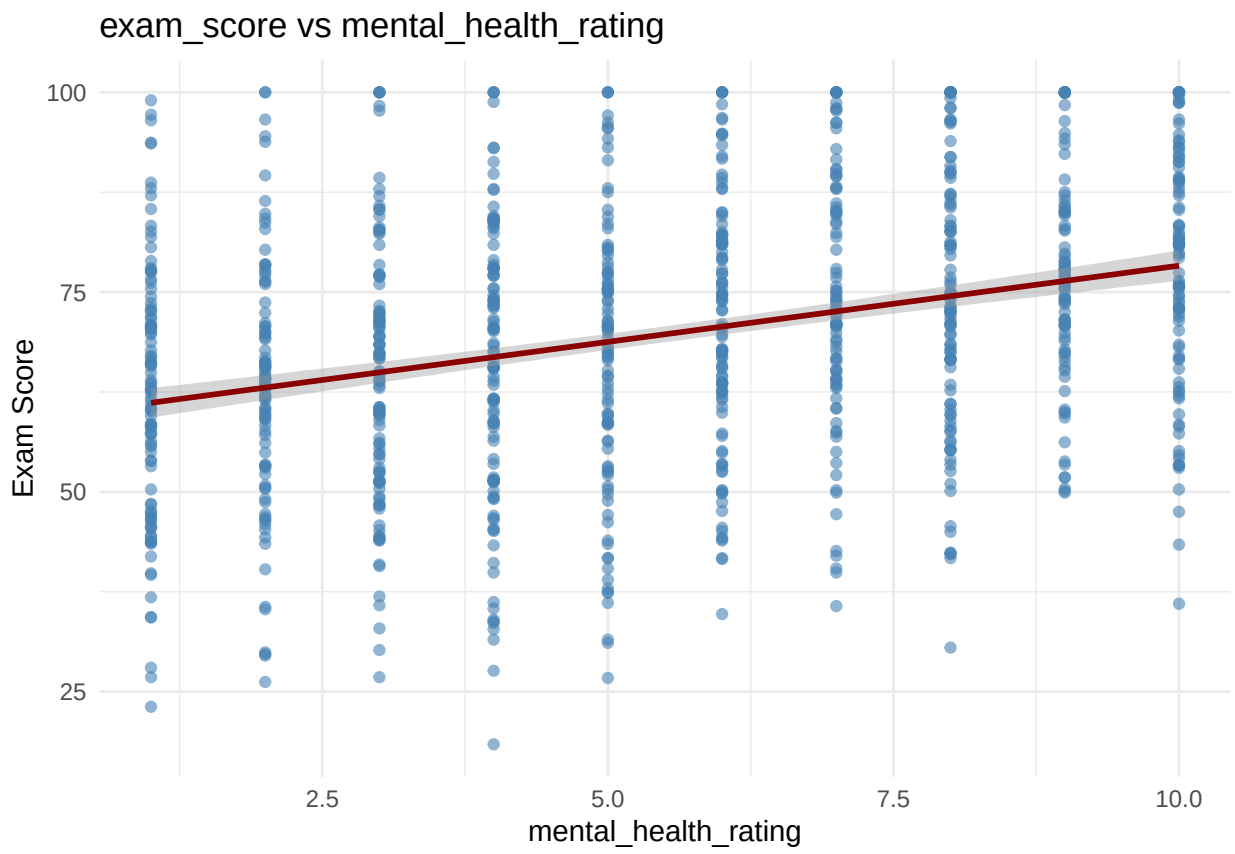

```
## `geom_smooth()` using formula = 'y ~ x'
```



