p8130_final

Zihan Lin

R Markdown

```
# Load the dataset
data <- read.csv("/Users/suwa/Desktop/p8130_final/data/Project_1_data.csv")

# Inspect the data structure
glimpse(data)  # Overview of the dataset
```

```
## Rows: 948
## Columns: 14
## $ Gender              <chr> "female", "female", "female", "male", "male", "fem~
## $ EthnicGroup         <chr> "", "group C", "group B", "group A", "group C", "g~
## $ ParentEduc          <chr> "bachelor's degree", "some college", "master's deg~
## $ LunchType           <chr> "standard", "standard", "standard", "free/reduced"~
## $ TestPrep            <chr> "none", "", "none", "none", "none", "none", "compl~
## $ ParentMaritalStatus <chr> "married", "married", "single", "married", "marrie~
## $ PracticeSport       <chr> "regularly", "sometimes", "sometimes", "never", "s~
## $ IsFirstChild        <chr> "yes", "yes", "yes", "no", "yes", "yes", "no", "ye~
## $ NrSiblings          <int> 3, 0, 4, 1, 0, 1, 1, 1, 3, NA, 1, 1, 1, 1, 2, 0, 0~
## $ TransportMeans      <chr> "school_bus", "", "school_bus", "", "school_bus", ~
## $ WklyStudyHours      <chr> "< 5", "10-May", "< 5", "10-May", "10-May", "10-Ma~
## $ MathScore           <int> 71, 69, 87, 45, 76, 73, 85, 41, 65, 37, 58, 40, 66~
## $ ReadingScore        <int> 71, 90, 93, 56, 78, 84, 93, 43, 64, 59, 54, 52, 82~
## $ WritingScore        <int> 74, 88, 91, 42, 75, 79, 89, 39, 68, 50, 52, 43, 74~
```

```
# Check for missing values
cat("Missing Values Summary:\n")
```

```
## Missing Values Summary:
```

```
colSums(is.na(data))  # Count missing values per column
```

```
##              Gender         EthnicGroup          ParentEduc           LunchType
##                   0                   0                   0                   0
##            TestPrep ParentMaritalStatus       PracticeSport       IsFirstChild
##                   0                   0                   0                   0
##          NrSiblings      TransportMeans      WklyStudyHours          MathScore
##                  46                   0                   0                   0
##        ReadingScore        WritingScore
##                   0                   0
```

```r
# Handle missing values
data <- data %>%
  mutate(across(where(is.numeric), ~ ifelse(is.na(.), mean(., na.rm = TRUE), .)))

# For categorical variables, impute missing with the mode
get_mode <- function(x) {
  unique_x <- unique(na.omit(x))
  unique_x[which.max(tabulate(match(x, unique_x)))]
}

data <- data %>%
  mutate(across(where(is.character), ~ ifelse(is.na(.), get_mode(.), .)))

# Check for duplicates
cat("Checking for duplicate rows:\n")
```

```
## Checking for duplicate rows:
```

```r
sum(duplicated(data))  # Count duplicate rows
```

```
## [1] 0
```

```r
# Remove duplicates if any
data <- data %>% distinct()

# Check for invalid or inconsistent values
cat("Summary of score variables:\n")
```

```
## Summary of score variables:
```

```r
summary(data$MathScore)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   56.00   66.00   65.98   76.00  100.00
```

```r
summary(data$ReadingScore)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   17.00   59.00   69.50   68.84   80.00  100.00
```

```r
summary(data$WritingScore)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   10.00   57.00   68.00   67.93   78.25  100.00
```

```r
# Replace invalid scores (e.g., >100 or <0) with NA
data <- data %>%
  mutate(
    MathScore = ifelse(MathScore < 0 | MathScore > 100, NA, MathScore),
```

```
    ReadingScore = ifelse(ReadingScore < 0 | ReadingScore > 100, NA, ReadingScore),
    WritingScore = ifelse(WritingScore < 0 | WritingScore > 100, NA, WritingScore)
  )

# Recheck missing values after cleaning
cat("Missing Values Summary After Cleaning:\n")
```

```
## Missing Values Summary After Cleaning:
```

