p8130_final_project_2

2024-12-19

Project 2: Breast cancer survival prediction

Data exploration

Descriptive table with summary statistics

```
data <- read.csv("Project_2_data.csv")
head(data,10)
```

```
##    Age  Race Marital.Status T.Stage N.Stage X6th.Stage
## 1   68 White        Married      T1      N1        IIA
## 2   50 White        Married      T2      N2       IIIA
## 3   58 White       Divorced      T3      N3       IIIC
## 4   58 White        Married      T1      N1        IIA
## 5   47 White        Married      T2      N1        IIB
## 6   51 White        Single       T1      N1        IIA
## 7   51 White        Married      T1      N1        IIA
## 8   40 White        Married      T2      N1        IIB
## 9   40 White       Divorced      T4      N3       IIIC
## 10  69 White        Married      T4      N3       IIIC
##                differentiate Grade  A.Stage Tumor.Size Estrogen.Status
## 1      Poorly differentiated     3 Regional          4        Positive
## 2  Moderately differentiated     2 Regional         35        Positive
## 3  Moderately differentiated     2 Regional         63        Positive
## 4      Poorly differentiated     3 Regional         18        Positive
## 5      Poorly differentiated     3 Regional         41        Positive
## 6  Moderately differentiated     2 Regional         20        Positive
## 7        Well differentiated     1 Regional          8        Positive
## 8  Moderately differentiated     2 Regional         30        Positive
## 9      Poorly differentiated     3 Regional        103        Positive
## 10       Well differentiated     1  Distant         32        Positive
##    Progesterone.Status Regional.Node.Examined Reginol.Node.Positive
## 1             Positive                    24                    1
## 2             Positive                    14                    5
## 3             Positive                    14                    7
## 4             Positive                     2                    1
## 5             Positive                     3                    1
## 6             Positive                    18                    2
## 7             Positive                    11                    1
## 8             Positive                     9                    1
## 9             Positive                    20                   18
## 10            Positive                    21                   12
##    Survival.Months Status
## 1               60  Alive
## 2               62  Alive
```

```
## 3                75  Alive
## 4                84  Alive
## 5                50  Alive
## 6                89  Alive
## 7                54  Alive
## 8                14   Dead
## 9                70  Alive
## 10               92  Alive
```

```
numerical_summary <- data %>%
  select_if(is.numeric) %>%
  summarise_all(list(
    count = ~sum(!is.na(.)),
    mean = mean,
    std = sd,
    min = min,
    median = median,
    max = max
  )) %>%
  pivot_longer(cols = everything(), names_to = "Variable", values_to = "Value") %>%
  separate(Variable, into = c("Variable", "Statistic"), sep = "_")

formatted_summary <- numerical_summary %>%
  pivot_wider(names_from = Statistic, values_from = Value)

kable(formatted_summary, col.names = c("Variable", "Count", "Mean", "Std", "Min", "Median", "Max"), cap
```

Table 1: Numerical Variables Summary Statistics

Variable	Count	Mean	Std	Min	Median	Max
Age	4024	53.972167	8.963134	30	54	69
Tumor.Size	4024	30.473658	21.119696	1	25	140
Regional.Node.Examined	4024	14.357107	8.099675	1	14	61
Reginol.Node.Positive	4024	4.158052	5.109331	1	2	46
Survival.Months	4024	71.297962	22.921429	1	73	107

```
categorical_vars <- data %>% select_if(is.character)

category_summary <- categorical_vars %>%
  gather(Variable, Category) %>%
  group_by(Variable, Category) %>%
  summarise(Count = n()) %>%
  mutate(Percentage = round((Count / sum(Count)) * 100, 2)) %>%
  arrange(Variable, desc(Count))
```

```
## `summarise()` has grouped output by 'Variable'. You can override using the
## `.groups` argument.
```

```
formatted_summary <- category_summary %>%
  group_by(Variable) %>%
  mutate(Variable = ifelse(row_number() == 1, Variable, ""))

kable(formatted_summary, col.names = c("Variable", "Category", "Count", "Percentage (%)"), caption = "Ca
```

Table 2: Category Distribution of Categorical Variables

Variable	Category	Count	Percentage (%)
A.Stage	Regional	3932	97.71
	Distant	92	2.29
Estrogen.Status	Positive	3755	93.32
	Negative	269	6.68
Grade	2	2351	58.42
	3	1111	27.61
	1	543	13.49
	anaplastic; Grade IV	19	0.47
Marital.Status	Married	2643	65.68
	Single	615	15.28
	Divorced	486	12.08
	Widowed	235	5.84
	Separated	45	1.12
N.Stage	N1	2732	67.89
	N2	820	20.38
	N3	472	11.73
Progesterone.Status	Positive	3326	82.65
	Negative	698	17.35
Race	White	3413	84.82
	Other	320	7.95
	Black	291	7.23
Status	Alive	3408	84.69
	Dead	616	15.31
T.Stage	T2	1786	44.38
	T1	1603	39.84
	T3	533	13.25
	T4	102	2.53
X6th.Stage	IIA	1305	32.43
	IIB	1130	28.08
	IIIA	1050	26.09
	IIIC	472	11.73
	IIIB	67	1.67
differentiate	Moderately differentiated	2351	58.42
	Poorly differentiated	1111	27.61
	Well differentiated	543	13.49
	Undifferentiated	19	0.47

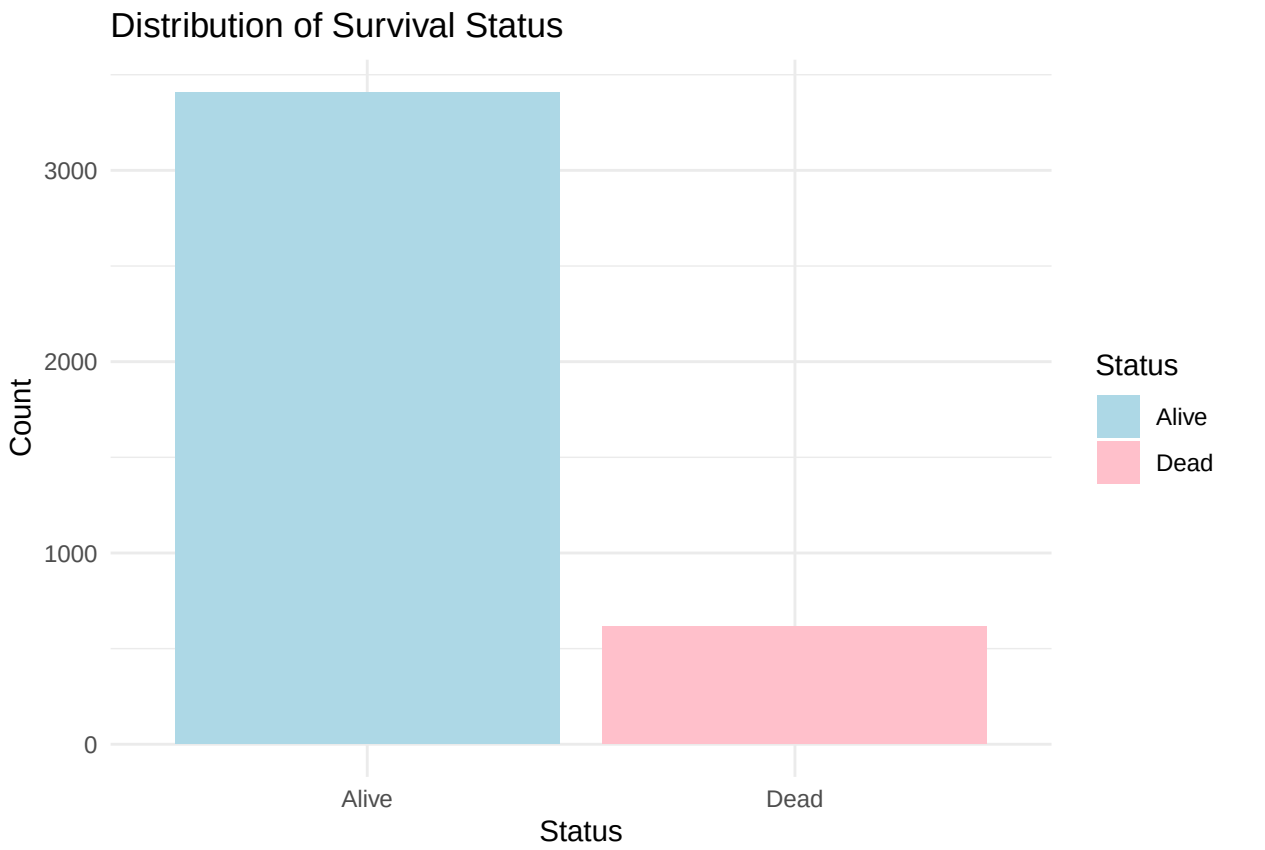
Explore the Distribution of the Outcome (Status: Dead / Alive)

```
status_distribution <- data %>%
  group_by(Status) %>%
  summarise(Count = n()) %>%
  mutate(Proportion = Count / sum(Count))

kable(status_distribution, col.names = c("Status", "Count", "Proportion"), caption = "Distribution of St
```

Table 3: Distribution of Survival Status (Dead/Alive)

Status	Count	Proportion
Alive	3408	0.8469185
Dead	616	0.1530815

```
ggplot(data, aes(x = Status, fill = Status)) +
  geom_bar() +
  labs(title = "Distribution of Survival Status", x = "Status", y = "Count") +
  theme_minimal() +
  scale_fill_manual(values = c("lightblue", "pink"))
```



For logistic regression, the binary outcome variable (Status: Dead/Alive) does not require transformation, as logistic regression inherently models binary outcomes.

