Misinformation Qualitative Analysis

Kendall Beaver

2025-03-24

```
# Filtered data set to columns I need

# Columns I need:

# PARTICIPANT ID
# GROUP
# CHANGED_VIEWS_QQ1
# CHANGED_LENGTH_QQ1
# CHANGED_VIEWS_QQ2
# CHANGED_LENGTH_QQ2
# GCBS PRE SCORE
# GCBS POST SCORE
# PRE & POST SCORE
# GCBS CHANGE
# MIST-20 PRE SCORE
# MIST-20 POST SCORE
# MIST-20 CHANGE
# ITMIST PRE SCORE
# ITMIST POST SCORE
# ITMIST CHANGE
# UNDERSTANDABILITY
# SENTIMENT
# ADDITIONAL THOUGHTS

misinfo_filtered <- misinfo %>%
  select(`PARTICIPANT ID`, GROUP, CHANGED_VIEWS_QQ1, CHANGED_LENGTH_QQ1,
         CHANGED_VIEWS_QQ2, CHANGED_LENGTH_QQ2, `GCBS PRE SCORE`,
         `GCBS POST SCORE`, `PRE & POST SCORE`, `GCBS CHANGE`,
         `MIST-20 PRE SCORE`, `MIST-20 POST SCORE`, `MIST-20 CHANGE`,
         `ITMIST PRE SCORE`, `ITMIST POST SCORE`, `ITMIST CHANGE`,
         UNDERSTANDABILITY, SENTIMENT, `ADDITIONAL THOUGHTS`)
```

```
# Turn "character" columns into factors

# Columns that need to be turned into factors:

# GROUP
#   "Test Group (Visual)"
#   "Control Group (Visual)"
#   "Test Group (Audio)"
#   "Control Group (Audio)"
```

```r
# CHANGED_VIEWS_QQ1
#   "View stayed about the same"
#   "View had some/moderate change"
#   "View significantantly changed"

# CHANGED_LENGTH_QQ1
#   "Comment length became slightly or significantly shorter"
#   "Comment length became slightly or significantly longer"
#   "Comment length stayed about the same"

# CHANGED_VIEWS_QQ2

# CHANGED_LENGTH_QQ2

# UNDERSTANDABILITY

# SENTIMENT

# `ADDITIONAL THOUGHTS`

# Mapping categorical variables to numerical values
misinfo_filtered <- misinfo_filtered %>%
  mutate(
    GROUP = case_when(
      GROUP == "Test Group (Visual)" ~ 1,
      GROUP == "Control Group (Visual)" ~ 2,
      GROUP == "Test Group (Audio)" ~ 3,
      GROUP == "Control Group (Audio)" ~ 4
    ),
    CHANGED_VIEWS_QQ1 = case_when(
      CHANGED_VIEWS_QQ1 == "View stayed about the same" ~ 1,
      CHANGED_VIEWS_QQ1 == "View had some/moderate change" ~ 2,
      CHANGED_VIEWS_QQ1 == "View significantly changed" ~ 3
    ),
    CHANGED_LENGTH_QQ1 = case_when(
      CHANGED_LENGTH_QQ1 == "Comment length became slightly or significantly shorter" ~ 1,
      CHANGED_LENGTH_QQ1 == "Comment length became slightly or significantly longer" ~ 2,
      CHANGED_LENGTH_QQ1 == "Comment length stayed about the same" ~ 3
    ),
    SENTIMENT = case_when(
      SENTIMENT == "Positive" ~ 1,
      SENTIMENT == "Negative" ~ 2,
      SENTIMENT == "Neutral" ~ 3
    )
  )

# Check the updated dataset
head(misinfo_filtered)
```

```
## # A tibble: 6 x 19
##   `PARTICIPANT ID` GROUP CHANGED_VIEWS_QQ1 CHANGED_LENGTH_QQ1 CHANGED_VIEWS_QQ2
##              <dbl> <dbl>             <dbl>              <dbl> <chr>
## 1                1     1                 1                  1 View stayed about~
```

```
## 2                2     2                 1                  1 View stayed about~
## 3                3     1                 1                  1 View stayed about~
## 4                4     2                 1                  2 View stayed about~
## 5                5     3                 2                  2 View stayed about~
## 6                6     3                 1                  1 View stayed about~
## # i 14 more variables: CHANGED_LENGTH_QQ2 <chr>, `GCBS PRE SCORE` <dbl>,
## #   `GCBS POST SCORE` <dbl>, `PRE & POST SCORE` <dbl>, `GCBS CHANGE` <dbl>,
## #   `MIST-20 PRE SCORE` <chr>, `MIST-20 POST SCORE` <dbl>,
## #   `MIST-20 CHANGE` <dbl>, `ITMIST PRE SCORE` <chr>,
## #   `ITMIST POST SCORE` <dbl>, `ITMIST CHANGE` <dbl>, UNDERSTANDABILITY <chr>,
## #   SENTIMENT <dbl>, `ADDITIONAL THOUGHTS` <chr>
```

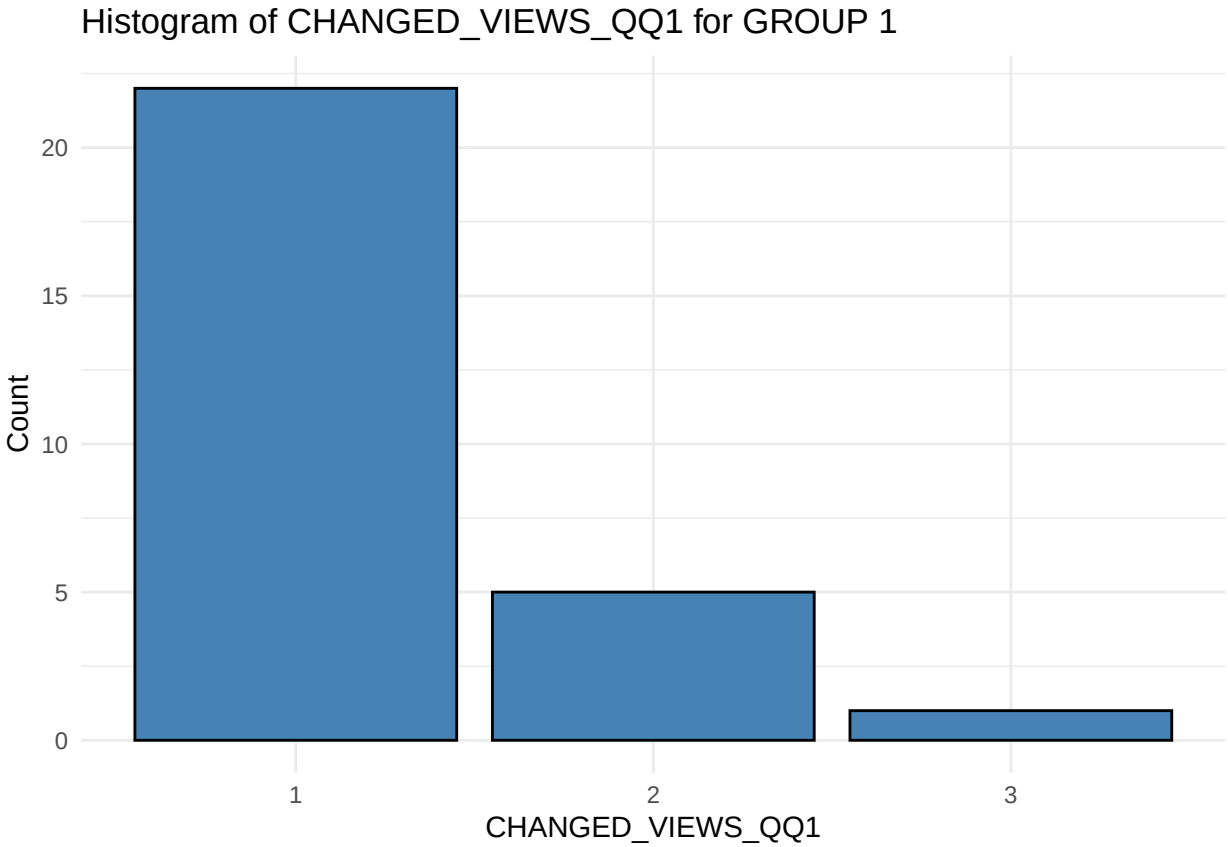
```r
# Check for missing values in the dataset
colSums(is.na(misinfo_filtered))
```

