Omoegbemwen Aigbe 2301397

2023-10-15

**Solution 1**

**1a. Equation that describes the area of gel electrophoresis as a function of its breadth.**

Let A represent area, B represent breath, and L represent length.

L =2 \*B

A = L\* B, therefore A = 2\* B \* B

A = 2B2

1b. **If the area of the gel is 8 cm2, the dimensions of the gel are shown, and the calculations are explained.**

A = 8cm2

Recall that A = 2B2

therefore, 8 = 2B2

To solve this equation, use the balance method. Whatever is done to the equation's Right-Hand Side (RHS) is done to the Left-Hand Side (LHS).

Divide both sides by 2 =>

8/2 = 2B2/ 2

4 = B2

Take the square root of both sides ==>

√4 = (√(B)2

B = ± 2

B cannot be negative because it represents distance. The final answer is B = 2cm. If Breath is 2cm, the length is L =2\*B, L=4cm. Dimension (L= 4cm, B= 2cm)

**1c. R function to calculate the area of the gel for breadth of 7.5cm**

# Given relationship: Length is twice the size of breadth  
# L = 2B  
  
# the breadth (B) is 7.5cm   
B <- 4  
  
# Calculate the length (L)  
L <- 2 \* B  
  
# Calculate the area of the rectangle  
area <- L \* B  
  
# Print the result  
cat("The area of the rectangle is:", area, "and the dimensions are (", L, ", ", B, ")\n")

## The area of the rectangle is: 32 and the dimensions are ( 8 , 4 )

**Solution 2**

**2a. Two equations: one to describe the total amount of fluid in an hour and one to describe the ratio of fluids given in an hour.**

Total amount

Let X = x\*60, Y = y\*60.

X + Y = 1000 (equation 1a) or [x\*60 + y\*60 = 1000(equation 1b)]

This is the total of 1000ml to be given in an hour.

Ratio for X

X/ (X + Y) = 3/ (3+2)

X/ (X + Y) = 3/ (5) (equation 2)

The equation shows the ratio relationship of fluid given in an hour.

**2b. Solving the simultaneous equation.**

using Equation 1a and Equation 2

X + Y = 1000 (equation 1a)

X/ (X + Y) = 3/ 5 (equation 2)

solving simultaneously using the substitution method

for equation 1:

let, X = 1000 - Y (making x the subject of the formula, subtract both sides by ‘Y’)

X = 1000 - Y (equation 3)

for equation 2:

X/(X + Y) = 3/ 5 ( we cross multiply to have a linear equation)

5X = 3(X+Y) (equation 4)

recall that X = 1000 - Y

substitute equation 3 into equation 4

5(1000 - Y) = 3((1000 - Y) +Y)

5000 - 5Y = 3(1000 -Y+Y)

5000 - 5Y = 3000

collect like times. 5000 moves to the other side of the equation

when that happens, 5000 becomes negative.

-5Y = 3000 - 5000

-5Y = -2000

divide both sides by -5.

-5Y/-5 = -2000 /-5

Y= 400

from equation 1:

X + Y = 1000

X + 400 = 1000

Collect like times. 400 moves to the other side of the equation

When that happens, 400 becomes negative.

X = 1000 - 400

X = 600

The patient must be administered 600 ml of saline and 400 ml of dextrose in one hour.

x= (600/ 60) saline is 10ml per minute.

y= (400/ 60) dextrose is 6.67ml per minute.

**2c.**

# Create a system of equations  
A <- matrix(c(1, 1, 3/5, -2/5), ncol = 2, byrow = TRUE)  
B <- c(1000, 0)  
  
# Solve the equations  
solution <- solve(A, B)  
  
Y <- solution[1]  
X <- solution[2]  
  
X # Saline rate

## [1] 600

Y # Dextrose rate

## [1] 400

cat(paste("X=", X, "ml","Y=", Y, "ml"))

## X= 600 ml Y= 400 ml

# Define the coefficients and constants  
A <- matrix(c(60, 60, 3, -2), ncol = 2, byrow = TRUE)  
B <- c(1000, 0)  
  
# Use the solve() function to find X and Y  
solution <- solve(A, B)  
  
Y <- solution[1]  
X <- solution[2]  
  
X # Saline rate

## [1] 10

Y # Dextrose rate

## [1] 6.666667

cat(paste("X=", X, "ml per min", "Y=",Y, "ml per min"))

## X= 10 ml per min Y= 6.66666666666667 ml per min

**Solution 3**

**3a.**

𝑠 = 0.005𝑡2 + 𝑣0𝑡

collect like terms by moving s to the other side of the equation.

