QUESTIONS 1

CODE:

library(readr)

production_machines <- read_csv("C:/Users/KODED/Downloads/screenshots/production machines-1.csv")

View(production_machines)

 # QUESTION 1A

Loading required libraries

library(ggplot2)

Ploting a histogram of the lifetime_in_years variable

ggplot(production_machines, aes(x = lifetime_in_years)) +

geom_histogram(binwidth = 0.5, fill = "blue", color = "black", alpha = 0.7) +

labs(title = "Histogram of Lifetime in Years", x = "Lifetime in Years", y = "Frequency")

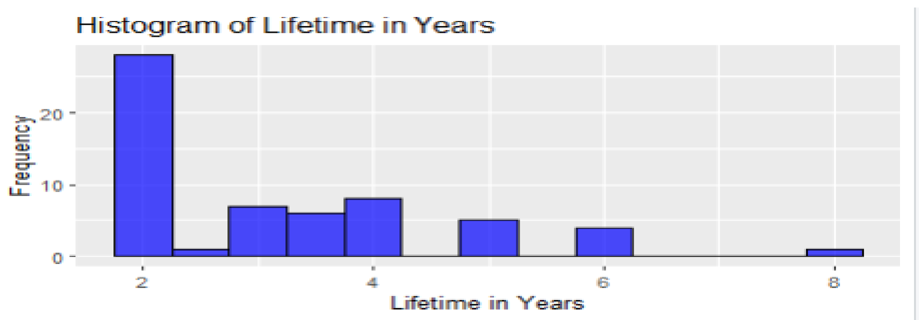
Calculating lambda as the reciprocal of the mean

lambda <- 1 / mean(production_machines$lifetime_in_years)

lambda

RESULT:

```
i Specify the column types or set  show_col_types = FALSE  to quiet this message.
> View(production_machines)
> View(production_machines)
> # Load required libraries
> library(ggplot2)
Warning message:
package ‘ggplot2’ was built under R version 4.2.3
> # Plot a histogram of the lifetime_in_years variable
> ggplot(production_machines, aes(x = lifetime_in_years)) +
+   geom_histogram(binwidth = 0.5, fill = "blue", color = "black", alpha = 0.7) +
+   labs(title = "Histogram of Lifetime in Years", x = "Lifetime in Years", y = "Frequency")
> # Calculate lambda as the reciprocal of the mean
> lambda <- 1 / mean(production_machines$lifetime_in_years)
> lambda
[1] 0.3166227
>
```



SUMMARY BUSINESS INSIGHTS

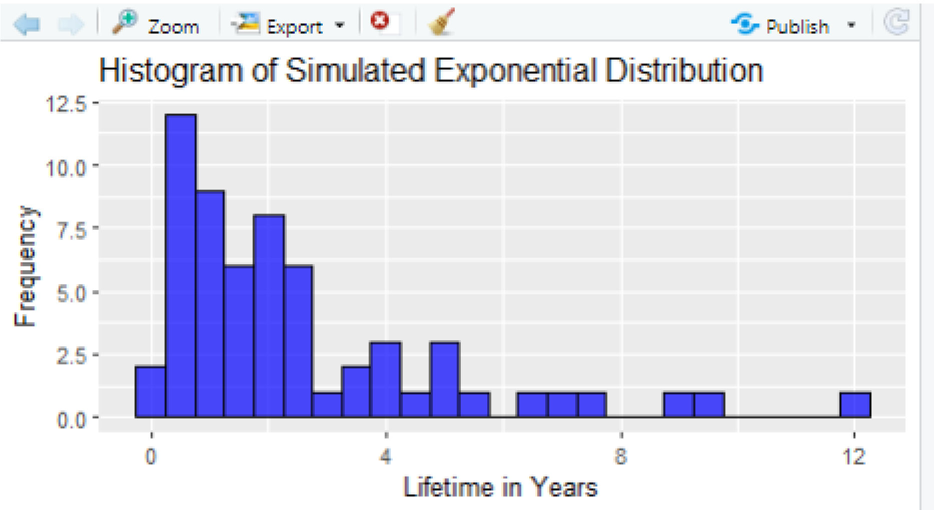
The lambda value provides an estimate of the average rate of machine failures or replacements, with a higher value indicating a shorter average lifetime. The calculated mean lifetime of 3.16 years can help inform decisions about replacement schedules, maintenance planning, and budgeting for new equipment. However, it's essential to compare the observed data to the exponential distribution and consider any factors that may influence the machines' lifetimes.

