```
title: "Notebook"
author: "Thao Nguyen"
output: pdf_document
```

```r
# Install necessary packages

install.packages('ggplot2')
```

```
## Installing package into 'C:/Users/songt/AppData/Local/R/win-library/4.4'
## (as 'lib' is unspecified)
```

```
## package 'ggplot2' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\songt\AppData\Local\Temp\RtmpCw0goI\downloaded_packages
```

```r
install.packages('dplyr')
```

```
## Installing package into 'C:/Users/songt/AppData/Local/R/win-library/4.4'
## (as 'lib' is unspecified)
```

```
## package 'dplyr' successfully unpacked and MD5 sums checked
```

```
## Warning: cannot remove prior installation of package 'dplyr'
```

```
## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
## C:\Users\songt\AppData\Local\R\win-library\4.4\00LOCK\dplyr\libs\x64\dplyr.dll
## to C:\Users\songt\AppData\Local\R\win-library\4.4\dplyr\libs\x64\dplyr.dll:
## Permission denied
```

```
## Warning: restored 'dplyr'
```

```
##
## The downloaded binary packages are in
##  C:\Users\songt\AppData\Local\Temp\RtmpCw0goI\downloaded_packages
```

```r
install.packages('tidyr')
```

```
## Installing package into 'C:/Users/songt/AppData/Local/R/win-library/4.4'
## (as 'lib' is unspecified)
```

```
## package 'tidyr' successfully unpacked and MD5 sums checked
```

```
## Warning: cannot remove prior installation of package 'tidyr'
```

```
## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
## C:\Users\songt\AppData\Local\R\win-library\4.4\00LOCK\tidyr\libs\x64\tidyr.dll
## to C:\Users\songt\AppData\Local\R\win-library\4.4\tidyr\libs\x64\tidyr.dll:
## Permission denied
## Warning: restored 'tidyr'
```

```
##
## The downloaded binary packages are in
##  C:\Users\songt\AppData\Local\Temp\RtmpCw0goI\downloaded_packages
```

```r
install.packages('gridExtra')
```

```
## Installing package into 'C:/Users/songt/AppData/Local/R/win-library/4.4'
## (as 'lib' is unspecified)
```

```
## package 'gridExtra' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\songt\AppData\Local\Temp\RtmpCw0goI\downloaded_packages
```

```r
install.packages('ggExtra')
```

```
## Installing package into 'C:/Users/songt/AppData/Local/R/win-library/4.4'
## (as 'lib' is unspecified)
```

```
## package 'ggExtra' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\songt\AppData\Local\Temp\RtmpCw0goI\downloaded_packages
```

```r
install.packages('ggridges')
```

```
## Installing package into 'C:/Users/songt/AppData/Local/R/win-library/4.4'
## (as 'lib' is unspecified)
```

```
## package 'ggridges' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\songt\AppData\Local\Temp\RtmpCw0goI\downloaded_packages
```

```r
install.packages('corrplot')
```

```
## Installing package into 'C:/Users/songt/AppData/Local/R/win-library/4.4'
## (as 'lib' is unspecified)
```

```
## package 'corrplot' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\songt\AppData\Local\Temp\RtmpCw0goI\downloaded_packages
```

```r
install.packages('rsample')
```

```
## Installing package into 'C:/Users/songt/AppData/Local/R/win-library/4.4'
## (as 'lib' is unspecified)
## package 'rsample' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\songt\AppData\Local\Temp\RtmpCw0goI\downloaded_packages
```

```
# Load the installed packages
library(tidyr)
library(gridExtra)
library(dplyr)
```

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

## The following object is masked from 'package:gridExtra':
##
##     combine

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

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

```
library(ggplot2)
library(ggExtra)
library(ggridges)
library(corrplot)
```

```
## corrplot 0.94 loaded
```

```
library(rsample)
```

```
df = read.csv('C:/Users/songt/R projects/Medical Cost Prediction/insurance.csv', header = TRUE)
head(df)
```

```
##   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
```

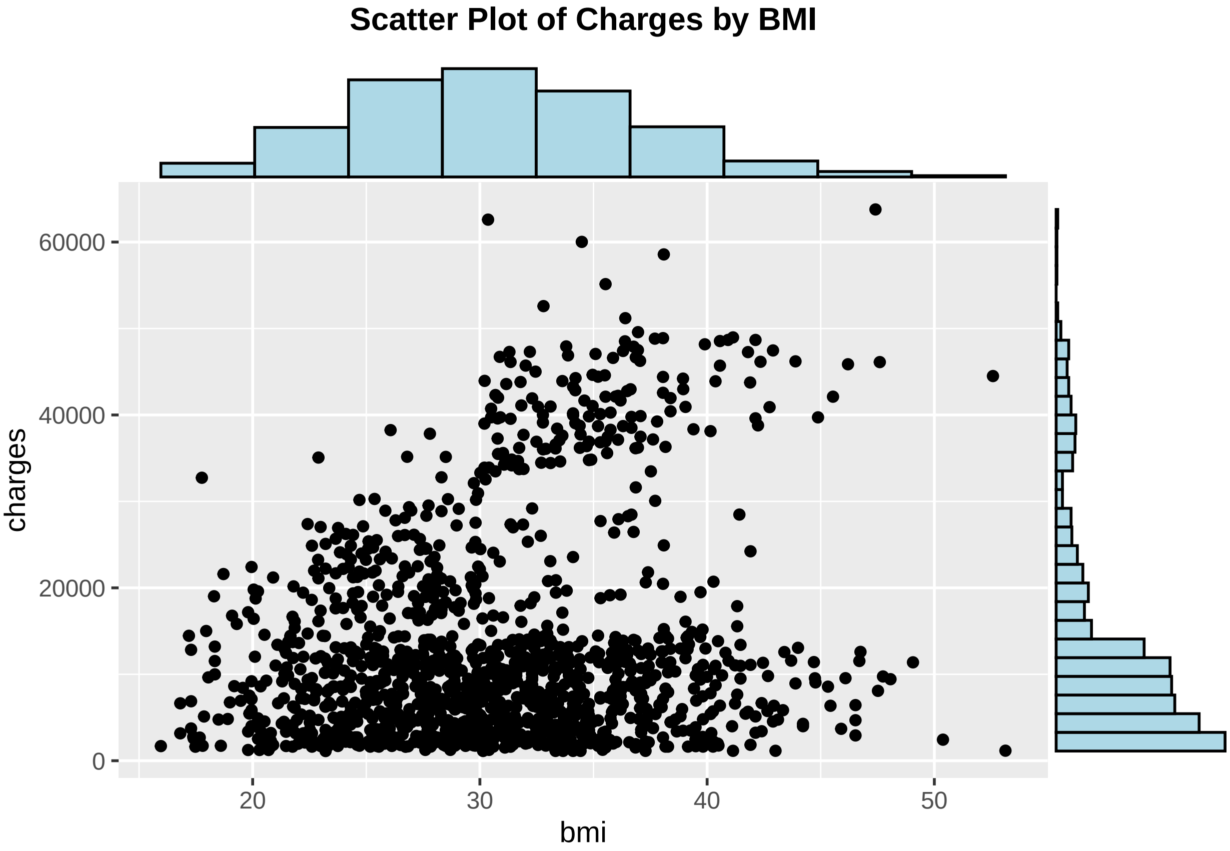
```
summary(df)
```

```
##       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
```

