|  |  |
| --- | --- |
| **PRACTICAL**  (Based on MSMS 301, MSMS 302, MSMS 303, MSMS304, MSMS306, MSMS306) | |
| NAME | AYANTANI PAUL |
| CLASS ROLL NUMBER | 09 |
| EXAM ROLL NUMBER | 23419STC011 |
| PAPER TITLE | PRACTICAL |
| PAPER CODE | MSMS -308 |
| COURSE | STATISTICS & COMPUTING |

Table of Contents

# MSMS – 301 Time Series Analysis

1. Practical 1

# MSMS – 302 Statistical Machine Learning

1. Practical 1
2. Practical 2
3. Practical 3

# MSMS – 303 Multivariate Analysis

1. Practical 1
2. Practical 2

# MSMS – 304 Life Time Data Analysis

1. Practical 1
2. Practical 2
3. Practical 3

# MSMS – 307 SAS and Computing

1. Practical 1
2. Practical 2

# Biostatistics

1. Practical 1
2. Practical 2
3. Practical 3
4. Practical 4

# MSMS – 301 Time Series Analysis

### Practical 1

For the “Nile” dataset available in R, obtain the results given below by writing a suitable R program

1. Autocorrelation up to order three
2. Partial autocorrelation
3. Also plot these using suitable diagram

## CODE:

library(datasets) data=Nile plot(data) require(graphics)

data1<-ts(data,start=c(1871,1), end=c(1970,12), frequency=12)

decomp<-decompose(data1) # to decompose the data into different components plot(decomp)

# (a) Autocorrelation up to order three Autocorr<-acf(data1,lag.max = 3,plot=FALSE)

Autocorr

# (b) Partial Autocorrelation

Partial\_Autocorr<-acf(data1,lag.max = 3,type=c("partial"),plot=FALSE)

Partial\_Autocorr

# (c) Plot the autocorrelation and partial autocorrelation functions par(mfrow = c(2, 1)) # Set layout for two plots in one column acf(data1, lag.max = 10, main = "Autocorrelation Function") pacf(data1, main = "Partial Autocorrelation Function")

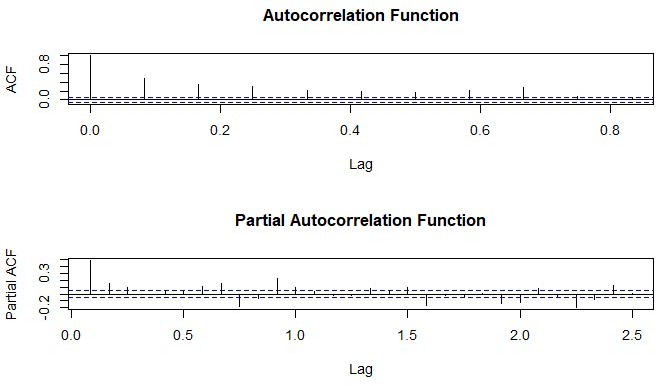
## OUTPUT:

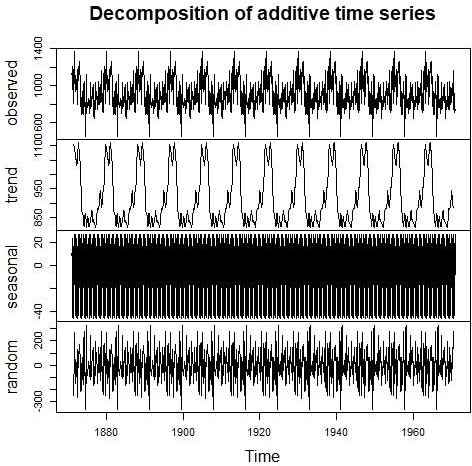
>Autocorrelations of series ‘data1’, by lag 0.0000 0.0833 0.1667 0.2500

1.000 0.487 0.357 0.296

>Partial autocorrelations of series ‘data1’, by lag 0.0833 0.1667 0.2500

0.487 0.158 0.0986





# MSMS – 302 Statistical Machine Learning

### Practical 1

1. For the following dataset, obtain kernel density estimate and Naïve density estimator. Also plot both the estimators.

|  |  |  |
| --- | --- | --- |
| 5.65746599 | 5.38283914 | 2.79892121 2.85423660 2.95252721 5.42626667 |
| 7.66239113 | -0.18001073 | 0.65083500 2.40276530 -0.09929884 6.32619215 |
| 5.03650752 | 2.07470777 | 1.78019174 6.12891558 4.05352439 2.02686971 |
| 3.50834853 -2.76449768 | | 4.98428763 3.01292677 2.82448038 3.98110437 |
| 5.09371862 5.97961648 | | 4.56968496 -0.48814532 5.08736697 2.41757609 |

## CODE:

# Dataset

data <- c(5.65746599, 5.38283914, 2.79892121, 2.85423660, 2.95252721, 5.42626667,

7.66239113,-0.18001073, 0.65083500, 2.40276530,-0.09929884, 6.32619215,

5.03650752, 2.07470777, 1.78019174, 6.12891558, 4.05352439, 2.02686971,

3.50834853,-2.76449768, 4.98428763, 3.01292677, 2.82448038, 3.98110437,

5.09371862, 5.97961648, 4.56968496,-0.48814532, 5.08736697, 2.41757609)

# Kernel Density Estimate kde <- density(data)

kde

# Naïve Density Estimator naive\_density <- function(x, data, h) { n <- length(data)

return(sum((abs(x- data) <= h) / (2 \* h)) / n)

}

# Naïve Density Estimator for a range of x

h <- 1.0 # Bandwidth for Naïve Density Estimation

x\_range <- seq(min(data)- 1, max(data) + 1, length.out = 100) naive\_estimates <- sapply(x\_range, naive\_density, data = data, h = h)

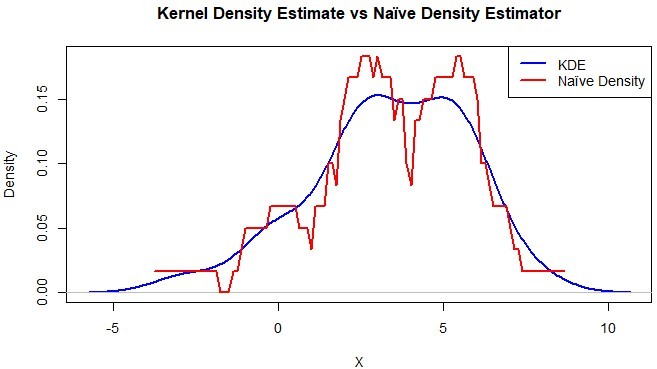
# Plotting both

plot(kde, main = "Kernel Density Estimate vs Naïve Density Estimator",

col = "blue", lwd = 2, ylim = c(0, max(kde$y, naive\_estimates)), xlab = "X", ylab = "Density") lines(x\_range, naive\_estimates, col = "red", lwd = 2)

legend("topright", legend = c("KDE", "Naïve Density"), col = c("blue", "red"), lty = 1, lwd = 2)

### Graph:

****

**Interpretation:**

KDE (blue) and the Naïve Density Estimator (red) are contrasted in the graph.

* The KDE efficiently captures the general shape and offers a smooth and reliable estimation of the data distribution.

Because of its restricted bandwidth and sensitivity to data points,

* The Naïve Density Estimator is imprecise and unable to accurately depict the distribution. KDE is favored because it represents density in a more fluid and understandable manner.

### Practical 2

1. Hosmer & Lerneshow (1989) give a dataset (“birthwt” available in R MASS library) on 189 births at a US hospital, with the main interest being in low birth weight. The main variable of interest is low birth weight, a binary response variable low. You can use variable “low” as binary response variable and remining variables as regressor variable. Divide the whole dataset into training and test dataset as solve perform following task
   1. Learn logistic classification model from training dataset and predict response using test dataset predictors.
   2. Obtain specificity, sensitivity, positive predictive value, negative predictive value of the model using test data set.