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

```
##              Gender         EthnicGroup          ParentEduc           LunchType
##                   0                   0                   0                   0
##            TestPrep ParentMaritalStatus       PracticeSport        IsFirstChild
##                   0                   0                   0                   0
##          NrSiblings      TransportMeans      WklyStudyHours           MathScore
##                   0                   0                   0                   0
##        ReadingScore        WritingScore
##                   0                   0
```

```
# Recheck data types and convert if necessary
cat("Converting categorical variables to factors...\n")
```

```
## Converting categorical variables to factors...
```

```
data <- data %>%
  mutate(
    Gender = as.factor(Gender),
    EthnicGroup = as.factor(EthnicGroup),
    ParentEduc = as.factor(ParentEduc),
    LunchType = as.factor(LunchType),
    TestPrep = as.factor(TestPrep),
    ParentMaritalStatus = as.factor(ParentMaritalStatus),
    PracticeSport = as.factor(PracticeSport),
    IsFirstChild = as.factor(IsFirstChild),
    TransportMeans = as.factor(TransportMeans),
    WklyStudyHours = as.factor(WklyStudyHours)
  )

# Standardize numeric variables (if needed for modeling)
# Example: Scale test scores
data <- data %>%
  mutate(across(c(MathScore, ReadingScore, WritingScore), scale))

# Save Cleaned Data
write.csv(data, "/Users/suwa/Desktop/p8130_final/data/data_cleaned.csv", row.names = FALSE)
```

```
# Reload the dataset
data <- read.csv("/Users/suwa/Desktop/p8130_final/data/data_cleaned.csv")

# Generate a summary table
skim(data)
```

Table 1: Data summary

Name	data
Number of rows	948
Number of columns	14
Column type frequency:	
character	10
numeric	4
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Gender	0	1	4	6	0	2	0
EthnicGroup	0	1	0	7	59	6	0
ParentEduc	0	1	0	18	53	7	0
LunchType	0	1	8	12	0	2	0
TestPrep	0	1	0	9	55	3	0
ParentMaritalStatus	0	1	0	8	49	5	0
PracticeSport	0	1	0	9	16	4	0
IsFirstChild	0	1	0	3	30	3	0
TransportMeans	0	1	0	10	102	3	0
WklyStudyHours	0	1	0	6	37	4	0