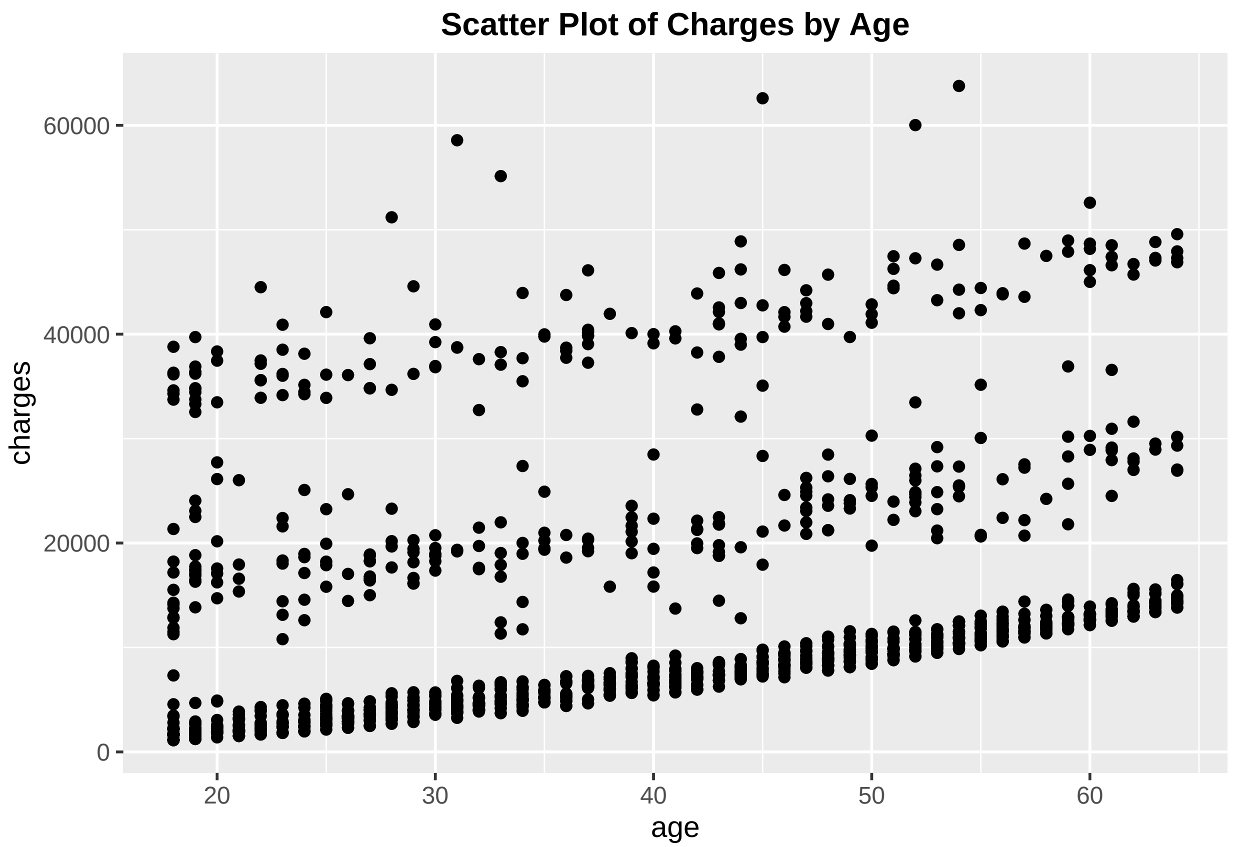
```
str(df)
```

```
## ’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 ...
```

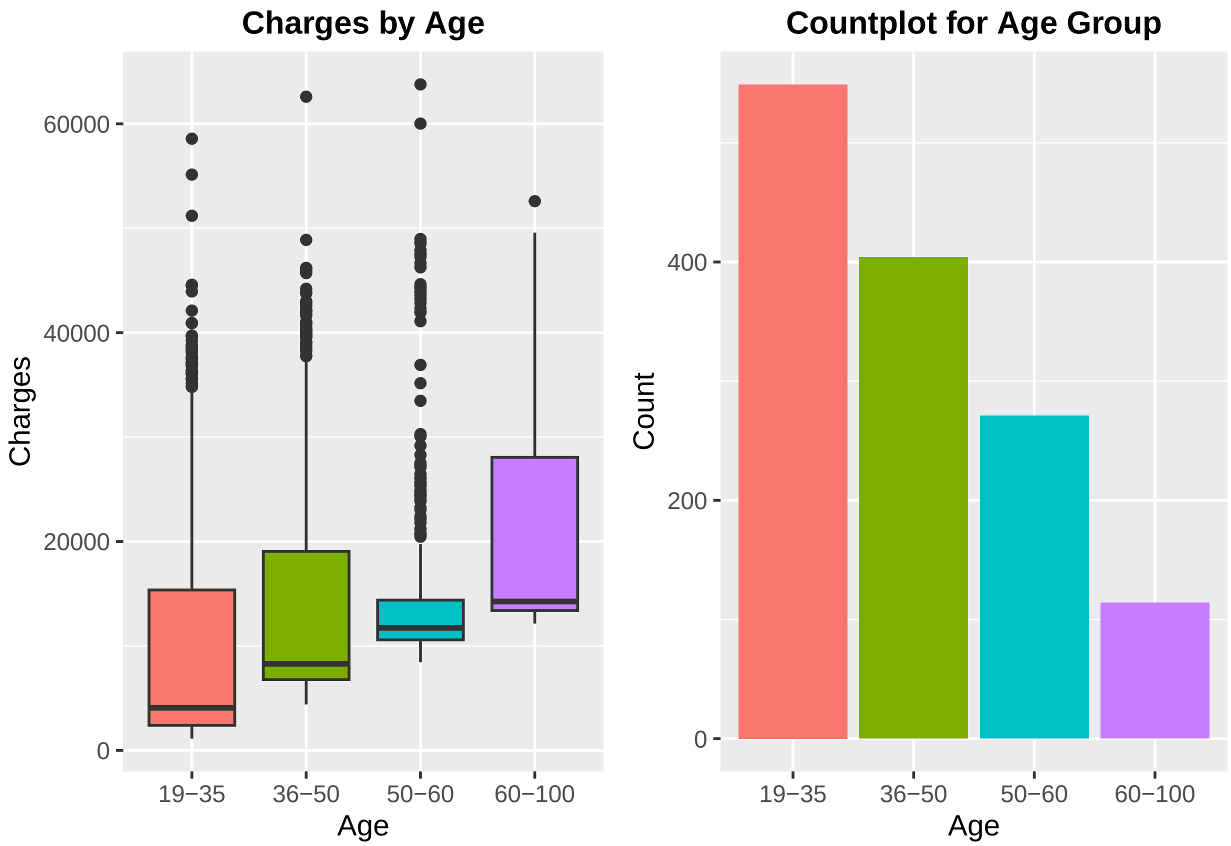
```
# Create the scatter plot of 'charges' versus 'bmi'
g <- ggplot(df, aes(x = bmi, y = charges)) +
  geom_point() +  # Add points to the plot
  theme(legend.position = 'none') +
  ggtitle("Scatter Plot of Charges by BMI") +
  theme(
    plot.title = element_text(size = 12,
                              face = "bold",
                              hjust = 0.5,
                              vjust = 0.5)
  )

# Add marginal histograms to the scatter plot
g1 <- ggMarginal(
  g, type = "histogram",
  fill = 'lightblue',
  xparams = list(bins = 10)
)

# Display the final plot
g1
```



```r
# Create the scatter plot of 'charges' versus 'age'
p <- ggplot(df, aes(x = age, y = charges)) +
  geom_point() +
  theme(legend.position = 'none') +
  ggtitle("Scatter Plot of Charges by Age") +
  theme(
    plot.title = element_text(size = 12,
                              face = "bold",
                              hjust = 0.5,
                              vjust = 0.5)
  )

# Display the plot
p
```



