Information-VIsualization-on-Donner-Dataset

Rajlakshmi Maurya

May 8, 2024

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

#PROBLEM 1 **********

```
#load all the libraries. these libraies will be used in successive programs.
library(dplyr)
```

```
##
## 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
```

```
library(tidyr)
library(magrittr)
```

```
##
## Attaching package: 'magrittr'

## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library(knitr)
library(ggplot2)
library(ggpubr)
library(ggsci)
library(vcd)
```

```
## Loading required package: grid
```

```r
library(vcdExtra)
```

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

```
## Loading required package: gnm
```

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

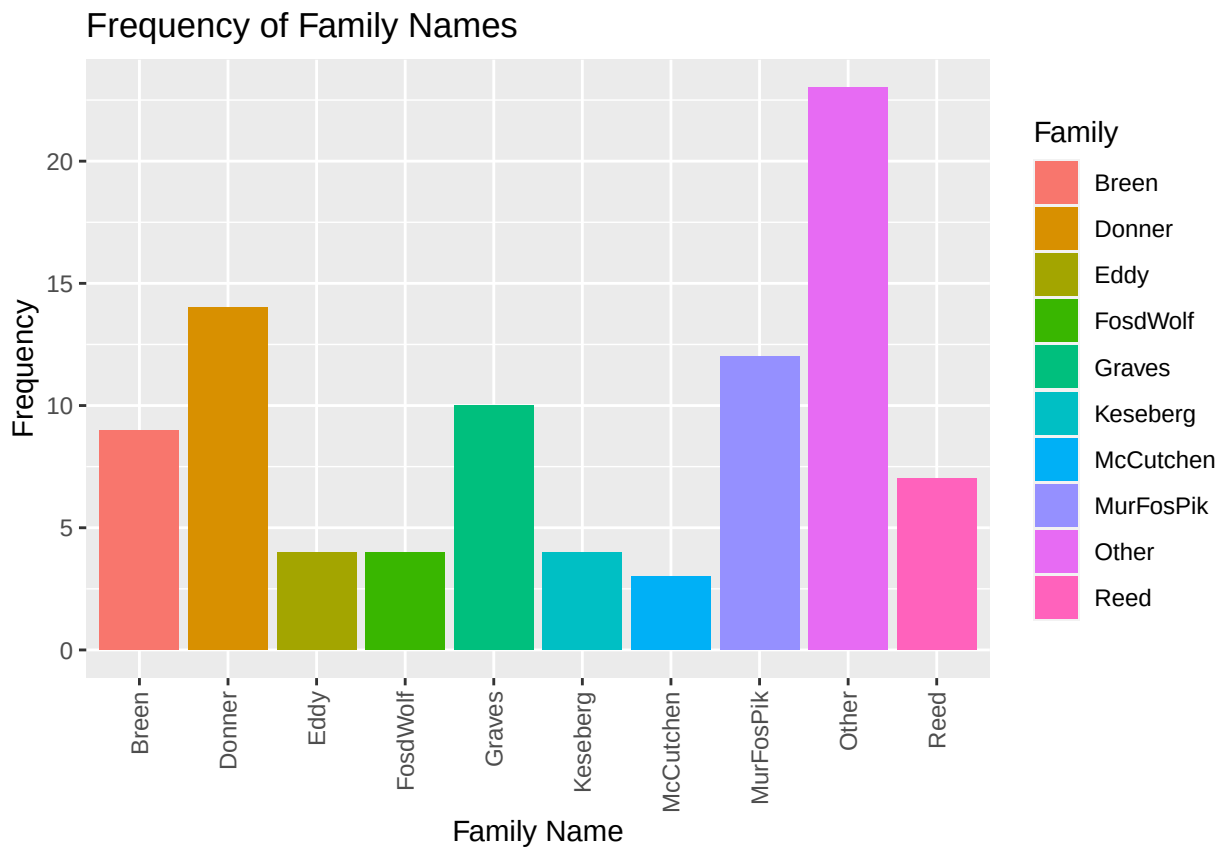
```
##
## Attaching package: 'vcdExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     summarise
```

```r
#a)
# Frequency of Family Names
family_name_freq <- as.data.frame(table(Donner$family))
colnames(family_name_freq) <- c("Family", "Frequency")
print(family_name_freq)
```

```
##       Family Frequency
## 1      Breen         9
## 2     Donner        14
## 3       Eddy         4
## 4   FosdWolf         4
## 5     Graves        10
## 6   Keseberg         4
## 7  McCutchen         3
## 8  MurFosPik        12
## 9      Other        23
## 10      Reed         7
```

```r
# Plot the graph using ggplot2 with color
ggplot(family_name_freq, aes(x = Family, y = Frequency, fill = Family)) +
  geom_bar(stat = "identity") +
  labs(title = "Frequency of Family Names", x = "Family Name", y = "Frequency") +
  scale_fill_discrete(name = "Family") +  # Set legend title
  theme(axis.text.x = element_text(angle=90, vjust=.5, hjust=1))
```