Transformation

```
# Identify numerical variables
numerical_vars <- data %>%
  select_if(is.numeric) %>%
  select(-`Survival.Months`)

# Display the list of numerical variables
names(numerical_vars)
```

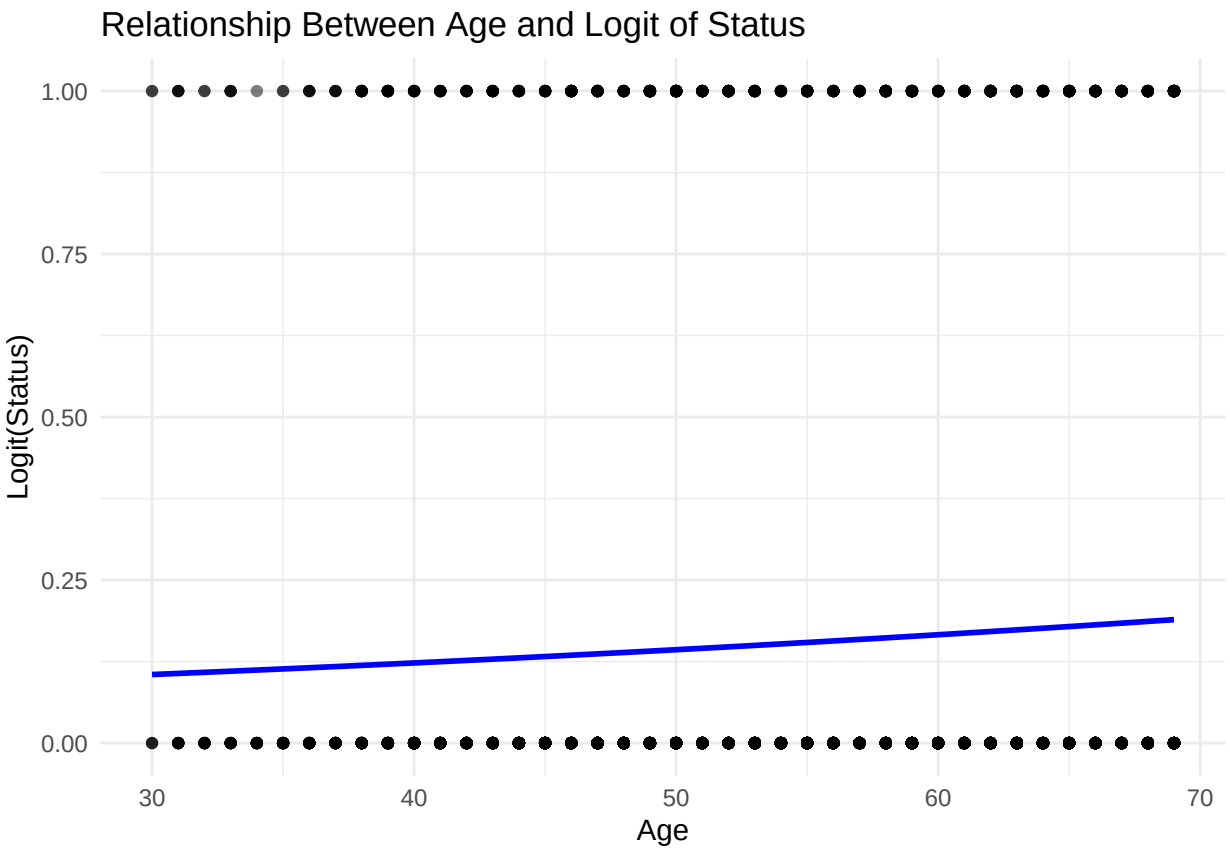
```
## [1] "Age"                    "Tumor.Size"             "Regional.Node.Examined"
```

```
## [4] "Reginol.Node.Positive"
```

```r
# Convert Status to a binary numeric variable
data$Status <- ifelse(data$Status == "Dead", 1, 0)

# Scatterplots for each numerical variable against the logit
logit <- function(p) log(p / (1 - p))  # Logit function

numerical_vars %>%
  names() %>%
  map(~ ggplot(data, aes(x = .data[[.x]], y = Status)) +
        stat_smooth(method = "glm", method.args = list(family = "binomial"), se = FALSE, color = "blue")
        geom_point(alpha = 0.5) +
        labs(title = paste("Relationship Between", .x, "and Logit of Status"),
             x = .x,
             y = "Logit(Status)") +
        theme_minimal())
```

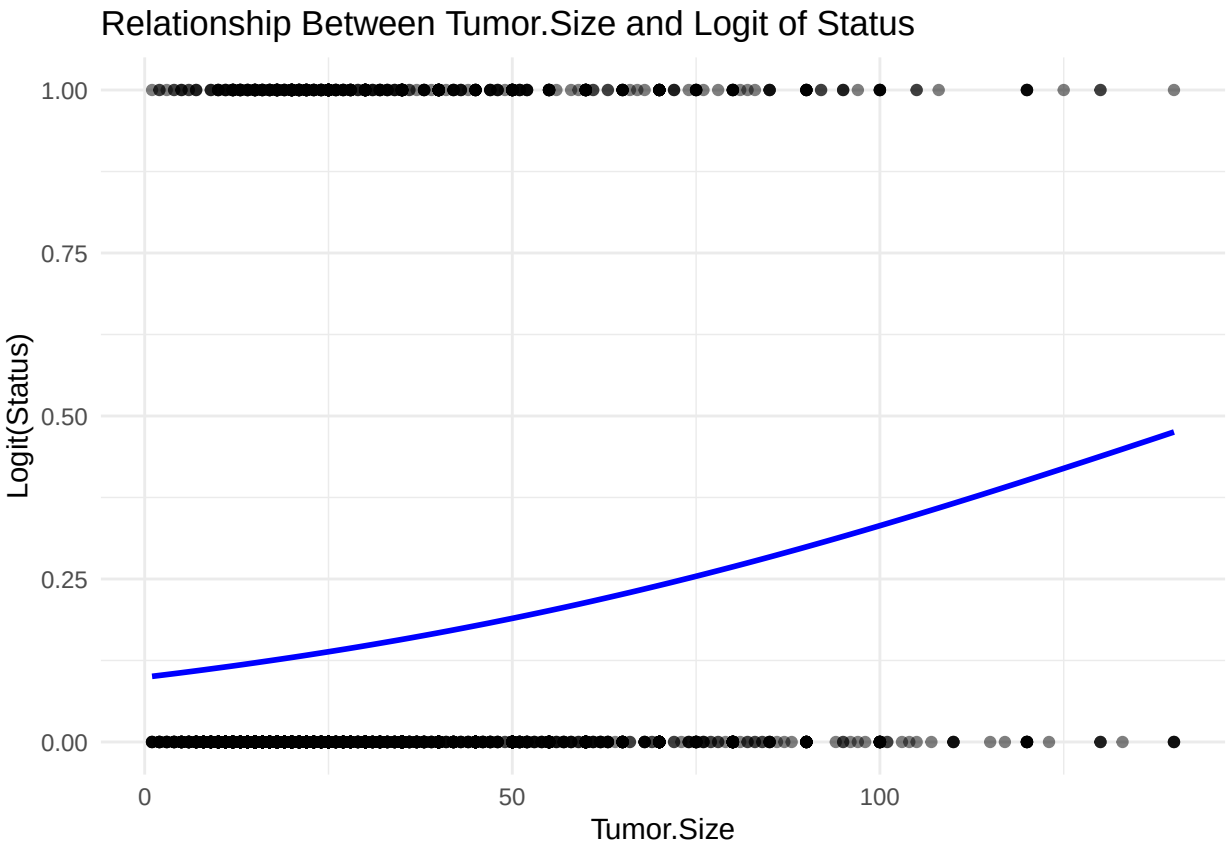
```
## [[1]]
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



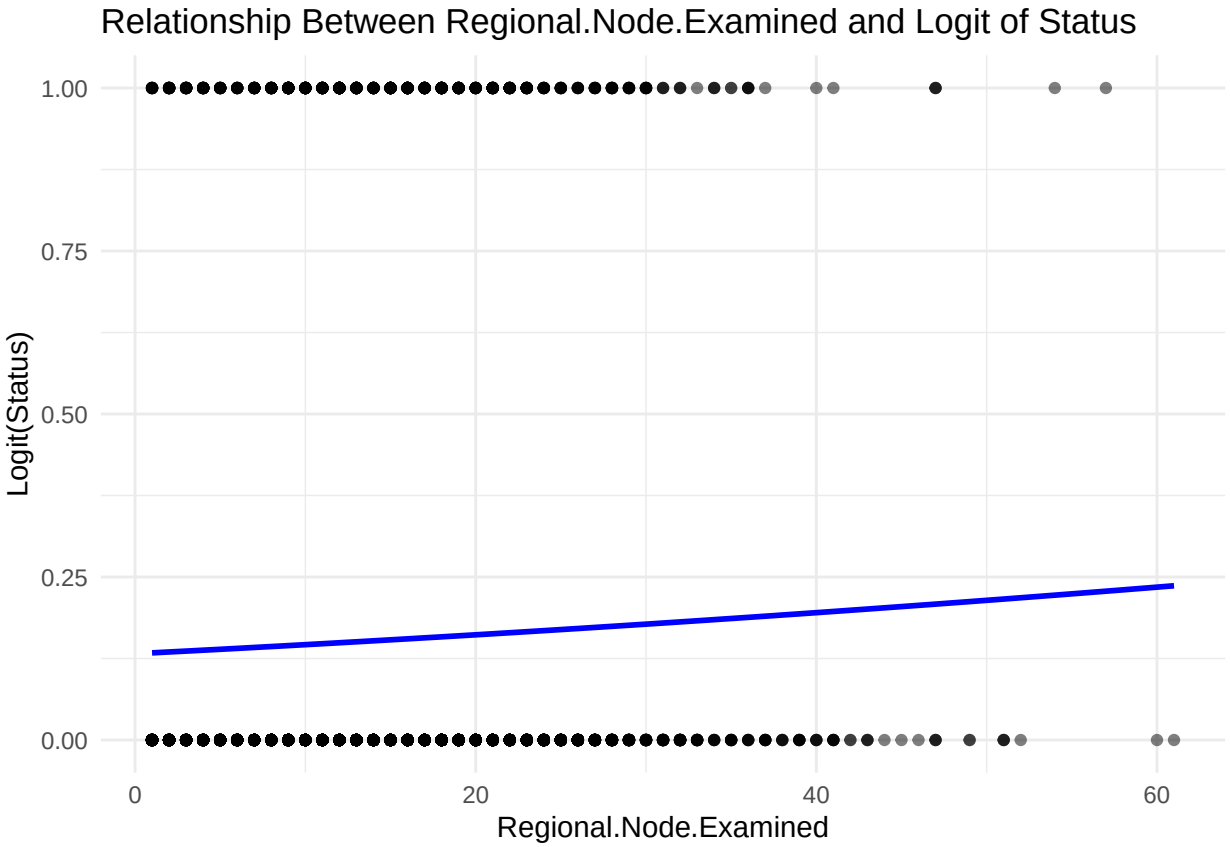