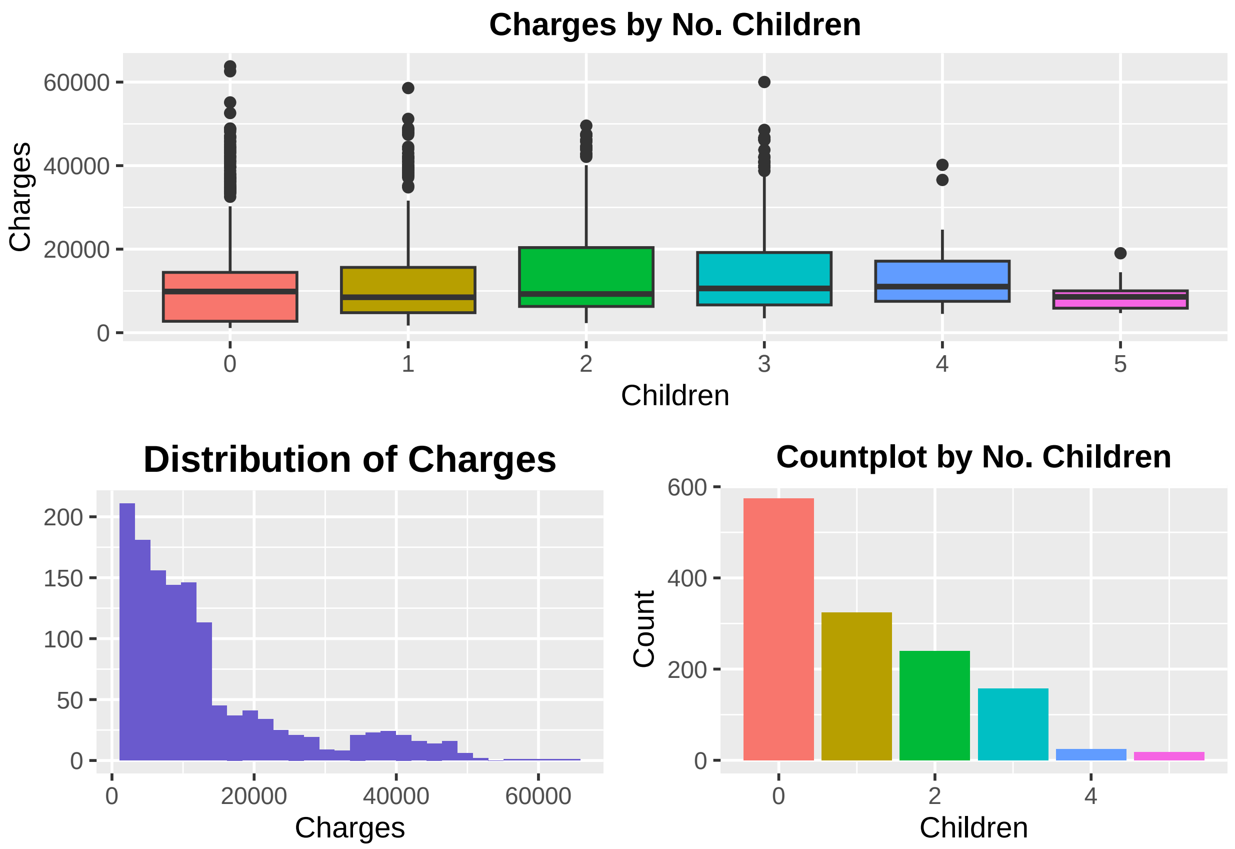
```
# Create age groups in the data frame
df$age_group <- cut(
  df$age,
  breaks = c(0, 18, 35, 50, 60, 100),  # Define age group boundaries
  labels = c('0-18', '19-35', '36-50', '50-60', '60-100'),  # Label each age group
  right = FALSE  # Ensure the interval includes the left value but excludes the right
)

# Boxplot: Charges by Age Group
g1 <- ggplot(df, aes(x = factor(age_group), y = charges, fill = age_group)) +
  geom_boxplot() +
  theme(legend.position = 'none') +
  ggtitle("Charges by Age") +
  theme(plot.title = element_text(colour = 'black',
                                  face = 'bold',
                                  size = 12,
                                  hjust = 0.5,
                                  vjust = 0.5)) +
  xlab('Age') +
  ylab('Charges')

# Barplot: Count of Age Groups
g2 <- ggplot(df, aes(x = age_group, fill = factor(age_group))) +
  geom_bar() +  # Create a bar plot
  theme(legend.position = 'none') +
  ggtitle("Countplot for Age Group") +
  theme(plot.title = element_text(colour = 'black',
                                  face = 'bold',
                                  size = 12,
                                  hjust = 0.5,
                                  vjust = 0.5)) +
  xlab('Age') +
  ylab('Count')

# Arrange the plots side by side
grid.arrange(g1, g2, ncol = 2)
```



