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")
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(), .groups = "drop") %>%
  mutate(Percentage = round((Count / sum(Count)) * 100, 1)) %>%
```

```
  arrange(Variable, desc(Count))

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	8.9
	Distant	92	0.2
Estrogen.Status	Positive	3755	8.5
	Negative	269	0.6
Grade	2	2351	5.3
	3	1111	2.5
	1	543	1.2
	anaplastic; Grade IV	19	0.0
Marital.Status	Married	2643	6.0
	Single	615	1.4
	Divorced	486	1.1
	Widowed	235	0.5
	Separated	45	0.1
N.Stage	N1	2732	6.2
	N2	820	1.9
	N3	472	1.1
Progesterone.Status	Positive	3326	7.5
	Negative	698	1.6
Race	White	3413	7.7
	Other	320	0.7
	Black	291	0.7
Status	Alive	3408	7.7
	Dead	616	1.4
T.Stage	T2	1786	4.0
	T1	1603	3.6
	T3	533	1.2
	T4	102	0.2
X6th.Stage	IIA	1305	2.9
	IIB	1130	2.6
	IIIA	1050	2.4
	IIIC	472	1.1
	IIIB	67	0.2
differentiate	Moderately differentiated	2351	5.3
	Poorly differentiated	1111	2.5
	Well differentiated	543	1.2
	Undifferentiated	19	0.0

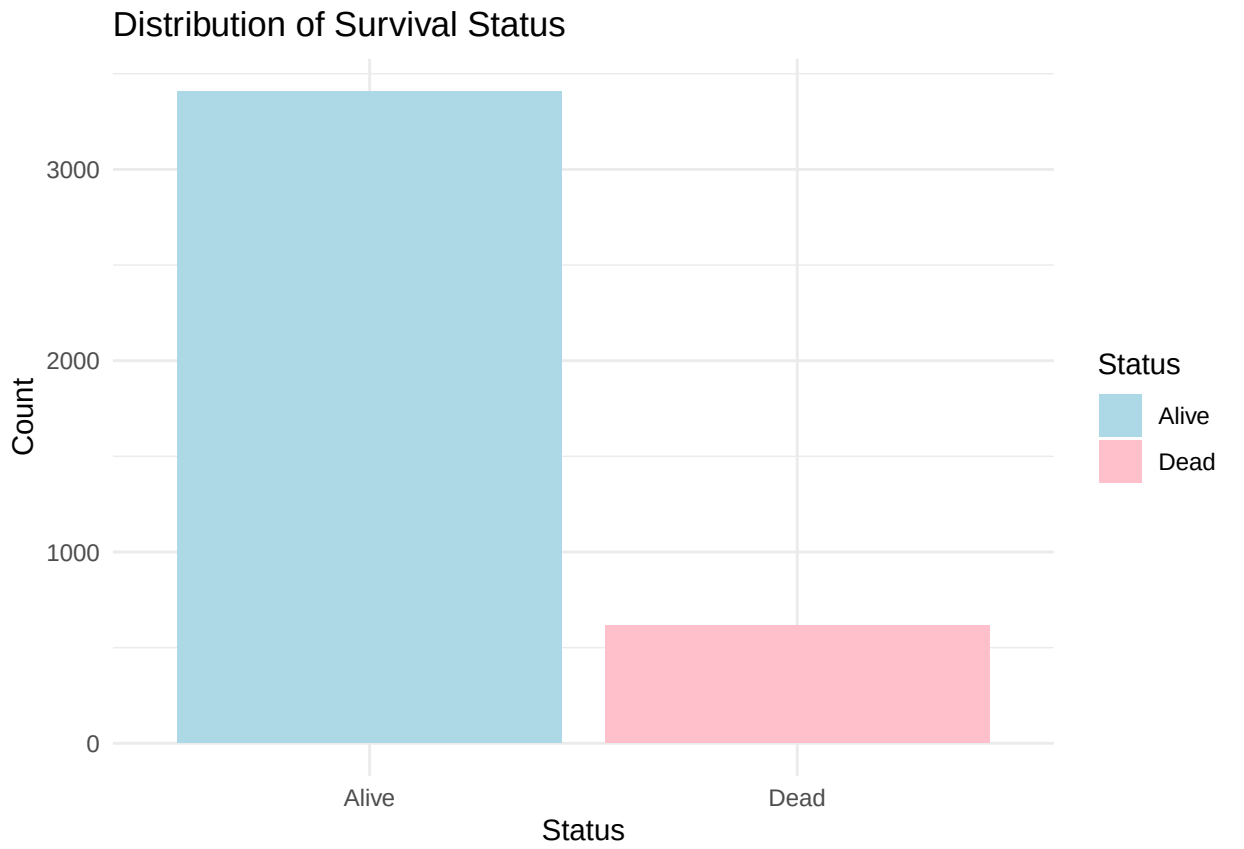
Explore the Distribution of the Outcome (Status: Dead / Alive)

```
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)
```

