BM_final

2024-12-16

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(data.table)
```

```
##
## Attaching package: 'data.table'
##
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
##
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
##
## The following object is masked from 'package:purrr':
##
##     transpose
```

load data

```r
data = read.csv("Project_2_data.csv")
```

preprocess

```r
colnames(data)
```

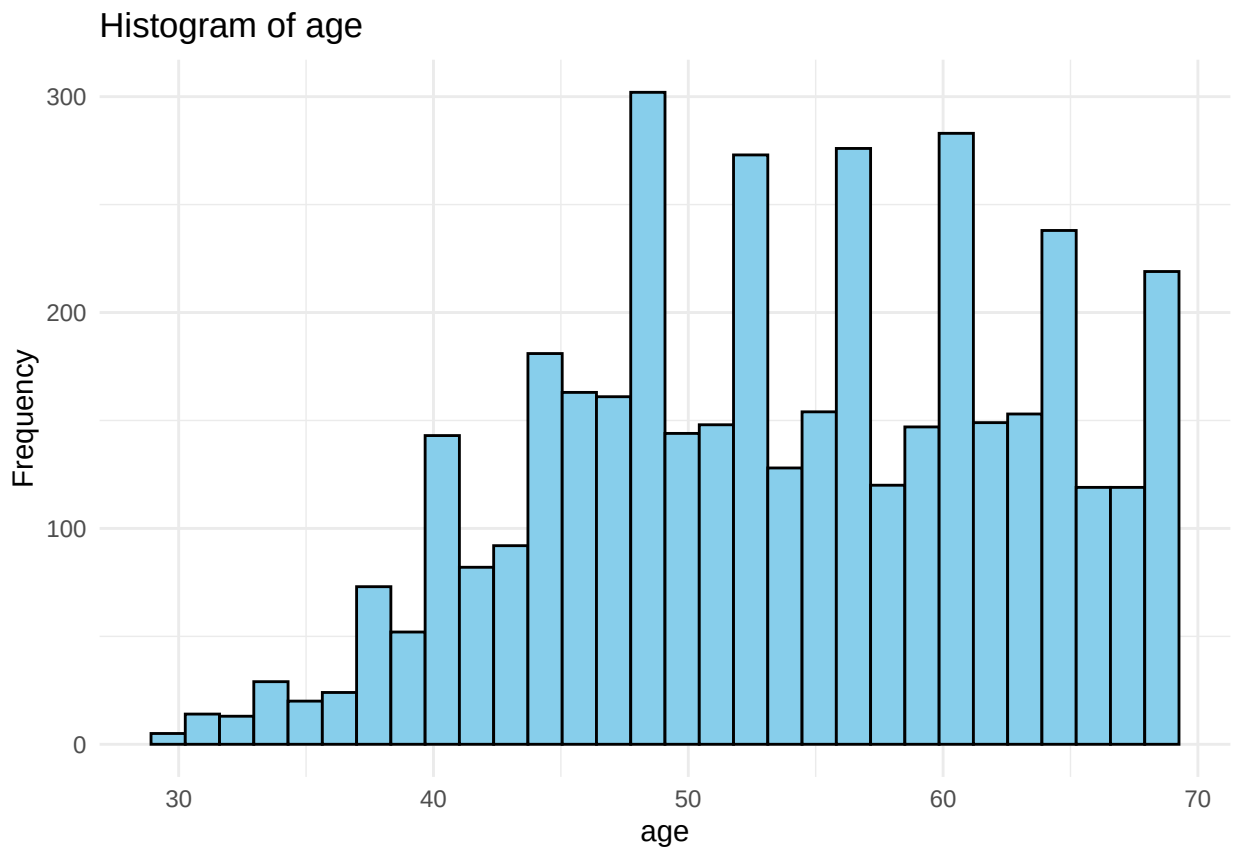
```
##  [1] "Age"                    "Race"                   "Marital.Status"
##  [4] "T.Stage"                "N.Stage"                "X6th.Stage"
##  [7] "differentiate"          "Grade"                  "A.Stage"
## [10] "Tumor.Size"             "Estrogen.Status"        "Progesterone.Status"
## [13] "Regional.Node.Examined" "Reginol.Node.Positive"  "Survival.Months"
## [16] "Status"
```

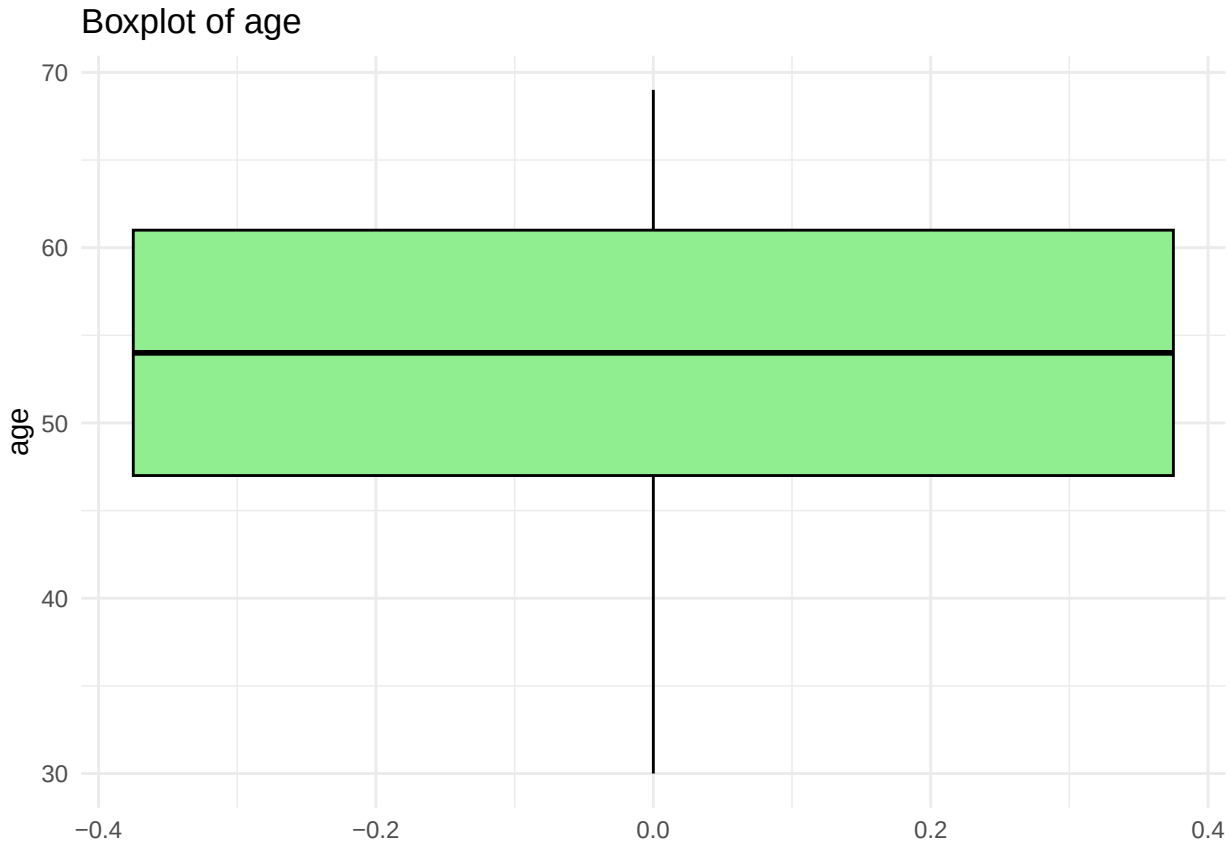
```r
data <- data %>%
  drop_na()%>%
  mutate(
    Race = if_else(Race != "White", "Not White", "White")) %>%
  mutate(Main.Stage = case_when(
    X6th.Stage %in% c("IIIA", "IIIC") ~ "III",
    X6th.Stage %in% c("IIA", "IIB") ~ "II",
    X6th.Stage %in% c("IVA", "IVB") ~ "IV",
    TRUE ~ X6th.Stage
  ))%>%
  mutate(
    T.Stage = as.numeric(gsub("T", "", T.Stage)),
    N.Stage = as.numeric(gsub("N", "", N.Stage))
  )%>%
  mutate(
    Race = as.factor(Race),
    Marital.Status = as.factor(Marital.Status),
    Estrogen.Status = as.factor(Estrogen.Status),
    Progesterone.Status = as.factor(Progesterone.Status),
    differentiate = as.factor(differentiate),
    A.Stage = as.factor(A.Stage),
    Status = as.factor(Status),
    Main.Stage = as.factor(Main.Stage)
  )%>%
  select(-X6th.Stage)%>%
  mutate(
    Grade = as.numeric(gsub("[^0-9]", "", Grade))
  )%>%
  rename(Regional.Node.Positive = "Reginol.Node.Positive")%>%
  janitor::clean_names()
```

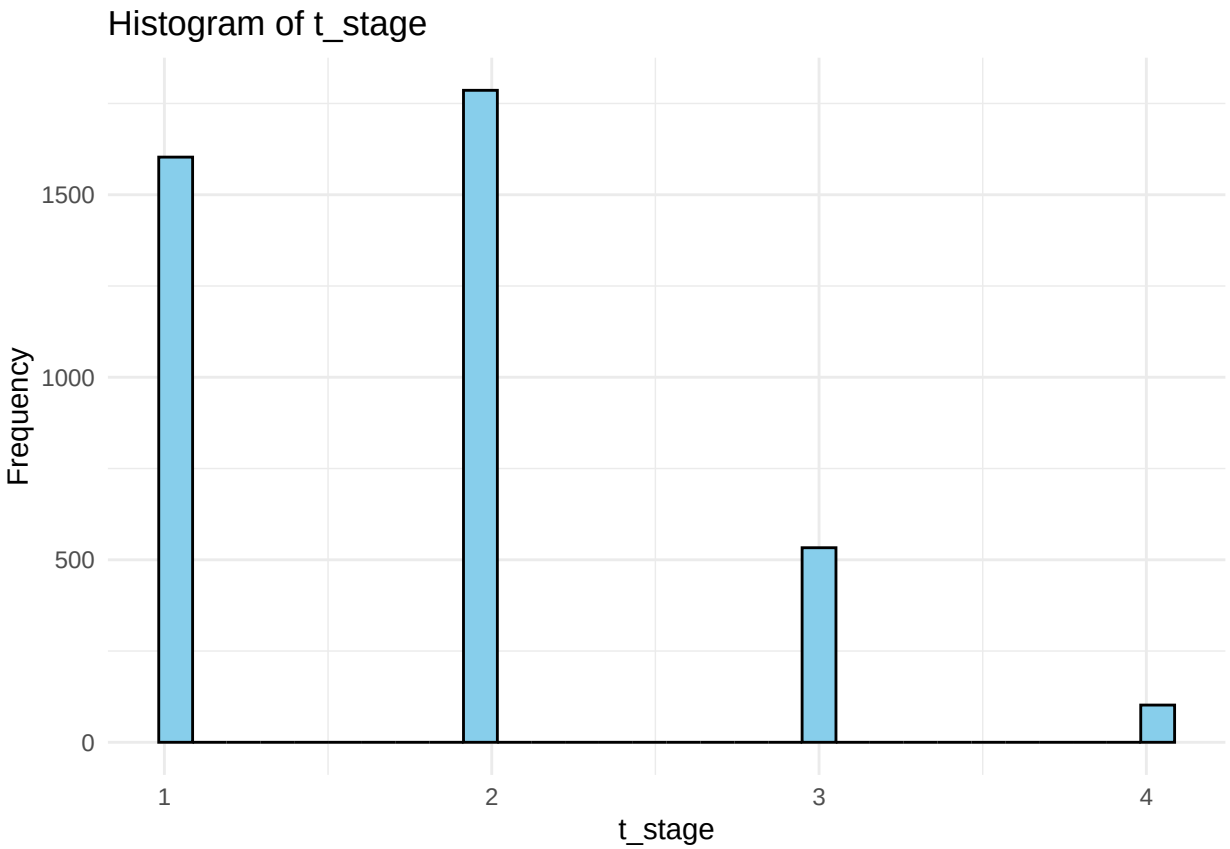
see the plot after first preprocess

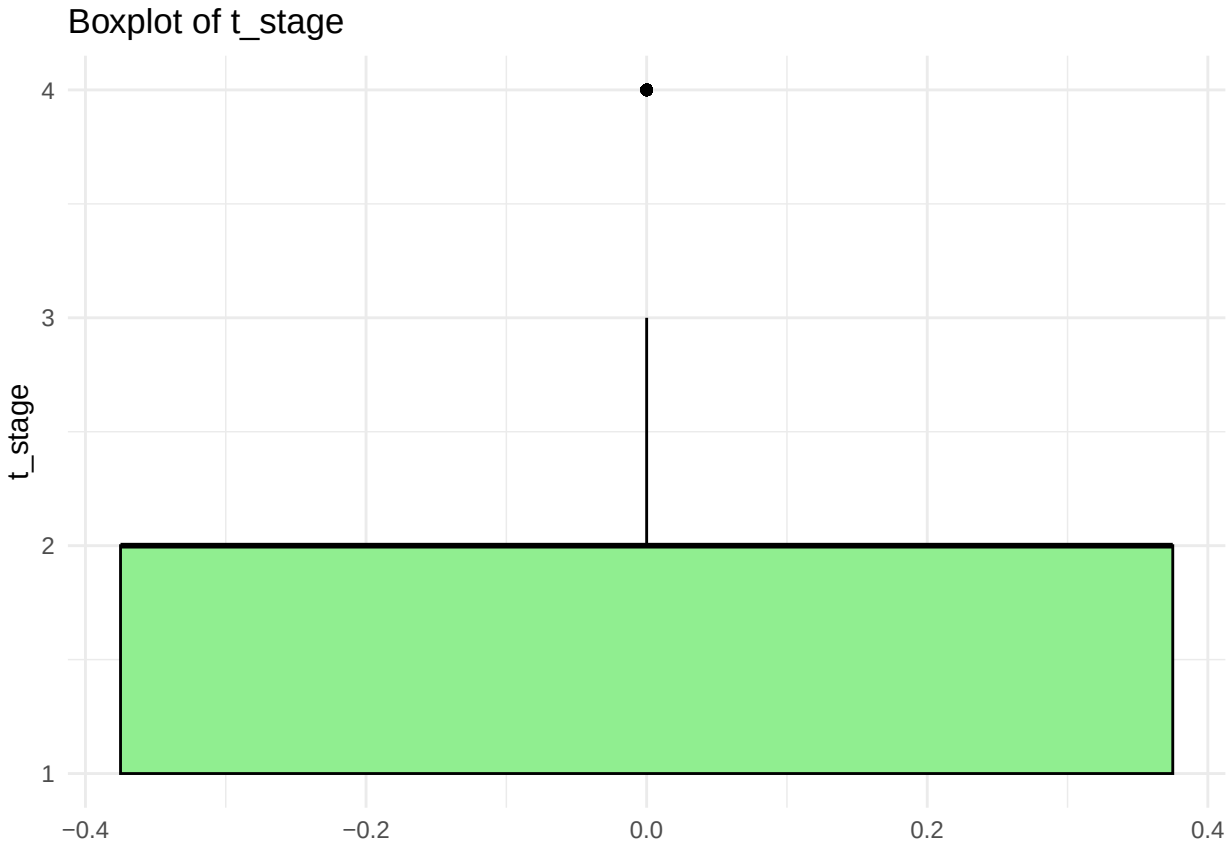
```r
for (var in names(data)) {
  # Skip if the column is not numeric
  if (is.numeric(data[[var]])) {

    # Histogram
    p1 <- ggplot(data, aes_string(x = var)) +
      geom_histogram(fill = "skyblue", color = "black", bins = 30) +
      labs(title = paste("Histogram of", var), x = var, y = "Frequency") +
      theme_minimal()
    print(p1)

    # Boxplot
    p2 <- ggplot(data, aes_string(y = var)) +
      geom_boxplot(fill = "lightgreen", color = "black") +
      labs(title = paste("Boxplot of", var), y = var) +
      theme_minimal()
    print(p2)
  }
}
```

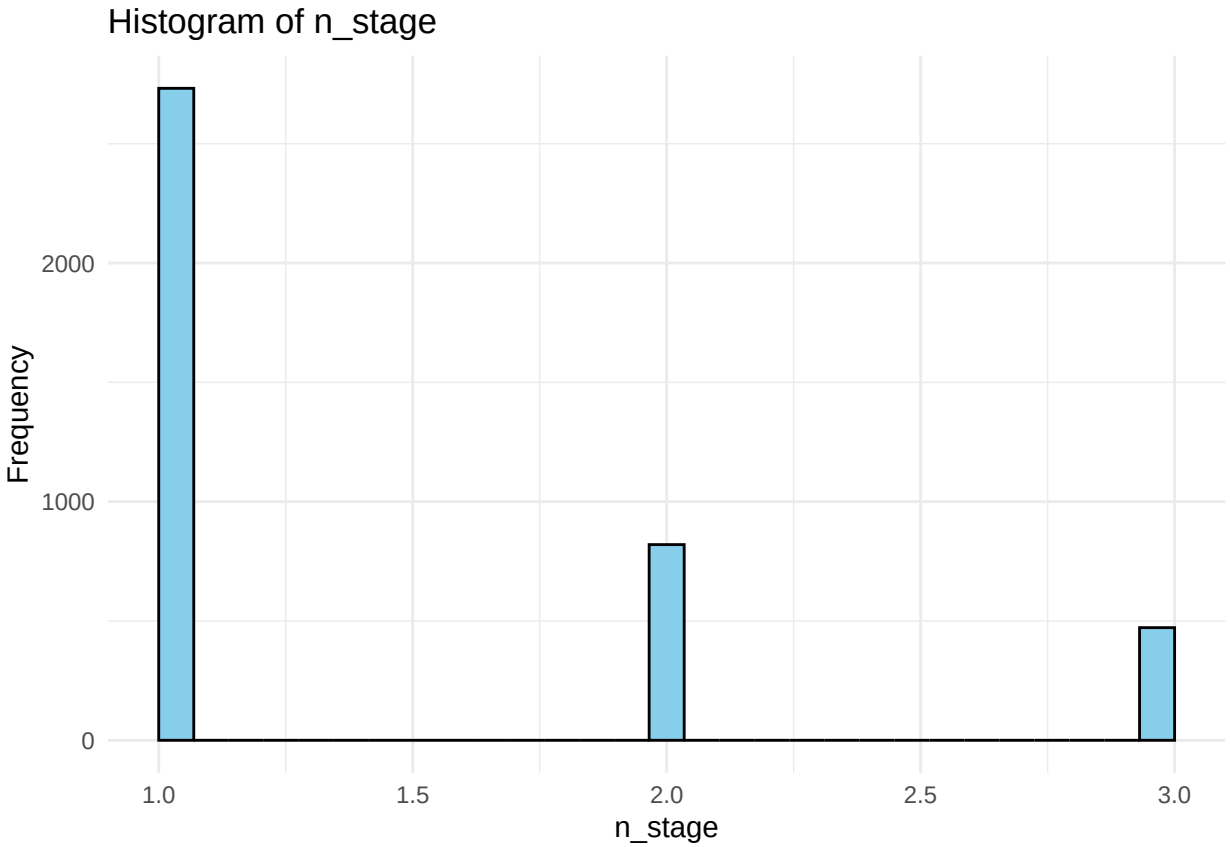
```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this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

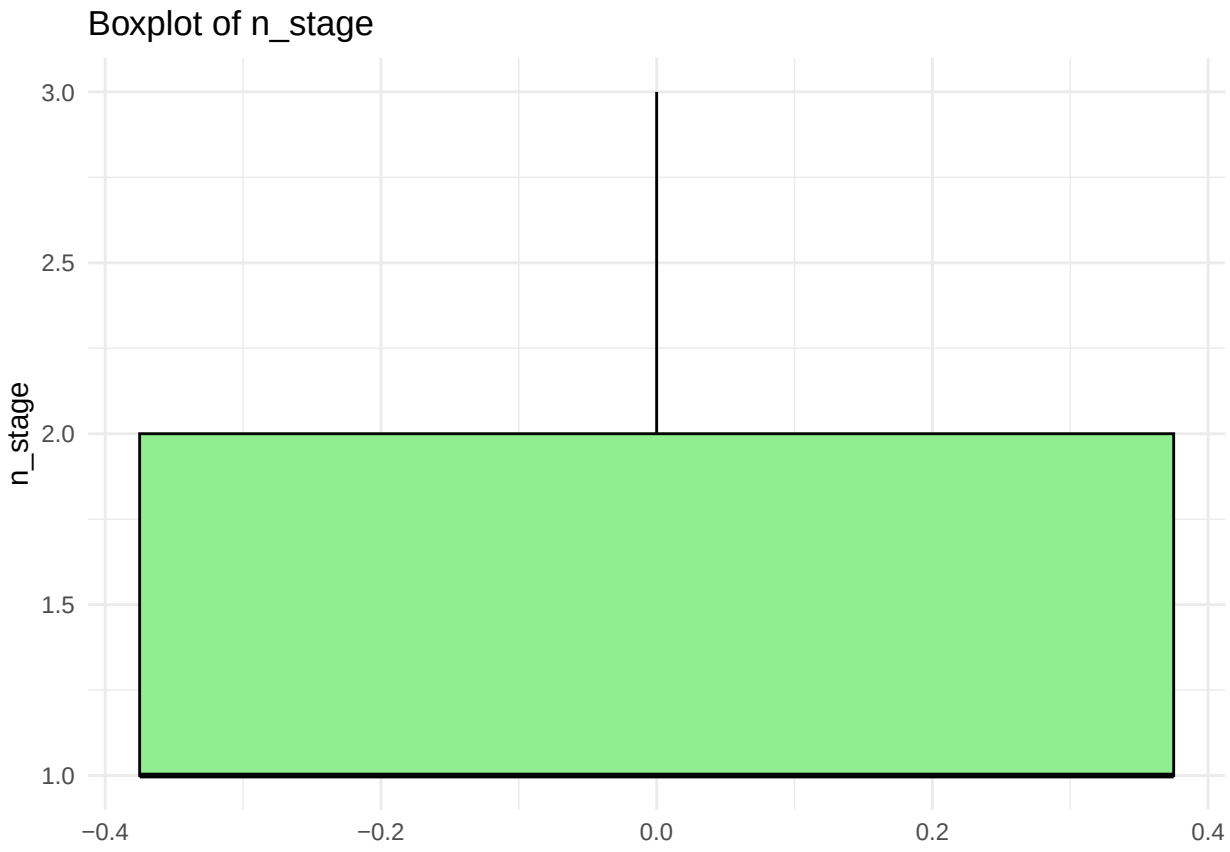

Histogram of age
Frequency
300
200
100
0
30
40
50
60
70
age


Boxplot of age
age
70
60
50
40
30
−0.4
−0.2
0.0
0.2
0.4

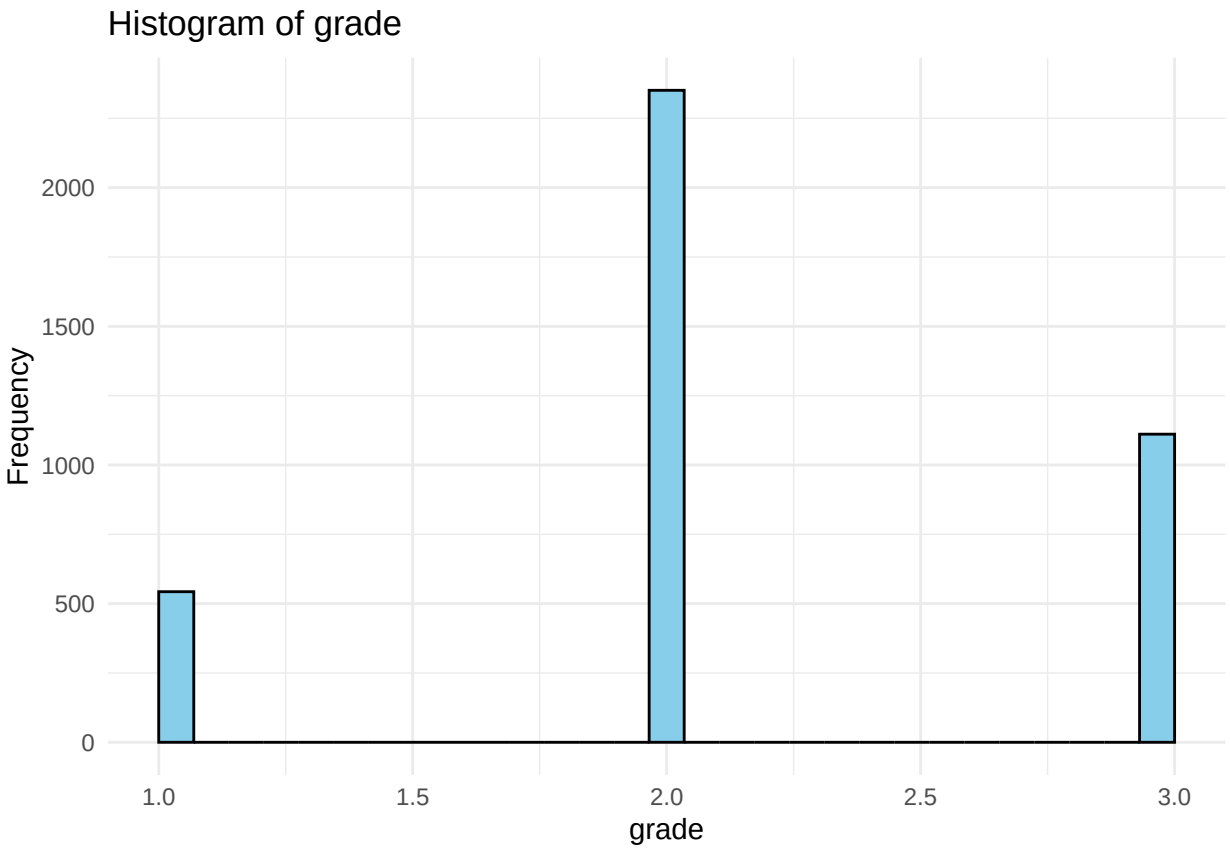

Histogram of t_stage
Frequency
1500
1000
500
0
1
2
3
4
t_stage


Boxplot of t_stage
t_stage
4
3
2
1
−0.4
−0.2
0.0
0.2
0.4

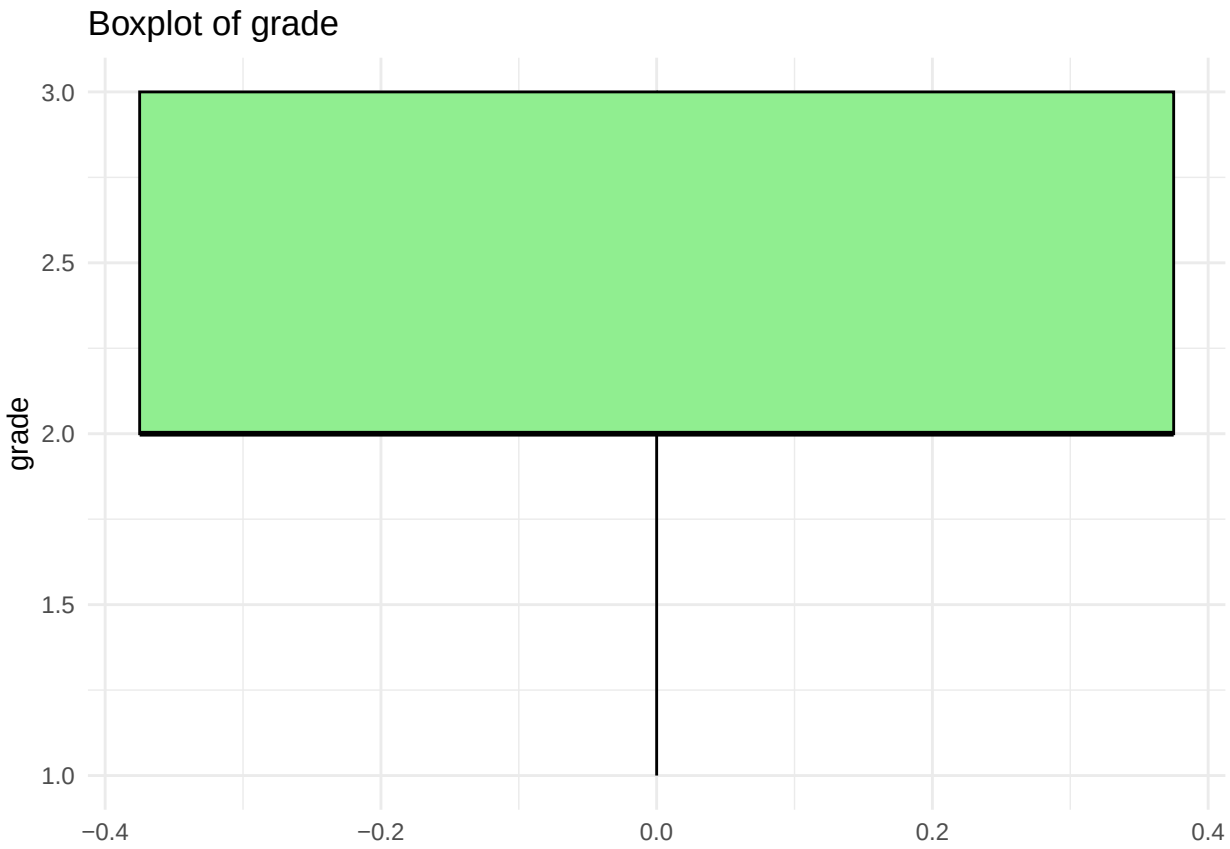

Histogram of n_stage
Frequency
2000
1000
0
1.0
1.5
2.0
2.5
3.0
n_stage

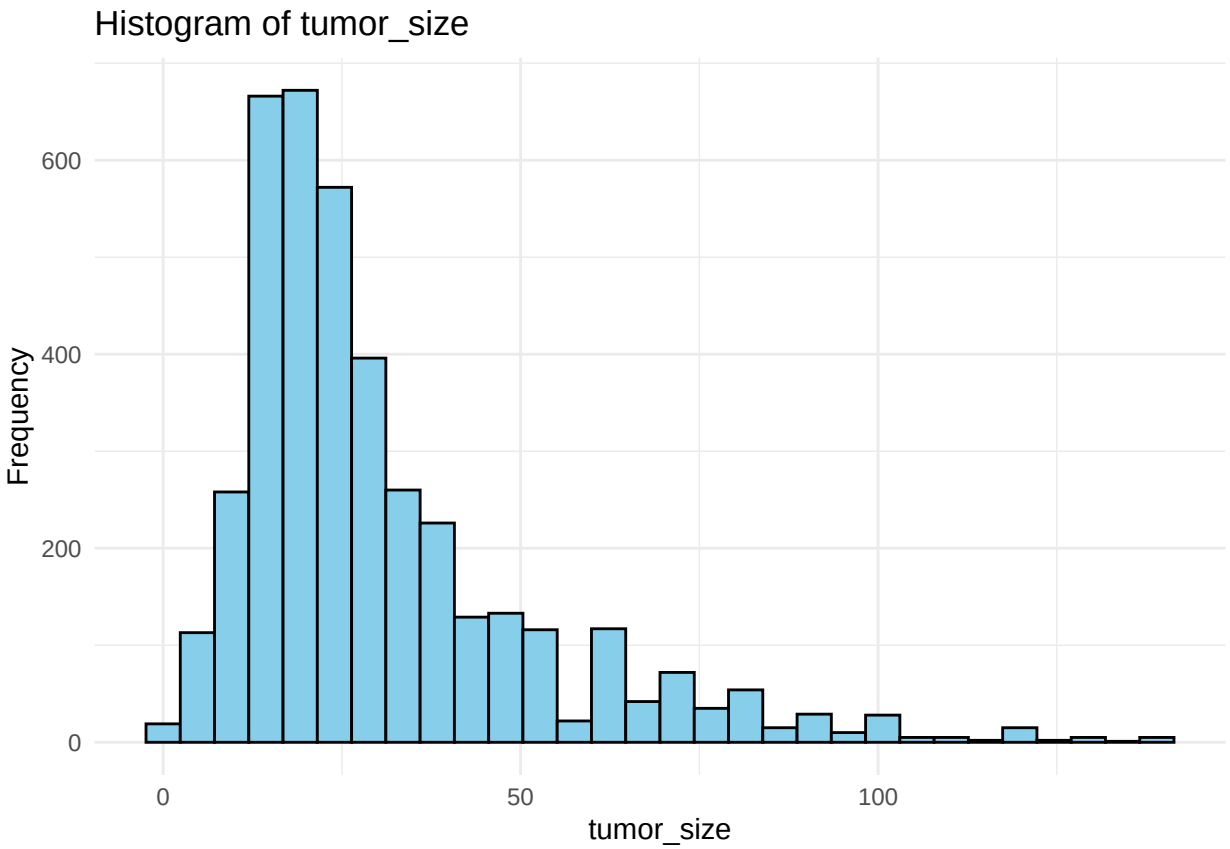


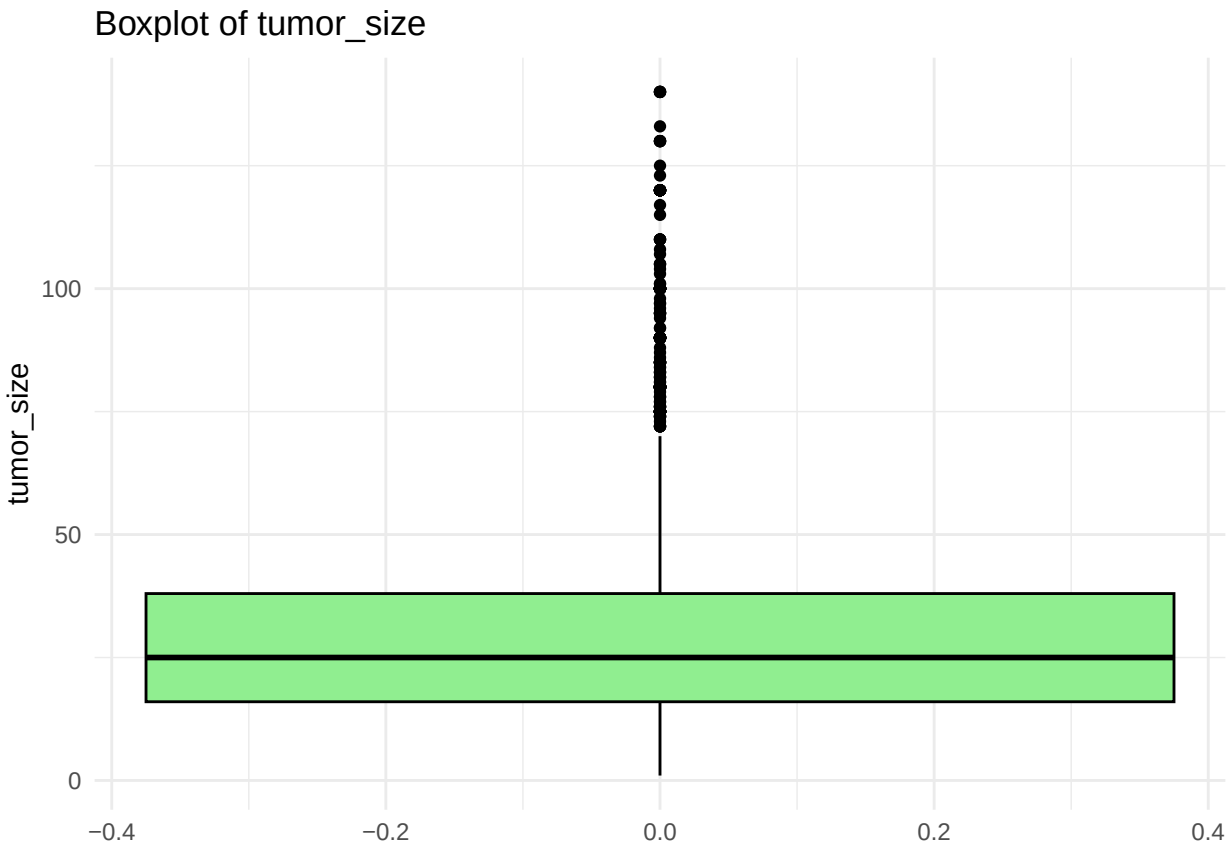

```
## Warning: Removed 19 rows containing non-finite outside the scale range
## (‘stat_bin()‘).
```