```
##
## [[2]]
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



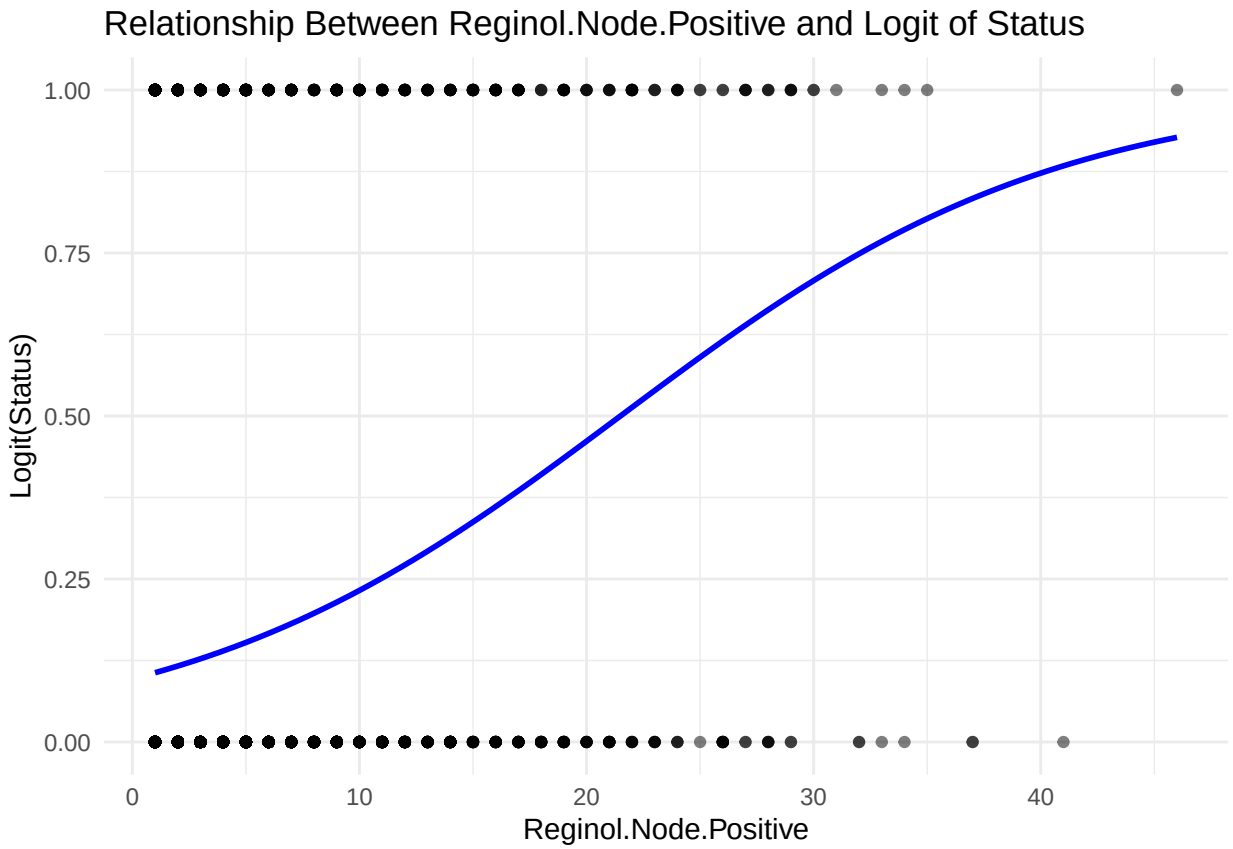

```
##
## [[3]]
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
##
## [[4]]
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
# Calculate skewness for numerical variables
numerical_skewness <- numerical_vars %>%
  map_df(~ tibble(Variable = deparse(substitute(.)),
                  Skewness = skewness(., na.rm = TRUE)))

# Correct the Variable column
numerical_skewness <- tibble(
  Variable = colnames(numerical_vars),
  Skewness = sapply(numerical_vars, skewness, na.rm = TRUE)
)

# Display the skewness table
kable(numerical_skewness, col.names = c("Variable", "Skewness"), caption = "Skewness of Numerical Varial
```

Table 4: Skewness of Numerical Variables

Variable	Skewness
Age	-0.2202085
Tumor.Size	1.7384530
Regional.Node.Examined	0.8286556
Reginol.Node.Positive	2.7005214

After our initial detection, we found out that:

- Reginol.Node.Positive variable show slightly nonlinear with the logit of Status, it need transformation.
- The skewness analysis reveals that Age (-0.22) has a roughly symmetric distribution, requiring no

transformation. Tumor Size (1.74) shows moderate right skewness, suggesting a potential log transformation to normalize the distribution, though it may not be strictly necessary. Regional Node Examined (0.83) has mild positive skewness and can likely be retained in its current form unless further diagnostics indicate otherwise. Reginol Node Positive (2.70), with significant right skewness, would benefit from a log transformation to reduce skewness and stabilize its relationship with the logit in the logistic regression model. These adjustments ensure numerical variables are well-prepared for regression analysis.

Base on the analysis above, try to make log transformation on Reginol Node Positive & Tumor Size.

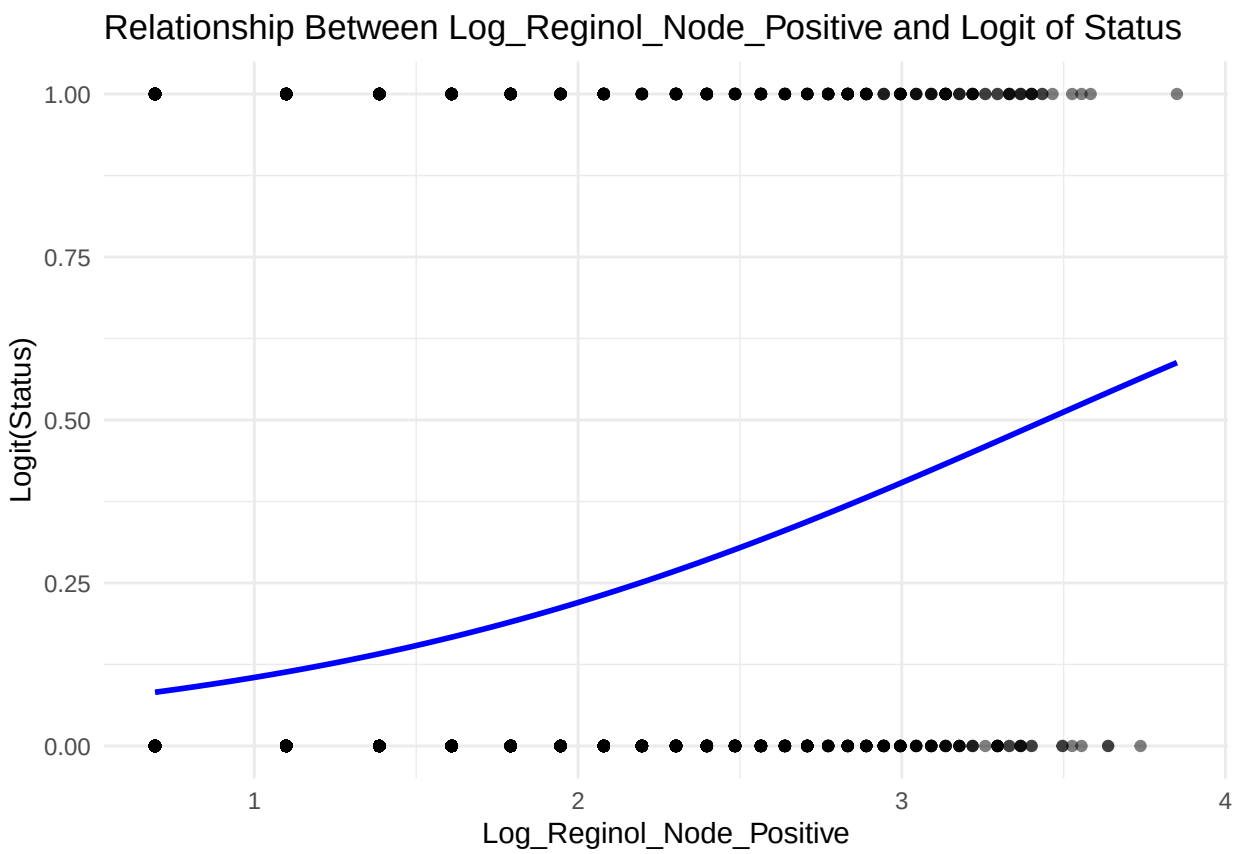
```
data <- data %>%
  mutate(
    Log_Reginol_Node_Positive = log1p(`Reginol.Node.Positive`),
    Log_Tumor_Size = log1p(`Tumor.Size`)
  )
transformed_skewness <- data %>%
  select(Log_Reginol_Node_Positive, Log_Tumor_Size) %>%
  summarise_all(~ skewness(.))

# Combine with variable names
transformed_skewness_table <- tibble(
  Variable = c("Log_Reginol_Node_Positive", "Log_Tumor_Size"),
  Skewness = as.numeric(transformed_skewness)
)

# Display the updated skewness table
kable(transformed_skewness_table, col.names = c("Variable", "Skewness"), caption = "Skewness of Transfo
```

