Assignment_3

Htet Khant Linn

2025-11-15

Importing Necessary Libraries

```
library(ggplot2)
```

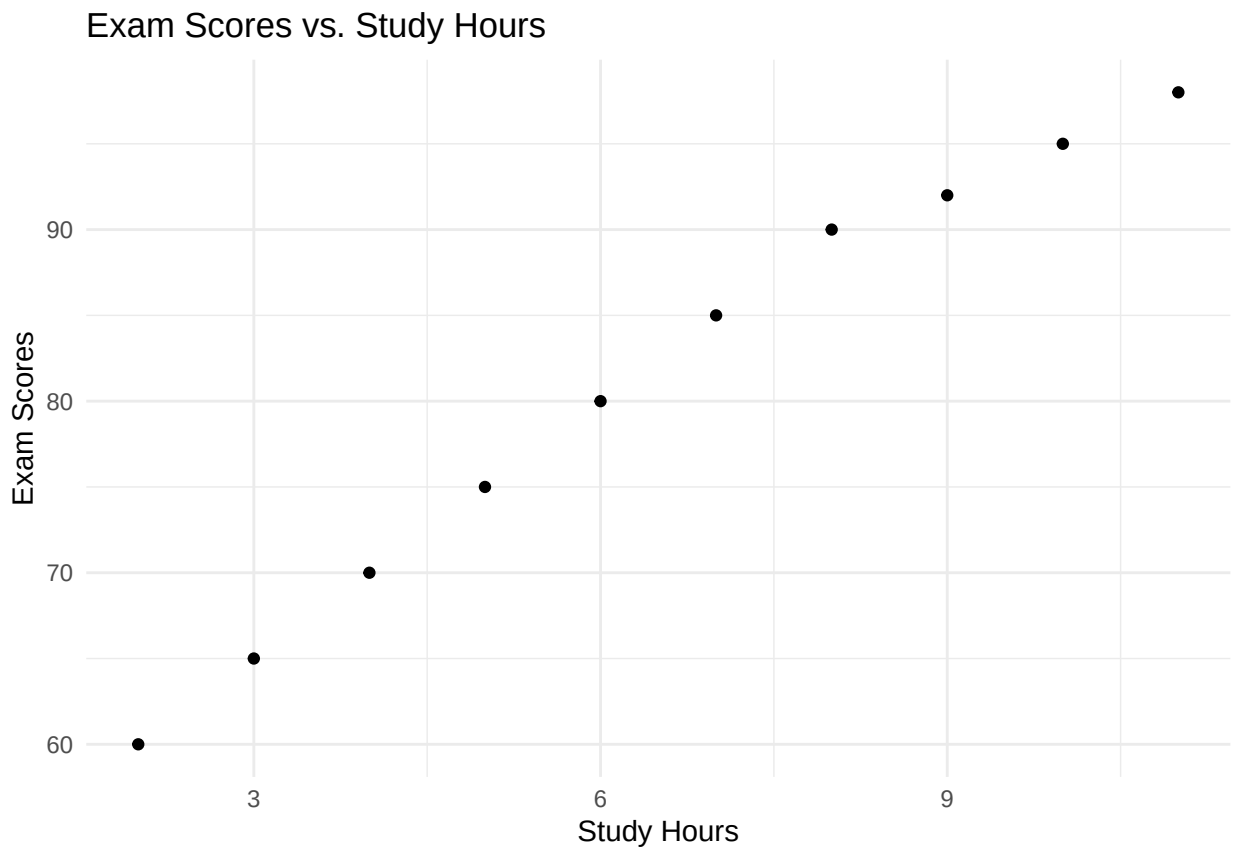
Question 1

```
study_hours <- c(2, 3, 4, 5, 6, 7, 8, 9, 10, 11)
exam_scores <- c(60, 65, 70, 75, 80, 85, 90, 92, 95, 98)
df <- data.frame(study_hours, exam_scores)
df
```

```
##    study_hours exam_scores
## 1            2          60
## 2            3          65
## 3            4          70
## 4            5          75
## 5            6          80
## 6            7          85
## 7            8          90
## 8            9          92
## 9           10          95
## 10          11          98
```

1. Visualizing The Data

```
ggplot(df, aes(study_hours, exam_scores)) +
  geom_point() +
  labs(title = "Exam Scores vs. Study Hours",
          x = "Study Hours",
          y = "Exam Scores") +
  theme_minimal()
```



2. Build Simple Linear Regression Model

```
model1 <- lm(exam_scores ~ study_hours)
model1
```

```
##
## Call:
## lm(formula = exam_scores ~ study_hours)
##
## Coefficients:
## (Intercept)  study_hours
##      52.952        4.315
```