### Code:

data("birthwt", package = "MASS") birthwt <- birthwt

birthwt$low <- as.numeric(birthwt$low)

# Split data into training and test sets

train\_indices <- sample(1:nrow(birthwt), size = 0.7 \* nrow(birthwt)) # 70% training train\_data <- birthwt[train\_indices, ]

test\_data <- birthwt[-train\_indices, ]

# Logistic function logistic <- function(x) { 1 / (1 + exp(-x))

}

# Fit logistic regression using Newton-Raphson fit\_logistic <- function(X, y, max\_iter = 100, tol = 1e-6) { n <- nrow(X)

p <- ncol(X)

# Initialize beta coefficients

beta <- matrix(0, nrow = p, ncol = 1) epsilon <- 1e-8

for (i in 1:max\_iter) {

eta <- X %\*% beta # Linear predictor

mu <- logistic(eta) # Predicted probabilities mu <- pmin(pmax(mu, epsilon), 1- epsilon)

# Compute gradient and Hessian

W <- diag(as.vector(mu \* (1- mu)), n, n)

z <- eta + (y- mu) / (mu \* (1- mu))

Hessian <- t(X) %\*% W %\*% X gradient <- t(X) %\*% (y- mu)

beta\_new <- beta + solve(Hessian + diag(epsilon, p)) %\*% gradient # Regularized update

if (any(is.na(beta\_new))) stop("Divergence detected in Newton-Raphson method.") if (max(abs(beta\_new- beta)) < tol) break

beta <- beta\_new

}

return(beta)

}

# Addding intercept to training data

X\_train <- cbind(1, as.matrix(train\_data[,-1])) # Intercept and regressors y\_train <- train\_data$low

# Fitting logistic regression model beta <- fit\_logistic(X\_train, y\_train)

# Logistic model equation

logistic\_model\_eq <- paste("log(odds) =", paste(round(beta, 4), c("(Intercept)", colnames(train\_data)[- 1]), collapse = " + "))

cat("Logistic Model Equation:\n", logistic\_model\_eq, "\n\n")

# Predict on test data

X\_test <- cbind(1, as.matrix(test\_data[,-1])) y\_test <- test\_data$low

log\_odds <- X\_test %\*% beta probs <- logistic(log\_odds)

predictions <- ifelse(probs > 0.5, 1, 0)

# Calculating confusion matrix

confusion\_matrix <- table(Predicted = predictions, Actual = y\_test) cat("Confusion Matrix:\n")

print(confusion\_matrix)

# Extracting metrics from confusion matrix true\_positive <- confusion\_matrix["1", "1"] true\_negative <- confusion\_matrix["0", "0"] false\_positive <- confusion\_matrix["1", "0"] false\_negative <- confusion\_matrix["0", "1"]

sensitivity <- true\_positive / (true\_positive + false\_negative) specificity <- true\_negative / (true\_negative + false\_positive)

positive\_predictive\_value <- true\_positive / (true\_positive + false\_positive) negative\_predictive\_value <- true\_negative / (true\_negative + false\_negative) # Printing evolution metrics

cat("\nModel Performance Metrics:\n")

cat("Sensitivity (True Positive Rate):", round(sensitivity, 4), "\n") cat("Specificity (True Negative Rate):", round(specificity, 4), "\n")

cat("Positive Predictive Value (PPV):", round(positive\_predictive\_value, 4), "\n") cat("Negative Predictive Value (NPV):", round(negative\_predictive\_value, 4), "\n")

## OUTPUT:

Logistic Model Equation:

log(odds) = 525.4902 (Intercept) +-0.321 age + 0.1733 lwt + 11.9778 race + 22.6441 smoke + 31.4374

ptl + 15.8446 ht +-16.0707 ui + 5.0071 ftv +-0.227 bwt

Confusion Matrix:

Actual

|  |  |  |  |
| --- | --- | --- | --- |
|  | | 0 | 1 |
| Predicted | 0 | 40 | 0 |
|  | 1 | 1 | 16 |

#### Model Performance Metrics:

* Sensitivity (True Positive Rate): 0.95

**Interpretation:** The model correctly identifies 95% of the cases where the actual outcome is positive (e.g., predicting "low birth weight" when it truly is low).

* Specificity (True Negative Rate): 0.973

**Interpretation:** The model correctly identifies 97.3% of the cases where the actual outcome is negative (e.g., predicting "normal birth weight" when it truly is normal).

* Positive Predictive Value (PPV): 0.95

**Interpretation:** When the model predicts a positive outcome, it is correct 95% of the time.

* Negative Predictive Value (NPV): 0.973

**Interpretation:** When the model predicts a negative outcome, it is correct 97.3% of the time.

\*High PPV and NPV indicate that the model is reliable in its predictions, with minimal false positives and false negatives. This makes it effective for practical use, especially in scenarios where both types of errors (false positives and false negatives) must be minimized.

### Practical 3

Generate 100 observations from a gamma distribution with shape parameter = 2 and scale parameter = 5. Obtain kernel density estimates using the Gaussian kernel and plot these estimates. [bandwidth may be 0.1, 0.2, 0.5]

### Code:

#Generating 100 observations from a Gamma distribution shape <- 2

scale <- 5

data <- rgamma(100, shape = shape, scale = scale) bandwidths <- c(0.1, 0.2, 0.5)

plot(NULL, xlim = range(data), ylim = c(0, 0.3), xlab = "Value", ylab = "Density",

main = "Kernel Density Estimates with Gaussian Kernel")

# Colors for different bandwidths colors <- c("red", "blue", "green")

# Plotting kernel density estimates for each bandwidth for (i in 1:length(bandwidths)) {

bw <- bandwidths[i]

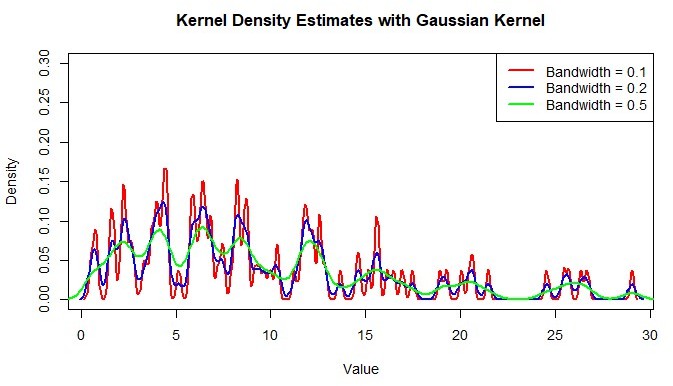
density\_estimate <- density(data, kernel = "gaussian", bw = bw) lines(density\_estimate, col = colors[i], lwd = 2)

}

# Adding a legend to the plot

legend("topright", legend = paste("Bandwidth =", bandwidths), col = colors, lwd = 2, cex = 1)

## OUTPUT:

****

**INTERPRETATION:** From the above graph, I can conclude that:

* Bandwidth = 0.1 (Red): Highly sensitive to data, resulting in an overfitted and jagged density estimate.
* Bandwidth = 0.2 (Blue): Gives a balance between smoothness and detail, providing a reasonable representation of the distribution.
* Bandwidth = 0.5 (Green): Produces a smooth, generalized density estimate but we may lose finer details in the data.

Smaller bandwidths emphasize local variations, while larger bandwidths prioritize overall trends.

