ins_pro

Srihari Myla Venkata

2024-05-05

```r
# Load necessary libraries
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.3.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(Hmisc)
```

```
## Warning: package 'Hmisc' was built under R version 4.3.3
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```r
# Read the data
data <- read.csv("C:\\Users\\mvsri\\Downloads\\insurance.csv")

# View the first few rows and summary statistics of the data
head(data)
```

```
##   age    sex    bmi children smoker    region   charges
## 1  19 female 27.900        0    yes southwest 16884.924
## 2  18   male 33.770        1     no southeast  1725.552
## 3  28   male 33.000        3     no southeast  4449.462
## 4  33   male 22.705        0     no northwest 21984.471
## 5  32   male 28.880        0     no northwest  3866.855
## 6  31 female 25.740        0     no southeast  3756.622
```

```
summary(data)
```

```
##       age            sex                 bmi           children
##  Min.   :18.00   Length:1338        Min.   :15.96   Min.   :0.000
##  1st Qu.:27.00   Class :character   1st Qu.:26.30   1st Qu.:0.000
##  Median :39.00   Mode  :character   Median :30.40   Median :1.000
##  Mean   :39.21                      Mean   :30.66   Mean   :1.095
##  3rd Qu.:51.00                      3rd Qu.:34.69   3rd Qu.:2.000
##  Max.   :64.00                      Max.   :53.13   Max.   :5.000
##     smoker             region            charges
##  Length:1338        Length:1338        Min.   : 1122
##  Class :character   Class :character   1st Qu.: 4740
##  Mode  :character   Mode  :character   Median : 9382
##                                        Mean   :13270
##                                        3rd Qu.:16640
##                                        Max.   :63770
```

```
# Check the dimensions (rows and columns) of the dataframe
dimensions <- dim(data)
num_rows <- dimensions[1]
num_columns <- dimensions[2]

# Print the number of rows and columns
cat("Number of rows:", num_rows, "\n")
```

```
## Number of rows: 1338
```

```
cat("Number of columns:", num_columns, "\n")
```

```
## Number of columns: 7
```

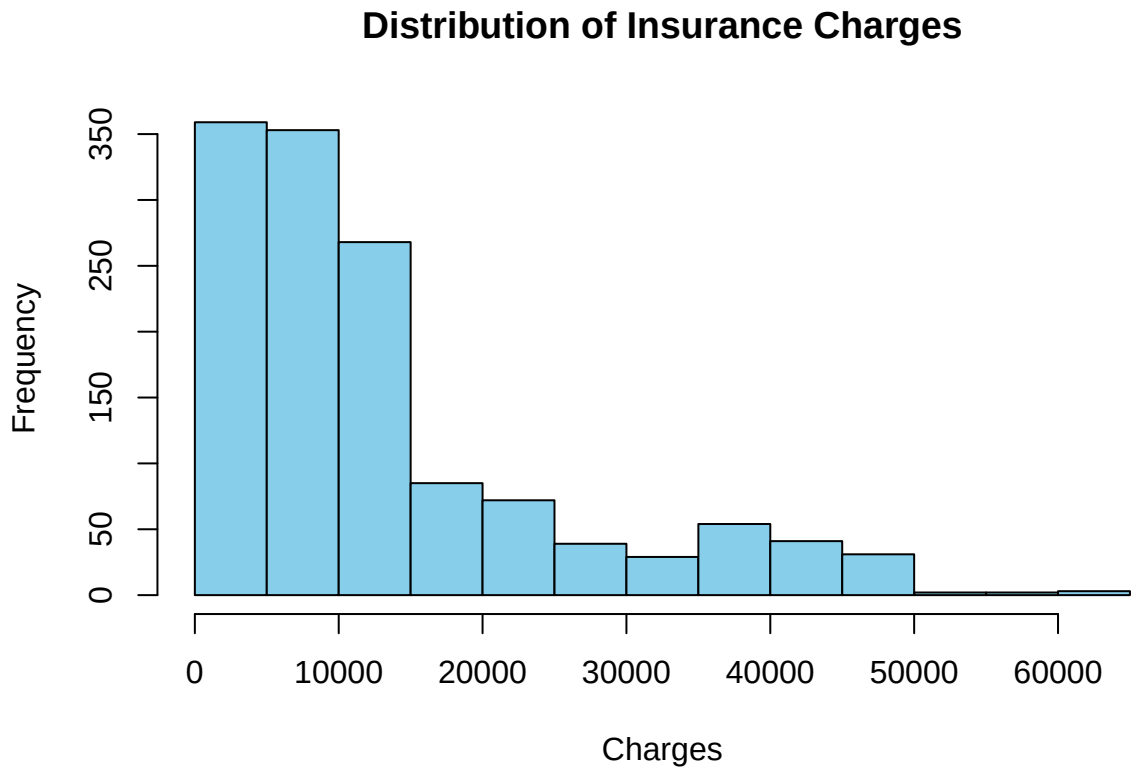
```
# Check the structure of the dataframe to identify variable types
str(data)
```

```
## 'data.frame':    1338 obs. of  7 variables:
##  $ age     : int  19 18 28 33 32 31 46 37 37 60 ...
##  $ sex     : chr  "female" "male" "male" "male" ...
##  $ bmi     : num  27.9 33.8 33 22.7 28.9 ...
##  $ children: int  0 1 3 0 0 0 1 3 2 0 ...
##  $ smoker  : chr  "yes" "no" "no" "no" ...
##  $ region  : chr  "southwest" "southeast" "southeast" "northwest" ...
##  $ charges : num  16885 1726 4449 21984 3867 ...
```

