Q2.Import the "Students Performance in Exams" dataset from Kaggle, which analyzes students' performance based on various factors like gender, parental education, lunch type, and test preparation

In [ ]:

```
install.packages(c("ggplot2", "dplyr", "tidyverse", "caret", "cluster", "future"
library(ggplot2)
library(dplyr)
library(tidyverse)
library(caret)
library(cluster)
library(future)
library(foreach)
library(e1071)
library(doParallel)
```

loading dataset

In [46]:

```
students <- read.csv("/content/StudentsPerformance.csv")
```

1. Display the first six rows of the dataset.

In [8]:

```
head(students)
```

A data.frame: 6 × 8

	gender	race.ethnicity	parental.level.of.education	lunch	test.preparation.course
	<chr>	<chr>	<chr>	<chr>	<chr>
1	female	group B	bachelor's degree	standard	none
2	female	group C	some college	standard	completed
3	female	group B	master's degree	standard	none
4	male	group A	associate's degree	free/reduced	none
5	male	group C	some college	standard	none
6	female	group B	associate's degree	standard	none

2. Check structure

In [9]:

```
str(students)
```

```
'data.frame':	1000 obs. of  8 variables:
 $ gender                     : chr  "female" "female" "female" "male" ...
 $ race.ethnicity             : chr  "group B" "group C" "group B" "group A" ...
 $ parental.level.of.education: chr  "bachelor's degree" "some college" "master's degree" "associate's degree" ...
 $ lunch                      : chr  "standard" "standard" "standard" "free/reduced" ...
 $ test.preparation.course    : chr  "none" "completed" "none" "none" ...
 $ math.score                 : int  72 69 90 47 76 71 88 40 64 38 ...
 $ reading.score              : int  72 90 95 57 78 83 95 43 64 60 ...
 $ writing.score              : int  74 88 93 44 75 78 92 39 67 50 ...
```

3. Identify missing values

In [10]:

```
sum(is.na(students))
```

0

4. Compute mean, median, and standard deviation for scores

In [19]:

```
score_stats <- students %>%
  summarize(
    math_mean = mean(math.score, na.rm = TRUE),
    math_median = median(math.score, na.rm = TRUE),
    math_sd = sd(math.score, na.rm = TRUE),

    reading_mean = mean(reading.score, na.rm = TRUE),
    reading_median = median(reading.score, na.rm = TRUE),
    reading_sd = sd(reading.score, na.rm = TRUE),

    writing_mean = mean(writing.score, na.rm = TRUE),
    writing_median = median(writing.score, na.rm = TRUE),
    writing_sd = sd(writing.score, na.rm = TRUE)
  )
print(score_stats)
```

```
  math_mean math_median  math_sd reading_mean reading_median reading_sd
1    66.089          66 15.16308       69.169             70   14.60019
  writing_mean writing_median writing_sd
1       68.054             69   15.19566
```

5. Student with highest total score

In [21]:

```
students$total_score <- students$math.score + students$reading.score + students$
highest_score_student <- students[which.max(students$total_score), ]
highest_score_student
```

A data.frame: 1 × 9

	gender	race.ethnicity	parental.level.of.education	lunch	test.preparation.course
	<chr>	<chr>	<chr>	<chr>	<chr>
459	female	group E	bachelor's degree	standard	none

6. Percentage of students scoring above 90 in math

In [23]:

```
above_90 <- sum(students$math.score > 90) / nrow(students) * 100
above_90
```

5

7. Compare average scores by gender

In [25]:

```
aggregate(cbind(math.score, reading.score, writing.score) ~ gender, data = stude
```