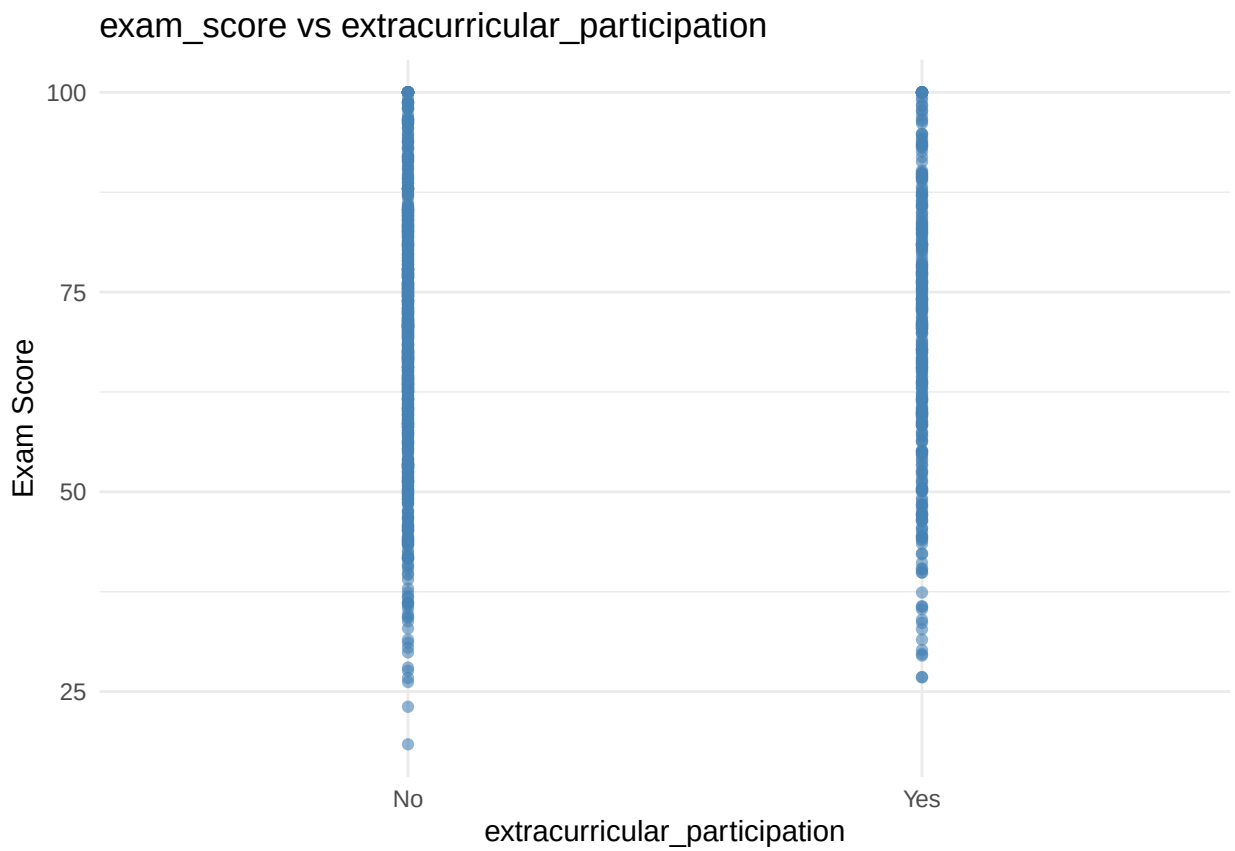

```
## `geom_smooth()` using formula = 'y ~ x'
```




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




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



2.4 Data Cleaning and Preprocessing

There is no missing values in the dataset and no further preprocessing at this step.

```
colSums(is.na(data))
```

```
##                    student_id                           age
##                             0                             0
##                        gender            study_hours_per_day
##                             0                             0
##             social_media_hours                  netflix_hours
##                             0                             0
##                  part_time_job         attendance_percentage
##                             0                             0
##                   sleep_hours                   diet_quality
##                             0                             0
##             exercise_frequency      parental_education_level
##                             0                             0
##               internet_quality          mental_health_rating
##                             0                             0
## extracurricular_participation                    exam_score
##                             0                             0
```

3. Regression Assumptions Verification

3.1 Independence of Observation

By the definition of this dataset, it is clear that each observation of this dataset is independent of others.