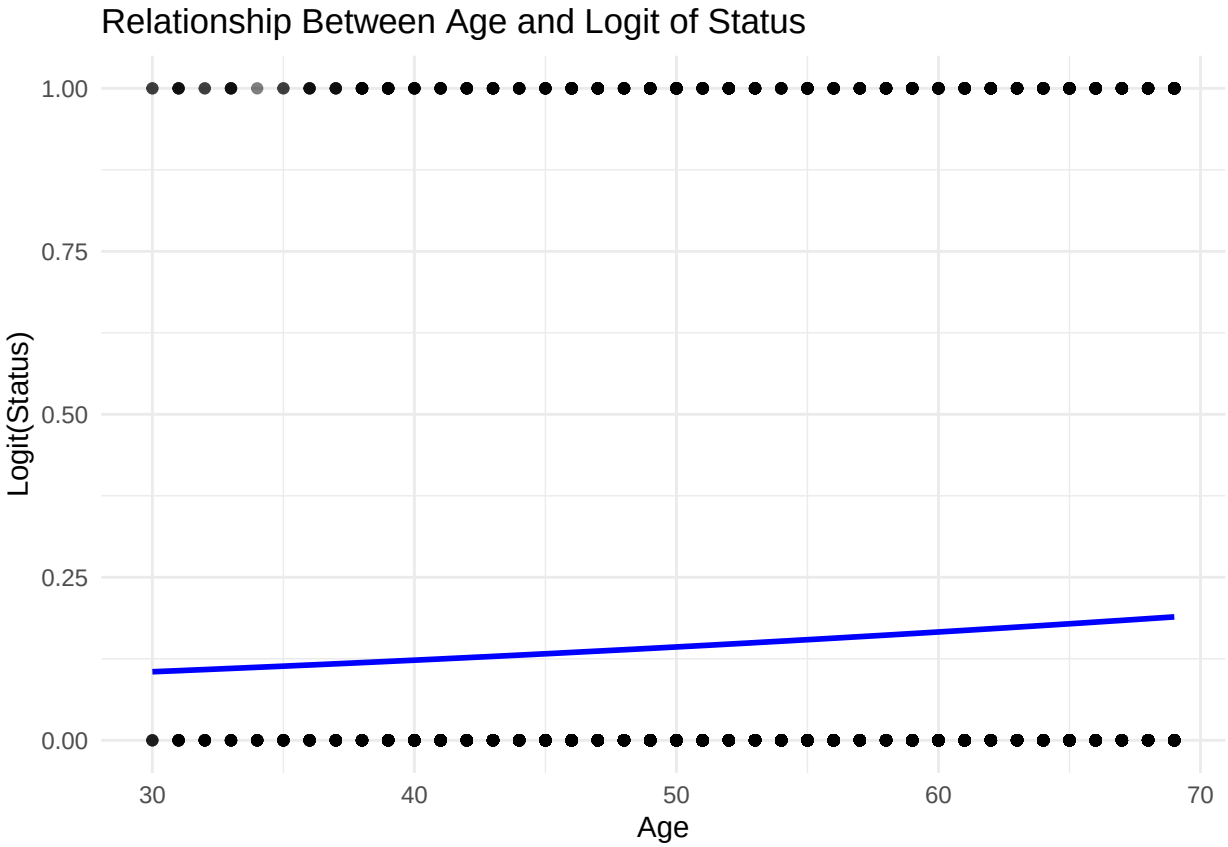
```
## [1] "Age"                    "Tumor.Size"             "Regional.Node.Examined"
## [4] "Reginol.Node.Positive"
```

```
# 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_string(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", ., "and Logit of Status"), x = ., y = "Logit(Status)
        theme_minimal())
```