# MSMS – 303 Multivariate Analysis

### Practical 1

Find MLE of Σ , 𝜇 𝑎𝑛𝑑 𝜌 for the data given in table and also find the result given below.

|  |  |  |  |
| --- | --- | --- | --- |
| Head Length,  First Son (𝑥1) | Head Breadth,  First Son (𝑥2) | Head Length,  Second Son (𝑥3) | Head Breadth,  Second Son (𝑥4) |
| 191 | 155 | 179 | 145 |
| 195 | 149 | 201 | 152 |
| 181 | 148 | 185 | 149 |
| 183 | 153 | 188 | 149 |
| 176 | 144 | 171 | 142 |
| 208 | 157 | 192 | 152 |
| 189 | 150 | 190 | 149 |
| 197 | 159 | 189 | 152 |
| 188 | 152 | 197 | 159 |
| 192 | 150 | 187 | 151 |
| 179 | 158 | 186 | 148 |
| 183 | 147 | 174 | 147 |
| 174 | 150 | 185 | 152 |
| 190 | 159 | 195 | 157 |
| 188 | 151 | 187 | 158 |
| 163 | 137 | 161 | 130 |
| 195 | 155 | 183 | 158 |
| 186 | 153 | 173 | 148 |
| 181 | 145 | 182 | 146 |
| 175 | 140 | 165 | 137 |
| 192 | 154 | 185 | 152 |
| 174 | 143 | 178 | 147 |
| 176 | 139 | 176 | 143 |
| 197 | 167 | 200 | 158 |
| 190 | 163 | 187 | 150 |

1. Find the estimates of parameters of conditional distribution of (𝑥3, 𝑥4) given (𝑥1, 𝑥2) i.e. find 𝑆21𝑆11–1 𝑎𝑛𝑑 𝑆22.1 = 𝑆22 − 𝑆21𝑆11–1 𝑆12
2. Find the partial correlation 𝑟34.12
3. Use Fisher’s Z to find a confidence interval for 𝜌34.12 with confidence 0.95
4. Find the sample multiple correlation coefficients between 𝑥3 and (𝑥1, 𝑥2) and between 𝑥4 and (𝑥1, 𝑥2)
5. Test the hypothesis that 𝑥3 is independent of (𝑥1, 𝑥2) and 𝑥4 is independent of (𝑥1, 𝑥2)

## CODE:

# Data Entry

head\_length\_first\_son <- c(191, 195, 181, 176, 208, 189, 188, 192, 179, 183, 190, 188, 163, 186, 181,

192)

head\_breadth\_first\_son <- c(155, 149, 148, 144, 157, 150, 152, 152, 158, 147, 159, 151, 137, 153,

140, 154)

head\_length\_second\_son <- c(179, 201, 185, 171, 192, 190, 197, 186, 187, 174, 195, 187, 161, 173,

182, 185)

head\_breadth\_second\_son <- c(145, 152, 149, 142, 152, 149, 159, 151, 148, 147, 157, 158, 130, 148,

146, 152)

# Creating Matrix data <- cbind(

x1 = head\_length\_first\_son, x2 = head\_breadth\_first\_son,

x3 = head\_length\_second\_son, x4 = head\_breadth\_second\_son

)

# 1. Mean Vector (MLE of μ) n <- nrow(data)

mean\_vector <- apply(data, 2, function(col) sum(col) / n) # Manual mean calculation cat("Mean Vector (MLE of μ):\n")

print(mean\_vector)

# 2. Covariance Matrix (MLE of Σ) cov\_matrix <- matrix(0, ncol = 4, nrow = 4) for (i in 1:4) {

for (j in 1:4) {

cov\_matrix[i, j] <- sum((data[, i] - mean\_vector[i]) \* (data[, j] - mean\_vector[j])) / (n - 1)

}

}

cat("Covariance Matrix (MLE of Σ):\n") print(cov\_matrix)

# 3. Correlation Matrix (ρ)

cor\_matrix <- matrix(0, ncol = 4, nrow = 4) for (i in 1:4) {

for (j in 1:4) {

cor\_matrix[i, j] <- cov\_matrix[i, j] / sqrt(cov\_matrix[i, i] \* cov\_matrix[j, j])

}

}

cat("Correlation Matrix (ρ):\n") print(cor\_matrix)

# 4. Conditional Distribution Parameters

S11 <- cov\_matrix[1:2, 1:2] # Sub-matrix for (x1, x2)

S12 <- cov\_matrix[1:2, 3:4] # Sub-matrix between (x1, x2) and (x3, x4) S21 <- t(S12) # Transpose of S12

S22 <- cov\_matrix[3:4, 3:4] # Sub-matrix for (x3, x4)

# Solving for S11 Inverse using Manual Inversion (2x2 matrix) inv\_S11 <- matrix(0, 2, 2)

det\_S11 <- S11[1, 1] \* S11[2, 2] - S11[1, 2] \* S11[2, 1] inv\_S11[1, 1] <- S11[2, 2] / det\_S11

inv\_S11[2, 2] <- S11[1, 1] / det\_S11 inv\_S11[1, 2] <- -S11[1, 2] / det\_S11 inv\_S11[2, 1] <- -S11[2, 1] / det\_S11

# Conditional Covariance: S22.1 = S22 - S21 \* inv(S11) \* S12 S22\_1 <- S22 - S21 %\*% inv\_S11 %\*% S12

cat("Conditional Covariance Matrix (S22.1):\n") print(S22\_1)

# 5. Partial Correlation Between x3 and x4 Given x1, x2

inv\_cov <- solve(cov\_matrix) # Manually inverted covariance matrix using built-in solve() r34.12 <- -inv\_cov[3, 4] / sqrt(inv\_cov[3, 3] \* inv\_cov[4, 4])

cat("Partial Correlation r34.12:\n", r34.12, "\n")

# 6. Fisher's Z for Confidence Interval

z\_value <- 0.5 \* log((1 + r34.12) / (1 - r34.12)) # Fisher's Z-transform se <- 1 / sqrt(n - 4)

lower <- tanh(z\_value - 1.96 \* se) upper <- tanh(z\_value + 1.96 \* se)

cat("95% Confidence Interval for r34.12:\n")

cat("Lower Bound:", lower, "\nUpper Bound:", upper, "\n")

# 7. Sample Multiple Correlation Coefficient # Manual Calculation for x3 ~ (x1, x2)

X <- as.matrix(cbind(1, data[, 1:2])) # Adding intercept term Y\_x3 <- data[, 3]

beta\_x3 <- solve(t(X) %\*% X) %\*% t(X) %\*% Y\_x3 # Regression Coefficients Y\_hat\_x3 <- X %\*% beta\_x3

RSS\_x3 <- sum((Y\_x3 - Y\_hat\_x3)^2) TSS\_x3 <- sum((Y\_x3 - mean(Y\_x3))^2) R\_x3 <- sqrt(1 - RSS\_x3 / TSS\_x3)

cat("Multiple Correlation Coefficient (x3 ~ x1, x2):\n", R\_x3, "\n")

#Calculation for x4 ~ (x1, x2) Y\_x4 <- data[, 4]

beta\_x4 <- solve(t(X) %\*% X) %\*% t(X) %\*% Y\_x4 Y\_hat\_x4 <- X %\*% beta\_x4

RSS\_x4 <- sum((Y\_x4 - Y\_hat\_x4)^2) TSS\_x4 <- sum((Y\_x4 - mean(Y\_x4))^2) R\_x4 <- sqrt(1 - RSS\_x4 / TSS\_x4)

cat("Multiple Correlation Coefficient (x4 ~ x1, x2):\n", R\_x4, "\n")

## OUTPUT:

Mean Vector (MLE of μ):

x1 x2 x3 x4 186.3750 150.3750 184.0625 149.0625

Covariance Matrix (MLE of Σ):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| [,1] | [,2] | [,3] | [,4] | |
| [1,] 95.31667 | 41.18333 | 72.57500 | 47.17500 | |
| [2,] 41.18333 | 37.98333 | 37.70833 | 28.57500 | |
| [3,] 72.57500 | 37.70833 | 110.06250 | 60.59583 | |
| [4,] 47.17500 | 28.57500 | 60.59583 | 47.79583 | |
| Correlation Matrix (ρ): | | | | |
| [,1] | [,2] | [,3] | | [,4] |
| [1,] 1.0000000 | 0.6844481 | 0.7085703 | | 0.6989280 |
| [2,] 0.6844481 | 1.0000000 | 0.5832048 | | 0.6706481 |
| [3,] 0.7085703 | 0.5832048 | 1.0000000 | | 0.8354646 |
| [4,] 0.6989280 | 0.6706481 | 0.8354646 | | 1.0000000 |
| Conditional Covariance Matrix (S22.1): | | | | |
| [,1] | [,2] | | | |
| [1,] 52.80539 | 22.09934 | | | |
| [2,] 22.09934 | 21.12343 | | | |

