STA234 HW6

Derin Gezgin

2025-04-17

100/100 Great job!

Importing the Necessary Libraries

```
library(tidyverse)
```

Problem 1 [20 points]

A data set from kaggle(https://www.kaggle.com/szamil/who-suicide-statistics/home), which shows basic historical (1979-2016) data by country, year and demographic groups. Original data comes from WHO Mortality Database. Please use the data set named as who_suicide(download from moodle or email attachment), and work on the following questions:

1.1

Import the dataset to your R studio, and then use the following R functions to get an idea of the datase: `View()`: open and check the dataset; `head()`: look at only the first few rows of the dataset; `names()`: check all the variables of the data set.

```
suicide.data = read.csv("who_suicide.csv")

#View(suicide.data) # Not running this line in the Markdown

head(suicide.data)

##   country year    sex         age suicides_no population
## 1 Albania 1985 female 15-24 years          NA     277900
## 2 Albania 1985 female 25-34 years          NA     246800
## 3 Albania 1985 female 35-54 years          NA     267500
## 4 Albania 1985 female  5-14 years          NA     298300
## 5 Albania 1985 female 55-74 years          NA     138700
## 6 Albania 1985 female   75+ years          NA      34200

names(suicide.data)

## [1] "country"     "year"        "sex"         "age"         "suicides_no"
## [6] "population"
```

1.2

Create a new dataset, named as WHO_S, and make it includes 6 columns of data: country **with only Iceland and United States of America**, year, sex, age, suicides_no **without missing values,** population.

```
WHO_S = suicide.data %>%
    filter(country %in% c("Iceland", "United States of America"),
           !is.na(suicides_no))
```

We can use the `filter` function to select these specific countries and filter out the `NA` values in the `suicides_no` column.

1.3

Add a new column to WHO_S and name it as ratio. The new column ratio is defined as the ratio of suicides_no and population. For example, if ratio is 0.001, that means about one person suicided per 1000 people. Make sure you View() your dataset WHO_S to check your operations are correct.

```
WHO_S = WHO_S %>%
    mutate(ratio = suicides_no / population)

# View(WHO_S) # Not running this line in the Markdown
```

We can use the `mutate` function to create a new variable which is the ratio of `suicides_no` and `population`.

1.4

Use `table()` to check the observations that you have from each country of your new data set WHO_S.

```
table(WHO_S$country)

## 
##                    Iceland United States of America 
##                        432                      444
```

1.5

If you want to know two numbers to summarize the suicide ratio of each country, would you use mean/variance or median/range? please explain why, and create a table showing the summarizing information.

```
country_summary_table = WHO_S %>%
    group_by(country) %>%
    summarise(data_count = n(),
            suicide_ratio_variance = var(ratio, na.rm = TRUE),
            min_ratio = min(ratio, na.rm = TRUE),
            max_ratio = max(ratio, na.rm = TRUE),
            lowerQuantile = quantile(ratio, 0.25),
            upperQuantile = quantile(ratio, 0.75))

country_summary_table

## # A tibble: 2 × 7
##   country    data_count suicide_ratio_variance min_ratio max_ratio
lowerQuantile
##   <chr>           <int>                  <dbl>     <dbl>     <dbl>
<dbl>
## 1 Iceland           432           0.0000000189   0         0.000696     0
## 2 United St…        444           0.0000000175   1.53e-6   0.000590
0.0000416
## # ℹ 1 more variable: upperQuantile <dbl>
```

In this case, I would use median as outliers on lower/upper side of the data can result in a skewed mean while they will not have as significant effect in the median. Min/Max, 25th and 75th percentile values would also provide me additional important information on the distribution of the data.

