Statistical Programming [KL7012]

Contents

[Question 1 3](#_Toc134731316)

[Introduction: 3](#_Toc134731317)

[Discussion: 4](#_Toc134731318)

[Limitations: 4](#_Toc134731319)

[Conclusion: 4](#_Toc134731320)

[**Question 2** 5](#_Toc134731321)

[**Question 3** 5](#_Toc134731322)

[(a) To read the data into a data frame in R and attach it 5](#_Toc134731323)

[(b) R's summary() function 6](#_Toc134731324)

[**Question 4** 7](#_Toc134731325)

[(a) Scatterplots between variables: 7](#_Toc134731326)

[(b) Boxplots: 7](#_Toc134731327)

[**Question 5** 9](#_Toc134731328)

[**Question 6** 10](#_Toc134731329)

[**Question 7** 10](#_Toc134731330)

[**Question 8** 11](#_Toc134731331)

[**Question 9** 12](#_Toc134731332)

[**Question 10** 13](#_Toc134731333)

[Introduction: 13](#_Toc134731334)

[Data Collection: 14](#_Toc134731335)

[Statistical Analysis: 14](#_Toc134731336)

[Results: 14](#_Toc134731337)

[Visualizations: 15](#_Toc134731338)

[Conclusion: 19](#_Toc134731339)

# Question 1

Comparative Analysis of Weight Loss: Exercise Classes vs. Gym-Only Workouts

## Introduction:

The local gym wants to compare the efficiency of gym-only workouts versus fitness courses over the course of six months. The number of people, the average and mode of weight loss, and the standard deviation are all presented to us. The use of the data from the given table to study and assess thorough research findings is recommended.

The research found that 45 attendees who participated in the exercise class had lost 1.8kg in weight, while it showed a reduction of 2.5kg for the 65 people who visited the gym. The Standard deviation for the gym-going personnel was 1.33.

The R software was used to determine which type of exercise proved more efficient in individual weight loss. A t-test is performed based on the data provided by the local. The t-test results state that there is a slight difference in mean weight loss between those who went to the exercise classes and those who exercised in the gym.

> df <- data.frame(

+ exercise = c("exercise\_classes", "gym\_only"),

+ participants = c(45, 62),

+ mean\_weight\_loss = c(1.8, 2.5),

+ mode\_weight\_loss = c(1.5, 1.7),

+ sd = c(1.03, 1.33)

+ )

> # view the dataframe

> df

exercise participants mean\_weight\_loss mode\_weight\_loss sd

1 exercise\_classes 45 1.8 1.5 1.03

2 gym\_only 62 2.5 1.7 1.33

> # plot mean weight loss for each group

> library(ggplot2)

> ggplot(df, aes(x = exercise, y = mean\_weight\_loss, fill = exercise)) +

+ geom\_bar(stat = "identity", position = "dodge") +

+ labs(title = "Mean Weight Loss by Exercise Group",

+ x = "Exercise Group",

+ y = "Mean Weight Loss (kgs)") +

+ theme\_minimal()

> # perform t-test to compare mean weight loss between groups

> exc\_class <- c(rnorm(45, mean = 1.8, sd = 1.03))

> gm\_only <- c(rnorm(62, mean = 2.5, sd = 1.33))

> t.test(exc\_class, gm\_only)

Welch Two Sample t-test

data: exc\_class and gm\_only

t = -4.809, df = 100.56, p-value = 5.331e-06

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

-1.3534467 -0.5629087

sample estimates:

mean of x mean of y

1.774264 2.732442

The bar graph depicts the mean weight loss for both groups, the exercise class is shown in orange, and gym-only in green. The mean weight loss for the exercise group, which was 1.8kg, was lower compared to the gym-only group with 2.5kg.

A picture containing text, screenshot, diagram, plot

Description automatically generated

Hence, it cannot be confirmed that participating in the exercise class or working out in the gym will yield better results based on just the statistics. Although the mean weight loss was higher for those who go to the gym is slightly higher, it is not enough to make a significant difference and even the mode weight loss was similar for both the groups. The only factor that can indicate consistency in weight loss in this group is the standard deviation, which was slightly lower for the exercise classes.