Table 5: Skewness of Transformed Variables

Variable	Skewness
Log_Reginol_Node_Positive	0.9887072
Log_Tumor_Size	-0.0874903

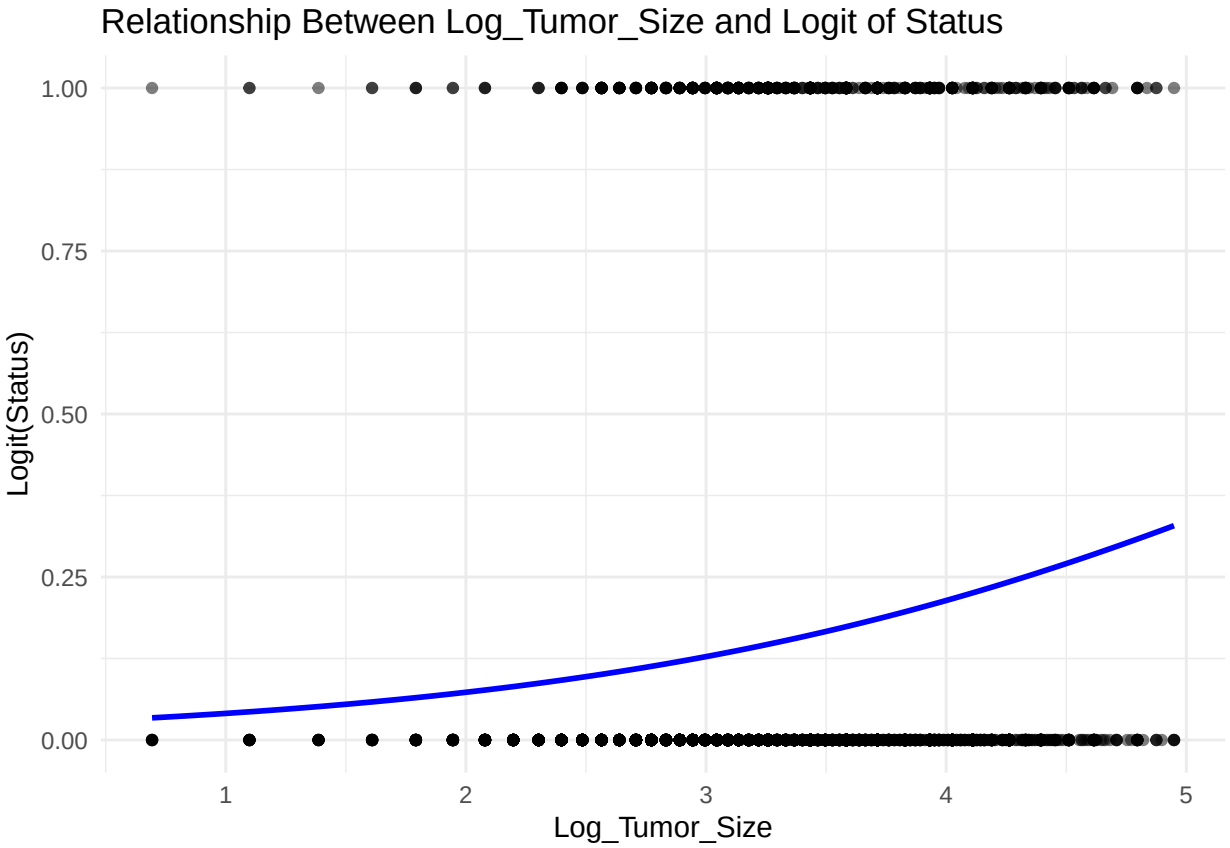
```
## Plots for Transformed Variables Against Logit of Status
logit <- function(p) log(p / (1 - p))  # Logit function

# Plot for Log_Reginol_Node_Positive
ggplot(data, aes(x = Log_Reginol_Node_Positive, y = Status)) +
  stat_smooth(method = "glm", method.args = list(family = "binomial"), se = FALSE, color = "blue") +
  geom_point(alpha = 0.5) +
  labs(title = "Relationship Between Log_Reginol_Node_Positive and Logit of Status", x = "Log_Reginol_N
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
# Plot for Log_Tumor_Size
ggplot(data, aes(x = Log_Tumor_Size, y = Status)) +
  stat_smooth(method = "glm", method.args = list(family = "binomial"), se = FALSE, color = "blue") +
  geom_point(alpha = 0.5) +
  labs(title = "Relationship Between Log_Tumor_Size and Logit of Status", x = "Log_Tumor_Size", y = "Log
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Comments:

- For Log_Reginol_Node_Positive, the skewness improved from 2.70 to 0.99, indicating a significant reduction in skewness. While still slightly positively skewed, the value is now within an acceptable range for modeling.
- For Log_Tumor_Size, the skewness reduced from 1.74 to -0.09, making it almost symmetric. This transformation effectively normalized the variable.
- For Log_Reginol_Node_Positive, the log transformation on Reginol.Node.Positive likely improved its relationship with the logit as well.
- For Log_Tumor_Size, after the transformation the linearity with the logit has not significantly improved, but at least the skewness reduced.

As a result, we should definitely conduct a log transformation on Reginol.Node.Positive and Tumor.Size (so the two original variables can be removed).

```
data=data |>
  select(-`Reginol.Node.Positive`, -`Tumor.Size`)
```

```
# Compute correlation coefficients among numeric variables
selected_vars <- c("Age", "Log_Reginol_Node_Positive", "Log_Tumor_Size","Regional.Node.Examined")

subset_data <- data[, selected_vars]

if (all(sapply(subset_data, is.numeric))) {
  correlation_matrix <- cor(subset_data, use = "pairwise.complete.obs")
  print(correlation_matrix)
}
```

```
##                                    Age Log_Reginol_Node_Positive Log_Tumor_Size
## Age                         1.000000000              -0.005830503    -0.08138322
## Log_Reginol_Node_Positive -0.005830503               1.000000000     0.30720387
## Log_Tumor_Size            -0.081383224               0.307203871     1.00000000
## Regional.Node.Examined    -0.033345483               0.395974275     0.11628210
##                           Regional.Node.Examined
## Age                                -0.03334548
## Log_Reginol_Node_Positive           0.39597428
## Log_Tumor_Size                      0.11628210
## Regional.Node.Examined              1.00000000
```

None of the correlation coefficients between numeric variables exceed 0.5, indicating that there is no strong linear relationship between each pair of numeric variables.

```
# Identify highly consistent category variables
contingency_table <- table(data[["differentiate"]], data[["Grade"]])
print(contingency_table)
```

```
##
##                              anaplastic; Grade IV    1    2    3
##   Moderately differentiated                     0    0 2351    0
##   Poorly differentiated                         0    0    0 1111
##   Undifferentiated                             19    0    0    0
##   Well differentiated                           0  543    0    0
```

We discover that complete linear dependency exist among Grade and differentiate, so we can only include one of them in the prediction model, so differentiate is excluded.