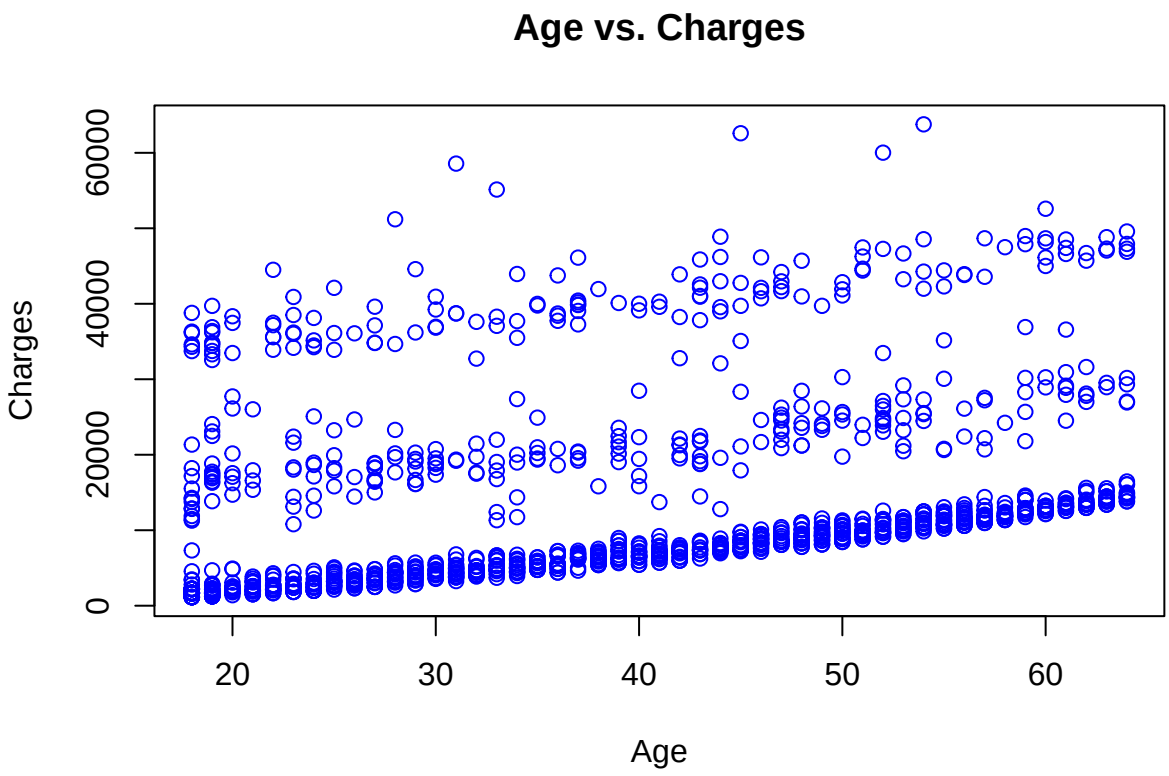
```r
# Check for null values in the dataframe
null_values <- is.na(data)

# Check if there are any null values overall
if (any(null_values)) {
  cat("There are null values in the dataset.\n")
  # Print the count of null values for each column
  print(colSums(null_values))
} else {
  cat("There are no null values in the dataset.\n")
}
```

```
## There are no null values in the dataset.
```

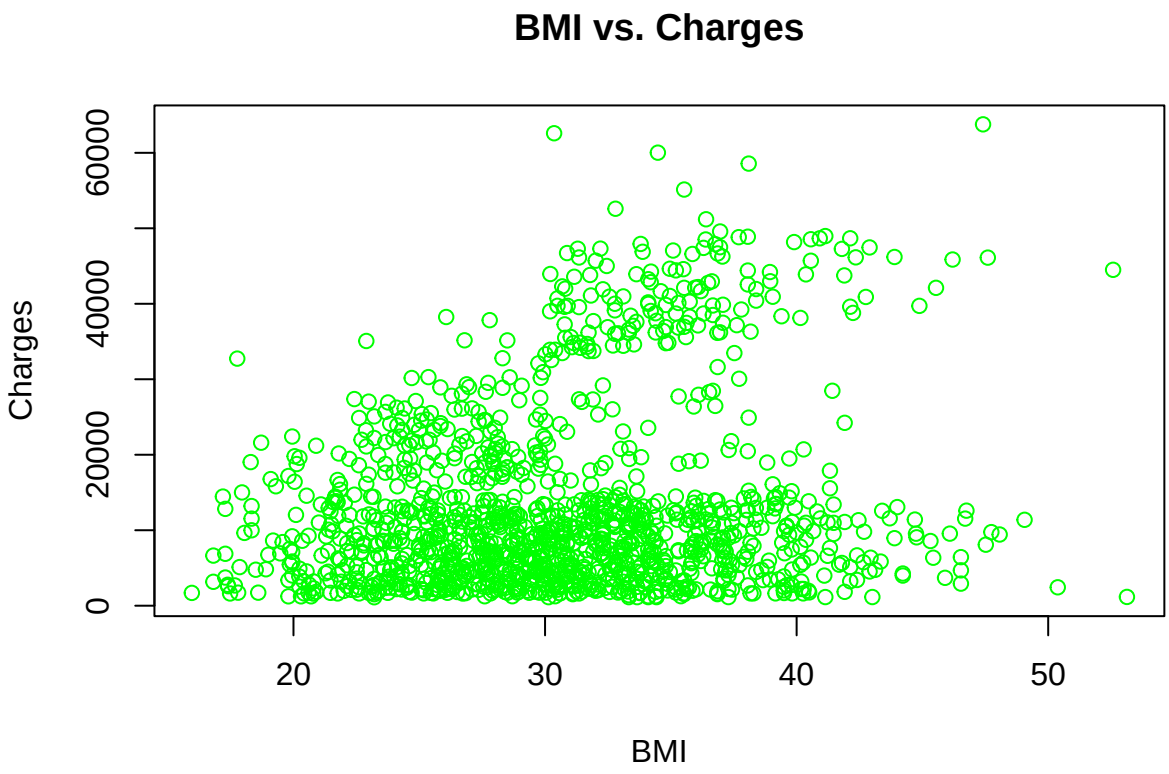
```r
# Visualize the distribution of charges
hist(data$charges, main = "Distribution of Insurance Charges", xlab = "Charges", col = "skyblue")
```



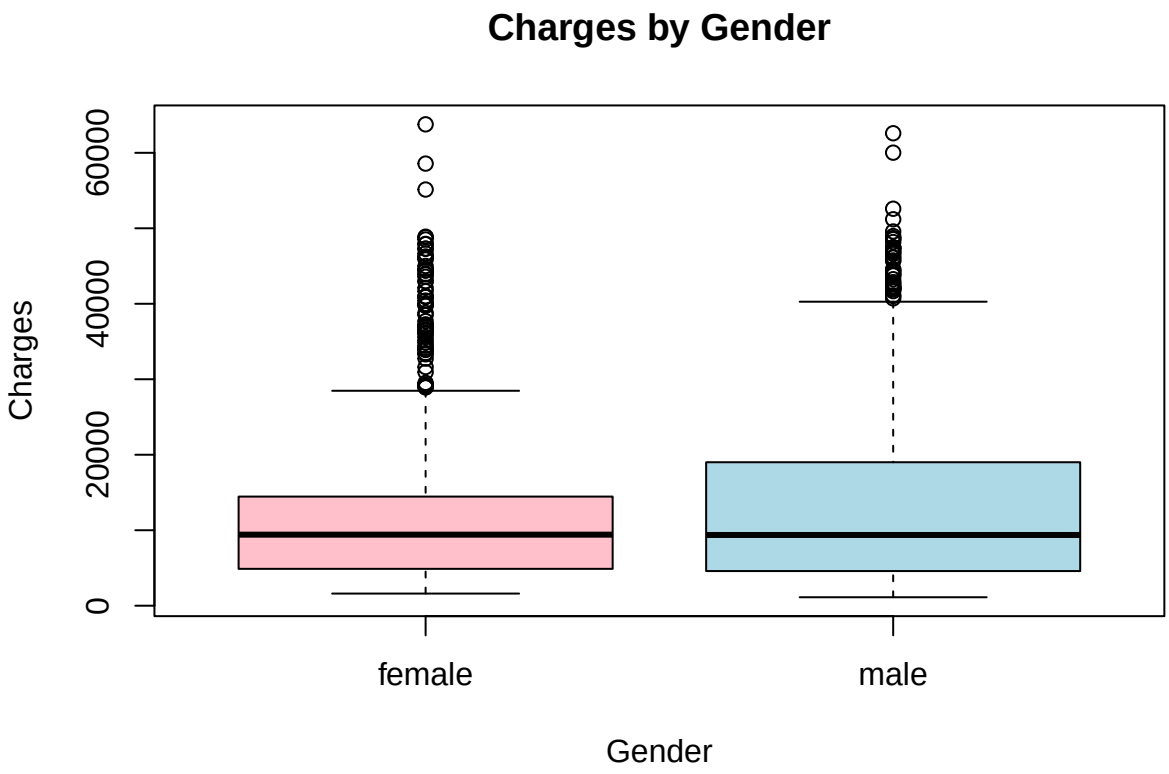

```r
# Explore the relationship between age and charges
plot(data$age, data$charges, main = "Age vs. Charges", xlab = "Age", ylab = "Charges", col = "blue")
```