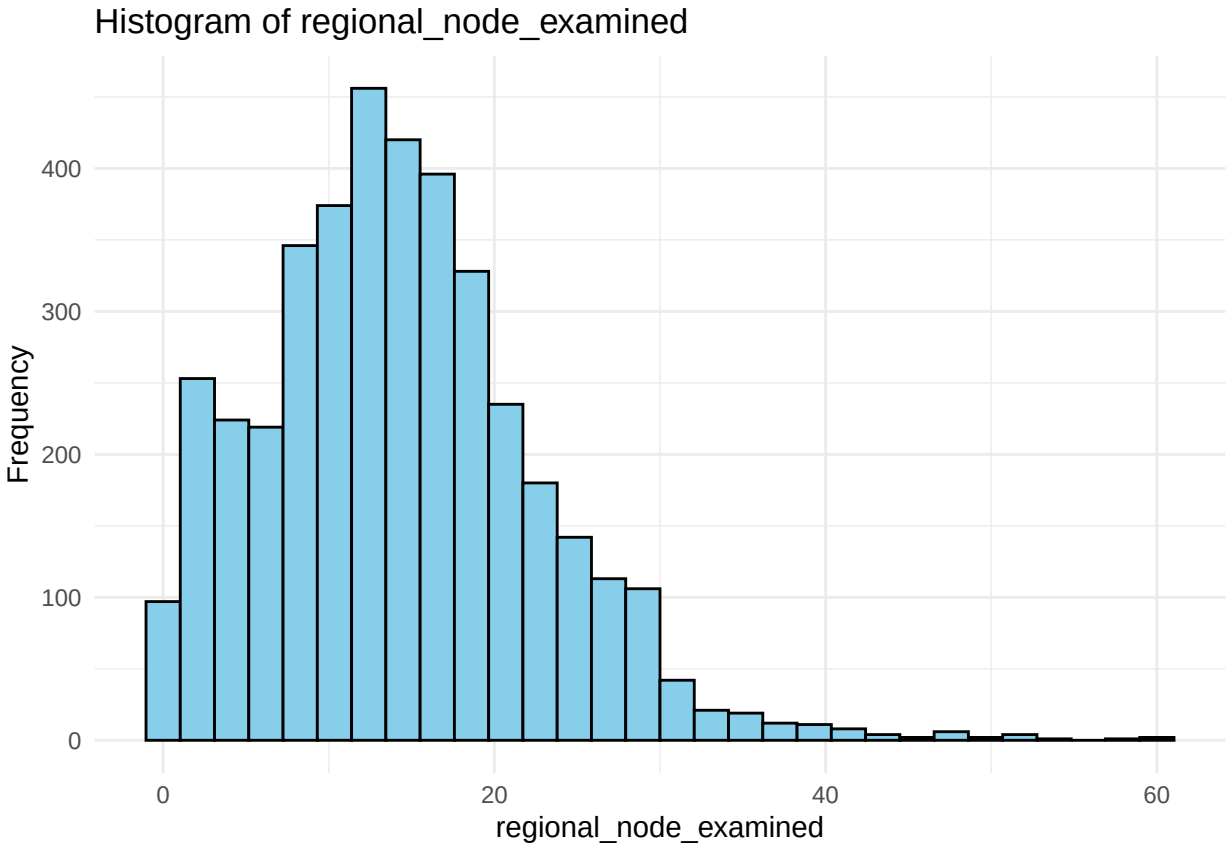


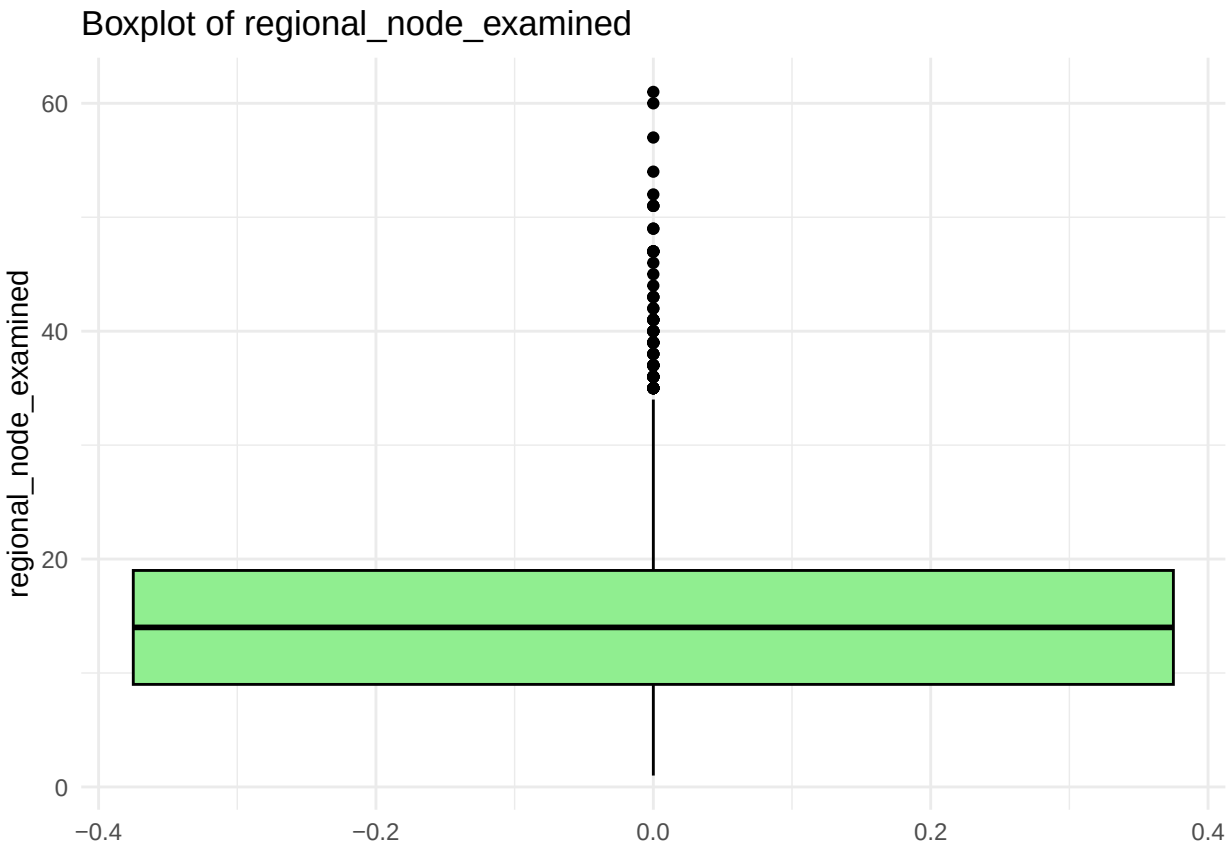

```
## Warning: Removed 19 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```

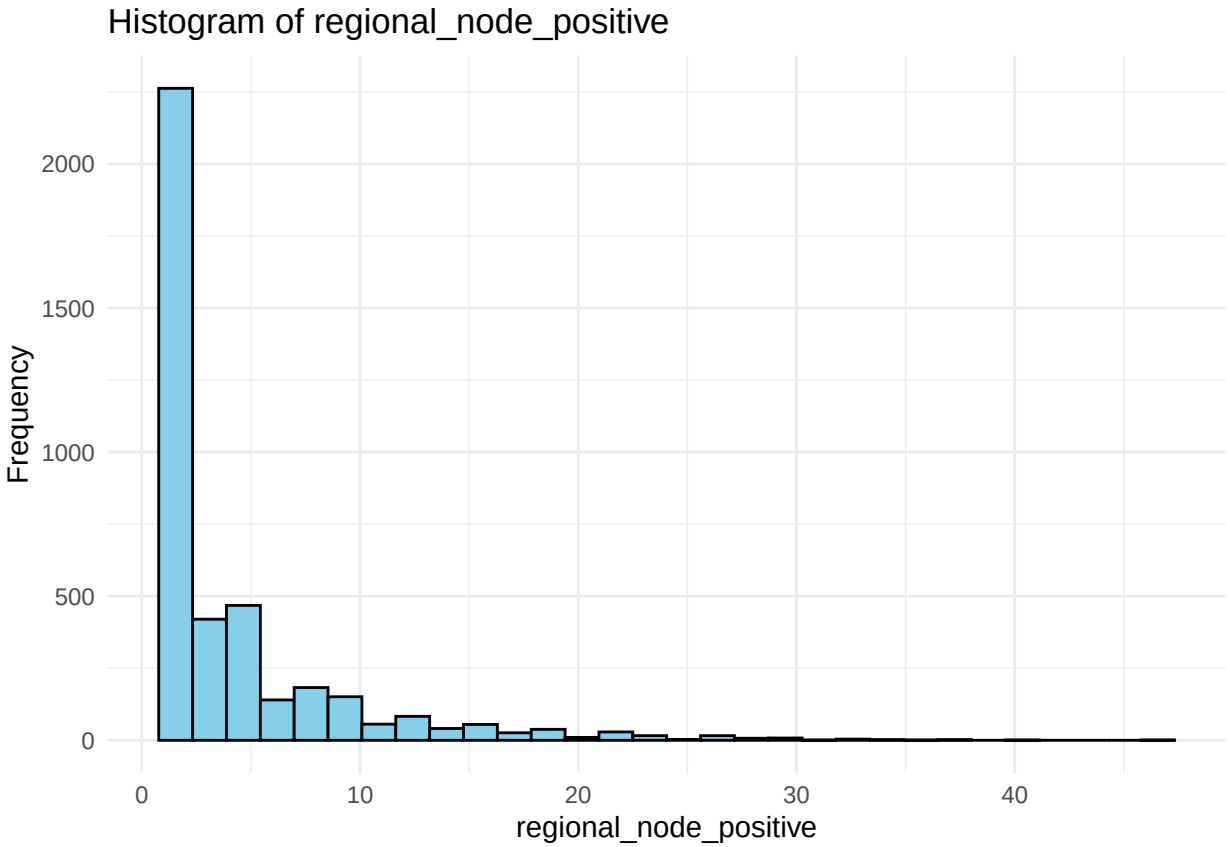

Boxplot of grade
grade
3.0
2.5
2.0
1.5
1.0
−0.4
−0.2
0.0
0.2
0.4

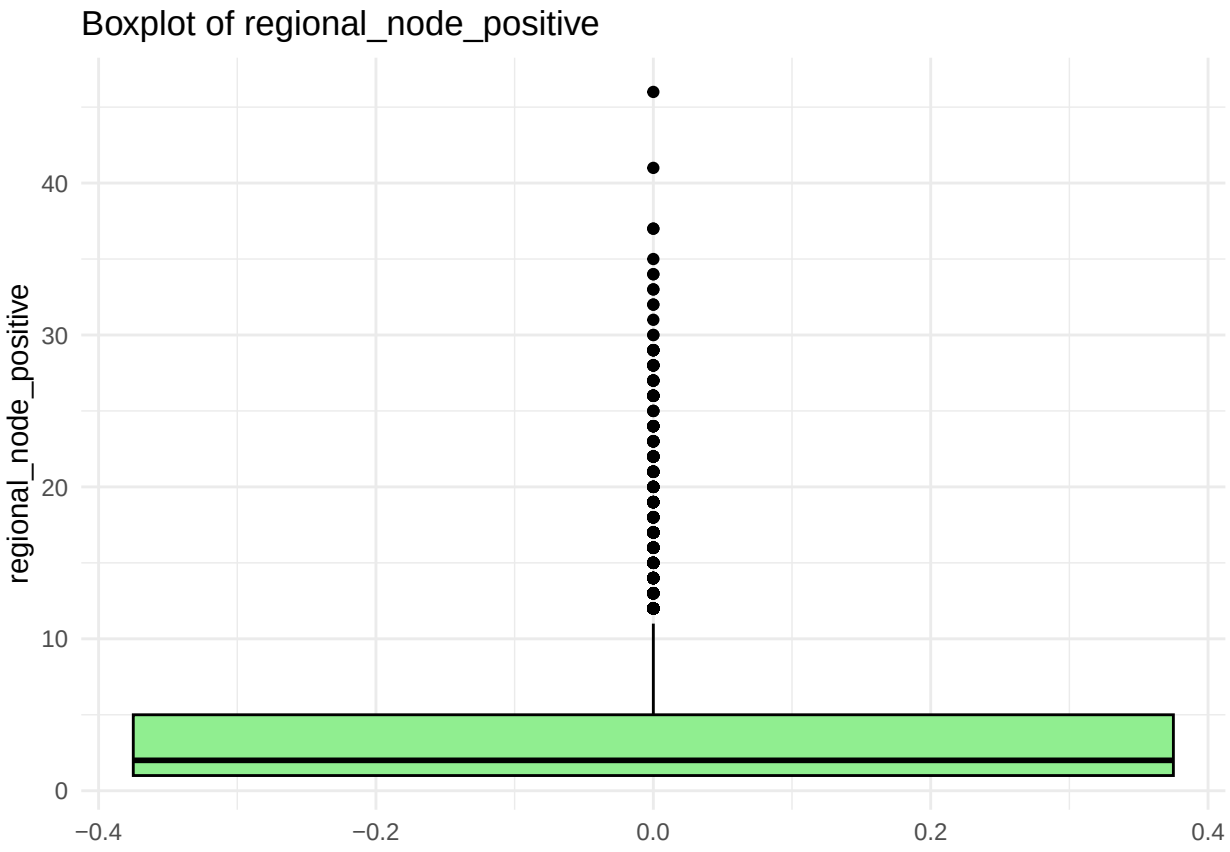

Histogram of tumor_size
Frequency
600
400
200
0
0
50
100
tumor_size

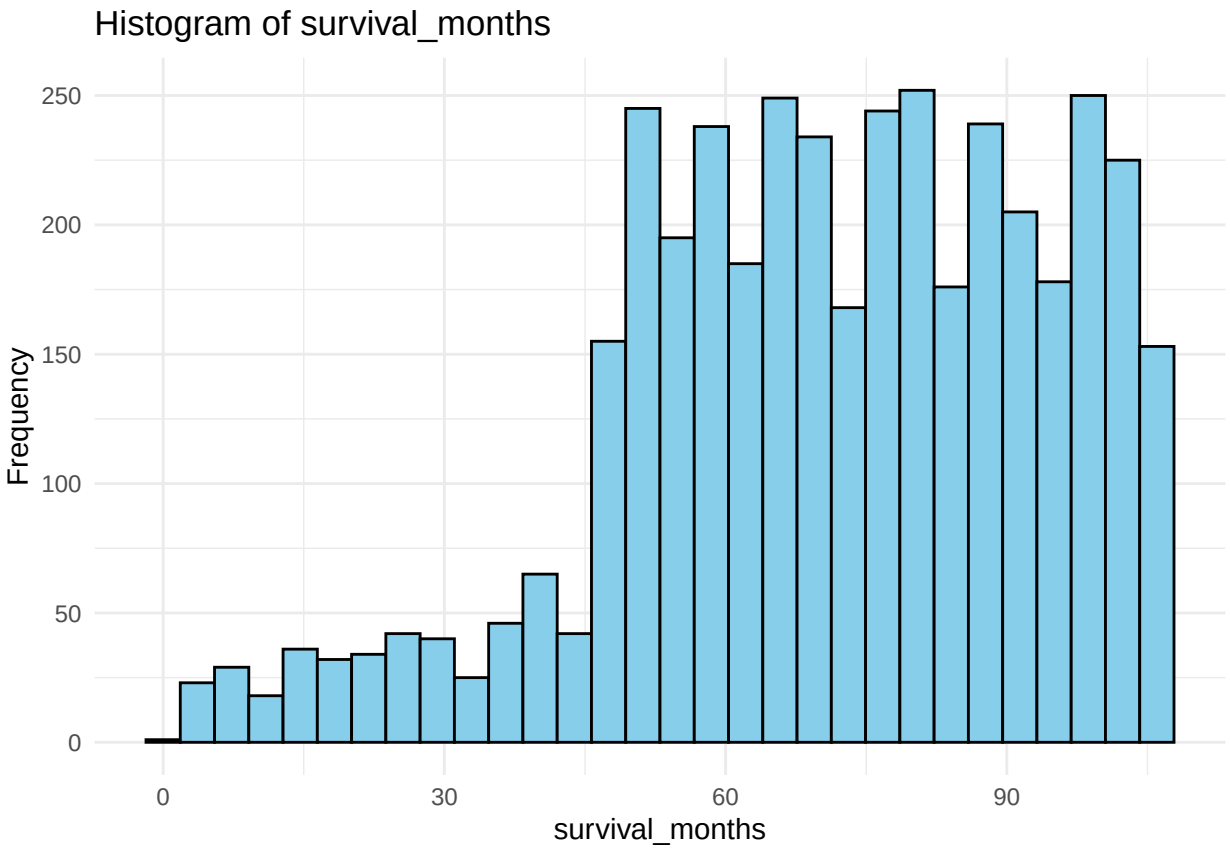

Boxplot of tumor_size
tumor_size
100
50
0
−0.4
−0.2
0.0
0.2
0.4

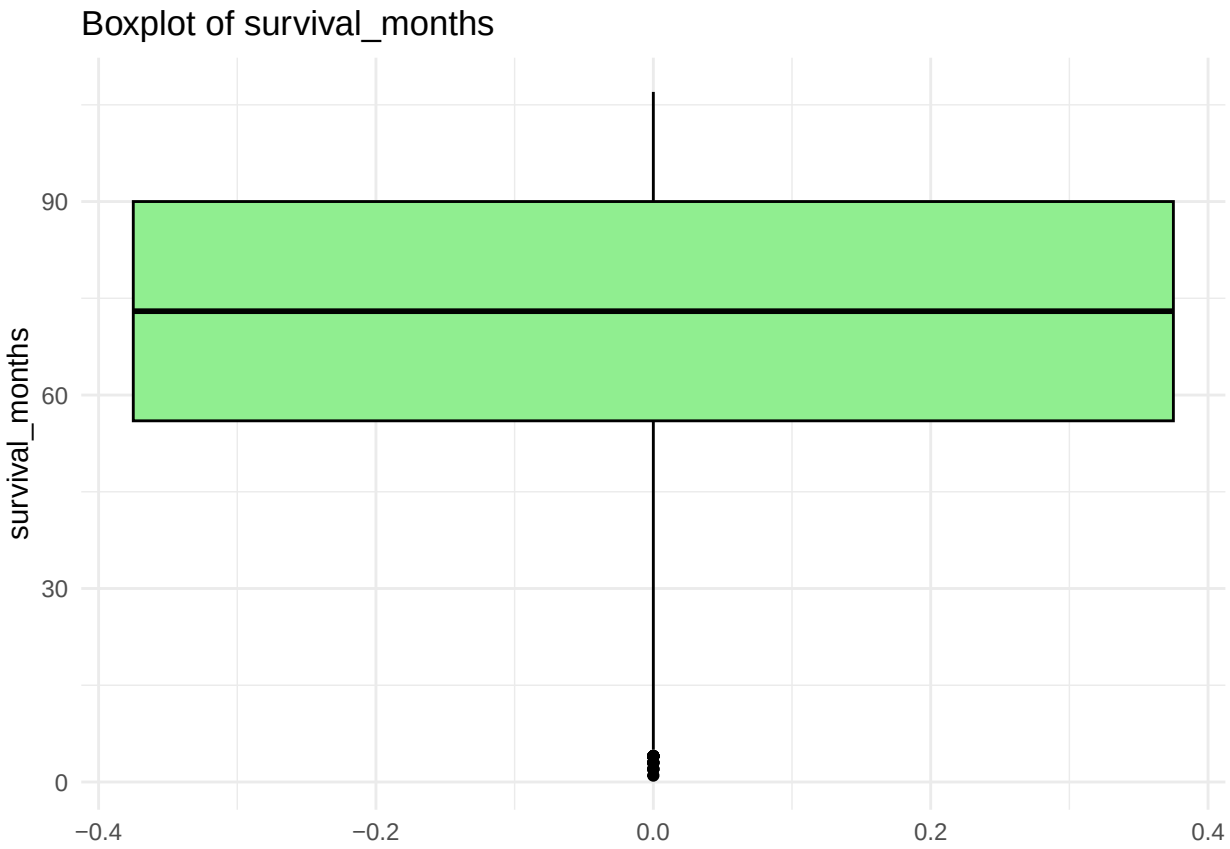

Histogram of regional_node_examined
Frequency
400
300
200
100
0
0
20
40
60
regional_node_examined


Boxplot of regional_node_examined
regional_node_examined
60
40
20
0
−0.4
−0.2
0.0
0.2
0.4


Histogram of regional_node_positive
Frequency
2000
1500
1000
500
0
0
10
20
30
40
regional_node_positive


Boxplot of regional_node_positive
regional_node_positive
40
30
20
10
0
−0.4
−0.2
0.0
0.2
0.4

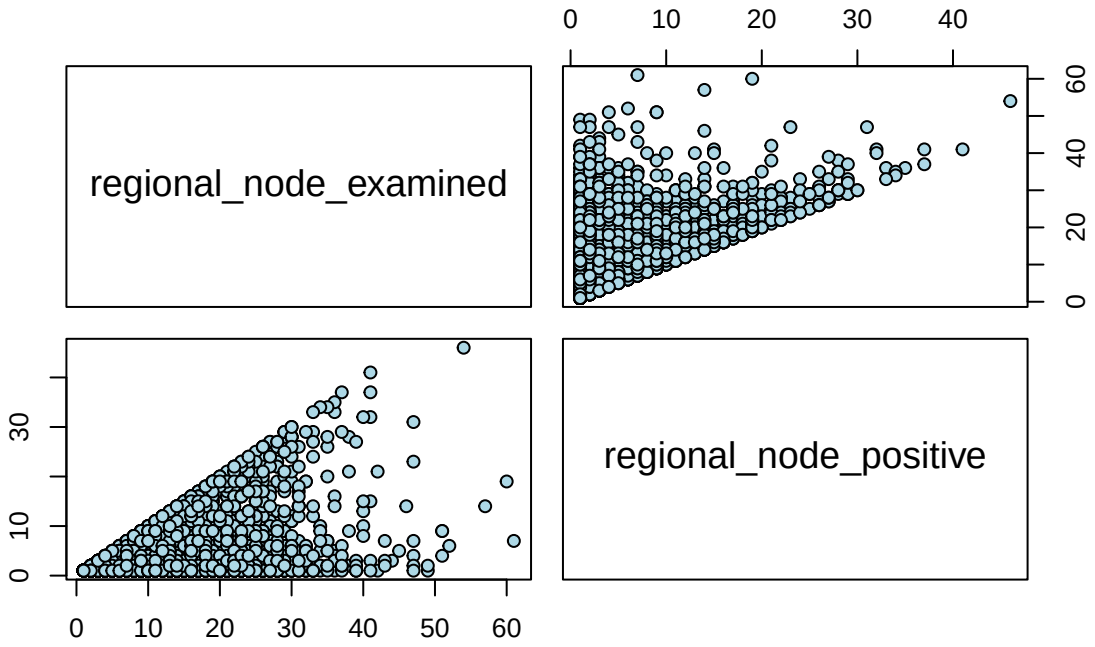

Histogram of survival_months
Frequency
250
200
150
100
50
0
0
30
60
90
survival_months



see the correlation between regional.node.examined and regional.node.positive

```
pairs_data <- data[, c("regional_node_examined", "regional_node_positive")]

pairs(pairs_data, main = "Pairs Plot: Regional Node Examined vs Regional Node Positive",
      pch = 21, bg = "lightblue")
```

Pairs Plot: Regional Node Examined vs Regional Node Positive



Linear Trend: There appears to be a positive association between Regional.Node.Examined and Regional.Node.Positive. As the number of nodes examined increases, the number of positive nodes also tends to increase. However, the relationship is not perfectly linear; there is noticeable spread in the points.

High Variability: There is significant variability in Regional.Node.Positive values for a given range of Regional.Node.Examined. This suggests that other factors might influence the number of positive nodes beyond the number of examined nodes.

Outliers: A few observations stand out as potential outliers, particularly where Regional.Node.Examined is high, but the number of Regional.Node.Positive remains low (or vice versa). These outliers could be influential points worth further investigation.

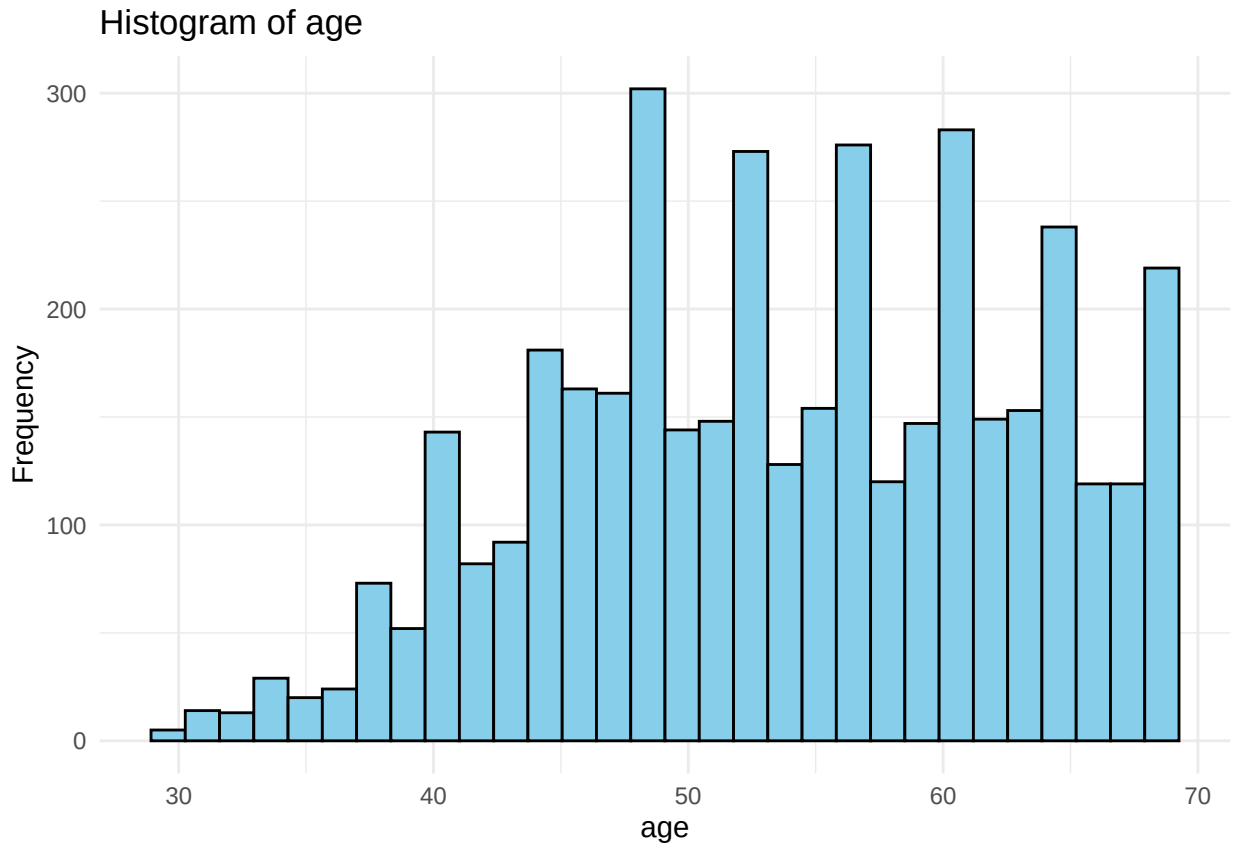
introduce a new variable proportion

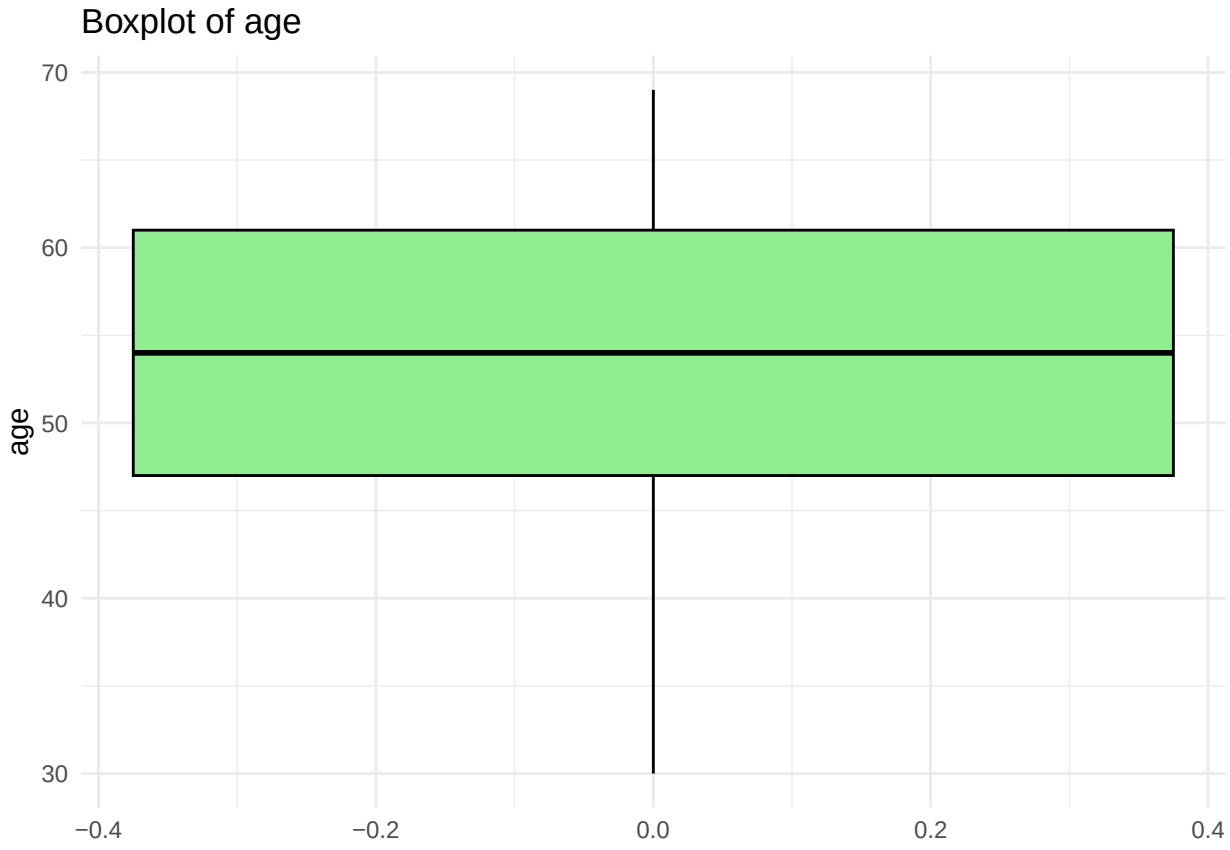
```
data = data%>%
  mutate(node_proportion = regional_node_examined/regional_node_positive)
```

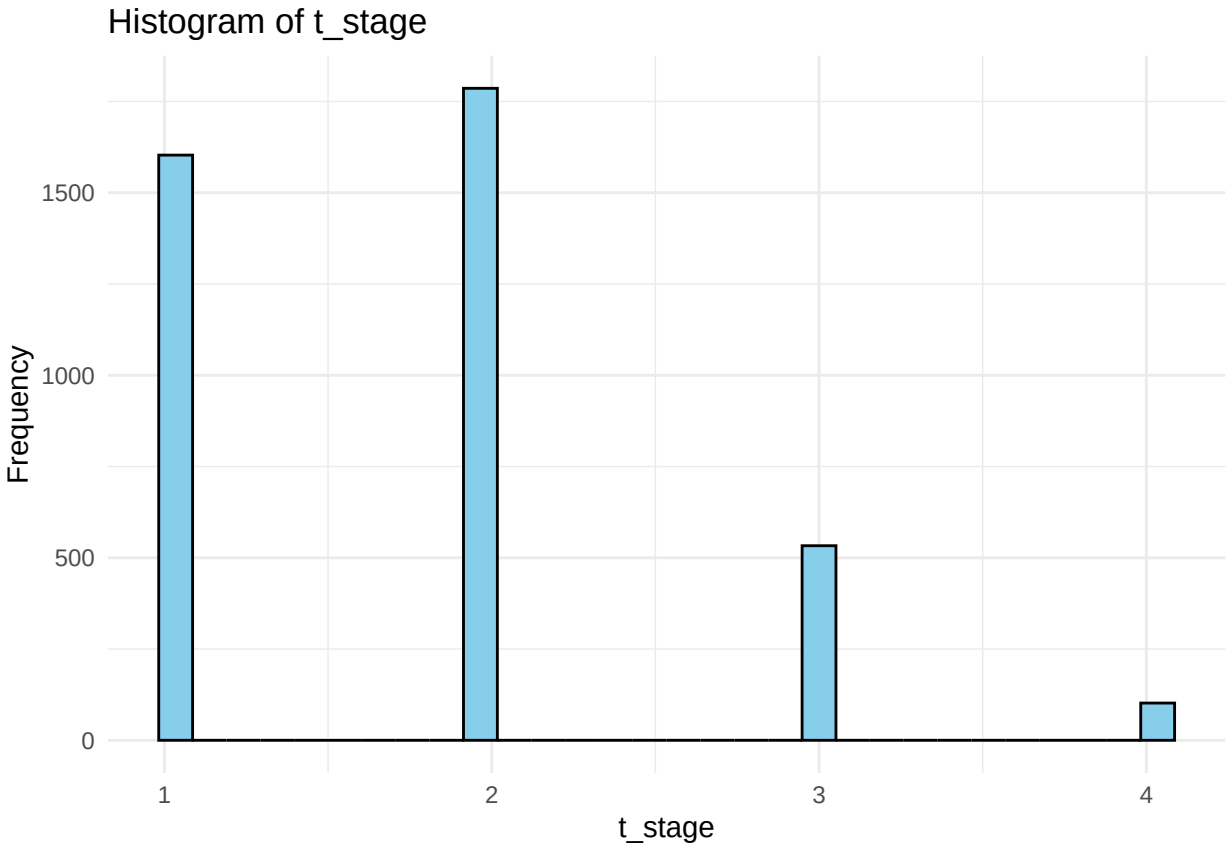
see the plot after the second preprocess

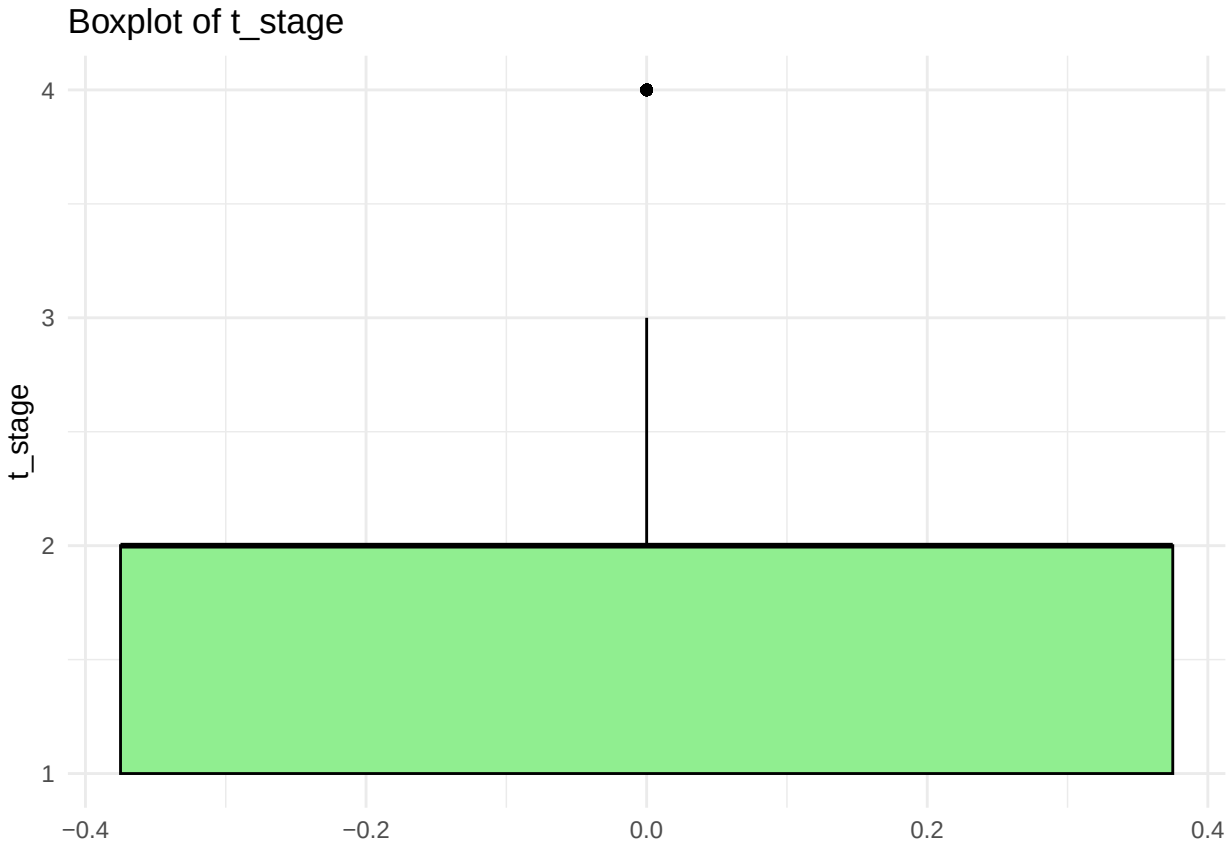
```
for (var in names(data)) {
  # Skip if the column is not numeric
  if (is.numeric(data[[var]])) {