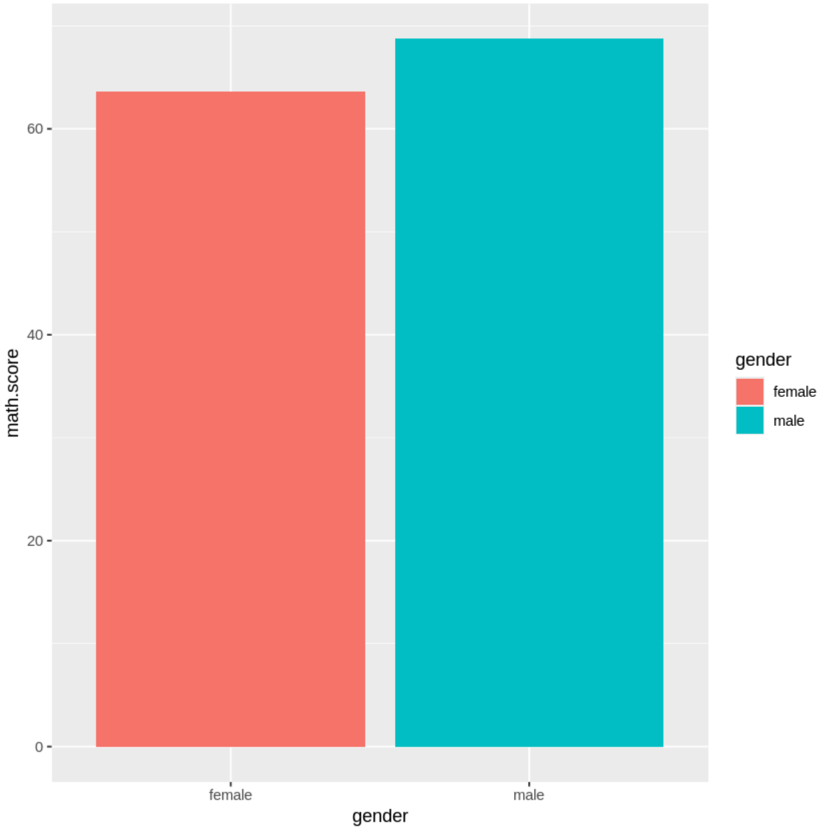
A data.frame: 2 × 4

gender	math.score	reading.score	writing.score
<chr>	<dbl>	<dbl>	<dbl>
female	63.63320	72.60811	72.46718
male	68.72822	65.47303	63.31120

8. Bar plot of math scores by gender

In [26]:

```
ggplot(students, aes(x = gender, y = math.score, fill = gender)) + geom_bar(stat
```



9. T-test for reading scores by gender

In [28]:

```
t.test(reading.score ~ gender, data = students)
```

```
	Welch Two Sample t-test

data:  reading.score by gender
t = 7.9684, df = 996.36, p-value = 4.376e-15
alternative hypothesis: true difference in means between group female and group m
ale is not equal to 0
95 percent confidence interval:
 5.377941 8.892218
sample estimates:
mean in group female   mean in group male 
            72.60811             65.47303 
```

10. Average math score by parental education

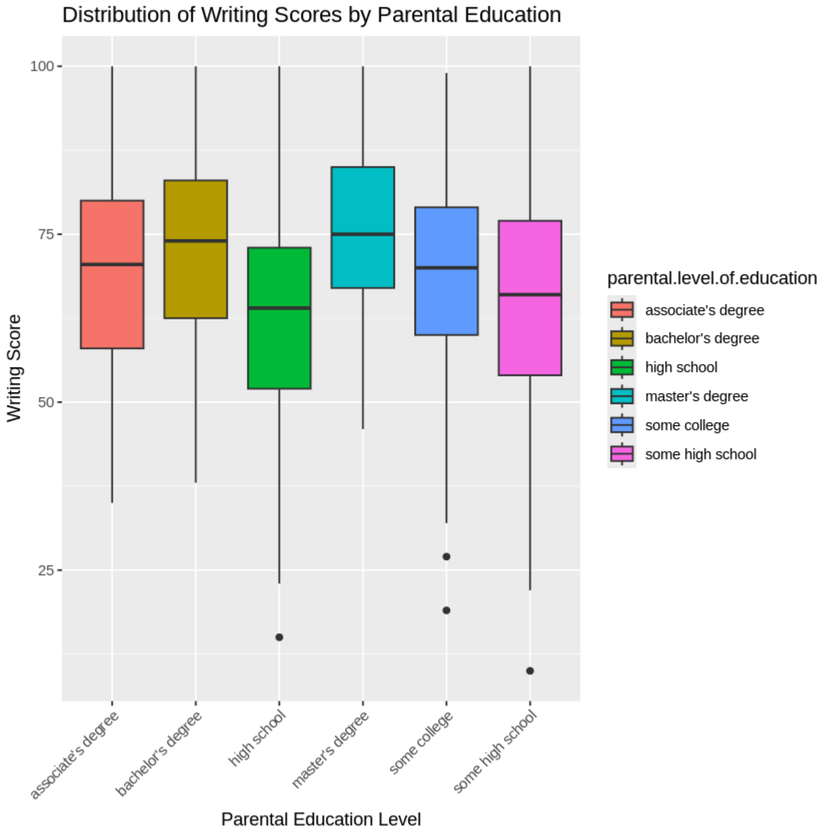
```
In [29]: aggregate(math.score ~ parental.level.of.education, data = students, FUN = mean)
```