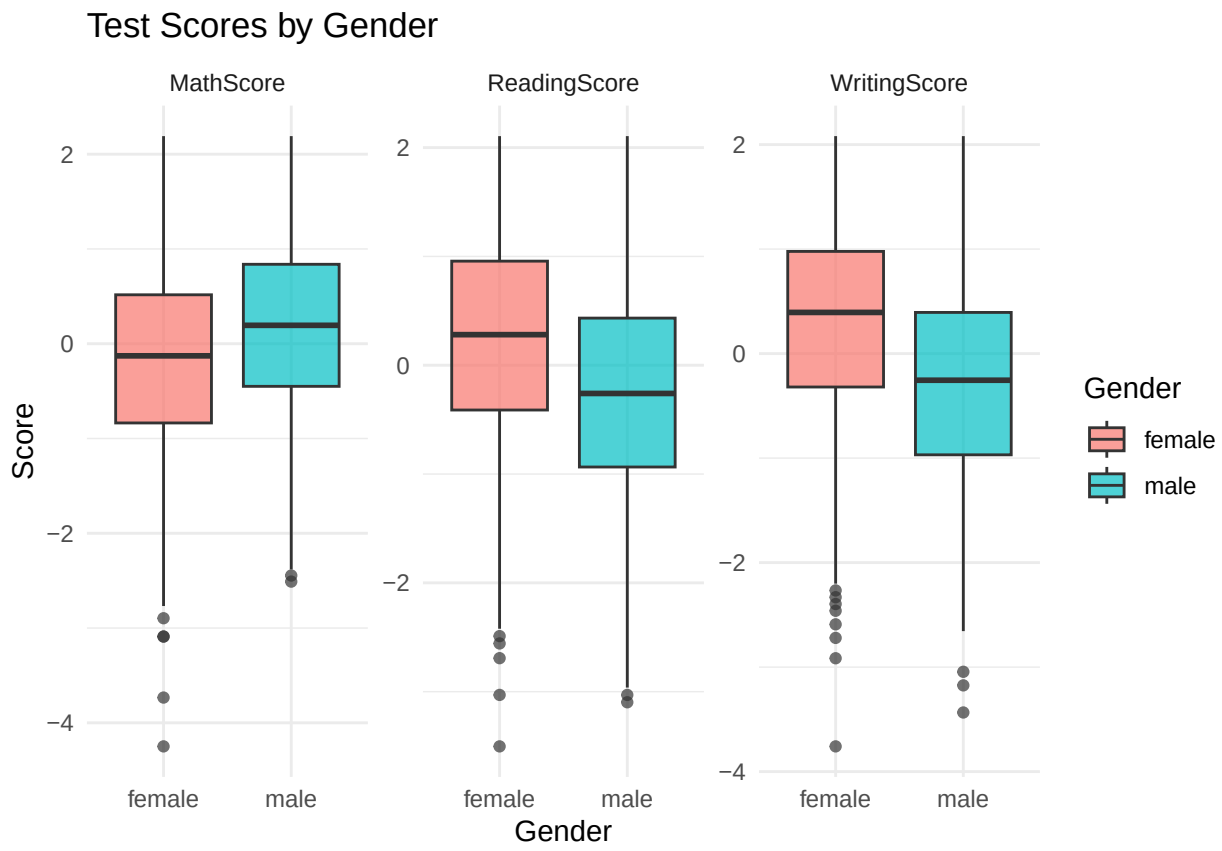
Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
NrSiblings	0	1	2.16	1.45	0.00	1.00	2.00	3.00	7.00	
MathScore	0	1	0.00	1.00	-4.25	-0.64	0.00	0.65	2.19	
ReadingScore	0	1	0.00	1.00	-3.50	-0.67	0.04	0.75	2.11	
WritingScore	0	1	0.00	1.00	-3.76	-0.71	0.00	0.67	2.08	

```
# Distributions of Test Scores
# Histograms for each test score
data %>%
  select(MathScore, ReadingScore, WritingScore) %>%
  pivot_longer(everything(), names_to = "Test", values_to = "Score") %>%
  ggplot(aes(x = Score, fill = Test)) +
  geom_histogram(binwidth = 5, alpha = 0.7, position = "dodge") +
  labs(title = "Distributions of Test Scores", x = "Score", y = "Frequency") +
  theme_minimal()
```


```
# Boxplots for test scores
data %>%
  select(MathScore, ReadingScore, WritingScore) %>%
  pivot_longer(everything(), names_to = "Test", values_to = "Score") %>%
  ggplot(aes(x = Test, y = Score, fill = Test)) +
  geom_boxplot(alpha = 0.7) +
  labs(title = "Boxplots of Test Scores", x = "Test", y = "Score") +
  theme_minimal()
```