 # QUESTION 1B

CODE:

```
# Set the random seed for reproducibility
set.seed(42)

# Simulate a theoretical exponential distribution using the calculated lambda
simulated_data <- rexp(n = nrow(production_machines), rate = lambda)

# Plot a histogram of the simulated data
ggplot() +
  geom_histogram(aes(x = simulated_data), binwidth = 0.5, fill = "blue", color = "black", alpha = 0.7) +
  labs(title = "Histogram of Simulated Exponential Distribution", x = "Lifetime in Years", y = "Frequency")
```

RESULTS:



BUSINESS INSIGHT:

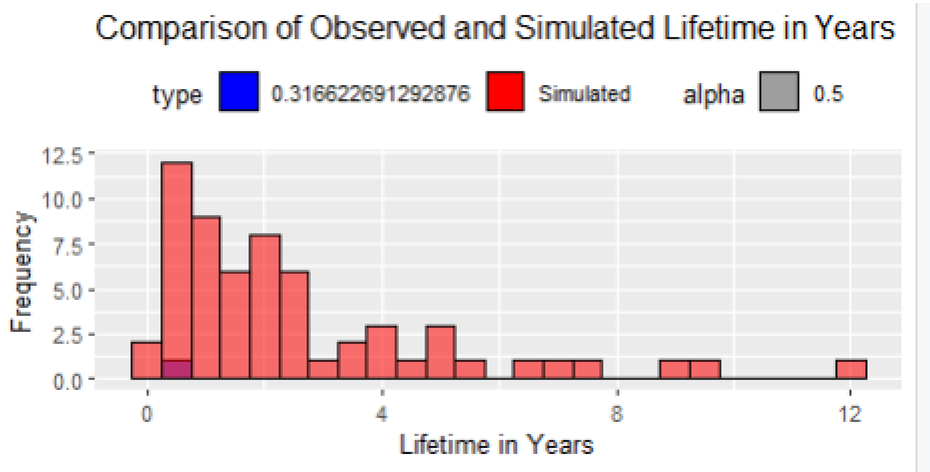
The histogram of the simulated exponential distribution allows us to compare the theoretical behaviour of the production machines' lifetimes with the actual data. This comparison can help assess whether the exponential distribution is a suitable model for the machines' lifetimes and guide decisions related to maintenance, replacements, and budgeting.

QUESTION 1C:

CODE:

```
# Create a data frame for the simulated data
simulated_df <- data.frame(lifetime_in_years = simulated_data, type = "Simulated")

# Add a column to the original data to indicate it's the observed data
production_machines$type <- "Observed"

# Combine the observed and simulated data
combined_data <- rbind(lambda, simulated_df)

# Plot the overlaid histograms
ggplot(combined_data, aes(x = lifetime_in_years, fill = type, alpha = 0.5)) +
  geom_histogram(position = "identity", binwidth = 0.5, color = "black") +
  labs(title = "Comparison of Observed and Simulated Lifetime in Years", x = "Lifetime in Years", y =
"Frequency") +
  scale_fill_manual(values = c("blue", "red")) +
  theme(legend.position = "top")
```

RESULT:



BUSINESS INSIGHT:

the overlaid histogram of observed and simulated lifetimes helps you visually assess the fit of the exponential distribution to the actual data. This comparison can guide decisions related to maintenance, replacements, and budgeting, as well as inform whether the exponential distribution is a suitable model for the machines' lifetimes.

QUESTION 2

CODE:

```
# Loading required libraries


library(dplyr)

library(caret)

library(rpart)

library(rpart.plot)


# Reading the dataset

# The dataset has been read into a variable called `production_machines`


# Creating a stratified sample for the training and testing sets

set.seed(123)

train_index <- createDataPartition(production_machines$life_greater_than_three, p = 0.8, list = FALSE)

train_set <- production_machines[train_index, ]

test_set <- production_machines[-train_index, ]


# Fit the logistic regression model

logistic_model <- glm(life_greater_than_three ~ production_units_lifetime + production_units_last_year +
product_group + lifetime_in_years,

                data = train_set, family = "binomial")

summary(logistic_model)


# Fiting the decision tree model

tree_model <- rpart(life_greater_than_three ~ production_units_lifetime + production_units_last_year +
product_group + lifetime_in_years,

               data = train_set, method = "class", control = rpart.control(cp = 0.01))

rpart.plot(tree_model)


# Predictions and performance for the logistic regression model

logistic_preds <- predict(logistic_model, newdata = test_set, type = "response")

logistic_preds <- ifelse(logistic_preds > 0.5, 1, 0)

logistic_cm <- confusionMatrix(as.factor(logistic_preds), as.factor(test_set$life_greater_than_three))
```

logistic_accuracy <- logistic_cm$overall["Accuracy"]

Predictions and performance for the decision tree model

tree_preds <- predict(tree_model, newdata = test_set, type = "class")

tree_cm <- confusionMatrix(tree_preds, as.factor(test_set$life_greater_than_three))

