Data Interpretation

Coraline Zhu

2025-10-13

```
library(readr)
data <-read_csv("Desktop/PSTAT197A_ Standardized_Proficiency_Score.csv")
```

```
## Rows: 51 Columns: 32
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): Num_Of_UpperDiv
## dbl (31): Prog_Level, Prog_Comfort, Prog_Proficiency, Math_Level, Math_Comfo...
##
## 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.
```

```
# Basic summaries
summary(data$Score_Avg)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.600   2.020   2.140   2.128   2.230   2.480
```

```
summary(data[, c("Math_Proficiency", "Prog_Proficiency", "Stat_Proficiency")])
```

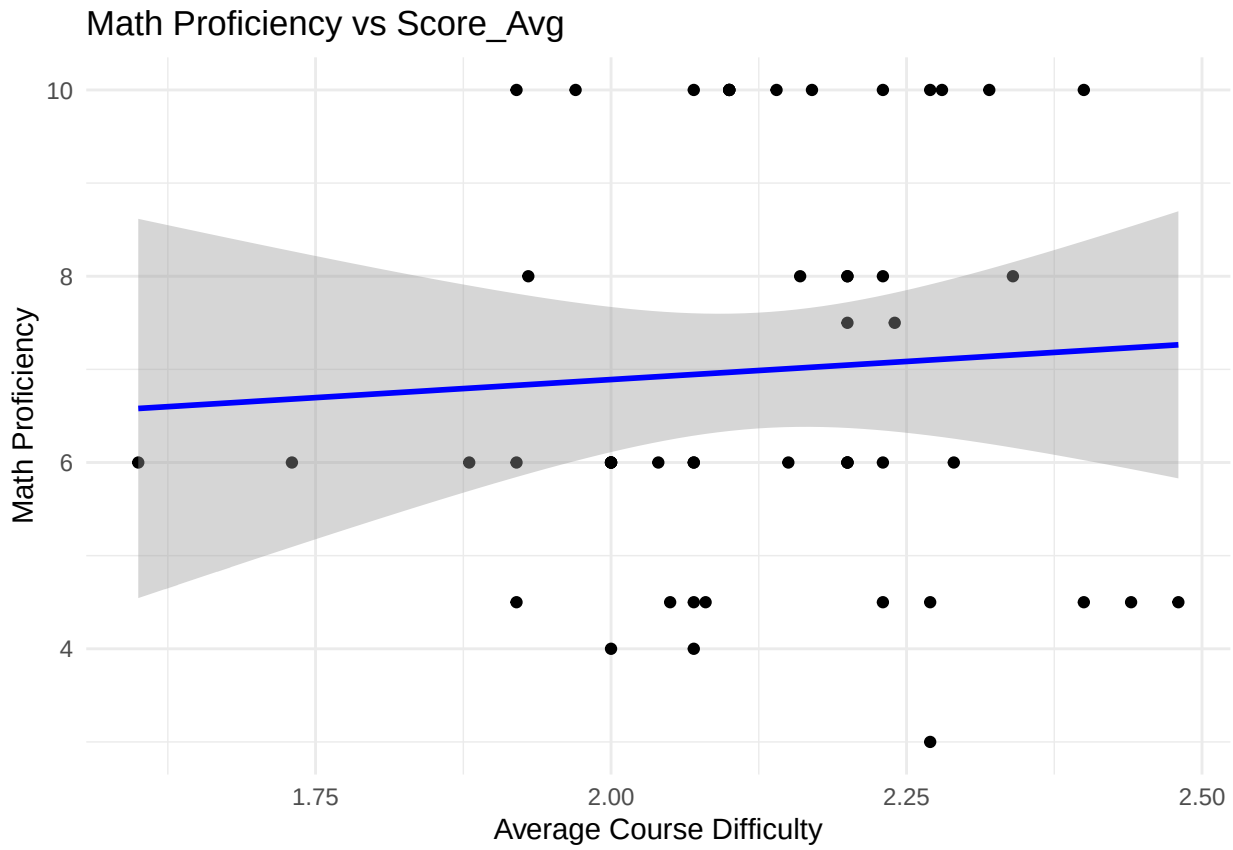
```
##  Math_Proficiency Prog_Proficiency Stat_Proficiency
##  Min.   : 3.00    Min.   : 3.000   Min.   : 2.000
##  1st Qu.: 6.00    1st Qu.: 5.250   1st Qu.: 6.000
##  Median : 6.00    Median : 6.000   Median : 8.000
##  Mean   : 6.99    Mean   : 6.559   Mean   : 7.392
##  3rd Qu.:10.00    3rd Qu.: 8.000   3rd Qu.:10.000
##  Max.   :10.00    Max.   :10.000   Max.   :10.000
```

```
# Correlation matrix
cor(data[, c("Score_Avg", "Math_Proficiency", "Prog_Proficiency", "Stat_Proficiency")])
```

```
##                   Score_Avg Math_Proficiency Prog_Proficiency Stat_Proficiency
## Score_Avg         1.00000000       0.06060925        0.1618892        0.3953879
## Math_Proficiency 0.06060925       1.00000000        0.2172544        0.6149421
## Prog_Proficiency 0.16188923       0.21725436        1.0000000        0.4087050
## Stat_Proficiency 0.39538785       0.61494213        0.4087050        1.0000000
```