A data.frame: 6 × 2

parental.level.of.education	math.score
<chr>	<dbl>
associate's degree	67.88288
bachelor's degree	69.38983
high school	62.13776
master's degree	69.74576
some college	67.12832
some high school	63.49721

11. Boxplot of writing scores by parental education

```
In [30]: ggplot(students, aes(x = parental.level.of.education, y = writing.score, fill = 
           geom_boxplot() +
           labs(title = "Distribution of Writing Scores by Parental Education", x = "Pare
           theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



12. ANOVA test for parental education and writing scores

In [31]:

```
anova_result <- aov(writing.score ~ parental.level.of.education, data = students
summary(anova_result)
```

```
                             Df Sum Sq Mean Sq F value   Pr(>F)    
parental.level.of.education   5  15623  3124.6   14.44 1.12e-13 ***
Residuals                   994 215054   216.4                     
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

13. Compare test preparation impact on scores

In [32]:

```
aggregate(cbind(math.score, reading.score, writing.score) ~ test.preparation.cou
```

A data.frame: 2 × 4

test.preparation.course	math.score	reading.score	writing.score
<chr>	<dbl>	<dbl>	<dbl>
completed	69.69553	73.89385	74.41899
none	64.07788	66.53427	64.50467

14. T-test for test preparation impact

In [33]:

```
t.test(math.score ~ test.preparation.course, data = students)
```

```
	Welch Two Sample t-test

data:  math.score by test.preparation.course
t = 5.787, df = 770.08, p-value = 1.043e-08
alternative hypothesis: true difference in means between group completed and grou
p none is not equal to 0
95 percent confidence interval:
 3.712041 7.523257
sample estimates:
mean in group completed      mean in group none 
               69.69553                64.07788 
```

15. Correlation between scores

In [34]:

```
cor(students[c("math.score", "reading.score", "writing.score")])
```