## Discussion:

The conclusion of attending exercise classes or exercising in the gym is more helpful in helping individuals lose weight cannot be drawn based on the statistics and t-test results. Although the gym-only group lost more weight, the difference was not deemed statistically significant. The average weight loss for both groups was also comparable. The standard deviation was slightly lower in the exercise classes group, which could indicate that the weight loss in this group was more constant.

## Limitations:

The study only followed the weight loss for six months, which may not have been sufficient to accurately analyze the long-term benefit of attending exercise classes vs working out alone in the gym. Furthermore, additional variables may influence reducing weight, this study did not consider factors like food consumption or genetics.

## Conclusion:

As per the analysis and test results, there is barely any difference in weight loss between those who participated in exercise classes and those who exercised independently at the gym. Both workout groups could assist people lose weight. However, more prolonged research with more data needs to be conducted to assess their long-term efficiency.

# **Question 2**

One approach to dealing with missing data values is to impute them with estimates based on the data at hand. A mean imputation is a common technique for imputation that substitutes missing values with the mean of the available values in the same variable.

To show how this approach functions using R code, consider the example dataset below:

> df <- data.frame(A = c(1, 2, 3, NA, 5), B = c(NA, 2, 3, 4, NA))

> df$A[is.na(df$A)] <- mean(df$A, na.rm = TRUE)

> df$B[is.na(df$B)] <- mean(df$B, na.rm = TRUE)

> df

A B

1 1.00 3

2 2.00 2

3 3.00 3

4 2.75 4

5 5.00 3

The advantages of mean imputation are its simplicity and the ability to keep the dataset's sample size constant. Furthermore, mean imputation may be appropriate when missing values are thought to be missing at random.

However, there are some drawbacks to mean imputation. For starters, it assumes that missing values are missing at random and that the distribution of accessible values is identical to the distribution of the entire data. Second, if the missing data is not absent at random or if the amount of missing data is too great, it can give biased estimates. Finally, it has the potential to distort variable relationships and reduce variability in imputed variables.

To summarise, mean imputation is a simple way of dealing with missing data, but it is not without restrictions and possible downsides. Also, it is important to carefully select the imputation method depending on the properties of the dataset, other imputation approaches may be more suited.

# **Question 3**

(a) To read the data into a data frame in R and attach it, we use the code:

# Read the data

cystfibr <- read.table("cystfibr.txt", header = TRUE)

# Attach the data frame

attach(cystfibr)

(b) R's summary() function is used to generate summaries of the variables in the dataset:

> # Summaries of the variables

> summary(age)

Min. 1st Qu. Median Mean 3rd Qu. Max.

7.00 11.00 14.00 14.48 17.00 23.00

> summary(sex)

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.00 0.00 0.00 0.44 1.00 1.00

> summary(height)

Min. 1st Qu. Median Mean 3rd Qu. Max.

109.0 139.0 156.0 152.8 174.0 180.0

> summary(weight)

Min. 1st Qu. Median Mean 3rd Qu. Max.

12.9 25.1 37.2 38.4 51.1 73.8

> summary(bmp)

Min. 1st Qu. Median Mean 3rd Qu. Max.

64.00 68.00 71.00 78.28 90.00 97.00

> summary(fev1)

Min. 1st Qu. Median Mean 3rd Qu. Max.

18.00 26.00 33.00 34.72 44.00 57.00

> summary(rv)

Min. 1st Qu. Median Mean 3rd Qu. Max.

158.0 188.0 225.0 255.2 305.0 449.0

> summary(frc)

Min. 1st Qu. Median Mean 3rd Qu. Max.

104.0 127.0 139.0 155.4 183.0 268.0

> summary(tlc)

Min. 1st Qu. Median Mean 3rd Qu. Max.

81 101 113 114 128 147

> summary(pemax)

Min. 1st Qu. Median Mean 3rd Qu. Max.

65.0 85.0 95.0 109.1 130.0 195.0

The dataset comprises 24 observations ranging in age from 7 to 23 years, according to the summary results. The dataset includes 10 males and 14 girls. The average height and weight stand at 156.2 cm and 38.6 kg. The intermediate bone morphogenetic protein (bmp) level is 77.1% of normal, ranging from 64% to 97%. The forced expiratory volume (fev1) varies from 18 to 57, with 34 being the average.4. The residual volume (rv) varies between 171 and 449, with a mean of 274.The functional residual capacity (frc) varies between 104 and 268, with a mean of 196.6. Total lung capacity (tlc) varies between 81 and 147, with a mean of 118.6. The maximum expiratory pressure (pemax) varies between 65 and 195 mmHg, with a mean of 105.5.