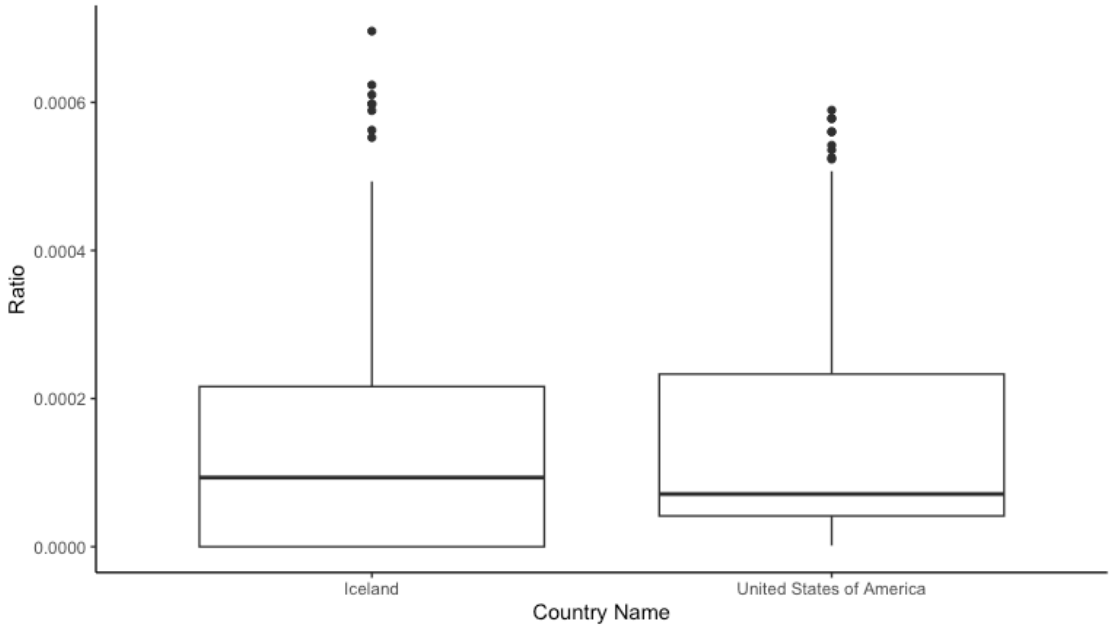
1.6

Draw a boxplot of ratio data to visualize the difference of Iceland and US. Summarize the important information from the boxplot.

```
suicide_bplot = ggplot(data = WHO_S,
                       aes(x = country,
                           y = ratio)) +
    geom_boxplot() +
    labs(title = "Fig.1 Comparision of Suicide Rates in Iceland and United
States of America",
         x = "Country Name",
         y = "Ratio",
         caption = "Source: World Health Organization") +
    theme_classic()

suicide_bplot
```

Fig.1 Comparision of Suicide Rates in Iceland and United States of America



Source: World Health Organization

From the boxplot, we can see that there are obvious outliers in the suicide ratio for both Iceland and United States of America. While the suicide ratios in Iceland has a higher median value, the 25th and 75th quantile values of Iceland's suicide ratios are lower.

1.7

Is suicide ratio from Iceland is different with the suicide ratio from U.S.? Explain.

Looking at Figure 1, it is not possible to say that they are very different than each other as they have a close median value, and 25th/75th percentile ranges. It is important to note that Iceland slightly has a larger 25th-75th range which points out to a greater variability.

1.8

Which factors do you think will have a difference in suicide ratio? Will age group affect that? or gender?

There can be many external factors that can have an effect in the suicide ratio such as gender, age, country of origin (in this case it does not have a significant effect), economical background, employment status. The year the data was collected can also have an effect on the suicide ratio as some periods in the history experienced social / economic crises that can increase the suicide ratio. In short, age group and/or gender would be one of the factors affecting the suicide ratio.

How would you check your suspicion?