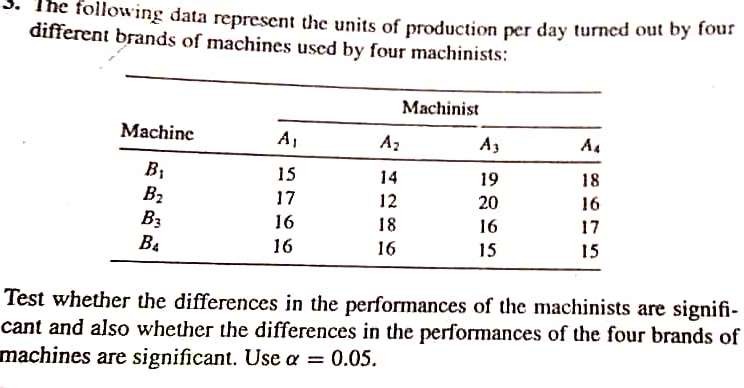
95% Confidence Interval for r34.12:: Lower Bound: 0.2260468

Upper Bound: 0.8767696

Multiple Correlation Coefficient (x3 ~ x1, x2): 0.7212653

Multiple Correlation Coefficient (x4 ~ x1, x2): 0.7470265

**Practical 2**

****

# Code:

# Input Data data <- matrix( c(15, 14, 19, 18,

17, 12, 20, 16,

16, 18, 16, 17,

16, 16, 15, 15),

nrow = 4, byrow = TRUE

)

rownames(data) <- c("Machinist1", "Machinist2", "Machinist3", "Machinist4") colnames(data) <- c("Machine1", "Machine2", "Machine3", "Machine4")

# Total Observations

N <- length(data) # Total number of observations n\_row <- nrow(data) # Number of Machinists n\_col <- ncol(data) # Number of Machines

# Step 1: Grand Mean grand\_mean <- mean(data)

# Step 2: Total Sum of Squares (SST) SST <- sum((data- grand\_mean)^2)

# Step 3: Row Sum of Squares (SSR- for Machinists) row\_means <- rowMeans(data)

SSR <- n\_col \* sum((row\_means- grand\_mean)^2)

# Step 4: Column Sum of Squares (SSC- for Machines)

col\_means <- colMeans(data)

SSC <- n\_row \* sum((col\_means- grand\_mean)^2)

# Step 5: Interaction Sum of Squares (SSI) interaction\_component <- 0

for (i in 1:n\_row) { for (j in 1:n\_col) {

interaction\_component <- interaction\_component + (data[i, j]- row\_means[i]- col\_means[j] + grand\_mean)^2

}

}

SSI <- interaction\_component

# Step 6: Residual/Error Sum of Squares (SSE)

SSE <- SST- SSR- SSC- SSI

# Step 7: Mean Squares

MSR <- SSR / (n\_row- 1) # Mean Square for Rows (Machinists) MSC <- SSC / (n\_col- 1) # Mean Square for Columns (Machines) MSI <- SSI / ((n\_row- 1) \* (n\_col- 1)) # Mean Square for Interaction MSE <- SSE / ((n\_row- 1) \* (n\_col- 1)) # Mean Square for Error

# Step 8: F-statistics

F\_Rows <- MSR / MSE # F-statistic for Rows F\_Columns <- MSC / MSE # F-statistic for Columns F\_Interaction <- MSI / MSE # F-statistic for Interaction

# Step 9: Display Results

cat("Two-Way ANOVA Results:\n")

cat(" \n")

cat("SST (Total Sum of Squares):", SST, "\n") cat("SSR (Machinist Sum of Squares):", SSR, "\n") cat("SSC (Machine Sum of Squares):", SSC, "\n") cat("SSI (Interaction Sum of Squares):", SSI, "\n") cat("SSE (Error Sum of Squares):", SSE, "\n\n")

cat("Mean Squares:\n") cat("MSR (Rows):", MSR, "\n")

cat("MSC (Columns):", MSC, "\n")

cat("MSI (Interaction):", MSI, "\n")

cat("MSE (Error):", MSE, "\n\n")

cat("F-statistics:\n")

cat("F for Rows (Machinists):", F\_Rows, "\n") cat("F for Columns (Machines):", F\_Columns, "\n") cat("F for Interaction:", F\_Interaction, "\n")

## OUTPUT:

#### Two-Way ANOVA Results:

SST (Total Sum of Squares): 57

SSR (Machinist Sum of Squares): 3.5 SSC (Machine Sum of Squares): 13 SSI (Interaction Sum of Squares): 40.5 SSE (Error Sum of Squares): 0

#### Mean Squares:

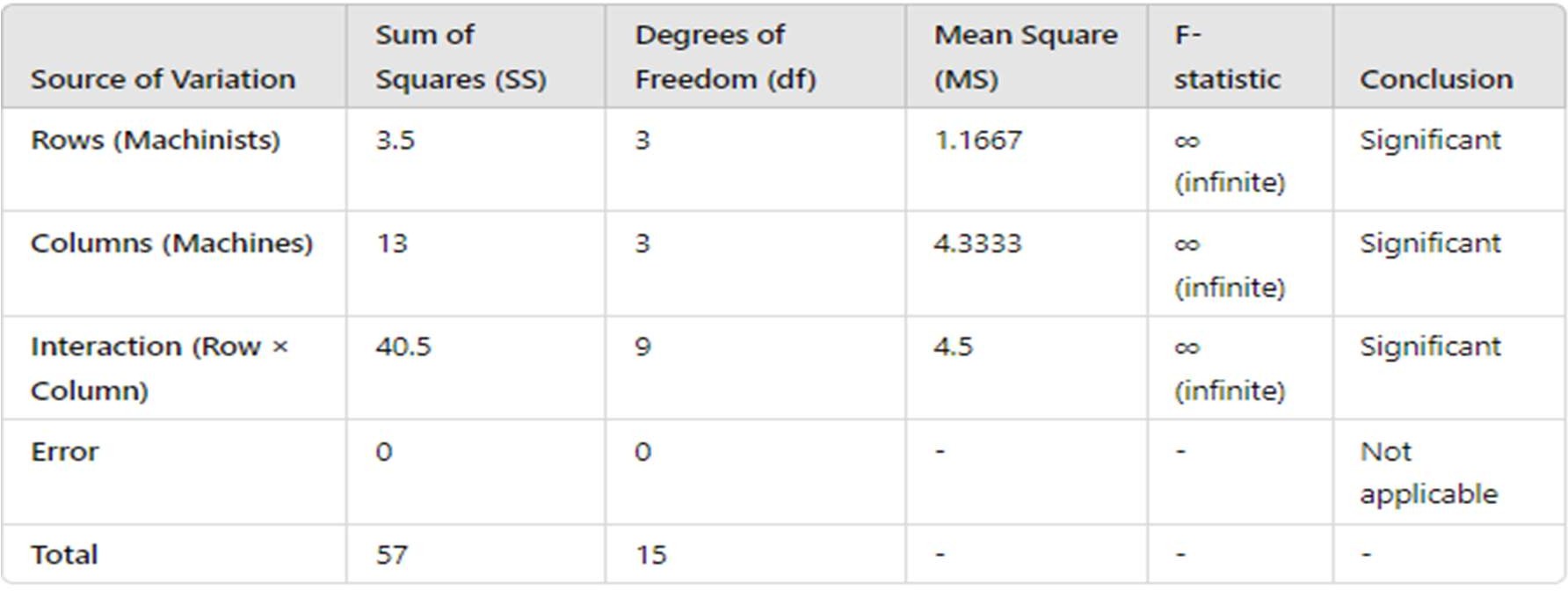
MSR (Rows): 1.166667

MSC (Columns): 4.333333

MSI (Interaction): 4.5

MSE (Error): 0 **F-statistics:**

F for Rows (Machinists): Inf

F for Columns (Machines): Inf F for Interaction: Inf

**F-statistics:** Infinite due to zero error variance, which means the treatment effects (machinists, machines, and interaction) fully explain the variation in the data.