    # Histogram
```

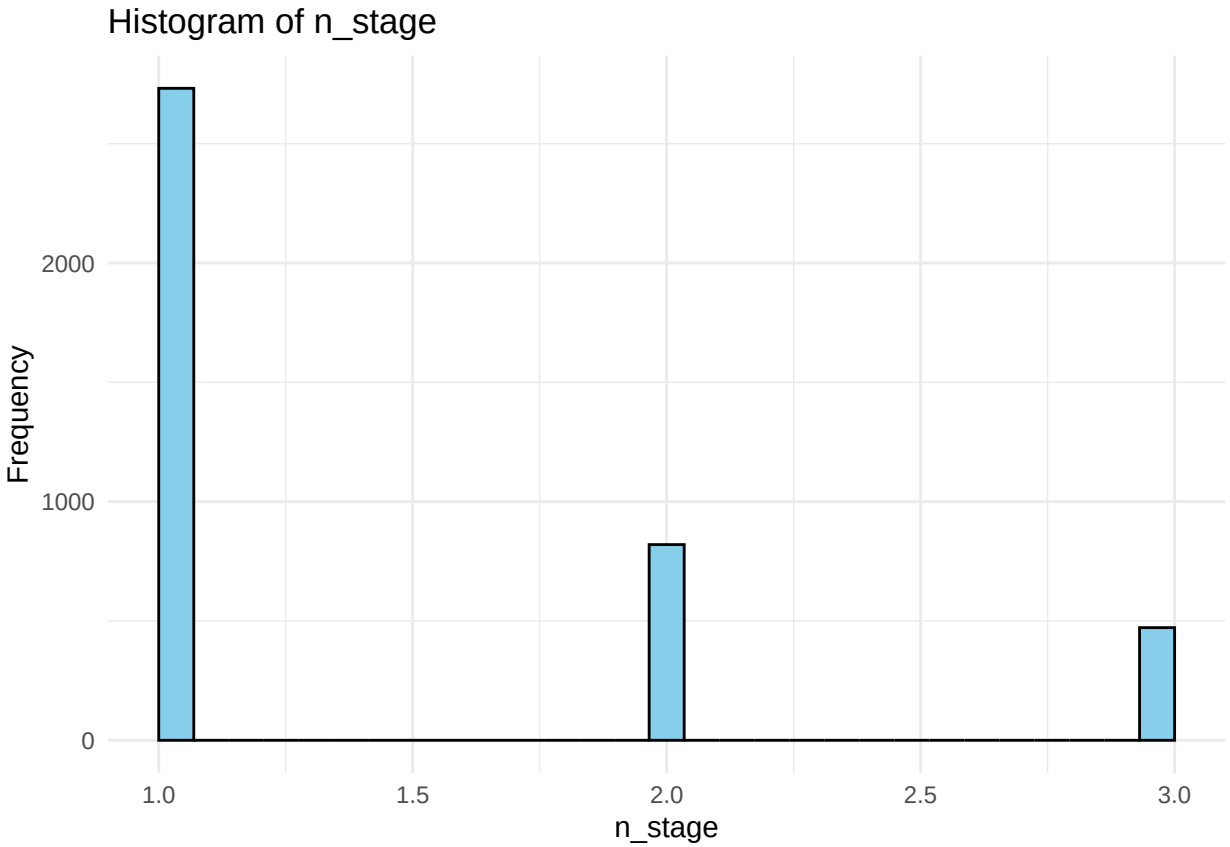
```
    p1 <- ggplot(data, aes_string(x = var)) +
      geom_histogram(fill = "skyblue", color = "black", bins = 30) +
      labs(title = paste("Histogram of", var), x = var, y = "Frequency") +
      theme_minimal()
    print(p1)

    # Boxplot
    p2 <- ggplot(data, aes_string(y = var)) +
      geom_boxplot(fill = "lightgreen", color = "black") +
      labs(title = paste("Boxplot of", var), y = var) +
      theme_minimal()
    print(p2)
  }
}
```

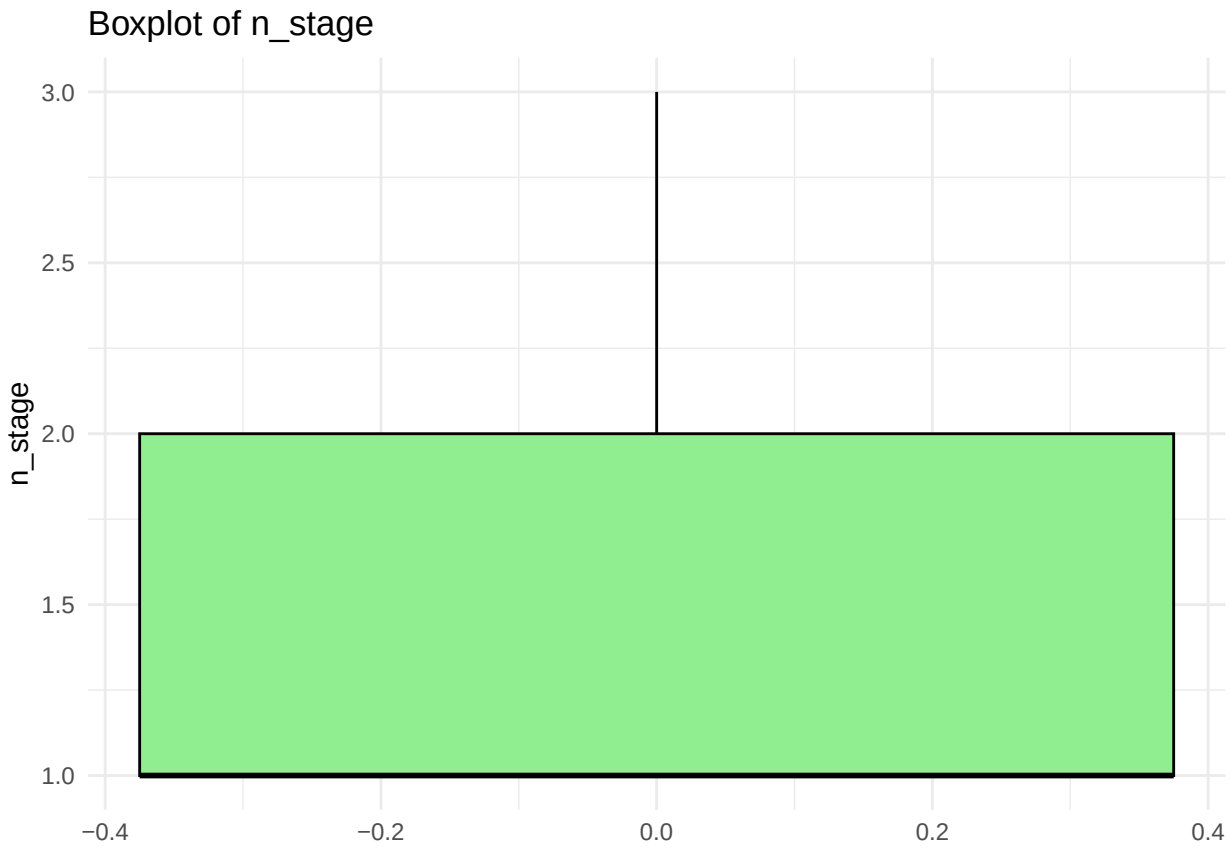



Boxplot of age
age
70
60
50
40
30
−0.4
−0.2
0.0
0.2
0.4

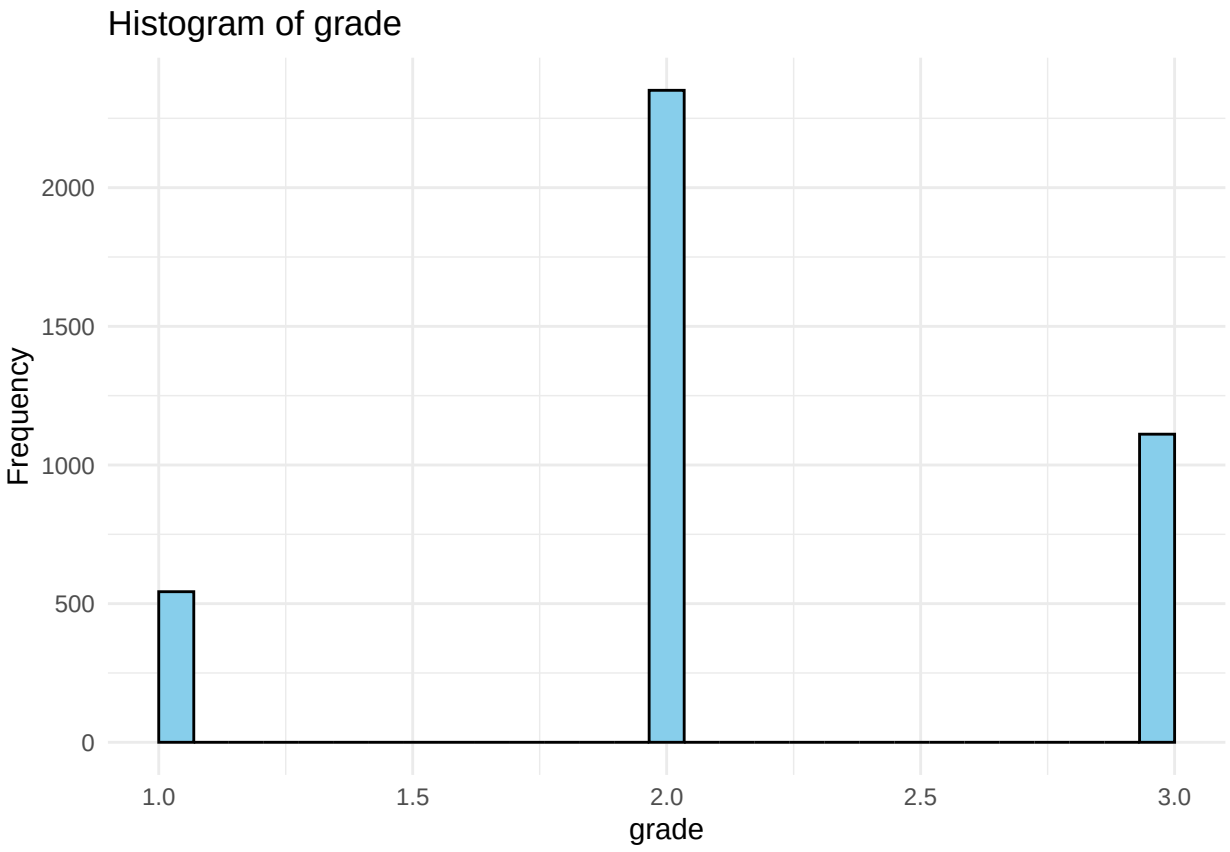

Histogram of t_stage
Frequency
1500
1000
500
0
1
2
3
4
t_stage


Boxplot of t_stage
t_stage
4
3
2
1
−0.4
−0.2
0.0
0.2
0.4

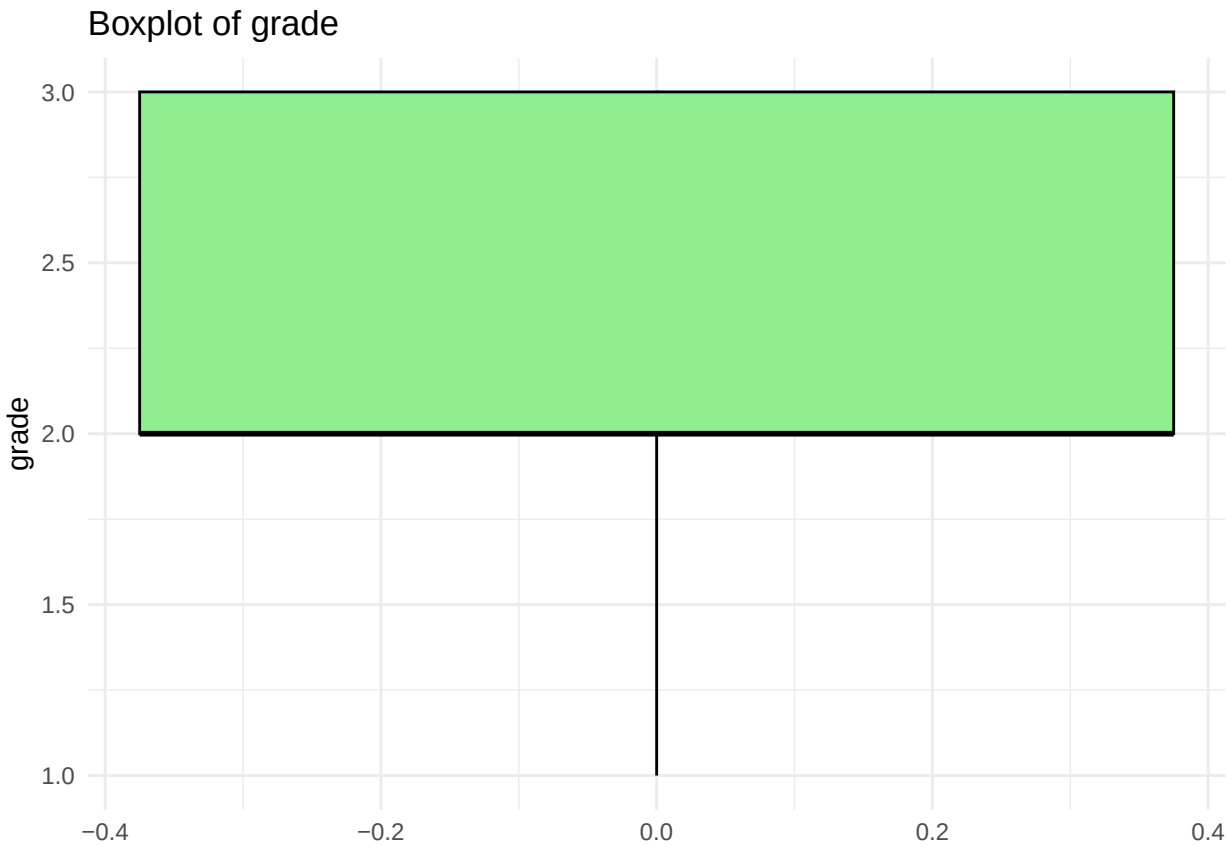

Histogram of n_stage
Frequency
2000
1000
0
1.0
1.5
2.0
2.5
3.0
n_stage

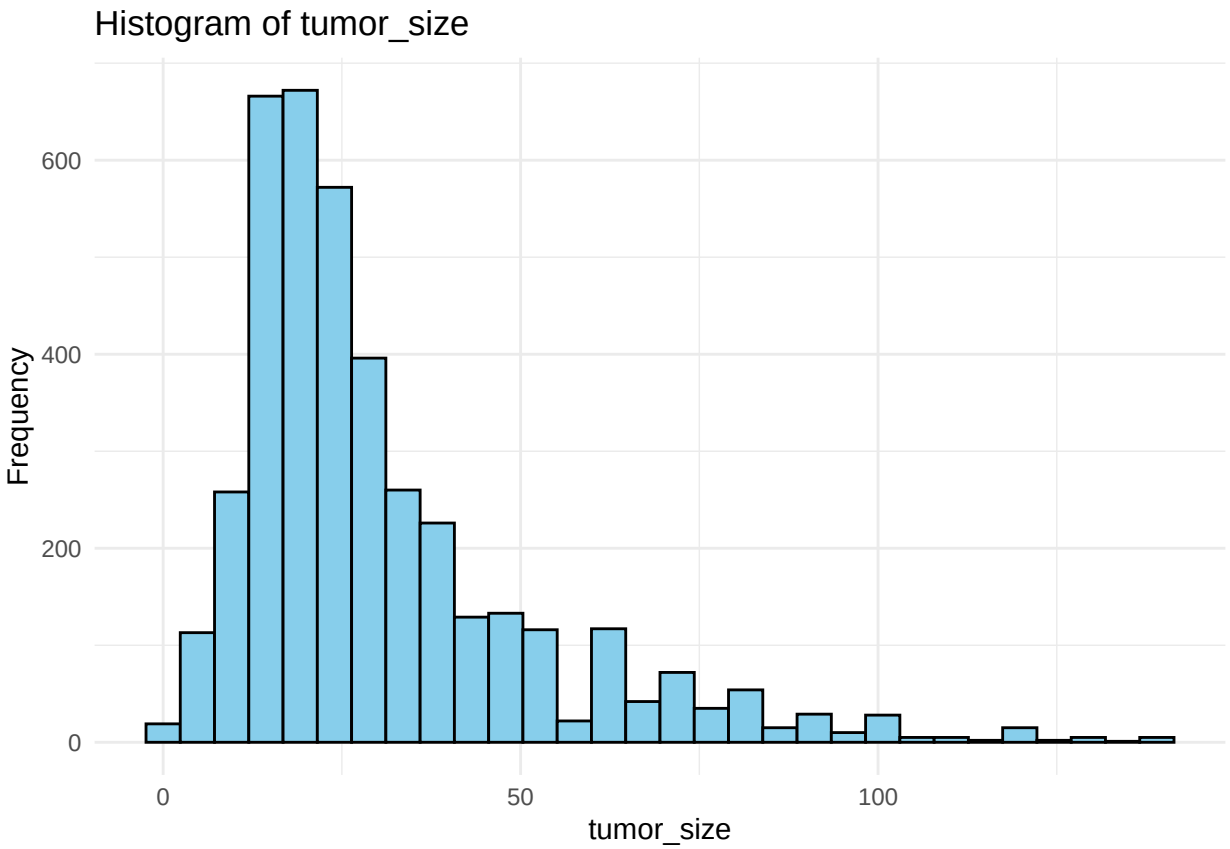


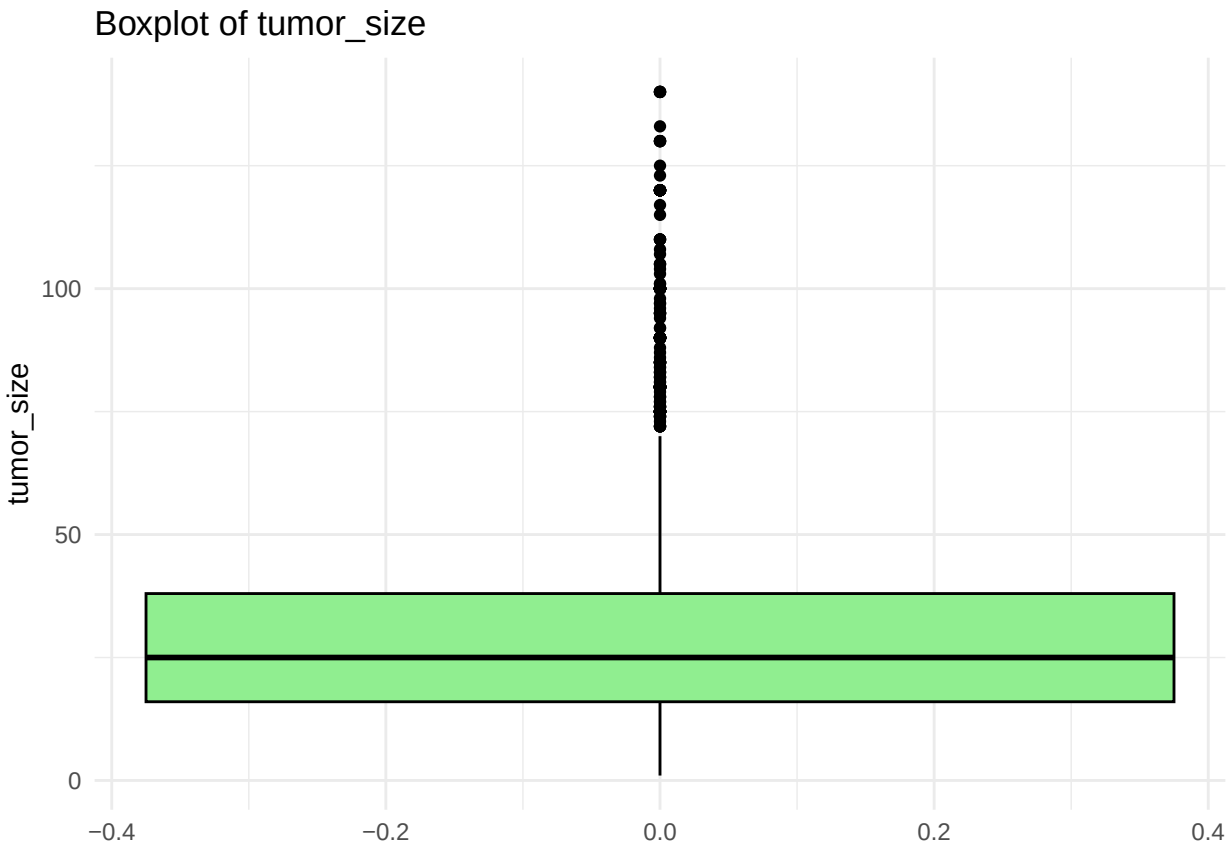

```
## Warning: Removed 19 rows containing non-finite outside the scale range
## (‘stat_bin()‘).
```

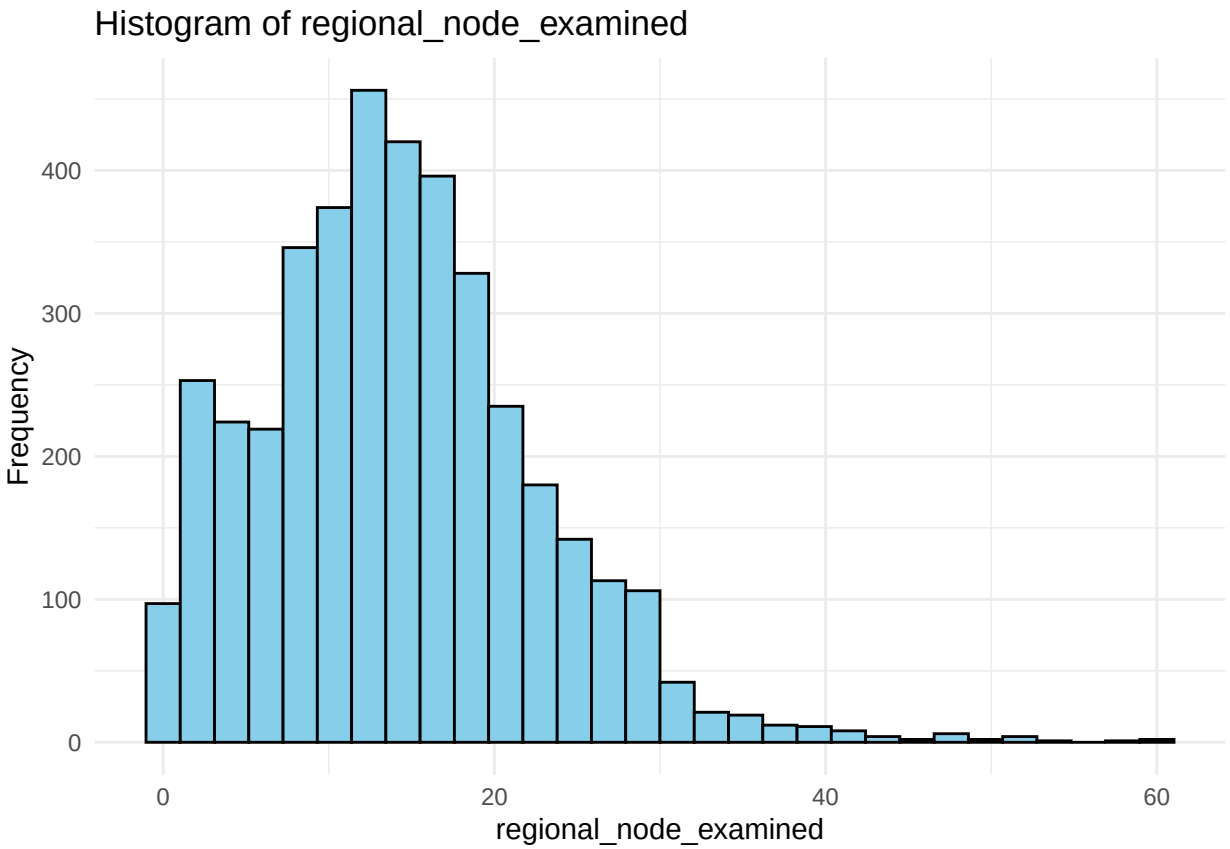


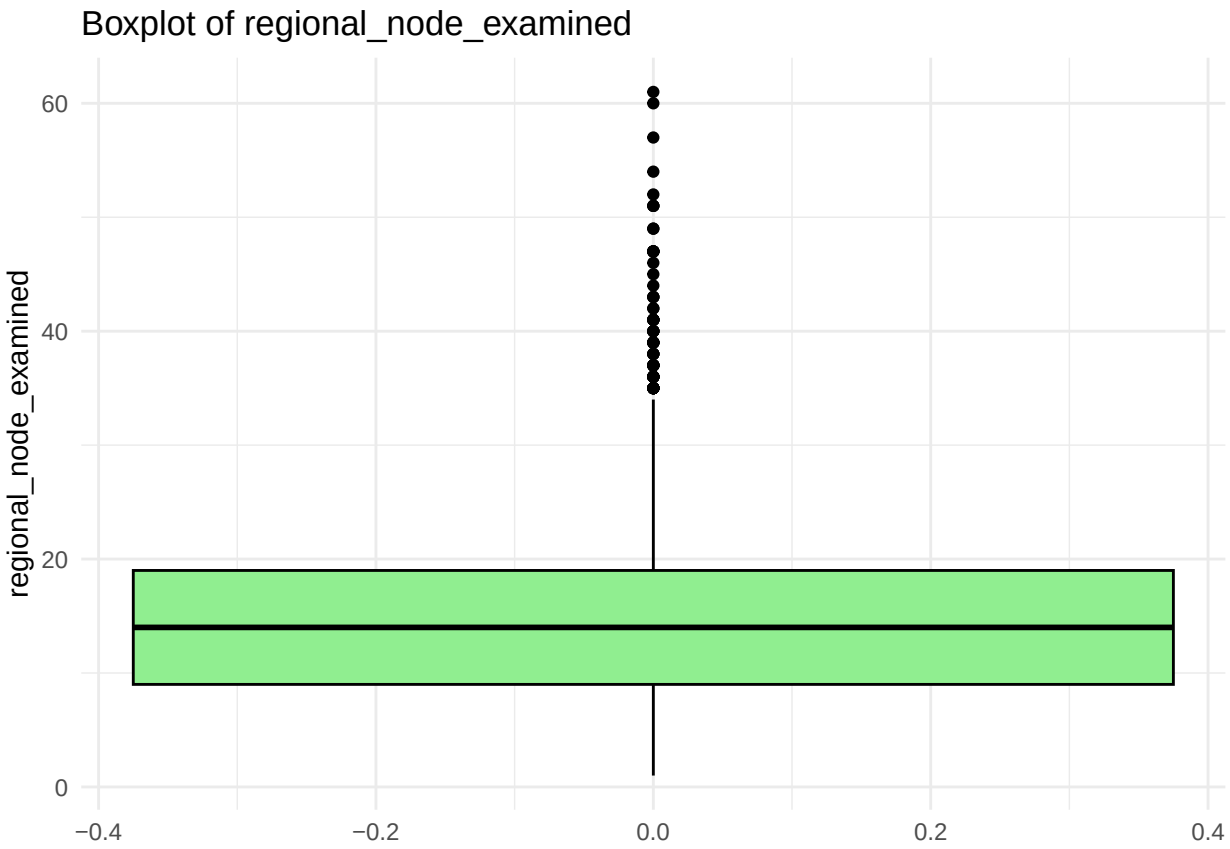

```
## Warning: Removed 19 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```