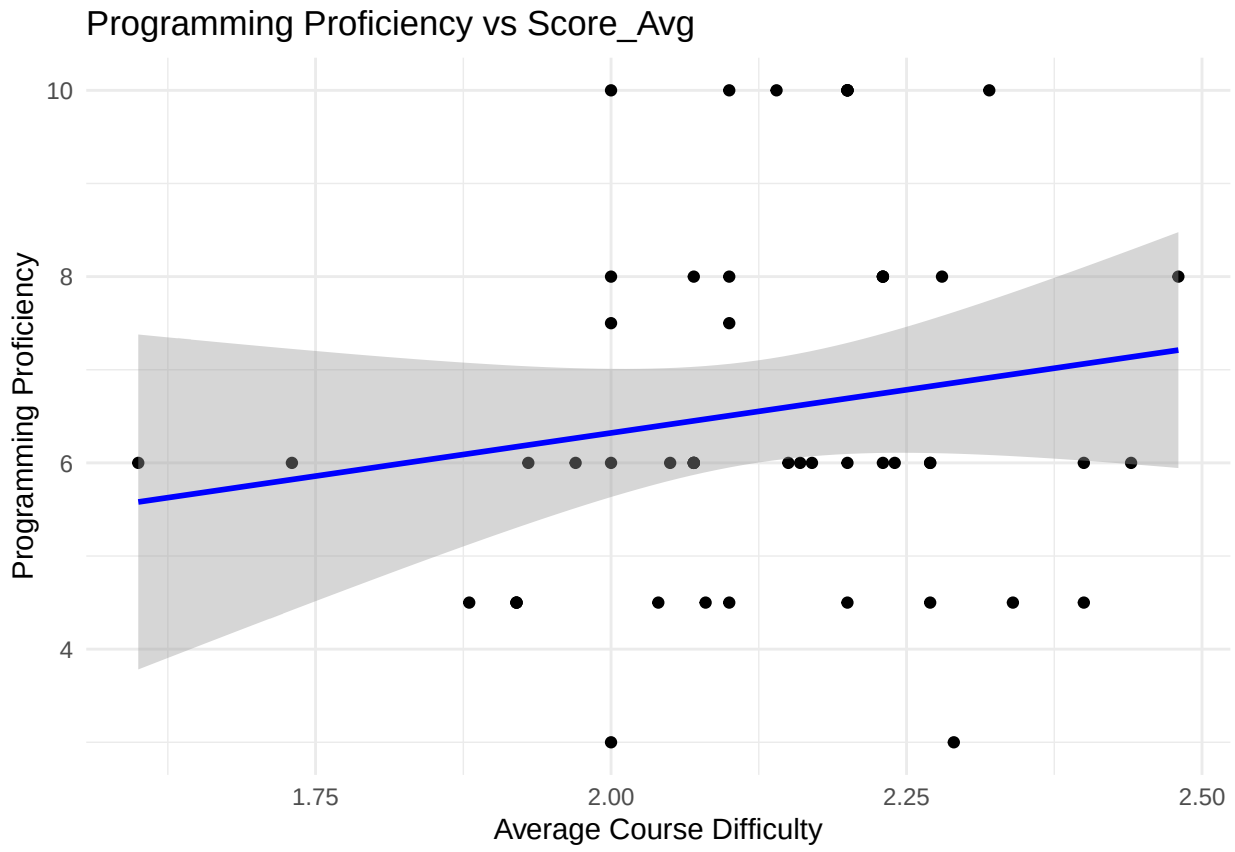
```r
library(ggplot2)

# Math Proficiency
ggplot(data, aes(x = Score_Avg, y = Math_Proficiency)) +
  geom_point() +
  geom_smooth(method = "lm", color = "blue") +
  labs(title = "Math Proficiency vs Score_Avg",
       x = "Average Course Difficulty",
       y = "Math Proficiency") +
  theme_minimal()
```