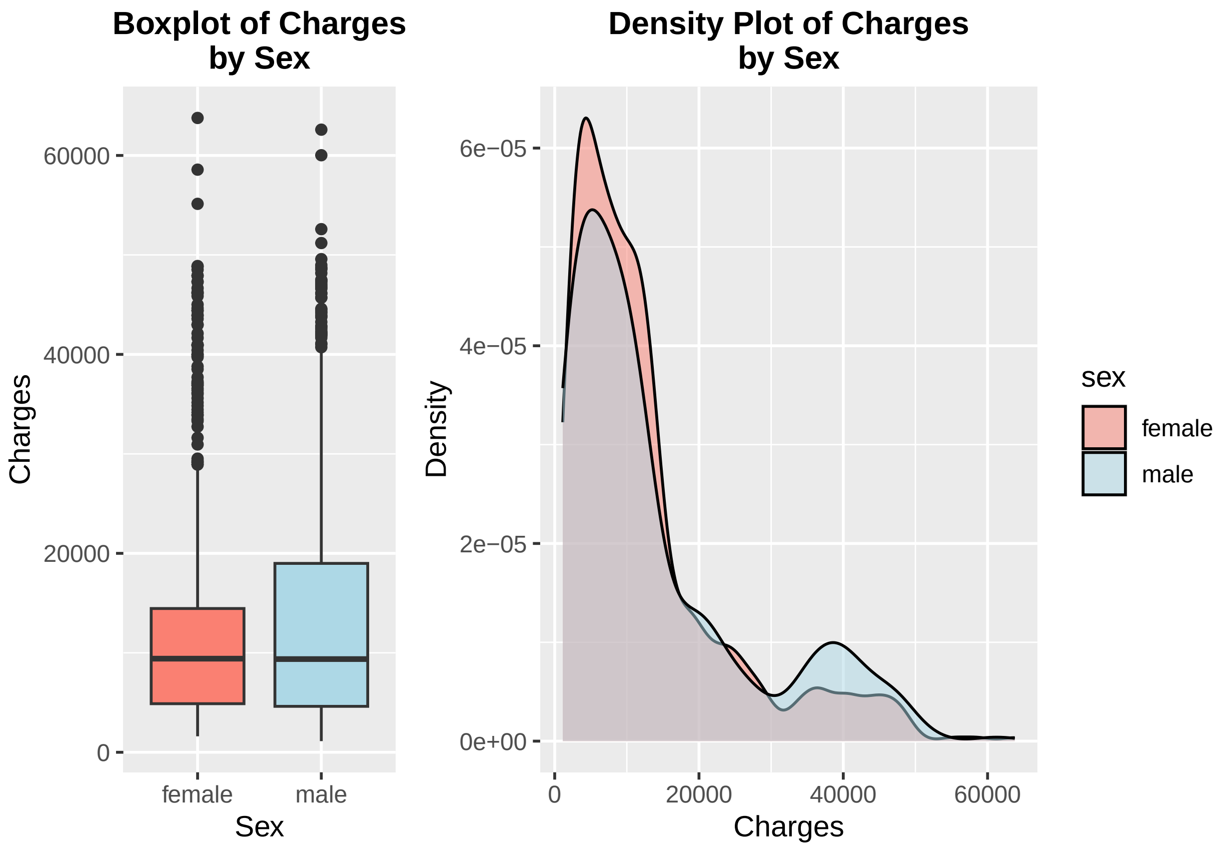
```
# Histogram: Distribution of Charges
g1 <- ggplot(df, aes(x = charges)) +
  geom_histogram(fill = 'slateblue') +
  theme(legend.position = 'none') +
  ggtitle("Distribution of Charges") +
  theme(plot.title = element_text(colour = 'black',
                                  face ='bold',
                                  size = 14,
                                  hjust = 0.5,
                                  vjust = 0.5)) +
  xlab('Charges') +
  ylab('')

# Boxplot: Charges by Number of Children
```

```r
g2 <- ggplot(df, aes(x = factor(children),
                     y = charges,
                     fill = factor(children))) +
  geom_boxplot() +  # Create boxplots
  theme(legend.position = 'none') +
  ggtitle("Charges by No. Children") +
  theme(plot.title = element_text(face = 'bold',
                                  colour = 'black',
                                  size = 12,
                                  hjust = 0.5, vjust = 0.5)) +
  xlab('Children') +
  ylab('Charges')

# Barplot: Count by Number of Children
g3 <- ggplot(df, aes(x = children, fill = factor(children))) +
  geom_bar() +
  theme(legend.position = 'none') +
  ggtitle("Countplot by No. Children") +
  theme(plot.title = element_text(colour = 'black',
                                  face ='bold',
                                  size = 12,
                                  hjust = 0.5, vjust = 0.5)) +
  xlab('Children') +  # Label the x-axis
  ylab('Count')  # Label the y-axis

# Arrange the plots
grid.arrange(
  g2,
  arrangeGrob(g1, g3, ncol = 2),
  nrow = 2
)
```

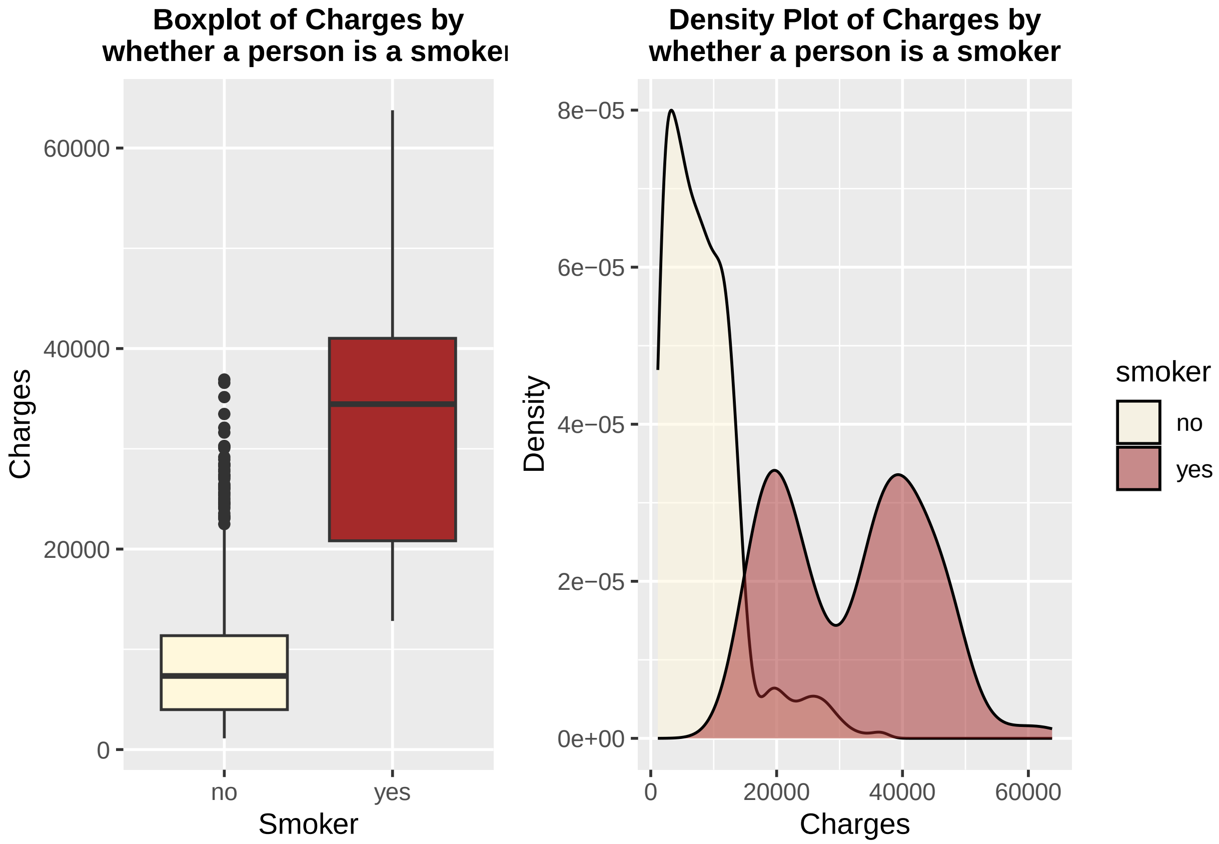
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
# Density Plot: Charges by Sex
g1 <- ggplot(data = df, aes(x = charges, fill = sex)) +
  geom_density(alpha = 0.5) +
  scale_fill_manual(values = c('salmon', 'lightblue')) +
  ggtitle("Density Plot of Charges\nby Sex") +
  theme(
    plot.title = element_text(colour = 'black',
                              face ='bold',
                              size = 12,
                              hjust = 0.5, vjust = 0.5)
  ) +
  xlab('Charges') +
  ylab('Density')

# Boxplot: Charges by Sex
g2 <- ggplot(data = df, aes(x = factor(sex),
                            y = charges,
                            fill = sex)) +
  geom_boxplot() +  # Create boxplots
  scale_fill_manual(values = c('salmon', 'lightblue')) +
  ggtitle("Boxplot of Charges\nby Sex") +
  theme(
    plot.title = element_text(colour = 'black',
                              face ='bold',
                              size = 12,
                              hjust = 0.5, vjust = 0.5),
    legend.position = 'none'
  ) +
  xlab('Sex') +
  ylab('Charges')