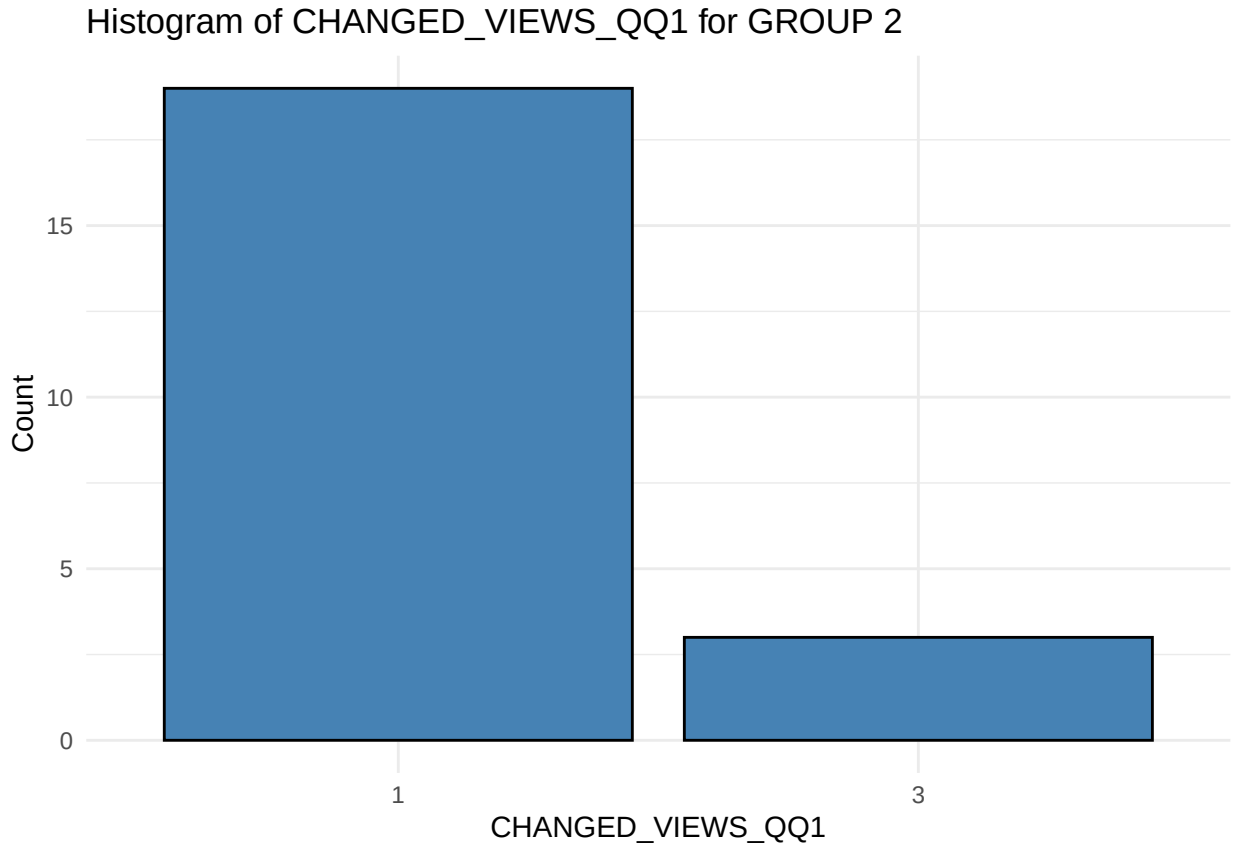
```
##      PARTICIPANT ID               GROUP   CHANGED_VIEWS_QQ1  CHANGED_LENGTH_QQ1
##                   0                   0                   0                   0
##   CHANGED_VIEWS_QQ2  CHANGED_LENGTH_QQ2      GCBS PRE SCORE     GCBS POST SCORE
##                   0                   0                   0                   0
##    PRE & POST SCORE         GCBS CHANGE   MIST-20 PRE SCORE  MIST-20 POST SCORE
##                   0                   0                   0                   0
##      MIST-20 CHANGE    ITMIST PRE SCORE   ITMIST POST SCORE       ITMIST CHANGE
##                   0                   0                   0                   0
##   UNDERSTANDABILITY           SENTIMENT ADDITIONAL THOUGHTS
##                   0                   0                   0
```