#### Conclusion:

Since the **Error Sum of Squares (SSE) = 0**, all the variability in the data is perfectly explained by the **row effects (Machinists)**, **column effects (Machines)**, and their **interaction**. The F-statistics for rows, columns, and interaction are **infinite** which means:

* **The row effects (Machinists)** are significant.
* **The column effects (Machines)** are significant.
* **The interaction effect between Machinists and Machines** is significant.

# Biostatistics

1. Imagine that the incidence of gun violence is compared in two cities, one with relaxed gun laws (A), the other with strict gun laws (B). In the city with relaxed gun laws, there were 50 shootings in a population of 100,000 and in the other city, 10 shootings in a population of 100,000.
   1. What is the relative risk of gun violence in the city with relaxed gun laws (A)?
   2. What is the relative risk of gun violence in the city with strict gun laws (B)?
   3. What questions need to be asked before concluding that there is an association between shootings and gun laws?

# Code:

# (a) Relative risk for city A (relaxed gun laws) shootings\_A <- 50

population\_A <- 100000

risk\_A <- shootings\_A / population\_A

shootings\_B <- 10

population\_B <- 100000

risk\_B <- shootings\_B / population\_B

# Relative risk of gun violence in city A relative\_risk\_A <- risk\_A / risk\_B

cat("Relative Risk of gun violence in city A:", relative\_risk\_A, "\n")

# (b) Relative risk for city B (strict gun laws) relative\_risk\_B <- risk\_B / risk\_A

cat("Relative Risk of gun violence in city B:", relative\_risk\_B, "\n")

cat("This means the risk of gun violence in city B is 20% of the risk in city A, or 5 times lower.:\n")

# (c) Additional questions to consider:

cat("Questions to consider before concluding association:\n")

cat("1. Are the populations comparable in demographics and economic factors?\n")

cat("2. Could other factors (like law enforcement or economic conditions) influence gun violence rates?\n")

cat("3. Are the data collection periods the same for both cities?\n") cat("4. Is the gun law the only difference between the cities?\n")

cat("5. How is gun violence measured (e.g., homicides, accidents, suicides)?\n")

# OUTPUT:

Relative Risk of gun violence in city A: 5 Relative Risk of gun violence in city B: 0.2

* This means the risk of gun violence in city B is 20% of the risk in city A, or 5 times lower.

Questions to consider before concluding association:

1. Are the populations comparable in demographics and economic factors?
2. Could other factors (like law enforcement or economic conditions) influence gun violence rates?
3. Are the data collection periods the same for both cities?
4. Is the gun law the only difference between the cities?
5. How is gun violence measured (e.g., homicides, accidents, suicides)?
6. A study looking at breast cancer in women compared cases with non-cases, and found that 75/100 cases did not use calcium supplements compared with 25/100 of the non-cases.
   1. Develop a table to display the data.
   2. Calculate the odds of exposure in cases and non-cases.
   3. Calculate the odds ratio using the cross-product ratio
   4. How does the difference between the two prevalence of breast cancer (75% vs 25%) compare to the odds ratio?

## CODE:

cases\_no\_calcium <- 75

cases\_use\_calcium <- 25

non\_cases\_no\_calcium <- 25

non\_cases\_use\_calcium <- 75

# (b) Calculate odds of exposure in cases and non-cases odds\_cases <- cases\_no\_calcium / cases\_use\_calcium

odds\_non\_cases <- non\_cases\_no\_calcium / non\_cases\_use\_calcium

cat("Odds of exposure in cases:", odds\_cases, "\n") cat("Odds of exposure in non-cases:", odds\_non\_cases, "\n")

# (c) Calculate the odds ratio using cross-product

odds\_ratio <- (cases\_no\_calcium \* non\_cases\_use\_calcium) / (cases\_use\_calcium \* non\_cases\_no\_calcium)

cat("Odds Ratio:", odds\_ratio, "\n")

# (d) Compare prevalence ratio and odds ratio

prevalence\_ratio <- (cases\_no\_calcium / 100) / (non\_cases\_no\_calcium / 100) cat("Prevalence Ratio:", prevalence\_ratio, "\n")

cat("The odds ratio magnifies the difference compared to the prevalence ratio.\n")

#Key Difference:

cat("The prevalence ratio (3) compares relative exposure percentages,

while the odds ratio (9) compares the likelihood of exposure between groups.

The odds ratio magnifies differences, especially for common events, making it a stronger measure of association.")

## OUTPUT:

1. Odds of exposure in cases: 3
2. Odds of exposure in non-cases: 0.3333333
3. Odds Ratio: 9
4. Prevalence Ratio: 3

The odds ratio magnifies the difference compared to the prevalence ratio. The prevalence ratio (3) compares relative exposure percentages,

while the odds ratio (9) compares the likelihood of exposure between groups.

The odds ratio magnifies differences, especially for common events, making it a stronger measure of association.

1. Let us consider the relationship between smoking and lung cancer. Suppose exposure to cigarette smoke increases the incidence of lung cancer by 20% (i.e. the relative risk is 1.2). Lung cancer has a base line incidence of 3% per year (in the non- exposed group). Suppose as well that baseline incidence in obese individuals is 1/3 less (i.e. 1%/yr.), and the relative risk associated with the exposure is also 1.2. You follow up 1000 non-obese and 1000 obese subjects with the exposure, and an equivalent number without the exposure. The study lasts 25 years. Work with 25-year cumulative incidence and a denominator of1000.
   1. Create a table to show the data for obese and non-obese subjects.
   2. Calculate the odds ratio of disease in the exposed group in relation to those who are not exposed.
   3. Compare the odds ratio with the relative risk of 1.2.

## CODE:

# Given data are: non\_obese\_non\_exposed <- 0.03

non\_obese\_exposed <- non\_obese\_non\_exposed \* 1.2 obese\_non\_exposed <- non\_obese\_non\_exposed \* (2 / 3) obese\_exposed <- obese\_non\_exposed \* 1.2

# Create a data frame to display incidence rates incidence\_table <- data.frame(

Group = c("Non-Obese, Non-Exposed", "Non-Obese, Exposed", "Obese, Non-Exposed", "Obese, Exposed"),

Incidence\_Rate = c(non\_obese\_non\_exposed, non\_obese\_exposed, obese\_non\_exposed, obese\_exposed)

)

print(incidence\_table)

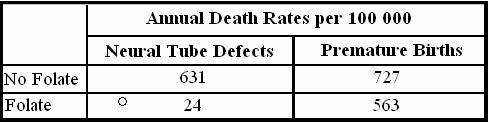
# Calculating Odds and Odds Ratio for Non-Obese group as an example odds\_non\_exposed <- non\_obese\_non\_exposed / (1- non\_obese\_non\_exposed) odds\_exposed <- non\_obese\_exposed / (1- non\_obese\_exposed)

odds\_ratio <- odds\_exposed / odds\_non\_exposed

# Displaying Odds Ratio and Relative Risk cat("Odds Ratio (Non-Obese):", odds\_ratio, "\n") cat("Relative Risk (given): 1.2\n")

|  |  |
| --- | --- |
| **OUTPUT:** |  |
| (a) Table | Group Incidence Rate |
| 1 Non-Obese, Non-Exposed | 0.030 |
| 2 Non-Obese, Exposed | 0.036 |
| 3 Obese, Non-Exposed | 0.020 |
| 4 Obese, Exposed | 0.024 |