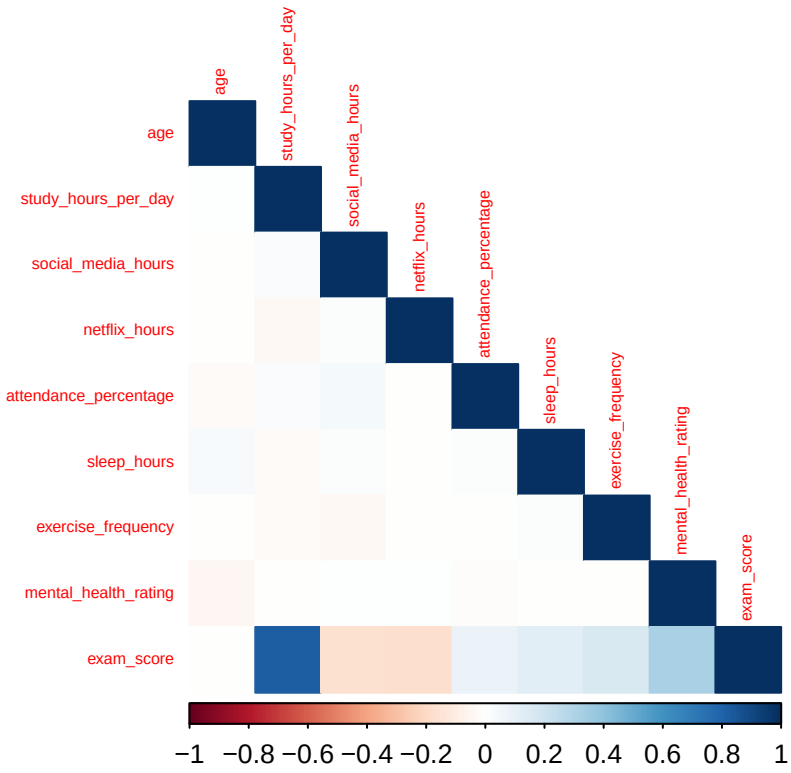
3.2 Multicollinearity

Before running the linear regression, we performed the Pearson correlation analysis first among the numerical predictors. And all values are between -0.05 and 0.05, indicating no strong correlations between any pairs of the tested predictors.

```
numeric_predictors <- data %>%
  select(where(is.numeric))

cor_matrix <- cor(numeric_predictors, use = "complete.obs", method = "pearson")
round(cor_matrix, 2)
```

```
##                         age study_hours_per_day social_media_hours
## age                    1.00                0.00              -0.01
## study_hours_per_day    0.00                1.00               0.02
## social_media_hours    -0.01                0.02               1.00
## netflix_hours          0.00               -0.03               0.01
## attendance_percentage -0.03                0.03               0.04
## sleep_hours            0.04               -0.03               0.02
## exercise_frequency     0.00               -0.03              -0.04
## mental_health_rating  -0.05                0.00               0.00
## exam_score            -0.01                0.83              -0.17
##                       netflix_hours attendance_percentage sleep_hours
## age                            0.00                -0.03        0.04
## study_hours_per_day           -0.03                 0.03       -0.03
## social_media_hours             0.01                 0.04        0.02
## netflix_hours                  1.00                 0.00        0.00
## attendance_percentage          0.00                 1.00        0.01
## sleep_hours                    0.00                 0.01        1.00
## exercise_frequency            -0.01                -0.01        0.02
## mental_health_rating           0.01                -0.02       -0.01
## exam_score                    -0.17                 0.09        0.12
##                       exercise_frequency mental_health_rating exam_score
## age                                 0.00                -0.05      -0.01
## study_hours_per_day                -0.03                 0.00       0.83
## social_media_hours                 -0.04                 0.00      -0.17
## netflix_hours                      -0.01                 0.01      -0.17
## attendance_percentage              -0.01                -0.02       0.09
## sleep_hours                         0.02                -0.01       0.12
## exercise_frequency                  1.00                 0.00       0.16
## mental_health_rating                0.00                 1.00       0.32
## exam_score                          0.16                 0.32       1.00
```

```
corrplot(cor_matrix, method = "color", type = "lower", tl.cex = 0.5)
```



3.3 Linearity & Homoscedasticity Assessment

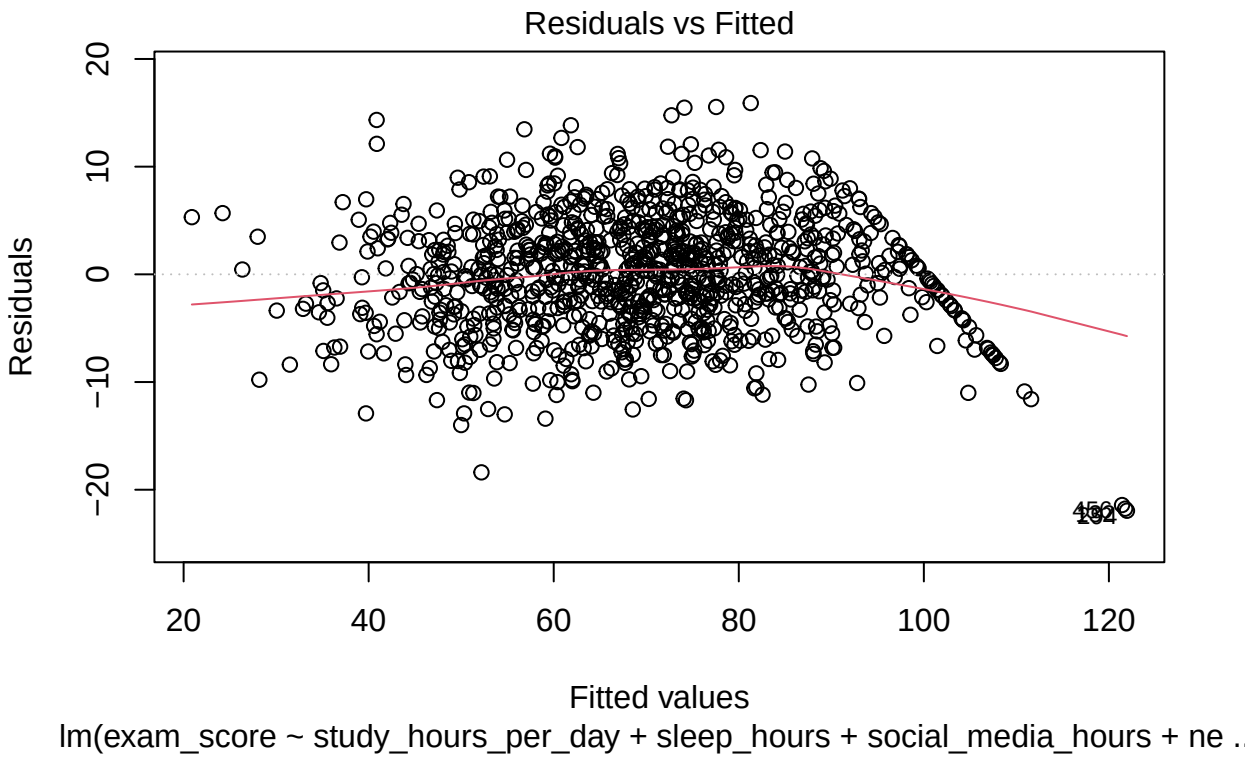
The plot of residuals vs. fitted is shown below. The red line shows a curved pattern, especially rising and then dropping on the right, suggesting that the relationship between predictors and outcome may not be fully linear. Also, the spread of residuals shrinks as fitted values increase. This indicates heteroscedasticity, violating the homoscedasticity assumption.

```
# Fit the linear model
model1 <- lm(exam_score ~ study_hours_per_day + sleep_hours + social_media_hours + netflix_hours + atte
summary(model1)
```

```
##
## Call:
## lm(formula = exam_score ~ study_hours_per_day + sleep_hours +
##     social_media_hours + netflix_hours + attendance_percentage +
##     exercise_frequency + mental_health_rating, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -21.9509  -3.3953  -0.0283   3.6680  15.9059
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            6.15722    1.89252   3.253  0.00118 **
## study_hours_per_day    9.57456    0.11503  83.238  < 2e-16 ***
```

```
## sleep_hours            2.00462    0.13764  14.564  < 2e-16 ***
## social_media_hours    -2.61978    0.14413 -18.177  < 2e-16 ***
## netflix_hours         -2.27708    0.15697 -14.507  < 2e-16 ***
## attendance_percentage  0.14473    0.01797   8.054 2.28e-15 ***
## exercise_frequency     1.45187    0.08338  17.413  < 2e-16 ***
## mental_health_rating   1.94891    0.05924  32.897  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.331 on 992 degrees of freedom
## Multiple R-squared:  0.9011, Adjusted R-squared:  0.9004
## F-statistic:  1291 on 7 and 992 DF,  p-value: < 2.2e-16
```