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



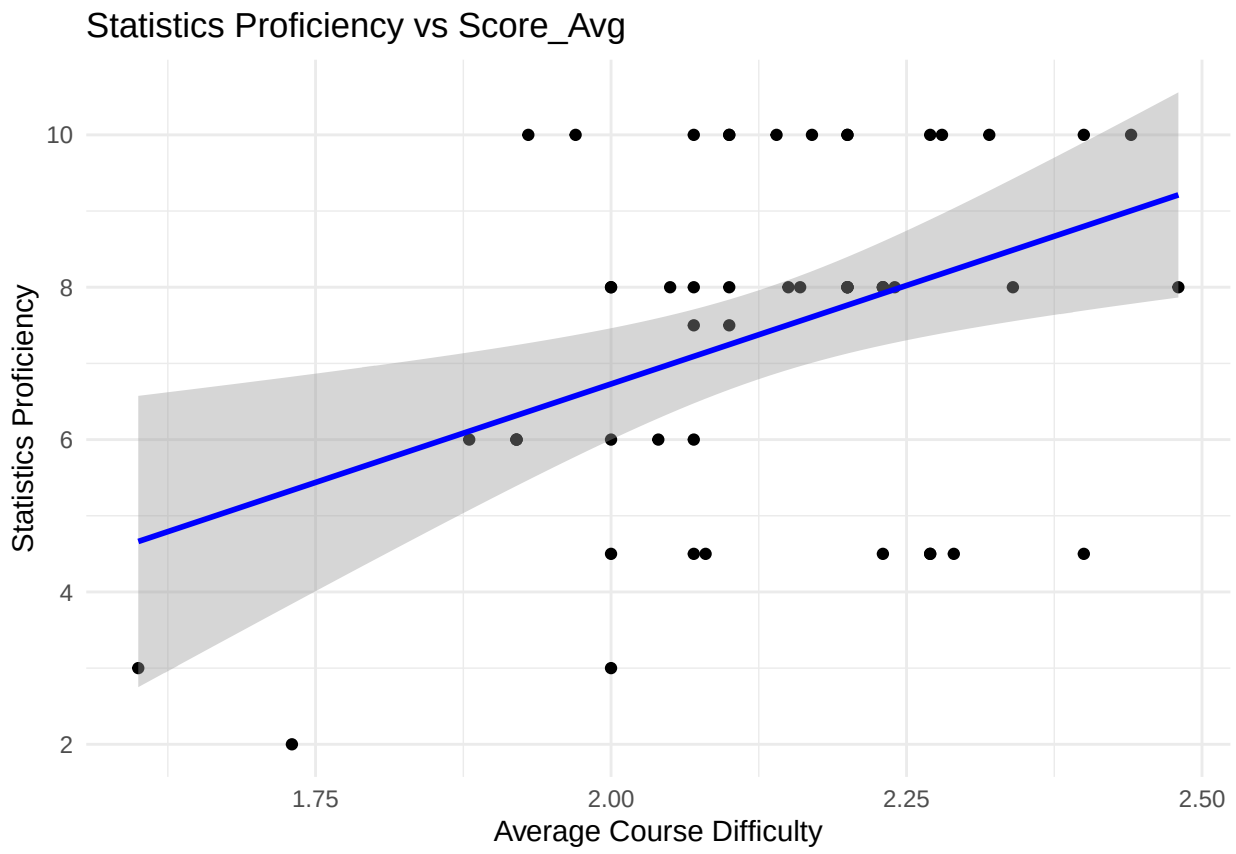

```r
# Programming Proficiency
ggplot(data, aes(x = Score_Avg, y = Prog_Proficiency)) +
  geom_point() +
  geom_smooth(method = "lm", color = "blue") +
  labs(title = "Programming Proficiency vs Score_Avg",
       x = "Average Course Difficulty",
       y = "Programming Proficiency") +
  theme_minimal()
```