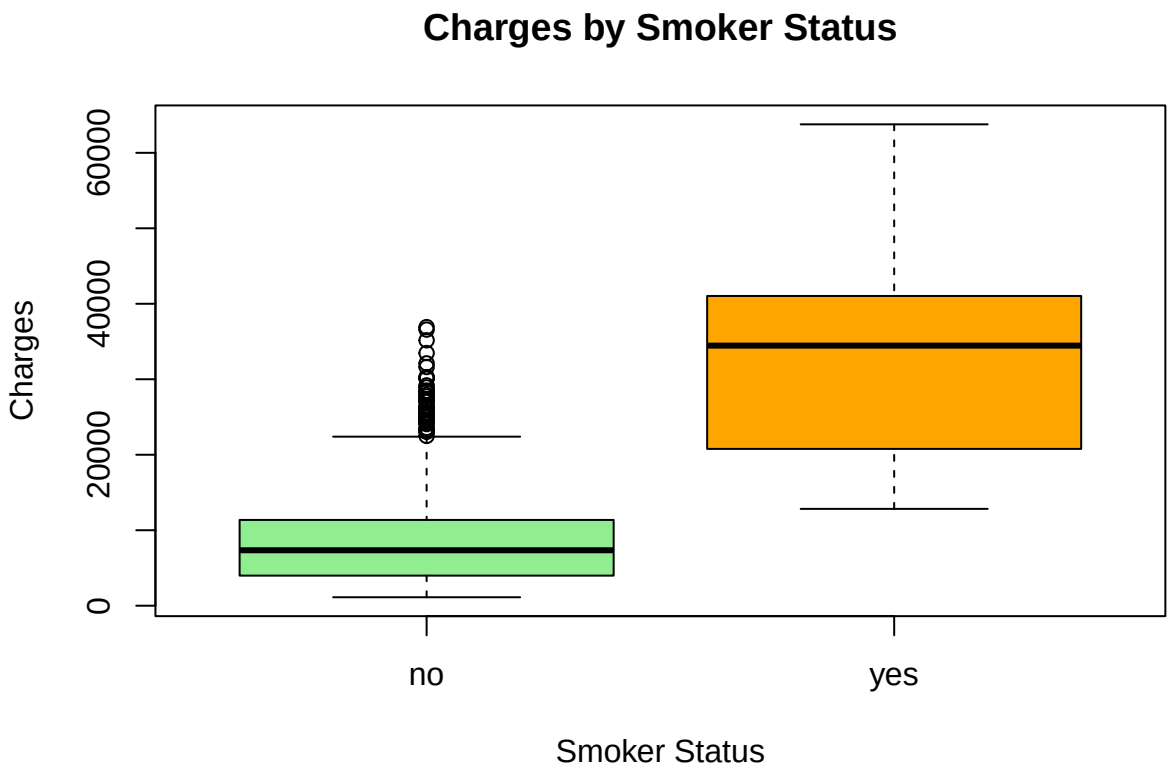

```r
# Explore the relationship between BMI and charges
plot(data$bmi, data$charges, main = "BMI vs. Charges", xlab = "BMI", ylab = "Charges", col = "green")
```



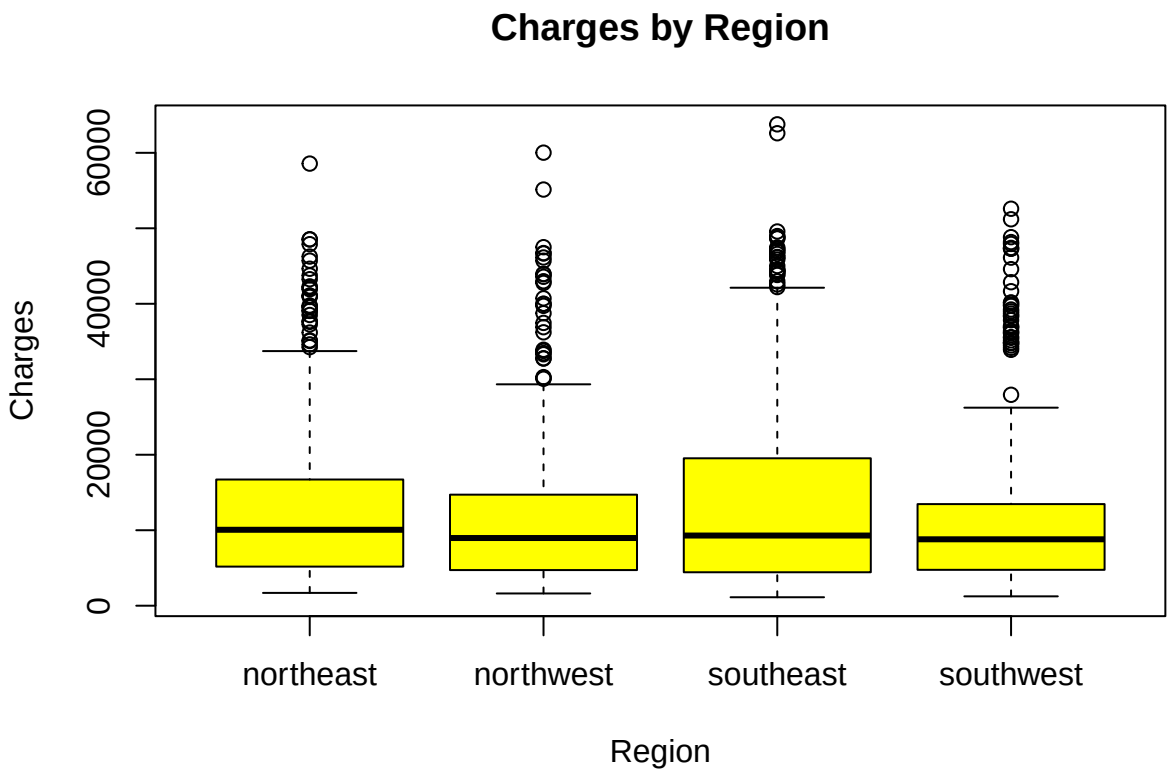

```
# Compare charges by gender
boxplot(charges ~ sex, data = data, main = "Charges by Gender", xlab = "Gender", ylab = "Charges", col =
```