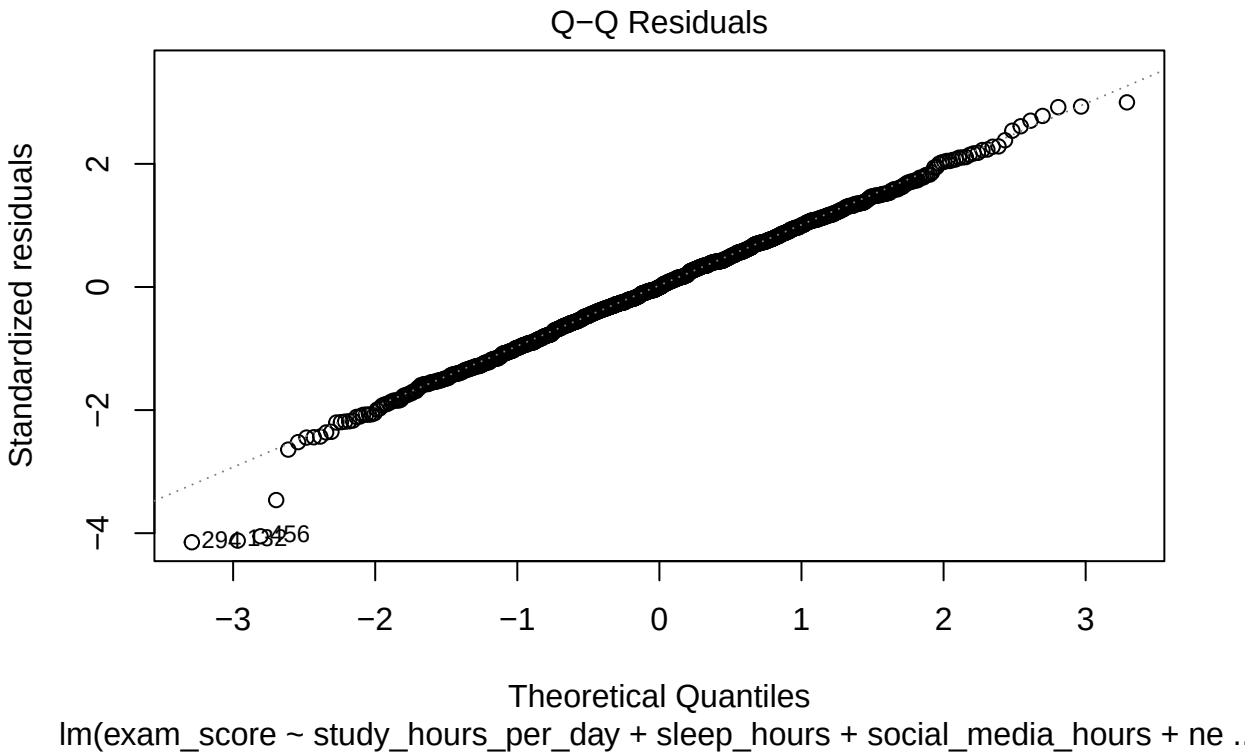
```
# Residuals vs Fitted
plot(model1, which = 1)
```