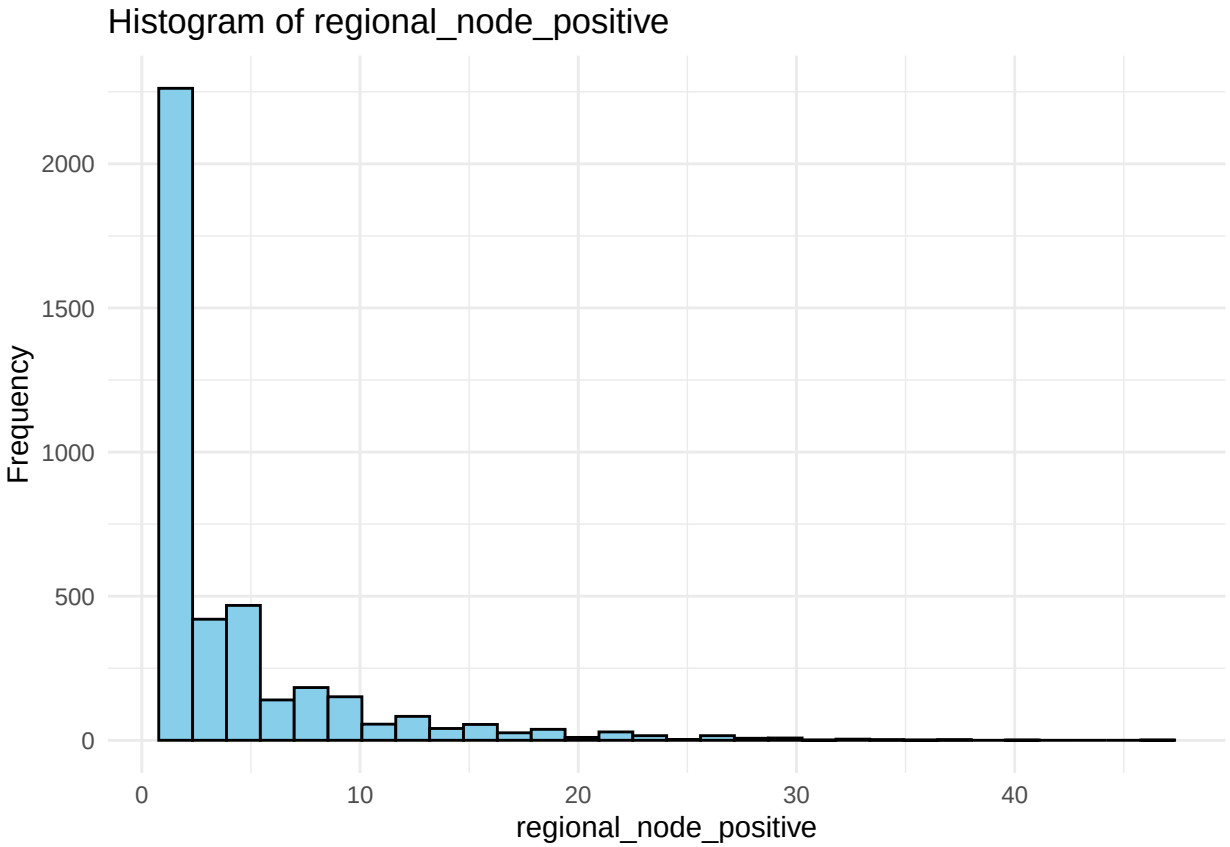

Boxplot of grade
grade
3.0
2.5
2.0
1.5
1.0
−0.4
−0.2
0.0
0.2
0.4

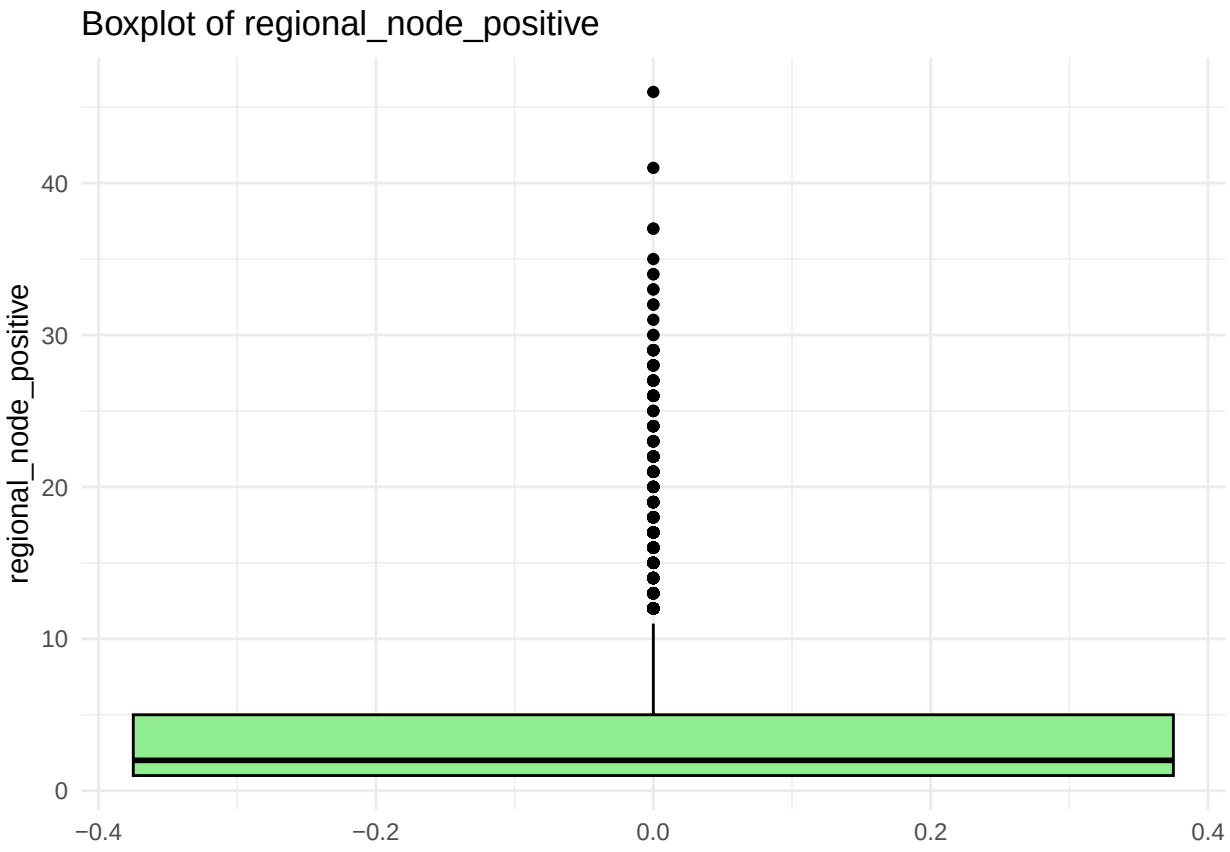

Histogram of tumor_size
Frequency
600
400
200
0
0
50
100
tumor_size

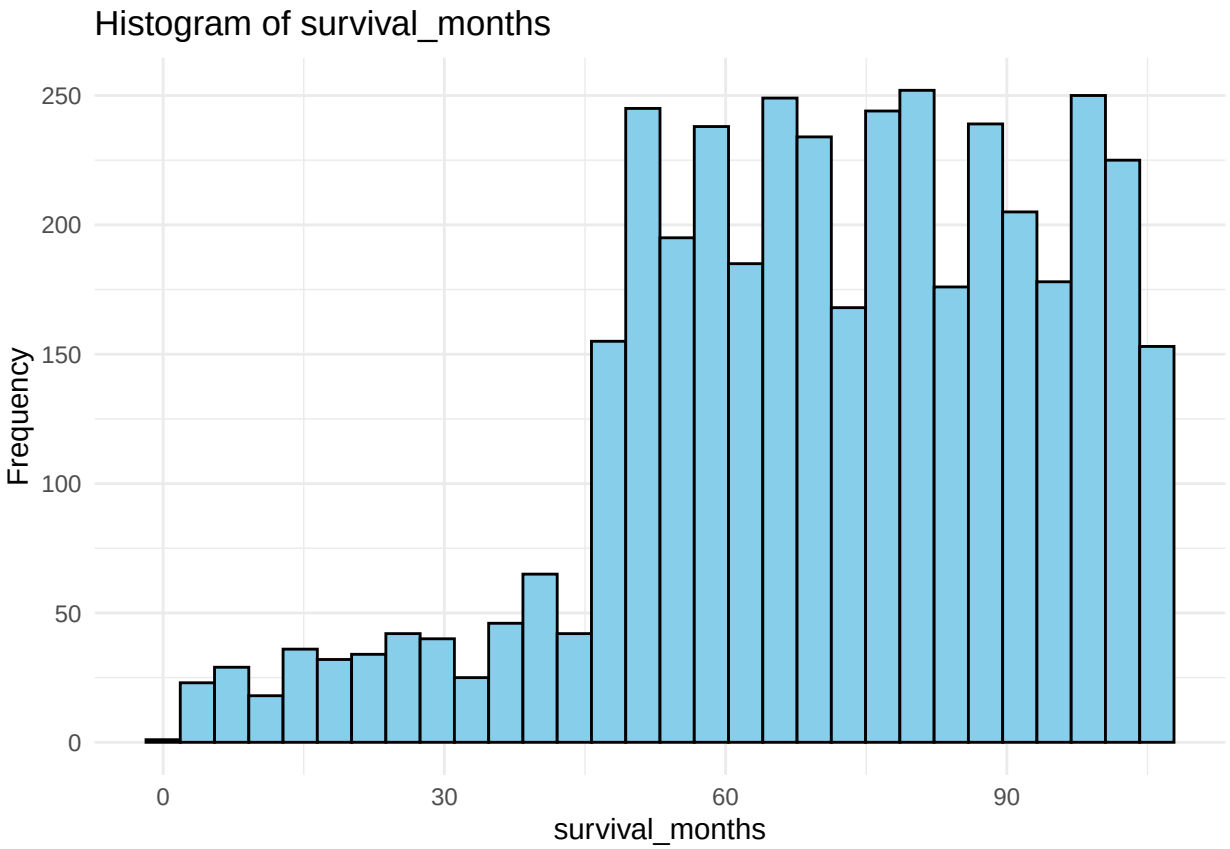

Boxplot of tumor_size
tumor_size
100
50
0
−0.4
−0.2
0.0
0.2
0.4

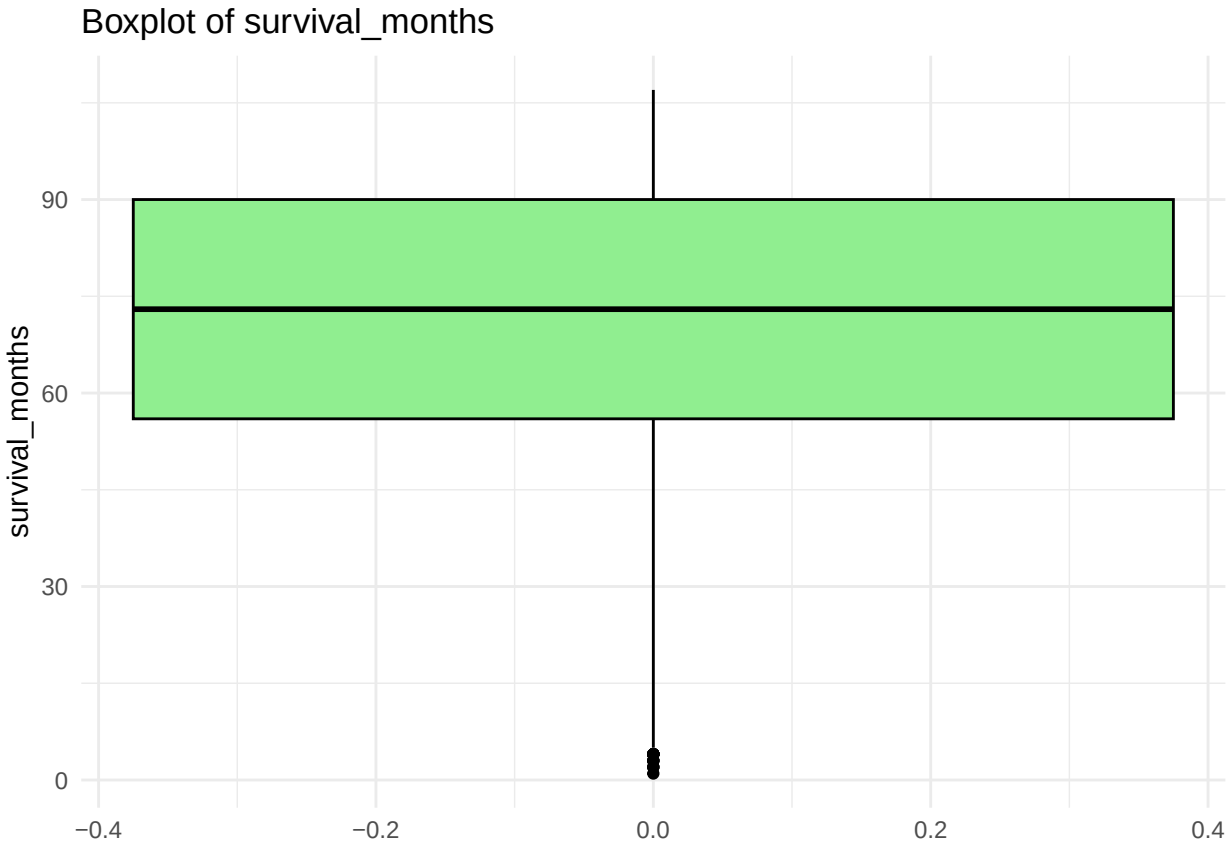

Histogram of regional_node_examined
Frequency
400
300
200
100
0
0
20
40
60
regional_node_examined

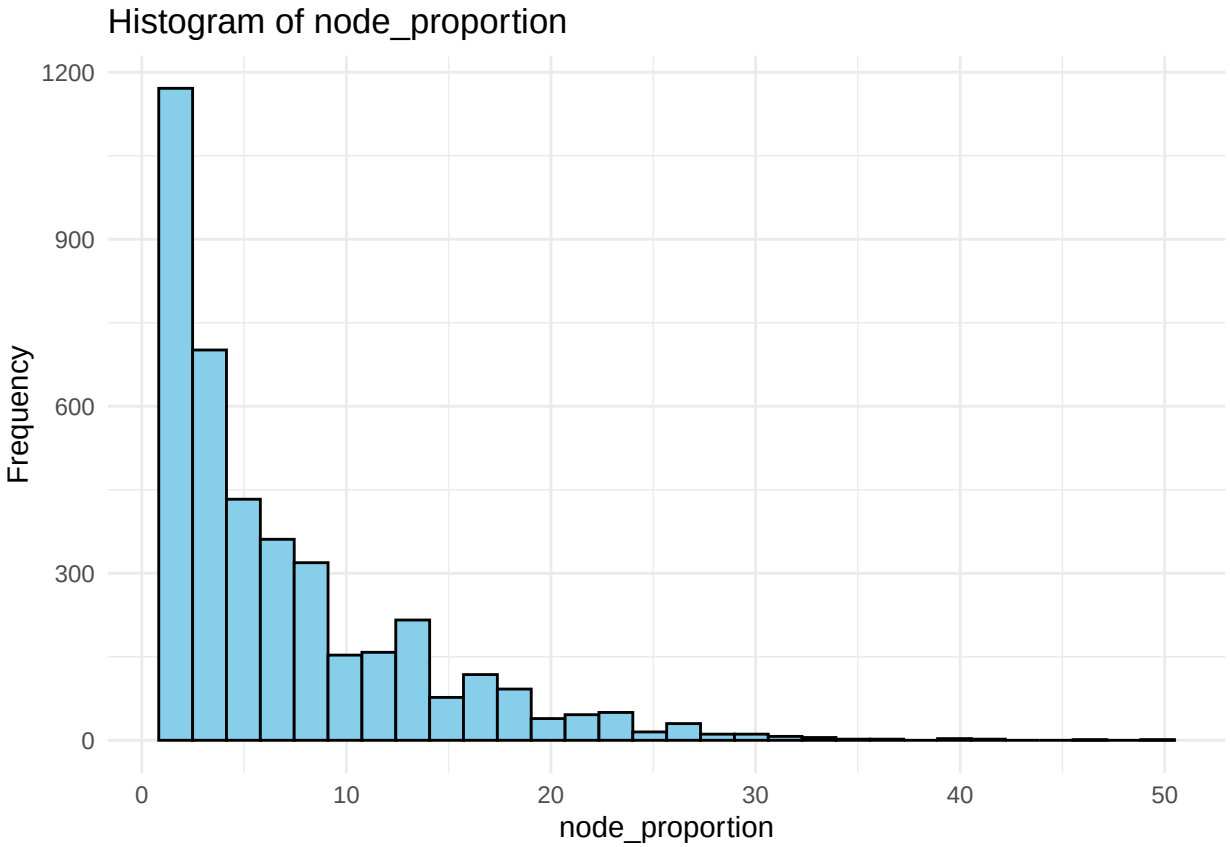

Boxplot of regional_node_examined
regional_node_examined
60
40
20
0
−0.4
−0.2
0.0
0.2
0.4

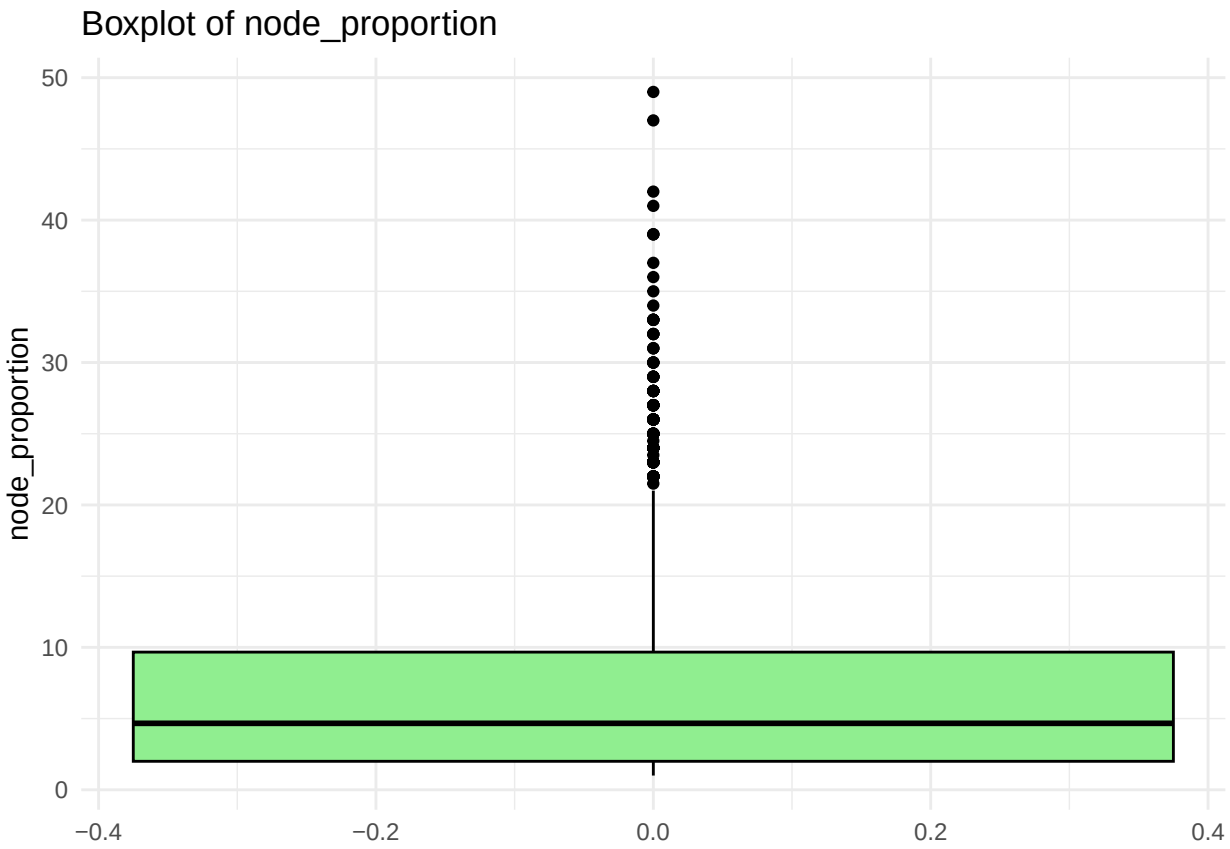

Histogram of regional_node_positive
Frequency
2000
1500
1000
500
0
0
10
20
30
40
regional_node_positive


Boxplot of regional_node_positive
regional_node_positive
40
30
20
10
0
−0.4
−0.2
0.0
0.2
0.4


Histogram of survival_months
Frequency
250
200
150
100
50
0
0
30
60
90
survival_months

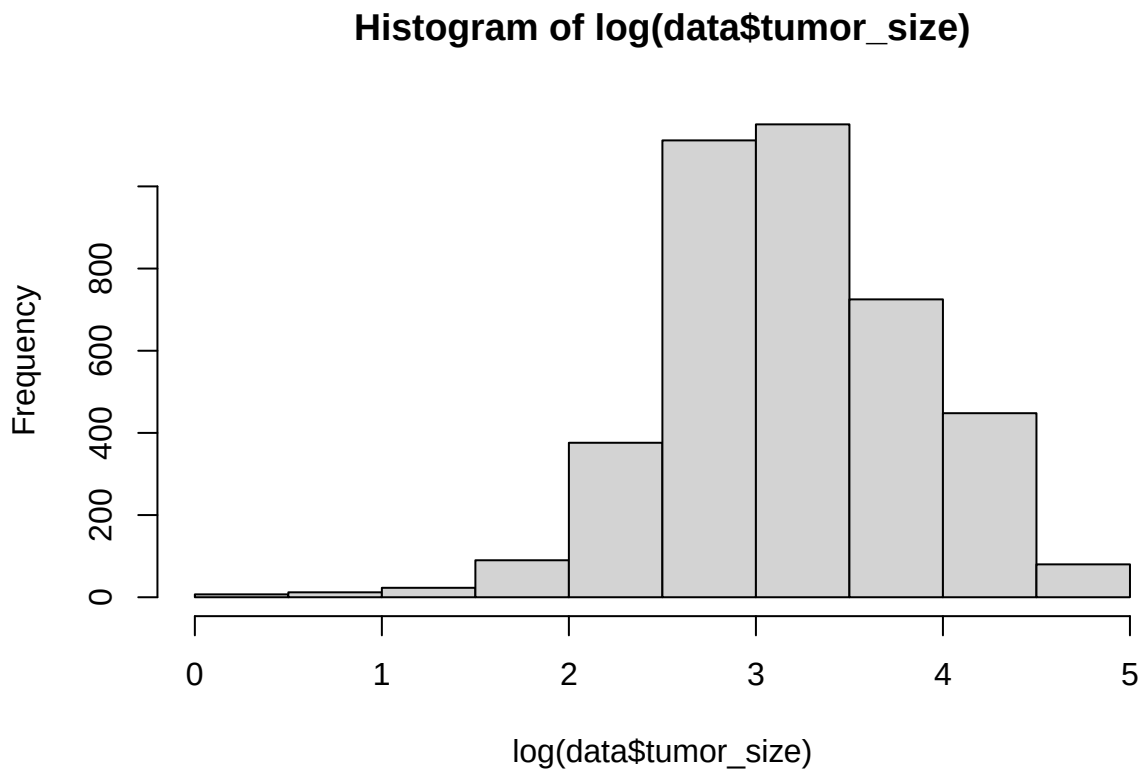

Boxplot of survival_months
survival_months
90
60
30
0
−0.4
−0.2
0.0
0.2
0.4


Histogram of node_proportion
Frequency
1200
900
600
300
0
0
10
20
30
40
50
node_proportion

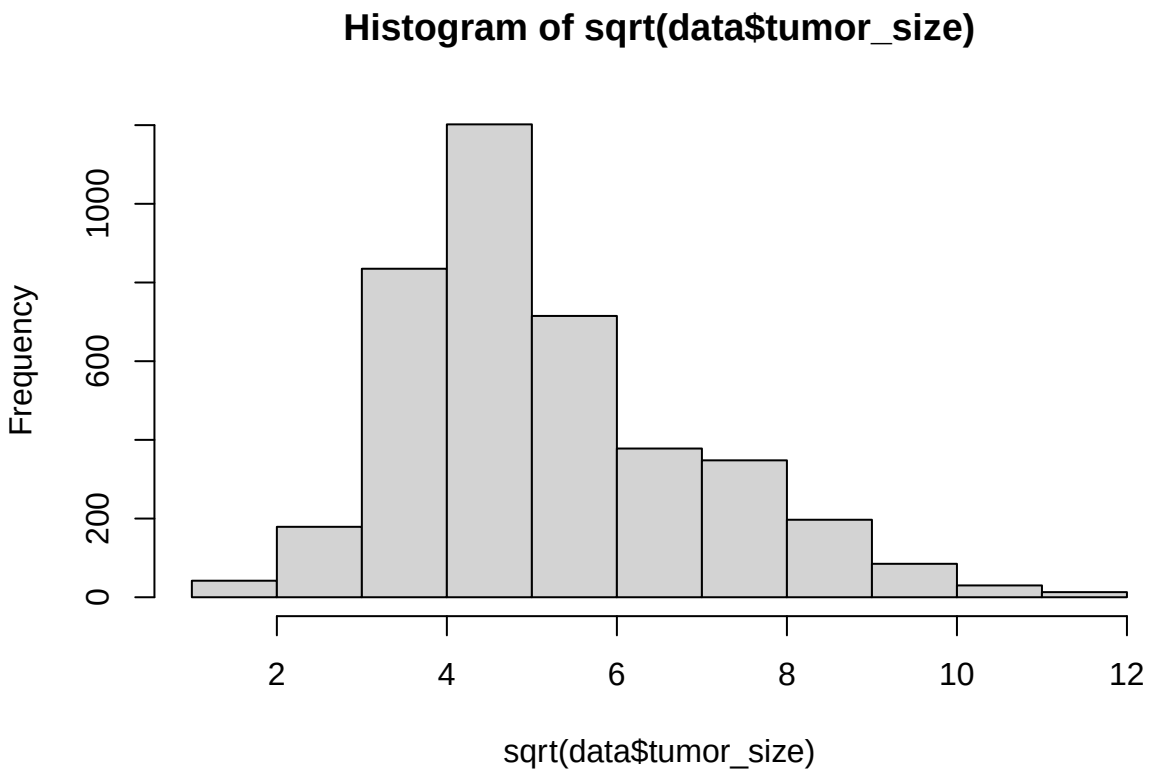


skewness of Tumor.size

```
hist(log(data$tumor_size))
```




```
hist(sqrt(data$tumor_size))
```



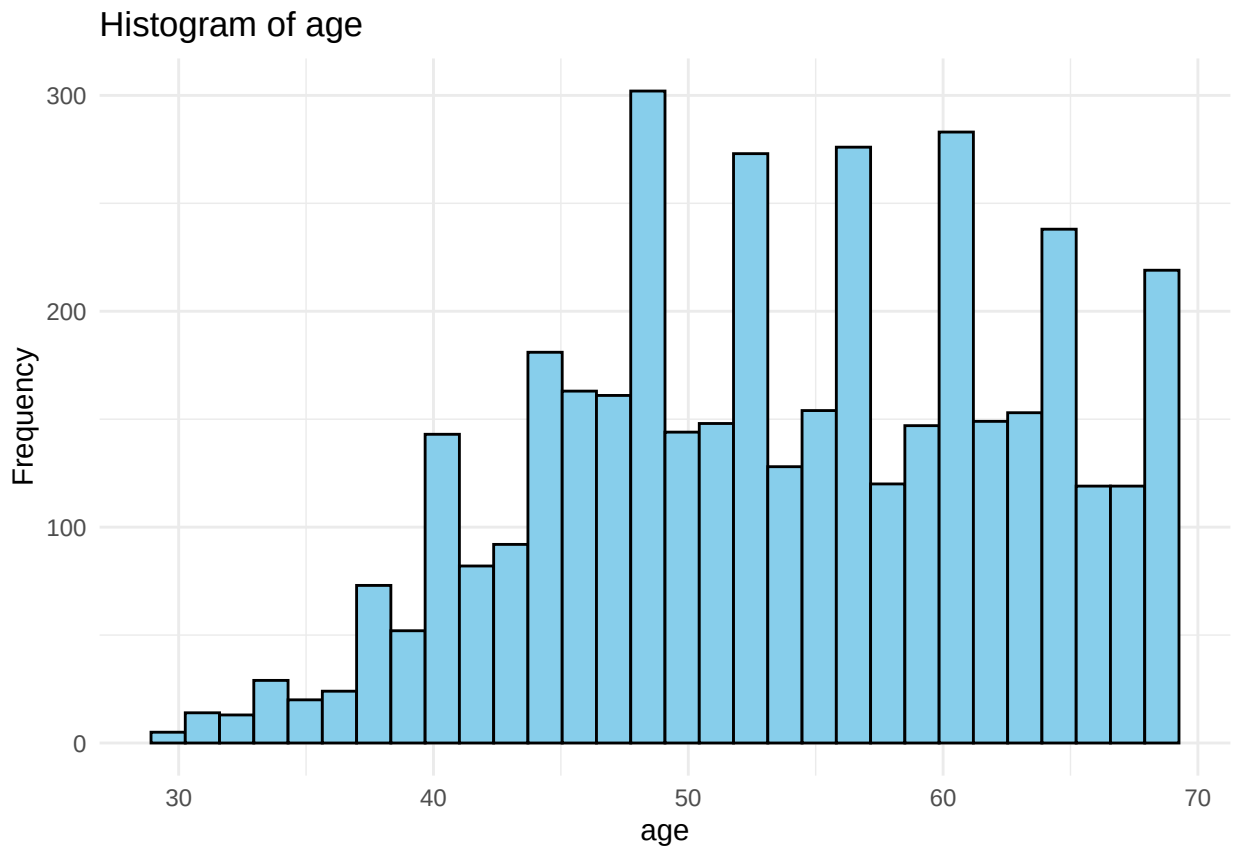
log transformation for tumor.size

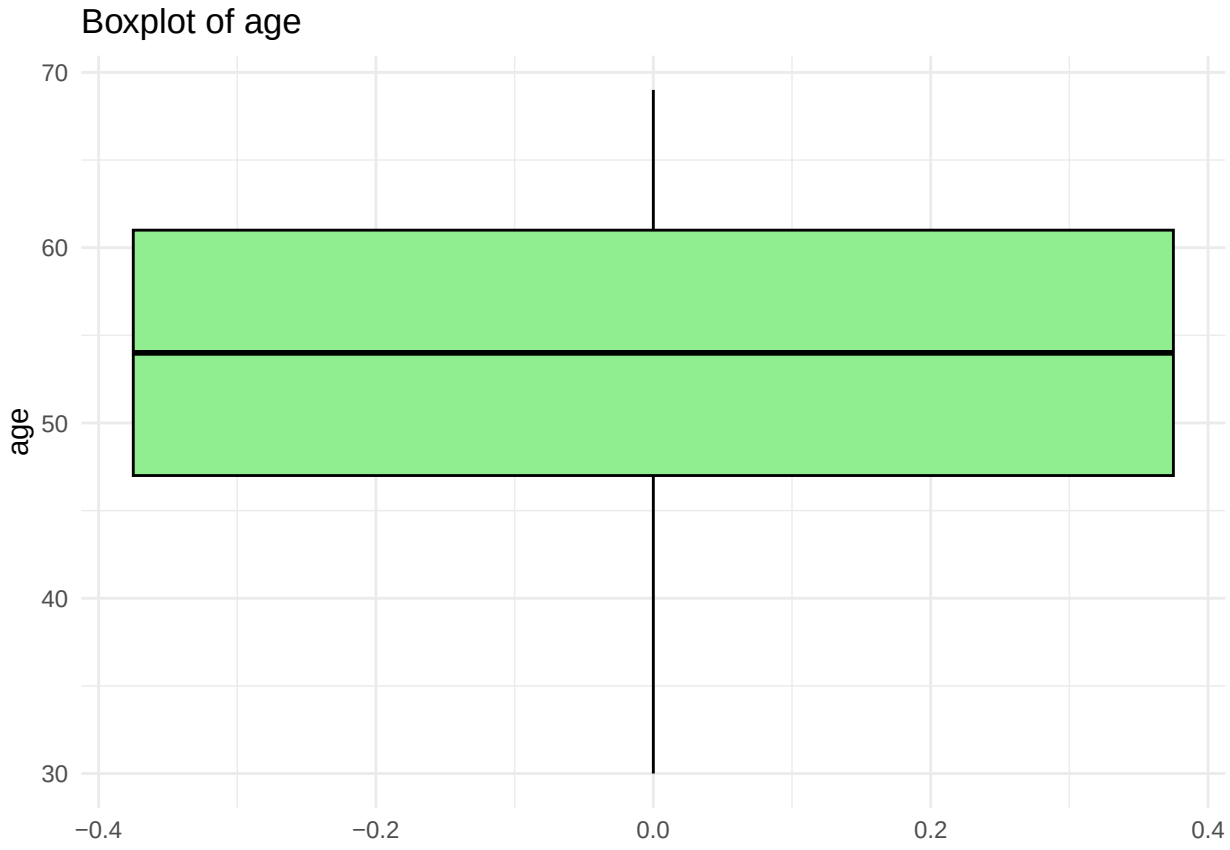
```
data = data %>%
  mutate(tumor_size = log(tumor_size))
```