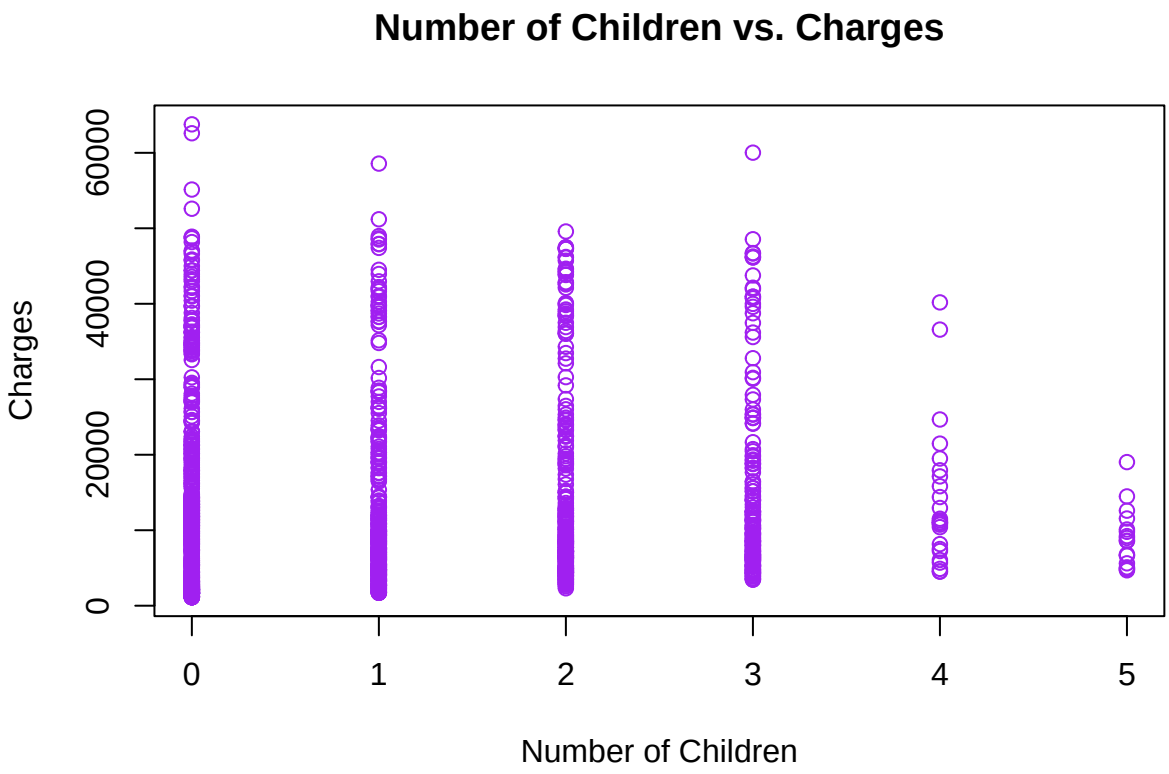

```r
# Compare charges by smoker status
boxplot(charges ~ smoker, data = data, main = "Charges by Smoker Status", xlab = "Smoker Status", ylab =
```



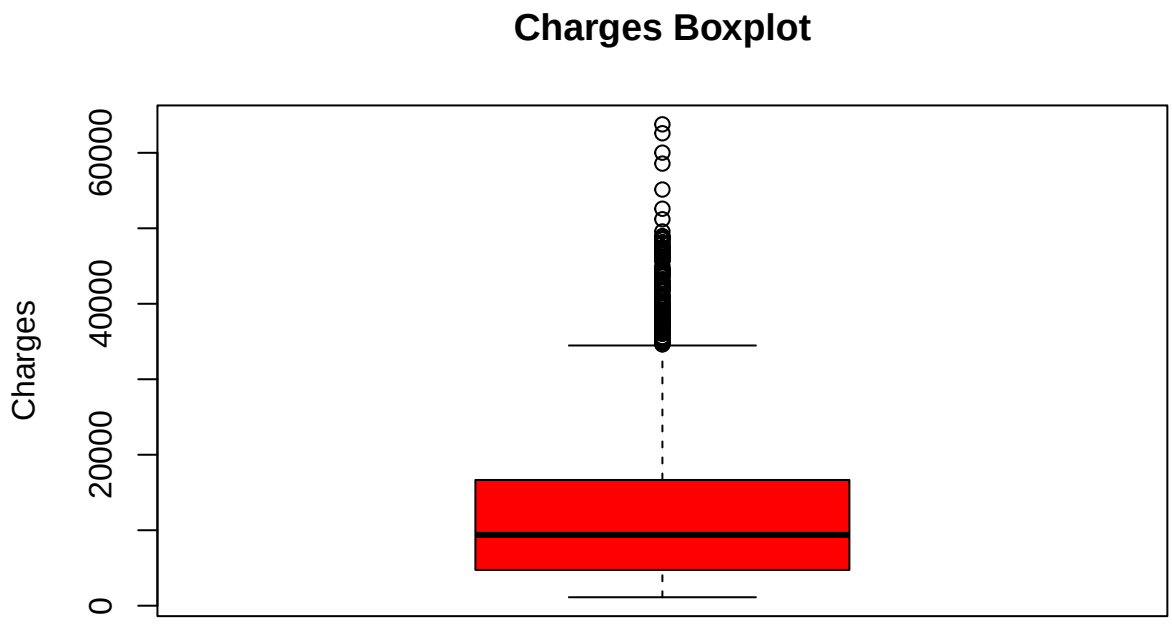

```r
# Compare charges by region
boxplot(charges ~ region, data = data, main = "Charges by Region", xlab = "Region", ylab = "Charges", co
```