3.4 Normality of Residuals

The QQ plot is given below. The residuals follow the diagonal line closely with only slight deviation at the tails. Although a few outliers exist on both ends, the normality assumption is reasonably satisfied.

```
# Normal Q-Q plot
plot(model1, which = 2)
```



4. Assumption Violation Handling

4.1 Polynomial Transformation

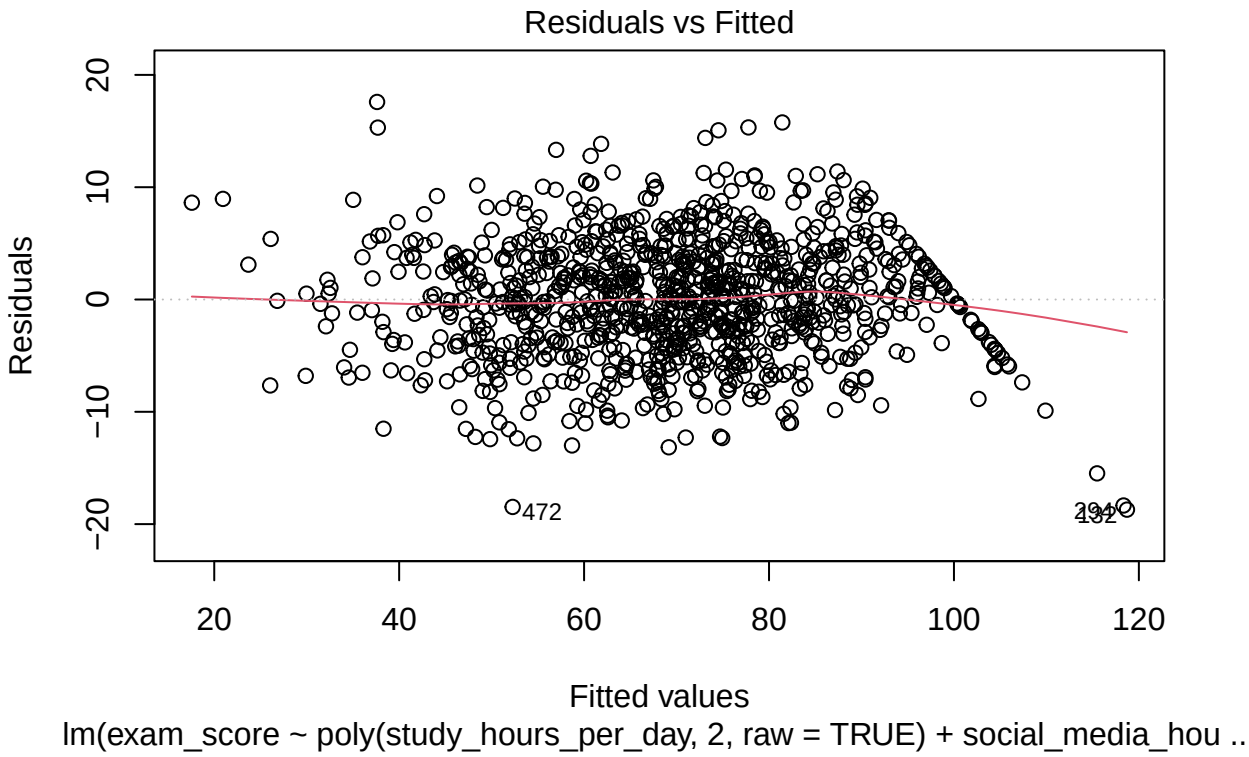
Based the discussion from the last section, we need to handle the violated assumptions (linearity and heteroscedasticity) for the current model. study_hours_per_day shows a strong curved trend in the previous plot, so we will use a 2nd-degree polynomial for this variable to improve the model linearity. The red trend line is now flatter across the middle range of fitted values, indicating that the previously observed curved trend has been partially corrected.

```
model2 <- lm(exam_score ~ poly(study_hours_per_day, 2, raw = TRUE) +
                              social_media_hours +
                              sleep_hours +
                              netflix_hours +
                              attendance_percentage +
                              exercise_frequency +
                              mental_health_rating,
                 data = data)
summary(model2)
```

```
##
## Call:
## lm(formula = exam_score ~ poly(study_hours_per_day, 2, raw = TRUE) +
##     social_media_hours + sleep_hours + netflix_hours + attendance_percentage +
```

```
##     exercise_frequency + mental_health_rating, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.7105  -3.5752   0.1022   3.5642  17.5902
##
## Coefficients:
##                                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)                              2.15516    2.00724   1.074    0.283
## poly(study_hours_per_day, 2, raw = TRUE)1 11.73782    0.41545  28.253  < 2e-16
## poly(study_hours_per_day, 2, raw = TRUE)2 -0.30132    0.05567  -5.413 7.79e-08
## social_media_hours                       -2.60421    0.14215 -18.321  < 2e-16
## sleep_hours                               2.01479    0.13573  14.844  < 2e-16
## netflix_hours                            -2.27694    0.15478 -14.711  < 2e-16
## attendance_percentage                     0.15145    0.01776   8.527  < 2e-16
## exercise_frequency                        1.47658    0.08234  17.932  < 2e-16
## mental_health_rating                      1.95323    0.05842  33.434  < 2e-16
##
## (Intercept)
## poly(study_hours_per_day, 2, raw = TRUE)1 ***
## poly(study_hours_per_day, 2, raw = TRUE)2 ***
## social_media_hours                       ***
## sleep_hours                              ***
## netflix_hours                            ***
## attendance_percentage                    ***
## exercise_frequency                       ***
## mental_health_rating                     ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.256 on 991 degrees of freedom
## Multiple R-squared:  0.9039, Adjusted R-squared:  0.9031
## F-statistic:  1165 on 8 and 991 DF,  p-value: < 2.2e-16
```

```
plot(model2, which=1)
```