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

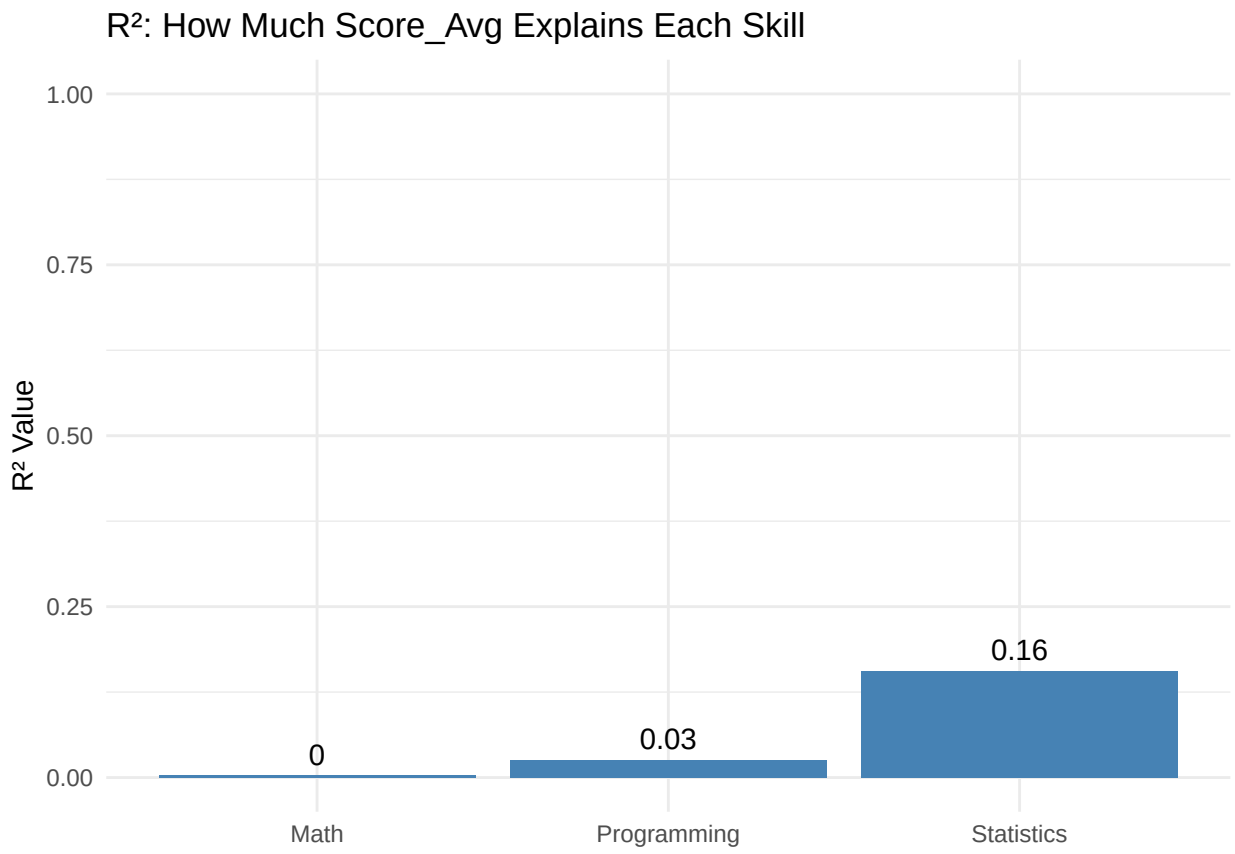



```
# Statistics Proficiency
ggplot(data, aes(x = Score_Avg, y = Stat_Proficiency)) +
  geom_point() +
  geom_smooth(method = "lm", color = "blue") +
  labs(title = "Statistics Proficiency vs Score_Avg",
       x = "Average Course Difficulty",
       y = "Statistics Proficiency") +
  theme_minimal()
```