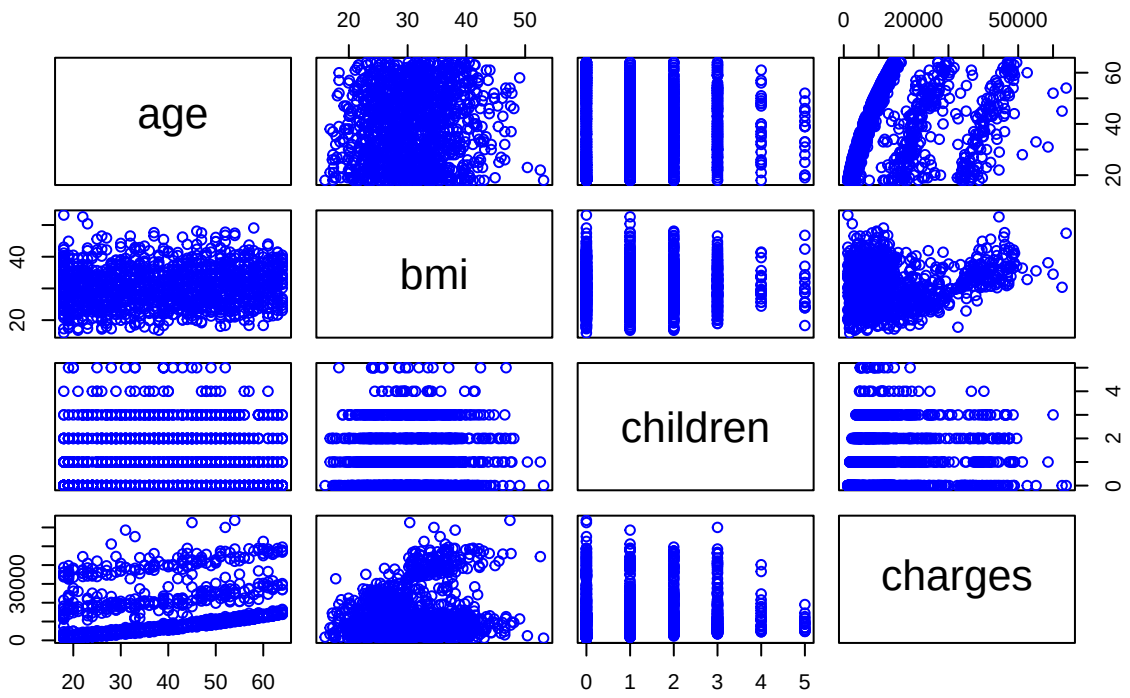

```
# Explore the relationship between the number of children and charges
plot(data$children, data$charges, main = "Number of Children vs. Charges", xlab = "Number of Children",
```



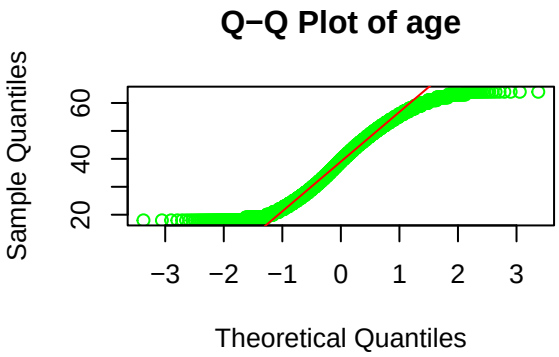

```r
# Identify outliers or anomalies in charges
boxplot(data$charges, main = "Charges Boxplot", ylab = "Charges", col = "red")
```

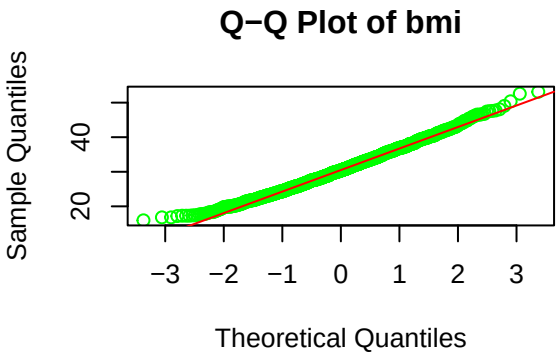


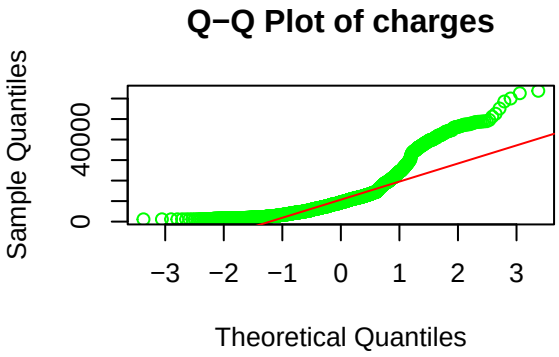

```
# Create a scatterplot matrix to visualize relationships between variables
pairs(data[, c("age", "bmi", "children", "charges")], main = "Scatterplot Matrix", col = "blue")
```




```r
# Add Q-Q plots
par(mfrow = c(2, 2)) # Set the layout to 2x2 for Q-Q plots
for (variable in c("age", "bmi", "charges")) {
  if (is.numeric(data[[variable]])) {
    qqnorm(data[[variable]], main = paste("Q-Q Plot of", variable), col = "green")
    qqline(data[[variable]], col = "red")
  }
}
```








```r
# Load necessary library
library(Hmisc)

# Load the dataset (update the path as needed)
insurance_data <- read.csv("insurance.csv")  # Make sure the file path is correct

# Check for missing values
sum(is.na(insurance_data))
```

```
## [1] 0
```

```r
# Remove rows with missing values (if any)
insurance_data <- na.omit(insurance_data)

# Convert non-numeric columns to numeric if possible
insurance_data <- type.convert(insurance_data, as.is = TRUE)

# Convert categorical variables to factors
insurance_data$sex <- as.factor(insurance_data$sex)
insurance_data$smoker <- as.factor(insurance_data$smoker)
insurance_data$region <- as.factor(insurance_data$region)

# Identify categorical columns
categorical_cols <- c("sex", "smoker", "region")
```

```r
# Convert categorical columns to numerical using one-hot encoding
encoded_data <- model.matrix(~ 0 + ., data = insurance_data[, categorical_cols])

# Combine the encoded columns with the original dataframe
insurance_data_numeric <- cbind(insurance_data[, !names(insurance_data) %in% categorical_cols], encoded

# Compute Spearman's correlation
cor_matrix <- rcorr(as.matrix(insurance_data_numeric), type = "spearman")

# Print the correlation matrix
print(cor_matrix$r)
```