```
data=data |>
  select(-differentiate)
```

Finally, we need to change all the catagorical variables to dummy variables:

```
categorical_vars <- data %>%
  select_if(is.character) %>%
  names()

data_final <- data %>%
  mutate(across(all_of(categorical_vars), ~ as.factor(.))) %>%
  model.matrix(~ . - 1, data = .) %>%
  as.data.frame()

data_final = data_final |>
  select(-`Survival.Months`)

head(data_final,10)
```

```
##    Age RaceBlack RaceOther RaceWhite Marital.StatusMarried
## 1   68         0         0         1                     1
## 2   50         0         0         1                     1
## 3   58         0         0         1                     0
## 4   58         0         0         1                     1
## 5   47         0         0         1                     1
## 6   51         0         0         1                     0
## 7   51         0         0         1                     1
## 8   40         0         0         1                     1
## 9   40         0         0         1                     0
```

```
## 10  69         0         0         1                    1
##    Marital.StatusSeparated Marital.StatusSingle  Marital.StatusWidowed
## 1                        0                    0                     0
## 2                        0                    0                     0
## 3                        0                    0                     0
## 4                        0                    0                     0
## 5                        0                    0                     0
## 6                        0                    1                     0
## 7                        0                    0                     0
## 8                        0                    0                     0
## 9                        0                    0                     0
## 10                       0                    0                     0
##    T.StageT2 T.StageT3 T.StageT4 N.StageN2 N.StageN3 X6th.StageIIB
## 1          0         0         0         0         0             0
## 2          1         0         0         1         0             0
## 3          0         1         0         0         1             0
## 4          0         0         0         0         0             0
## 5          1         0         0         0         0             1
## 6          0         0         0         0         0             0
## 7          0         0         0         0         0             0
## 8          1         0         0         0         0             1
## 9          0         0         1         0         1             0
## 10         0         0         1         0         1             0
##    X6th.StageIIIA X6th.StageIIIB X6th.StageIIIC Grade1 Grade2 Grade3
## 1               0              0              0      0      0      1
## 2               1              0              0      0      1      0
## 3               0              0              1      0      1      0
## 4               0              0              0      0      0      1
## 5               0              0              0      0      0      1
## 6               0              0              0      0      1      0
## 7               0              0              0      1      0      0
## 8               0              0              0      0      1      0
## 9               0              0              1      0      0      1
## 10              0              0              1      1      0      0
##    A.StageRegional Estrogen.StatusPositive Progesterone.StatusPositive
## 1                1                       1                           1
## 2                1                       1                           1
## 3                1                       1                           1
## 4                1                       1                           1
## 5                1                       1                           1
## 6                1                       1                           1
## 7                1                       1                           1
## 8                1                       1                           1
## 9                1                       1                           1
## 10               0                       1                           1
##    Regional.Node.Examined Status Log_Reginol_Node_Positive Log_Tumor_Size
## 1                      24      0                0.6931472       1.609438
## 2                      14      0                1.7917595       3.583519
## 3                      14      0                2.0794415       4.158883
## 4                       2      0                0.6931472       2.944439
## 5                       3      0                0.6931472       3.737670
## 6                      18      0                1.0986123       3.044522
## 7                      11      0                0.6931472       2.197225
## 8                       9      1                0.6931472       3.433987
```

```
## 9                      20      0                2.9444390       4.644391
## 10                     21      0                2.5649494       3.496508
```

After basic data preprocessing, we use lasso regression to help select the variables used as predictors.

```
x <- model.matrix(Status ~ ., data = data_final)[, -1]
y <- data_final$Status

# Perform cross-validation for Lasso regression
lasso_cv <- cv.glmnet(x, y, family = "binomial", alpha = 1)

# Get the optimal regularization parameter lambda
best_lambda <- lasso_cv$lambda.min
print(paste("Optimal lambda:", best_lambda))
```

```
## [1] "Optimal lambda: 0.00103892293095532"
```

```
# Fit the Lasso model using the optimal lambda
lasso_model <- glmnet(x, y, family = "binomial", alpha = 1, lambda = best_lambda)

# Extract the coefficients from the Lasso model
lasso_coefficients <- coef(lasso_model)

# Convert coefficients to a standard matrix format
lasso_coefficients_matrix <- as.matrix(lasso_coefficients)

# Extract variable names with non-zero coefficients (excluding the intercept)
selected_vars <- rownames(lasso_coefficients_matrix)[lasso_coefficients_matrix[, 1] != 0][-1]

# Output the selected variables
print("Selected variables:")
```

```
## [1] "Selected variables:"
```

```
print(selected_vars)
```

```
##  [1] "Age"                          "RaceBlack"
##  [3] "RaceOther"                    "Marital.StatusMarried"
##  [5] "Marital.StatusSeparated"      "Marital.StatusWidowed"
##  [7] "T.StageT2"                    "T.StageT3"
##  [9] "T.StageT4"                    "N.StageN2"
## [11] "N.StageN3"                    "X6th.StageIIB"
## [13] "X6th.StageIIIB"               "X6th.StageIIIC"
## [15] "Grade1"                       "Grade2"
## [17] "A.StageRegional"              "Estrogen.StatusPositive"
## [19] "Progesterone.StatusPositive" "Regional.Node.Examined"
## [21] "Log_Reginol_Node_Positive"    "Log_Tumor_Size"
```

```
# Construct the logistic regression formula
final_formula <- as.formula(paste("Status ~", paste(selected_vars, collapse = " + ")))

# Fit the final logistic regression model
final_model <- glm(final_formula, data = data_final, family = "binomial")

# Output the summary of the final model
summary(final_model)
```

```
##
## Call:
## glm(formula = final_formula, family = "binomial", data = data_final)
##
## Coefficients: (1 not defined because of singularities)
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -2.519832   0.604361  -4.169 3.05e-05 ***
## Age                          0.024975   0.005576   4.479 7.51e-06 ***
## RaceBlack                    0.493886   0.160715   3.073 0.002119 **
## RaceOther                   -0.422082   0.202125  -2.088 0.036778 *
## Marital.StatusMarried       -0.183285   0.106633  -1.719 0.085643 .
## Marital.StatusSeparated      0.711849   0.370708   1.920 0.054828 .
## Marital.StatusWidowed        0.048349   0.200724   0.241 0.809655
## T.StageT2                    0.193251   0.196814   0.982 0.326150
## T.StageT3                    0.407837   0.275094   1.483 0.138197
## T.StageT4                    0.865917   0.440379   1.966 0.049264 *
## N.StageN2                    0.283352   0.200269   1.415 0.157111
## N.StageN3                    0.657838   0.294561   2.233 0.025530 *
## X6th.StageIIB                0.248681   0.196218   1.267 0.205021
## X6th.StageIIIB               0.128930   0.477963   0.270 0.787353
## X6th.StageIIIC                     NA         NA      NA       NA
## Grade1                      -0.927008   0.192261  -4.822 1.42e-06 ***
## Grade2                      -0.402494   0.103945  -3.872 0.000108 ***
## A.StageRegional             -0.056295   0.264024  -0.213 0.831157
## Estrogen.StatusPositive     -0.777335   0.177491  -4.380 1.19e-05 ***
## Progesterone.StatusPositive -0.561011   0.127532  -4.399 1.09e-05 ***
## Regional.Node.Examined      -0.032028   0.006960  -4.601 4.20e-06 ***
## Log_Reginol_Node_Positive    0.644056   0.143833   4.478 7.54e-06 ***
## Log_Tumor_Size               0.033313   0.149431   0.223 0.823591
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3444.7  on 4023  degrees of freedom
## Residual deviance: 2962.1  on 4002  degrees of freedom
## AIC: 3006.1
##
## Number of Fisher Scoring iterations: 5
```

```
# Check for linear dependencies (aliased variables)
alias_info <- alias(final_model)
print(alias_info)
```

```
## Model :
## Status ~ Age + RaceBlack + RaceOther + Marital.StatusMarried +
##     Marital.StatusSeparated + Marital.StatusWidowed + T.StageT2 +
##     T.StageT3 + T.StageT4 + N.StageN2 + N.StageN3 + X6th.StageIIB +
##     X6th.StageIIIB + X6th.StageIIIC + Grade1 + Grade2 + A.StageRegional +
##     Estrogen.StatusPositive + Progesterone.StatusPositive + Regional.Node.Examined +
##     Log_Reginol_Node_Positive + Log_Tumor_Size
##
## Complete :
##                (Intercept) Age RaceBlack RaceOther Marital.StatusMarried
## X6th.StageIIIC 0           0   0         0         0
```

```
##                 Marital.StatusSeparated Marital.StatusWidowed T.StageT2
## X6th.StageIIIC 0                       0                     0
##                 T.StageT3 T.StageT4 N.StageN2 N.StageN3 X6th.StageIIB
## X6th.StageIIIC 0         0         0         1         0
##                 X6th.StageIIIB Grade1 Grade2 A.StageRegional
## X6th.StageIIIC 0              0      0      0
##                 Estrogen.StatusPositive Progesterone.StatusPositive
## X6th.StageIIIC 0                        0
##                 Regional.Node.Examined Log_Reginol_Node_Positive Log_Tumor_Size
## X6th.StageIIIC 0                       0                         0
```

```
if (!is.null(alias_info$Complete)) {
  aliased_vars <- rownames(alias_info$Complete)
  print("Aliased (linearly dependent) variables:")
  print(aliased_vars)
} else {
  print("No aliased coefficients found.")
}
```

```
## [1] "Aliased (linearly dependent) variables:"
## [1] "X6th.StageIIIC"
```

X6th.StageIIIC was identified as an aliased (linearly dependent) variable, being perfectly correlated with N.StageN3. This redundancy can cause multicollinearity issues and instability in coefficient estimation. Both X6th.StageIIIC and N.StageN3 were removed to ensure a more stable and interpretable model.