# Arrange the plots side by side
grid.arrange(g2, g1, ncol = 2, widths = c(0.5, 1))
```

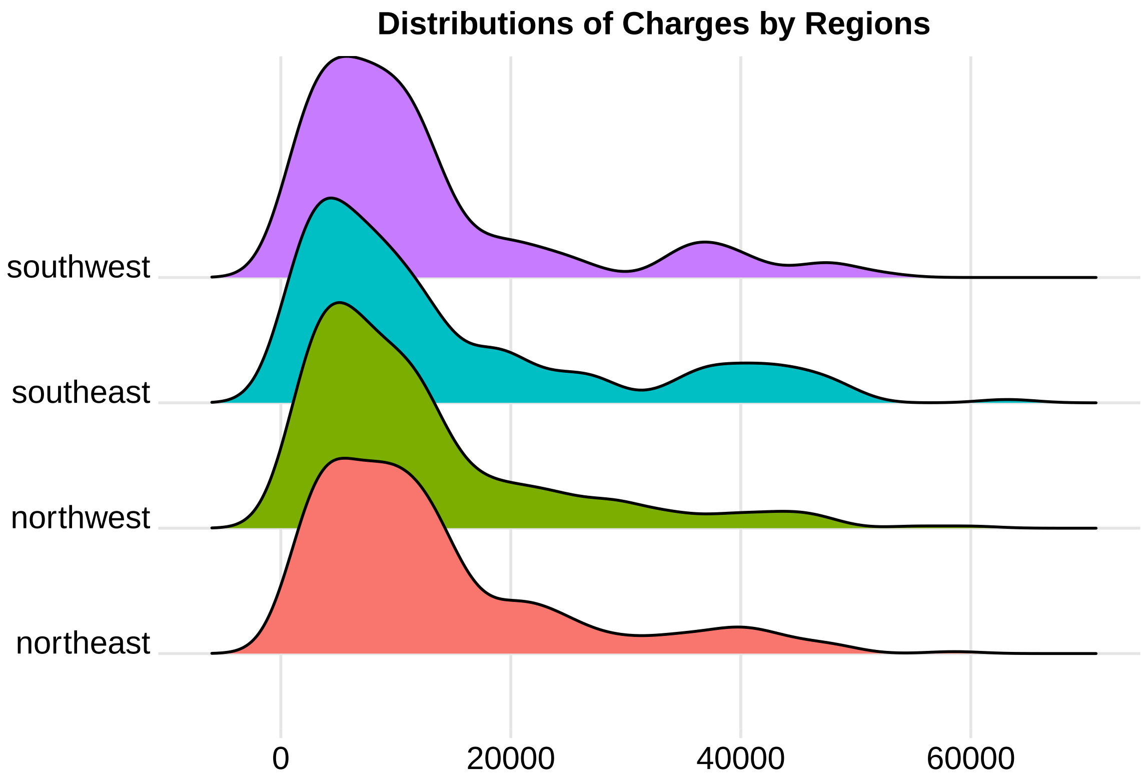


```
# Density Plot: Charges by Smoking Status
g1 <- ggplot(data = df, aes(x = charges, fill = smoker)) +
  geom_density(alpha = 0.5) +  # Create a density plot
  scale_fill_manual(values = c('cornsilk1', 'brown')) +
  ggtitle("Density Plot of Charges by\nwhether a person is a smoker") +
  theme(
    plot.title = element_text(colour = 'black',
                              face ='bold',
                              size = 11,
                              hjust = 0.5, vjust = 0.5)
  ) +
  xlab('Charges') +
  ylab('Density')

# Boxplot: Charges by Smoking Status
g2 <- ggplot(data = df, aes(x = factor(smoker),
                            y = charges,
                              fill = smoker)) +
  geom_boxplot() +
  scale_fill_manual(values = c('cornsilk1', 'brown')) +
  ggtitle("Boxplot of Charges by\nwhether a person is a smoker") +
  theme(
    plot.title = element_text(colour = 'black',
                              face ='bold',
                              size = 11,
                              hjust = 0.5, vjust = 0.5),
    legend.position = 'none'
  ) +
  xlab('Smoker') +
  ylab('Charges')

grid.arrange(g2, g1, ncol = 2, widths = c(0.7, 1))
```

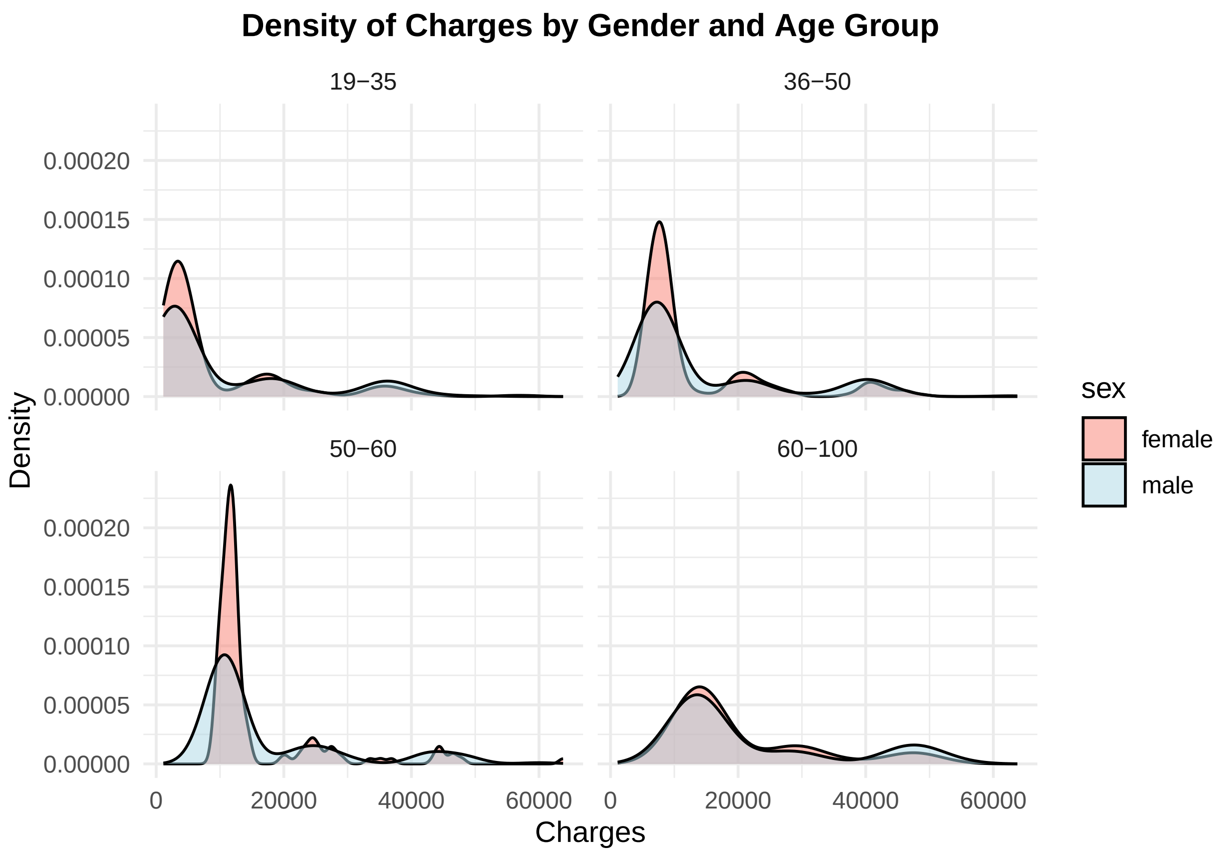


```
# Density Ridges Plot: Distributions of Charges by Region
ggplot(df, aes(x = charges,
               y = region,
               fill = region)) +
  geom_density_ridges() +
  theme_ridges() +
  theme(
    legend.position = 'none',
    plot.title = element_text(size = 12,
                              face = 'bold',
                              hjust = 0.5)
  ) +
  ggtitle('Distributions of Charges by Regions') +
  xlab('') +
  ylab('')
```