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




```r
# Build regression models
model_math <- lm(Math_Proficiency ~ Score_Avg, data = data)
model_prog <- lm(Prog_Proficiency ~ Score_Avg, data = data)
model_stat <- lm(Stat_Proficiency ~ Score_Avg, data = data)

# Extract R²
r2_values <- data.frame(
  Skill = c("Math", "Programming", "Statistics"),
  R2 = c(summary(model_math)$r.squared,
         summary(model_prog)$r.squared,
         summary(model_stat)$r.squared)
)

# Bar chart
ggplot(r2_values, aes(x = Skill, y = R2)) +
  geom_col(fill = "steelblue") +
  geom_text(aes(label = round(R2, 2)), vjust = -0.5) +
  labs(title = "R²: How Much Score_Avg Explains Each Skill",
       x = NULL, y = "R² Value") +
  ylim(0, 1) +
  theme_minimal()
```

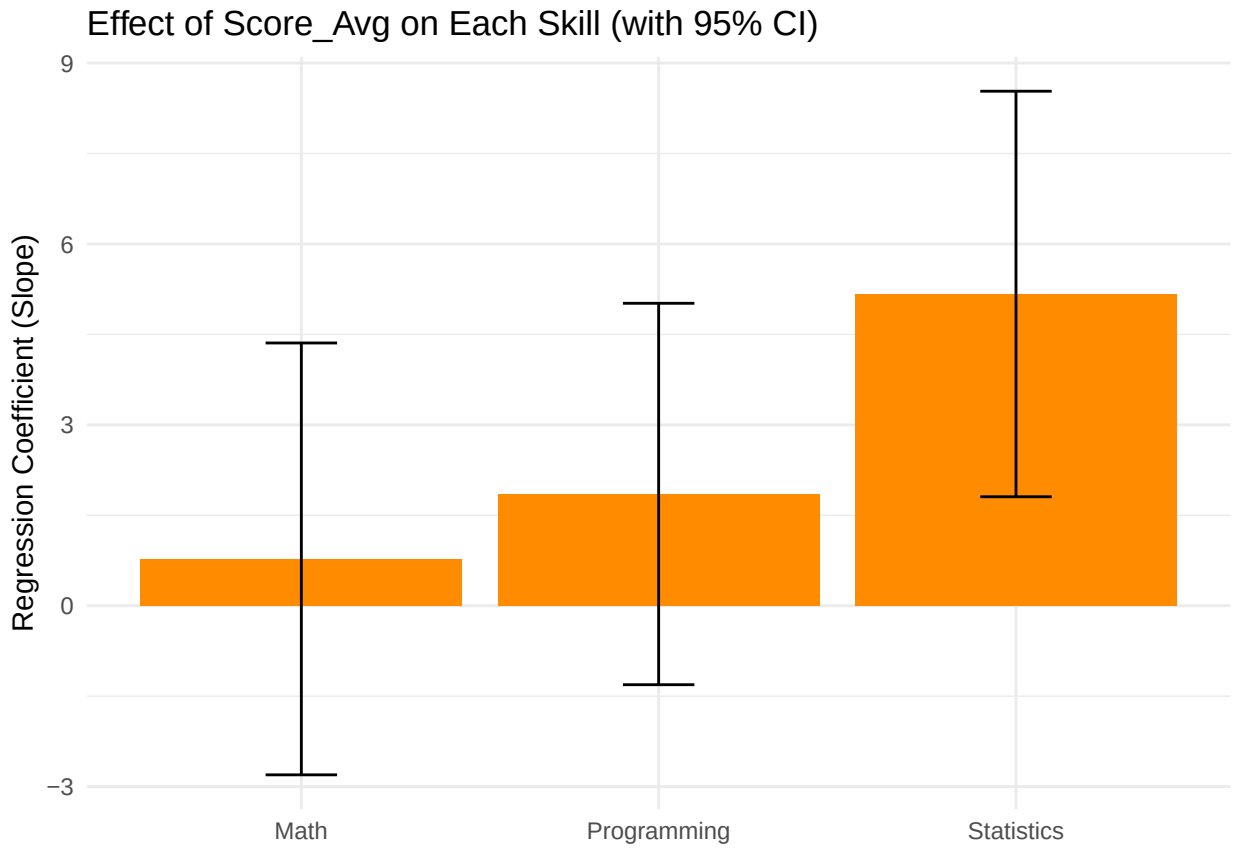



```r
# Load required libraries
library(ggplot2)
library(broom)
library(dplyr)
```