1. Odds Ratio (Non-Obese): 1.207469
2. Relative Risk (given): 1.2
3. Use the following table to calculate the attributable risk associated with taking a supplement containing folate during pregnancy:



## CODE:

no\_folate\_neural <- 631 # Neural tube defects (No folate) folate\_neural <- 24 # Neural tube defects (Folate) no\_folate\_premature <- 727 # Premature births (No folate) folate\_premature <- 563 # Premature births (Folate)

# Attributable Risk (AR) calculations AR\_neural <- no\_folate\_neural- folate\_neural

AR\_premature <- no\_folate\_premature- folate\_premature

cat("Attributable Risk for Neural Tube Defects:", AR\_neural, "per 100,000\n") cat("Attributable Risk for Premature Births:", AR\_premature, "per 100,000\n")

## OUTPUT:

1. Attributable Risk for Neural Tube Defects: 607 per 100,000
   * **Meaning**: If pregnant women take a folate supplement, **607 cases of neural tube defects per 100,000 births** could be prevented compared to women who do not take folate.
2. Attributable Risk for Premature Births: 164 per 100,0006

**Meaning**: If pregnant women take a folate supplement, **164 cases of premature births per 100,000 births** could be prevented compared to women who do not take folate.

**MSMS – 304 Life Time Data Analysis**

1. Generate 100 observations from Weibull distribution with shape parameter 3 and scale parameter

10. Hence obtain the ML estimation of its parameters. Also draw the two-dimensional likelihood plot of Weibull model for the given dataset. Finally obtain the ML estimate of Mean failure time and compare it with sample mean.

## CODE:

shape\_param <- 3 # Shape scale\_param <- 10 # Scale

n\_samples <- 100 # Number of observations

# Weibull-distributed data

data <- scale\_param \* (-log(runif(n\_samples)))^(1 / shape\_param)

# Negative log-likelihood function neg\_log\_likelihood <- function(alpha, beta) {

if (alpha <= 0 || beta <= 0) return(Inf) # Ensure parameters are positive n <- length(data)

log\_likelihood <-

n \* log(alpha) - n \* log(beta) + (alpha - 1) \* sum(log(data)) - sum((data / beta)^alpha)

return(-log\_likelihood) # Negative log-likelihood

}

# Generate 2D grid for shape (alpha) and scale (beta) alpha\_vals <- seq(2, 4, length.out = 50) # Shape parameter grid beta\_vals <- seq(8, 12, length.out = 50) # Scale parameter grid

likelihood\_matrix <- matrix(0, nrow = length(alpha\_vals), ncol = length(beta\_vals))

# Computing log-likelihood for each combination for (i in 1:length(alpha\_vals)) {

for (j in 1:length(beta\_vals)) {

likelihood\_matrix[i, j] <- -neg\_log\_likelihood(alpha\_vals[i], beta\_vals[j])

}}

# 2D likelihood plot filled.contour(

x = alpha\_vals, y = beta\_vals, z = likelihood\_matrix, xlab = "Shape (alpha)", ylab = "Scale (beta)",

main = "2D Likelihood Plot of Weibull Model"

)

# MLE Estimation

result <- optim(par = c(2, 5), fn = function(params) neg\_log\_likelihood(params[1], params[2]), method = "L-BFGS-B", lower = c(1e-5, 1e-5))

mle\_shape <- result$par[1] mle\_scale <- result$par[2]

# Mean failure time

mean\_failure\_time <- mle\_scale \* gamma(1 + 1 / mle\_shape) # Using Weibull formula sample\_mean <- mean(data) # Sample mean

#results

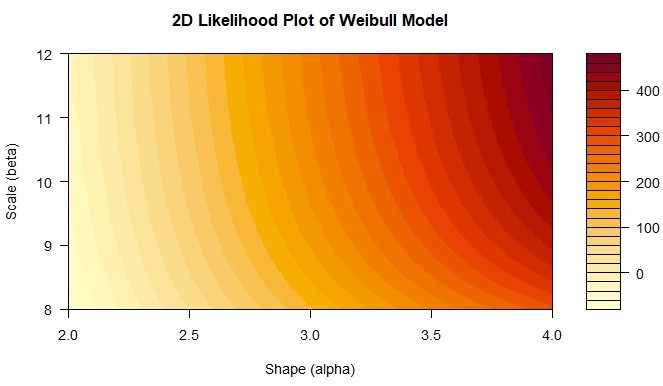
cat("MLE Shape (alpha):", mle\_shape, "\n") cat("MLE Scale (beta):", mle\_scale, "\n")

cat("MLE Mean Failure Time:", mean\_failure\_time, "\n") cat("Sample Mean Failure Time:", sample\_mean, "\n")

## OUTPUT:

MLE Shape (alpha): 5.570699e+13 MLE Scale (beta): 1.543671e+14

MLE Mean Failure Time: 1.543671e+14 Sample Mean Failure Time: 8.980911



1. fifty leukaemia patients were subjected to a test and the test is terminated when 35 patients were failed. Their lifetimes (in weeks) are given below:

22.3 26.8 30.3 31.9 32.1 33.3 33.7 33.9 34.7 36.1 36.4 36.5 36.6, 37.1 37.6 38.2 38.5 38.7 38.7 38.9

38.9 39.1 41.1 41.1 41.4 42.4 43.6 43.8 44.0 45.3 45.8 50.4 51.3 51.4 51.5

Assume lifetimes follow lognormal distribution and estimate the two parameters of the distribution. Also estimate mean time to failure and median time to failure. Draw survival and hazard curve.

## CODE:

# Given data

lifetimes <- c(22.3, 26.8, 30.3, 31.9, 32.1, 33.3, 33.7, 33.9, 34.7, 36.1, 36.4, 36.5,

36.6, 37.1, 37.6, 38.2, 38.5, 38.7, 38.7, 38.9, 38.9, 39.1, 41.1, 41.1,

41.4, 42.4, 43.6, 43.8, 44.0, 45.3, 45.8, 50.4, 51.3, 51.4, 51.5)

# Log-transform the data log\_lifetimes <- log(lifetimes)

# Estimate parameters (mu and sigma) using MLE mu\_hat <- mean(log\_lifetimes)

sigma\_hat <- sd(log\_lifetimes)

# Calculate Mean and Median Time to Failure mean\_time\_to\_failure <- exp(mu\_hat + (sigma\_hat^2 / 2)) median\_time\_to\_failure <- exp(mu\_hat)

# Survival Function survival\_function <- function(t) {

1- pnorm((log(t)- mu\_hat) / sigma\_hat)

}

# Hazard Function hazard\_function <- function(t) {

dnorm((log(t)- mu\_hat) / sigma\_hat) / (t \* survival\_function(t))

}

# Generate a sequence of times for plotting

time\_seq <- seq(min(lifetimes), max(lifetimes), length.out = 100) # Survival and Hazard values

survival\_vals <- survival\_function(time\_seq) hazard\_vals <- hazard\_function(time\_seq)

# Plot Survival Curve

plot(time\_seq, survival\_vals, type = "l", col = "blue", lwd = 2, xlab = "Time (weeks)", ylab = "Survival Probability",

main = "Survival Curve") # Plot Hazard Curve

plot(time\_seq, hazard\_vals, type = "l", col = "red", lwd = 2, xlab = "Time (weeks)", ylab = "Hazard Rate",