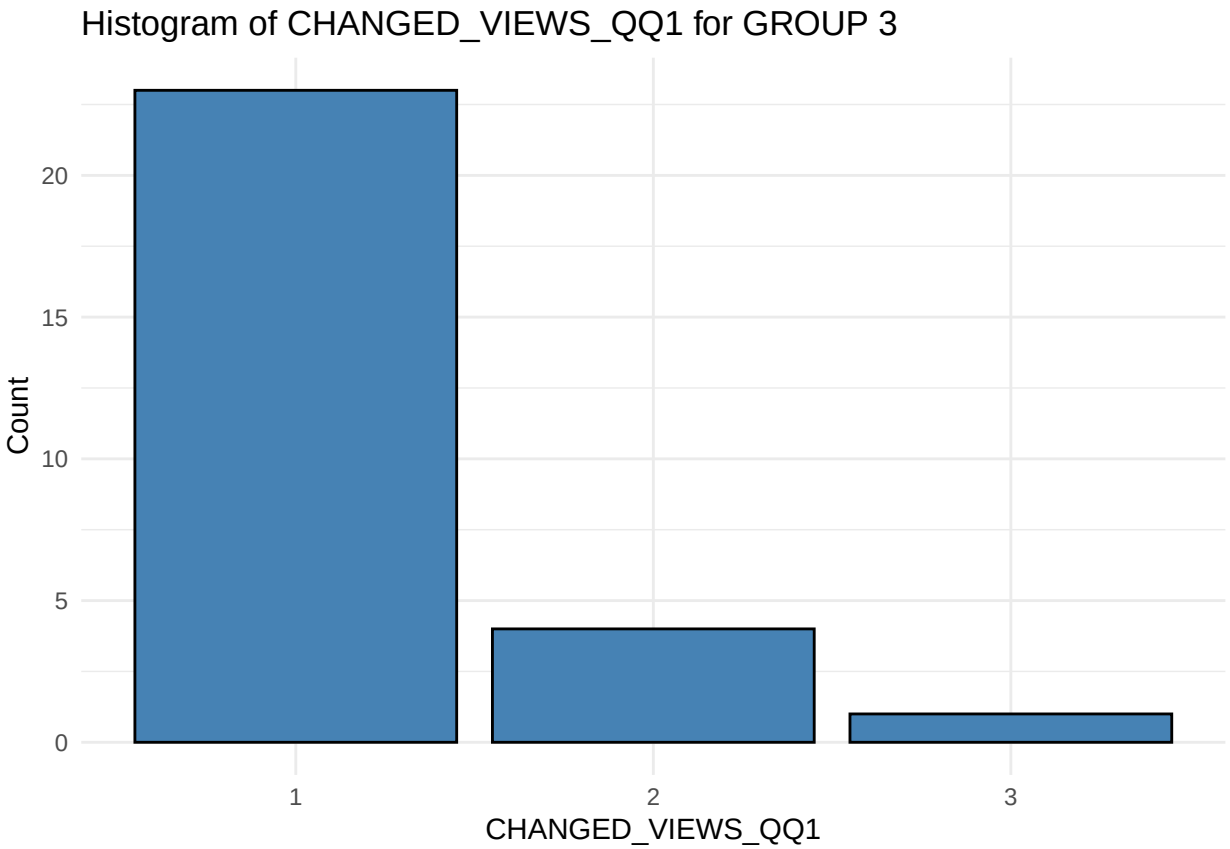
```r
# Ensure CHANGED_VIEWS_QQ1 is treated as a factor
misinfo_filtered$CHANGED_VIEWS_QQ1 <- as.factor(misinfo_filtered$CHANGED_VIEWS_QQ1)
```

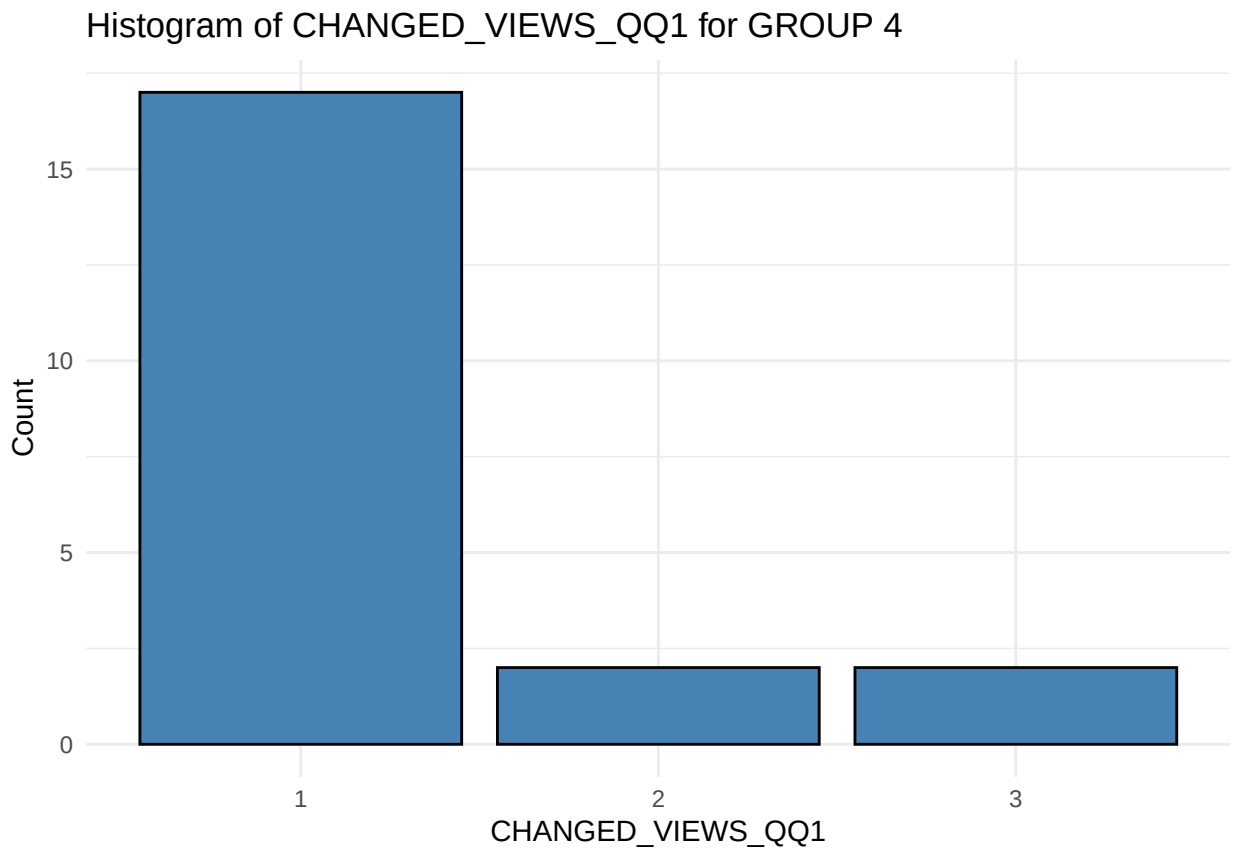
Histograms

```r
# HISTOGRAMS

# Ensure CHANGED_VIEWS_QQ1 is treated as a factor
misinfo_filtered$CHANGED_VIEWS_QQ1 <- as.factor(misinfo_filtered$CHANGED_VIEWS_QQ1)

# Loop through each GROUP value and create histograms
for (group_value in unique(misinfo_filtered$GROUP)) {
  p <- ggplot(misinfo_filtered %>% filter(GROUP == group_value), aes(x = CHANGED_VIEWS_QQ1)) +
    geom_bar(fill = "steelblue", color = "black") +
    labs(title = paste("Histogram of CHANGED_VIEWS_QQ1 for GROUP", group_value),
         x = "CHANGED_VIEWS_QQ1",
         y = "Count") +
    theme_minimal()

  print(p)  # Ensure the plot is displayed inside the loop
}
```


Histogram of CHANGED_VIEWS_QQ1 for GROUP 1
Count
20
15
10
5
0
1
2
3
CHANGED_VIEWS_QQ1