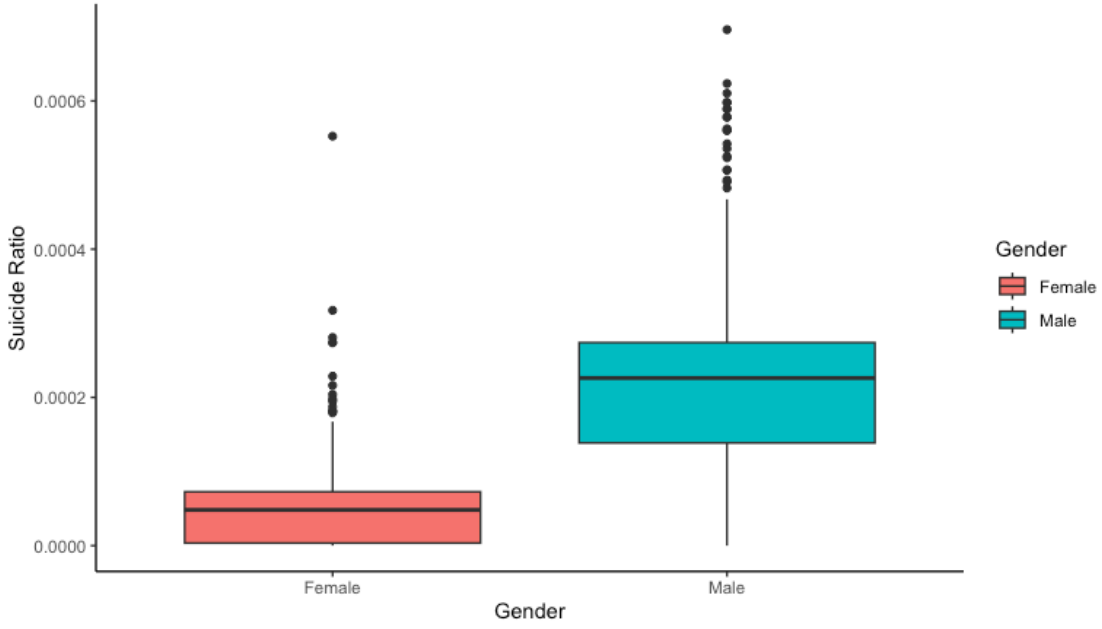
To check my suspicion, I should see the distribution of the suicide data for each variable of interest (in this case age and/or gender), to understand if that variable has a significant effect on the suicide ratio or not. I will make my interpretations in the following part.

I can start by creating a boxplot of suicide ratio colored by gender.

```
WHO_S$sex = factor(WHO_S$sex,
                   levels = c("female", "male"),
                   labels = c("Female", "Male"))
genderPlot = ggplot(data = WHO_S,
                    aes(x = sex,
                        y = ratio,
                        fill = sex)) +
  geom_boxplot() +
  labs(title = "Fig.2 Suicide Ratio Colored by Gender",
       x = "Gender",
       y = "Suicide Ratio",
       caption = "Source: World Health Organization",
       fill = "Gender") +
  theme_classic()

genderPlot
```

Fig.2 Suicide Ratio Colored by Gender



Source: World Health Organization

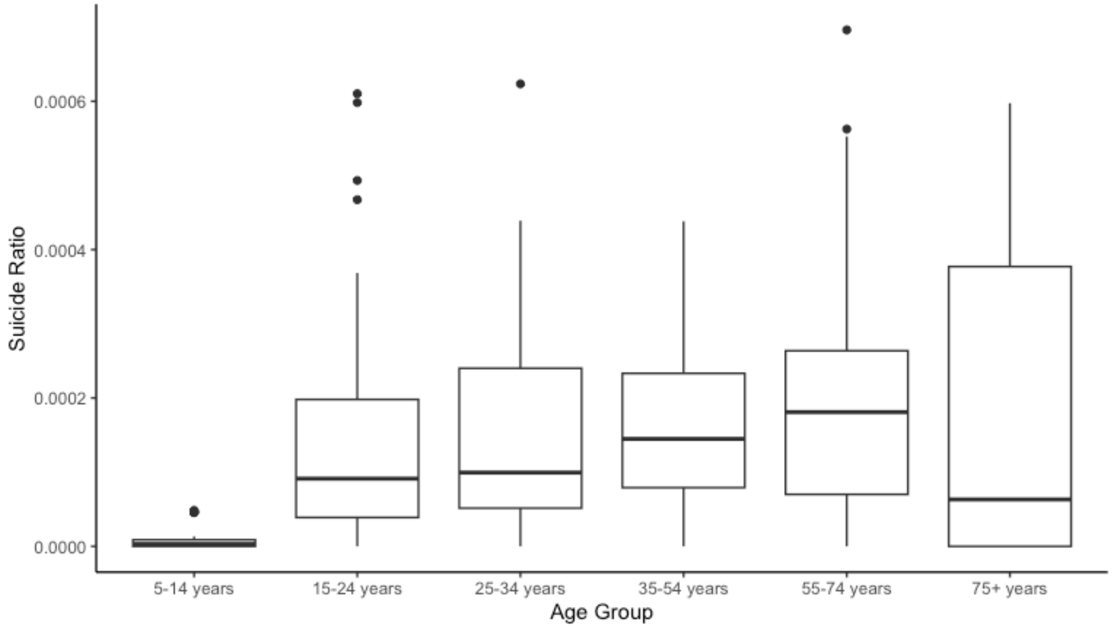
Following this, I can look at boxplot of suicide ratio for different age groups.

```
WHO_S$age = factor(WHO_S$age,
                   levels = c("5-14 years", "15-24 years", "25-34 years",
                              "35-54 years", "55-74 years", "75+ years"))

agePlot = ggplot(data = WHO_S,
                 aes(x = age,
                     y = ratio)) +
  geom_boxplot() +
  labs(title = "Fig.3 Suicide Ratio by Age Group",
       x = "Age Group",
       y = "Suicide Ratio",
       caption = "Source: World Health Organization") +
  theme_classic()

agePlot
```