```
## Picking joint bandwidth of 2370
```



```
# Density Plot: Charges by Gender and Age Group
ggplot(df, aes(x = charges, fill = sex)) +
  geom_density(alpha = 0.5) +
  facet_wrap(~age_group) +
  scale_fill_manual(values = c('salmon', 'lightblue')) +
  ggtitle("Density of Charges by Gender and Age Group") +
  xlab('Charges') +
  ylab('Density') +
  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold",
                              hjust = 0.5,
                              size=12)
  )
```

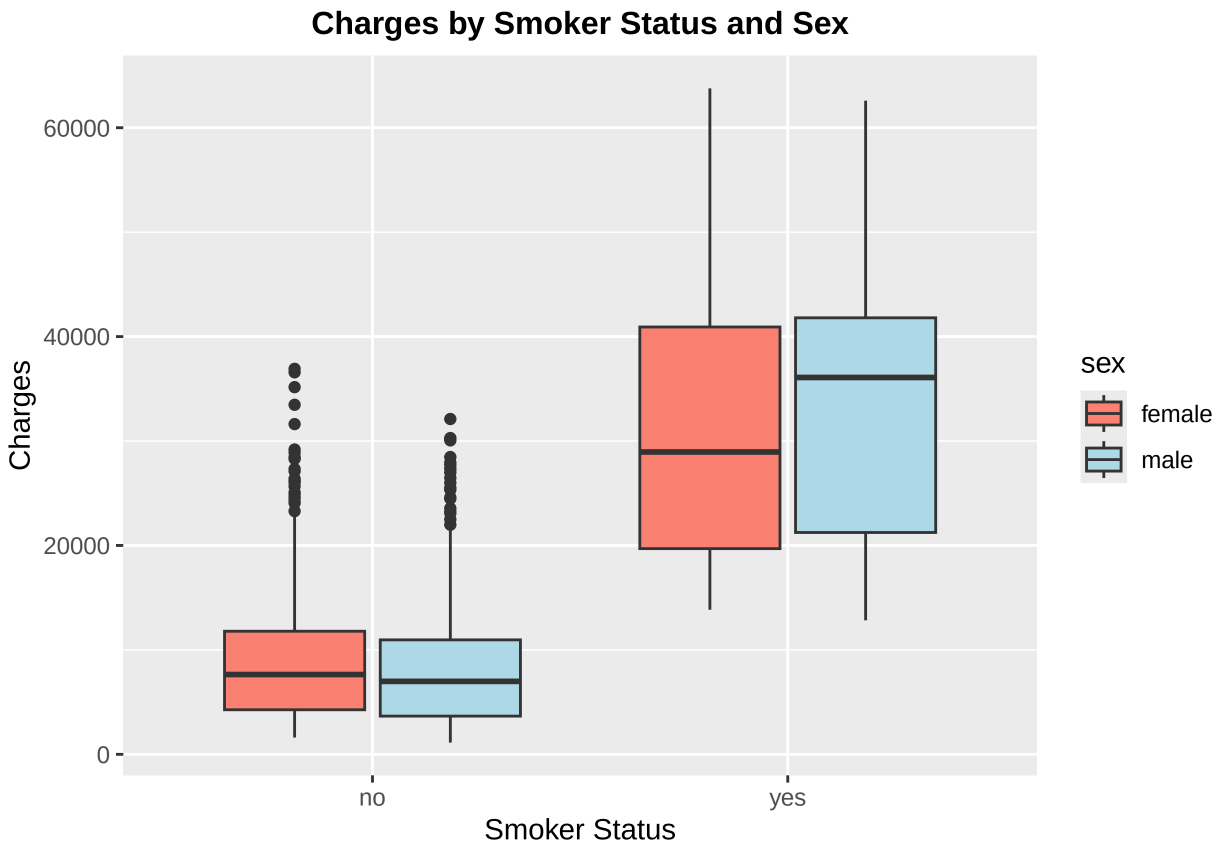


```
# Calculate mean charges by smoker status and gender
charges_by_smoker_gender <- df %>%
  group_by(smoker, sex) %>%
  # Calculate mean charges and drop grouping
  summarize(mean_charges = mean(charges), .groups = 'drop')

# Print the summarized data
print(charges_by_smoker_gender)
```

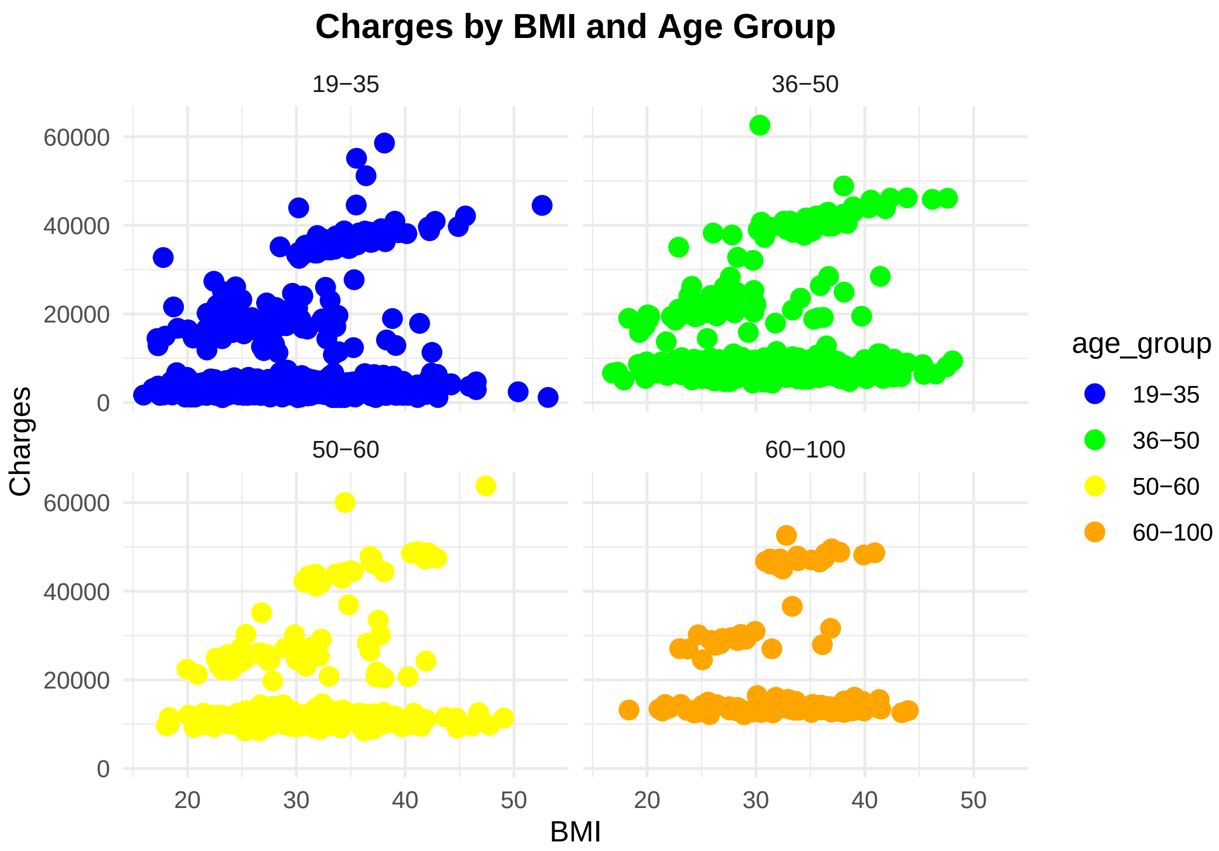
```
## # A tibble: 4 x 3
##   smoker sex    mean_charges
##   <chr>  <chr>         <dbl>
## 1 no     female        8762.
## 2 no     male          8087.
## 3 yes    female       30679.
## 4 yes    male         33042.
```