Histogram of CHANGED_VIEWS_QQ1 for GROUP 2
Count
15
10
5
0
1
3
CHANGED_VIEWS_QQ1


Histogram of CHANGED_VIEWS_QQ1 for GROUP 3
Count
20
15
10
5
0
1
2
3
CHANGED_VIEWS_QQ1

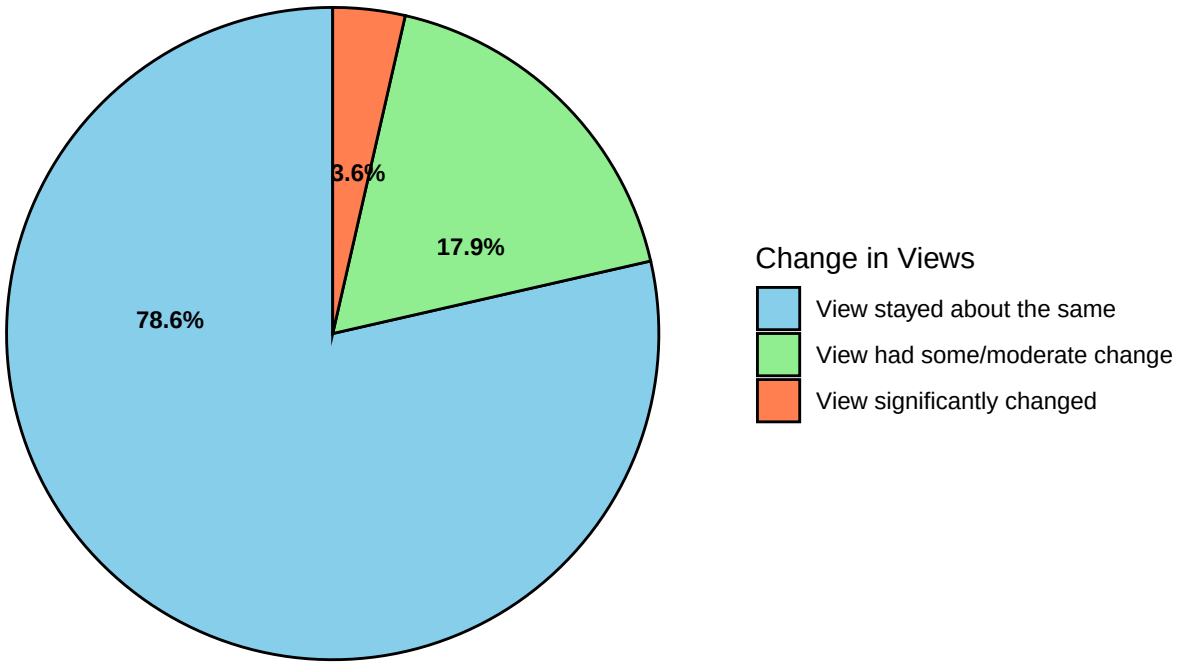


Pie Charts

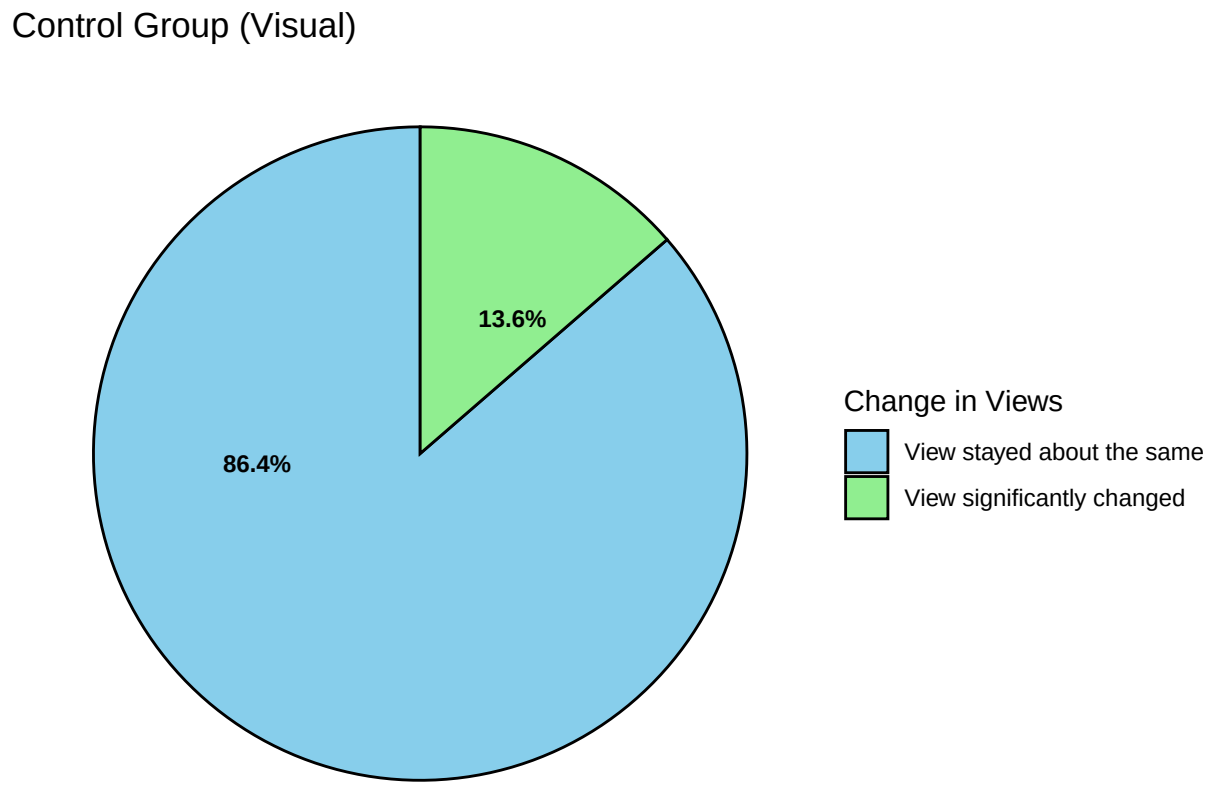
```
library(ggplot2)
library(dplyr)

# Create an empty list to store pie charts
pie_charts <- list()

# Define a vector to map numeric values to descriptions for CHANGED_VIEWS_QQ1
view_labels <- c("1" = "View stayed about the same",
                 "2" = "View had some/moderate change",
                 "3" = "View significantly changed")

# Custom colors for each section (You can change these to any colors you prefer)
custom_colors <- c("skyblue", "lightgreen", "coral")

# Define a vector to map numeric GROUP values to their corresponding titles
group_titles <- c(
  "1" = "Test Group (Visual)",
  "2" = "Control Group (Visual)",
  "3" = "Test Group (Audio)",
  "4" = "Control Group (Audio)"
)
```

```r
# Loop through each GROUP value (numeric) and create pie charts
for (group_value in unique(misinfo_filtered$GROUP)) {

  # Convert group_value to a character (because group_titles uses character keys)
  group_name <- as.character(group_value)

  # Prepare data for pie chart
  pie_data <- misinfo_filtered %>%
    filter(GROUP == group_value) %>%
    count(CHANGED_VIEWS_QQ1) %>%
    mutate(percentage = n / sum(n) * 100,
           label = paste0(round(percentage, 1), "%"))  # Create labels with percentage signs

  # Ensure CHANGED_VIEWS_QQ1 is a factor
  pie_data$CHANGED_VIEWS_QQ1 <- factor(pie_data$CHANGED_VIEWS_QQ1, levels = c("1", "2", "3"))

  # Generate pie chart with custom colors and black borders
  p <- ggplot(pie_data, aes(x = "", y = n, fill = CHANGED_VIEWS_QQ1)) +
    geom_bar(stat = "identity", width = 1, color = "black") +  # Add black border around sections
    coord_polar(theta = "y") +
    geom_text(aes(label = label), position = position_stack(vjust = 0.7), color = "black", size = 3, fo
    labs(title = group_titles[group_name],  # Custom title based on group_name
         fill = "Change in Views") +  # Rename the legend
    scale_fill_manual(values = custom_colors, labels = view_labels) +  # Apply custom colors
    theme_void() +  # Remove unnecessary gridlines
    theme(legend.position = "right")  # Ensure the legend is visible

  # Store the plot in the list with the group value as the name
  pie_charts[[paste0("Group_", group_name)]] <- p
}

# View stored pie charts
pie_charts[["Group_1"]]  # View the pie chart for "Test Group (Visual)"
```