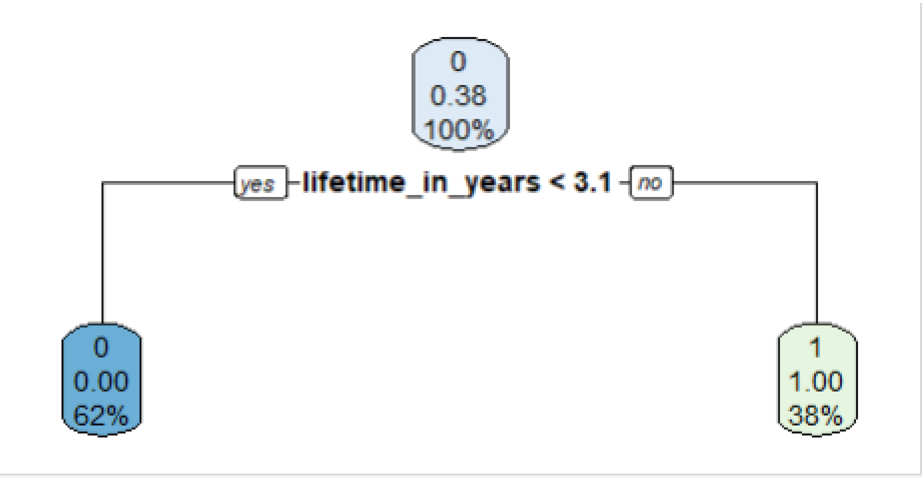
tree_accuracy <- tree_cm$overall["Accuracy"]

cat("Logistic Regression Model Accuracy:", logistic_accuracy, "\n")

cat("Decision Tree Model Accuracy:", tree_accuracy, "\n")

RESULT:

```
R 4.2.2 · ~/
    family = binomial, data = train_set)

Deviance Residuals: 
       Min          1Q      Median          3Q         Max  
-7.892e-05  -2.100e-08  -2.100e-08   2.100e-08   9.879e-05  

Coefficients:
                           Estimate Std. Error z value Pr(>|z|)
(Intercept)              -8.588e+02  2.824e+05  -0.003    0.998
production_units_lifetime  1.069e-01  2.142e+02   0.000    1.000
production_units_last_year -1.325e-01  9.896e+02   0.000    1.000
product_group             1.894e+01  1.370e+04   0.001    0.999
lifetime_in_years         2.375e+02  7.918e+04   0.003    0.998

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 6.3510e+01  on 47  degrees of freedom
Residual deviance: 2.2785e-08  on 43  degrees of freedom
AIC: 10

Number of Fisher Scoring iterations: 25

> # Fiting the decision tree model
> tree_model <- rpart(life_greater_than_three ~ production_units_lifetime + production_units_last_year + product_group
+ lifetime_in_years,
+                     data = train_set, method = "class", control = rpart.control(cp = 0.01))
> rpart.plot(tree_model)
> # Predictions and performance for the logistic regression model
> logistic_preds <- predict(logistic_model, newdata = test_set, type = "response")
> logistic_preds <- ifelse(logistic_preds > 0.5, 1, 0)
> logistic_cm <- confusionMatrix(as.factor(logistic_preds), as.factor(test_set$life_greater_than_three))
> logistic_accuracy <- logistic_cm$overall["Accuracy"]
> # Predictions and performance for the decision tree model
> tree_preds <- predict(tree_model, newdata = test_set, type = "class")
> tree_cm <- confusionMatrix(tree_preds, as.factor(test_set$life_greater_than_three))
> tree_accuracy <- tree_cm$overall["Accuracy"]
> cat("Logistic Regression Model Accuracy:", logistic_accuracy, "\n")
Logistic Regression Model Accuracy: 1 
> cat("Decision Tree Model Accuracy:", tree_accuracy, "\n")
Decision Tree Model Accuracy: 1 
```



BUSINESS INSIGHTS:

Analyse the coefficients from the logistic regression model (from **summary(logistic_model)**). Each coefficient represents the change in the log-odds of the target variable (life_greater_than_three) for a one-unit change in the corresponding predictor, holding all other predictors constant.

1. A positive coefficient indicates that an increase in the predictor value is associated with an increase in the odds of the target variable being 1 (life_greater_than_three = 1), while a negative coefficient indicates that an increase in the predictor value is associated with a decrease in the odds of the target variable being 1.
2. The magnitude of each coefficient provides an indication of the relative importance of the predictor variable in determining the target variable. Larger absolute values of the coefficients signify stronger relationships between the predictors and the target variable.