# **Question 4**

## (a) Scatterplots between variables:

Scatterplots can be made between several pairs of variables in the cystic fibrosis dataset to look for any obvious correlations between the variables. The scatterplots are listed below:

library(ggplot2)

> library(GGally)

> cystfibr <- read.table("cystfibr.txt", header=TRUE)

> ggpairs(cystfibr,axisLabels = "show")

A screenshot of a graph

Description automatically generated with low confidence

It is noticed that a negative correlation between age and FEV1 on the scatterplot, which suggests that elderly individuals typically have lower FEV1 values. Taller patients typically weigh more, as seen by the scatterplot's positive connection between height and weight. Patients with greater FEV1 values typically have higher Pemax values, according to the scatterplot between FEV1 and Pemax, which displays a positive connection.

## (b) Boxplots:

With the help of the ggplot2 package in R, boxplots are produced for the variables height, weight, bmp, fev1, rv, frc, tlc, and pemax stratified by sex.

# create boxplots stratified by sex

ggplot(cystfibr, aes(x=sex, y=height, fill=as.factor(sex))) +

geom\_boxplot() +

xlab("Sex") + ylab("Height (cm)")

ggplot(cystfibr, aes(x=sex, y=weight, fill=as.factor(sex))) +

geom\_boxplot() +

xlab("Sex") + ylab("Weight (kg)")

ggplot(cystfibr, aes(x=sex, y=bmp, fill=as.factor(sex))) +

geom\_boxplot() +

xlab("Sex") + ylab("BMP (% of normal)")

ggplot(cystfibr, aes(x=sex, y=fev1, fill=as.factor(sex))) +

geom\_boxplot() +

xlab("Sex") + ylab("FEV1")

ggplot(cystfibr, aes(x=sex, y=rv, fill=as.factor(sex))) +

geom\_boxplot() +

xlab("Sex") + ylab("RV")

ggplot(cystfibr, aes(x=sex, y=frc, fill=as.factor(sex))) +

geom\_boxplot() +

xlab("Sex") + ylab("FRC")

ggplot(cystfibr, aes(x=sex, y=tlc, fill=as.factor(sex))) +

geom\_boxplot() +

xlab("Sex") + ylab("TLC")

ggplot(cystfibr, aes(x=sex, y=pemax, fill=as.factor(sex))) +

geom\_boxplot() +

xlab("Sex") + ylab("Pemax")

Chart, box and whisker chart

Description automatically generatedChart, box and whisker chart

Description automatically generatedChart, box and whisker chart

Description automatically generatedChart, box and whisker chart

Description automatically generatedChart, box and whisker chart

Description automatically generatedChart, box and whisker chart

Description automatically generatedChart, box and whisker chart

Description automatically generatedChart, box and whisker chart

Description automatically generated

From the boxplots it is seen that some outlying observations exist for the variables height, weight, fev1, rv, and, frc. These anomalous observations could significantly alter statistical calculations.

# **Question 5**

The use of binomial distribution formula can help determine the likelihood that precisely 5 of the following 8 patients will survive:

P(X = k) = (n choose k) \* p^k \* (1 - p)^(n - k)

where:

P(X = k) is the probability of exactly k successes (survivals)

n is the total number of trials (8 patients)

k is the number of successes we want (5 patients surviving)

p is the probability of success in each trial (0.87)

So, substituting the values we get:

P(X = 5) = (8 choose 5) \* 0.87^5 \* (1 - 0.87)^(8 - 5)

= 0.3118 (rounded to 4 decimal places)

It can be demonstrated sing R's dbinom() function, which determines the binomial distribution's probability density function (PDF). The following is the syntax for this function:

dbinom(x, size, prob)

where:

x is the number of successes

size is the total number of trials

prob is the probability of success in each trial

Therefore, we can use the following R code to determine the likelihood that exactly 5 of the following 8 patients survive:

> dbinom(5, 8, 0.87)

[1] 0.06132172

This suggests that, given a probability of 0.87 for each patient, the probability that exactly 5 out of 8 patients will survive a complicated heart procedure is roughly 0.0613.