```
# Boxplot: Charges by Smoker Status and Gender
ggplot(df, aes(x = smoker, y = charges, fill = sex)) +
  geom_boxplot() +
  scale_fill_manual(values = c('salmon', 'lightblue')) +
  ggtitle("Charges by Smoker Status and Sex") +
  theme(
    plot.title = element_text(size = 12,
                              face = 'bold',
```

```r
                              hjust = 0.5)
  ) +
  xlab('Smoker Status') +
  ylab('Charges')
```



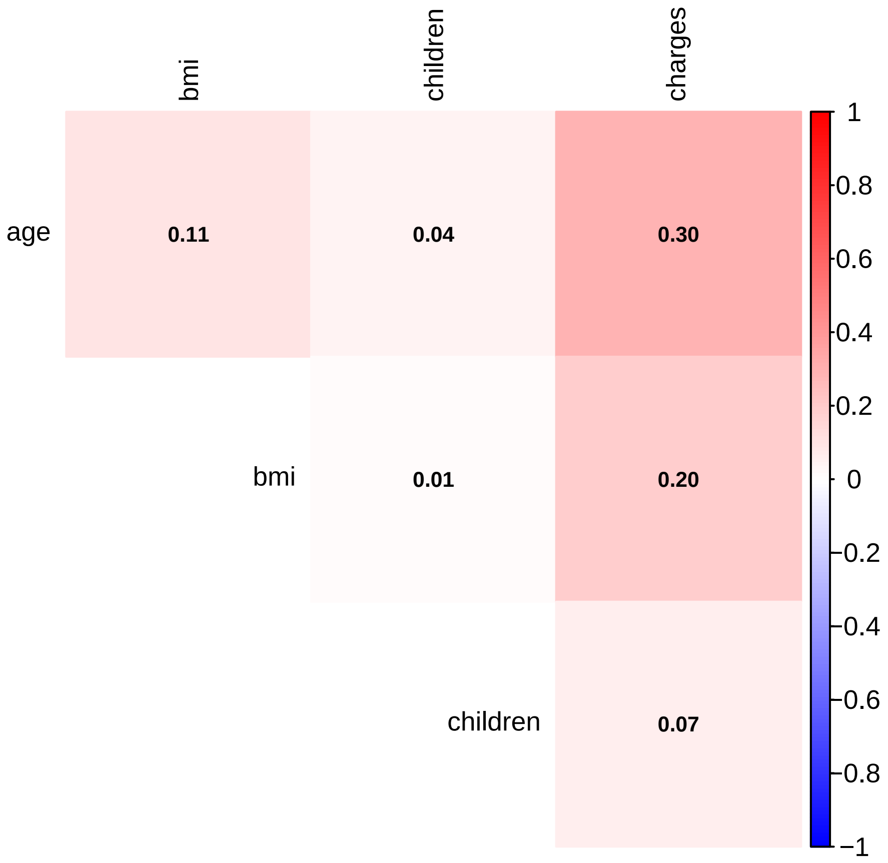
```r
# Scatter Plot: Charges by BMI and Age Group
ggplot(df, aes(x = bmi, y = charges, color = age_group)) +
  geom_point(size = 3) +
  facet_wrap(~age_group) +
  scale_color_manual(values = c('blue', 'green', 'yellow', 'orange')) +
  ggtitle("Charges by BMI and Age Group") +
  xlab("BMI") +
  ylab("Charges") +
  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold", hjust = 0.5)
  )
```



```r
# Select only numeric columns from the dataframe
df_numeric <- df[sapply(df, is.numeric)]

# Calculate the correlation matrix for numeric columns
corr_matrix <- cor(df_numeric, use = 'pairwise.complete.obs')

# Print the correlation matrix
corr_matrix
```

```
##                 age       bmi   children    charges
## age       1.0000000 0.1092719 0.04246900 0.29900819
## bmi       0.1092719 1.0000000 0.01275890 0.19834097
## children  0.0424690 0.0127589 1.00000000 0.06799823
## charges   0.2990082 0.1983410 0.06799823 1.00000000
```

```r
# Visualize the correlation matrix with customized formatting
corrplot(corr_matrix,
         method = 'color',
         type = 'upper',
         tl.cex = 0.8,
         tl.col = 'black',
         number.cex = 0.7,
         col = colorRampPalette(c('blue', 'white', 'red'))(200),
         addCoef.col = 'black',
         diag = FALSE)
```



```r
# Convert categorical variables to factors
df$sex <- as.factor(df$sex)
df$smoker <- as.factor(df$smoker)
df$region <- as.factor(df$region)
df$age_group <- as.factor(df$age_group)

# Split the dataset into training and testing sets
split <- initial_split(df, prop = 0.8)  # 80% training data, 20% testing data
train_data <- training(split)  # Training data
test_data <- testing(split)    # Testing data

# Fit a linear regression model
model <- lm(charges ~ bmi + age_group + sex + region + smoker, data = train_data)

# Display the summary of the linear regression model
summary(model)
```

```
##
## Call:
## lm(formula = charges ~ bmi + age_group + sex + region + smoker,
##     data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12138.3  -3283.2   -863.7   2088.6  30669.0
##
```

```
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -5594.13    1063.21  -5.262 1.73e-07 ***
## bmi                363.28      33.43  10.865  < 2e-16 ***
## age_group36-50    3836.28     456.40   8.406  < 2e-16 ***
## age_group50-60    7942.92     514.48  15.439  < 2e-16 ***
## age_group60-100  10361.39     727.99  14.233  < 2e-16 ***
## sexmale           -299.76     381.25  -0.786   0.4319
## regionnorthwest   -566.69     548.62  -1.033   0.3019
## regionsoutheast   -993.62     544.23  -1.826   0.0682 .
## regionsouthwest  -1011.32     542.91  -1.863   0.0628 .
## smokeryes        23945.95     468.19  51.146  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 6200 on 1060 degrees of freedom
## Multiple R-squared:  0.7496, Adjusted R-squared:  0.7475
## F-statistic: 352.6 on 9 and 1060 DF,  p-value: < 2.2e-16
```

```
# Make predictions on the test data
predictions <- predict(model, newdata = test_data)

# Add the predictions to the test data
test_data$predicted_charges <- predictions

# Display the first few rows of the test data with predictions
head(test_data)
```

```
##   age    sex    bmi children smoker    region   charges age_group
## 1  28   male 33.000        3     no southeast  4449.462     19-35
## 2  33   male 22.705        0     no northwest 21984.471     19-35
## 3  31 female 25.740        0     no southeast  3756.622     19-35
## 4  60 female 25.840        0     no northwest 28923.137    60-100
## 5  19   male 20.425        0     no northwest  1625.434     19-35
## 6  26   male 20.800        0     no southwest  2302.300     19-35
##   predicted_charges
## 1         5100.6555
## 2         1787.6358
## 3         2763.0163
## 4        13587.6581
## 5          959.3630
## 6          650.9669
```

```
# Calculate Mean Absolute Error (MAE)
mae <- mean(abs(test_data$charges - test_data$predicted_charges))
mae
```