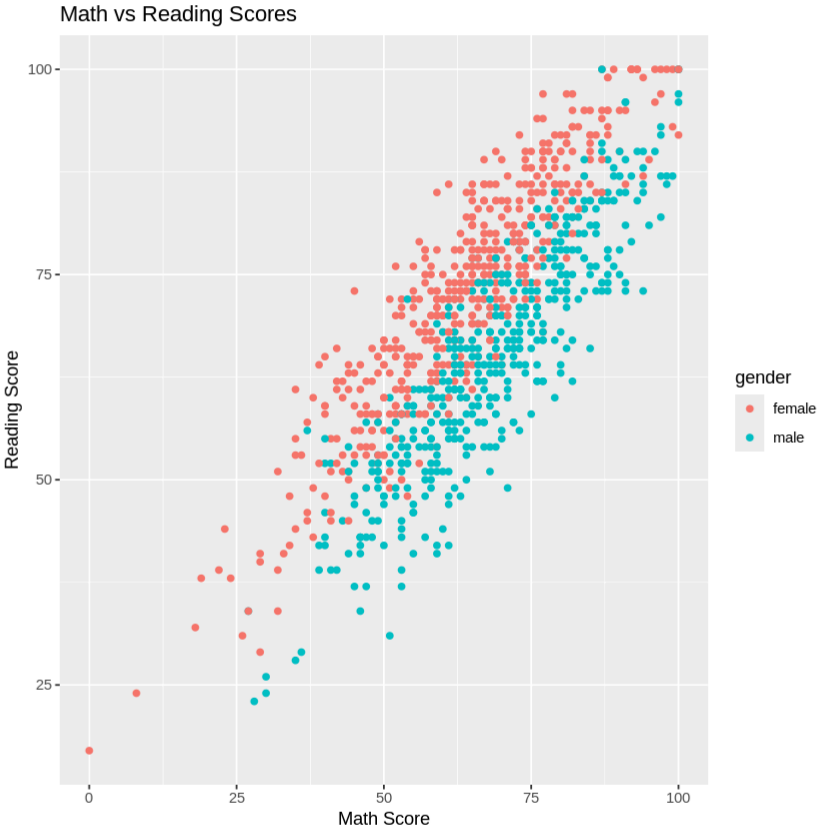
A matrix: 3 × 3 of type dbl

	math.score	**reading.score**	**writing.score**
math.score	1.0000000	0.8175797	0.8026420
reading.score	0.8175797	1.0000000	0.9545981
writing.score	0.8026420	0.9545981	1.0000000

16. Scatter plot of math vs. reading scores

In [35]:

```
ggplot(students, aes(x = math.score, y = reading.score)) +
  geom_point(aes(color = gender)) +
  labs(title = "Math vs Reading Scores", x = "Math Score", y = "Reading Score")
```



17. Linear regression to predict math scores

In [36]:
```
lm_model <- lm(math.score ~ parental.level.of.education + test.preparation.cours
summary(lm_model)
```

```
Call:
lm(formula = math.score ~ parental.level.of.education + test.preparation.course +
    lunch, data = students)

Residuals:
    Min      1Q  Median      3Q     Max 
-53.573  -9.388   0.183  10.122  35.885 

Coefficients:
                                             Estimate Std. Error t value
(Intercept)                                   64.1150     1.2294  52.153
parental.level.of.educationbachelor's degree   1.6840     1.5610   1.079
parental.level.of.educationhigh school        -5.1486     1.3449  -3.828
parental.level.of.educationmaster's degree     2.7151     2.0075   1.352
parental.level.of.educationsome college       -0.5594     1.2948  -0.432
parental.level.of.educationsome high school   -4.8032     1.3773  -3.487
test.preparation.coursenone                   -5.7389     0.9081  -6.319
lunchstandard                                 11.3097     0.9060  12.483
                                             Pr(>|t|)    
(Intercept)                                   < 2e-16 ***
parental.level.of.educationbachelor's degree 0.280954    
parental.level.of.educationhigh school       0.000137 ***
parental.level.of.educationmaster's degree   0.176536    
parental.level.of.educationsome college      0.665795    
parental.level.of.educationsome high school  0.000509 ***
test.preparation.coursenone                  3.96e-10 ***
lunchstandard                                 < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 13.7 on 992 degrees of freedom
Multiple R-squared:  0.1894,	Adjusted R-squared:  0.1837 
F-statistic: 33.12 on 7 and 992 DF,  p-value: < 2.2e-16
```

18. Evaluate model performance

In [37]:
```
summary(lm_model)$r.squared
```

0.189447865286474

19-21. Study hours vs. math score regressio

In [51]:
```
set.seed(123)  # Set seed for reproducibility
students$study_hours <- runif(nrow(students), min = 0, max = 20)  # Random study
```