```
# Remove X6th.StageIIIC and N.StageN3 from selected_vars
vars_to_remove <- c("X6th.StageIIIC", "N.StageN3")
selected_vars_updated <- setdiff(selected_vars, vars_to_remove)

# Output the updated selected variables
print("Updated selected variables:")
```

```
## [1] "Updated selected variables:"
```

```
print(selected_vars_updated)
```

```
##  [1] "Age"                          "RaceBlack"
##  [3] "RaceOther"                    "Marital.StatusMarried"
##  [5] "Marital.StatusSeparated"      "Marital.StatusWidowed"
##  [7] "T.StageT2"                    "T.StageT3"
##  [9] "T.StageT4"                    "N.StageN2"
## [11] "X6th.StageIIB"                "X6th.StageIIIB"
## [13] "Grade1"                       "Grade2"
## [15] "A.StageRegional"              "Estrogen.StatusPositive"
## [17] "Progesterone.StatusPositive" "Regional.Node.Examined"
## [19] "Log_Reginol_Node_Positive"    "Log_Tumor_Size"
```

```
# Construct the updated logistic regression formula
updated_formula <- as.formula(paste("Status ~", paste(selected_vars_updated, collapse = " + ")))

# Fit the updated logistic regression model
final_model_updated <- glm(updated_formula, data = data_final, family = "binomial")

# Output the summary of the updated model
summary(final_model_updated)
```

```
##
```

```
## Call:
## glm(formula = updated_formula, family = "binomial", data = data_final)
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -2.586868   0.604324  -4.281 1.86e-05 ***
## Age                          0.024919   0.005571   4.473 7.71e-06 ***
## RaceBlack                    0.507115   0.160256   3.164  0.00155 **
## RaceOther                   -0.426369   0.202063  -2.110  0.03485 *
## Marital.StatusMarried       -0.185979   0.106429  -1.747  0.08056 .
## Marital.StatusSeparated      0.685081   0.371771   1.843  0.06537 .
## Marital.StatusWidowed        0.048188   0.200576   0.240  0.81014
## T.StageT2                    0.289094   0.192589   1.501  0.13333
## T.StageT3                    0.439822   0.273499   1.608  0.10781
## T.StageT4                    0.993099   0.437609   2.269  0.02325 *
## N.StageN2                   -0.066720   0.124254  -0.537  0.59129
## X6th.StageIIB                0.077969   0.180760   0.431  0.66622
## X6th.StageIIIB              -0.058609   0.471980  -0.124  0.90118
## Grade1                      -0.939837   0.191920  -4.897 9.73e-07 ***
## Grade2                      -0.414513   0.103628  -4.000 6.33e-05 ***
## A.StageRegional             -0.142220   0.263709  -0.539  0.58968
## Estrogen.StatusPositive     -0.791877   0.176913  -4.476 7.60e-06 ***
## Progesterone.StatusPositive -0.564563   0.127217  -4.438 9.09e-06 ***
## Regional.Node.Examined      -0.032821   0.006960  -4.715 2.41e-06 ***
## Log_Reginol_Node_Positive    0.898326   0.087884  10.222  < 2e-16 ***
## Log_Tumor_Size               0.033660   0.149132   0.226  0.82143
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3444.7  on 4023  degrees of freedom
## Residual deviance: 2967.1  on 4003  degrees of freedom
## AIC: 3009.1
##
## Number of Fisher Scoring iterations: 5
```

```
# Calculate Variance Inflation Factor (VIF) for the updated model
vif_values_updated <- vif(final_model_updated)
print("Variance Inflation Factors (VIF) for the updated model:")
```

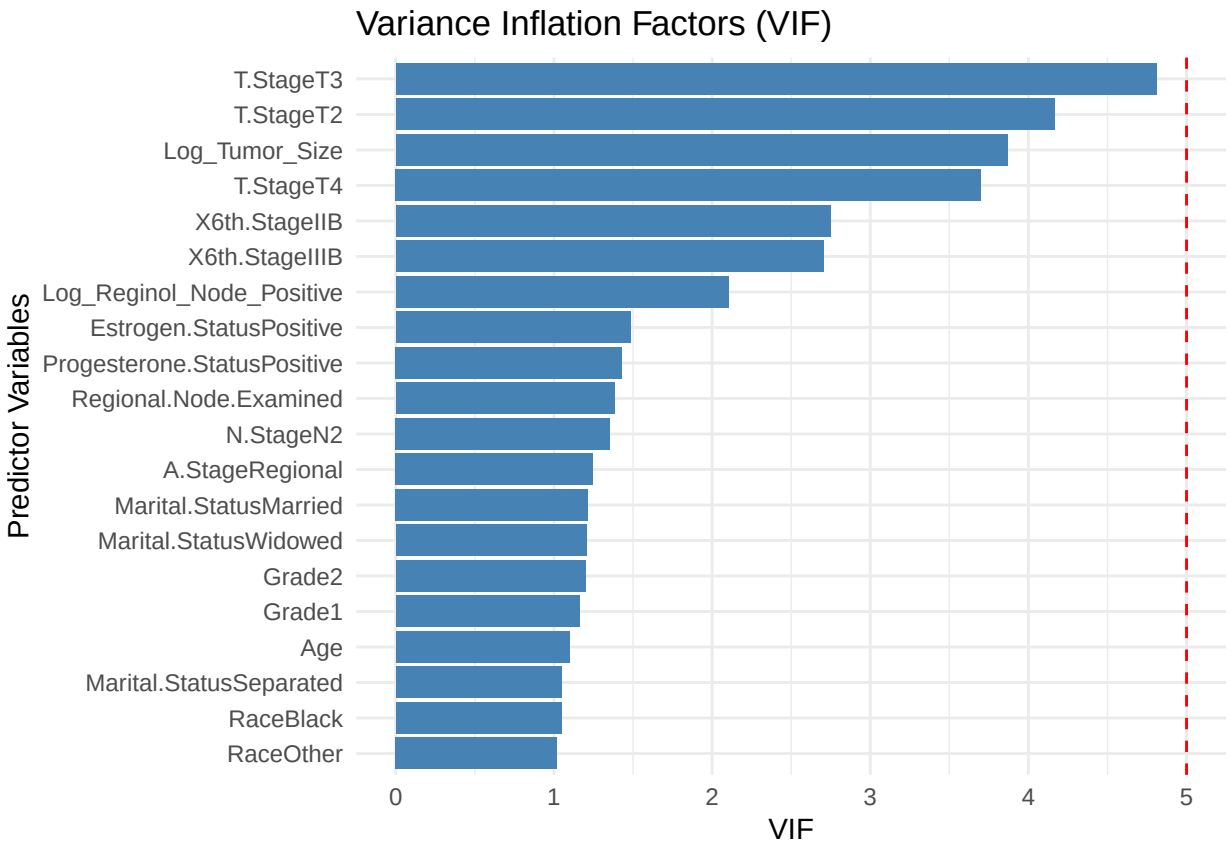
```
## [1] "Variance Inflation Factors (VIF) for the updated model:"
```

```
print(vif_values_updated)
```

```
##                          Age                    RaceBlack
##                     1.098778                     1.047286
##                    RaceOther       Marital.StatusMarried
##                     1.019144                     1.214976
##     Marital.StatusSeparated       Marital.StatusWidowed
##                     1.048147                     1.205814
##                    T.StageT2                    T.StageT3
##                     4.163568                     4.808712
##                    T.StageT4                    N.StageN2
##                     3.699579                     1.354531
```

```
##                X6th.StageIIB              X6th.StageIIIB
##                     2.749446                    2.704404
##                      Grade1                      Grade2
##                     1.161258                    1.198231
##             A.StageRegional     Estrogen.StatusPositive
##                     1.247616                    1.485066
## Progesterone.StatusPositive      Regional.Node.Examined
##                     1.430544                    1.381053
##   Log_Reginol_Node_Positive             Log_Tumor_Size
##                     2.105050                    3.867394
```

```r
vif_df <- data.frame(
  Variable = names(vif_values_updated),
  VIF = vif_values_updated
)

ggplot(vif_df, aes(x = reorder(Variable, VIF), y = VIF)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  geom_hline(yintercept = 5, color = "red", linetype = "dashed") +
  labs(title = "Variance Inflation Factors (VIF)",
       x = "Predictor Variables",
       y = "VIF") +
  coord_flip() +
  theme_minimal()
```



After removing X6th.StageIIIC and N.StageN3 and refitting the updated logistic regression model, the Variance Inflation Factor (VIF) values were recalculated. The results indicate that all predictor variables now have VIF values below 5, suggesting that multicollinearity among the independent variables has been

successfully resolved.

We can extract coefficients from the updated final model and show them.

```
# Extract coefficients from the updated final model
coefficients_updated <- summary(final_model_updated)$coefficients

# Create a summary table
summary_table_updated <- data.frame(
  Variable = rownames(coefficients_updated),
  Estimate = coefficients_updated[, "Estimate"],
  Std_Error = coefficients_updated[, "Std. Error"],
  z_value = coefficients_updated[, "z value"],
  p_value = coefficients_updated[, "Pr(>|z|)"]
)

# Calculate Odds Ratios and Confidence Intervals
summary_table_updated <- summary_table_updated %>%
  mutate(
    Odds_Ratio = exp(Estimate),
    CI_Lower = exp(Estimate - 1.96 * Std_Error),
    CI_Upper = exp(Estimate + 1.96 * Std_Error)
  ) %>%
  select(Variable, Estimate, Odds_Ratio, CI_Lower, CI_Upper, p_value)