Test Group (Visual)

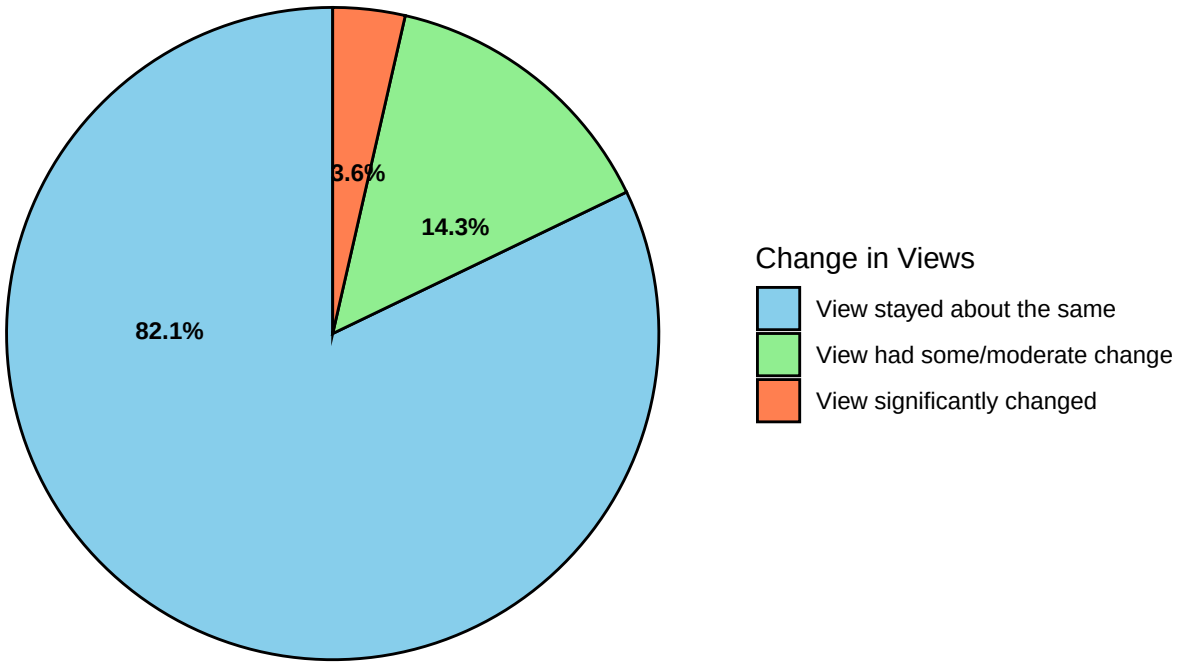



```
pie_charts[["Group_2"]]  # View the pie chart for "Control Group (Visual)"
```



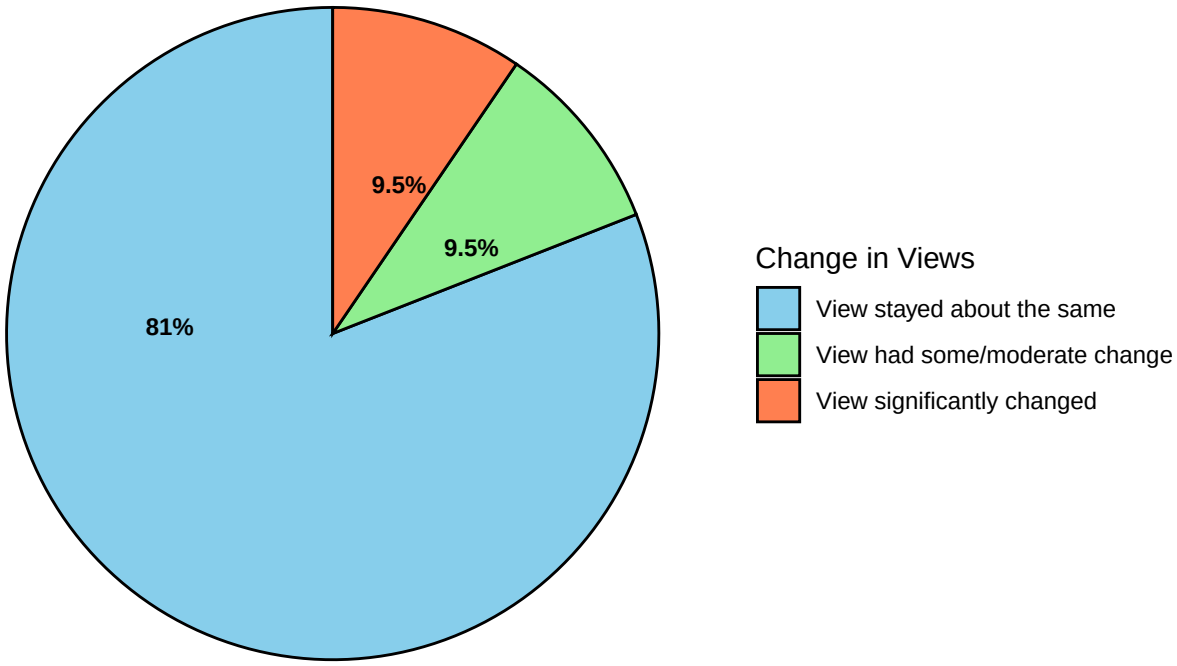

```
pie_charts[["Group_3"]]  # View the pie chart for "Test Group (Audio)"
```