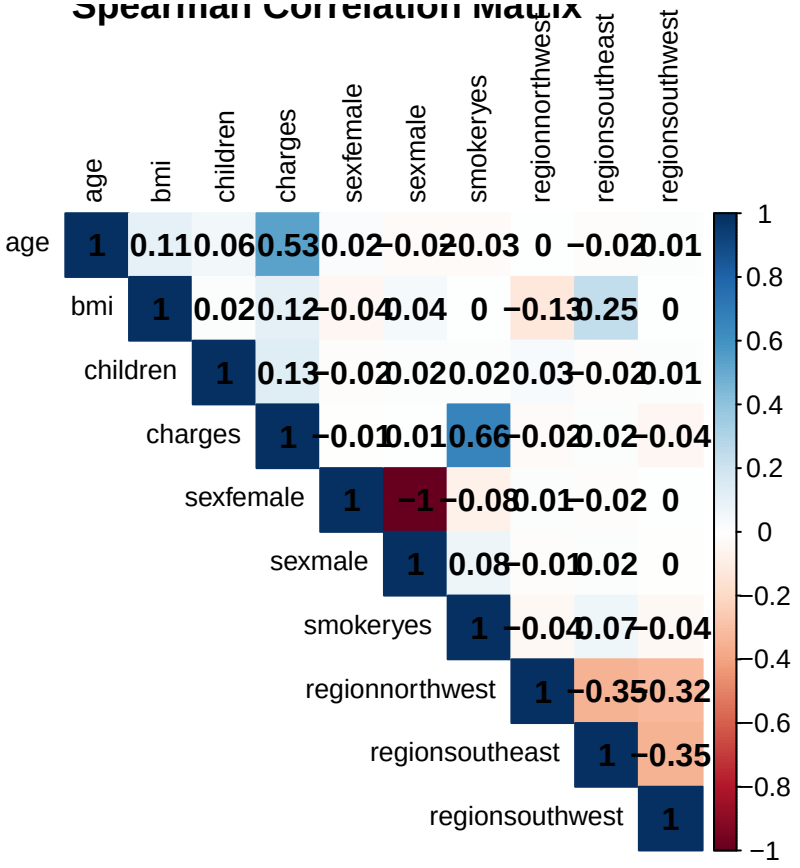
```
##                          age          bmi    children      charges    sexfemale
## age              1.000000000  0.107736035  0.05699222  0.534392134  0.020808830
## bmi              0.107736035  1.000000000  0.01560674  0.119395904 -0.044801536
## children         0.056992224  0.015606736  1.00000000  0.133338943 -0.015588577
## charges          0.534392134  0.119395904  0.13333894  1.000000000 -0.009489706
## sexfemale        0.020808830 -0.044801536 -0.01558858 -0.009489706  1.000000000
## sexmale         -0.020808830  0.044801536  0.01558858  0.009489706 -1.000000000
## smokeryes       -0.025210462  0.002203313  0.01658339  0.663460060 -0.076184817
## regionnorthwest  0.002683348 -0.127167912  0.03446494 -0.021633737  0.011155728
## regionsoutheast -0.015273341  0.249037111 -0.01953102  0.017275198 -0.017116875
## regionsouthwest  0.013315183  0.001710132  0.01146611 -0.042353754  0.004184049
##                       sexmale    smokeryes regionnorthwest regionsoutheast
## age             -0.020808830 -0.025210462     0.002683348     -0.01527334
## bmi              0.044801536  0.002203313    -0.127167912      0.24903711
## children         0.015588577  0.016583386     0.034464938     -0.01953102
## charges          0.009489706  0.663460060    -0.021633737      0.01727520
## sexfemale       -1.000000000 -0.076184817     0.011155728     -0.01711688
## sexmale          1.000000000  0.076184817    -0.011155728      0.01711688
## smokeryes        0.076184817  1.000000000    -0.036945474      0.06849841
## regionnorthwest -0.011155728 -0.036945474     1.000000000     -0.34626466
## regionsoutheast  0.017116875  0.068498410    -0.346264661      1.00000000
## regionsouthwest -0.004184049 -0.036945474    -0.320829220     -0.34626466
##                  regionsouthwest
## age                   0.013315183
## bmi                   0.001710132
## children              0.011466110
## charges              -0.042353754
## sexfemale             0.004184049
## sexmale              -0.004184049
## smokeryes            -0.036945474
## regionnorthwest      -0.320829220
## regionsoutheast      -0.346264661
## regionsouthwest       1.000000000
```

```r
# Load the corrplot library
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.3.3
```

```
## corrplot 0.92 loaded
```

```r
# Visualize the Spearman correlation matrix using a heatmap
corrplot(cor_matrix$r, method = "color",
         main = "Spearman Correlation Matrix",
         type = "upper",
         tl.col = "black",
         tl.cex = 0.8,
         addCoef.col = "black")
```




```r
# Filter all numerical columns (excluding the target variable 'charges')
numeric_cols <- data %>%
  select_if(is.numeric)

# Exclude the target variable 'charges'
numeric_cols <- numeric_cols %>%
  select(-charges)

# Filter encoded numerical columns (if you've encoded categorical variables)
encoded_numeric_cols <- data %>%
  select(starts_with("encoded_"))  # Adjust this based on the column names after encoding

# Combine all numerical columns
all_numeric_cols <- cbind(numeric_cols, encoded_numeric_cols)

# Define a function to binarize numeric variables
binarize_numeric <- function(x) {
```

```r
  ifelse(x >= median(x), "high", "low")
}

# Perform t-tests for each numeric column
t_test_results <- lapply(all_numeric_cols, function(col) {
  # Binarize the numeric variable
  binarized_col <- binarize_numeric(col)
  # Perform t-test
  t_test_result <- t.test(data$charges ~ binarized_col, data = data)
  return(t_test_result)
})

# Print t-test results
names(t_test_results) <- names(all_numeric_cols)
t_test_results
```

```
## $age
##
##  Welch Two Sample t-test
##
## data:  data$charges by binarized_col
## t = 9.6216, df = 1330.3, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group high and group low is not equal to 0
## 95 percent confidence interval:
##  4909.555 7424.311
## sample estimates:
## mean in group high  mean in group low
##           16261.71           10094.77
##
##
## $bmi
##
##  Welch Two Sample t-test
##
## data:  data$charges by binarized_col
## t = 7.3633, df = 1062.9, p-value = 3.589e-13
## alternative hypothesis: true difference in means between group high and group low is not equal to 0
## 95 percent confidence interval:
##  3504.587 6050.983
## sample estimates:
## mean in group high  mean in group low
##           15655.74           10877.96
##
##
## $children
##
##  Welch Two Sample t-test
##
## data:  data$charges by binarized_col
## t = 2.3753, df = 1240.3, p-value = 0.01769
## alternative hypothesis: true difference in means between group high and group low is not equal to 0
## 95 percent confidence interval:
##   275.6744 2892.2566
```

```
## sample estimates:
## mean in group high  mean in group low
##           13949.94           12365.98
```

```r
# Perform chi-square tests for each categorical column
chi_square_results <- lapply(categorical_cols, function(col) {