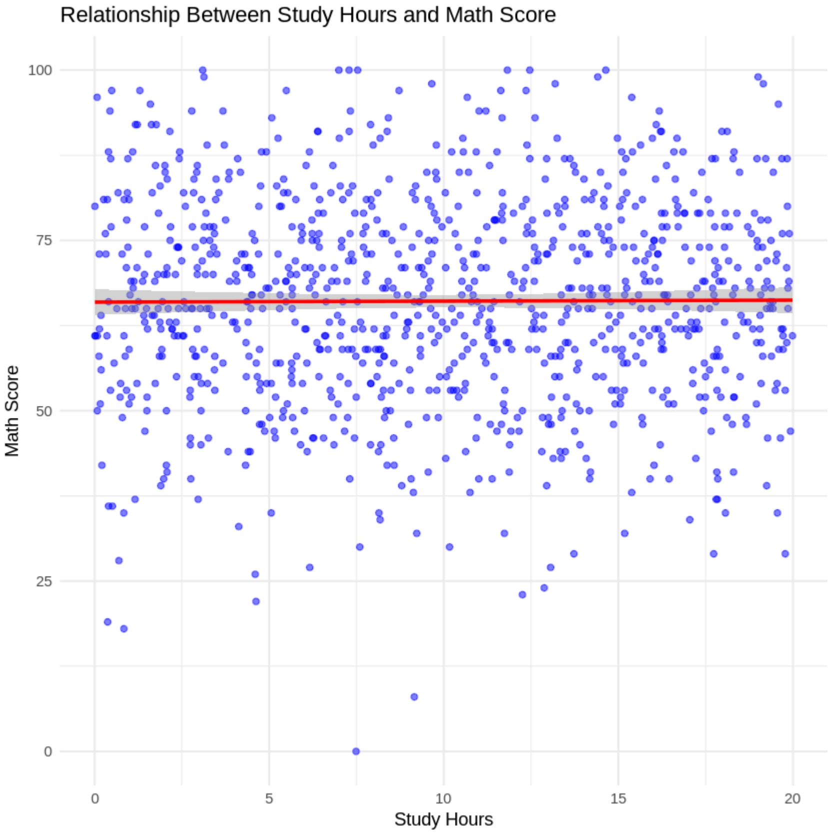
In [52]:
```
study_hours_model <- lm(math.score ~ study_hours, data = students)
summary(study_hours_model)
```

```
Call:
lm(formula = math.score ~ study_hours, data = students)

Residuals:
    Min      1Q  Median      3Q     Max 
-66.053  -9.167  -0.033  10.795  34.010 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 65.94525    0.95887  68.774   <2e-16 ***
study_hours  0.01445    0.08348   0.173    0.863    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 15.17 on 998 degrees of freedom
Multiple R-squared:  3.004e-05,	Adjusted R-squared:  -0.0009719 
F-statistic: 0.02998 on 1 and 998 DF,  p-value: 0.8626
```

22. Visualize the relationship using a scatter plot with a regression line

In [55]:

```r
ggplot(students, aes(x = study_hours, y = math.score)) +
  geom_point(color = "blue", alpha = 0.5) +  # Scatter plot
  geom_smooth(method = "lm", formula = y ~ x, color = "red") +  # Linear regress
  labs(title = "Relationship Between Study Hours and Math Score",
       x = "Study Hours",
       y = "Math Score") +
  theme_minimal()
```



23. Compare scores by lunch type

In [56]:

```r
students %>%
  group_by(lunch) %>%
  summarise(
    avg_math = mean(math.score, na.rm = TRUE),
    avg_reading = mean(reading.score, na.rm = TRUE),
    avg_writing = mean(writing.score, na.rm = TRUE)
  )
```

A tibble: 2 × 4

lunch	avg_math	avg_reading	avg_writing
<chr>	<dbl>	<dbl>	<dbl>
free/reduced	58.92113	64.65352	63.02254
standard	70.03411	71.65426	70.82326

24. Identify high performers

In [42]:

```
high_performers <- students[students$math.score > 85 &
                            students$reading.score > 85 &
                            students$writing.score > 85, ]

head(high_performers)

nrow(high_performers)
```

A data.frame: 6 × 10

	gender	race.ethnicity	parental.level.of.education	lunch	test.preparation.course
	<chr>	<chr>	<chr>	<chr>	<chr>
3	female	group B	master's degree	standard	none
7	female	group B	some college	standard	completed
17	male	group C	high school	standard	none
105	male	group C	some college	standard	completed
107	female	group D	master's degree	standard	none
115	female	group E	bachelor's degree	standard	completed

51