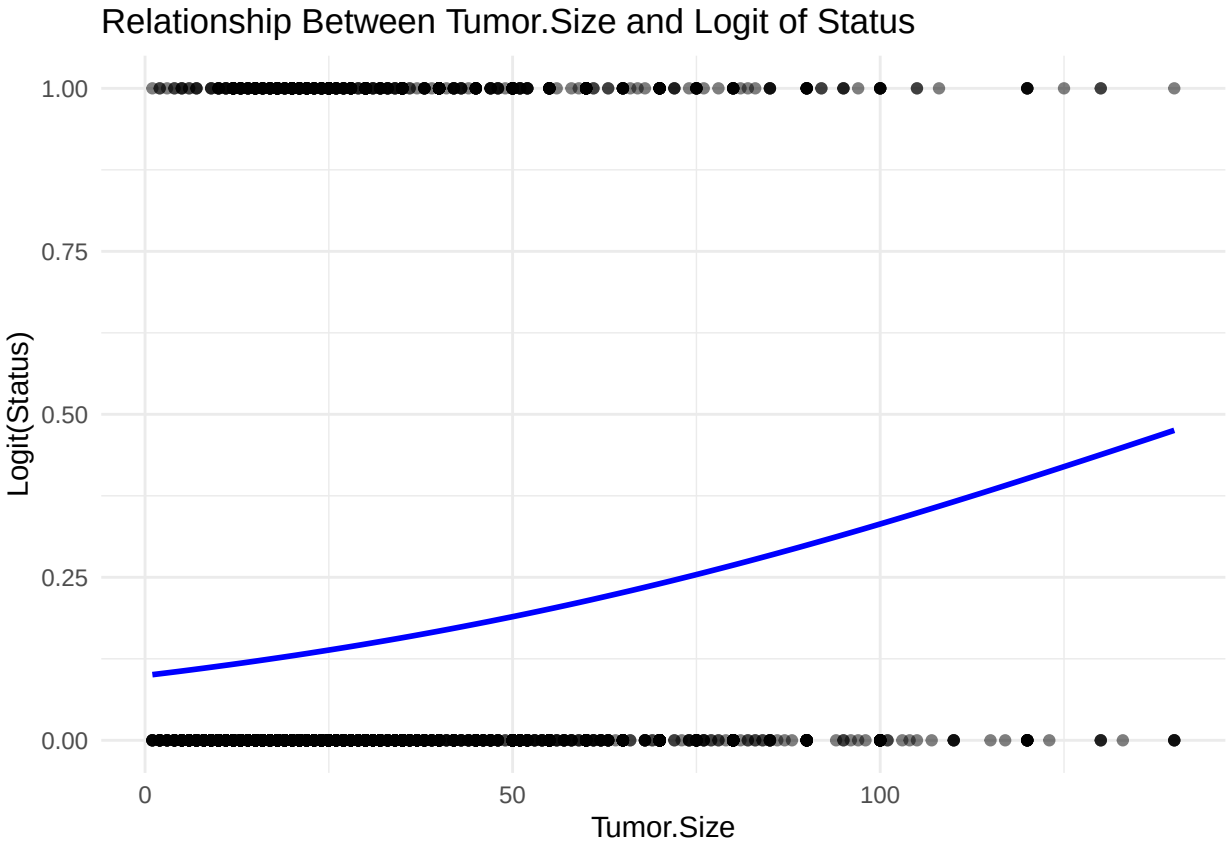
```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

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