4.2 Robust Standard Error

With the new model, we ran the Breusch-Pagan test first, and got the p-value of $0.02 < 0.05$. Therefore, we reject the null hypothesis that residuals have constant variance .

To handle the heteroscedasticity, we applied robust standard errors for valid inference in this model. While the coefficient estimates remained unchanged, the robust standard errors provide more reliable inference. All predictors remained statistically significant, confirming the robustness of the model's findings.

```
# Breusch-Pagan Test
bptest(model2)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model2
## BP = 17.768, df = 8, p-value = 0.02303
```

```
# Replace the default with the robust standard errors
coeftest(model2, vcov. = vcovHC(model2, type = "HC3"))
```

```
##
## t test of coefficients:
##
```

```
##                                          Estimate Std. Error  t value
## (Intercept)                              2.155155   2.129849   1.0119
## poly(study_hours_per_day, 2, raw = TRUE)1 11.737818   0.546783  21.4670
## poly(study_hours_per_day, 2, raw = TRUE)2 -0.301320   0.076816  -3.9226
## social_media_hours                       -2.604208   0.145504 -17.8979
## sleep_hours                               2.014790   0.137793  14.6219
## netflix_hours                            -2.276944   0.154537 -14.7340
## attendance_percentage                     0.151452   0.018307   8.2728
## exercise_frequency                        1.476580   0.082990  17.7923
## mental_health_rating                      1.953229   0.060739  32.1577
##                                          Pr(>|t|)
## (Intercept)                                0.3118
## poly(study_hours_per_day, 2, raw = TRUE)1 < 2.2e-16 ***
## poly(study_hours_per_day, 2, raw = TRUE)2 9.363e-05 ***
## social_media_hours                       < 2.2e-16 ***
## sleep_hours                              < 2.2e-16 ***
## netflix_hours                            < 2.2e-16 ***
## attendance_percentage                    4.187e-16 ***
## exercise_frequency                       < 2.2e-16 ***
## mental_health_rating                     < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

4.3 Model Comparison

We compared two linear regression models to predict exam scores: one assuming a linear relationship with study hours and another including a 2nd-degree polynomial term to capture non-linearity. The polynomial model showed a slight improvement in fit, with a lower residual standard error (5.256 vs. 5.331) and a higher adjusted R-squared (0.9031 vs. 0.9004). The squared term was statistically significant ($p < 0.001$), indicating diminishing returns to study time.

5. Variable Selection & Hypothesis Testing

5.1 Branch and Bound based on BIC

Now, we have the regression model with 7 predictors. But the contribution from each variable is significantly different. We want to select less variables to have better model generalizibility as well as reduce the risk of overfitting.

The first variable selection method applied was branch and bound based on BIC. The results showed that the best-performing subset included all 8 predictors, including the polynomial term for study hours. This suggests that each of the current variables contributes meaningfully to explaining variation in exam scores, and removing any one of them would result in a less optimal model. The adjusted R-squared is 0.9031.

```
# Include the square term
data$study_hours_squared <- data$study_hours_per_day^2

subset_data <- data %>%
  select(exam_score,
         study_hours_per_day,
         study_hours_squared,
         sleep_hours,
         social_media_hours,
```

```r
        netflix_hours,
        attendance_percentage,
        exercise_frequency,
        mental_health_rating)

# Perform Branch and Bound
model3 <- regsubsets(exam_score ~ ., data = subset_data, nvmax = 9, method = "exhaustive")
summary(model3)
```