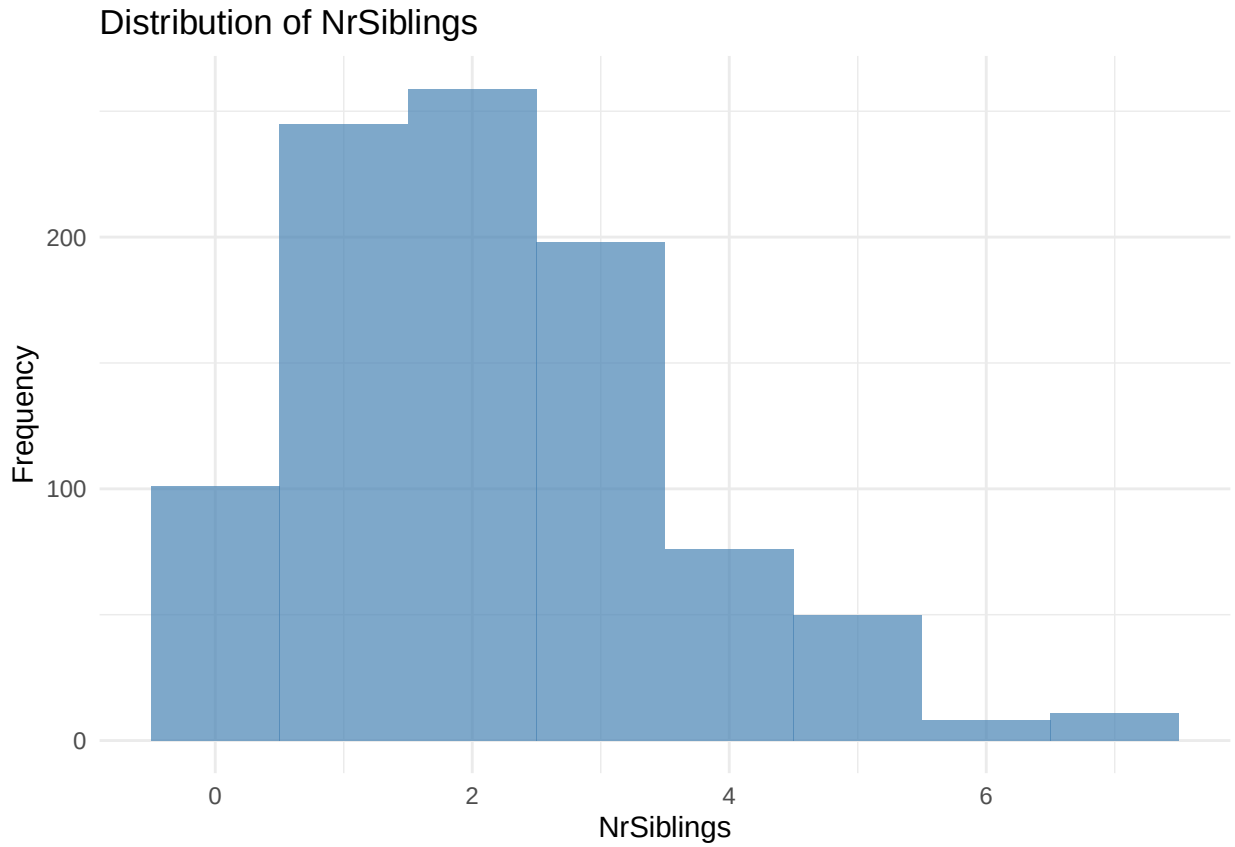

```r
# Boxplots for Categorical Covariates vs Test Scores
# Explore relationships between categorical covariates and test scores
data %>%
  pivot_longer(cols = c(MathScore, ReadingScore, WritingScore),
               names_to = "TestType", values_to = "Score") %>%
  ggplot(aes(x = Gender, y = Score, fill = Gender)) +
  geom_boxplot(alpha = 0.7) +
  facet_wrap(~TestType, scales = "free") +
  labs(title = "Test Scores by Gender",
       x = "Gender",
       y = "Score") +
  theme_minimal()
```