Test Group (Audio)

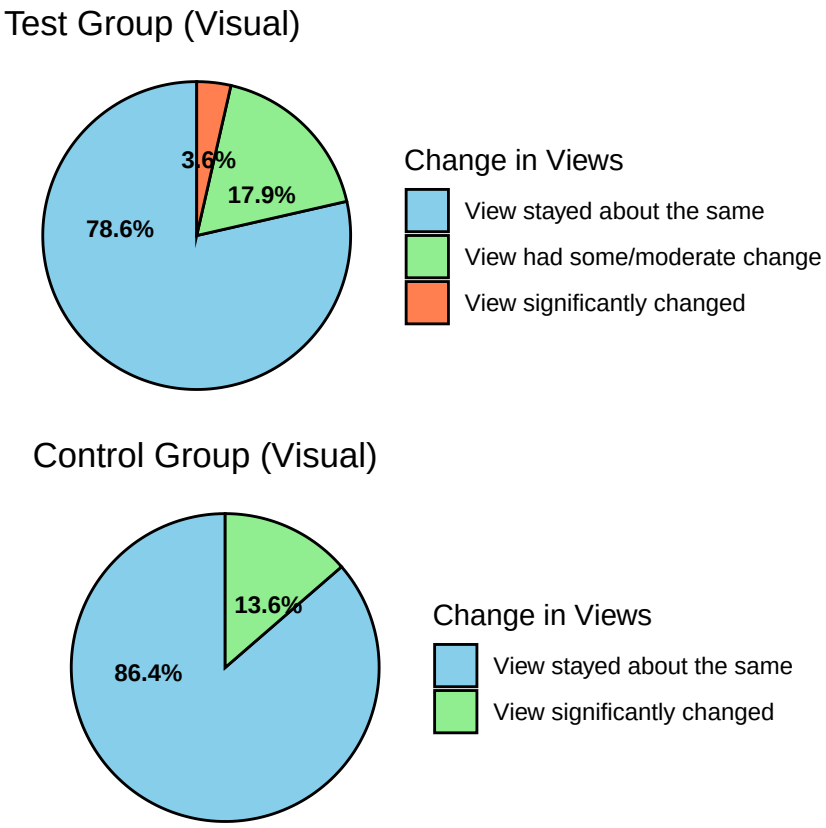



```
pie_charts[["Group_4"]]  # View the pie chart for "Control Group (Audio)"
```

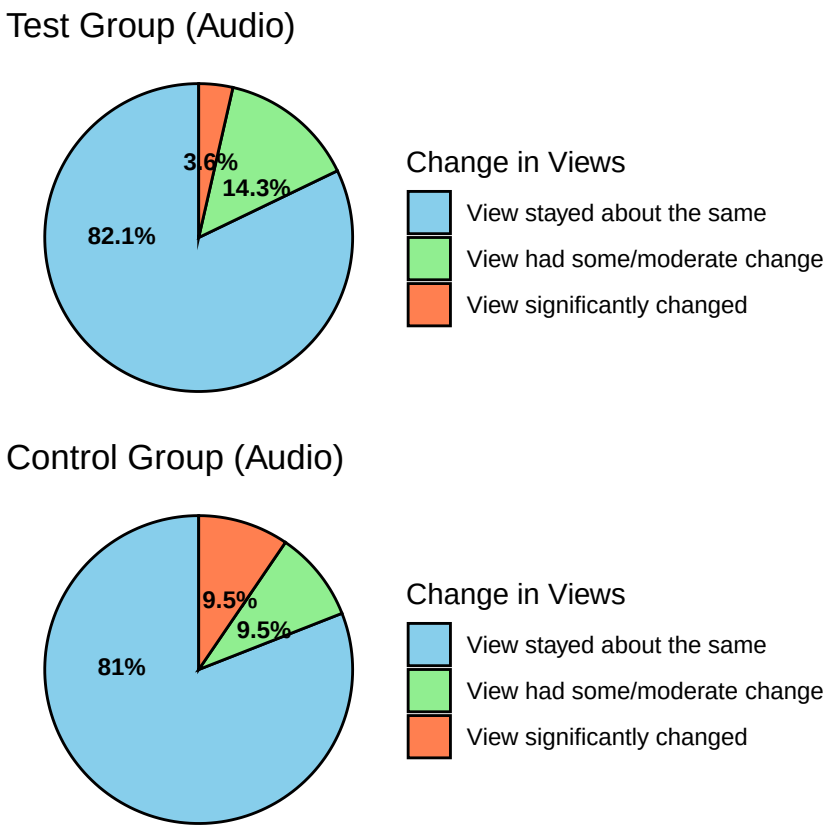
Control Group (Audio)




```
# Combine the first two and the last two pie charts in a 2x2 grid
grid.arrange(
  pie_charts[["Group_1"]],  # Test Group (Visual)
  pie_charts[["Group_2"]]  # Control Group (Visual)
  #pie_charts[["Group_3"]],  # Test Group (Audio)
  #pie_charts[["Group_4"]],  # Control Group (Audio)
  #nrow = 2  # Arrange in 2 rows
  #ncol = 1   # Arrange in 2 columns
)
```

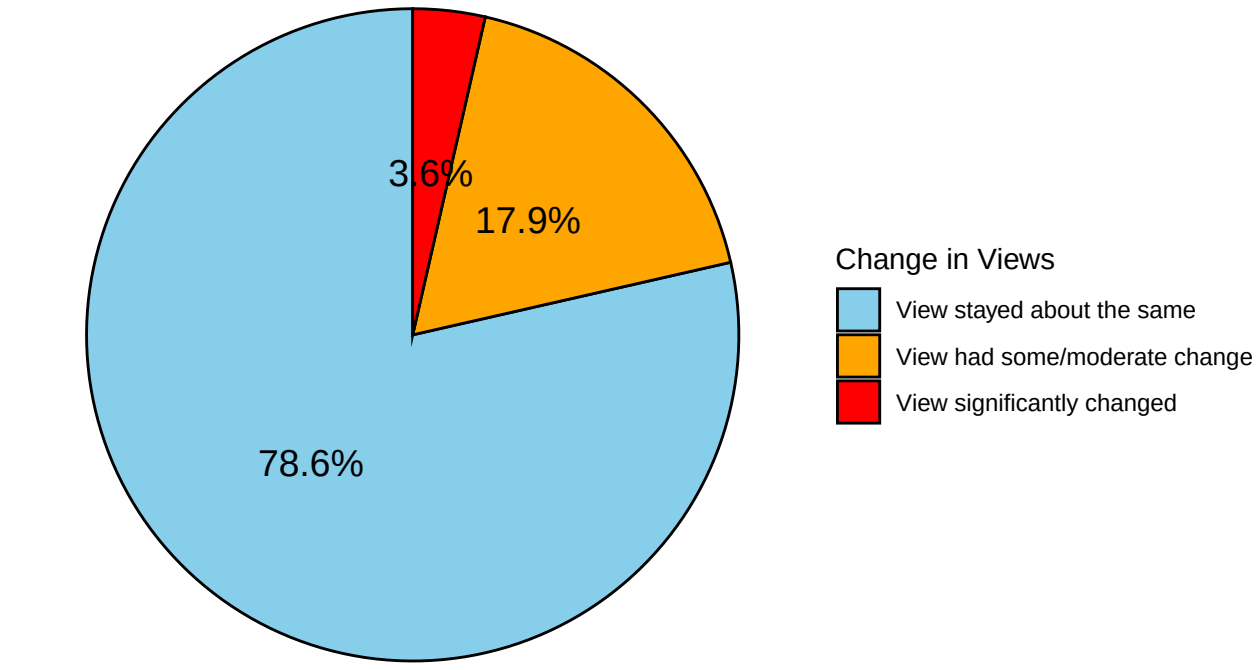



```
grid.arrange(
  pie_charts[["Group_3"]],  # Test Group (Audio)
  pie_charts[["Group_4"]]  # Control Group (Audio)
  #nrow = 2  # Arrange in 2 rows
  #ncol = 1   # Arrange in 2 columns
)
```