```
## Subset selection object
## Call: regsubsets.formula(exam_score ~ ., data = subset_data, nvmax = 9,
##     method = "exhaustive")
## 8 Variables  (and intercept)
##                       Forced in Forced out
## study_hours_per_day       FALSE      FALSE
## study_hours_squared       FALSE      FALSE
## sleep_hours               FALSE      FALSE
## social_media_hours        FALSE      FALSE
## netflix_hours             FALSE      FALSE
## attendance_percentage     FALSE      FALSE
## exercise_frequency        FALSE      FALSE
## mental_health_rating      FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          study_hours_per_day study_hours_squared sleep_hours social_media_hours
## 1  ( 1 ) "*"                 " "                 " "         " "
## 2  ( 1 ) "*"                 " "                 " "         " "
## 3  ( 1 ) "*"                 " "                 " "         "*"
## 4  ( 1 ) "*"                 " "                 " "         "*"
## 5  ( 1 ) "*"                 " "                 "*"         "*"
## 6  ( 1 ) "*"                 " "                 "*"         "*"
## 7  ( 1 ) "*"                 " "                 "*"         "*"
## 8  ( 1 ) "*"                 "*"                 "*"         "*"
##          netflix_hours attendance_percentage exercise_frequency
## 1  ( 1 ) " "           " "                   " "
## 2  ( 1 ) " "           " "                   " "
## 3  ( 1 ) " "           " "                   " "
## 4  ( 1 ) " "           " "                   "*"
## 5  ( 1 ) " "           " "                   "*"
## 6  ( 1 ) "*"           " "                   "*"
## 7  ( 1 ) "*"           "*"                   "*"
## 8  ( 1 ) "*"           "*"                   "*"
##          mental_health_rating
## 1  ( 1 ) " "
## 2  ( 1 ) "*"
## 3  ( 1 ) "*"
## 4  ( 1 ) "*"
## 5  ( 1 ) "*"
## 6  ( 1 ) "*"
## 7  ( 1 ) "*"
## 8  ( 1 ) "*"
```

```r
sout <- summary(model3)
print(sout$bic)
```

```
## [1] -1129.739 -1524.350 -1690.329 -1876.131 -2026.674 -2201.653 -2258.090
## [8] -2280.317
```

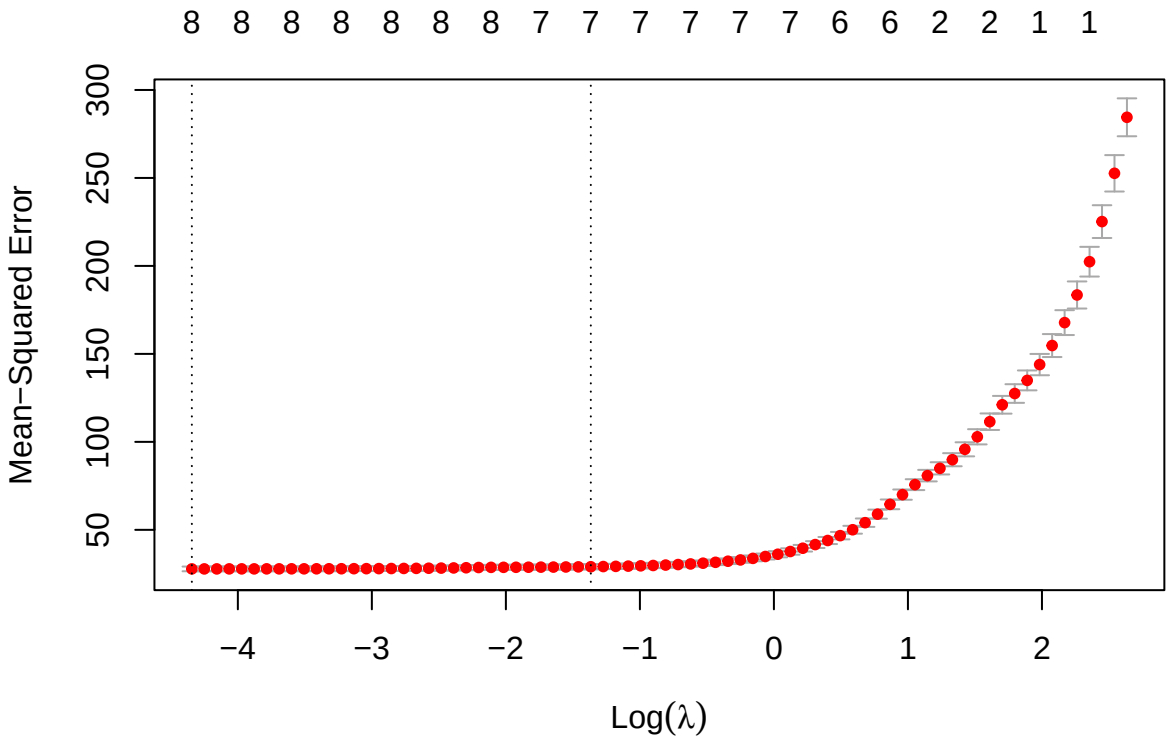
```r
which.min(sout$bic)
```

```
## [1] 8
```

5.2 LASSO Variable Selection

The second variable selection method used was LASSO regression. Based on cross-validation results, the model with the lowest prediction error included all eight predictors, including the polynomial term for study hours. A more parsimonious model selected using the 1-SE rule excluded the polynomial term and achieved a similar RMSE. However, as discussed in Chapter 2, omitting the polynomial term violates the assumption of linearity. Therefore, we retain all seven predictors along with the polynomial term in the final model to ensure both predictive performance and theoretical consistency.