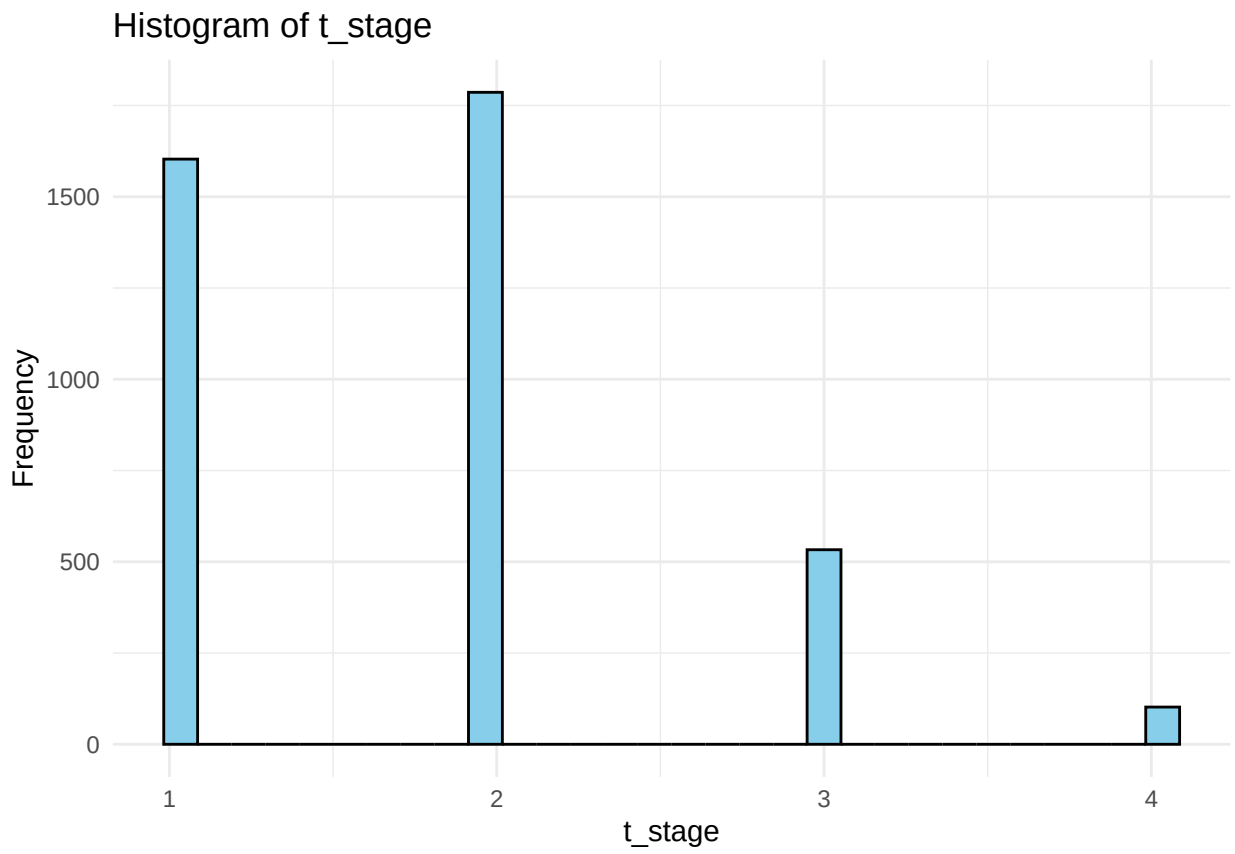
see the plot after the third preprocess

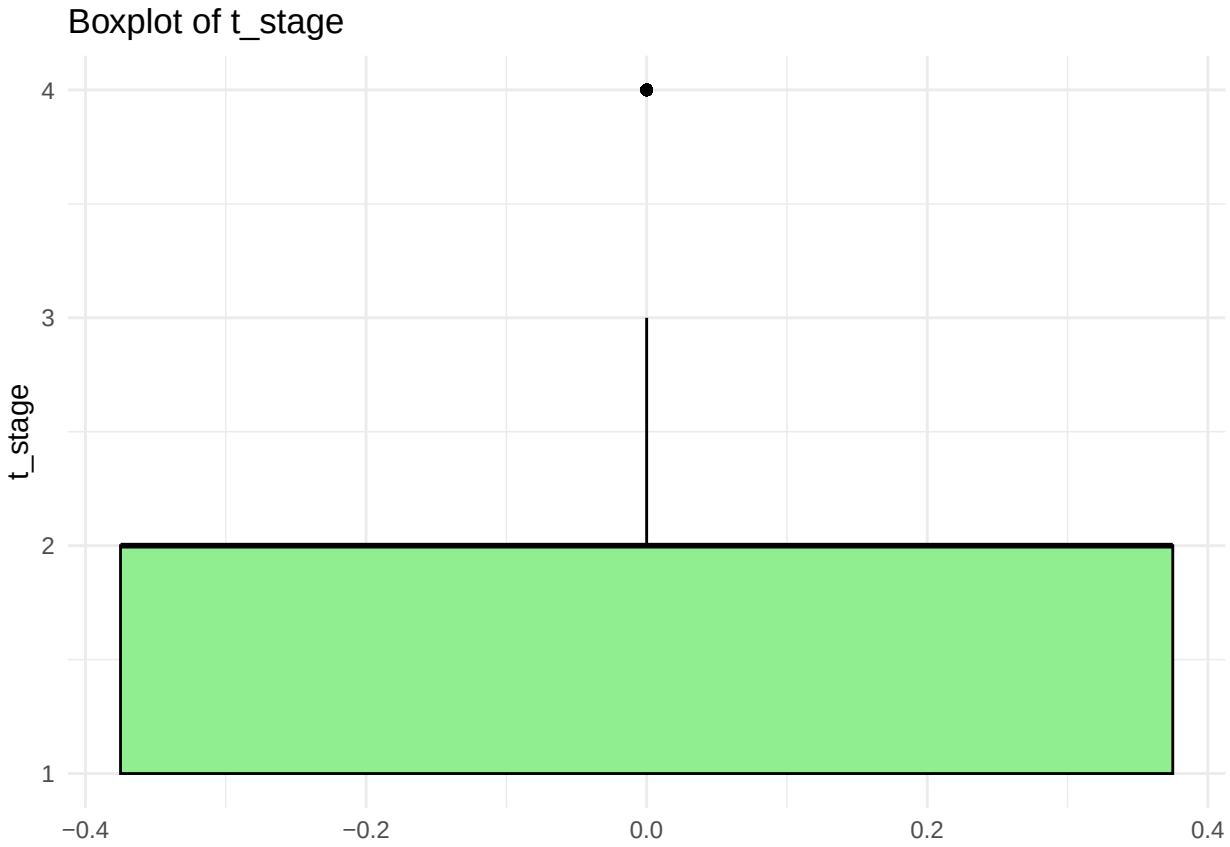
```
for (var in names(data)) {
  # Skip if the column is not numeric
  if (is.numeric(data[[var]])) {

    # Histogram
    p1 <- ggplot(data, aes_string(x = var)) +
      geom_histogram(fill = "skyblue", color = "black", bins = 30) +
      labs(title = paste("Histogram of", var), x = var, y = "Frequency") +
      theme_minimal()
    print(p1)

    # Boxplot
    p2 <- ggplot(data, aes_string(y = var)) +
      geom_boxplot(fill = "lightgreen", color = "black") +
      labs(title = paste("Boxplot of", var), y = var) +
```

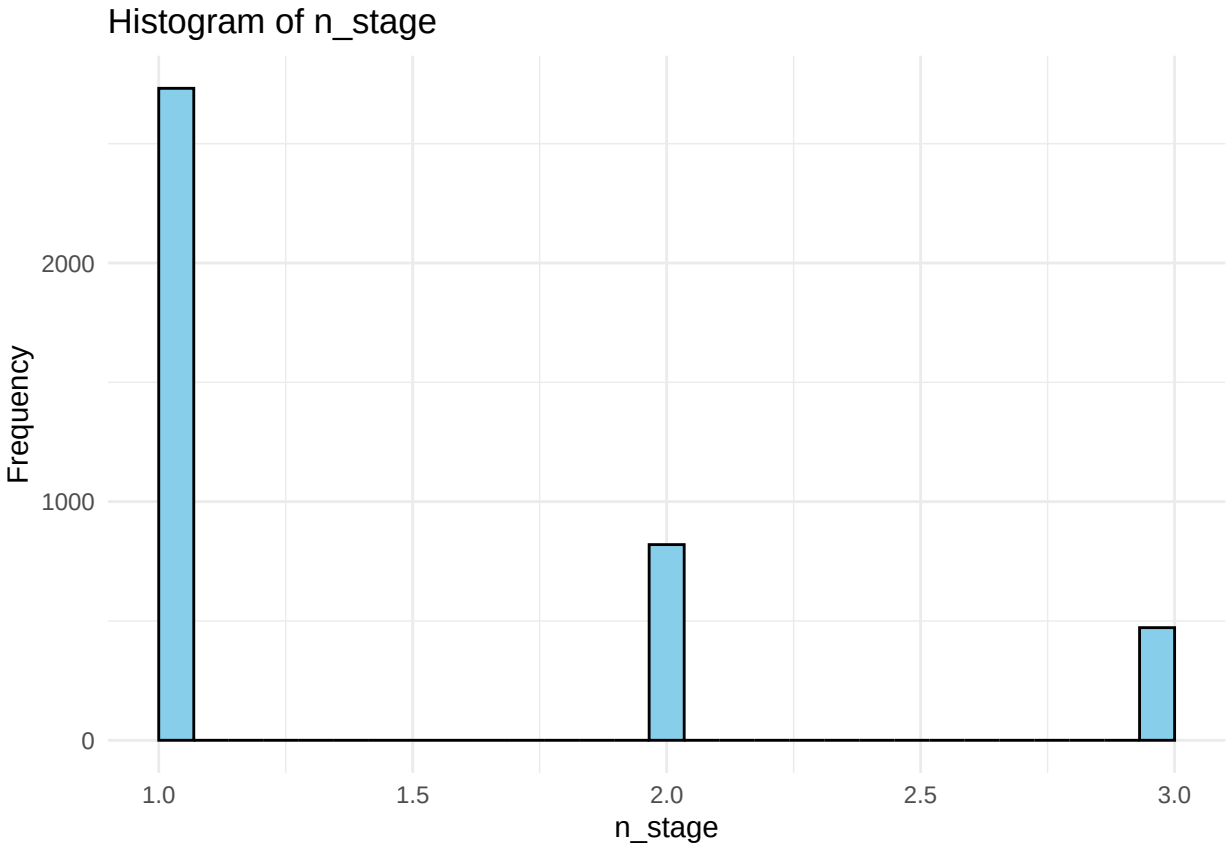
```
      theme_minimal()
    print(p2)
  }
}
```

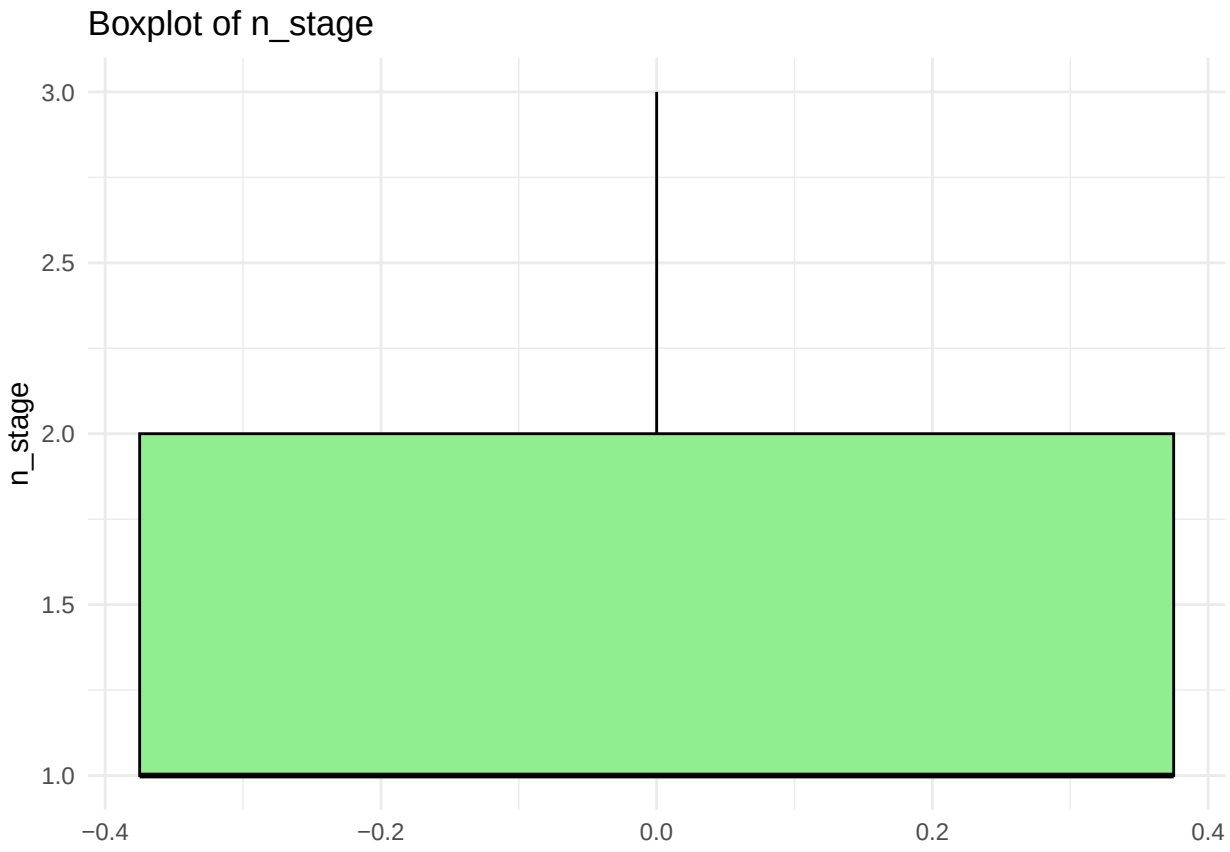



Boxplot of age
age
70
60
50
40
30
−0.4
−0.2
0.0
0.2
0.4

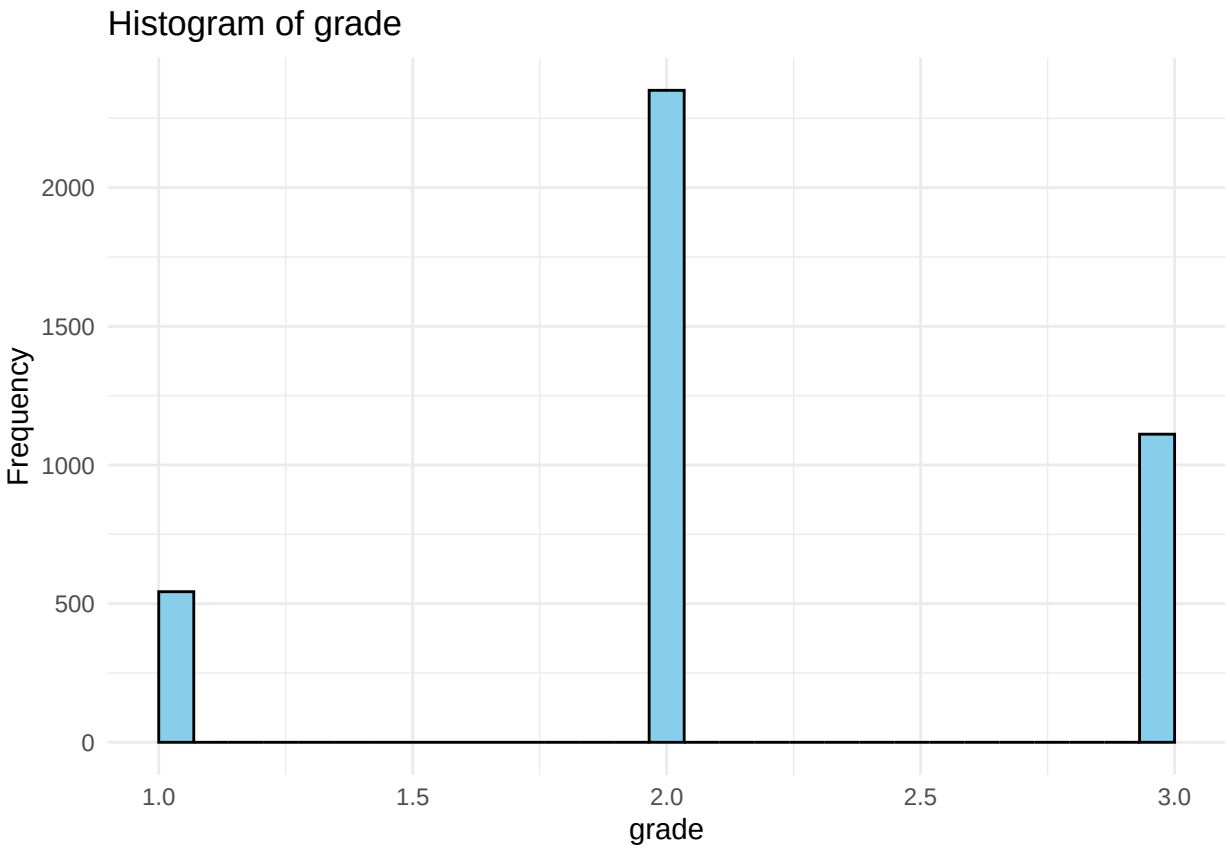

Histogram of t_stage
Frequency
1500
1000
500
0
1
2
3
4
t_stage


Boxplot of t_stage
t_stage
4
3
2
1
−0.4
−0.2
0.0
0.2
0.4

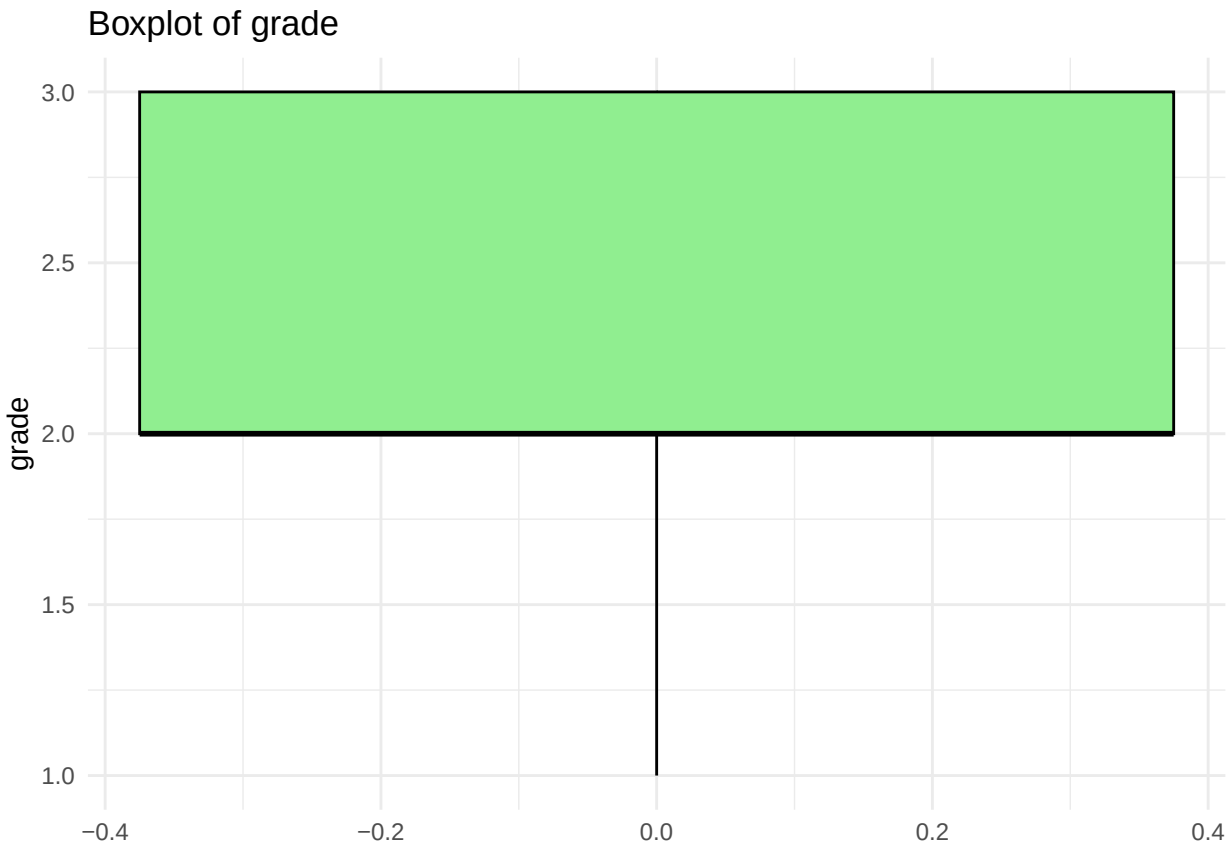

Histogram of n_stage
Frequency
2000
1000
0
1.0
1.5
2.0
2.5
3.0
n_stage

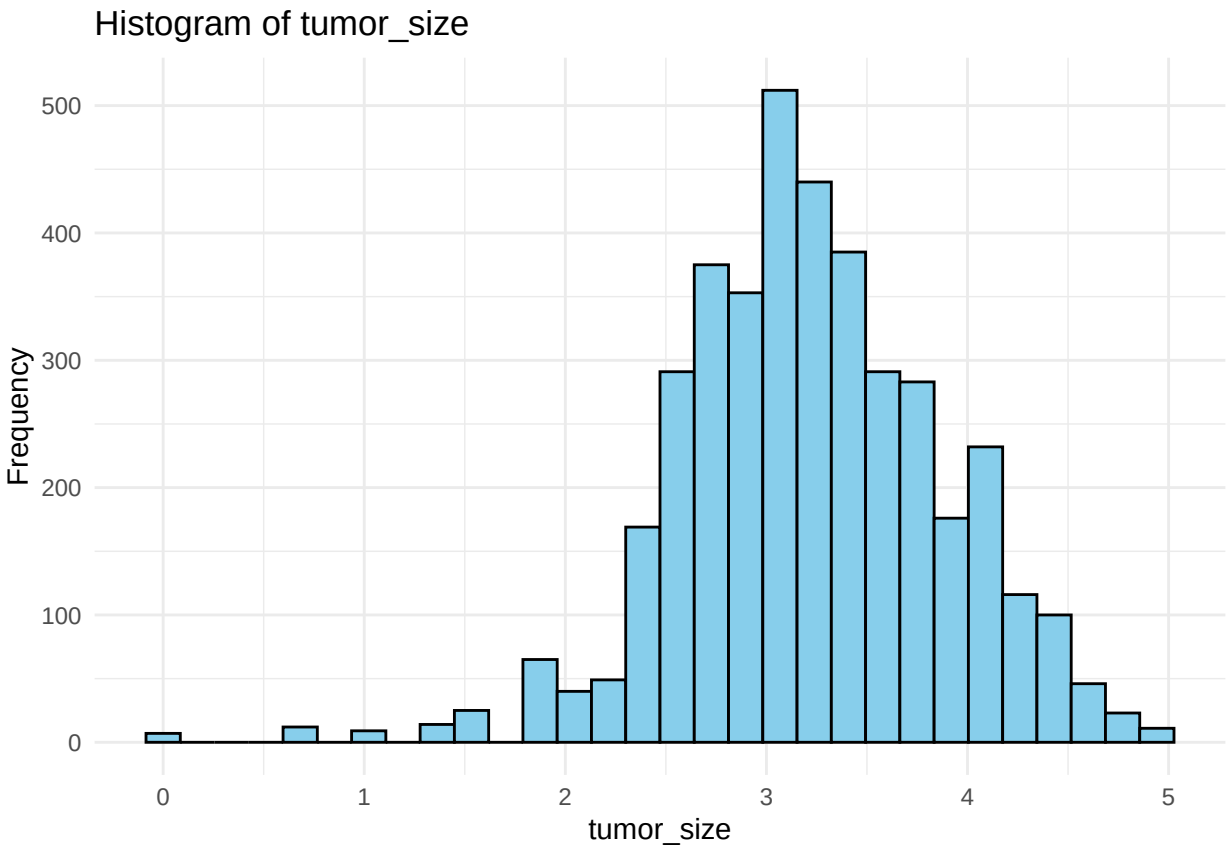


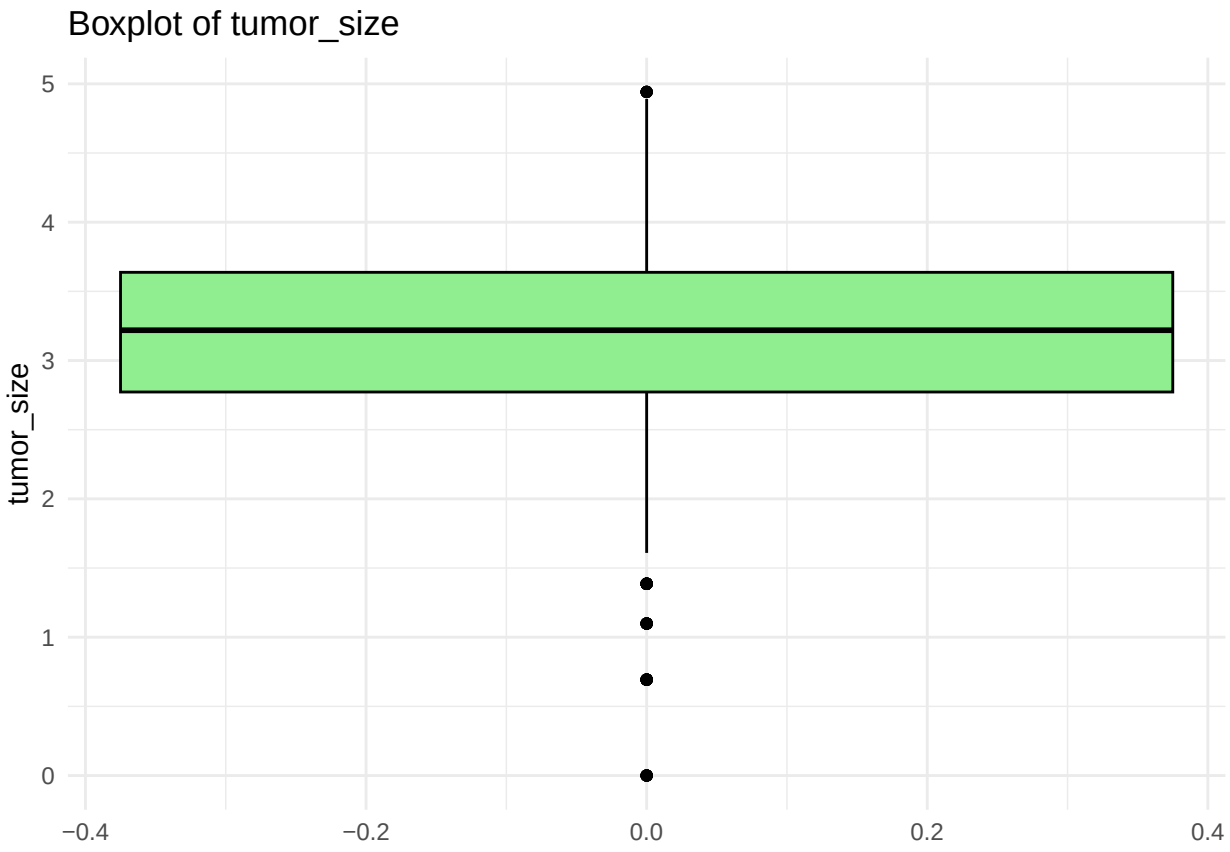

```
## Warning: Removed 19 rows containing non-finite outside the scale range
## (‘stat_bin()‘).
```