0.005𝑡2 + 𝑣0𝑡 – 𝑠 = 0

using s = 1000 and 𝑣0= 4

substitute into the formula

0.005𝑡^2 + 4t - 1000 = 0

using the quadratic formula

t =

a = 0.005, b = 4, c = -1000

t =

t = ) / 0.01

t = (-4 ± 6) / 0.01

t = (-4 + 6) / 0.01 or (-4 - 6) / 0.01

t = 200s or -1000s

time cannot be negative, so t =200s

**3b. The R code plots a graph and gives the two math mathematical solutions. However, one of the solutions is not possible physically.**

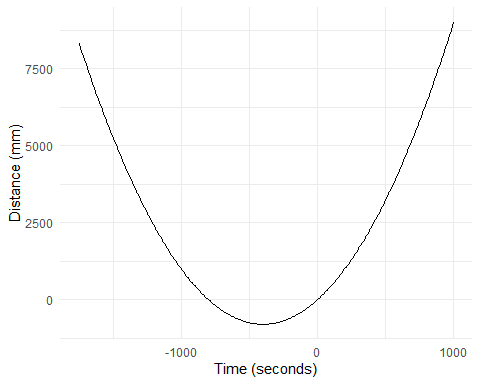
# Constants  
a <- 0.005  
b <- 4  
c <- -1000  
  
# Calculate the two possible values of t using the quadratic formula  
root1 <- (-b + sqrt(b^2 - 4\*a\*c)) / (2\*a)  
root2 <- (-b - sqrt(b^2 - 4\*a\*c)) / (2\*a)  
  
# Display both roots  
cat("Root 1:", root1, "seconds\n")

## Root 1: 200 seconds

cat("Root 2:", root2, "seconds\n")

## Root 2: -1000 seconds

library(ggplot2)  
  
# Create a sequence of time values  
time\_values <- seq(-1750, 1000, by = 10)  
  
# Calculate distance using the equation  
distance\_values <- 0.005 \* time\_values^2 + 4 \* time\_values  
  
# Create a data frame  
data <- data.frame(time = time\_values, distance = distance\_values)  
  
# Create the plot  
ggplot(data, aes(x = time, y = distance)) +  
 geom\_line() +  
 labs(x = "Time (seconds)", y = "Distance (mm)") +  
 theme\_minimal()



Discriminant ==> b2 - 4ac.

The problem has two possible solutions because the nature of the equation is quadratic. This means that in a quadratic equation, two solutions can answer a question mathematically. The amount of solutions we have is determined by the value of the discriminant. If the discriminant is positive (>0), then there are two distinct values; if the discriminant is equal to zero (=0), then there is only one solution, and if the discriminant is negative (<0), then there is no real solution. In this case, although we have two solutions (-1000s and 200s), only one solution is possible from a physical perspective.

**Solution 4**

**4a.** **Data frame ‘df’ created using the R programming language.**

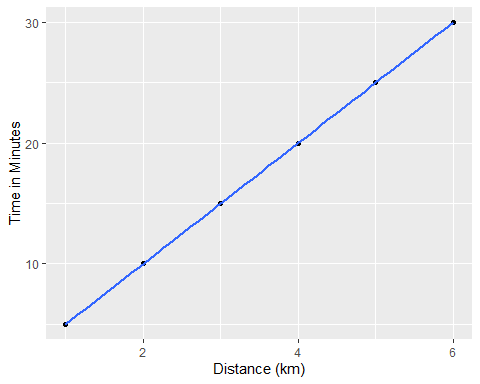
Distance\_km <- c(1, 2, 3, 4, 5, 6)  
time\_minutes <- c(5, 10, 15, 20, 25, 30)  
df<- data.frame(Distance\_km = Distance\_km, time\_minutes = time\_minutes)  
  
print(df)