```r
# Define predictor matrix and response
x <- model.matrix(exam_score ~ study_hours_per_day + I(study_hours_per_day^2) + sleep_hours + social_me
y <- data$exam_score

# Cross-validated LASSO
set.seed(123)
lasso_cv <- cv.glmnet(x, y, alpha = 1, standardize = TRUE)
plot(lasso_cv)
```




```
model4 <- glmnet(x, y, alpha = 1, lambda = lasso_cv$lambda.min)
coef(model4)
```

```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##                                    s0
## (Intercept)                 2.8726632
## study_hours_per_day        11.4605898
## I(study_hours_per_day^2)   -0.2643726
## sleep_hours                 2.0026366
## social_media_hours         -2.5945689
## netflix_hours              -2.2655356
## attendance_percentage       0.1492196
## exercise_frequency          1.4672486
## mental_health_rating        1.9479502
```

```
model5 <- glmnet(x, y, alpha = 1, lambda = lasso_cv$lambda.1se)
coef(model5)
```

```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##                                    s0
## (Intercept)                10.4083299
## study_hours_per_day         9.3965702
## I(study_hours_per_day^2)    .
## sleep_hours                 1.7923920
## social_media_hours         -2.3944452
```

```
## netflix_hours            -2.0503410
## attendance_percentage     0.1168679
## exercise_frequency        1.3293040
## mental_health_rating      1.8557737
```

5.3 Cross-Validation

We fit a linear regression model using all 8 predictors. All variables were statistically significant at the 0.001 level. The model achieved an adjusted R^2 of approximately 0.903, indicating that it explains over 90% of the variance in the response. To assess generalization performance, we performed 10-fold cross-validation. The cross-validated mean squared error (MSE) was 27.97, corresponding to a root mean squared error (RMSE) of 5.29. This suggests that the model's predictions deviate from actual exam scores by about 5.3 points on average, demonstrating both high accuracy and reliability.

```
model_cv <- glm(exam_score ~ ., data = subset_data)
summary(model_cv)
```

```
##
## Call:
## glm(formula = exam_score ~ ., data = subset_data)
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              2.15516    2.00724   1.074    0.283
## study_hours_per_day     11.73782    0.41545  28.253  < 2e-16 ***
## study_hours_squared     -0.30132    0.05567  -5.413 7.79e-08 ***
## sleep_hours              2.01479    0.13573  14.844  < 2e-16 ***
## social_media_hours      -2.60421    0.14215 -18.321  < 2e-16 ***
## netflix_hours           -2.27694    0.15478 -14.711  < 2e-16 ***
## attendance_percentage    0.15145    0.01776   8.527  < 2e-16 ***
## exercise_frequency       1.47658    0.08234  17.932  < 2e-16 ***
## mental_health_rating     1.95323    0.05842  33.434  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 27.62792)
##
##     Null deviance: 284938  on 999  degrees of freedom
## Residual deviance:  27379  on 991  degrees of freedom
## AIC: 6167.7
##
## Number of Fisher Scoring iterations: 2
```

```
err_cv <- cv.glm(subset_data, model_cv, K=10)$delta[1]
err_cv
```

```
## [1] 27.78616
```

```
rmse_cv <- sqrt(err_cv)
rmse_cv
```

```
## [1] 5.271258
```

6. Feature Impact Analysis

6.1 CI for Significant Coefficients

```
coefci(model_cv, level = .95, vcov. = vcovHC(model_cv, type = "HC3"))
```

```
##                              2.5 %     97.5 %
## (Intercept)           -2.0192721  6.3295829
## study_hours_per_day   10.6661432 12.8094934
## study_hours_squared   -0.4518768 -0.1507626
## sleep_hours            1.7447212  2.2848587
## social_media_hours    -2.8893904 -2.3190257
## netflix_hours         -2.5798307 -1.9740570
## attendance_percentage  0.1155703  0.1873331
## exercise_frequency     1.3139230  1.6392364
## mental_health_rating   1.8341826  2.0722757
```

6.2 Interpretation of Features

The final regression model included 8 predictors, all of which were statistically significant at the 0.001 level. The coefficient for study_hours_per_day was 11.74, indicating that each additional hour of study is associated with an average increase of 11.74 points in exam score, though this effect is tempered by the negative squared term (–0.30), reflecting diminishing returns at higher study durations. Lifestyle factors also showed meaningful impacts: each hour of sleep was associated with a 2.01-point increase, while each hour of social_media or Netflix use corresponded to 2.60 and 2.28-point decreases, respectively. Attendance_percentage, exercise_frequency, and mental_health_rating all had positive effects, with coefficients of 0.15, 1.48, and 1.95, respectively.

These results highlight that productive study habits and well-being factors such as sleep, mental health, and physical activity are positively related to academic performance, while excessive screen time has a negative impact.

6.3 Future Work

While this study effectively identified key lifestyle factors associated with academic performance using linear regression, future work could explore nonlinear or interaction effects in greater depth using more flexible models such as generalized additive models (GAMs) or tree-based methods like random forests.