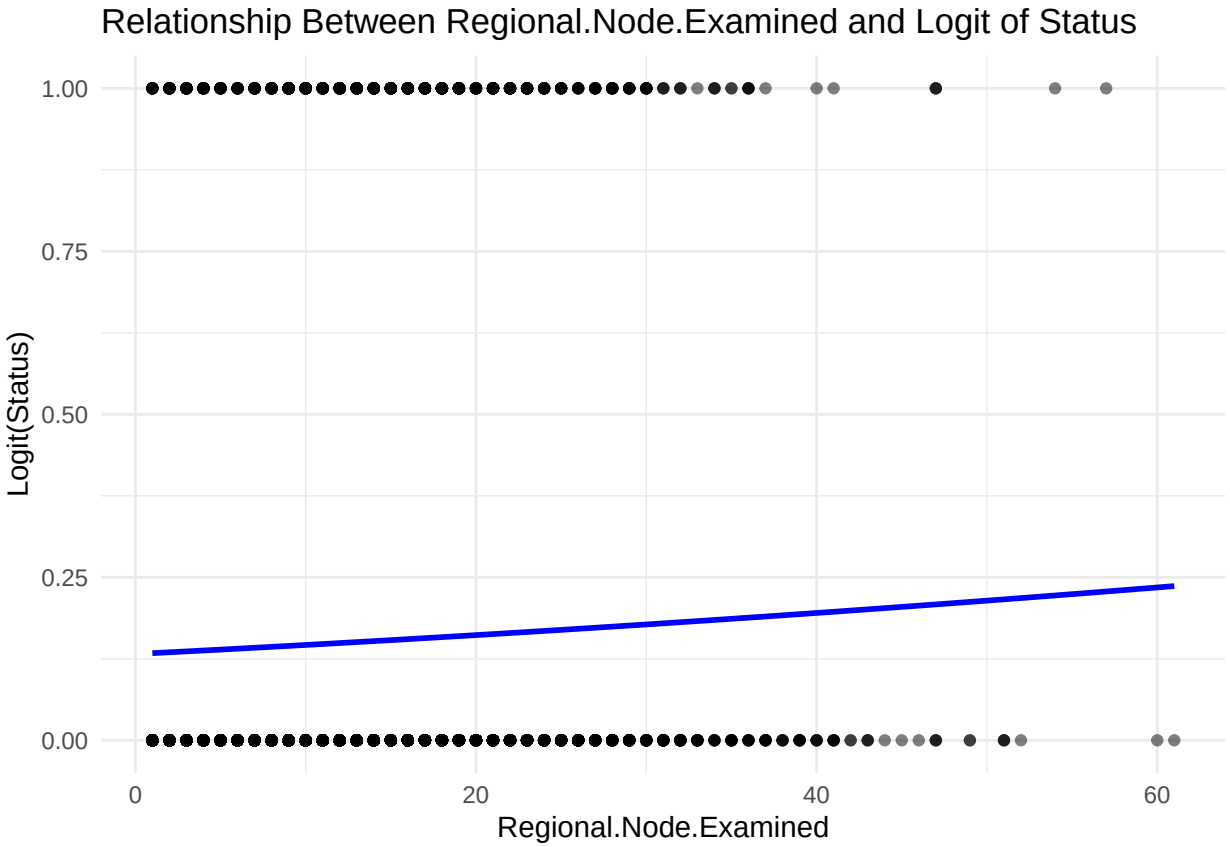
```
## `geom_smooth()` using formula = 'y ~ x'
```



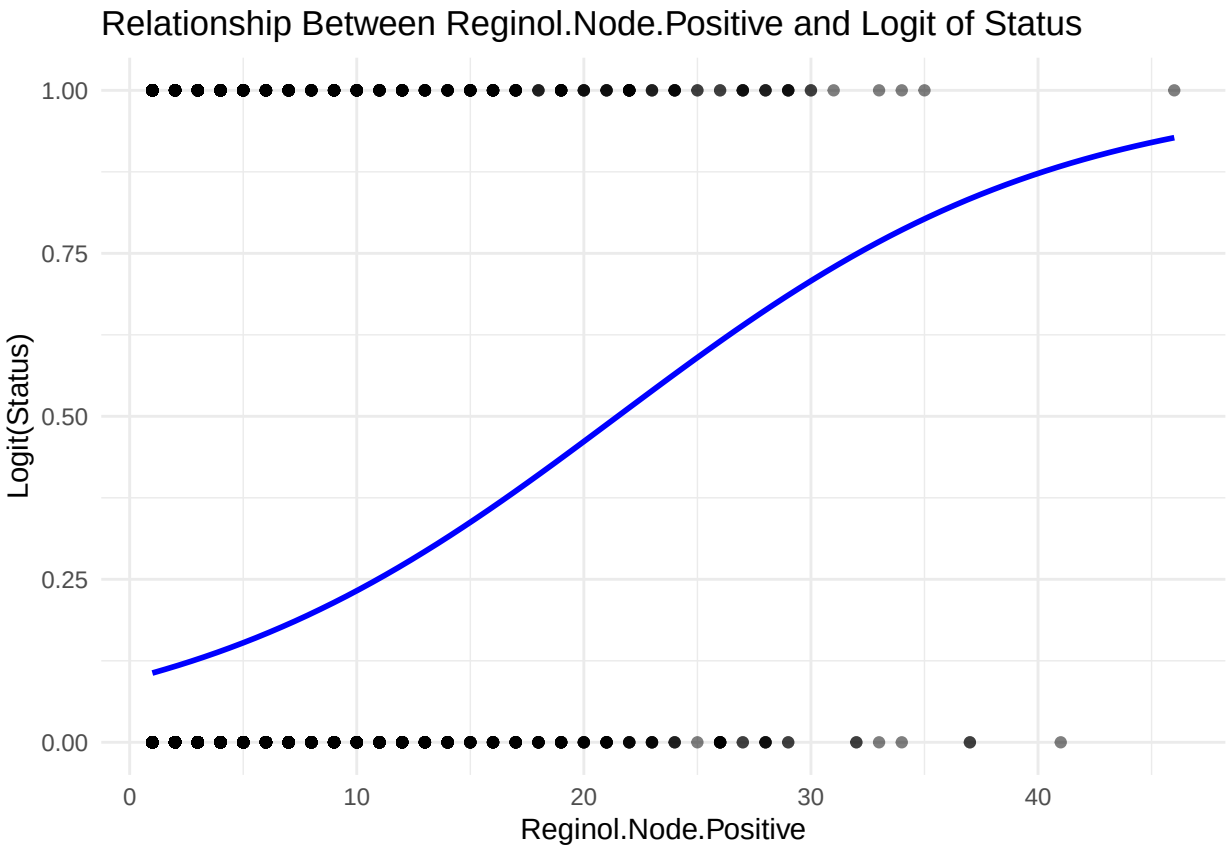