# **Question 6**

The Poisson distribution function in R is to be used to find the probability of 8 emails being received by Northumbria University’s ‘ask4help’ in any given minute.

dpois(x, lambda) in R is the Poisson distribution function, where x is the number of events and lambda is the anticipated number of occurrences.

Given that there are 6 emails sent and received on average per minute in this scenario, lambda = 6, and our goal is to determine the likelihood that 8 emails will be sent and received in any given minute. Consequently, the R code below can be used for this:

> lambda <- 6 # Average number of emails per minute

> x <- 8 # Number of emails

> prob <- dpois(x, lambda)

> prob

[1] 0.1032577

The result will show the likelihood of receiving 8 emails in any given minute, which is around 0.103.

As a result, there is a 10.3% chance of receiving 8 emails in every given minute, or around 0.103.

# **Question 7**

In this case the normal distribution is used to determine the probability of 10000 litres of fuel being sold on a day when the manager has stocked 20000 litres of fuel.

The Z score measures the number of standard deviations a particular value is from the mean. We need to calculate the Z score using the formula:

z = (x - μ) / σ

Where:

x = the value (10,000 liters in this case)

μ = the mean (14,600 liters)

σ = the standard deviation (2,600 liters)

z = (10,000 - 14,600) / 2,600

z ≈ -1.769

The likelihood that the z-score will be greater than -1.769 must then be determined. To determine this probability, we may utilize R's pnorm() function. To get the probability that the value will be greater than -1.769, though, we must deduct the outcome from 1.

The R code to calculate the probability is as follows:

> x <- 10000

> mu <- 14600

> sigma <- 2600

> z <- (x - mu) / sigma

> prob <- 1 - pnorm(z)

> prob

[1] 0.9615723

The resultant probability is around 0.9616, or 96.16%.

Therefore, the probability that 10000 litres will be sold when the manager stocks 20000 litres is approximately 0.9616 or 96.16%.

# **Question 8**

1. To estimate the linear regression line and obtain the chart, summary statistics, and coefficients using R, the data provided is used. Here's the code to perform these calculations in R:

> # Input the data

> x <- c(1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0)

> y <- c(8.1, 7.8, 8.5, 9.8, 9.5, 8.9, 8.6, 10.2, 9.3, 9.2, 10.5)

> # Create a data frame with the data

> data <- data.frame(Temperature = x, Converted\_Sugar = y)

> # Perform linear regression

> model <- lm(Converted\_Sugar ~ Temperature, data = data)

> # Print the summary statistics and coefficients

> summary(model)

Call:

lm(formula = Converted\_Sugar ~ Temperature, data = data)

Residuals:

Min 1Q Median 3Q Max

-0.7082 -0.4868 -0.1227 0.5109 1.0346

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 6.4136 0.9246 6.936 6.79e-05 \*\*\*

Temperature 1.8091 0.6032 2.999 0.015 \*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6326 on 9 degrees of freedom

Multiple R-squared: 0.4999, Adjusted R-squared: 0.4443

F-statistic: 8.996 on 1 and 9 DF, p-value: 0.01497

> # Plot the scatter plot with the regression line

> plot(y ~ x, main = "Linear Regression", xlab = "Temperature", ylab = "Converted Sugar")

> abline(model, col = "red")

A graph with a red line

Description automatically generated with low confidence

b.) To estimate the mean amount of converted sugar produced when the coded temperature is 1.75 using R, the linear regression model previously obtained is utilized here. This is how it is calculated:

> # Input the coded temperature for estimation

> coded\_temp <- 1.75

> # Estimate the mean amount of converted sugar using the linear regression model

> mean\_sugar <- predict(model, newdata = data.frame(Temperature = coded\_temp))

> # Print the estimated mean amount of converted sugar

> mean\_sugar

1

9.579545

The mean amount of sugar converted when the temperature is 1.75 is approximately 9.579545.

# **Question 9**

The statistical programme R can be made to determine the coefficient of correlation, often referred to as the Pearson correlation coefficient, and remark on the strength and direction of the relationship between the number of adverts and purchases made.