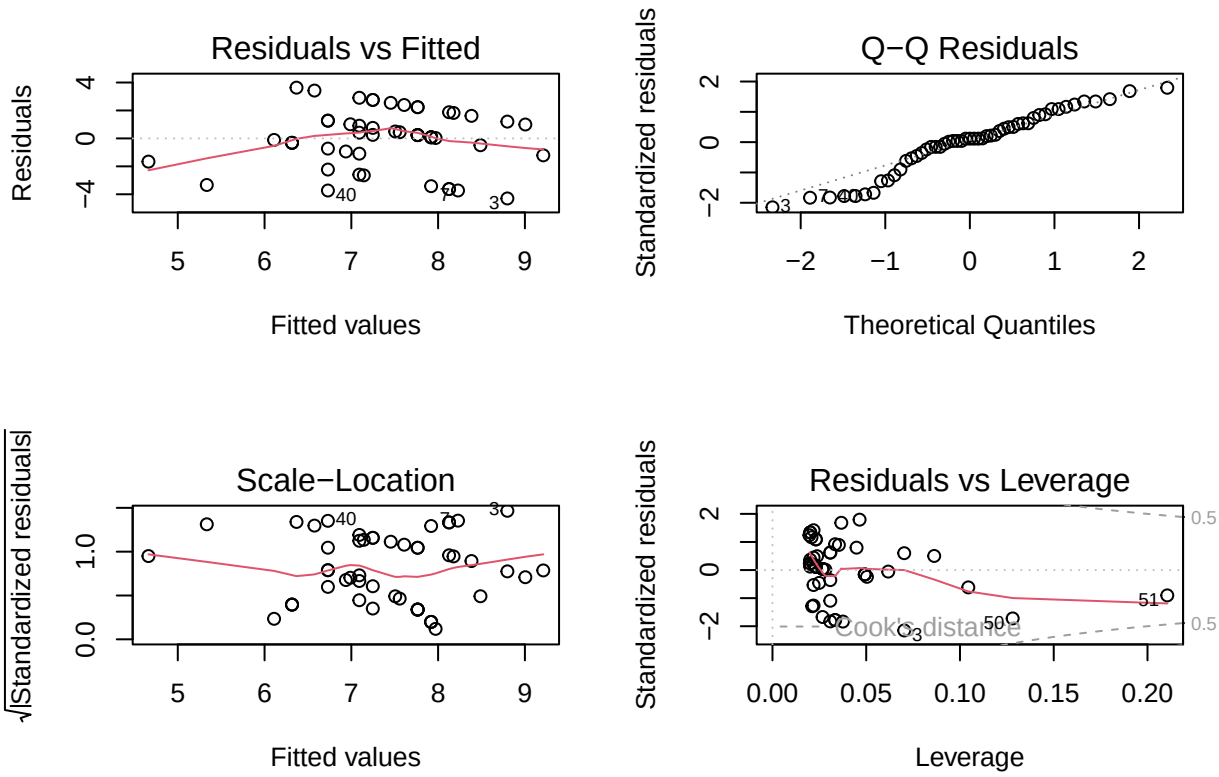
```
##
## Attaching package: 'dplyr'

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

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

```r
# Build models
model_math <- lm(Math_Proficiency ~ Score_Avg, data = data)
model_prog <- lm(Prog_Proficiency ~ Score_Avg, data = data)
model_stat <- lm(Stat_Proficiency ~ Score_Avg, data = data)

# Extract coefficients and standard errors
coef_data <- tibble(
  Skill = c("Math", "Programming", "Statistics"),
```

```
  Coef = c(coef(model_math)[2], coef(model_prog)[2], coef(model_stat)[2]),
  SE = c(tidy(model_math)$std.error[2],
         tidy(model_prog)$std.error[2],
         tidy(model_stat)$std.error[2])
)

# Plot with error bars
ggplot(coef_data, aes(x = Skill, y = Coef)) +
  geom_col(fill = "darkorange") +
  geom_errorbar(aes(ymin = Coef - 1.96 * SE, ymax = Coef + 1.96 * SE), width = 0.2) +
  labs(
    title = "Effect of Score_Avg on Each Skill (with 95% CI)",
    y = "Regression Coefficient (Slope)",
    x = NULL
  ) +
  theme_minimal()
```




```
# Check residuals for statistics model
par(mfrow = c(2,2))
plot(model_stat)
```


Residuals vs Fitted
Residuals
Fitted values
Q−Q Residuals
Standardized residuals
Theoretical Quantiles
Scale−Location
√|Standardized residuals|
Fitted values
Residuals vs Leverage
Standardized residuals
Cook's distance
Leverage