Fig.3 Suicide Ratio by Age Group



Source: World Health Organization

Other than these plots, we can also get the mean suicide ratio values for different age / gender values

```
suicideByGender = WHO_S %>%
    group_by(sex) %>%
    summarise(median_ratio = median(ratio, na.rm = TRUE))

suicideByGender

## # A tibble: 2 × 2
##   sex    median_ratio
##   <fct>         <dbl>
## 1 Female    0.0000482
## 2 Male      0.000226
```

```
suicideByAge = WHO_S %>%
    group_by(age) %>%
    summarise(median_ratio = median(ratio, na.rm = TRUE))

suicideByAge
```

```
## # A tibble: 6 × 2
##   age         median_ratio
##   <fct>              <dbl>
## 1 5-14 years    0.00000315
## 2 15-24 years   0.0000912
## 3 25-34 years   0.0000994
## 4 35-54 years   0.000145
## 5 55-74 years   0.000181
## 6 75+ years     0.0000632
```

It is also possible to fit a linear regression model to see how different factors such as age and gender influence the suicide ratio. This model can further reveal the relationship between the variables.

And what is your conclusion?

We can see that the median suicide ratio (in Figure 2 and the first tibble) for male is significantly higher than females. Similarly, the 25th/75th percentile ranges are wider for males pointing out a higher variation in suicide ratios of males. Both genders have outliers but males have more and higher outliers, pointing out to extreme suicide ratios for man in certain years. I can conclude that gender has an effect on the suicide ratio.

When we look at Figure.3 and the second tibble, we can see that the suicide ratio increases from teens (15-24) to middle ages (35-74) peeking at the 55-74 range. In the 15-74 range, there is a clear and stable pattern of increasing medians of suicide ratios. On the other hand, this pattern ends as 75+ individuals have a significantly lower suicide ratio the 55-74, even 15-24. While 75+ individuals have the second lowest median value, they have the highest 75th percentile value and highest whiskers. This points out a significantly larger variability than the other age groups. In conclusion, I can definitely say that age is an important factor affecting the suicide ratio of individuals.

Problem 2 [50 points]

2.a [10 points]

Create a function which is called even_odd which checks if a given number is even or odd. If a non-number if given, the function must give the error “Input must be a number”. Run function with input (a) 10, (b) 7, and (c) “yes”.

```
even_odd = function(x) {
  if (!is.numeric(x))
```

```
    stop("Input must be a number")

  if (x %% 2 == 0)
    return("even")

  return("odd")
}

even_odd(10)

## [1] "even"

even_odd(7)

## [1] "odd"

# Commenting this line as it will interrupt the knitting
# even_odd("yes")
```

2.b [10 points]

Write a function that accepts a numeric vector value, in miles, and converts the value(s) to kilometers.

```
miles_to_km = function(miles) {
  kilometers = miles * 1.60934
  return(kilometers)
}
```