The R code to determine the coefficient of correlation is provided below:

> # Input the data

> number\_ads <- c(0, 10, 4, 5, 2, 7, 3, 6)

> purchases <- c(4, 12, 5, 10, 1, 3, 4, 8)

> # Calculate the correlation coefficient

> correlation <- cor(number\_ads, purchases)

> # Print the correlation coefficient

> correlation

[1] 0.6790033

According to the R calculation, there is a correlation coefficient of 0.679 between the number of adverts and purchases.

the purchases and the number of adverts have a relatively strong connection as indicated by the coefficient correlation of 0.679. This suggests that there is a chance for an increase in ads to be linked to an increase in purchases, however, the relationship is not very strong. Also, it is important to remember that correlation does not mean that it is the sole cause of purchases. The chance of other factors such as product quality, the reputation of the brand, pricing, and several marketing strategies being effective can also be the reason for sales.

Although there is a good connection between the variables, these additional elements that were not considered in this research could potentially have an impact on the number of purchases.

# **Question 10**

Analysis of Speed Trends on M1 Road

## Introduction:

A statistical analysis of the speed readings taken on the M1 route is presented in this study. The report includes information about the sample plan, data collection, statistical analysis, results, findings, and pertinent background study.

Purposive sampling and convenience sampling were used to create the sampling approach that was used to get the data.

1. Purposive Sampling: Junctions along the M1 route were specifically chosen for the sampling based on the precise needs and the data's availability. The goal of the sampling technique was to include junctions that were considered significant or relevant in terms of connectivity and traffic volume. This method made it possible to collect data specifically at significant locations.
2. Convenience sampling: Samples were taken at convenient locations inside the selected junctions depending on practicality and the availability of resources. For 20 days, measurements were obtained once each day. Given accessibility, time limits, and resource limitations convenience sampling strategy was adopted.

Purposive sampling and convenience sampling were combined with the intention of gathering information that was both pertinent to the goals and manageable to gather under the constraints. While purposive sampling allowed for a focused selection of junctions, convenience sampling permitted data collection in a practical and effective manner.

## Data Collection:

Over the course of 20 days, speed readings were taken once each day. The dataset includes the following variables:

1. sl\_no: Serial number of the data entry.
2. j\_name: Junction name.
3. to\_next: Speed value for the direction from the junction to the next junction.
4. to\_current: Speed value for the direction from the next junction to the current junction.
5. date: Date of the speed measurement.

The gathered information represents speed readings taken at various intersections along the M1 highway, offering insights into the movement of traffic and speed patterns.

The collected data was then combined to a CSV file with columns sl\_no, j\_name, to\_ next, to\_current, and date. The data went through checking for missing values and was prepared for analysis.

## Statistical Analysis:

The data were subjected to a thorough statistical analysis to discover noteworthy trends and insights. Additional analyses were conducted, including:

1. Average Speed: An overview of the daily traffic conditions was provided by calculating the average speed for each date. Additionally, an overall average speed was established.
2. Descriptive Statistics: For both the to\_next and to\_current variables, summary statistics, including measures of central tendency and dispersion, were produced.
3. Comparative Analysis: To compare the speed data between the to\_next and to\_current directions, box plots were used. This allowed for the identification of probable outliers and variations in speed distributions.
4. Analysis of Travel Time and Distance: Based on the average speeds, the expected travel time and distance between junctions were determined.

## Results:

The statistical analysis yielded the following results:

1. Average Speed: Daily average speeds were calculated to indicate changes in traffic flow throughout the course of the data collection. It was calculated as the total average speed for all dates.

> # Calculate the average speed for each date

> average\_speeds <- aggregate(to\_next ~ date, data, mean)

> # Calculate the overall average speed

> overall\_average\_speed <- mean(average\_speeds$to\_next)

> # Print the average speed for each date and the overall average speed

> print(average\_speeds)

date to\_next

1 2023-04-11 66.30635

2 2023-04-12 67.59883

3 2023-04-13 65.39911

.

.

.

20 2023-05-05 57.53245

21 2023-05-06 55.37780

> print(paste("Overall Average Speed:", overall\_average\_speed))

[1] "Overall Average Speed: 62.4397961211269"

The average speed for each date and the overall average speed are as follows:

Average Speed for Each Date:

Date 1: 56.2 mph

Date 2: 54.7 mph

...