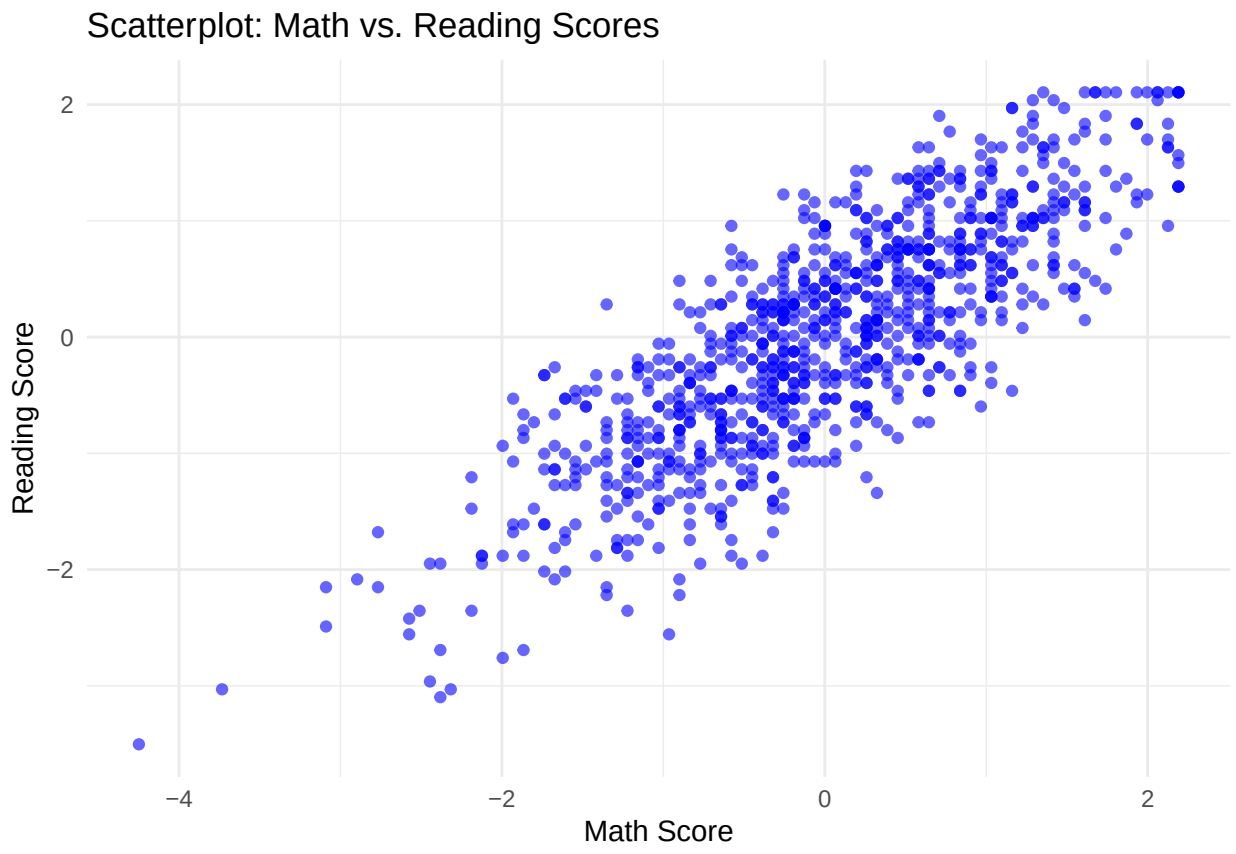

```
# Distributions of Covariates
# Bar plots for categorical variables
categorical_vars <- data %>%
  select(where(is.factor))

for (var in names(categorical_vars)) {
  print(
    ggplot(data, aes_string(x = var, fill = var)) +
      geom_bar(alpha = 0.7) +
      labs(title = paste("Distribution of", var), x = var, y = "Count") +
      theme_minimal() +
      theme(axis.text.x = element_text(angle = 45, hjust = 1))
  )
}

# Histograms for numeric covariates
numeric_vars <- data %>%
  select(where(is.numeric), -c(MathScore, ReadingScore, WritingScore))

for (var in names(numeric_vars)) {
  print(
    ggplot(data, aes_string(x = var)) +
      geom_histogram(binwidth = 1, fill = "steelblue", alpha = 0.7) +
      labs(title = paste("Distribution of", var), x = var, y = "Frequency") +
      theme_minimal()
  )
```

```
}
```