Testing the function

```
miles_to_km(100)

## [1] 160.934
```

2.c [10 points]

Take matrix “mym” and write a function which should return a new matrix which contains all the columns without an NA in it.

```
removeNACols = function(mym) {
    selectedCols = colSums(is.na(mym)) == 0
    cleanMatrix = mym[, selectedCols, drop = FALSE]
    return(cleanMatrix)
}
```

Testing the function

```
mym = matrix(c(1, 2, 3, 4, NA, 6, 7, 8, 9), nrow = 3)
mym
```

```
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2   NA    8
## [3,]    3    6    9
```

We can see our initial matrix has a column with a NA value in it

```
newMatrix = removeNACols(mym)
newMatrix
```

```
##      [,1] [,2]
## [1,]    1    7
## [2,]    2    8
## [3,]    3    9
```

We can use our new removeNACols function to remove this column. The output shows us the new matrix without that extra column.

2.d [10 points]

Write a function called invoice with input arguments pcs (which is the number of pieces) and unitprice and the function calculates the net price (pcs * unitprice) and gives a deduction of 10% for >25 pieces sold. Test the function for 56 pieces and unit price of 89$.

```
invoice = function(pcs, unitprice) {
  net_price = pcs * unitprice

  if (pcs > 25)
    net_price = net_price * 0.90

  return(net_price)
}

invoice(56, 89)
```

```
## [1] 4485.6
```

2.e [10 points]

Write the function called doreturn with input arguments x and y. The functions returns the following four calculations:

$$a = 5x + y \quad b = x + 7y \quad r = 3x + 9y \quad t = x/y - 115$$

```
doreturn = function(x, y) {
  a = 5 * x + y
  b = x + 7 * y
  r = 3 * x + 9 * y
  t = x / y - 115
  return(list(a = a, b = b, r = r, t = t))
}
```

Test for x=10 and y=15.

```
doreturn(10, 15)
```

```
## $a
## [1] 65
##
## $b
## [1] 115
##
## $r
## [1] 165
##
## $t
## [1] -114.3333
```

Problem 3 [30 points]

3.f [15 points]

Create a function which is called madness_check which is defined as the following: the madness_check function is related the input of a madness_grade. If the madness_grade is more than 7, print "Very mad", if the madness_grade is between 4 to 7, print "moderate mad", if the madness_grade is 1 to 3, print "light mad", if the madness_grade is 0, print "happy". A person is only allowed to input a number from 0 to 10, otherwise, print "Rate your madness using a number from 0 to 10!".

```
madness_check = function(madness_grade) {
  if (!is.numeric(madness_grade) || madness_grade < 0 || madness_grade > 10)
{
    print("Rate your madness using a number from 0 to 10!")
  } else if (madness_grade > 7) {
    print("Very mad")
  } else if (madness_grade >= 4) {
    print("Moderate mad")
  } else if (madness_grade >= 1) {
    print("Light mad")
  } else if (madness_grade == 0) {
    print("Happy")
  }
}
```

3.g [5 points]

Then, run the function when values of madness_grade are 10, 100, 0 separately. Check the information that is printed in your console is the right information or not. Make sure when you run madness_check(10) you get “Very mad”; when you run madness_check(100), you get “Please rate your madness using a number from 0 to 10!”; when you run madness_check(0) you get “happy”.

```
madness_check(0)

## [1] "Happy"

madness_check(10)

## [1] "Very mad"

madness_check(100)

## [1] "Rate your madness using a number from 0 to 10!"
```

3.h [10 points]