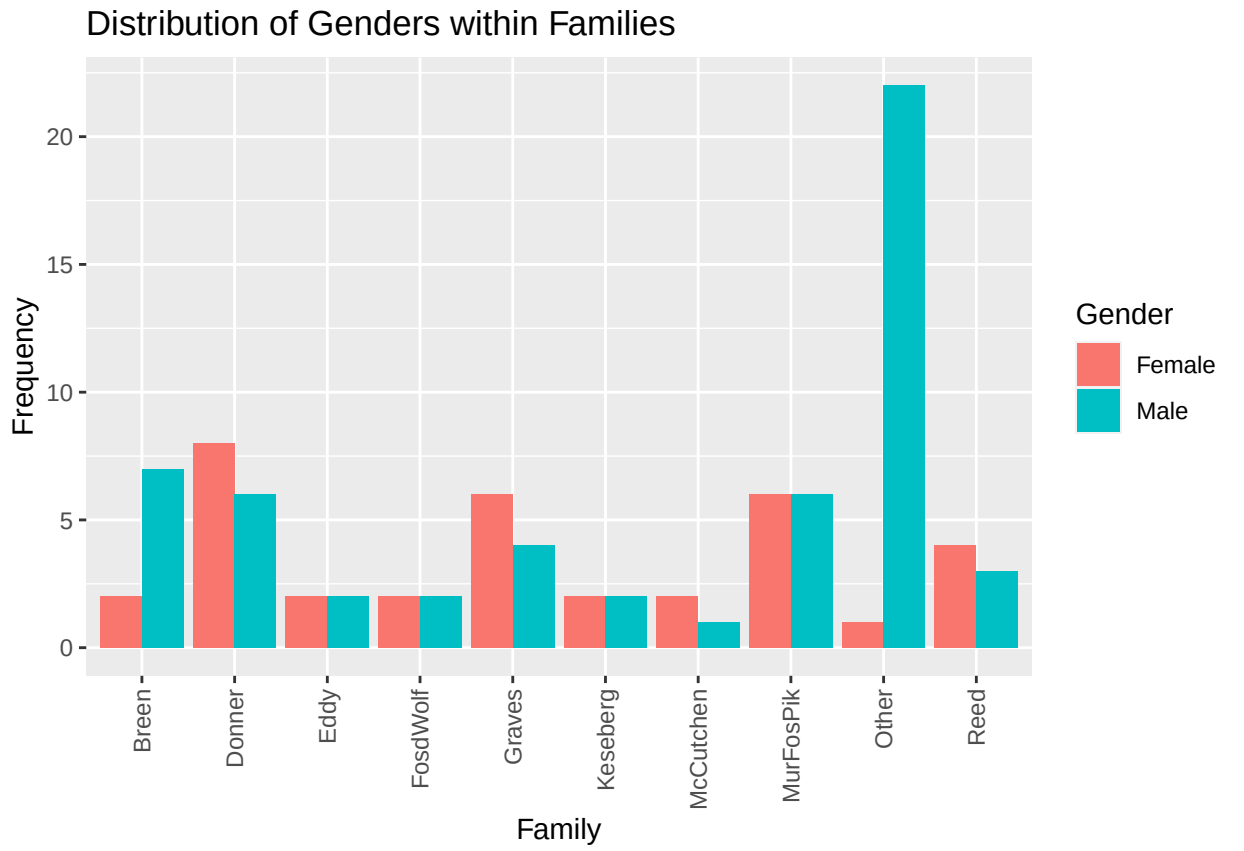



```r
#b)
# Distribution of Genders within Families
gender_in_families <- as.data.frame(table(Donner$family, Donner$sex))
colnames(gender_in_families) <- c("Family", "Gender", "Frequency")
print(gender_in_families)
```

```
##       Family Gender Frequency
## 1      Breen Female         2
## 2     Donner Female         8
## 3       Eddy Female         2
## 4   FosdWolf Female         2
## 5     Graves Female         6
## 6   Keseberg Female         2
## 7  McCutchen Female         2
## 8  MurFosPik Female         6
## 9      Other Female         1
## 10      Reed Female         4
## 11     Breen   Male         7
## 12    Donner   Male         6
## 13      Eddy   Male         2
## 14  FosdWolf   Male         2
## 15    Graves   Male         4
## 16  Keseberg   Male         2
## 17 McCutchen   Male         1
## 18 MurFosPik   Male         6
```

```
## 19     Other   Male        22
## 20      Reed   Male         3
```

```
ggplot(gender_in_families, aes(x = Family, y = Frequency, fill = Gender)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Distribution of Genders within Families", x = "Family", y = "Frequency", fill = "Gender
  theme(axis.text.x = element_text(angle=90, vjust=.5, hjust=1))
```