```
##
## [[2]]

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




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

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




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

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




```r
# 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.

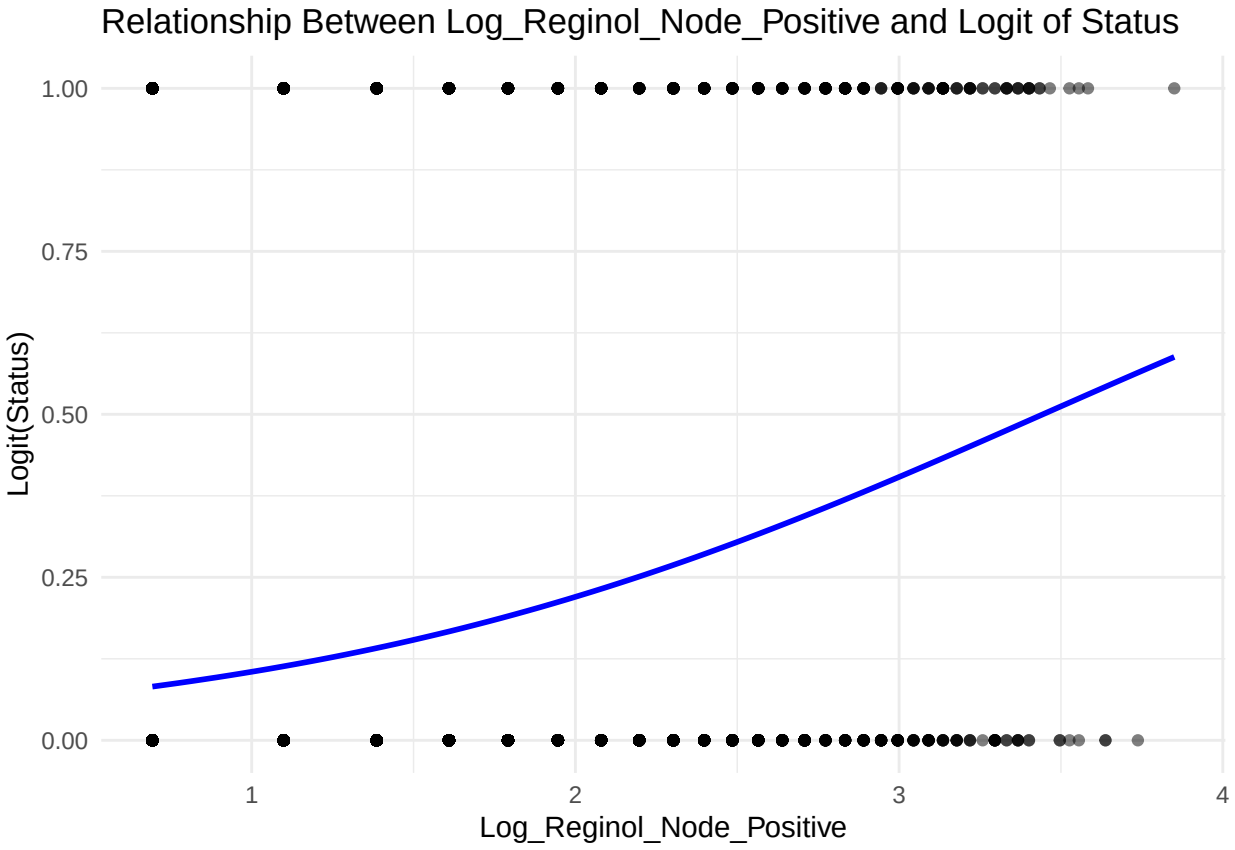
```
data <- data %>%
  mutate(
    Log_Reginol_Node_Positive = log1p(`Reginol.Node.Positive`),
    Log_Tumor_Size = log1p(`Tumor.Size`)
  )
transformed_skewness <- data %>%
  select(Log_Reginol_Node_Positive, Log_Tumor_Size) %>%
  summarise_all(~ skewness(.))