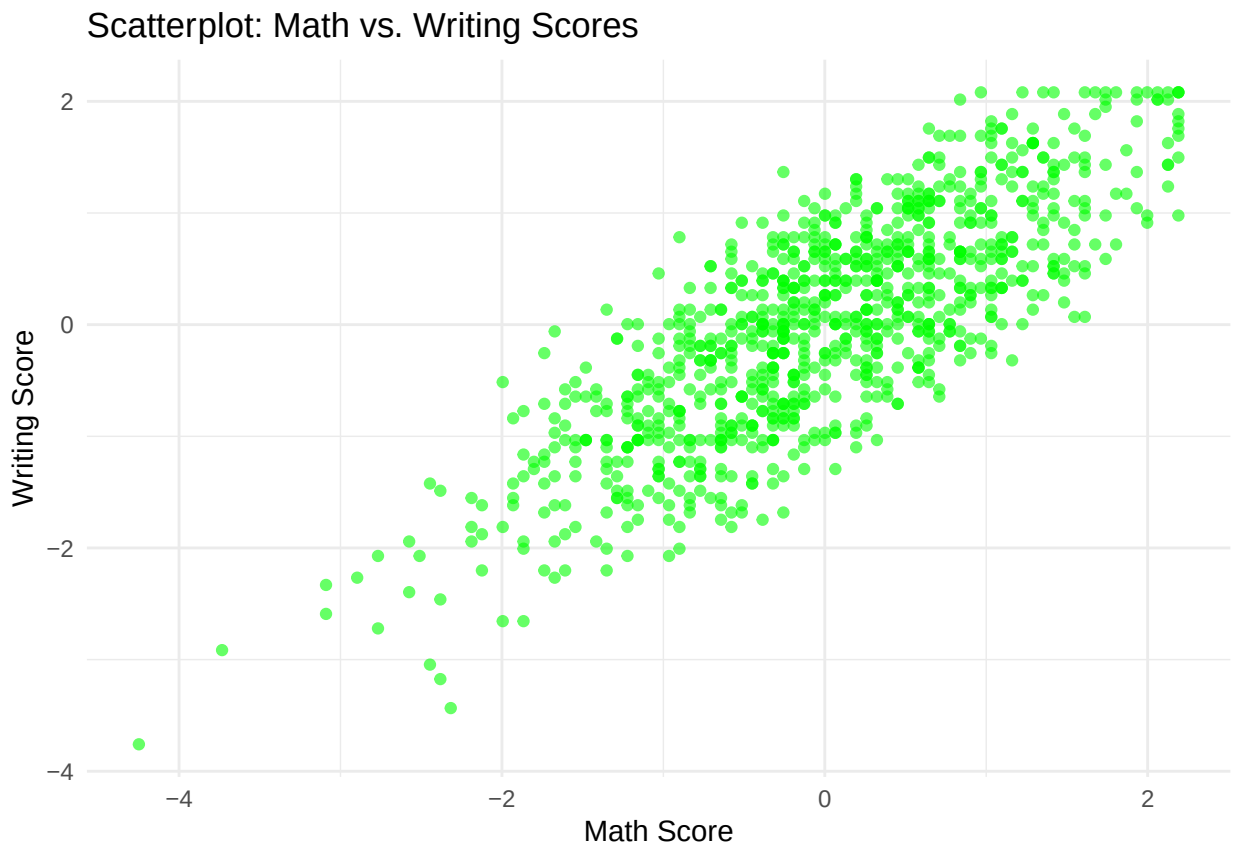
```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this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