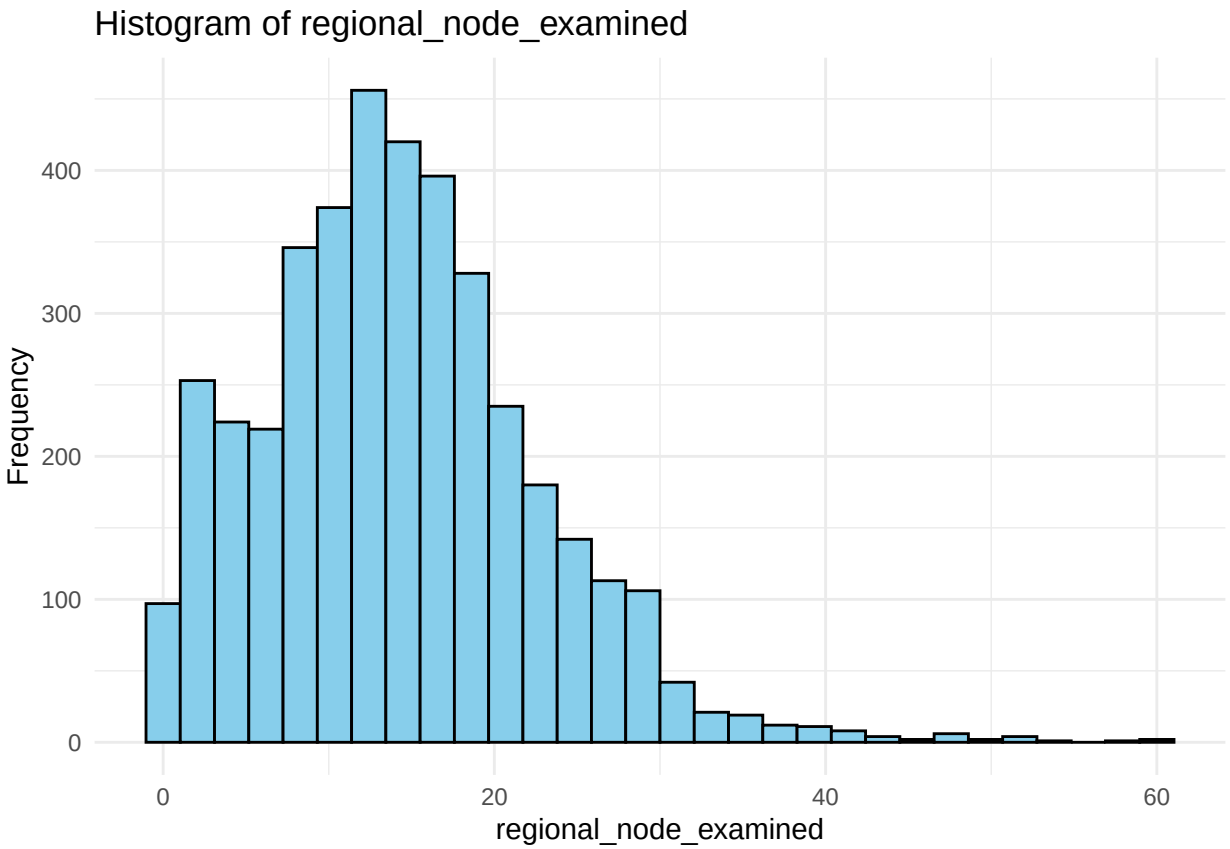


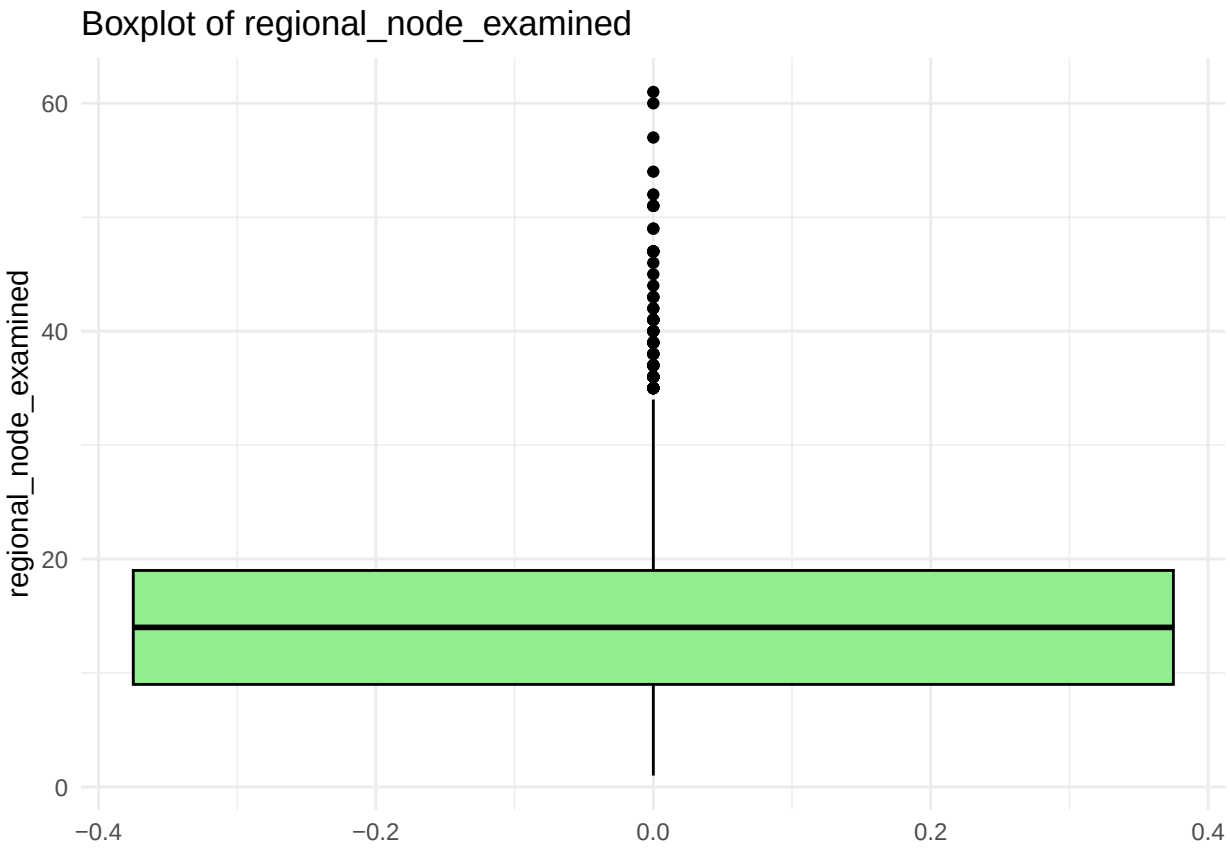

```
## Warning: Removed 19 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```

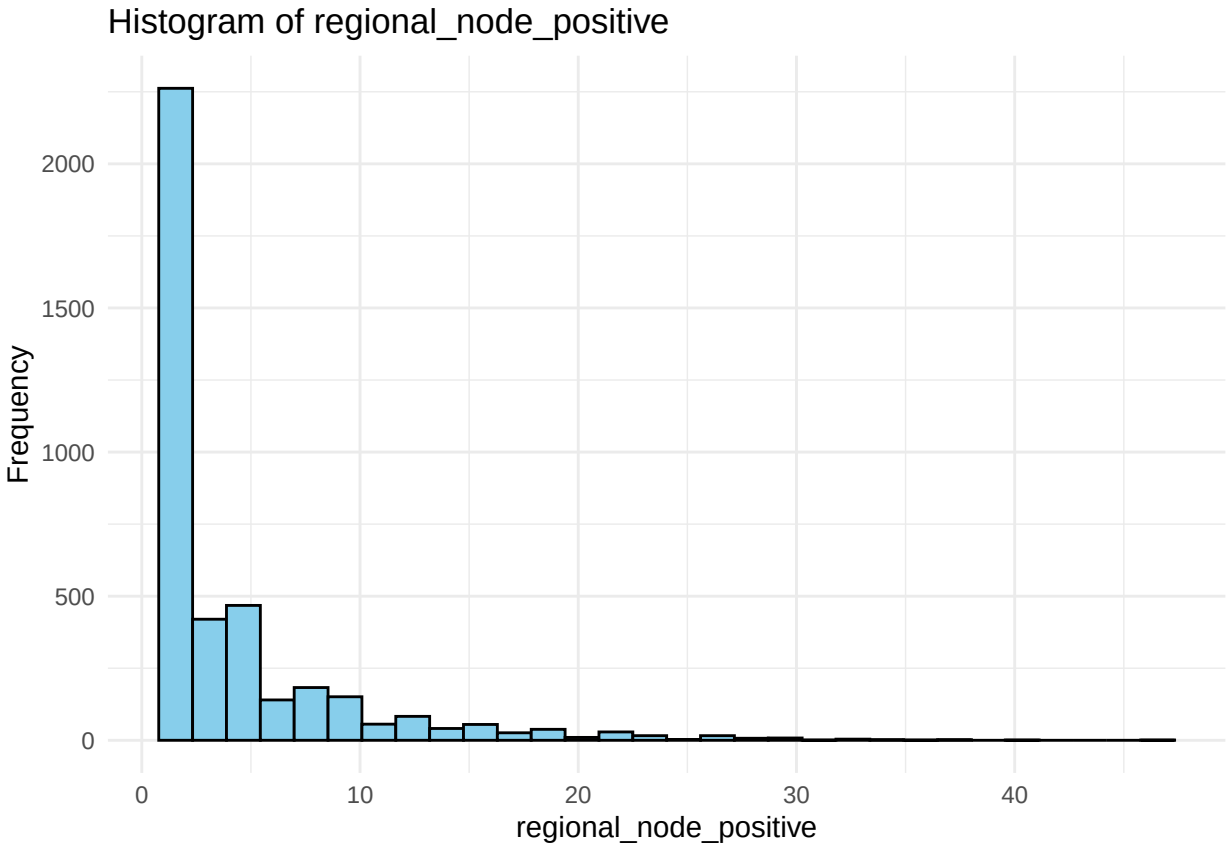

Boxplot of grade
grade
3.0
2.5
2.0
1.5
1.0
−0.4
−0.2
0.0
0.2
0.4

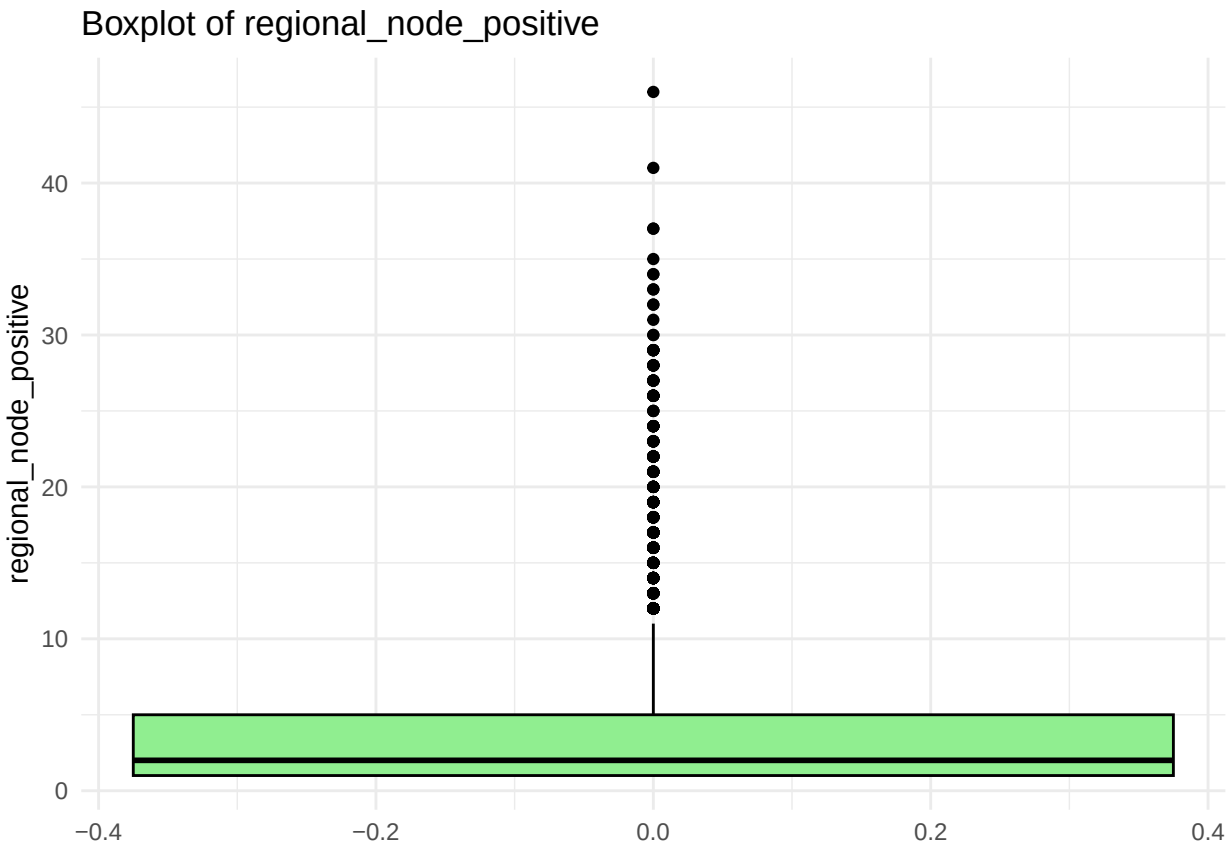

Histogram of tumor_size
Frequency
500
400
300
200
100
0
0
1
2
3
4
5
tumor_size

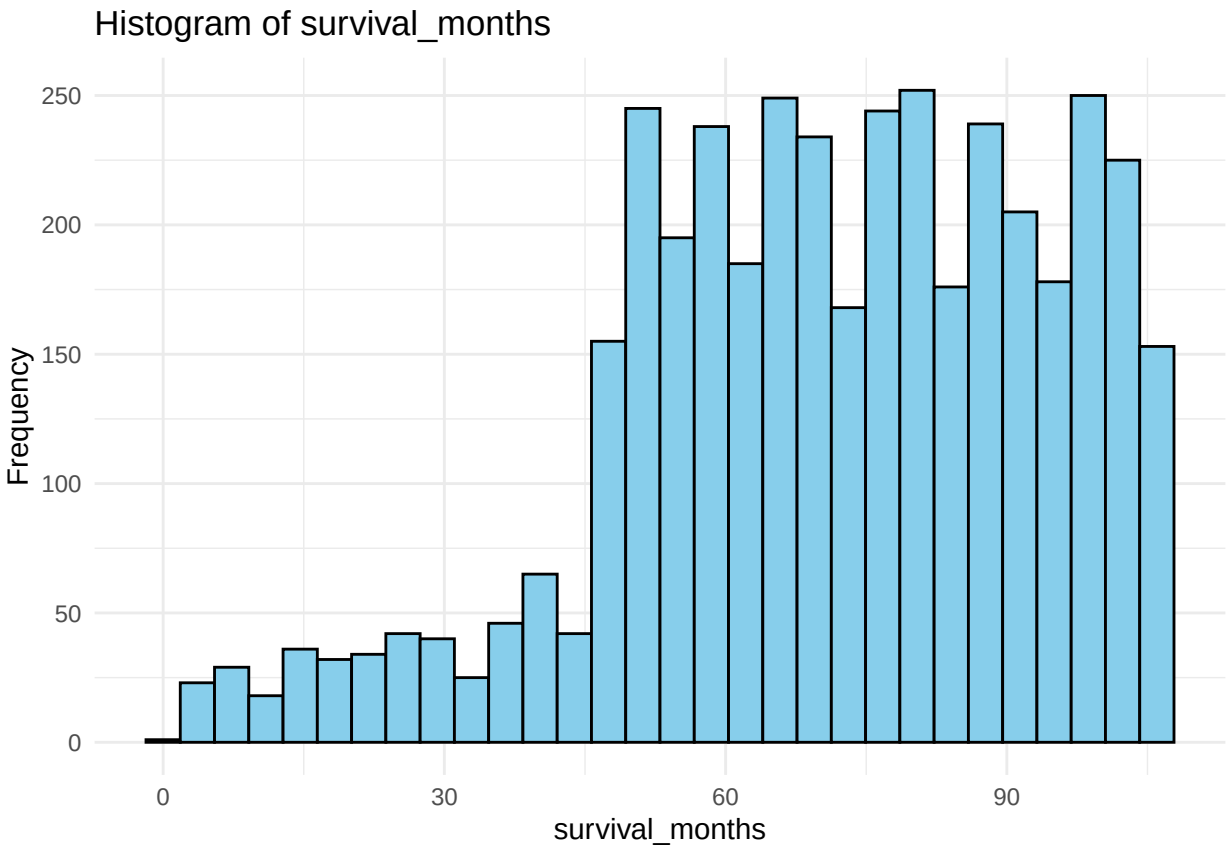

Boxplot of tumor_size
tumor_size
5
4
3
2
1
0
−0.4
−0.2
0.0
0.2
0.4

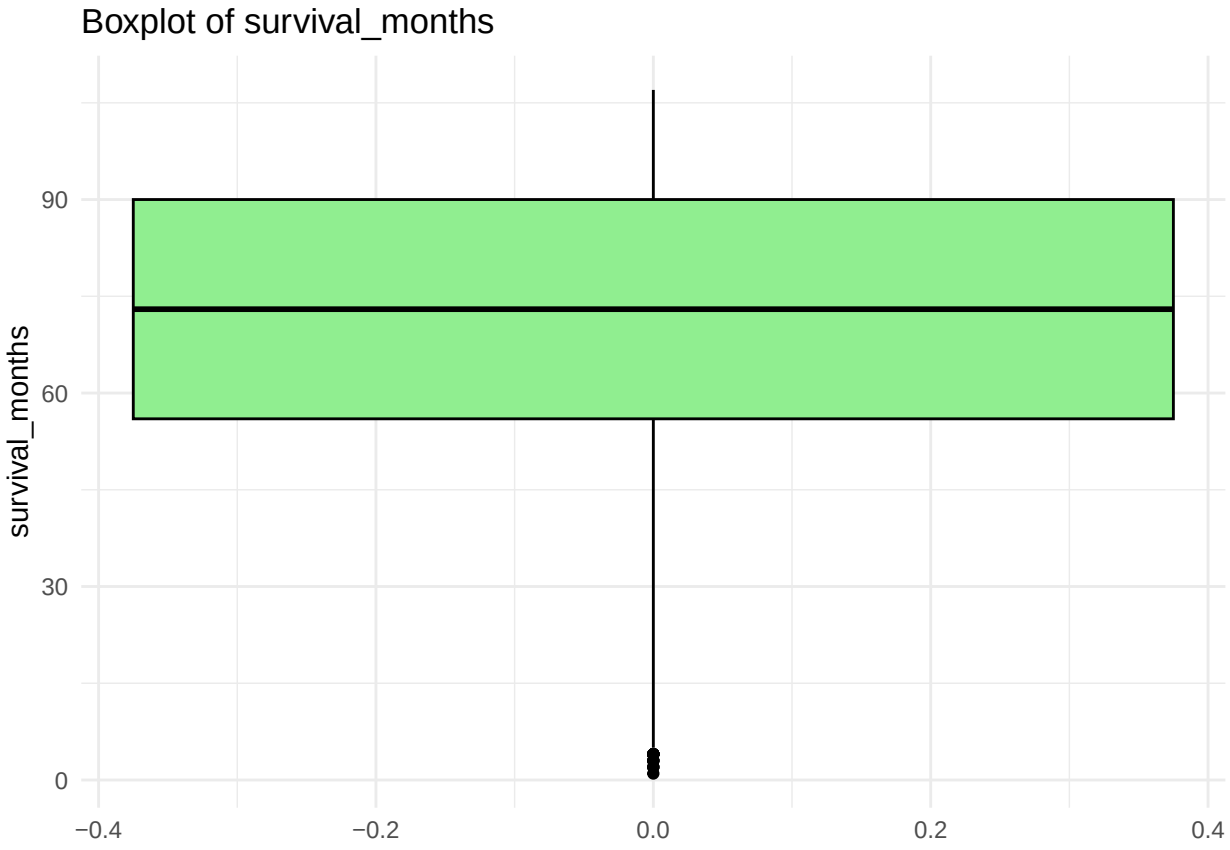

Histogram of regional_node_examined
Frequency
400
300
200
100
0
0
20
40
60
regional_node_examined

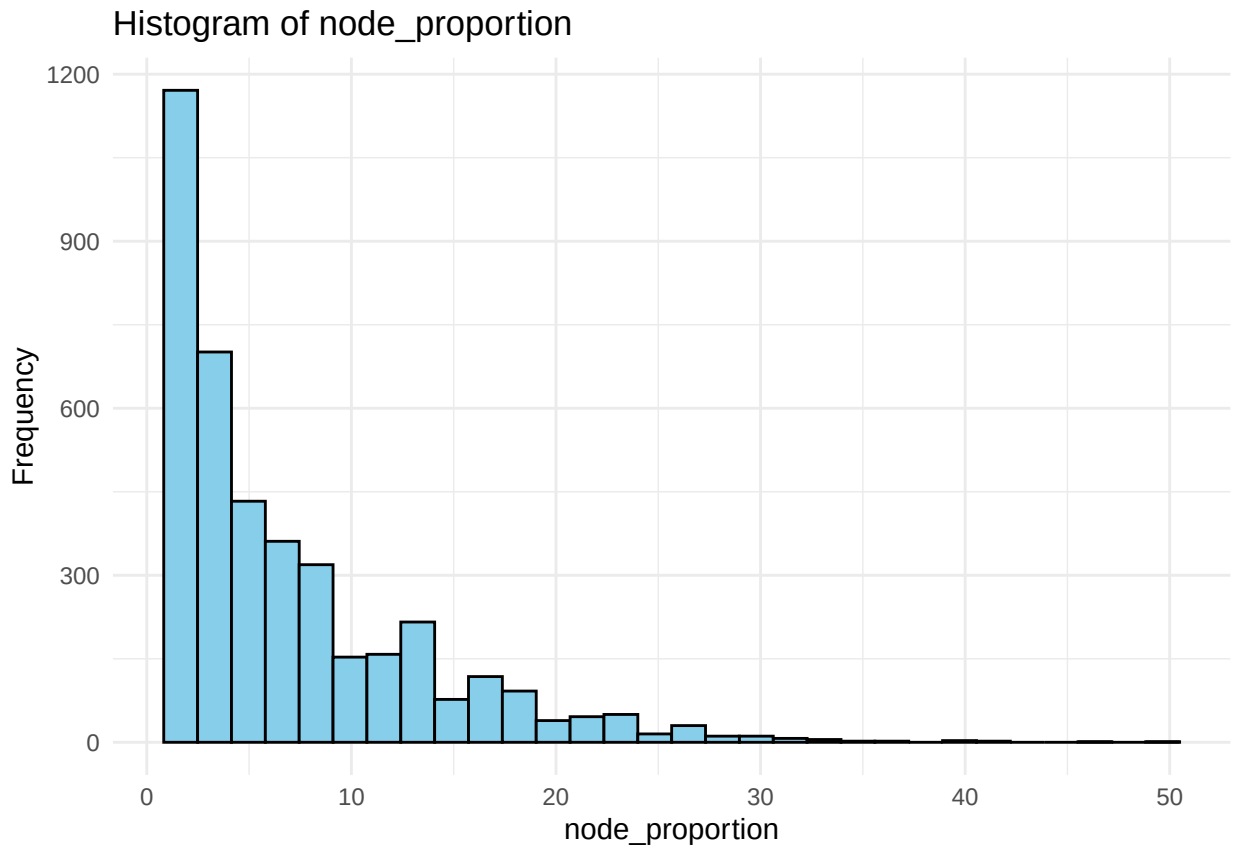

Boxplot of regional_node_examined
regional_node_examined
60
40
20
0
−0.4
−0.2
0.0
0.2
0.4

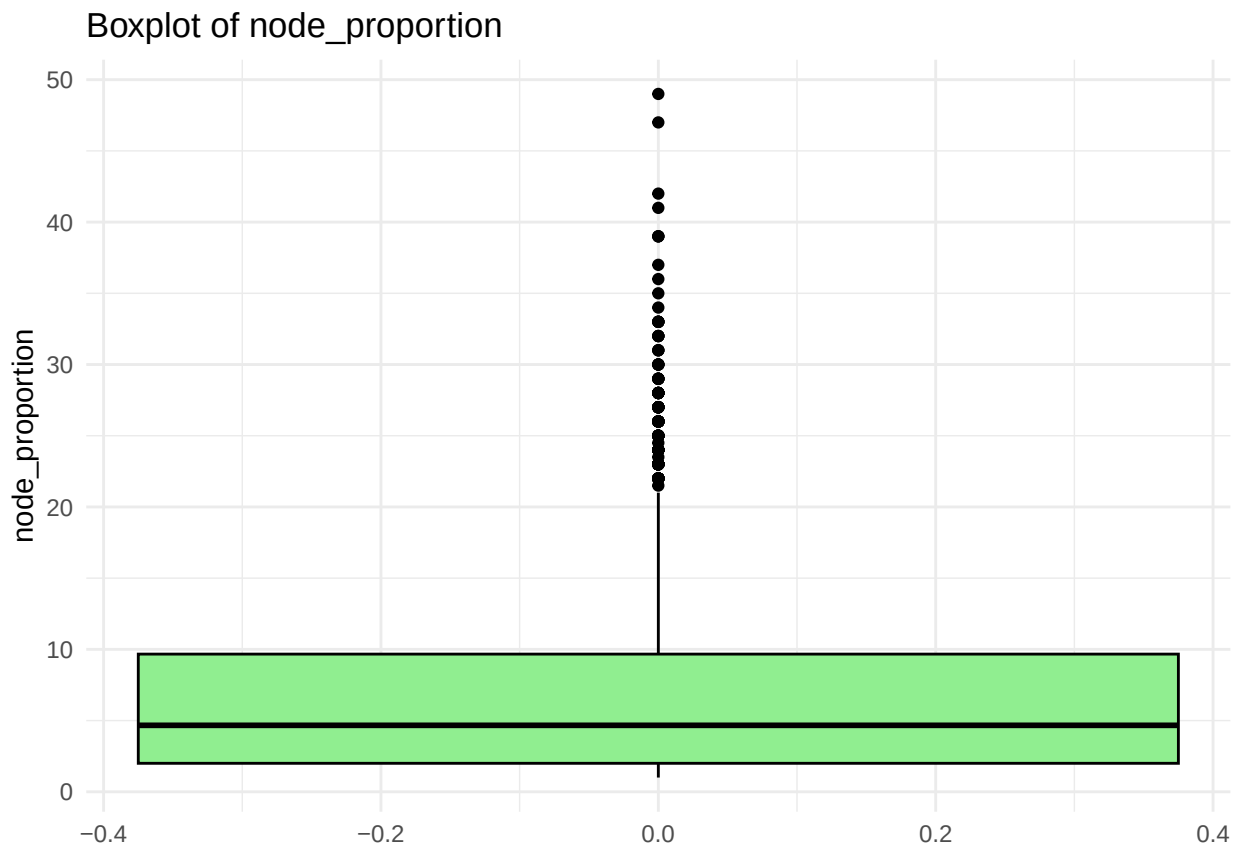

Histogram of regional_node_positive
Frequency
2000
1500
1000
500
0
0
10
20
30
40
regional_node_positive


Boxplot of regional_node_positive
regional_node_positive
40
30
20
10
0
−0.4
−0.2
0.0
0.2
0.4


Histogram of survival_months
Frequency
250
200
150
100
50
0
0
30
60
90
survival_months


Boxplot of survival_months
survival_months
90
60
30
0
−0.4
−0.2
0.0
0.2
0.4

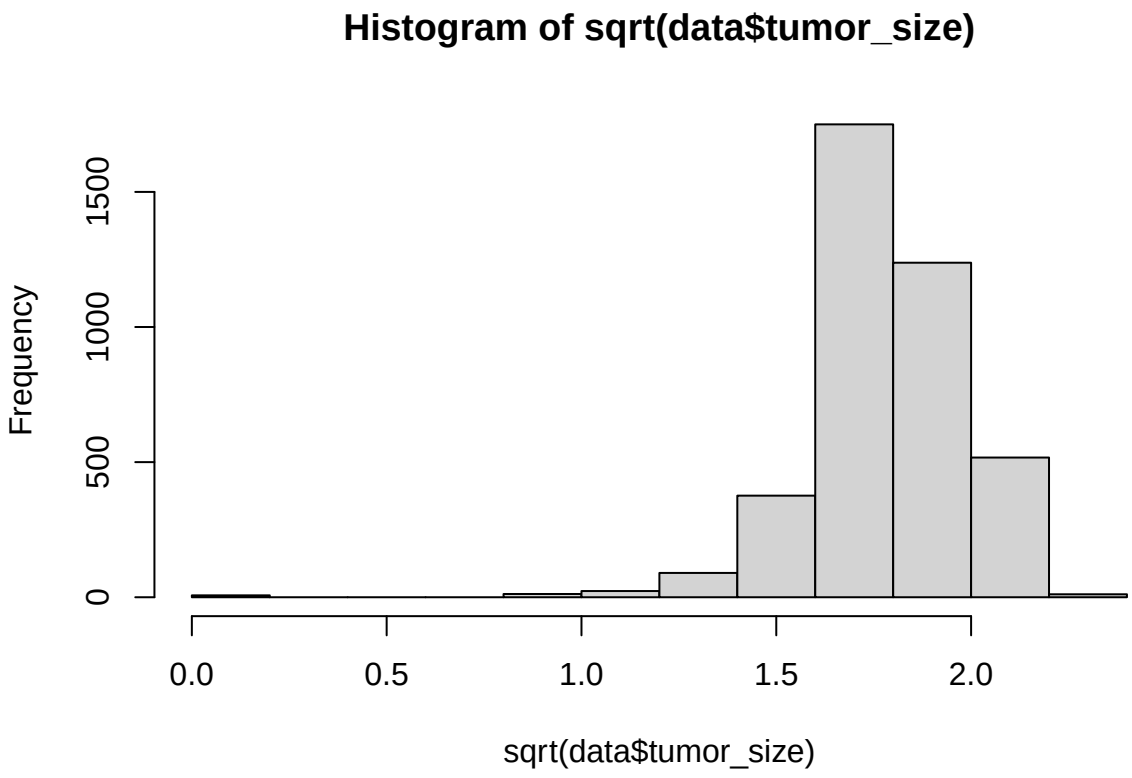

Histogram of node_proportion
Frequency
1200
900
600
300
0
0
10
20
30
40
50
node_proportion



skewness of Tumor.size

```
hist(log(data$tumor_size))
```


```r
hist(sqrt(data$tumor_size))
```



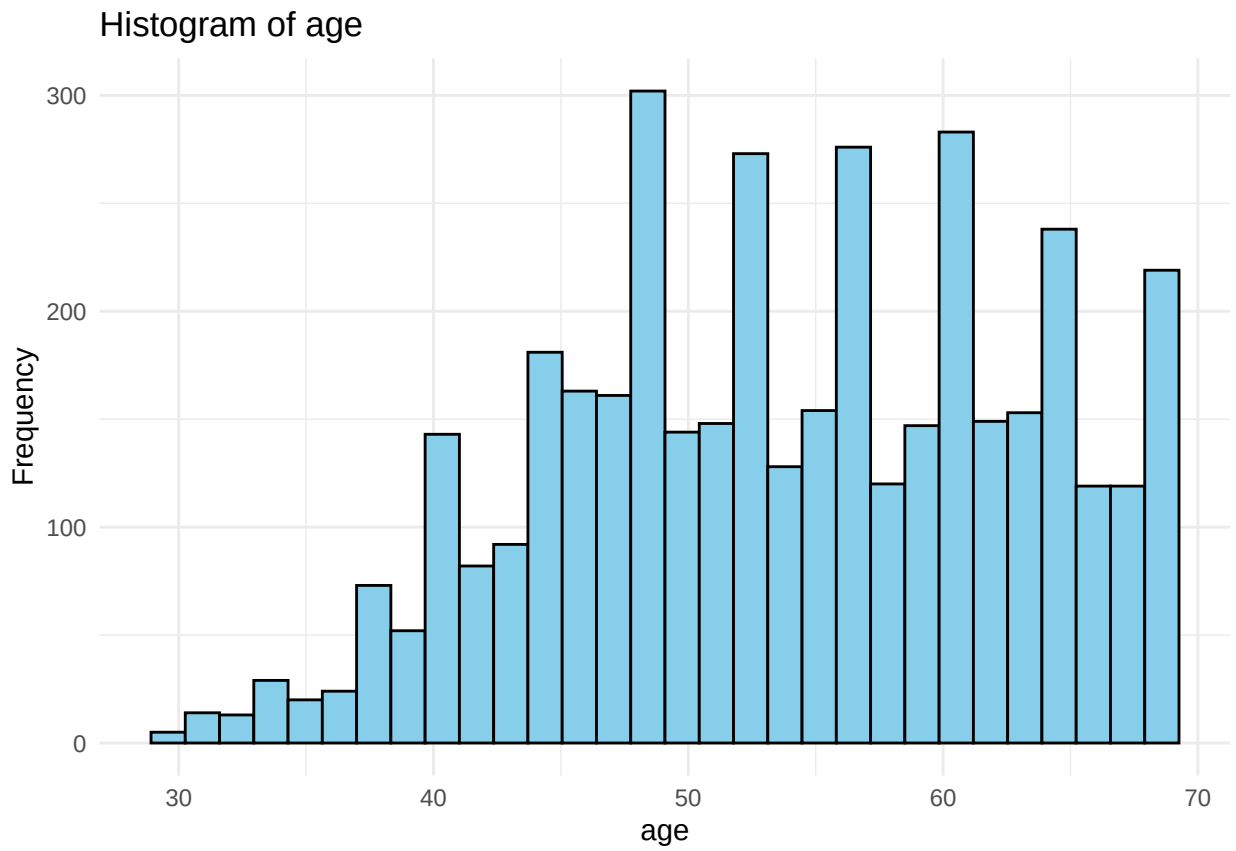
log transformation for tumor.size

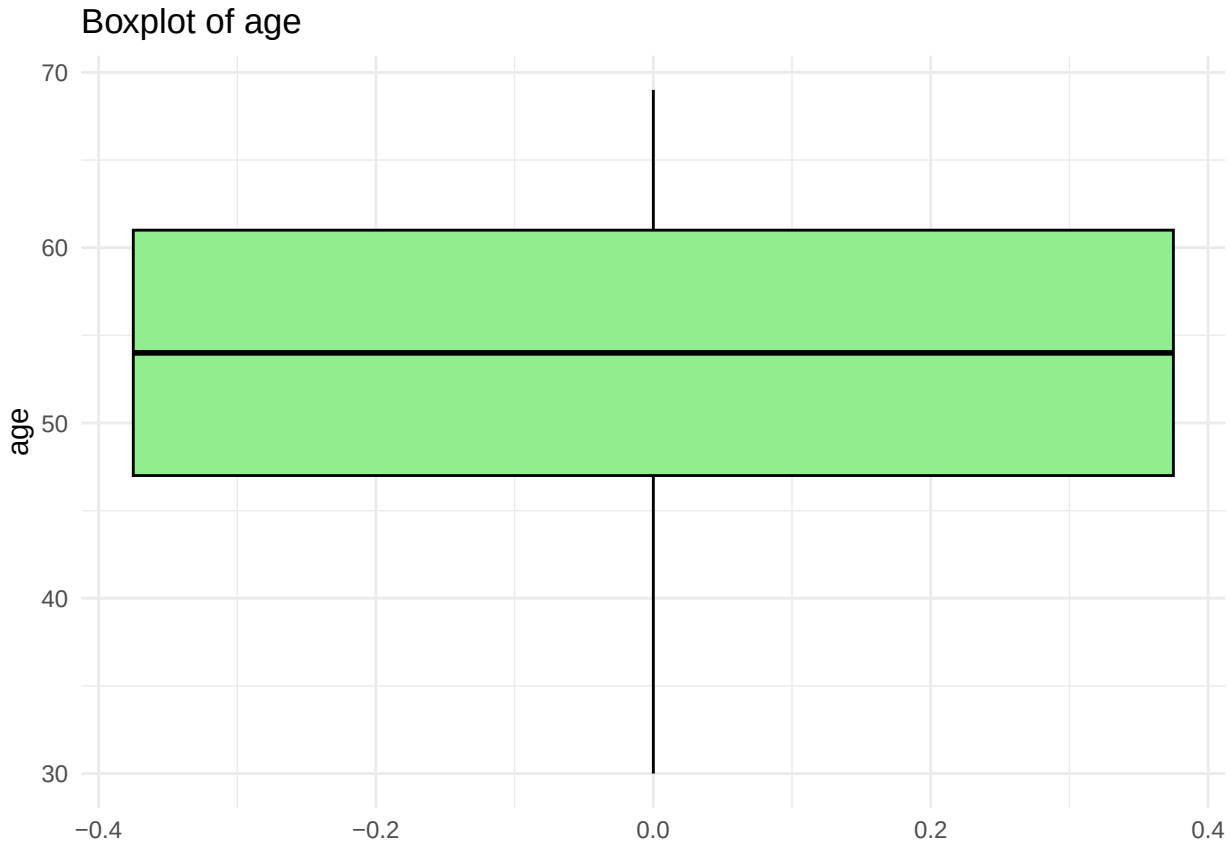
```
data = data %>%
  mutate(tumor_size = log(tumor_size))
```

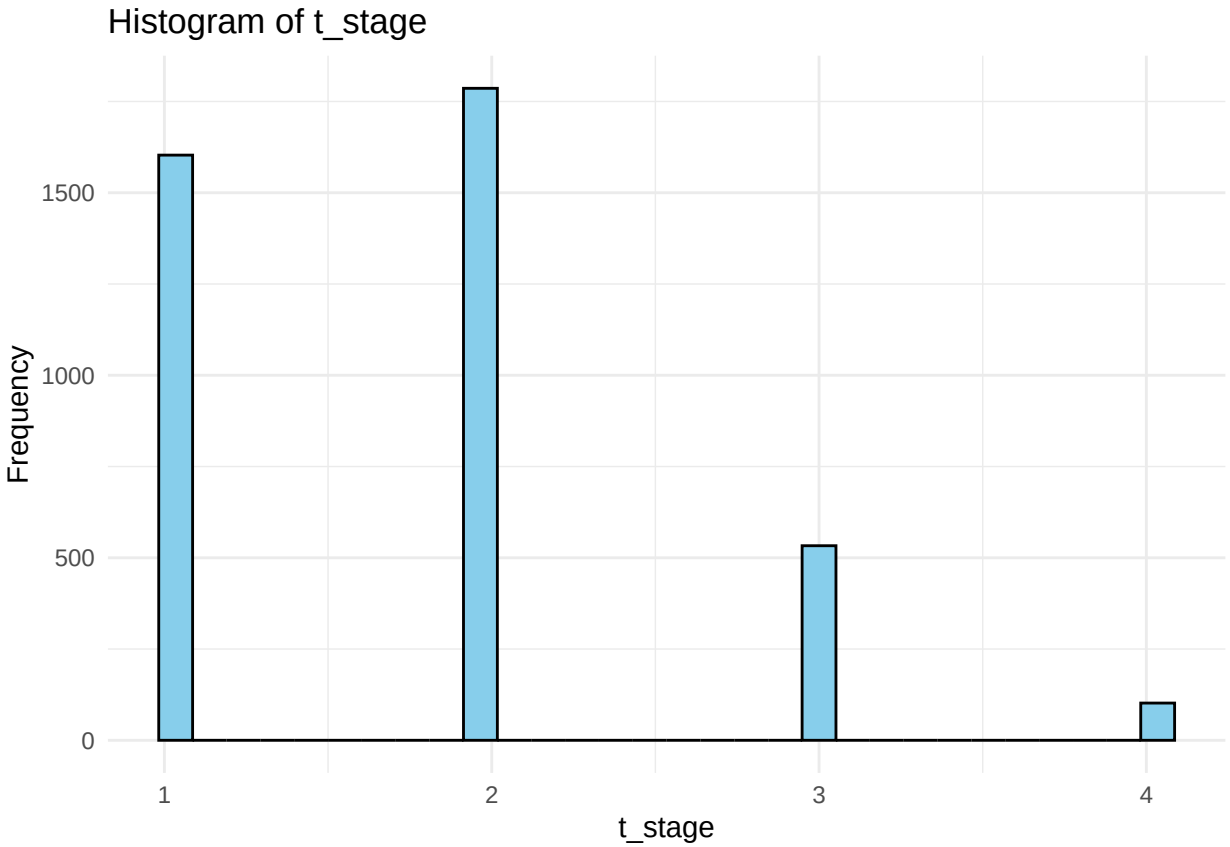
see the plot after the third preprocess

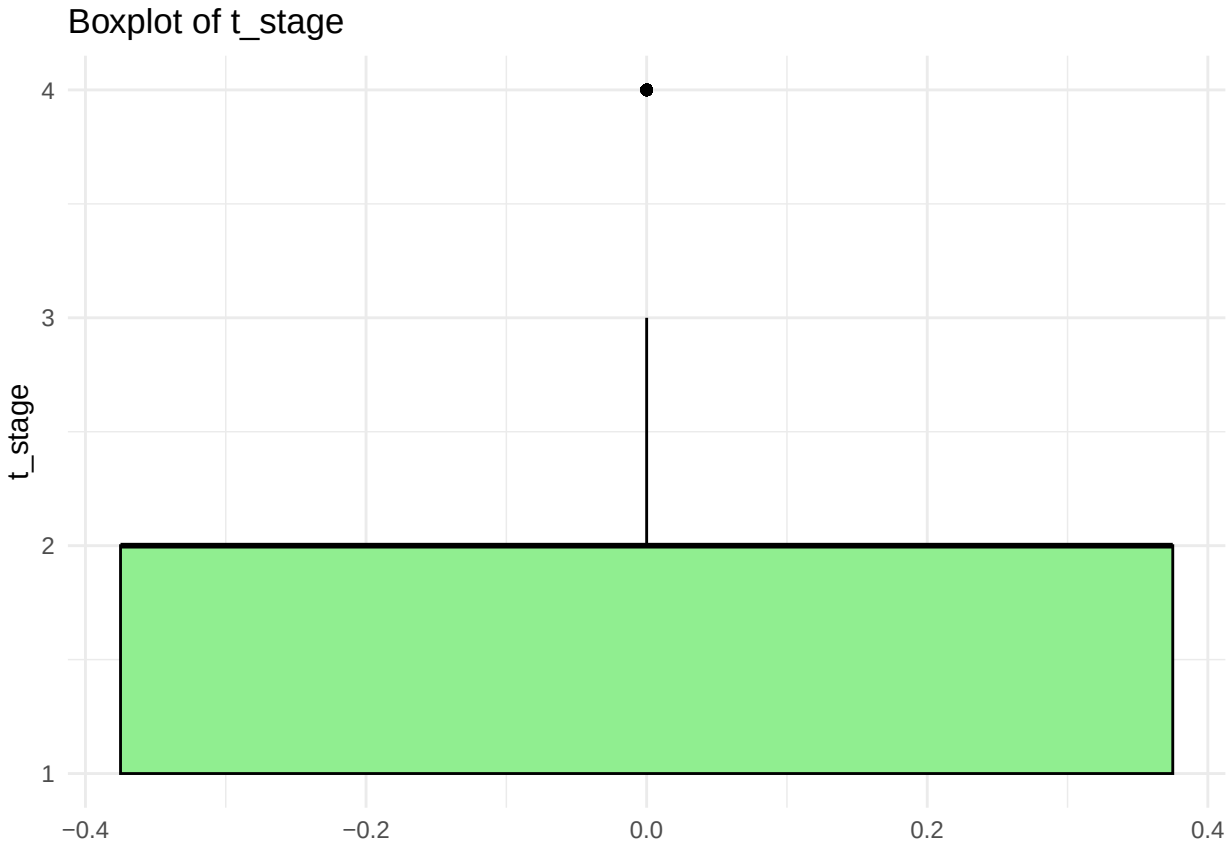
```
for (var in names(data)) {
  # Skip if the column is not numeric
  if (is.numeric(data[[var]])) {