# 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

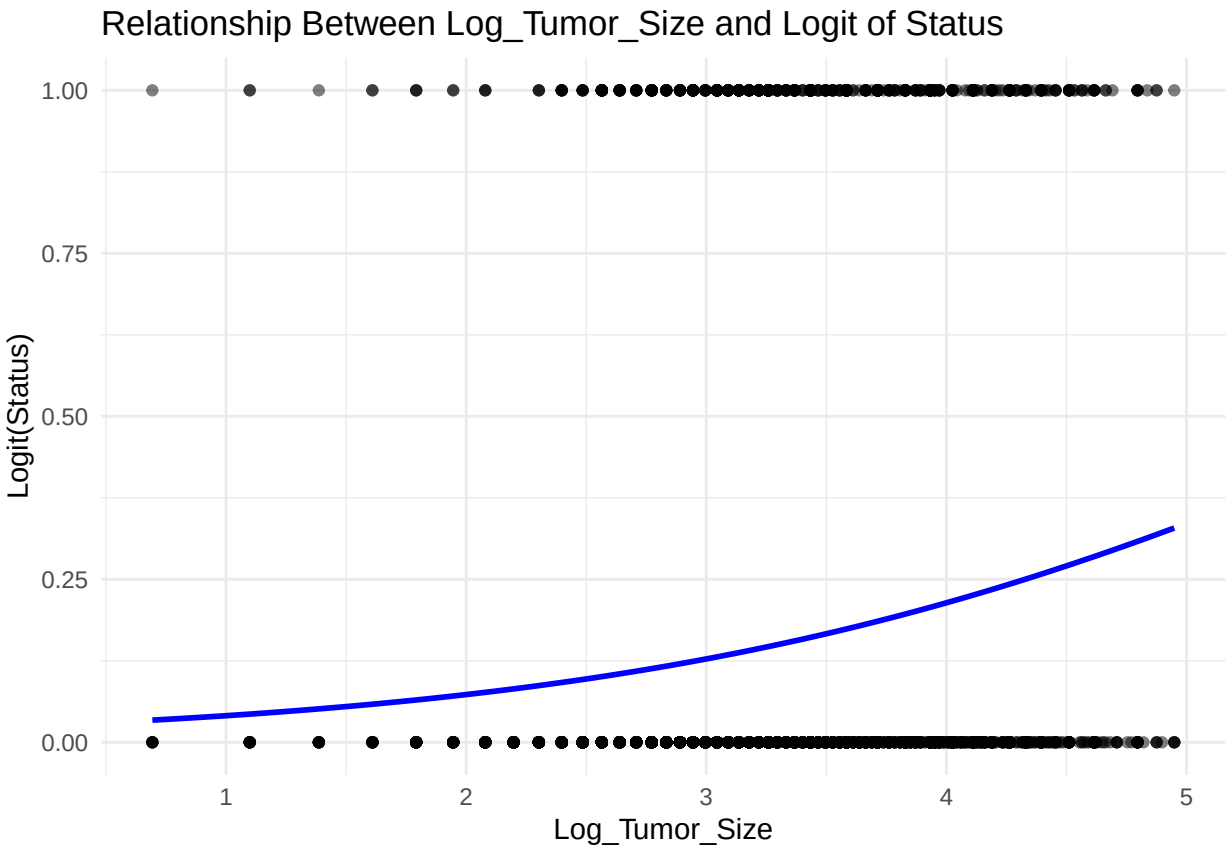
```
## 2. Plots for Transformed Variables Against Logit of Status
logit <- function(p) log(p / (1 - p))  # Logit function

# 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
- For Log_Tumor_Size, the curvature is still present after the transformation, and the linearity with the logit has not significantly improved.

As a result, we should definitely conduct a log transformation on Log_Reginol_Node_Positive, we are not sure on Log_Tumor_Size, we can evaluate it in model selection.

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

```
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()
```