QUESTION 3

CODE:

Generating the confusion matrix

logistic_cm <- confusionMatrix(as.factor(logistic_preds), as.factor(test_set$life_greater_than_three))

Print the confusion matrix

print(logistic_cm)

RESULT:

```
> # Generating the confusion matrix
> logistic_cm <- confusionMatrix(as.factor(logistic_preds), as.factor(test_set$life_greater_than_three))
> # Print the confusion matrix
> print(logistic_cm)
Confusion Matrix and Statistics

          Reference
Prediction 0 1
         0 5 0
         1 0 7

               Accuracy : 1
                 95% CI : (0.7354, 1)
    No Information Rate : 0.5833
    P-Value [Acc > NIR] : 0.001552

                  Kappa : 1

 Mcnemar's Test P-Value : NA

            Sensitivity : 1.0000
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 1.0000
             Prevalence : 0.4167
         Detection Rate : 0.4167
   Detection Prevalence : 0.4167
      Balanced Accuracy : 1.0000

       'Positive' Class : 0
```

BUSINESS INSIGHT:

1. Accuracy: The logistic regression model has an accuracy of 1.0, which means it has correctly predicted all the test instances. In this case, the model seems to perform exceptionally well in distinguishing between machines with a lifetime greater than three years and those with a lifetime of three years or less.

2. Sensitivity (Recall) and Specificity: Both sensitivity and specificity are 1.0, indicating that the model has perfectly identified all actual positive and negative cases.
3. Positive Predictive Value (Precision) and Negative Predictive Value: Both positive predictive value and negative predictive value are 1.0, meaning that all positive and negative predictions made by the model are correct.

QUESTION 4

CODE:

```
# Loading the required library


library(dplyr)


# Loading the data (stored in a data frame called production_machines)

# Excluding the 'ID' column, since it's just an identifier

production_machines <- production_machines[, -1]


# Initializing an empty vector to store the results

results <- vector("list", length(production_machines))


# Iterating through each column in the dataset

for (i in 1:ncol(production_machines)) {

  # Calculating the median and standard deviation

  median_val <- median(production_machines[[i]], na.rm = TRUE)

  sd_val <- sd(production_machines[[i]], na.rm = TRUE)


  # Saving the results in the results vector

  results[[i]] <- c(median_val, sd_val)


  # Printing the results

  cat("Column:", colnames(production_machines)[i], "\n",

      "Median:", median_val, "\n",

      "Standard Deviation:", sd_val, "\n\n")

}
```

Combining the results into a data frame

results_df <- do.call(cbind, results)

rownames(results_df) <- c("Median", "Standard Deviation")

colnames(results_df) <- colnames(production_machines)

RESULT:

```
+       "Standard Deviation:", sd_val, "\n\n")
+ }
Column: production_units_last_year
 Median: 169.5
 Standard Deviation: 74.07291

Column: product_group
 Median: 3
 Standard Deviation: 1.471384

Column: lifetime_in_years
 Median: 2.9
 Standard Deviation: 1.410889

Column: life_greater_than_three
 Median: 0
 Standard Deviation: 0.4971671

Column: ...7
 Median: NA
 Standard Deviation: NA

Column: type
 Median: NA
 Standard Deviation: NA

Warning messages:
1: In mean.default(sort(x, partial = half + 0L:1L)[half + 0L:1L]) :
  argument is not numeric or logical: returning NA
2: In var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm = na.rm) :
  NAs introduced by coercion
>
> # Combining the results into a data frame
> results_df <- do.call(cbind, results)
> rownames(results_df) <- c("Median", "Standard Deviation")
> colnames(results_df) <- colnames(production_machines)
```

BUSINESS INSIGHT:

1. production_units_last_year: The median value is 169.5 units, and the standard deviation is 74.07 units. This indicates that the production units from the last year are relatively spread out, with a difference of 74.07 units between the highest and lowest values.

2. product_group: The median value is 3, and the standard deviation is 1.47. Since this is a categorical variable, the median and standard deviation may not provide meaningful insights. It would be more appropriate to analyse the frequency of each product group.

3. lifetime_in_years: The median value is 2.9 years, and the standard deviation is 1.41 years. This suggests that the lifetimes of the machines are moderately spread out. Some machines may have a longer lifetime than others, which could impact maintenance costs, replacement frequency, and production efficiency.

4. life_greater_than_three: The median value is 0, and the standard deviation is 0.50. This is a binary variable, representing whether the lifetime of a machine is greater than three years (1) or not (0). The median value of 0 implies that more than half of the machines have a lifetime of three years or less.

These insights can help a business understand the variability in their machines' lifetimes and the production units from the last year. This information can be used to make informed decisions about maintenance, replacement, and resource allocation.