    # Histogram
    p1 <- ggplot(data, aes_string(x = var)) +
      geom_histogram(fill = "skyblue", color = "black", bins = 30) +
      labs(title = paste("Histogram of", var), x = var, y = "Frequency") +
      theme_minimal()
    print(p1)

    # Boxplot
    p2 <- ggplot(data, aes_string(y = var)) +
      geom_boxplot(fill = "lightgreen", color = "black") +
      labs(title = paste("Boxplot of", var), y = var) +
```

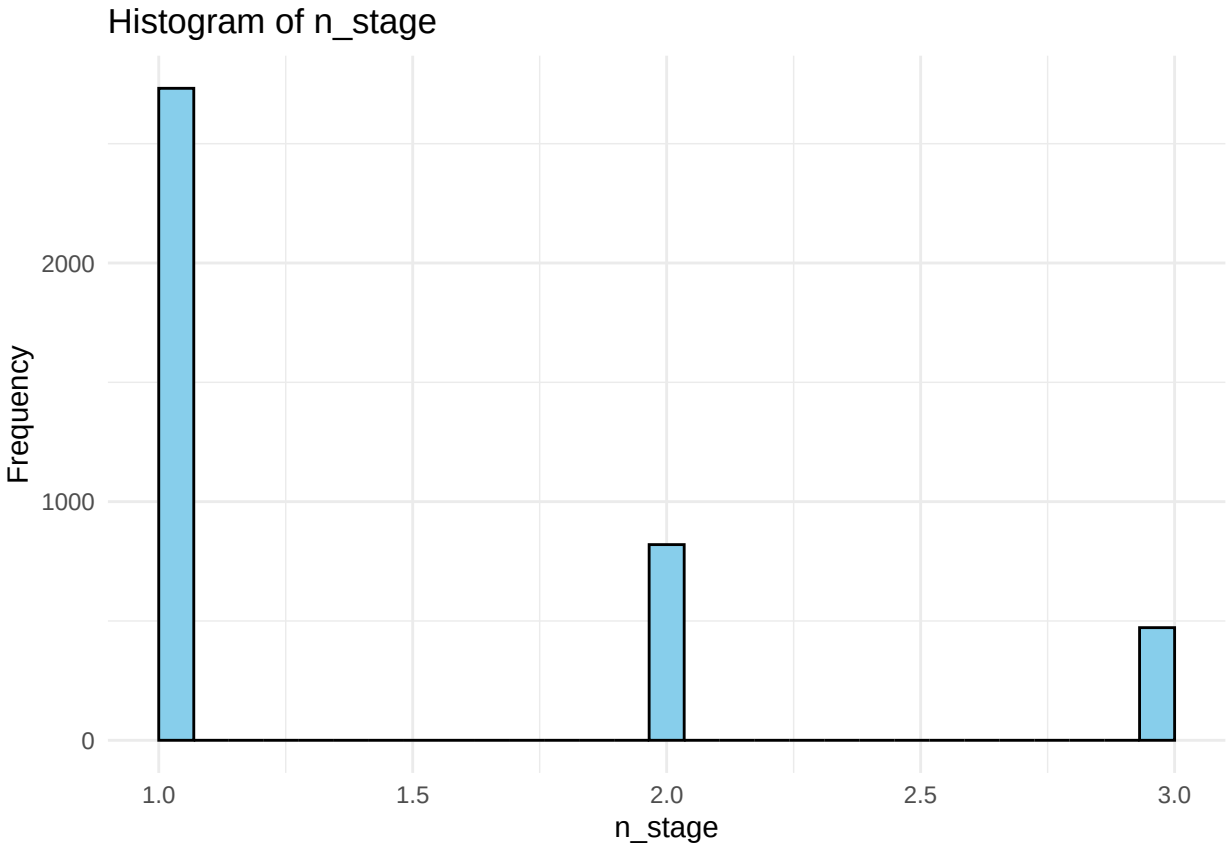
```
      theme_minimal()
    print(p2)
  }
}
```

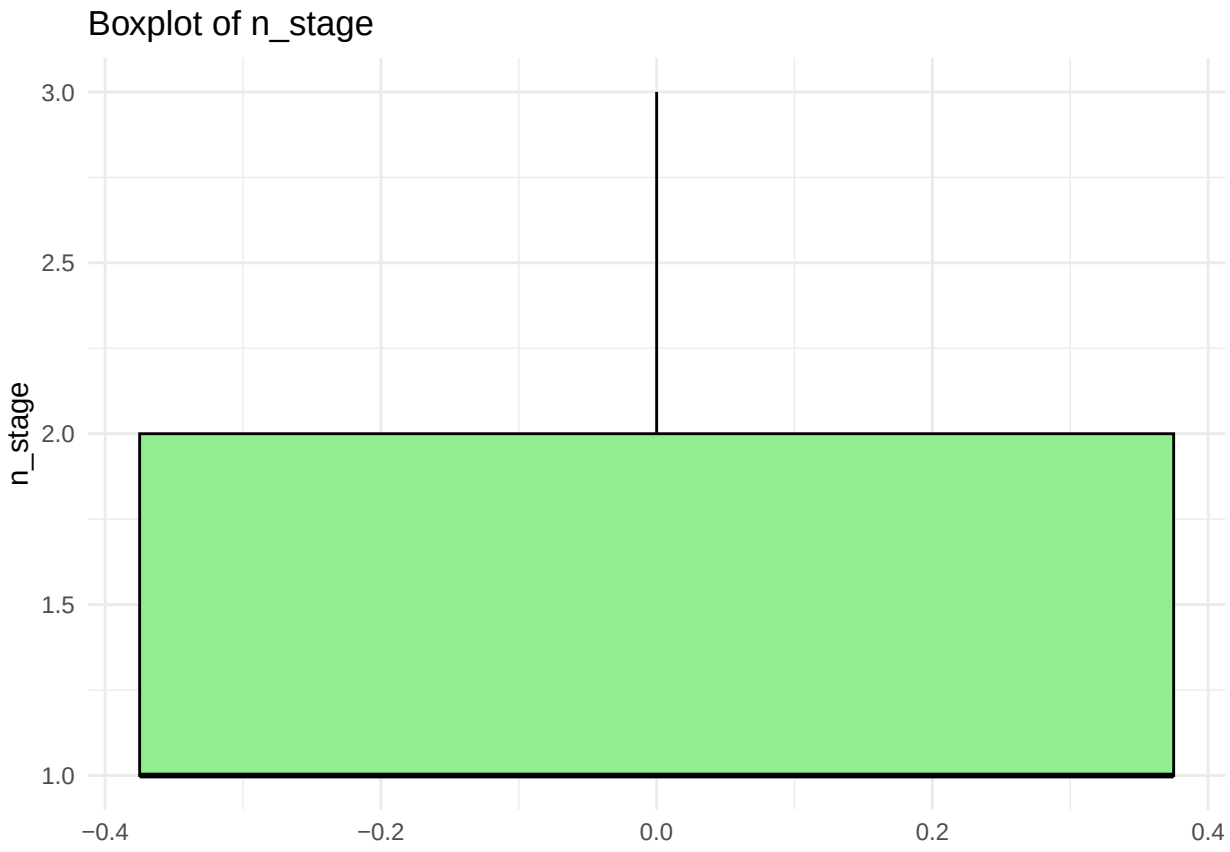



Boxplot of age
age
70
60
50
40
30
−0.4
−0.2
0.0
0.2
0.4

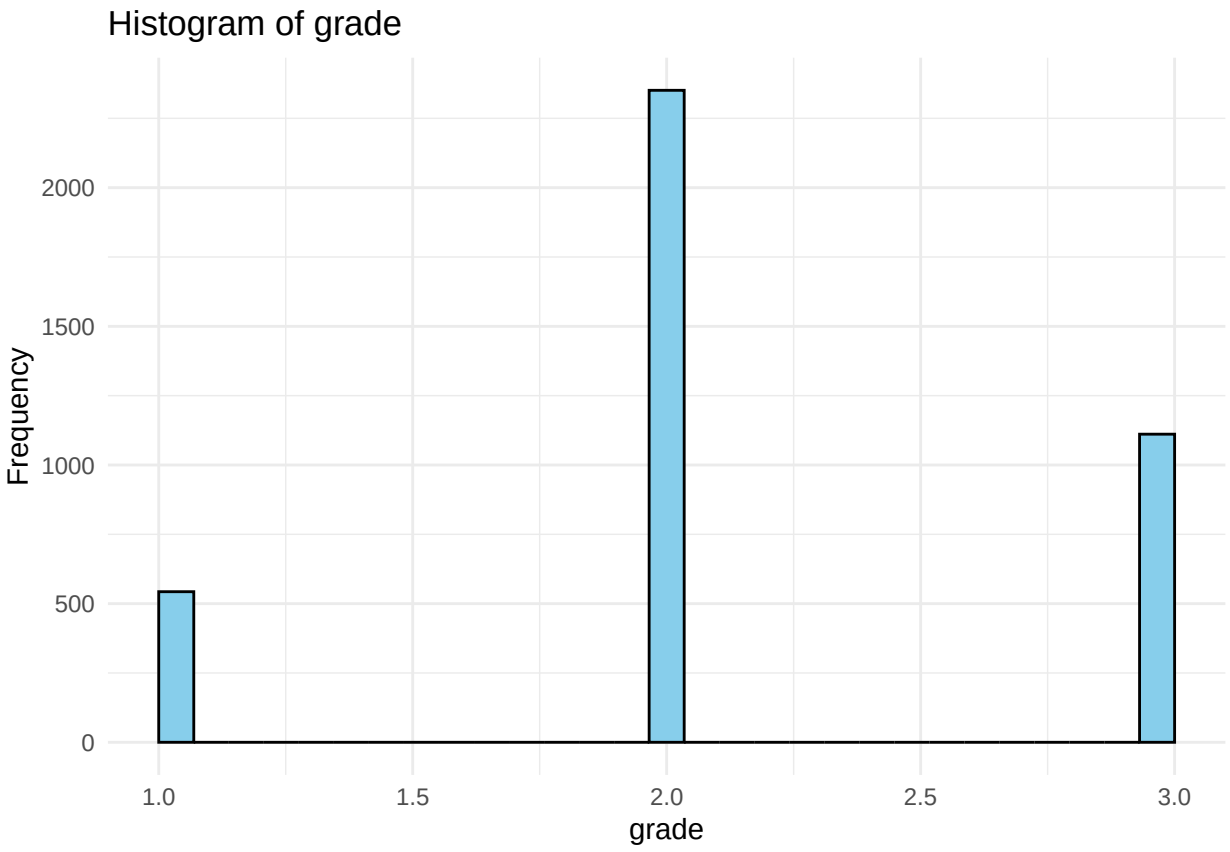

Histogram of t_stage
Frequency
1500
1000
500
0
1
2
3
4
t_stage


Boxplot of t_stage
t_stage
4
3
2
1
−0.4
−0.2
0.0
0.2
0.4

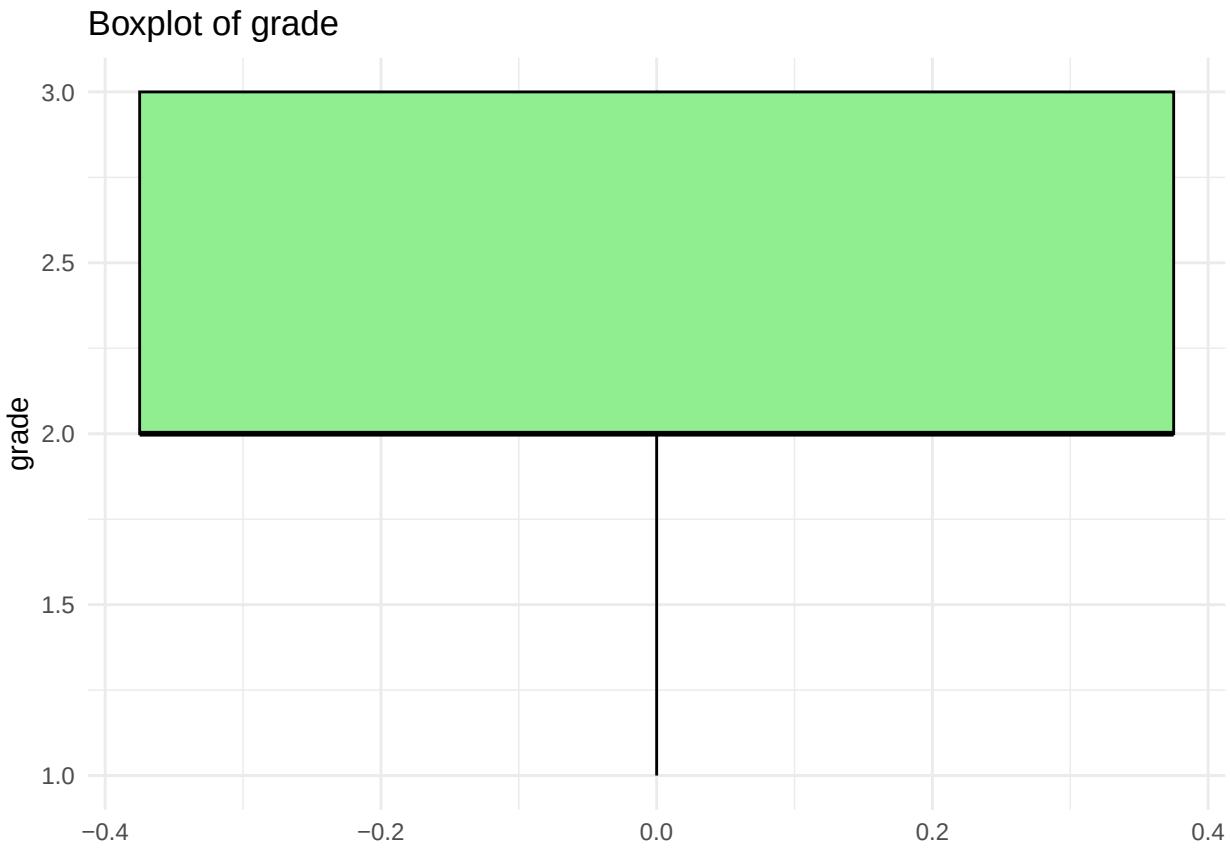

Histogram of n_stage
Frequency
2000
1000
0
1.0
1.5
2.0
2.5
3.0
n_stage



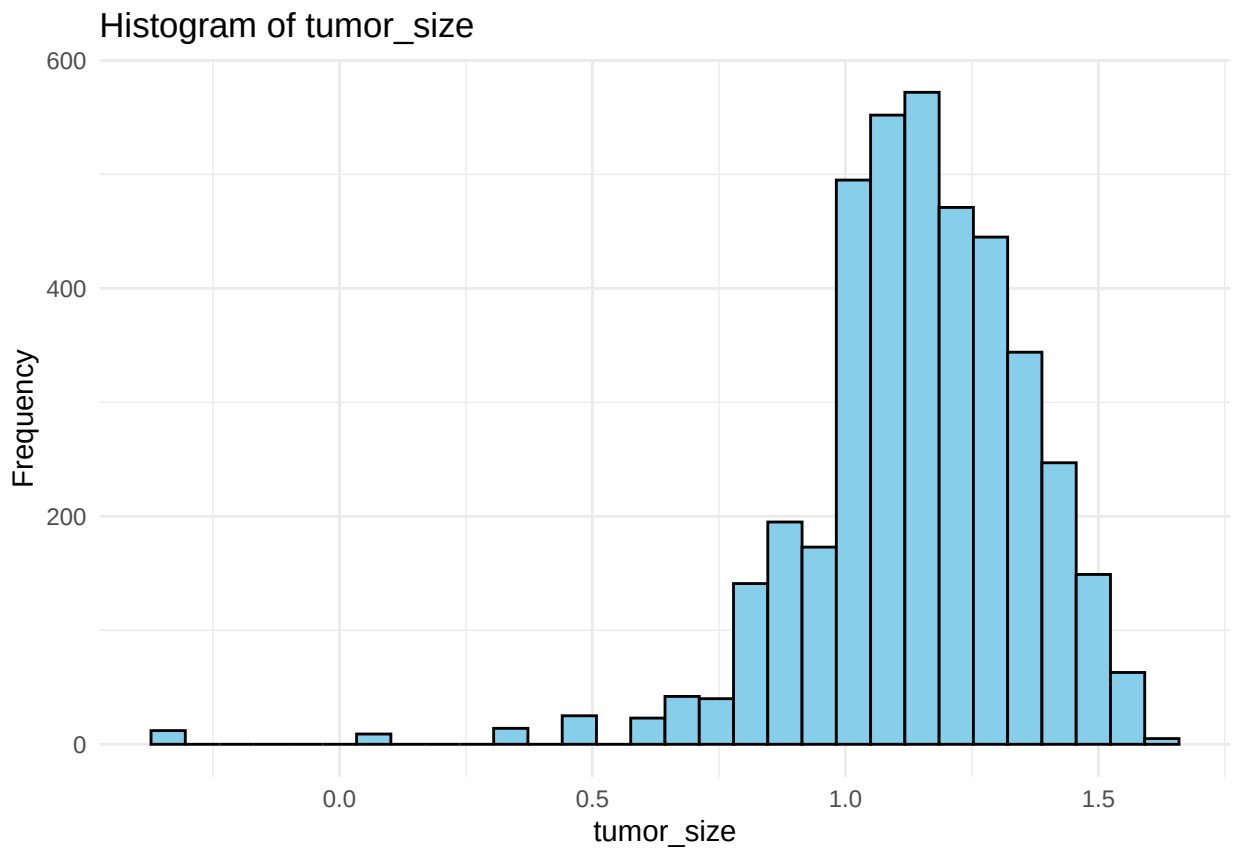

```
## Warning: Removed 19 rows containing non-finite outside the scale range
## (‘stat_bin()‘).
```



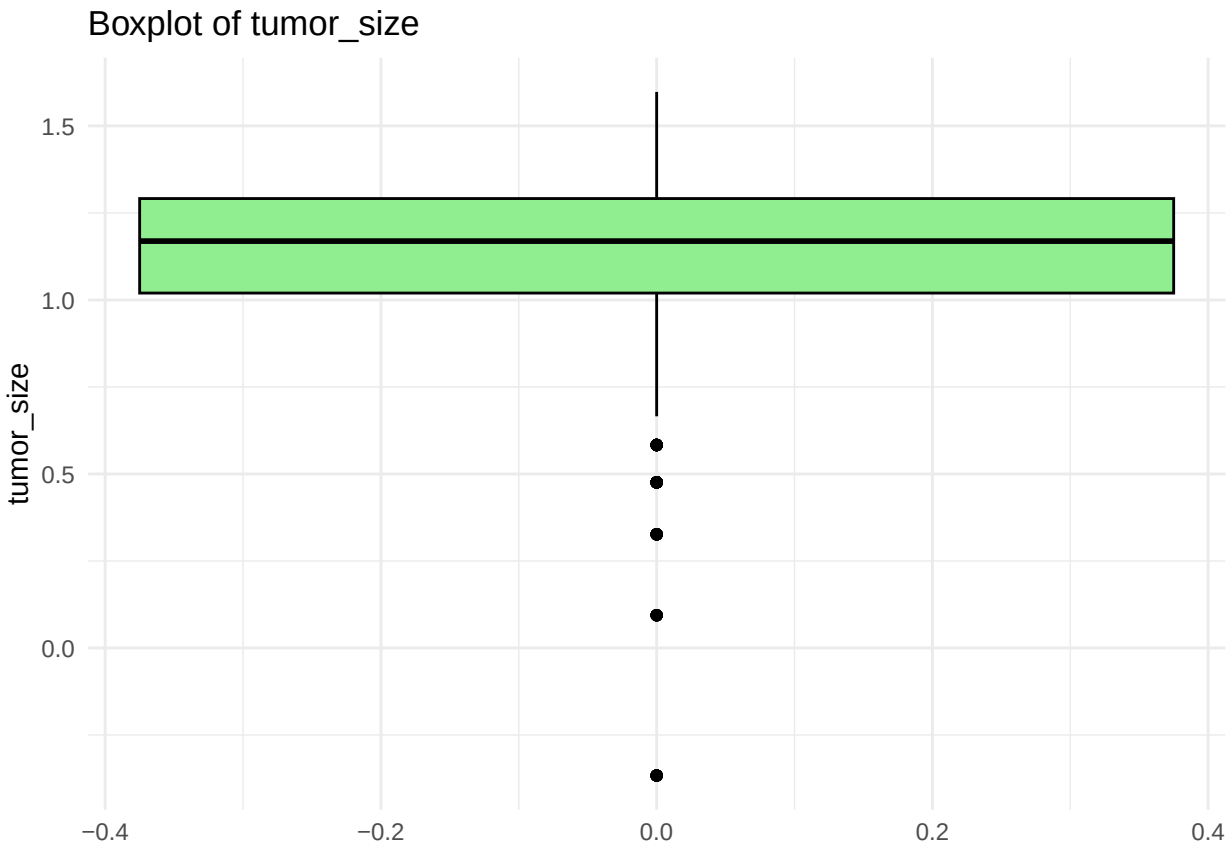

```
## Warning: Removed 19 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```

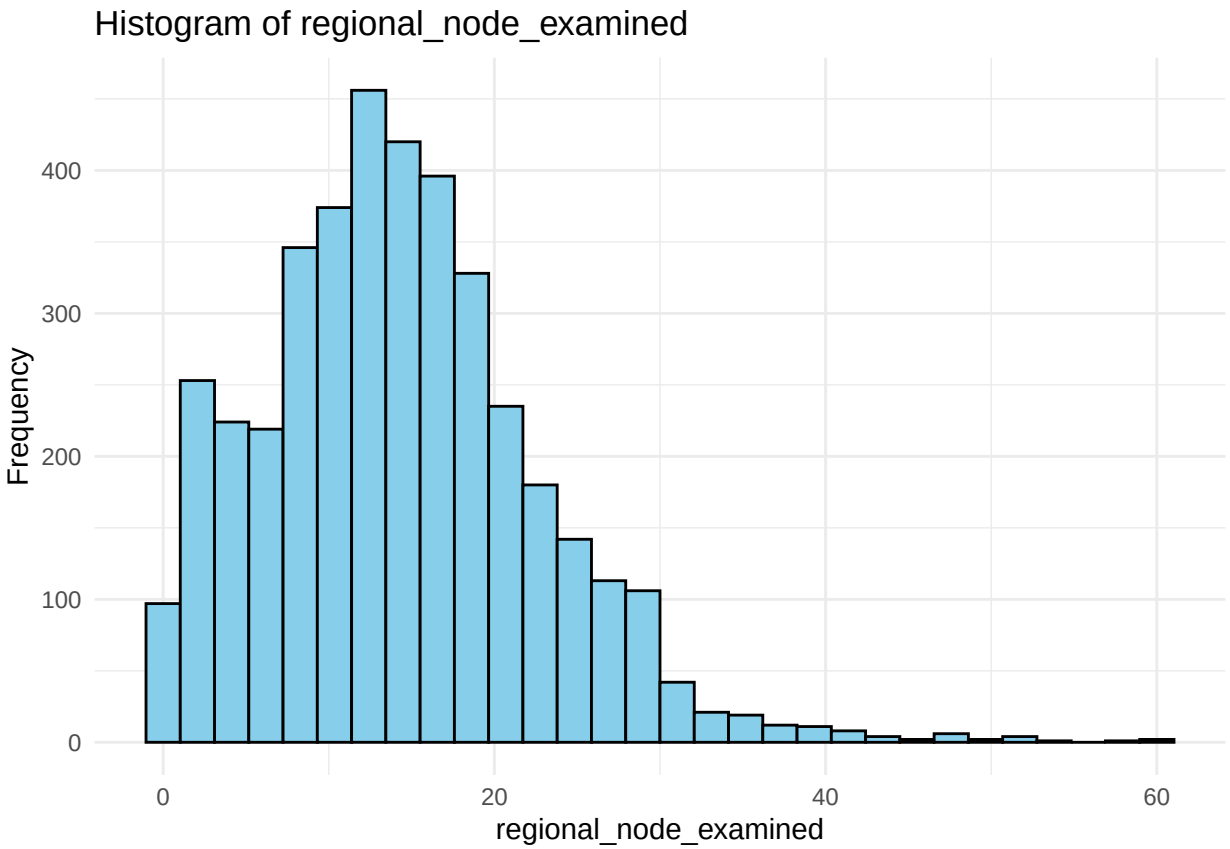


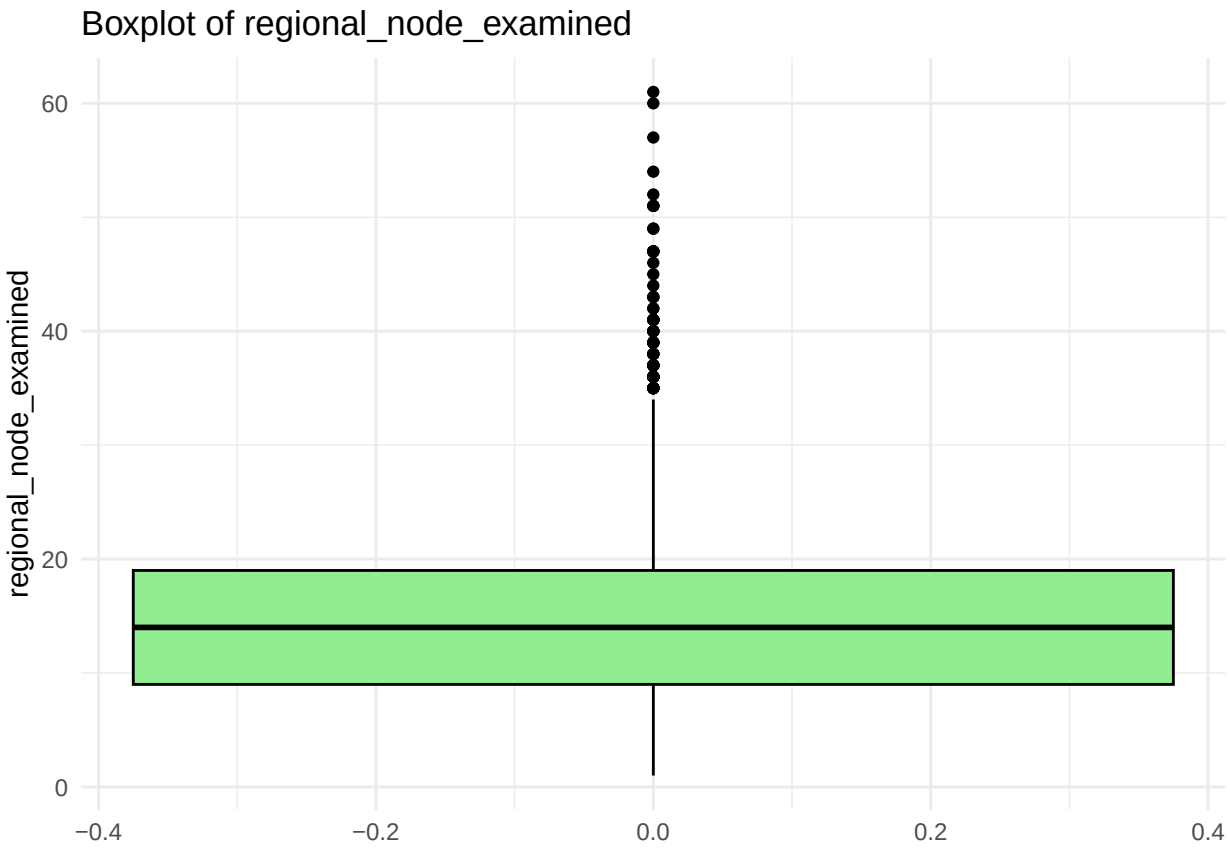

```
## Warning: Removed 7 rows containing non-finite outside the scale range
## (‘stat_bin()‘).
```