  chi_square_result <- chisq.test(table(data[[col]], data$charges > median(data$charges)))
  return(chi_square_result)
})

# Print chi-square test results
names(chi_square_results) <- categorical_cols
chi_square_results
```

```
## $sex
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(data[[col]], data$charges > median(data$charges))
## X-squared = 0.0029899, df = 1, p-value = 0.9564
##
##
## $smoker
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(data[[col]], data$charges > median(data$charges))
## X-squared = 342.05, df = 1, p-value < 2.2e-16
##
##
## $region
##
##  Pearson's Chi-squared test
##
## data:  table(data[[col]], data$charges > median(data$charges))
## X-squared = 4.466, df = 3, p-value = 0.2153
```

```r
# Perform Shapiro-Wilk test for normality
shapiro_test_result <- shapiro.test(data$charges)

# Print the test result
shapiro_test_result
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data$charges
## W = 0.81469, p-value < 2.2e-16
```

```r
# Filter categorical and numerical variables
categorical_vars <- c("sex", "smoker", "region")
numeric_vars <- c("age", "bmi", "children")
```

```r
# Perform Kruskal-Wallis test for categorical variables
kruskal_cat <- lapply(categorical_vars, function(var) {
  kruskal.test(charges ~ get(var), data = data)
})

# Perform Kruskal-Wallis test for numerical variables
kruskal_num <- lapply(numeric_vars, function(var) {
  kruskal.test(data$charges ~ data[[var]])
})

# Print results for categorical variables
cat_results <- data.frame(
  Variable = categorical_vars,
  Statistic = sapply(kruskal_cat, function(x) x$statistic),
  P_Value = sapply(kruskal_cat, function(x) x$p.value)
)
print("Kruskal-Wallis test results for categorical variables:")
```

```
## [1] "Kruskal-Wallis test results for categorical variables:"
```

```r
print(cat_results)
```

```
##   Variable   Statistic       P_Value
## 1      sex   0.1204029  7.285979e-01
## 2   smoker 588.5196584 5.259018e-130
## 3   region   4.7341812  1.923291e-01
```

```r
# Print results for numerical variables
num_results <- data.frame(
  Variable = numeric_vars,
  Statistic = sapply(kruskal_num, function(x) x$statistic),
  P_Value = sapply(kruskal_num, function(x) x$p.value)
)
print("Kruskal-Wallis test results for numerical variables:")
```

```
## [1] "Kruskal-Wallis test results for numerical variables:"
```

```r
print(num_results)
```

```
##   Variable Statistic      P_Value
## 1      age 420.28637 6.734492e-62
## 2      bmi 527.91984 7.134508e-01
## 3 children  29.48707 1.860485e-05
```

```r
# Fit a linear regression model
linear_model <- lm(charges ~ age + bmi + children + sex + smoker + region, data = data)

# Summary of the linear regression model
summary(linear_model)
```

```
##
## Call:
## lm(formula = charges ~ age + bmi + children + sex + smoker +
##     region, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11304.9  -2848.1   -982.1   1393.9  29992.8
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -11938.5      987.8 -12.086  < 2e-16 ***
## age                256.9       11.9  21.587  < 2e-16 ***
## bmi                339.2       28.6  11.860  < 2e-16 ***
## children           475.5      137.8   3.451 0.000577 ***
## sexmale           -131.3      332.9  -0.394 0.693348
## smokeryes        23848.5      413.1  57.723  < 2e-16 ***
## regionnorthwest   -353.0      476.3  -0.741 0.458769
## regionsoutheast  -1035.0      478.7  -2.162 0.030782 *
## regionsouthwest   -960.0      477.9  -2.009 0.044765 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6062 on 1329 degrees of freedom
## Multiple R-squared:  0.7509, Adjusted R-squared:  0.7494
## F-statistic: 500.8 on 8 and 1329 DF,  p-value: < 2.2e-16
```

```r
# Fit a multiple regression model
multiple_model <- lm(charges ~ ., data = data)

# Summary of the multiple regression model
summary(multiple_model)
```

```
##
## Call:
## lm(formula = charges ~ ., data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11304.9  -2848.1   -982.1   1393.9  29992.8
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -11938.5      987.8 -12.086  < 2e-16 ***
## age                256.9       11.9  21.587  < 2e-16 ***
## sexmale           -131.3      332.9  -0.394 0.693348
## bmi                339.2       28.6  11.860  < 2e-16 ***
## children           475.5      137.8   3.451 0.000577 ***
## smokeryes        23848.5      413.1  57.723  < 2e-16 ***
## regionnorthwest   -353.0      476.3  -0.741 0.458769
## regionsoutheast  -1035.0      478.7  -2.162 0.030782 *
## regionsouthwest   -960.0      477.9  -2.009 0.044765 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

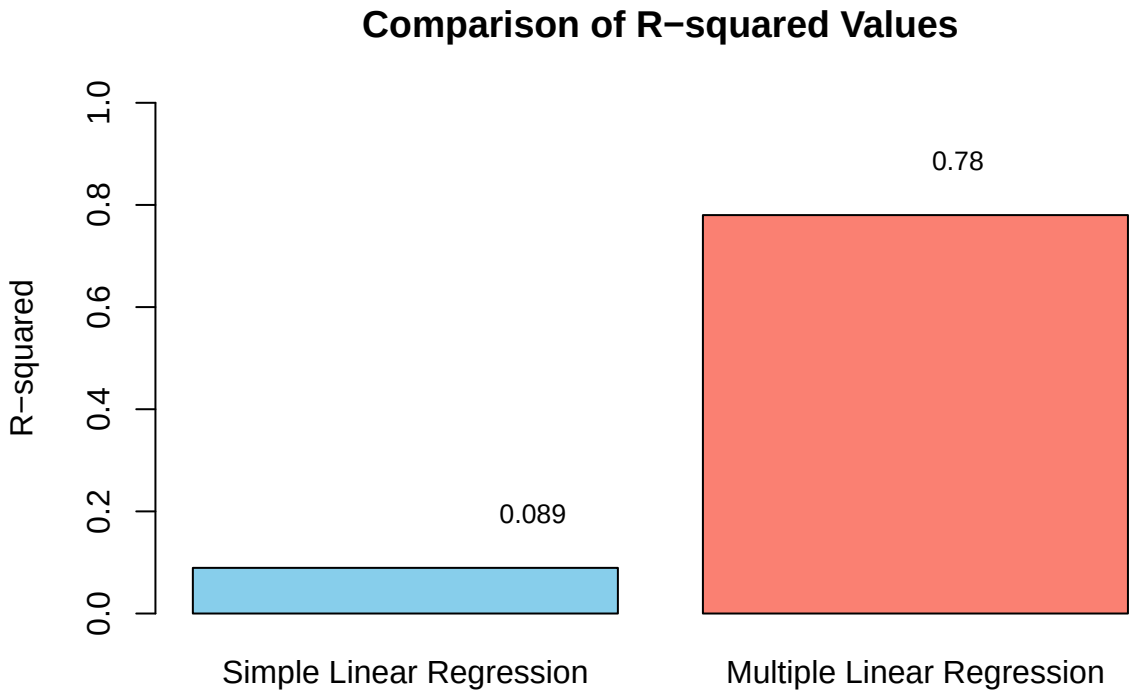
```
##
## Residual standard error: 6062 on 1329 degrees of freedom
## Multiple R-squared:  0.7509, Adjusted R-squared:  0.7494
## F-statistic: 500.8 on 8 and 1329 DF,  p-value: < 2.2e-16
```