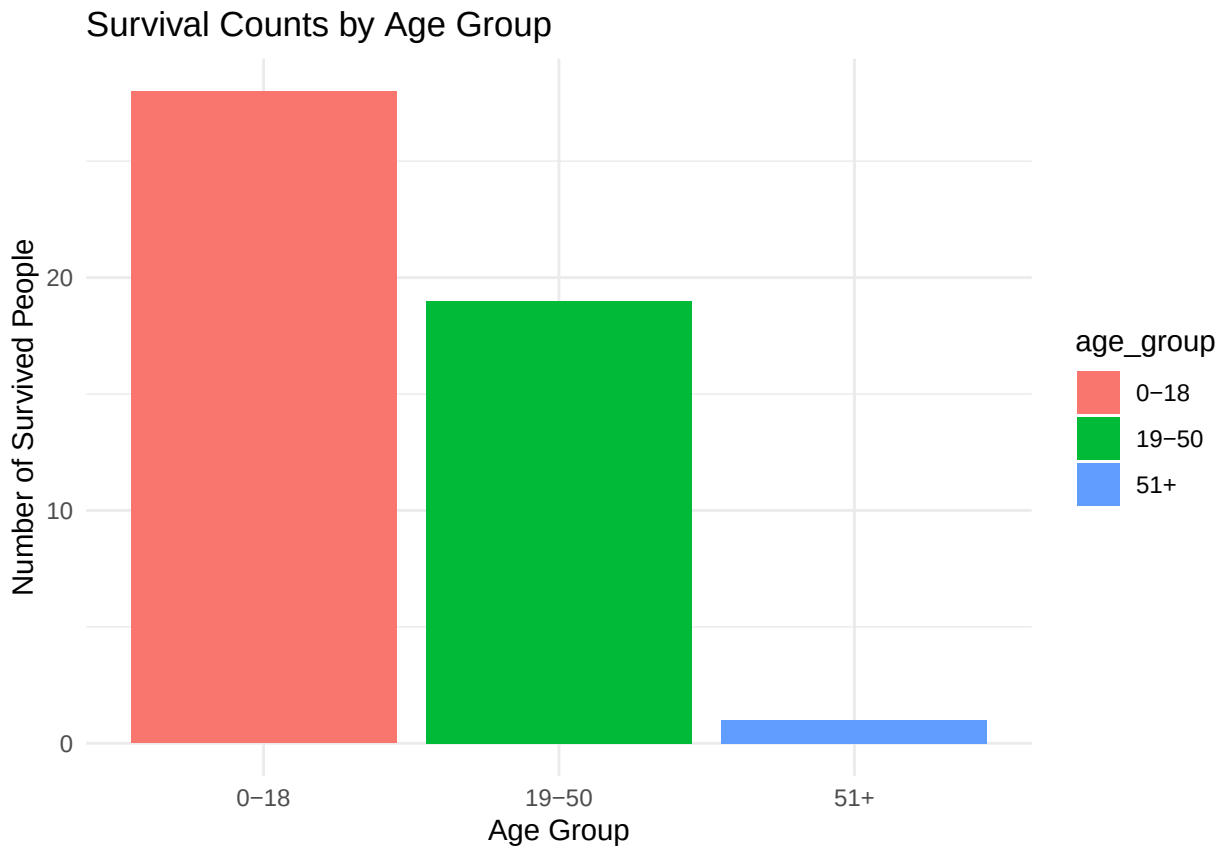



```
#c)
# Define age groups and count the number of people survive
donner_dataset<-Donner
donner_dataset$age_group <- cut(donner_dataset$age, breaks = c(0, 18, 50, max(Donner$age)), labels = c(
# Count the number of survived people in each age group
survived_counts <- aggregate(survived ~ age_group, data = donner_dataset, FUN = function(x) sum(x == 1))
print(survived_counts)
```

```
##   age_group survived
## 1      0-18       28
## 2     19-50       19
## 3       51+        1
```

```
# Plot the graph using ggplot2
ggplot(survived_counts, aes(x = age_group, y = survived, fill = age_group,palette="jco")) +
  geom_bar(stat = "identity") +
```

```
  labs(title = "Survival Counts by Age Group", x = "Age Group", y = "Number of Survived People",title.a
  theme_minimal()
```