3. Summary of the Model

```
summary(model1)
```

```
##
## Call:
## lm(formula = exam_scores ~ study_hours)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4182 -1.0515  0.0000  0.9864  2.5273
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  52.9515     1.2917   40.99 1.38e-10 ***
## study_hours   4.3152     0.1818   23.74 1.06e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.651 on 8 degrees of freedom
## Multiple R-squared:  0.986,  Adjusted R-squared:  0.9843
## F-statistic: 563.6 on 1 and 8 DF,  p-value: 1.055e-08
```

4. Estimated Intercept & Slope

The estimated intercept (c): **52.9515** The slope (m): **4.3152**

Regression Equation: y = mx + c -> `Exam Score = (4.3152 * Study Hours) + 52.9515`

5. The slope coefficient

The slope coefficient in this context represents that for each additional hour a student studies, on average, the exam score will increase by about **4.3152** points.

6. Relationship base on P-value

The p-value is **1.055e-08** which is extremely small and less then the common significant level of 0.05, we can say that the observed relationship between study hours and exam scores is **highly statistically significant**.

7. R-Squared Value

The R-squared Value is **0.986**. This indicates that approx. **98.6%** of the variability in exam scores can be explained by the number of study hours in the model. Also, since the R-Squared Value in very high, it also shows that the model is a great fit for the data.

8. Regression Line

```r
ggplot(df, aes(study_hours, exam_scores)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(title = "Exam Scores vs. Study Hours",
          x = "Study Hours",
          y = "Exam Scores") +
  theme_minimal()
```

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

9. Predicting Exam Score

```
new_data = data.frame(study_hours = 6.5)
predicted_score = predict(model1, newdata = new_data)
print(paste("Predicted Exam Score is:", predicted_score))
```

```
## [1] "Predicted Exam Score is: 81"
```

Question 2

```
temperature <- c(60, 65, 70, 75, 80, 85, 90)
coffees_sold <- c(40, 50, 60, 70, 80, 90, 100)

df2 = data.frame(temperature, coffees_sold)
df2
```

```
##   temperature coffees_sold
## 1          60           40
## 2          65           50
## 3          70           60
## 4          75           70
```

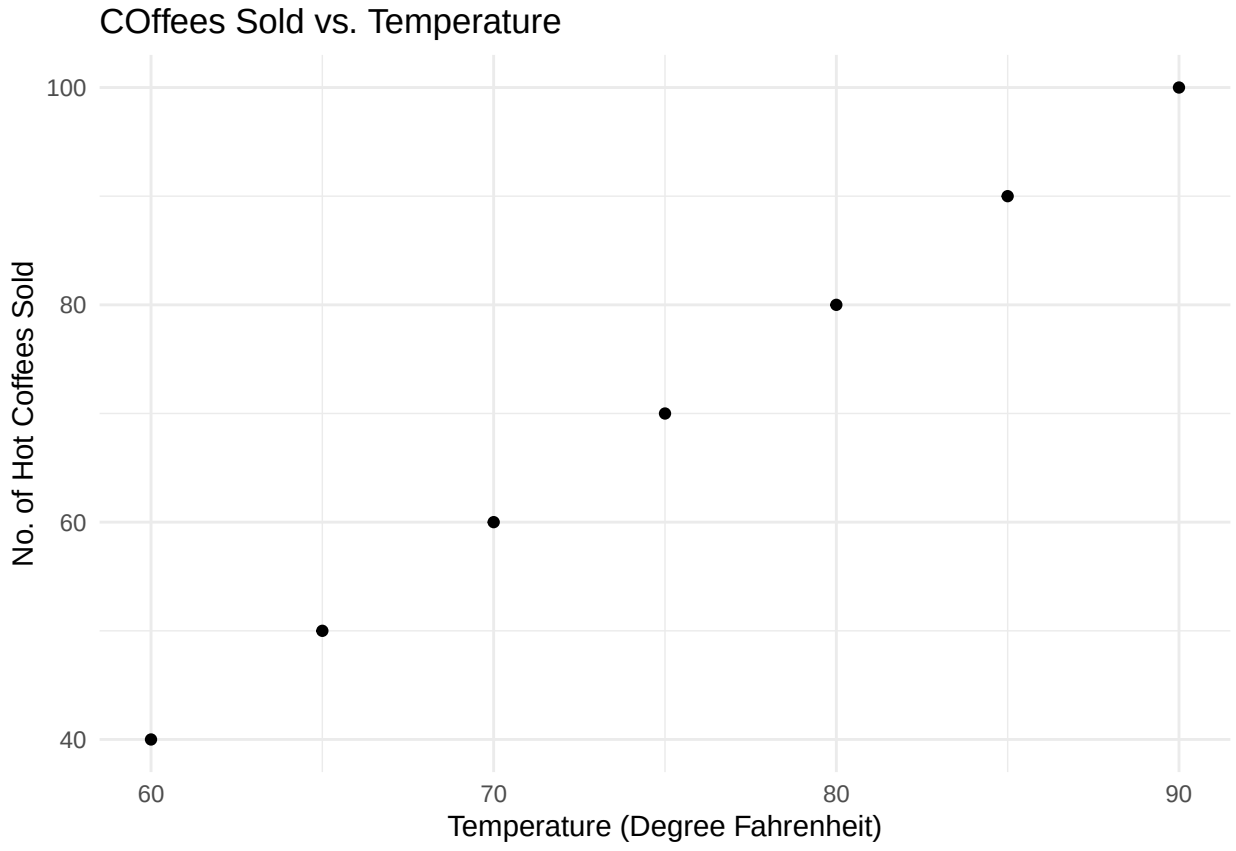
```
## 5           80           80
## 6           85           90
## 7           90          100
```