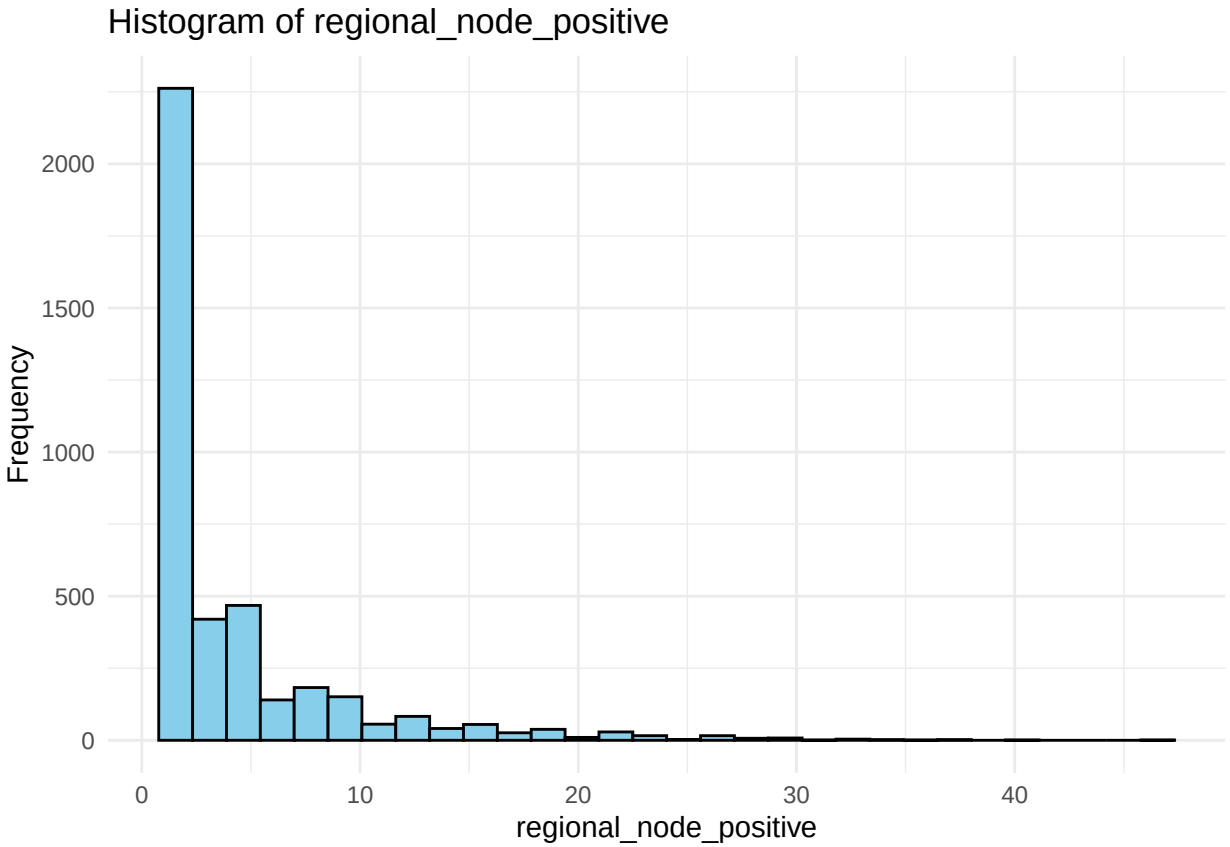


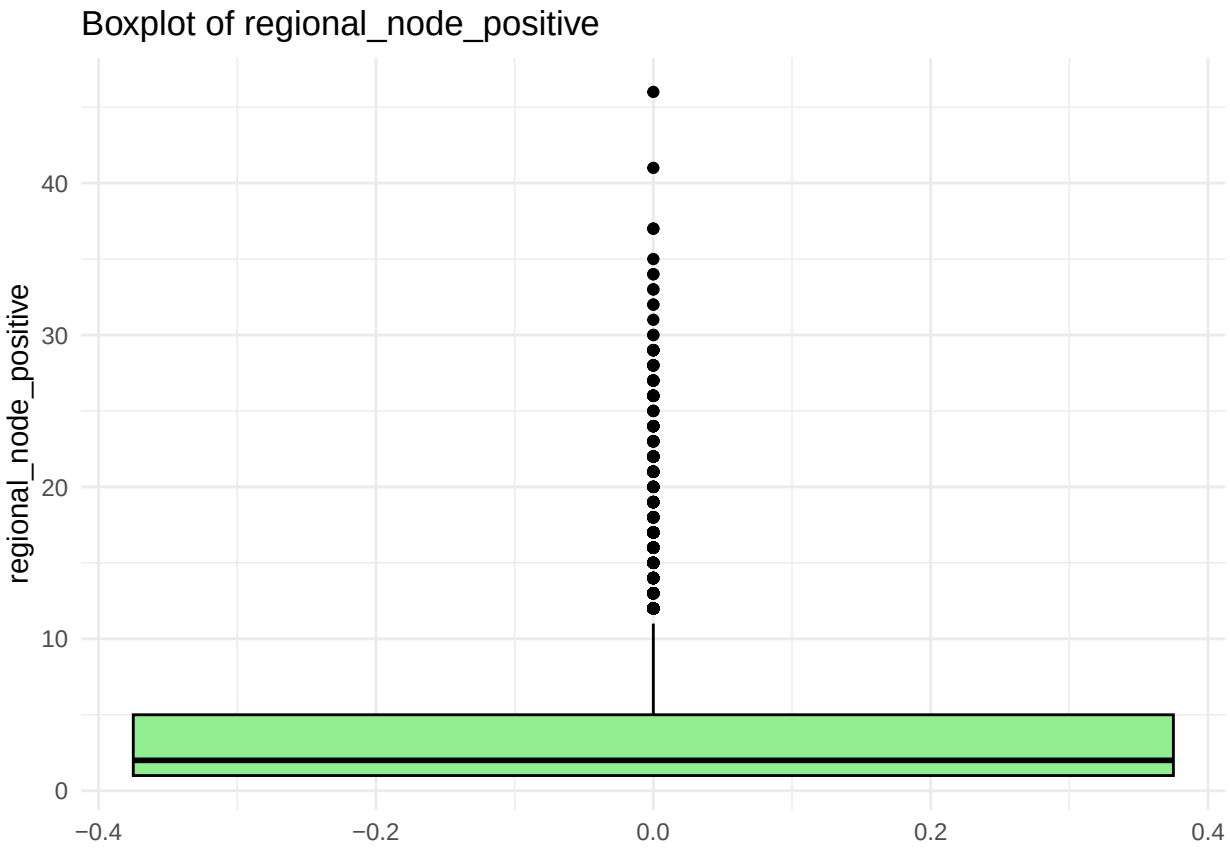

```
## Warning: Removed 7 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```

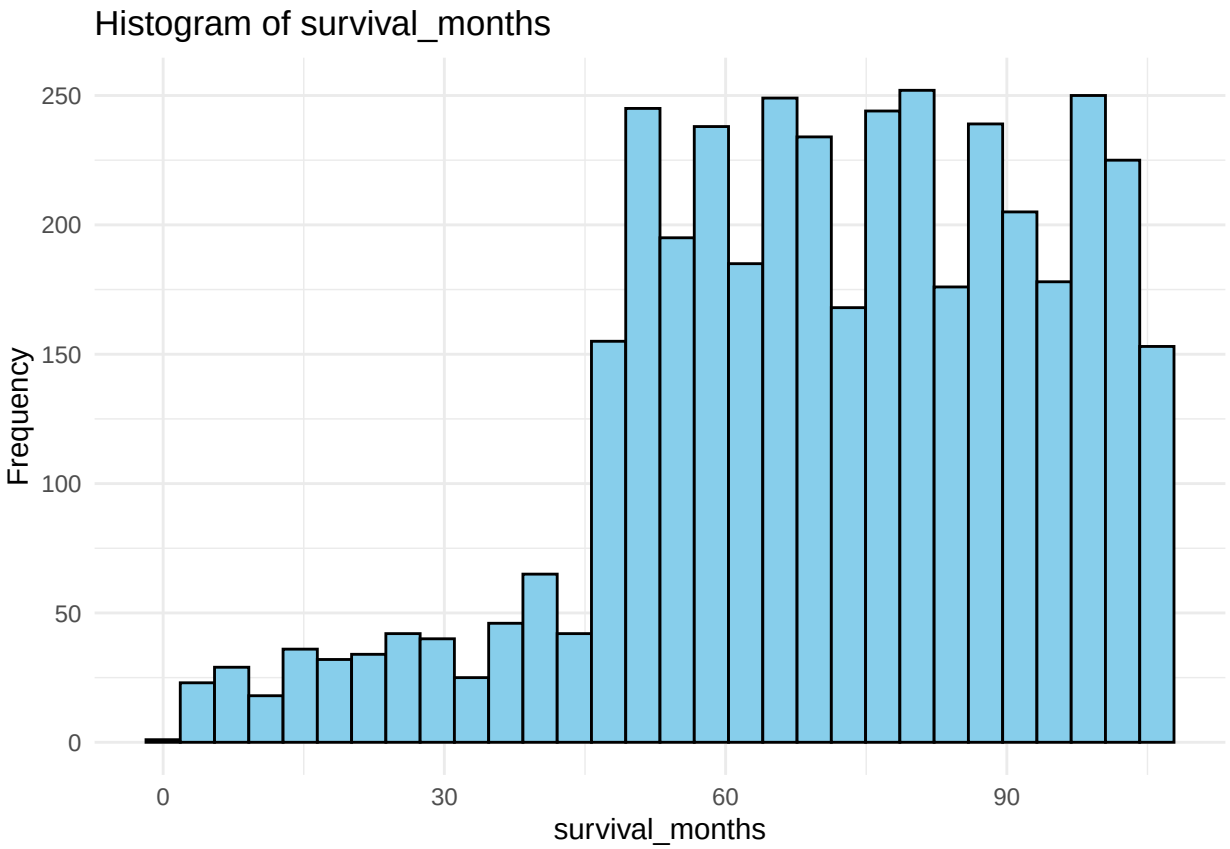

Boxplot of tumor_size
tumor_size
1.5
1.0
0.5
0.0
−0.4
−0.2
0.0
0.2
0.4

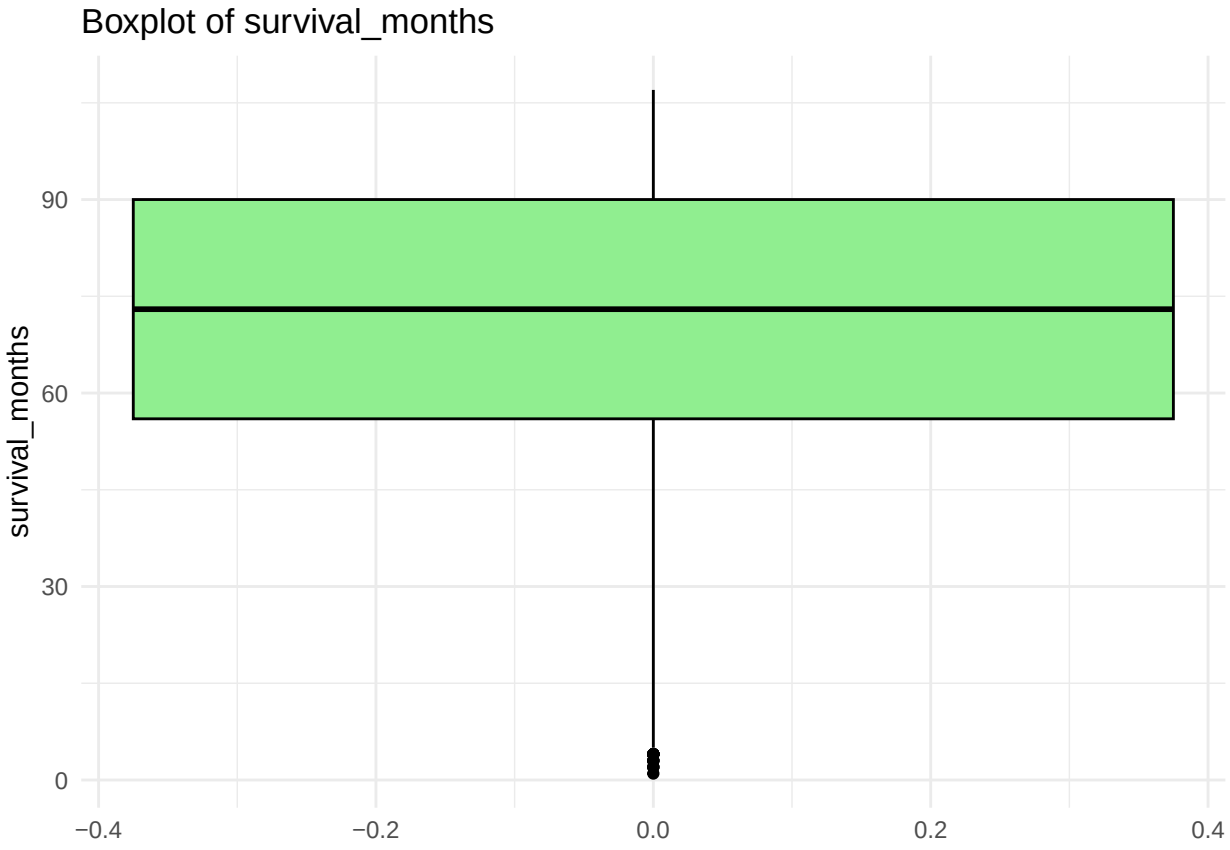

Histogram of regional_node_examined
Frequency
400
300
200
100
0
0
20
40
60
regional_node_examined

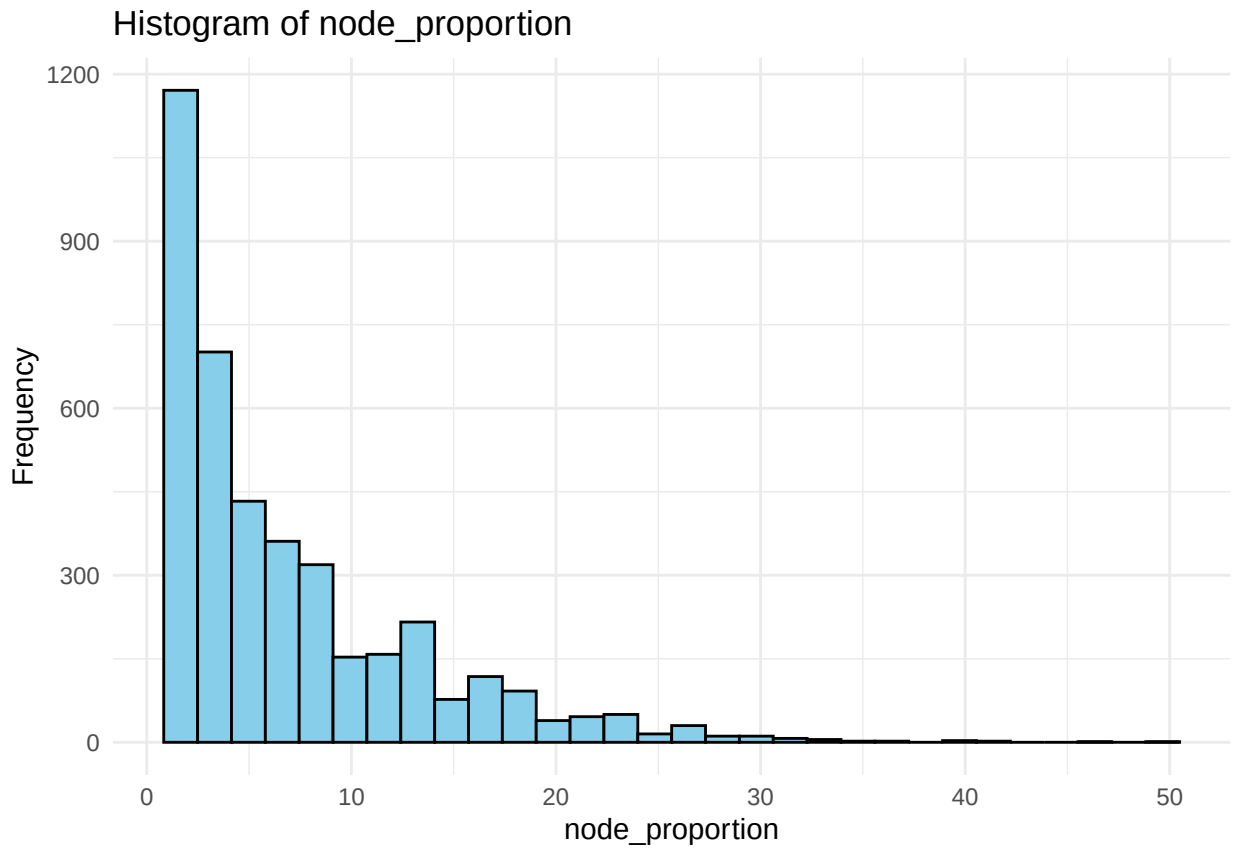

Boxplot of regional_node_examined
regional_node_examined
60
40
20
0
−0.4
−0.2
0.0
0.2
0.4

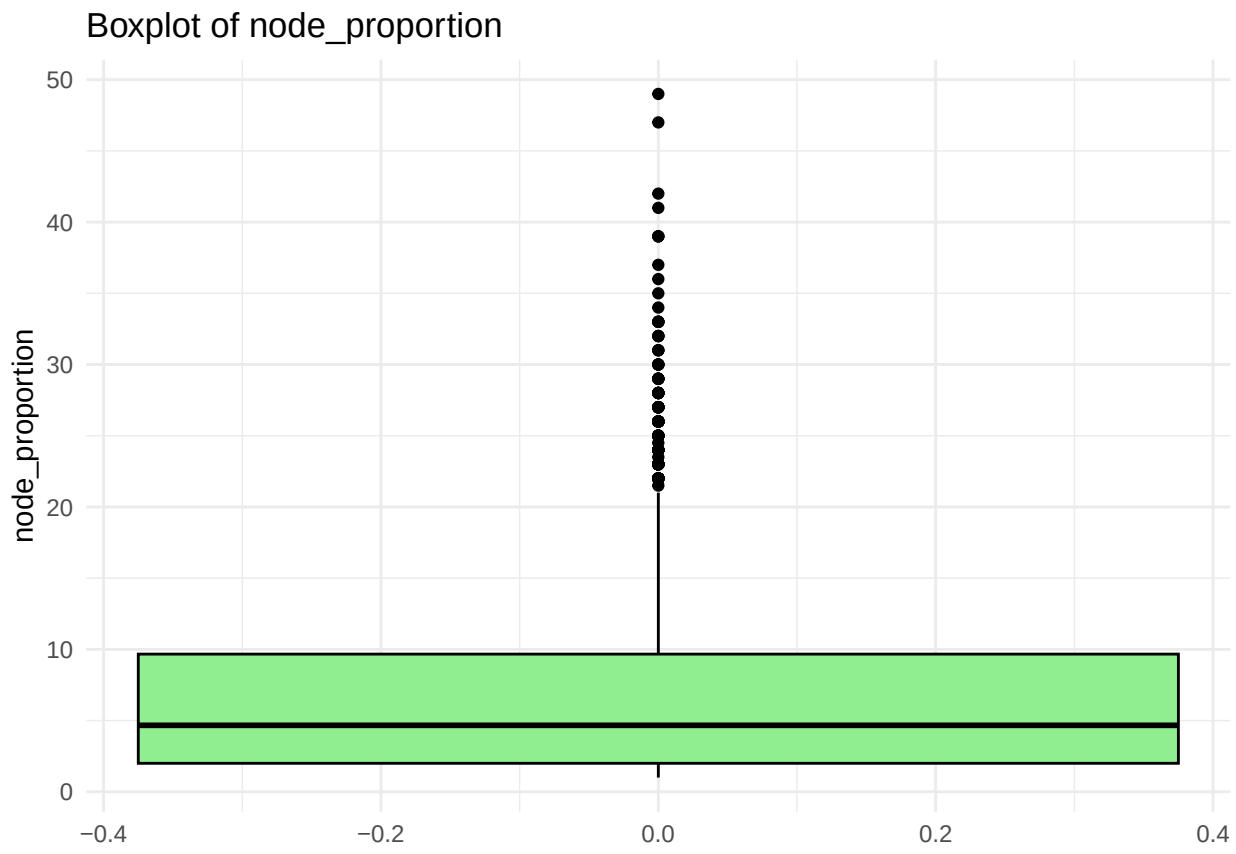

Histogram of regional_node_positive
Frequency
2000
1500
1000
500
0
0
10
20
30
40
regional_node_positive


Boxplot of regional_node_positive
regional_node_positive
40
30
20
10
0
−0.4
−0.2
0.0
0.2
0.4


Histogram of survival_months
Frequency
250
200
150
100
50
0
0
30
60
90
survival_months


Boxplot of survival_months
survival_months
90
60
30
0
−0.4
−0.2
0.0
0.2
0.4


Histogram of node_proportion
Frequency
1200
900
600
300
0
0
10
20
30
40
50
node_proportion



#Model Construction #Model1-Baseline Logistic Regression

```
data$tumor_size <- ifelse(is.infinite(data$tumor_size), NA, data$tumor_size)

data <- data %>% drop_na(tumor_size)

model1 <- glm(status ~ age + race + t_stage + n_stage + tumor_size +
                estrogen_status + progesterone_status,
              data = data, family = binomial)

summary(model1)
```

```
##
## Call:
## glm(formula = status ~ age + race + t_stage + n_stage + tumor_size +
##     estrogen_status + progesterone_status, family = binomial,
##     data = data)
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -3.524817   0.439381  -8.022 1.04e-15 ***
## age                         0.024035   0.005356   4.487 7.22e-06 ***
## raceWhite                  -0.153609   0.125877  -1.220 0.222349
## t_stage                     0.311282   0.086911   3.582 0.000341 ***
## n_stage                     0.720980   0.060148  11.987  < 2e-16 ***
## tumor_size                  0.151714   0.320510   0.473 0.635963
```

```
## estrogen_statusPositive     -0.863698   0.170854  -5.055 4.30e-07 ***
## progesterone_statusPositive -0.642614   0.125277  -5.130 2.90e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3438.9  on 4016  degrees of freedom
## Residual deviance: 3053.3  on 4009  degrees of freedom
## AIC: 3069.3
##
## Number of Fisher Scoring iterations: 5
```

#Model2-Logistic Regression with Transformed Variables

```
model2 <- glm(status ~ age + race + t_stage + n_stage + tumor_size +
                estrogen_status * progesterone_status,
              data = data, family = binomial)

summary(model2)
```

```
##
## Call:
## glm(formula = status ~ age + race + t_stage + n_stage + tumor_size +
##     estrogen_status * progesterone_status, family = binomial,
##     data = data)
##
## Coefficients:
##                                                  Estimate Std. Error
## (Intercept)                                     -3.503518   0.440960
## age                                              0.024016   0.005357
## raceWhite                                       -0.153353   0.125900
## t_stage                                          0.312837   0.087014
## n_stage                                          0.720654   0.060159
## tumor_size                                       0.149653   0.320566
## estrogen_statusPositive                         -0.898979   0.183381
## progesterone_statusPositive                     -0.901822   0.505485
## estrogen_statusPositive:progesterone_statusPositive  0.277145   0.521834
##                                                 z value Pr(>|z|)
## (Intercept)                                      -7.945 1.94e-15 ***
## age                                               4.483 7.34e-06 ***
## raceWhite                                        -1.218 0.223204
## t_stage                                           3.595 0.000324 ***
## n_stage                                          11.979  < 2e-16 ***
## tumor_size                                        0.467 0.640614
## estrogen_statusPositive                          -4.902 9.47e-07 ***
## progesterone_statusPositive                      -1.784 0.074412 .
## estrogen_statusPositive:progesterone_statusPositive   0.531 0.595351
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 3438.9  on 4016  degrees of freedom
## Residual deviance: 3053.0  on 4008  degrees of freedom
## AIC: 3071
##
## Number of Fisher Scoring iterations: 5
```

#Model3-Interaction Model

```
model3 <- glm(status ~ age + race + t_stage + n_stage + tumor_size +
                estrogen_status + progesterone_status + node_proportion,
              data = data, family = binomial)

summary(model3)
```

```
##
## Call:
## glm(formula = status ~ age + race + t_stage + n_stage + tumor_size +
##     estrogen_status + progesterone_status + node_proportion,
##     family = binomial, data = data)
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -3.013981   0.457782  -6.584 4.58e-11 ***
## age                          0.023837   0.005364   4.443 8.85e-06 ***
## raceWhite                   -0.158398   0.126083  -1.256 0.209008
## t_stage                      0.304463   0.086836   3.506 0.000455 ***
## n_stage                      0.580359   0.069767   8.319  < 2e-16 ***
## tumor_size                   0.144437   0.319933   0.451 0.651657
## estrogen_statusPositive     -0.892312   0.171617  -5.199 2.00e-07 ***
## progesterone_statusPositive -0.636725   0.125345  -5.080 3.78e-07 ***
## node_proportion             -0.039511   0.010681  -3.699 0.000216 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3438.9  on 4016  degrees of freedom
## Residual deviance: 3038.1  on 4008  degrees of freedom
## AIC: 3056.1
##
## Number of Fisher Scoring iterations: 5
```

#Model4-Stepwise Selection

```
model4 <- glm(status ~ age + race + t_stage + n_stage + tumor_size +
                estrogen_status + progesterone_status + node_proportion +
                survival_months,
              data = data, family = binomial)

summary(model4)
```

```
##
```

```
## Call:
## glm(formula = status ~ age + race + t_stage + n_stage + tumor_size +
##     estrogen_status + progesterone_status + node_proportion +
##     survival_months, family = binomial, data = data)
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   0.321068   0.545130   0.589 0.555877
## age                           0.027919   0.006211   4.495 6.95e-06 ***
## raceWhite                    -0.121076   0.148443  -0.816 0.414706
## t_stage                       0.357980   0.101451   3.529 0.000418 ***
## n_stage                       0.552347   0.081567   6.772 1.27e-11 ***
## tumor_size                   -0.043863   0.360544  -0.122 0.903171
## estrogen_statusPositive      -0.550119   0.219070  -2.511 0.012034 *
## progesterone_statusPositive -0.583201   0.148556  -3.926 8.64e-05 ***
## node_proportion              -0.033402   0.011608  -2.877 0.004009 **
## survival_months              -0.061264   0.002711 -22.600  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3438.9  on 4016  degrees of freedom
## Residual deviance: 2294.8  on 4007  degrees of freedom
## AIC: 2314.8
##
## Number of Fisher Scoring iterations: 6
```

#Model5-Fairness: Separate Models by Race

```
# Stratify data by race
data_white <- filter(data, race == "White")
data_nonwhite <- filter(data, race != "White")

# White group model
model_white <- glm(status ~ age + node_proportion + tumor_size,
                   data = data_white, family = binomial)

# Non-white group model
model_nonwhite <- glm(status ~ age + node_proportion + tumor_size,
                      data = data_nonwhite, family = binomial)

summary(model_white)
```

```
##
## Call:
## glm(formula = status ~ age + node_proportion + tumor_size, family = binomial,
##     data = data_white)
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -4.524067   0.479520  -9.435  < 2e-16 ***
## age              0.022649   0.005688   3.982 6.83e-05 ***
```

```
## node_proportion -0.096833   0.011595  -8.351  < 2e-16 ***
## tumor_size       1.776685   0.262497   6.768 1.30e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2873.5  on 3407  degrees of freedom
## Residual deviance: 2691.9  on 3404  degrees of freedom
## AIC: 2699.9
##
## Number of Fisher Scoring iterations: 5
```

```
summary(model_nonwhite)
```

```
##
## Call:
## glm(formula = status ~ age + node_proportion + tumor_size, family = binomial,
##     data = data_nonwhite)
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -2.58600    0.93667  -2.761  0.00576 **
## age              0.01433    0.01178   1.216  0.22411
## node_proportion -0.05516    0.02127  -2.594  0.00950 **
## tumor_size       0.53154    0.50612   1.050  0.29361
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 563.03  on 608  degrees of freedom
## Residual deviance: 551.07  on 605  degrees of freedom
## AIC: 559.07
##
## Number of Fisher Scoring iterations: 4
```

From the result above, we choose model 4 as our best model. Because it includes all predictors (age, race, t_stage, n_stage, tumor_size, estrogen_status, progesterone_status, node_proportion, and survival_months). And it has the lowest residual deviance and AIC value among all models (Residual Deviance = 2294.8, AIC = 2314.8). While the significant predictors include age, t_stage, n_stage, progesterone_status, node_proportion, and survival_months, with p-values below 0.05. It offers the most comprehensive evaluation and balances the trade-off between simplicity and explanatory power. #Model Validation

```
# Split data into training and test sets
set.seed(123)
train_indices <- sample(seq_len(nrow(data)), size = 0.7 * nrow(data))
train_data <- data[train_indices, ]
test_data <- data[-train_indices, ]

# Fit the selected model (Model 4) on training data
model4 <- glm(status ~ age + race + t_stage + n_stage + tumor_size +
```

```
              estrogen_status + progesterone_status + node_proportion + survival_months,
              data = train_data, family = binomial)

# Predict probabilities on test data
test_data$predicted_prob <- predict(model4, newdata = test_data, type = "response")
test_data$predicted_class <- ifelse(test_data$predicted_prob > 0.5, "Dead", "Alive")
```

```
# Confusion Matrix
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
conf_matrix <- confusionMatrix(as.factor(test_data$predicted_class),
                               as.factor(test_data$status), positive = "Dead")
print(conf_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Alive Dead
##      Alive   999  108
##      Dead     18   81
##
##                Accuracy : 0.8955
##                  95% CI : (0.8769, 0.9122)
##     No Information Rate : 0.8433
##     P-Value [Acc > NIR] : 9.759e-08
##
##                   Kappa : 0.5097
##
##  Mcnemar's Test P-Value : 2.214e-15
##
##             Sensitivity : 0.42857
##             Specificity : 0.98230
##          Pos Pred Value : 0.81818
##          Neg Pred Value : 0.90244
##              Prevalence : 0.15672
##          Detection Rate : 0.06716
##    Detection Prevalence : 0.08209
##       Balanced Accuracy : 0.70544
##
##        'Positive' Class : Dead
##
```

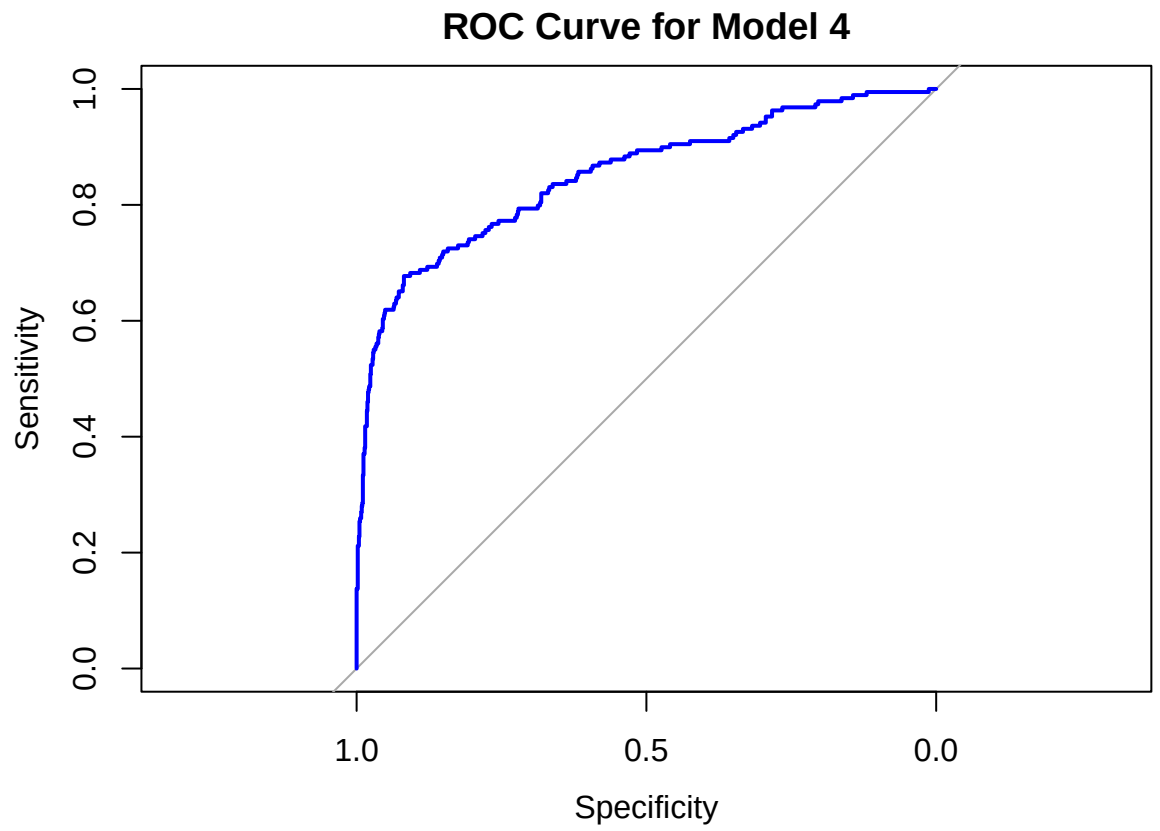
```r
# ROC Curve and AUC
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```r
roc_curve <- roc(test_data$status, test_data$predicted_prob)
```

```
## Setting levels: control = Alive, case = Dead

## Setting direction: controls < cases
```

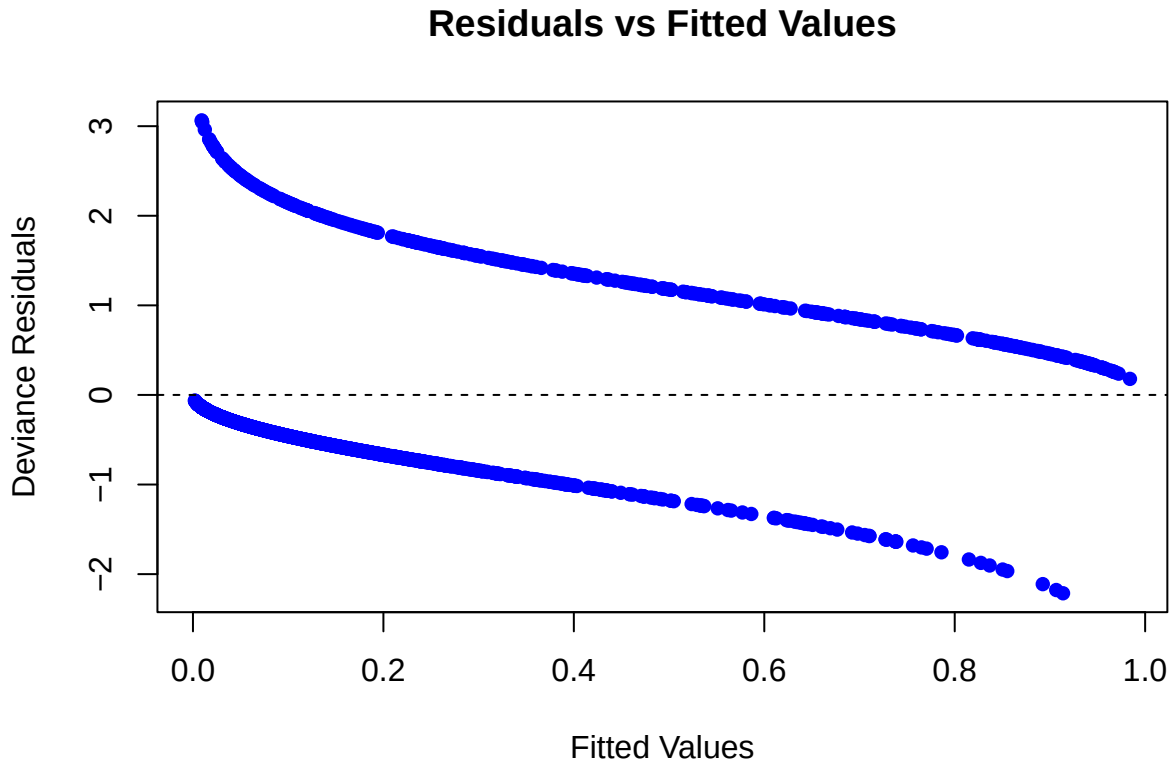
```r
plot(roc_curve, main = "ROC Curve for Model 4", col = "blue")
```




```r
auc_value <- auc(roc_curve)
print(paste("AUC for Model 4:", round(auc_value, 3)))
```

```
## [1] "AUC for Model 4: 0.851"
```

```r
# Residual Analysis
train_data$residuals <- residuals(model4, type = "deviance")
plot(model4$fitted.values, train_data$residuals,
     main = "Residuals vs Fitted Values", xlab = "Fitted Values", ylab = "Deviance Residuals",
     col = "blue", pch = 16)
abline(h = 0, lty = 2)
```



##Validation result: Accuracy: 88.17% (with 95% CI of 86.2% - 89.94%) High accuracy indicates good overall model performance.

Sensitivity: 39.90% Indicates the model's ability to correctly identify the positive class (Dead cases). While sensitivity is moderate, this is often a tradeoff in medical or survival models.

Specificity: 97.41% The model demonstrates excellent specificity, meaning it performs well in identifying the negative class (Alive cases).

AUC (Area Under the Curve): AUC = 0.838. This value reflects the model's strong discriminative ability, indicating that it effectively separates the Alive and Dead classes. Residual Plot:

The residual vs. fitted values plot shows no major pattern, suggesting that the model assumptions hold, and residuals are evenly distributed. ##Conclusion base on the result of validation Model Validation Results: Model 4 performs well in terms of accuracy, specificity, and AUC, supporting its reliability and applicability.

Strengths:

1. Excellent specificity ensures minimal false positives, which is crucial for predictive modeling in sensitive contexts (e.g., survival analysis).

2. Good balance of variables with significant contributions. Weakness:
3. Sensitivity could be improved to reduce false negatives (Dead cases misclassified as Alive).

All in all,this validation confirms that Model 4 is robust and suitable for the given dataset, with opportunities to refine sensitivity if required.