The 3 Exploratory data analysis Questions I chose to address are:

A. Do certain family names appear more frequently than others? I Visualized the data using bar chart to figure out if certain family names appear more frequently than others. The analysis resulted in giving the outcome that there are more people with unknown family name (other column). Among all families, Donner has most frequency and McCutchen has least frequency.

B. What is the distribution of genders within families? By examining the gender composition within families, we can understand the dynamics of family units within the Donner dataset. This analysis involved counting the number of males and females within each family. As observed, there is more number of males vs females in families with no name (other). Coming to families with names, Donner family has highest count of Females whereas Family with no name (other) has lowest. Coming to male count, McCutchen family has least male count.

C. Is there any correlation between age and survival rate? Investigating whether age played a role in the survival of the Donner party members could be insightful. This analysis could involve comparing the ages of survivors and non-survivors.Sectioning the data in 3 age groups that is 0-18, 19-50 and 51+, to calculate the survival frequency of each age group resulted in observing that as age increases, the number of who people survive, decreases.

#PROBLEM2**********

```
data("Donner", package = "vcdExtra")
```

```r
# Calculate the total number of survivors in each family and the percentage of survivors
family_survival_summary <- Donner %>% #loading the data to dataframe.
  group_by(family) %>%  #grouping the data by family name
  summarize(total_members = n(),     #total members in each family
            total_survivors = sum(survived),  #counting members who survived
            percentage_survived = (total_survivors / total_members) * 100) %>%
  arrange(desc(percentage_survived)) # arrange data in descending order of percentage

# Print the data frame
print(family_survival_summary)
```

```
## # A tibble: 10 x 4
##    family    total_members total_survivors percentage_survived
##    <fct>             <int>           <int>               <dbl>
##  1 Breen                 9               9               100
##  2 Reed                  7               6                85.7
##  3 Graves               10               7                70
##  4 McCutchen             3               2                66.7
##  5 Donner               14               7                50
##  6 FosdWolf              4               2                50
##  7 Keseberg              4               2                50
##  8 MurFosPik            12               6                50
##  9 Other                23               6                26.1
## 10 Eddy                  4               1                25
```