# Display the summary table
kable(summary_table_updated, digits = 3, caption = "Final Logistic Regression Model Summary")
```

Table 6: Final Logistic Regression Model Summary

	Variable	Estimate	Odds_Ratio	CI_Lower	CI_Upper	p_value
(Intercept)	(Intercept)	-2.587	0.075	0.023	0.246	0.000
Age	Age	0.025	1.025	1.014	1.036	0.000
RaceBlack	RaceBlack	0.507	1.660	1.213	2.273	0.002
RaceOther	RaceOther	-0.426	0.653	0.439	0.970	0.035
Marital.StatusMarried	Marital.StatusMarried	-0.186	0.830	0.674	1.023	0.081
Marital.StatusSeparated	Marital.StatusSeparated	0.685	1.984	0.957	4.111	0.065
Marital.StatusWidowed	Marital.StatusWidowed	0.048	1.049	0.708	1.555	0.810
T.StageT2	T.StageT2	0.289	1.335	0.915	1.948	0.133
T.StageT3	T.StageT3	0.440	1.552	0.908	2.654	0.108
T.StageT4	T.StageT4	0.993	2.700	1.145	6.365	0.023
N.StageN2	N.StageN2	-0.067	0.935	0.733	1.193	0.591
X6th.StageIIB	X6th.StageIIB	0.078	1.081	0.759	1.541	0.666
X6th.StageIIIB	X6th.StageIIIB	-0.059	0.943	0.374	2.379	0.901
Grade1	Grade1	-0.940	0.391	0.268	0.569	0.000
Grade2	Grade2	-0.415	0.661	0.539	0.809	0.000
A.StageRegional	A.StageRegional	-0.142	0.867	0.517	1.454	0.590
Estrogen.StatusPositive	Estrogen.StatusPositive	-0.792	0.453	0.320	0.641	0.000
Progesterone.StatusPositive	Progesterone.StatusPositive	-0.565	0.569	0.443	0.730	0.000
Regional.Node.Examined	Regional.Node.Examined	-0.033	0.968	0.955	0.981	0.000
Log_Reginol_Node_Positive	Log_Reginol_Node_Positive	0.898	2.455	2.067	2.917	0.000
Log_Tumor_Size	Log_Tumor_Size	0.034	1.034	0.772	1.385	0.821

Then we need to evaluate model performance by computing some matrix. Set the threshold to 0.5, meaning

that any predicted probability greater than or equal to 0.5 is classified as the positive class (1) (in our case, dead), while probabilities below 0.5 are classified as the negative class (0) (in our case, alive).

```
threshold <- 0.5
# Predict probabilities using the updated model
pred_probs_updated <- predict(final_model_updated, type = "response")

pred_classes_updated <- ifelse(pred_probs_updated >= threshold, 1, 0)

# Generate the confusion matrix for the updated model
conf_matrix_updated <- confusionMatrix(as.factor(pred_classes_updated), as.factor(y), positive = "1")
print(conf_matrix_updated)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 3358  533
##          1   50   83
##
##                Accuracy : 0.8551
##                  95% CI : (0.8439, 0.8659)
##     No Information Rate : 0.8469
##     P-Value [Acc > NIR] : 0.07661
##
##                   Kappa : 0.1769
##
##  Mcnemar's Test P-Value : < 2e-16
##
##             Sensitivity : 0.13474
##             Specificity : 0.98533
##          Pos Pred Value : 0.62406
##          Neg Pred Value : 0.86302
##              Prevalence : 0.15308
##          Detection Rate : 0.02063
##    Detection Prevalence : 0.03305
##       Balanced Accuracy : 0.56003
##
##        'Positive' Class : 1
##
```

We also need to draw the Receiver Operating Characteristic Curve.

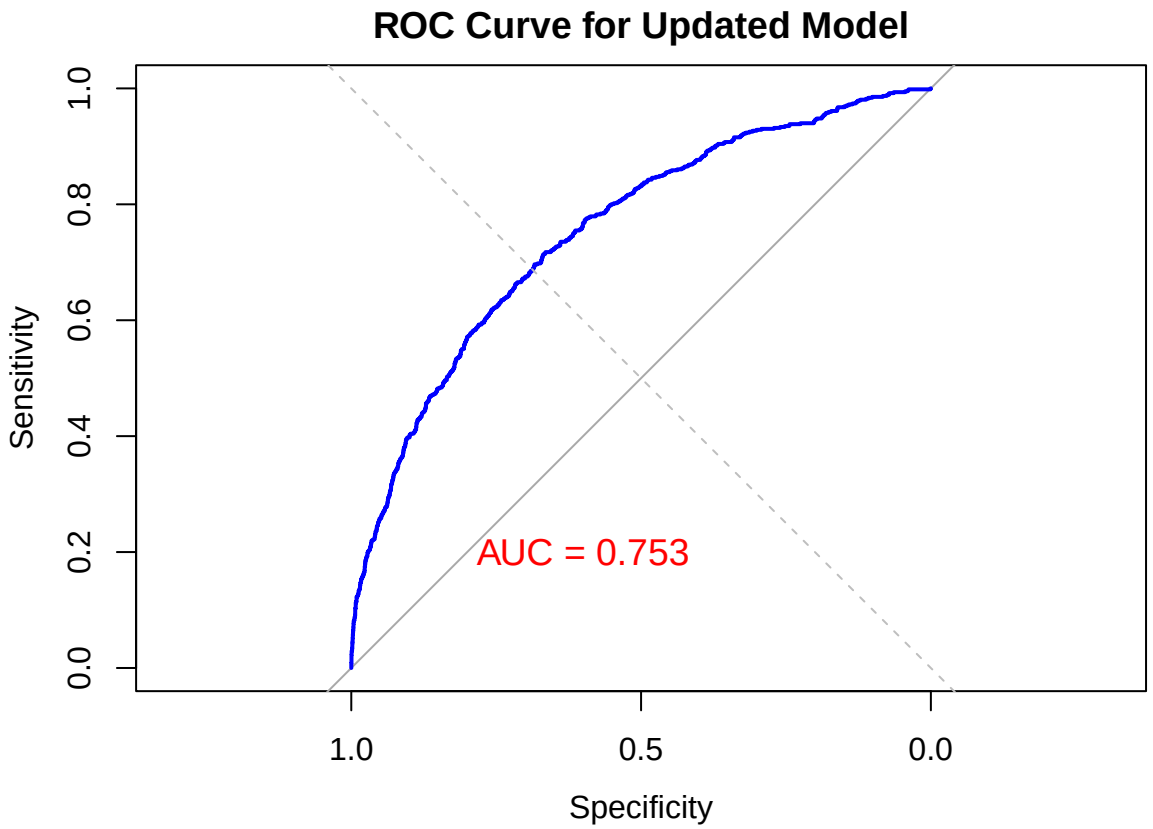
```
# Predict probabilities for the entire dataset
pred_probs_updated <- predict(final_model_updated, type = "response")

# Compute ROC curve
roc_curve <- roc(data_final$Status, pred_probs_updated)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
# Plot ROC curve
plot(roc_curve, col = "blue", lwd = 2, main = "ROC Curve for Updated Model")
abline(a = 0, b = 1, lty = 2, col = "gray")  # Add diagonal line (random guess)
```

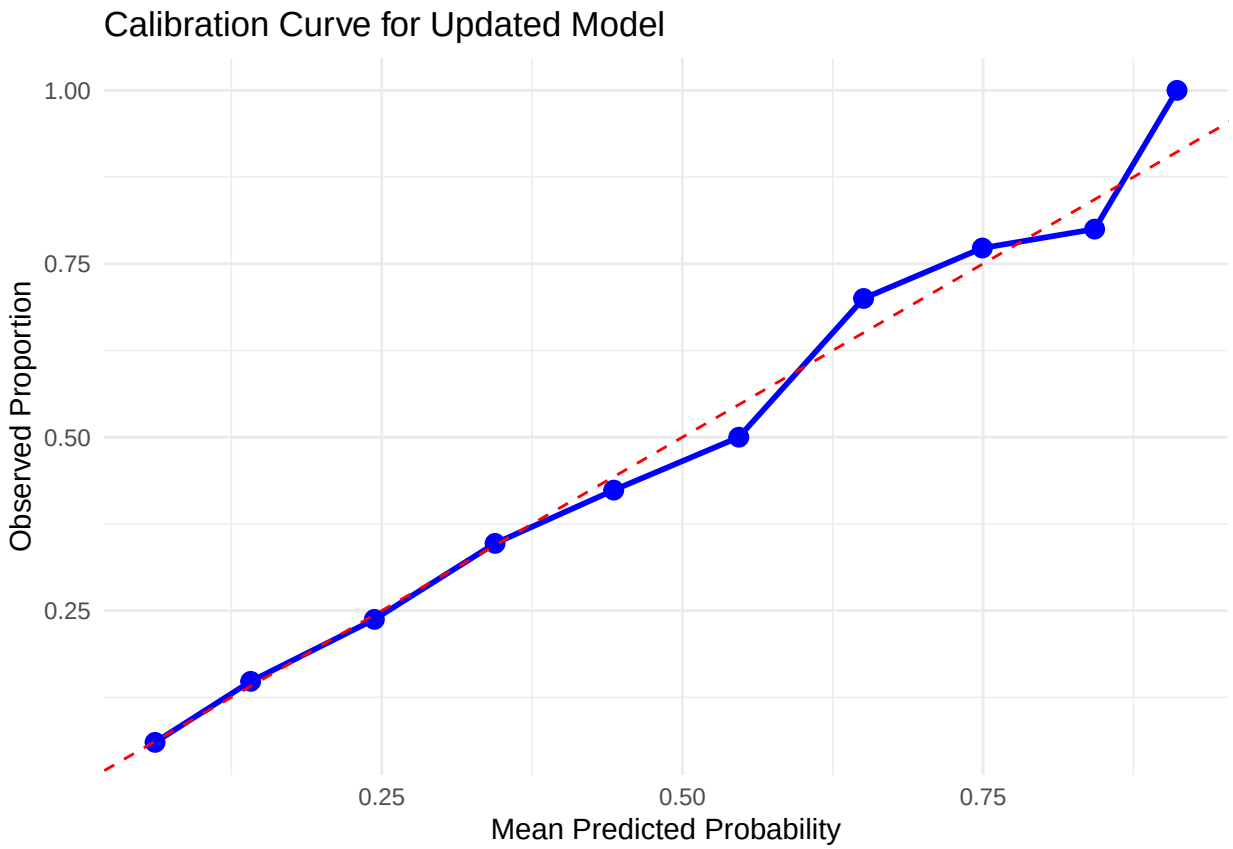
```r
# Add AUC value to the plot
auc_value <- auc(roc_curve)
text(0.6, 0.2, paste("AUC =", round(auc_value, 3)), col = "red", cex = 1.2)
```



On the entire dataset, ROC-AUC value is 0.753, indicating the model performance is acceptable but has room for improvement (0.7 - 0.8).