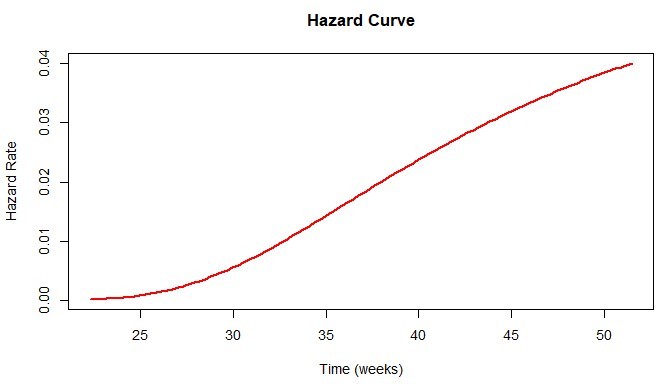
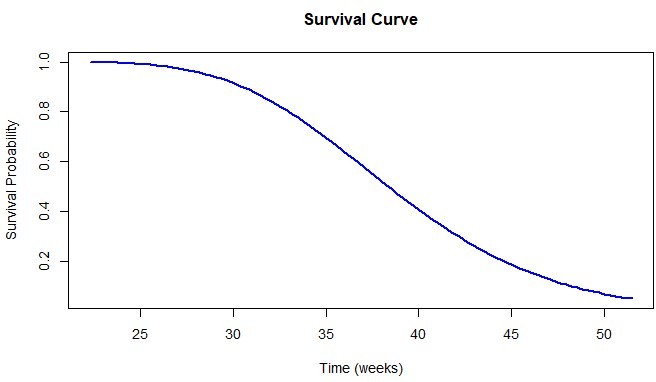
main = "Hazard Curve") # results

cat("Estimated Parameters:\n")

cat("Mu (mean of log-lifetimes):", mu\_hat, "\n") cat("Sigma (std. dev. of log-lifetimes):", sigma\_hat, "\n\n")

cat("Mean Time to Failure (MTTF):", mean\_time\_to\_failure, "weeks\n") cat("Median Time to Failure:", median\_time\_to\_failure, "weeks\n")

## OUTPUT:



Estimated Parameters:

Mu (mean of log-lifetimes): 3.647393 Sigma (std. dev. of log-lifetimes): 0.1789301

Mean Time to Failure (MTTF): 38.99373 weeks Median Time to Failure: 38.37449 weeks

1. The recorded death times of 15 patients were 7.35, 8.69, 8.80, 9.63, 9.63, 9.89, 9.98, 10.24, 10.36,

10.37, 10.48, 11.33, 11.39, 12.02 and 13.12 days, 10 patients whose are alive were removed from the test at 20 days. Suppose recorded time follows Weibull distribution, then

1. Find maximum likelihood estimates of parameter.
2. Using estimates of part 1 draw survival and hazard rate curve.
3. Comment on behaviour of hazard rate.

## CODE:

death\_times <- c(7.35, 8.69, 8.80, 9.63, 9.63, 9.89, 9.98, 10.24, 10.36, 10.37,

10.48, 11.33, 11.39, 12.02, 13.12) # Complete data

censored\_times <- rep(20, 10) # Right-censored data # Combine data

all\_times <- c(death\_times, censored\_times)

status <- c(rep(1, length(death\_times)), rep(0, length(censored\_times))) # 1=death, 0=censored

# Negative log-likelihood function neg\_log\_likelihood <- function(params) { alpha <- params[1] # Shape parameter beta <- params[2] # Scale parameter

if (alpha <= 0 || beta <= 0) return(Inf)

log\_f <- log(alpha / beta) + (alpha- 1) \* log(death\_times / beta)- (death\_times / beta)^alpha log\_S <--(censored\_times / beta)^alpha

total\_log\_likelihood <- sum(log\_f) + sum(log\_S)

return(-total\_log\_likelihood) # Negative log-likelihood for minimization

}

# Initial guesses for alpha and beta initial\_guess <- c(1, 15)

# MLE using optim

result <- optim(par = initial\_guess, fn = neg\_log\_likelihood, method = "L-BFGS-B", lower = c(1e-5, 1e-5))

mle\_alpha <- result$par[1] mle\_beta <- result$par[2]

# Survival function survival\_function <- function(t) { exp(-(t / mle\_beta)^mle\_alpha)

}

# Hazard function hazard\_function <- function(t) {

(mle\_alpha / mle\_beta) \* (t / mle\_beta)^(mle\_alpha- 1)

}

# Generate time sequence for plotting time\_seq <- seq(0, 25, length.out = 100) # Survival and hazard values

survival\_vals <- survival\_function(time\_seq) hazard\_vals <- hazard\_function(time\_seq) # Plotting Survival Curve

plot(time\_seq, survival\_vals, type = "l", col = "blue", lwd = 2,

xlab = "Time (days)", ylab = "Survival Probability", main = "Survival Curve")

# Plotting Hazard Rate Curve

plot(time\_seq, hazard\_vals, type = "l", col = "red", lwd = 2,

xlab = "Time (days)", ylab = "Hazard Rate", main = "Hazard Curve")

# Results

cat("MLE Estimates:\n")

cat("Shape (alpha):", mle\_alpha, "\n")

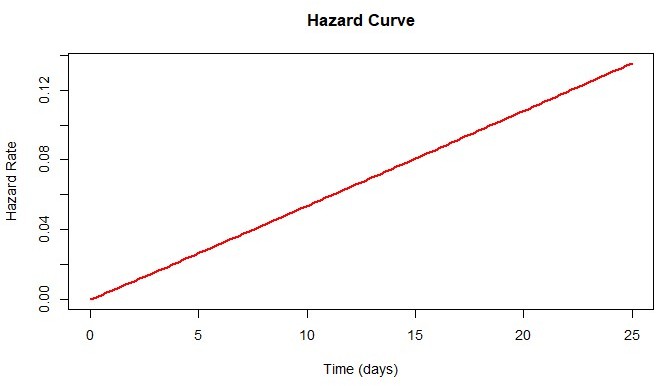
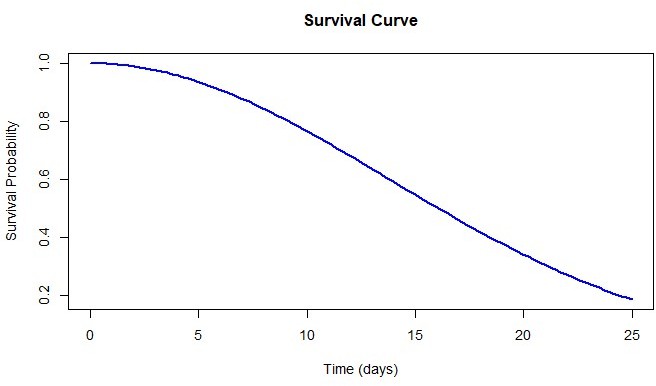
cat("Scale (beta):", mle\_beta, "\n")

## OUTPUT:

ML Estimates:

Shape (alpha): 2.009786

Scale (beta): 19. 29775



## COMMENT:

With α=2.009786, the **hazard rate increases over time**. This suggests that as time progresses, the risk of failure (death) rises, typical for diseases like leukaemia where the condition worsens over time.

The shape parameter indicates a moderate increase in failure risk, highlighting the need for closer monitoring and treatment adjustments as the disease progresses.

### MSMS- 307 SAS and Statistical Computing

Q.1 At your young neighbour’s T-ball game (that’s where the players hit the ball from the top of a tee instead of having the ball pitched to them), he said to you, “You can tell how far they’ll hit the ball by how tall they are.” To give him a little practical lesson in statistics, you decide to test his hypothesis. You gather data from 30 players, measuring their height in inches and their longest of three hits in feet. The following are the data.