Suppose we surveyed 100 people and we get the following observed data, surveydata, create a dataframe called surveyresult which includes two columns: first column named as surveydata, is the original survey data; second column named as madness, is the madness result using the scales from question a. Show the first six and last six rows of your data survey result.

```
set.seed(100)
surveydata=sample(1:11, 100, replace=T)

surveyresult = data.frame(surveydata = surveydata,
                          madness = sapply(surveydata, madness_check))

## [1] "Very mad"
## [1] "Moderate mad"
## [1] "Moderate mad"
## [1] "Light mad"
## [1] "Very mad"
## [1] "Very mad"
## [1] "Moderate mad"
## [1] "Moderate mad"
## [1] "Moderate mad"
## [1] "Moderate mad"
## [1] "Moderate mad"
## [1] "Moderate mad"
## [1] "Light mad"
## [1] "Moderate mad"
## [1] "Moderate mad"
## [1] "Moderate mad"
## [1] "Rate your madness using a number from 0 to 10!"
## [1] "Very mad"
## [1] "Light mad"
## [1] "Light mad"
## [1] "Light mad"
## [1] "Very mad"
## [1] "Light mad"
## [1] "Very mad"
## [1] "Light mad"
## [1] "Light mad"
## [1] "Moderate mad"
## [1] "Moderate mad"
## [1] "Moderate mad"
## [1] "Moderate mad"
## [1] "Rate your madness using a number from 0 to 10!"
## [1] "Moderate mad"
## [1] "Very mad"
## [1] "Rate your madness using a number from 0 to 10!"
## [1] "Moderate mad"
## [1] "Light mad"
## [1] "Moderate mad"
## [1] "Moderate mad"
## [1] "Light mad"
## [1] "Rate your madness using a number from 0 to 10!"
## [1] "Moderate mad"
## [1] "Very mad"
## [1] "Very mad"
## [1] "Very mad"
## [1] "Moderate mad"
## [1] "Very mad"
## [1] "Rate your madness using a number from 0 to 10!"
## [1] "Moderate mad"
## [1] "Light mad"
## [1] "Very mad"
## [1] "Moderate mad"
## [1] "Moderate mad"
## [1] "Rate your madness using a number from 0 to 10!"
## [1] "Very mad"
## [1] "Light mad"
## [1] "Moderate mad"
## [1] "Light mad"
## [1] "Light mad"
## [1] "Rate your madness using a number from 0 to 10!"
## [1] "Moderate mad"
## [1] "Rate your madness using a number from 0 to 10!"
## [1] "Moderate mad"
## [1] "Moderate mad"
## [1] "Moderate mad"
## [1] "Light mad"
## [1] "Very mad"
## [1] "Very mad"
## [1] "Moderate mad"
## [1] "Light mad"
## [1] "Moderate mad"
## [1] "Light mad"
## [1] "Moderate mad"
## [1] "Moderate mad"
## [1] "Moderate mad"
## [1] "Moderate mad"
## [1] "Very mad"
## [1] "Very mad"
## [1] "Very mad"
## [1] "Very mad"
## [1] "Moderate mad"
## [1] "Very mad"
## [1] "Very mad"
## [1] "Moderate mad"
## [1] "Very mad"
## [1] "Very mad"
## [1] "Very mad"
## [1] "Rate your madness using a number from 0 to 10!"
## [1] "Moderate mad"
## [1] "Rate your madness using a number from 0 to 10!"
## [1] "Light mad"
## [1] "Very mad"
## [1] "Very mad"
## [1] "Very mad"
## [1] "Moderate mad"
## [1] "Moderate mad"
## [1] "Rate your madness using a number from 0 to 10!"
## [1] "Light mad"
## [1] "Very mad"
## [1] "Rate your madness using a number from 0 to 10!"
## [1] "Very mad"
```

I can directly create a dataframe using the `data.frame` function.

```
head(surveyresult)
```

```
##   surveydata      madness
## 1         10     Very mad
## 2          7 Moderate mad
## 3          6 Moderate mad
## 4          3    Light mad
## 5          9     Very mad
## 6         10     Very mad
```

These are the first 6 rows of my results.

```
tail(surveyresult)

##     surveydata                                        madness
## 95           5                                   Moderate mad
## 96          11 Rate your madness using a number from 0 to 10!
## 97           2                                      Light mad
## 98           8                                       Very mad
## 99          11 Rate your madness using a number from 0 to 10!
## 100          8                                       Very mad
```

These are the last 6 rows of my results.

Problem 4 [25 points]

Read the article WritingExample-1 (shared in Moodle) and report the following:

4.a [5 points]

Explain the research goals of the study.