```r
# Create calibration data frame
calibration_df <- data.frame(
  Predicted = pred_probs_updated,
  Observed = as.numeric(data_final$Status)
)

# Group predicted probabilities into bins
calibration_df$Bin <- cut(calibration_df$Predicted, breaks = seq(0, 1, by = 0.1), include.lowest = TRUE)

# Calculate mean predicted probability and observed proportion for each bin
calibration_summary <- calibration_df %>%
  group_by(Bin) %>%
  summarise(
    Mean_Predicted = mean(Predicted),
    Mean_Observed = mean(Observed)
  )

# Plot calibration curve
ggplot(calibration_summary, aes(x = Mean_Predicted, y = Mean_Observed)) +
  geom_point(color = "blue", size = 3) +
  geom_line(color = "blue", lwd = 1) +
```

```
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Calibration Curve for Updated Model",
       x = "Mean Predicted Probability",
       y = "Observed Proportion") +
  theme_minimal()
```



The calibration curve demonstrates that the predicted probabilities align reasonably well with the observed proportions across most bins, as the points and blue line generally follow the diagonal red dashed line (perfect calibration) although some deviations exist. Overall, the model shows acceptable calibration.

Cross-validation provides a comprehensive method for model diagnosis by evaluating its performance across multiple data splits. This approach helps assess the model's generalization ability, reducing the risk of overfitting and ensuring robust performance on unseen data.

```
# Define 10-fold cross-validation
set.seed(123)  # For reproducibility
folds <- createFolds(y, k = 10, list = TRUE)

# Initialize a data frame to store results
cv_results <- data.frame(
  Fold = integer(),
  Accuracy = numeric(),
  Sensitivity = numeric(),
  Specificity = numeric(),
  ROC_AUC = numeric()
)

# Perform 10-fold cross-validation
```

```r
for (i in seq_along(folds)) {
  # Split data into training and testing sets
  train_indices <- unlist(folds[-i])  # Indices for training data
  test_indices <- unlist(folds[i])   # Indices for testing data

  train_data <- data_final[train_indices, ]
  test_data <- data_final[test_indices, ]

  # Refit the logistic regression model on the training set
  train_model <- glm(updated_formula, data = train_data, family = "binomial")

  # Predict probabilities on the testing set
  test_probs <- predict(train_model, newdata = test_data, type = "response")

  # Convert probabilities to binary predictions
  test_preds <- ifelse(test_probs >= threshold, 1, 0)

  # Generate the confusion matrix for the testing set
  fold_conf_matrix <- confusionMatrix(
    as.factor(test_preds),
    as.factor(test_data$Status),
    positive = "1"
  )

  # Calculate ROC-AUC
  roc_curve <- roc(as.numeric(test_data$Status), test_probs)
  roc_auc <- auc(roc_curve)

  # Store performance metrics for this fold
  cv_results <- rbind(cv_results, data.frame(
    Fold = i,
    Accuracy = fold_conf_matrix$overall["Accuracy"],
    Sensitivity = fold_conf_matrix$byClass["Sensitivity"],
    Specificity = fold_conf_matrix$byClass["Specificity"],
    ROC_AUC = as.numeric(roc_auc)
  ))
}
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```r
# Summarize cross-validation results
cv_summary <- data.frame(
  Metric = c("Accuracy", "Sensitivity", "Specificity", "ROC_AUC"),
  Mean = colMeans(cv_results[, -1], na.rm = TRUE),
  SD = apply(cv_results[, -1], 2, sd, na.rm = TRUE)
)

# Print the summary of cross-validation results
print("Cross-Validation Results:")
```

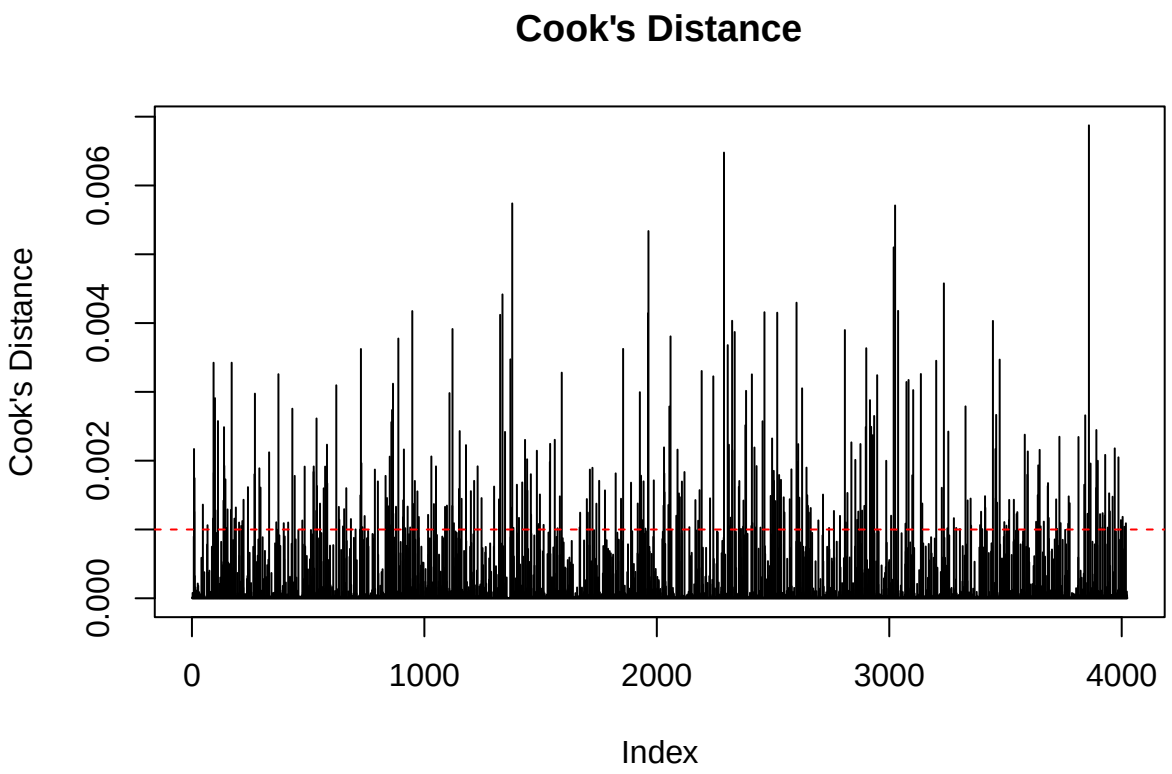
```
## [1] "Cross-Validation Results:"
```

```r
print(cv_summary)
```

```
##                   Metric      Mean          SD
## Accuracy        Accuracy 0.8531295 0.010397492
## Sensitivity Sensitivity 0.1334452 0.031295877
## Specificity Specificity 0.9833262 0.006557589
## ROC_AUC         ROC_AUC 0.7451556 0.023696947
```

The average accuracy is 85.29% (SD = 1.04%), indicating that the model performs well overall in classifying the observations correctly. The specificity is very high, averaging 98.36% (SD = 0.64%), which demonstrates the model's strong ability to correctly identify negative cases (Alive). However, the sensitivity is relatively low at 13.00% (SD = 3.01%), reflecting a limited capability to detect positive cases (Alive). The ROC-AUC is 0.744 (SD = 0.024), suggesting the model has acceptable discrimination ability but room for improvement.

```r
# Calculate Cook's Distance
cooks_d <- cooks.distance(final_model_updated)

# Plot Cook's Distance
plot(cooks_d, type = "h", main = "Cook's Distance", ylab = "Cook's Distance")
abline(h = 4/(nrow(data_final) - length(final_model_updated$coefficients) - 1), col = "red", lty = 2)
```




```
# Identify influential points
influential_threshold <- 4/(nrow(data_final) - length(final_model_updated$coefficients) - 1)
influential_points <- which(cooks_d > influential_threshold)
print(paste("Number of influential points:", length(influential_points)))
```

```
## [1] "Number of influential points: 332"
```

```
# Inspect influential observations
influential_data <- data_final[influential_points, ]
kable(head(influential_data, 10), caption = "Influential Observations")
```

Table 7: Influential Observations

	Age	Race																										
9	40	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	1	1	1	1	20	0	2.944	4.391
10	69	0	0	1	1	0	0	0	0	0	1	0	1	0	0	0	1	1	0	0	0	1	1	21	0	2.564	9.496
11	68	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	9	1	0.693	2.079
47	42	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	1	0	0	15	1	1.098	3.322
67	31	0	0	1	1	0	0	0	0	1	0	0	1	0	0	0	1	0	0	1	1	1	1	23	1	3.178	0.268
91	53	0	0	1	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	1	1	1	8	1	0.693	1.772
93	63	0	1	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	1	1	5	1	1.609	4.205
94	45	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	1	1	1	1	21	1	2.833	3.258
99	50	0	0	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	1	1	1	3	1	1.386	2.944
101	38	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	1	0	1	0	1	1	1	33	1	3.218	4.268

The red dashed line represents the commonly used threshold for identifying influential points. The majority of observations fall below the threshold, suggesting that they contribute reasonably and do not overly influence the model.