1. Identifying Variables

Temperature(Degree Fahrenheit) is the independent variable (x) as this variable influences the other variable **Coffees Sold**. On the other hand, **Coffees Sold** is dependent variable (y), and this is the variable we are trying to predict.

2. Visualizing the Data

```
ggplot(df2, aes(temperature, coffees_sold)) +
  geom_point() +
  labs(title = "COffees Sold vs. Temperature",
       x = "Temperature (Degree Fahrenheit)",
       y = "No. of Hot Coffees Sold") +
  theme_minimal()
```



From the above visual, we can see that there is a clear linear relationship exists - a perfect positive linear relationship. As the temperature increases, the number of hot coffees sold also increases.

3. Regression Equation

```
model2 <- lm(coffees_sold ~ temperature)
model2
```

```
##
## Call:
## lm(formula = coffees_sold ~ temperature)
##
## Coefficients:
## (Intercept)  temperature
##         -80            2
```

```
summary(model2)
```

```
## Warning in summary.lm(model2): essentially perfect fit: summary may be
## unreliable
```

```
##
## Call:
## lm(formula = coffees_sold ~ temperature)
##
## Residuals:
##          1          2          3          4          5          6          7
##  1.057e-14 -7.436e-15 -4.711e-15 -2.891e-15 -1.293e-15  3.858e-15  1.903e-15
##
## Coefficients:
##               Estimate Std. Error    t value Pr(>|t|)
## (Intercept) -8.000e+01  1.887e-14 -4.239e+15   <2e-16 ***
## temperature  2.000e+00  2.494e-16  8.018e+15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.6e-15 on 5 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 6.429e+31 on 1 and 5 DF,  p-value: < 2.2e-16
```

The intercept (c): **-80** The slope (m): **2**

Regression Equation: y = mx + c -> `Coffee Sold = (2 * Temperature) - 80`

4. Interpreting the Coefficients

The slope (m) is **2** and it represents that *for every one-degree increase in temperature*, on average, the coffee shop *is predicted to sell 2 additional hot coffees.*

The intercept (c) is **-80** and it is the predicted values when the temperature is 0 Degree Fahrenheit. The model predicts sales of -80 coffees at that temperature but this doesn't make sense as it has no practical meaning in this context.

5. Making a prediction

```
new_temp <- data.frame(temperature = 72)

predicted_coffee_sale <- predict(model2, newdata =  new_temp)

print(paste("Predicted Hot Coffee Sale is:", predicted_coffee_sale, "cups."))
```

```
## [1] "Predicted Hot Coffee Sale is: 64 cups."
```

6. Evaluating Limitations

```r
new_temp_2 <- data.frame(temperature = 30)

predicted_coffee_sale_2 <- predict(model2, newdata =  new_temp_2)

print(paste("Predicted Hot Coffee Sale is:", predicted_coffee_sale_2, "cups."))
```

```
## [1] "Predicted Hot Coffee Sale is: -20 cups."
```

Using this model to predict sales on a Temperature of 30 Degree Fahrenheit can be problematic as our model was built using data only within the temperature of 60 and 90 degree Fahrenheit. The linear trend may not hold at extremely low temperatures. Predictions outside the observed range are unreliable and nonsensical predictions, which is **-20 cups** in our case.