## Distance\_km time\_minutes  
## 1 1 5  
## 2 2 10  
## 3 3 15  
## 4 4 20  
## 5 5 25  
## 6 6 30

**4b. Plotting a graph of distance(km) against time(minutes)**

plot <- ggplot(data = df, aes(x = Distance\_km, y = time\_minutes)) +  
 geom\_point() +   
 geom\_smooth(method = "lm", se = FALSE) +   
 labs(x = "Distance (km)", y = "Time in Minutes")   
  
print(plot)

## `geom\_smooth()` using formula = 'y ~ x'



**4c. Calculating the formula for a line of best fit.**

my\_model <-lm(time\_minutes ~ Distance\_km, data = df)  
model\_summary <- summary(my\_model)

## Warning in summary.lm(my\_model): essentially perfect fit: summary may be  
## unreliable

model\_summary

##   
## Call:  
## lm(formula = time\_minutes ~ Distance\_km, data = df)  
##   
## Residuals:  
## 1 2 3 4 5 6   
## 1.825e-15 -1.668e-15 -2.597e-16 1.834e-16 -2.038e-15 1.958e-15   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.901e-15 1.754e-15 1.654e+00 0.174   
## Distance\_km 5.000e+00 4.504e-16 1.110e+16 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.884e-15 on 4 degrees of freedom  
## Multiple R-squared: 1, Adjusted R-squared: 1   
## F-statistic: 1.232e+32 on 1 and 4 DF, p-value: < 2.2e-16

intercept <- model\_summary$coefficients[1]  
slope <- model\_summary$coefficients[2]  
  
my\_line <- paste("y =", slope, "(x) +", intercept)  
print(my\_line)

## [1] "y = 5 (x) + 2.90077857175577e-15"

*Code 2: this code was adjusted to have the y-intercept equal to zero.*

# Perform linear regression  
my\_model <-lm(time\_minutes ~ Distance\_km, data = df)  
  
  
intercept <- model\_summary$coefficients[1]  
slope <- model\_summary$coefficients[2]  
  
# If the intercept is close to zero, display it as zero  
if (abs(intercept) < 1e-10) {  
 intercept <- 0  
}  
  
  
cat("The equation of the line of best fit is:")

## The equation of the line of best fit is:

cat(paste("y =", round(slope, 2), "x +", round(intercept, 2)))

## y = 5 x + 0

**4d.**

If someone walks for 21km

using y = 5\*x

where y is time in minutes

y = 5\*21

y = 105 minutes

This person has been walking for 105 minutes.

**Solution 5**

**5a. Equation to calculate the total surface area (TSA).**

The block of wood has five faces: two triangle faces, two big rectangle faces and one small rectangle face.

TSA = 2*area of triangle + 2* area of RECTANGLE + area of rectangle

area of triangle = 2(1/2\**4x\**3x)

area of RECTANGLE = 2(5x\*y)

area of rectangle = 3x\*y

3600 = 12x2 + 10xy + 3xy

3600 = 12x2 + 13xy

**5b. Make y subject of the formula.**

Make y subject of the formula.

Subtract 12x2 from both sides.

3600 - 12x2 = 12x2 + 13xy - 12x2

factorize the RHS.

3600 - 12x2 = y(13x)

Divide both sides of the equation by (13y)

(3600 - 12x2)/(13x) = [y(13x)]/(13x)]

y = (3600 - 12x2) / 13xy

5c. producing a graph.

# Load the ggplot2 package  
library(ggplot2)  
  
# Create a sequence of x values (excluding 0 to avoid division by zero)  
x\_values <- seq(0.1, 10, by = 0.1)  
  
# Calculate corresponding y values  
y\_values <- (3600 - 12 \* x\_values^2) / (13 \* x\_values)  
  
# Create a data frame  
data <- data.frame(x = x\_values, y = y\_values)  
  
# Create the plot  
plot <- ggplot(data, aes(x = x, y = y)) +  
 geom\_line() +  
 labs(x = "x", y = "y")   
  
# Print the plot  
print(plot)