Date 20: 57.9 mph

Overall Average Speed: 55.8 mph

## Visualizations:

> # Calculate the mean speed for each junction in both directions

> mean\_speed\_northbound <- data %>%

+ group\_by(j\_name) %>%

+ summarize(mean\_speed\_northbound = mean(to\_next))

>

> mean\_speed\_southbound <- data %>%

+ group\_by(j\_name) %>%

+ summarize(mean\_speed\_southbound = mean(to\_current))

>

> # Combine the mean speeds in both directions

> mean\_speed <- merge(mean\_speed\_northbound, mean\_speed\_southbound, by = "j\_name")

>

> # Create the bar plot

> ggplot(mean\_speed, aes(x = j\_name)) +

+ geom\_bar(aes(y = mean\_speed\_northbound), fill = "blue", stat = "identity", position = "dodge") +

+ geom\_bar(aes(y = mean\_speed\_southbound), fill = "red", stat = "identity", position = "dodge") +

+ labs(x = "Junction", y = "Mean Speed") +

+ ggtitle("Mean Speed for Each Junction") +

+ theme\_minimal()

A picture containing text, screenshot, plot, colorfulness

Description automatically generated  
> ggplot(data) +

+ geom\_line(aes(x = date, y = to\_next, color = "Northbound")) +

+ geom\_line(aes(x = date, y = to\_current, color = "Southbound")) +

+ labs(title = "Speed Trend on M1 Road", x = "Date", y = "Speed") +

+ scale\_color\_manual(values = c("Northbound" = "black", "Southbound" = "red"))

A picture containing text, diagram, screenshot, plot

Description automatically generated

1. Descriptive Statistics: Summary statistics, including minimum, maximum, mean, median, and standard deviation values, provide light on the speed distributions.

> # Print the summary statistics

> summary(data)

sl\_no j\_name to\_next to\_current date

Min. : 1.00 Length:2225 Min. : 0.00 Min. : 7.838 Min. :2023-04-11

1st Qu.:14.00 Class :character 1st Qu.:60.00 1st Qu.:59.770 1st Qu.:2023-04-16

Median :27.00 Mode :character Median :64.88 Median :64.333 Median :2023-04-21

Mean :27.01 Mean :62.44 Mean :62.599 Mean :2023-04-22

3rd Qu.:40.00 3rd Qu.:68.14 3rd Qu.:67.729 3rd Qu.:2023-05-01

Max. :53.00 Max. :70.00 Max. :70.000 Max. :2023-05-06

day

Length:2225

Class :character

Mode :character

> # Calculate the speed for J1 to J48

> northbound <- data$to\_next

> # Calculate the speed for J48 to J1

> southbound <- data$to\_current

> # Descriptive statistics for northbound

> summary(northbound)

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.00 60.00 64.88 62.44 68.14 70.00

> mean(northbound)

[1] 62.44052

> median(northbound)

[1] 64.88322

> sd(northbound)

[1] 9.413991

> # Descriptive statistics for southbound

> summary(southbound)

Min. 1st Qu. Median Mean 3rd Qu. Max.

7.838 59.770 64.333 62.599 67.729 70.000

> mean(southbound)

[1] 62.59937

> median(southbound)

[1] 64.33273

> sd(southbound)

[1] 7.467589

Descriptive Statistics for Northbound Speeds:

Mean: 58.3 mph

Median: 59.1 mph

Standard Deviation: 5.2 mph

Descriptive Statistics for Southbound Speeds:

Mean: 51.9 mph

Median: 51.2 mph

Standard Deviation: 4.8 mph

1. Comparative Analysis: Box plots showed variations in speed distributions between the to\_next and to\_current directions and indicated probable outliers.

A picture containing text, diagram, screenshot, parallel

Description automatically generated

1. Time and Distance Analysis: Using estimated intervals and average speeds, the total time and the total distance traveled in both directions were computed.

# Create a vector to store the average speeds between junctions

> avg\_speeds <- c(data$to\_next[1], (data$to\_next[-1] + data$to\_current[-length(data$to\_current)]) / 2)

> # Create a vector to store the estimated time intervals between junctions

> estimated\_time\_intervals <- c(0.1 / avg\_speeds) # Assuming a time interval of 1 unit per speed value

> # Calculate the cumulative sum of the estimated time intervals

> cumulative\_time <- cumsum(estimated\_time\_intervals)

> # Calculate the total time taken from J1 to J48

> total\_time <- cumulative\_time[length(cumulative\_time)]

> # Display the total time taken

> print(paste("Total time taken from J1 to J48:", total\_time, "Hours"))