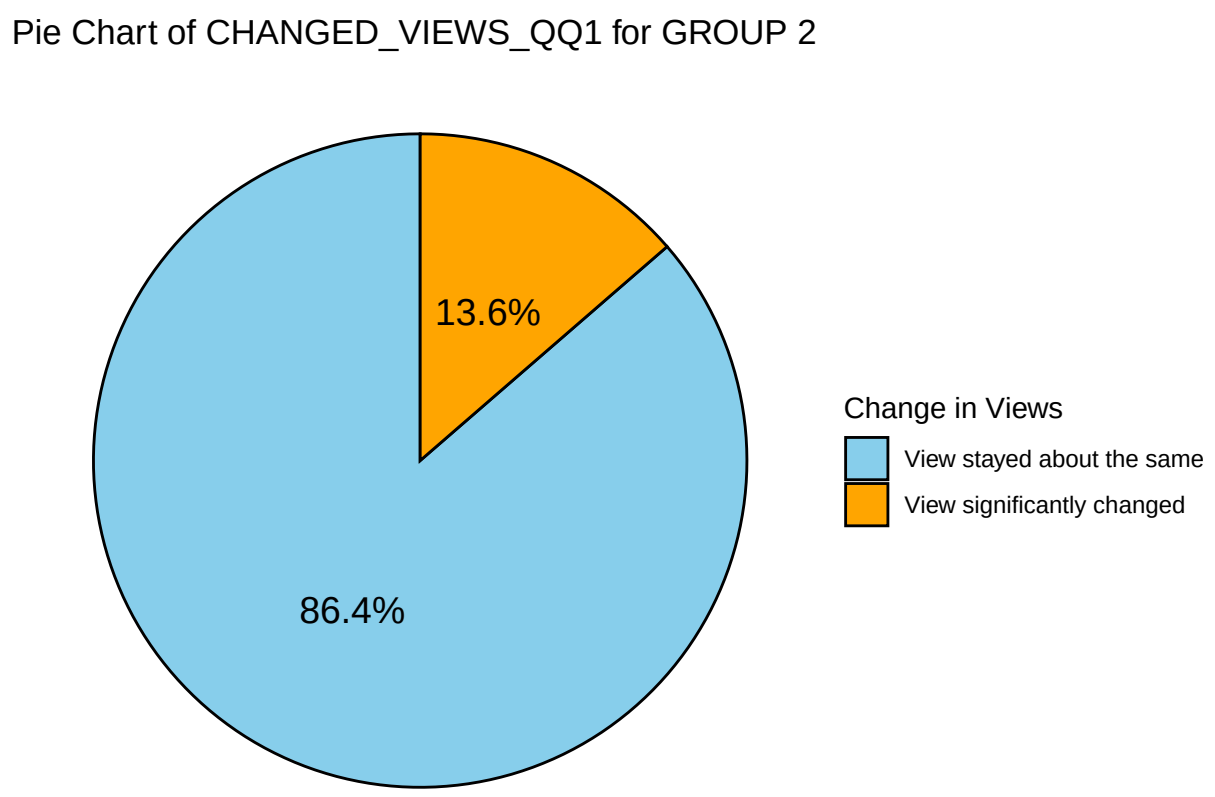



```
library(ggplot2)
library(dplyr)

# Create an empty list to store pie charts
pie_charts <- list()

# Define a vector to map numeric values to descriptions for CHANGED_VIEWS_QQ1
view_labels <- c("1" = "View stayed about the same",
                 "2" = "View had some/moderate change",
                 "3" = "View significantly changed")

# Custom colors for each section (You can change these to any colors you prefer)
custom_colors <- c("skyblue", "orange", "red")

# Loop through each GROUP value and create pie charts
for (group_value in unique(misinfo_filtered$GROUP)) {

  # Prepare data for pie chart
  pie_data <- misinfo_filtered %>%
    filter(GROUP == group_value) %>%
    count(CHANGED_VIEWS_QQ1) %>%
    mutate(percentage = n / sum(n) * 100,
           label = paste0(round(percentage, 1), "%"))  # Create labels with percentage signs

  # Ensure CHANGED_VIEWS_QQ1 is a factor
  pie_data$CHANGED_VIEWS_QQ1 <- factor(pie_data$CHANGED_VIEWS_QQ1, levels = c(1, 2, 3))
```