```
# Pairwise Relationships
# Scatterplots of test scores
data %>%
  ggplot(aes(x = MathScore, y = ReadingScore)) +
  geom_point(alpha = 0.6, color = "blue") +
  labs(title = "Scatterplot: Math vs. Reading Scores", x = "Math Score", y = "Reading Score") +
  theme_minimal()
```

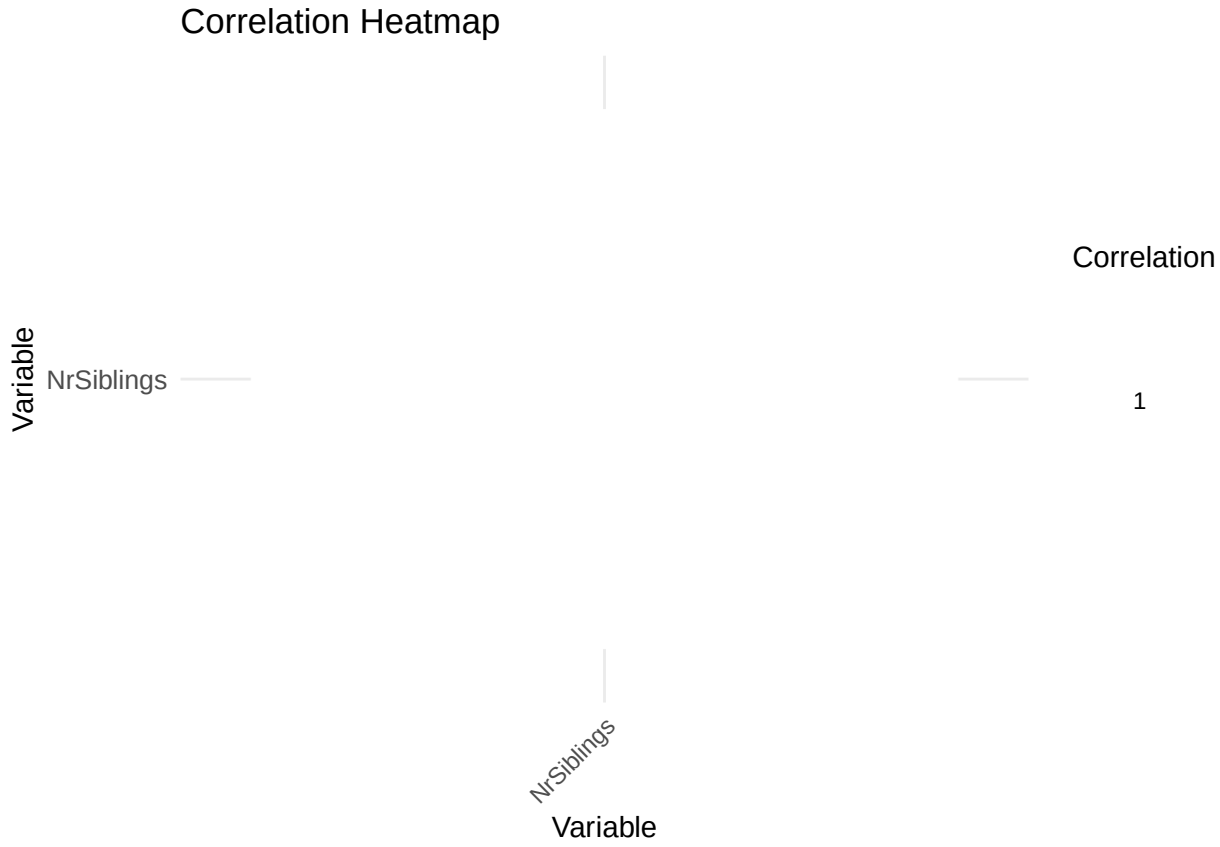



```
data %>%
  ggplot(aes(x = MathScore, y = WritingScore)) +
  geom_point(alpha = 0.6, color = "green") +
  labs(title = "Scatterplot: Math vs. Writing Scores", x = "Math Score", y = "Writing Score") +
  theme_minimal()
```

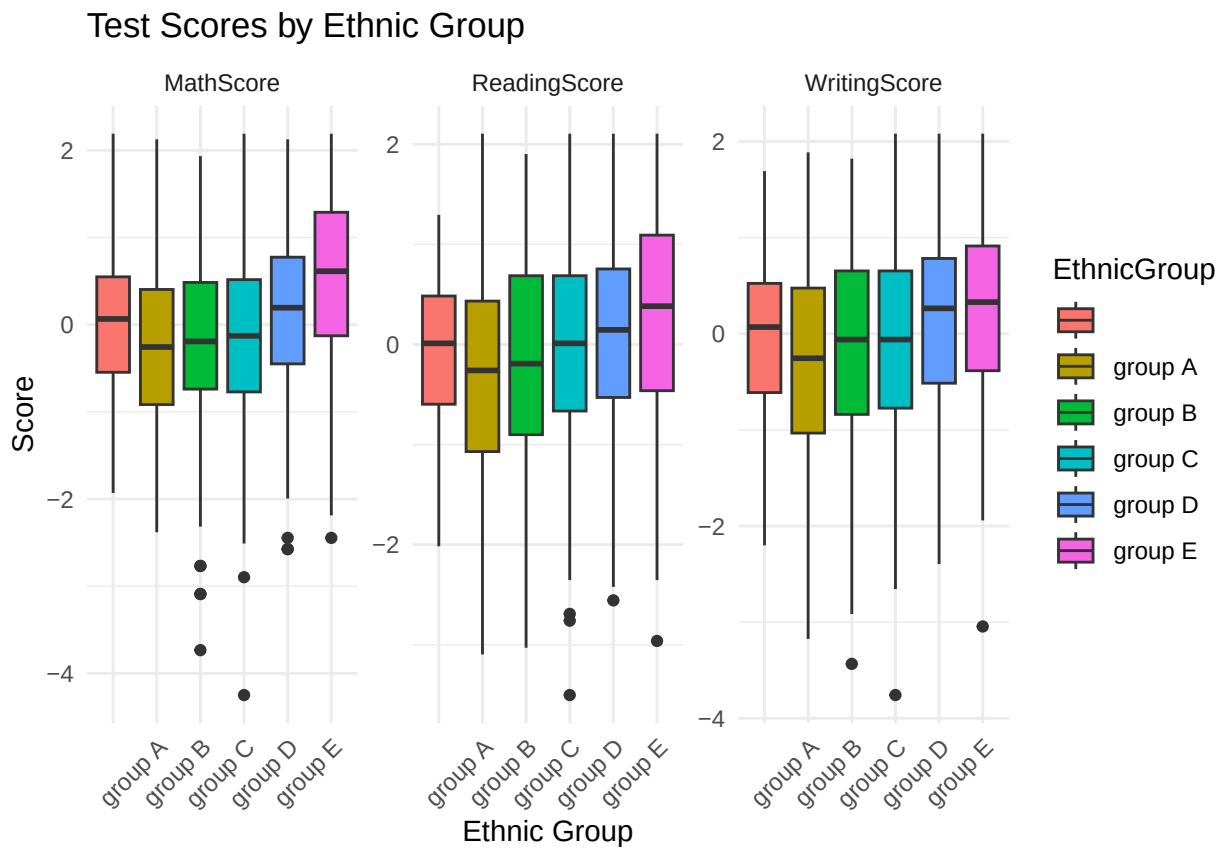