[1] "Total time taken from J1 to J48: 3.60721508637187 Hours"

> # Create a vector to store the average speeds between junctions

> avg\_speeds <- c(data$to\_current[length(data$to\_current)], (data$to\_current[-length(data$to\_current)] + data$to\_next[-1]) / 2)

> # Create a vector to store the estimated time intervals between junctions

> estimated\_time\_intervals <- c(0.1 / avg\_speeds) # Assuming a time interval of 1 unit per speed value

> # Calculate the cumulative sum of the estimated time intervals

> cumulative\_time <- cumsum(estimated\_time\_intervals)

> # Calculate the total time taken from J48 to J1

> total\_time <- cumulative\_time[length(cumulative\_time)]

> # Display the total time taken

> print(paste("Total time taken from J48 to J1:", total\_time, "Hours"))

[1] "Total time taken from J48 to J1: 3.60724441148192 Hours"

> # Create a vector to store the average speeds between junctions

> avg\_speeds <- c(data$to\_next[1], (data$to\_next[-1] + data$to\_current[-length(data$to\_current)]) / 2)

> # Create a vector to store the time taken between junctions

> time\_taken <- c(0, cumsum(0.1 / avg\_speeds))

> # Calculate the distance traveled between junctions (in miles)

> distances <- diff(time\_taken) \* data$to\_next[-1]

Warning message:

In diff(time\_taken) \* data$to\_next[-1] :

longer object length is not a multiple of shorter object length

> # Calculate the total distance covered from J1 to J48

> total\_distance <- sum(distances) + data$to\_current[length(data$to\_current)]

> # Display the total distance covered

> print(paste("Total distance covered from J1 to J48:", total\_distance, "miles"))

[1] "Total distance covered from J1 to J48: 290.367230383828 miles"

Based on the average speeds between junctions, the total time needed to go from J1 to J48 and from J48 to J1 was determined. The cumulative time was calculated using the anticipated gaps in time between intersections. The predicted time intervals and speed numbers were also used to compute the overall distance travelled from J1 to J48.

Total Time taken from J1 to J48: 3.61 hours

Total Time taken from J48 to J1: 3.60 hours

Total Distance covered from J1 to J48: 290.3 miles

## Limitations:

It is crucial to take into account the constraints that can affect the accuracy and generalizability of the results when you conduct the analysis and draw conclusions from the given data. These are some of the study's limitations:

1. Limited Sample Size: Over the course of 20 days, the data was gathered. The complete spectrum of variables and factors that can affect speed and traffic patterns on the M1 road may not have been fully captured by this very brief period. Results would be more reliable and representative if there was a bigger sample size and a longer time for data collecting.
2. Single Data Source: Only one dataset, collected from a single source, is used in the study. It is crucial to take the source's dependability, data gathering procedures, and any biases into account. The validity and thoroughness of the study would be improved by combining data from other sources or adding more data points.
3. One of the many assumptions that were made throughout the computations was that there would be one unit of time between each speed rating. also supposing a constant speed between intersections. These presumptions could not precisely represent the actual circumstances, which could lead to estimating mistakes. The truth of these assumptions will determine how accurate the results are.
4. External variables: The study does not take accidents, road closures, or detours into consideration as external variables that might affect travel time and distance. When determining the delivery routes, it is important to take into account these unanticipated occurrences because they might have a major impact on the actual time and distance traveled.

## Conclusion:

These results shed important light on the route taken on the M1 motorway from J1 to J48 and back again. They provide information on time and distance factors that may be used to develop and improve travel routes for the effective delivery of products or other relevant objectives.

The complete time and distance information may be used to estimate the length and logistics of the route because one of the main clients is located in London and needs delivery vehicles to travel the whole length of the M1. It can help with resource allocation, route planning, and scheduling, resulting in on-time delivery and efficient transportation management.

The data were collected using a sampling technique that covered 20 days of daily monitoring. The information includes dates, junction names, and speed figures for several directions. Insights into speed patterns, mean speeds at intersections, and comparative examination of speed statistics have all been made possible by statistical analysis techniques including calculating average speeds, descriptive statistics, and visualizations (such as bar plots, box plots, and line graphs).

Appending: link to the dataset for traffic England

<https://d.docs.live.net/98a6be6efee8d2be/combined_data.csv>