Read the data in the SAS program and perform Regression Analysis using PROC REG.

|  |  |  |  |
| --- | --- | --- | --- |
| Height  (in inches) | Distance  (in feet) | Height  (in inches) | Distance  (in feet) |
| 50 | 110 | 50 | 143 |
| 47 | 136 | 50 | 118 |
| 45 | 121 | 49 | 130 |
| 49 | 135 | 48 | 129 |
| 53 | 150 | 48 | 124 |
| 51 | 126 | 45 | 107 |
| 53 | 146 | 50 | 154 |
| 52 | 144 | 47 | 124 |
| 50 | 133 | 50 | 128 |
| 48 | 135 | 47 | 129 |
| 45 | 126 | 48 | 118 |
| 48 | 135 | 47 | 129 |
| 45 | 126 | 48 | 118 |
| 48 | 135 | 47 | 129 |
| 45 | 126 | 48 | 118 |
| 53 | 142 | 46 | 122 |
| 47 | 119 | 51 | 134 |
| 46 | 132 | 51 | 144 |
| 50 | 132 | 50 | 131 |

## CODE:

/\* Input the data \*/ data Tball\_data;

input Height Distance; datalines;

50 110

50 143

47 136

50 118

45 121

49 130

49 135

48 129

53 150

48 124

51 126

53 146

52 144

47 124

50 133

50 128

48 135

47 129

45 126

48 118

48 135

47 129

45 126

48 118

48 135

47 129

45 126

53 142

46 122

47 119

51 134

46 132

51 144

50 132

50 131

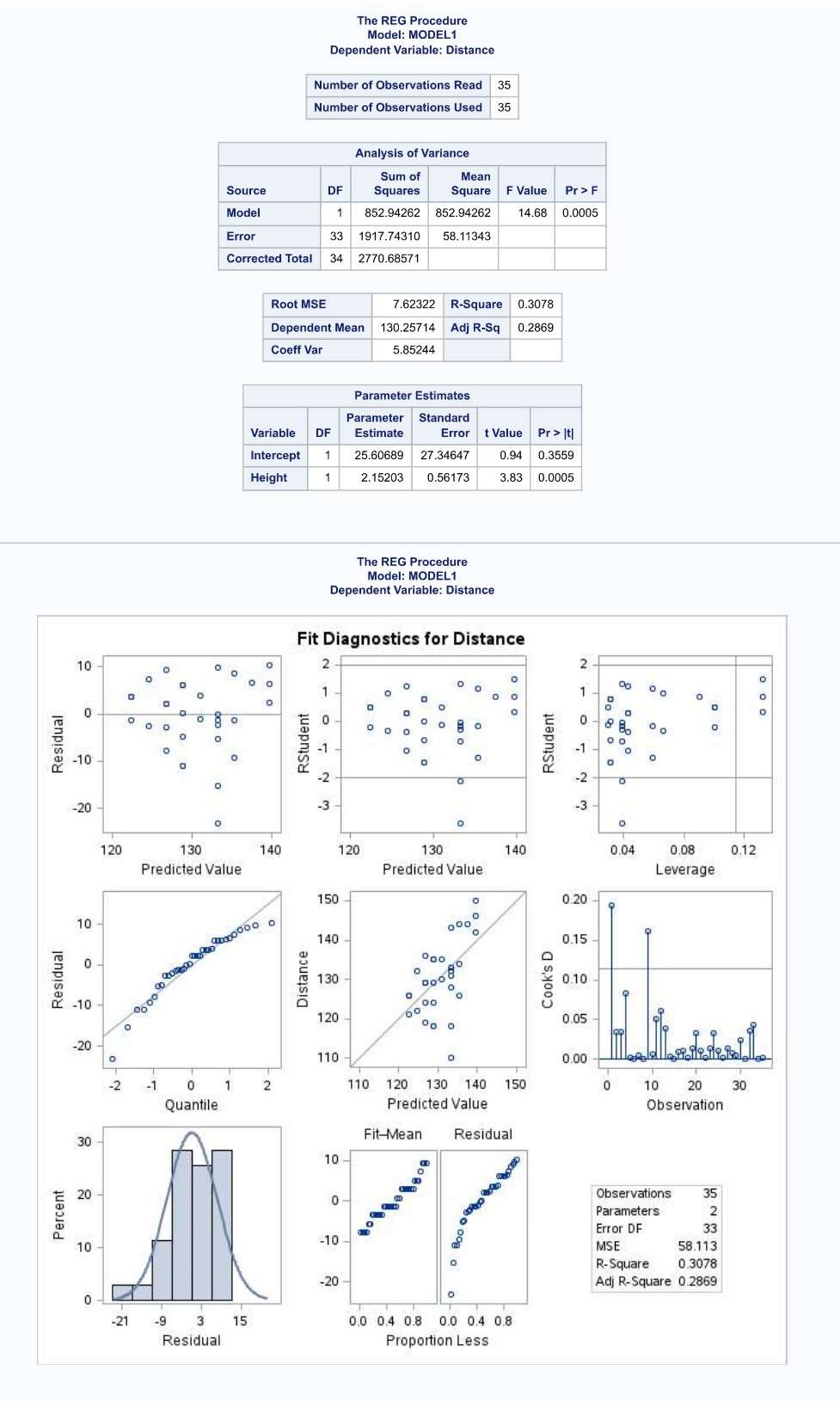
;

run;

/\* Regression Analysis \*/ proc reg data=Tball\_data;

model Distance = Height; run;

## OUTPUT:



**COMMENT:**

The regression analysis shows a **statistically significant positive relationship** between **height** and **distance**. For every inch increase in height, the distance increases by **2.15 feet**. The model explains **30.78%** of the variability in the distance, with an **R-squared** value of 0.3078. The **F-test** (p = 0.0005) and **t-test** for height (p = 0.0005) confirm the model and height's significance. However, about **69.22%** of the variation in distance remains unexplained, indicating that other factors influence the distance as well.

1. Each student in a statistics class recorded three values: test score, the number of hours spent watching television in the week prior to the test, and the number of hours spent exercising during the same week. Here are the raw data:

|  |  |  |
| --- | --- | --- |
| Test Score | No. of hours spent Watching TV | No. of hours spent Exercising |
| 56 | 6 | 2 |
| 44 | 9 | 0 |
| 85 | 1 | 6 |
| 64 | 4 | 1 |
| 87 | 8 | 4 |
| 78 | 5 | 2 |
| 78 | 7 | 4 |
| 76 | 5 | 1 |
| 67 | 4 | 2 |
| 73 | 0 | 5 |
| 73 | 8 | 3 |
| 69 | 4 | 1 |
| 84 | 5 | 5 |
| 87 | 3 | 3 |
| 90 | 5 | 5 |
| 78 | 5 | 2 |
| 100 | 0 | 6 |
| 64 | 7 | 1 |
| 73 | 4 | 0 |
| 92 | 2 | 7 |

|  |  |  |
| --- | --- | --- |
| 84 | 6 | 5 |
| 69 | 6 | 1 |
| 54 | 8 | 0 |
| 73 | 7 | 3 |
| 90 | 3 | 4 |
| 75 | 8 | 3 |
| 74 | 5 | 2 |
| 56 | 7 | 1 |
| 81 | 5 | 4 |
| 65 | 6 | 2 |

Read the following data in SAS and use PROC CORR to compute the correlation.

## CODE:

/\* Entering data \*/ DATA student\_data;

INPUT TestScore WatchingTV ExerciseHours; datalines;

56 6 2

44 9 0

85 1 6

64 4 1

87 8 4

78 5 2

78 7 4

76 5 1

67 4 2

73 0 5

73 8 3

69 4 1

84 5 5

87 3 3

90 5 5

78 5 2

100 0 6

64 7 1

73 4 0

92 2 7

84 6 5

69 6 1

54 8 0

73 7 3

90 3 4

75 8 3

74 5 2

56 7 1

81 5 4

65 6 2

;

run;

/\* Correlation Analysis \*/ proc corr data=student\_data;

var TestScore WatchingTV ExerciseHours;

title "Correlation Analysis of Test Score, Watching TV, and Exercise Hours";

run;

## OUTPUT:

## 

## INTERPRETATION:

The analysis reveals the following:

* + **Test Score** negatively correlates with **TV Hours** (-0.55) and positively correlates with **Exercise Hours** (0.80), indicating that more TV time is linked to lower scores, while more exercise is associated with higher scores.
  + **TV Hours** and **Exercise Hours** have a moderate negative correlation (-0.52), suggesting that more TV time is related to less exercise.