```
data %>%
  ggplot(aes(x = ReadingScore, y = WritingScore)) +
  geom_point(alpha = 0.6, color = "purple") +
  labs(title = "Scatterplot: Reading vs. Writing Scores", x = "Reading Score", y = "Writing Score") +
  theme_minimal()
```


```
# Correlation heatmap
corr_matrix <- cor(numeric_vars, use = "complete.obs")

corr_matrix %>%
  as.data.frame() %>%
  rownames_to_column(var = "Variable1") %>%
  pivot_longer(cols = -Variable1, names_to = "Variable2", values_to = "Correlation") %>%
  ggplot(aes(x = Variable1, y = Variable2, fill = Correlation)) +
  geom_tile(color = "white") +  # Add grid lines
  scale_fill_gradient2(low = "blue", mid = "white", high = "red", midpoint = 0) +
  labs(title = "Correlation Heatmap", x = "Variable", y = "Variable", fill = "Correlation") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        axis.text.y = element_text(size = 10))
```




```
# Boxplots for test scores by EthnicGroup
data %>%
  pivot_longer(cols = c(MathScore, ReadingScore, WritingScore), names_to = "TestType", values_to = "Sco
  ggplot(aes(x = EthnicGroup, y = Score, fill = EthnicGroup)) +
  geom_boxplot() +
  facet_wrap(~ TestType, scales = "free") +
  theme_minimal() +
  labs(title = "Test Scores by Ethnic Group", x = "Ethnic Group", y = "Score") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Findings from Exploratory Data Analysis (EDA)

Pairwise Relationships:

Strong correlations between MathScore, ReadingScore, and WritingScore (r 0.95), suggesting redundancy in predictors for individual models. Weak correlation between NrSiblings and test scores. Visualizations indicate potential interaction effects, for example, between Gender and LunchType on MathScore.

Distributions:

Numeric variables like MathScore, ReadingScore, and WritingScore exhibit nearly normal distributions but with some skewness in scores below 50. NrSiblings is positively skewed with most values concentrated around 1 to 3.

Interactions and Covariate Effects:

Boxplots reveal that WklyStudyHours and EthnicGroup significantly impact test scores. Students with more than 10 hours of study time score higher across all test types.

Covariate Analysis:

Weekly study hours (WklyStudyHours) and test preparation (TestPrep) have clear separations in performance, suggesting strong predictive potential. Interaction plots highlight a differential impact of LunchType based on Gender.