```
# Convert 'charges' to a binary variable based on a threshold (e.g., median)
data$charges_binary <- ifelse(data$charges > median(data$charges), 1, 0)

# Fit a logistic regression model
logistic_model <- glm(charges_binary ~ age + bmi + children + sex + smoker + region, data = data, family

# Summary of the logistic regression model
summary(logistic_model)
```

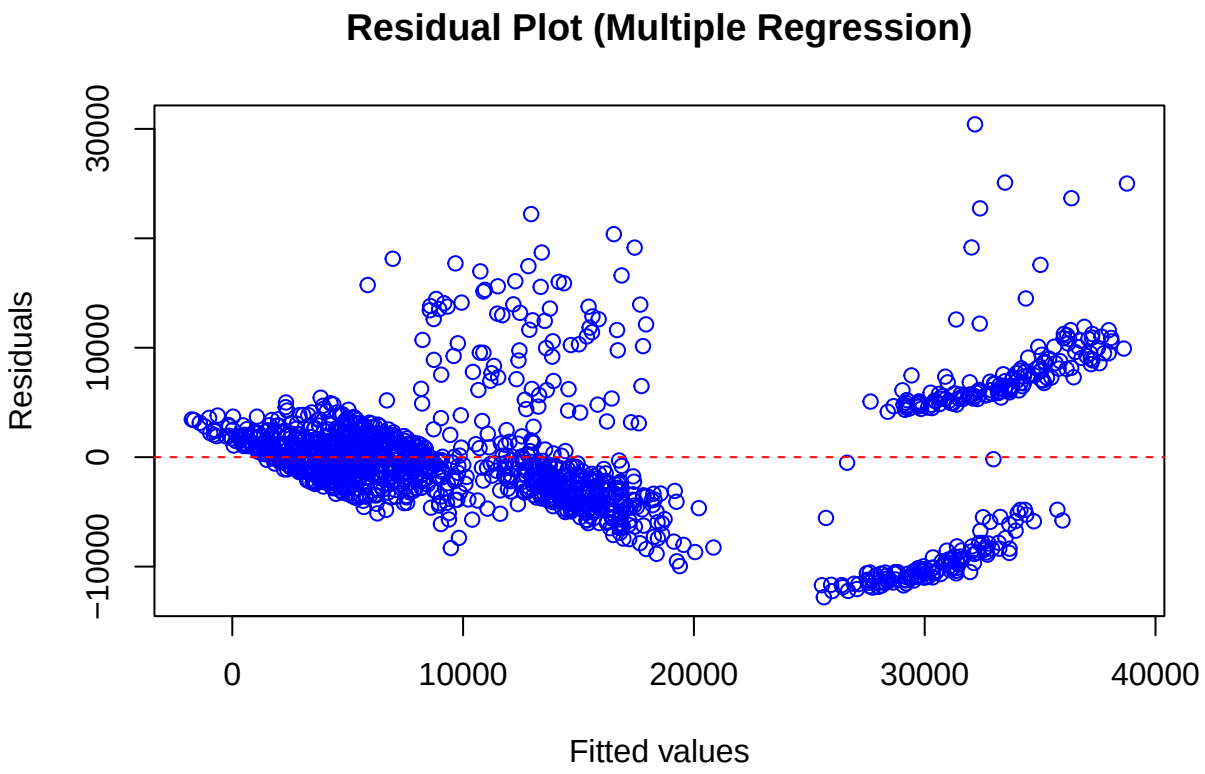
```
##
## Call:
## glm(formula = charges_binary ~ age + bmi + children + sex + smoker +
##     region, family = binomial, data = data)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -8.17993    0.67948 -12.038  < 2e-16 ***
## age               0.16683    0.01004  16.624  < 2e-16 ***
## bmi               0.03268    0.01582   2.065  0.03891 *
## children          0.14483    0.07495   1.932  0.05333 .
## sexmale          -0.35313    0.18188  -1.942  0.05219 .
## smokeryes        22.32977  509.88463   0.044  0.96507
## regionnorthwest  -0.41109    0.25915  -1.586  0.11267
## regionsoutheast  -0.86119    0.26801  -3.213  0.00131 **
## regionsouthwest  -0.77646    0.25872  -3.001  0.00269 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1854.86  on 1337  degrees of freedom
## Residual deviance:  773.45  on 1329  degrees of freedom
## AIC: 791.45
##
## Number of Fisher Scoring iterations: 18
```

```
# Linear Regression
# Simple Linear Regression
simple_lm <- lm(charges ~ age, data = data)
r_squared_simple <- summary(simple_lm)$r.squared

# Multiple Linear Regression
multiple_lm <- lm(charges ~ ., data = data)
r_squared_multiple <- summary(multiple_lm)$r.squared

# Combine R-squared values with model names
model_names <- c("Simple Linear Regression", "Multiple Linear Regression")
r_squared_values <- c(r_squared_simple, r_squared_multiple)
```

```r
# Create a bar plot to visualize R-squared values
barplot(r_squared_values,
        names.arg = model_names,
        ylab = "R-squared", main = "Comparison of R-squared Values",
        col = c("skyblue", "salmon"), ylim = c(0, 1))

# Add text labels for R-squared values
text(x = 1:2, y = r_squared_values + 0.05, round(r_squared_values, 3), pos = 3, cex = 0.8, col = "black
```




```r
# Fit a multiple linear regression model
lm_multiple <- lm(charges ~ ., data = data)

# Obtain residuals from the model
residuals_multiple <- residuals(lm_multiple)

# Create residual plot with color
plot(fitted(lm_multiple), residuals_multiple,
     xlab = "Fitted values",
     ylab = "Residuals",
     main = "Residual Plot (Multiple Regression)",
     col = "blue")  # Specify color here

# Add a horizontal line at y = 0 for reference
abline(h = 0, col = "red", lty = 2)
```


Residual Plot (Multiple Regression)
Residuals
30000
10000
0
−10000
0
10000
20000
30000
40000
Fitted values