```
## [1] 4162.317
```

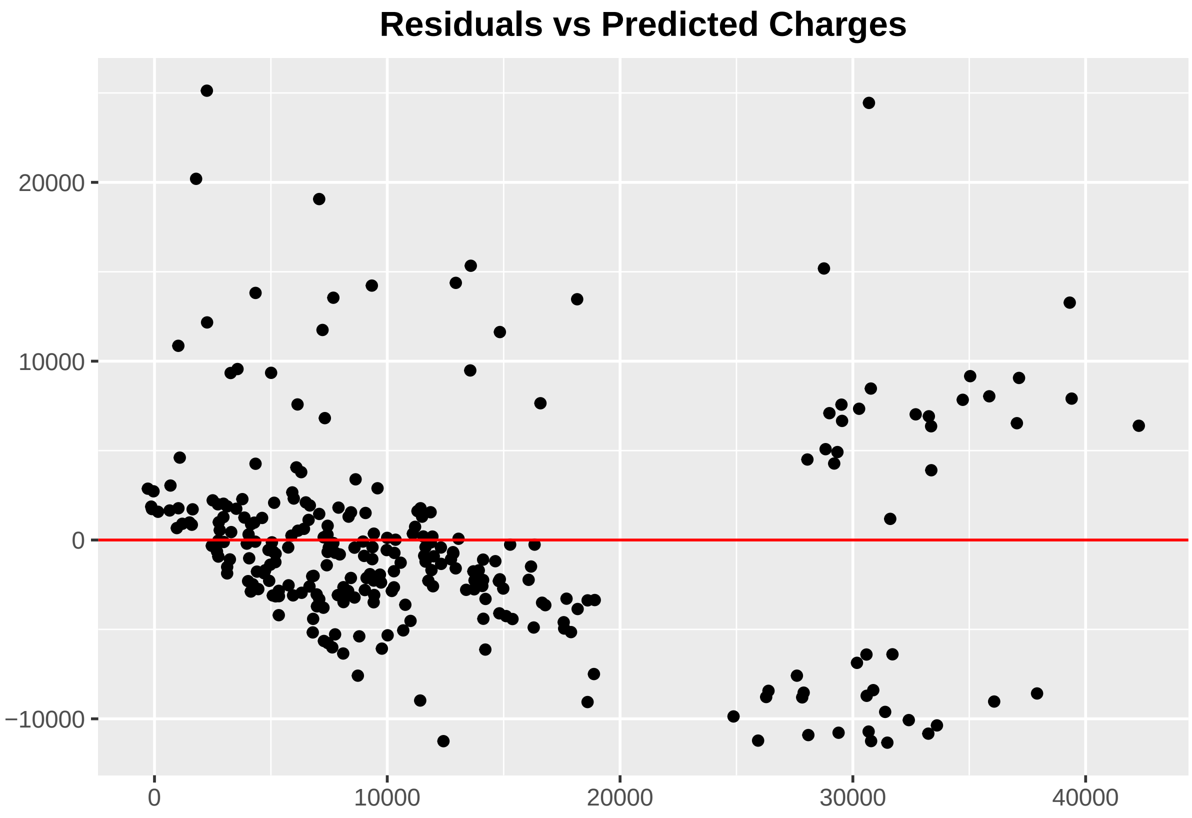
```
# Calculate Mean Squared Error (MSE)
mse <- mean((test_data$charges - test_data$predicted_charges)^2)
mse
```

```
## [1] 34970585
```

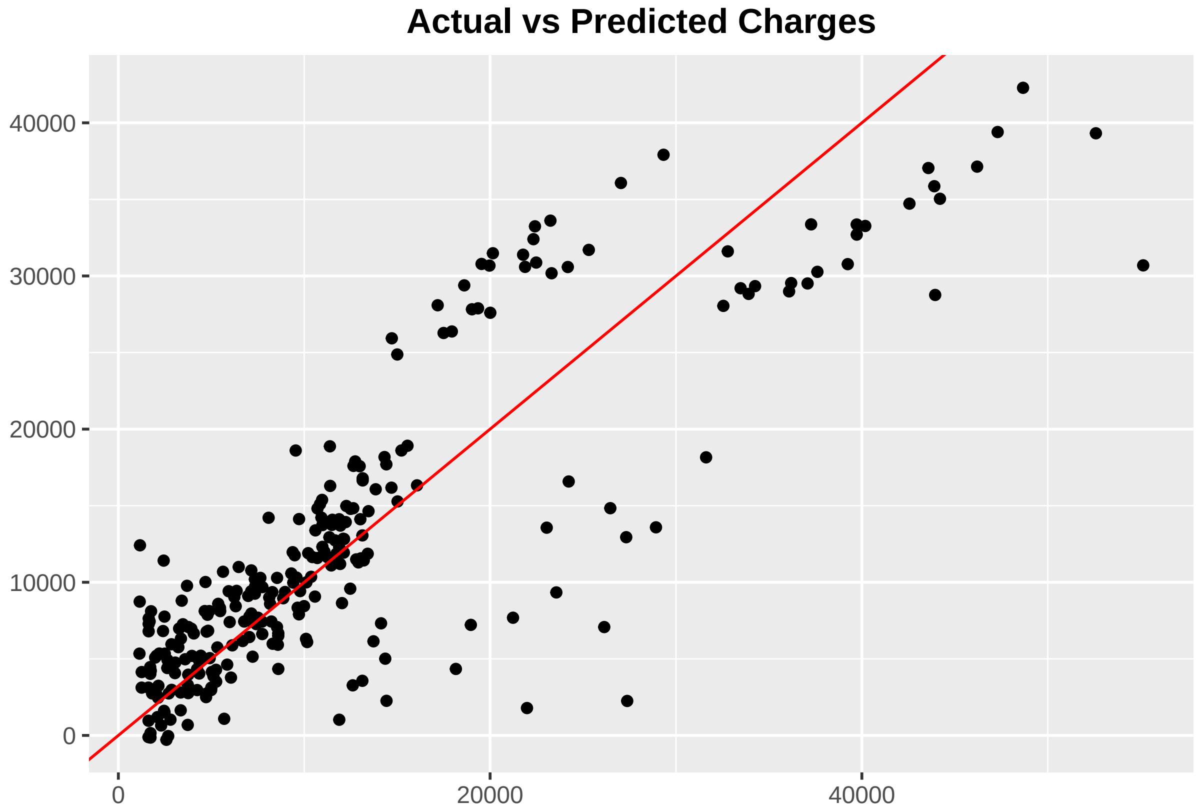
```
# Calculate Root Mean Squared Error (RMSE)
rmse <- sqrt(mse)
rmse
```

```
## [1] 5913.593
```

```
# Plot residuals versus predicted charges
ggplot(test_data, aes(x = predicted_charges,
                      y = charges - predicted_charges)) +
  geom_point() +
  geom_hline(yintercept = 0, color = "red") +
  ggtitle("Residuals vs Predicted Charges") +
  xlab("") +
  ylab("") +
  theme(
    plot.title = element_text(face = "bold",
                              hjust = 0.5)
  )
```



```
# Plot actual vs predicted values
ggplot(test_data, aes(x = charges, y = predicted_charges)) +
geom_point() +
geom_abline(intercept = 0, slope = 1, color = "red") +
ggtitle("Actual vs Predicted Charges") +
xlab("") +
ylab("") +
   theme(
    plot.title = element_text(face = "bold", hjust = 0.5)
  )
```


Actual vs Predicted Charges