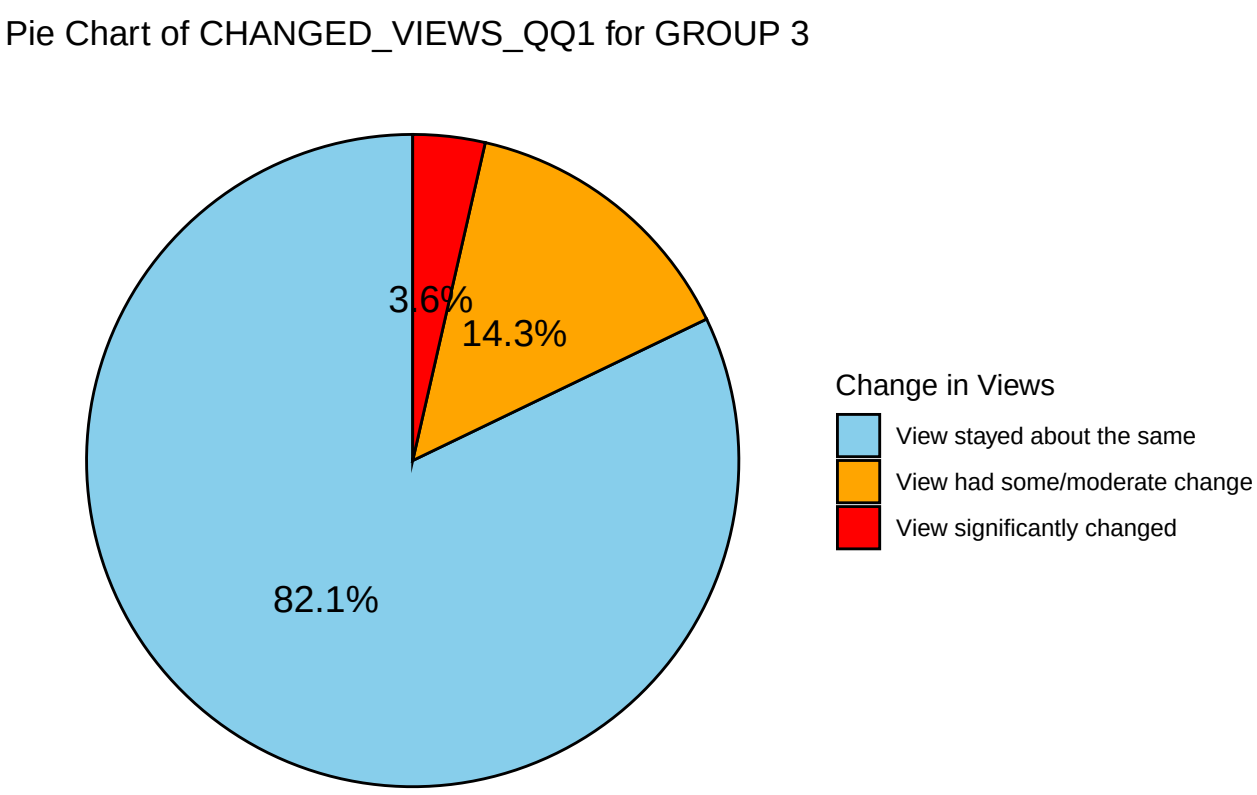
```r
  # Generate pie chart with custom colors and black borders
  p <- ggplot(pie_data, aes(x = "", y = n, fill = CHANGED_VIEWS_QQ1)) +
    geom_bar(stat = "identity", width = 1, color = "black") +  # Add black border around sections
    coord_polar(theta = "y") +
    geom_text(aes(label = label), position = position_stack(vjust = 0.5), color = "black", size = 5) +
    labs(title = paste("Pie Chart of CHANGED_VIEWS_QQ1 for GROUP", group_value),
         fill = "Change in Views") +  # Rename the legend
    scale_fill_manual(values = custom_colors, labels = view_labels) +  # Apply custom colors
    theme_void() +  # Remove unnecessary gridlines
    theme(legend.position = "right")  # Ensure the legend is visible

  # Store the plot in the list with the group value as the name
  pie_charts[[paste0("Group_", group_value)]] <- p
}

# View stored pie charts
pie_charts[["Group_1"]]  # Example: Display pie chart for Group 1
```



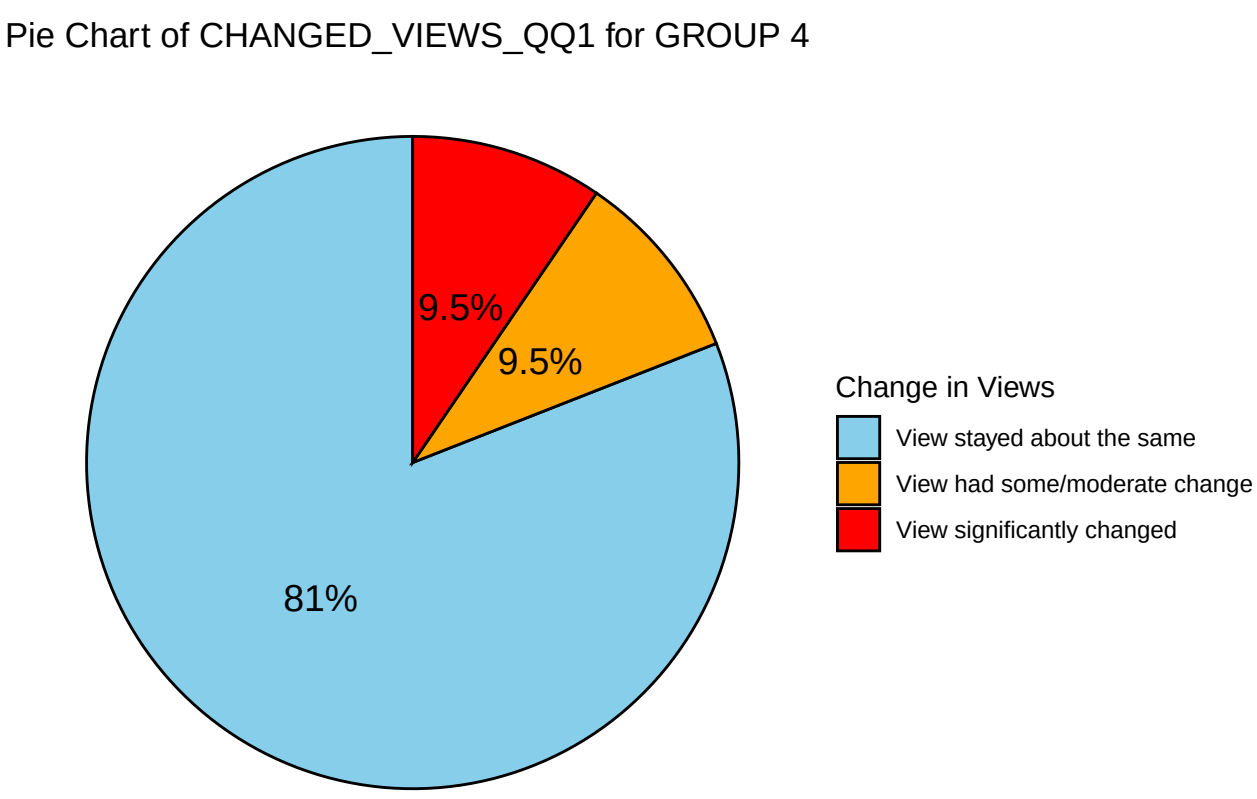

```r
pie_charts[["Group_2"]]  # Example: Display pie chart for Group 2
```




```
pie_charts[["Group_3"]]  # Example: Display pie chart for Group 3
```




```
pie_charts[["Group_4"]]  # Example: Display pie chart for Group 4
```



GCBS Pre-Score Q5 (Most Important)

Large groups of scientists manipulate, fabricate, or suppress evidence in order to deceive the public.

GCBS Pre-Score Q4 (Least Important)

The spread of most viruses and/or diseases is the result of the deliberate, concealed efforts of some organization.

Control Group (Audio): 21 (42.98%)
Control Group (Visual): 22 (44%)
Test Group (Audio): 28 (57.14%)
Test Group (Visual): 28 (56%)

Total: 99

They're trying to see if an advertisement influences their likelihood to believe in conspiracy theories and fake news? Hence asking questions about podcasts and social media.