This research focuses on a digital learning platform, CourseKata, and explores how student engagement with the material and student performance can be analyzed and improved with data analysis and natural language processing techniques. It analyses the student interaction with the material, pulse check responses, student performance, and student reflections. It also focuses on the correlation between students' self-reported satisfaction ratings with their actual performance.

4.b [2.5 points]

Explain the dataset used in the study.

The dataset used in this study comes from the CourseKata platform which is an online platform for college students used in statistics education using the textbook "Advanced Statistics with R". The data had 3 main types:

1. **Student Interaction Logs:** Log of how students interact with the platform, i.e. how long they spend on a chapter / activity
2. **Pulse Check Responses:** Short, informal surveys students complete throughout the course on how they feel about the material. This is not a knowledge-based test.
3. **End of Chapter Performance Scores:** Actual knowledge-based tests that measure how much of the material the student has understood. These tests are hold at the end of each chapter.

This dataset was cleaned and pre-processed using Python's Pandas library.

4.c [2.5 points]

Explain the methodology used.

The study uses wide-range of methods in the different parts of the study.

- Pandas library in Python to preprocess the data (removing missing values. filtering out outliers, normalizing values, etc.).
- Sperman's rank correlation to check if students' pulse check scores are correlated to their actual end-of-chapter assessment scores.
- The student feedback for each chapter was analyzed using t-SNE and a locally hosted Large Language Model (LLM). This feedback was grouped into clusters based on their meaning, and the local LLM is used to summarize each group to highlight their main themes.
- Lastly, various data visualization tools like boxplots, scatterplots, line charts, etc. were used to show the visual trends in the data.

4.d [15 points]

Briefly explain what is presented in the 11 figures and findings to answer the research questions.

Figure 1

Figure 1 shows the rated constructs: Expectancy, Utility, Intrinsic Value, and Course Cost; and their frequency. According to the paper, over 70% of the responses were positive across all constructs, which points out a generally positive student opinion toward course experience.

Figure 2

Figure 2 shows the end-of-chapter performance versus the self-reported expectancy. It shows the misalignment between these parameters as the stable expectancy trend does is not even closely followed by the actual performance of the student which fluctuates heavily. This plot shows that positive shift in the expectancy does not come with a positive change in the performance, showing a need for better feedback measures.

Figure 3

Figure 3 shows the visual map of student sentiments, which was summarised by a local Large Language Model (LLM). It helps us to identify dominant themes such as time-management concerns, understanding of the content, which can be used to tailor the platform specifically for these concerns.

Figure 4

Figure 4 shows us the distribution of pulse check responses for each chapter. This plot shows us an important insight as despite the rise in the difficulty, the pulse checks remain consistent, which shows a potential overestimation of skill by students.

Figure 5

Figure 5 can be interpreted with Figure 4 as it shows the significant decline in the student performance as the chapters progress. This Figure (5) contradicts with Figure 4 as it was showing constant student confidence, which does not align with this decline in performance.

Figure 6

Figure 6 shows the average time spent versos the End of Chapter scores for each chapter. Despite the engagement time stays similar throughout th chapters, the performance still decreases showing that the engagement time -alone- cannot assess the learning challenges.

Figure 7

Figure 7 shows that students watched over 90% of the video content, which shows that visual content (like video) has a high engagement rate, pointing out that the visual contents should be offered more to increase student engagement with the content.

Figure 8

Figure 8 shows the distribution of question types, and we can see that most of the questions (70%) are in multiple-choice type, and there are not many interactive or varying question types. The paper suggests that it is possible to vary out the question diversity to increase the student engagement and th effectiveness of the assessments.

Figure 9

Figure 9 maps the open-ended question responses into clusters based on their semantic similarity. This is used as an interactive environment for users to se why they were classified in that location.

Figure 10

Figure 10 shows the manual analysis of the cluster in Figure 9 highlighting different themes such as “time constraints”, “conceptual understanding”, etc. which can guide the further investigations and feature development.

Figure 11

Lastly, Figure 11 shows the LLM-generated summaries for each cluster in the Figure 9/10, which summarizes the overall theme of each cluster which can give a brief summary to help us understand the data.