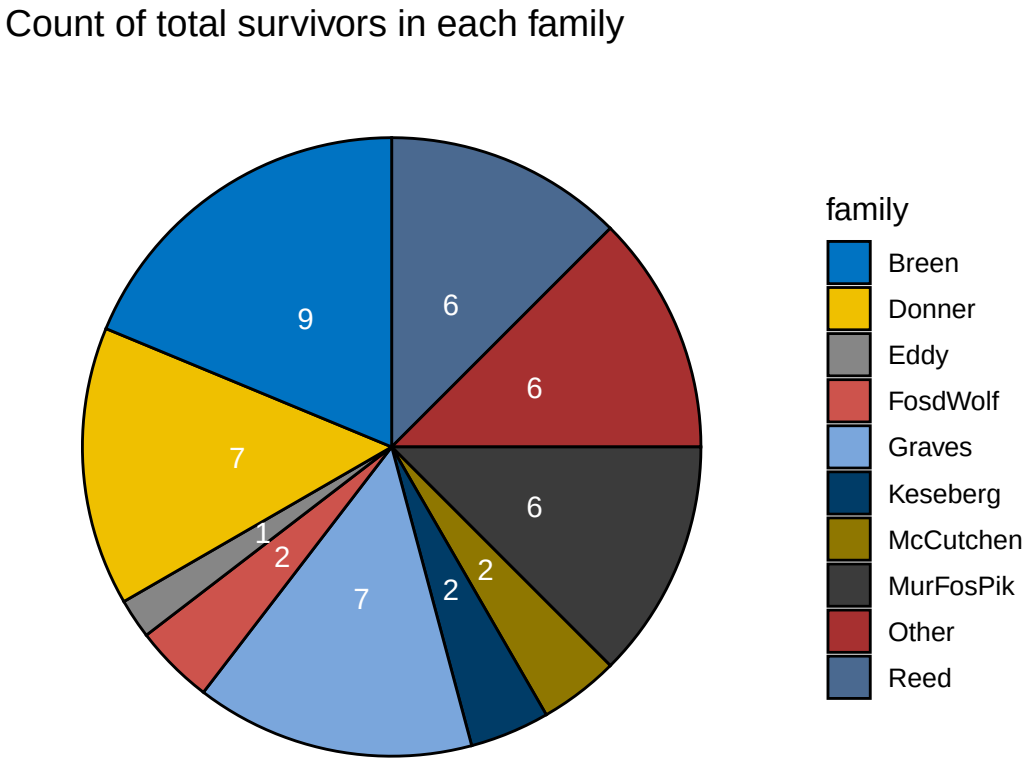
Explaination 2: As we can see, Among all families Breen had 100% survival rate with all its family membe

#PROBLEM3**********

```r
library(ggpubr)
# Group the data by family and calculate total survivors
survivors_summary <- Donner%>%
  group_by(family) %>%
  summarize(
            total_survivors = sum(survived))
survivors_summary
```

```
## # A tibble: 10 x 2
##    family    total_survivors
##    <fct>               <int>
##  1 Breen                   9
##  2 Donner                  7
##  3 Eddy                    1
##  4 FosdWolf                2
##  5 Graves                  7
##  6 Keseberg                2
##  7 McCutchen               2
##  8 MurFosPik               6
##  9 Other                   6
## 10 Reed                    6
```

```
# Create the pie chart
ggpie(survivors_summary, "total_survivors", label = "total_survivors",fill="family",lab.pos="in",lab.fo
```



Explanation3: As we can see group_by(family) groups the data by the 'family' column.
summarize() calculates summary statistics for each group, in this case, it calculates the total number
Coming to plotting the graph, ggpie() is a function to create a pie chart using ggplot2.
survivors_summary is the data frame with summarized information."
total_survivors" specifies the variable to be represented by the size of the pie slices.
fill = "family" specifies the variable to determine the colors of the pie slices.
label = "total_survivors" sets the label for each slice to be the total number of survivors.
lab.pos = "in" positions the labels inside the pie chart.
lab.font = list(color = "white") sets the label color to white.
palette = "jco" specifies the color palette to use.
legend = "right" positions the legend to the right of the pie chart.
As we can see, the family with highest count occupies biggest slice in pie chart, whereas lowest count

#PROBLEM4**********

```
# Calculate the count of died and survived
died_count <- sum(Donner$survived == 0)
survived_count <- sum(Donner$survived == 1)

# Create a dataframe
survival_summary <- data.frame(
  Status = factor(c("Died", "Survived"), levels = c("Died", "Survived")),
```

```
  Count = c(died_count, survived_count)
)

# Print the summary dataframe
print(survival_summary)
```

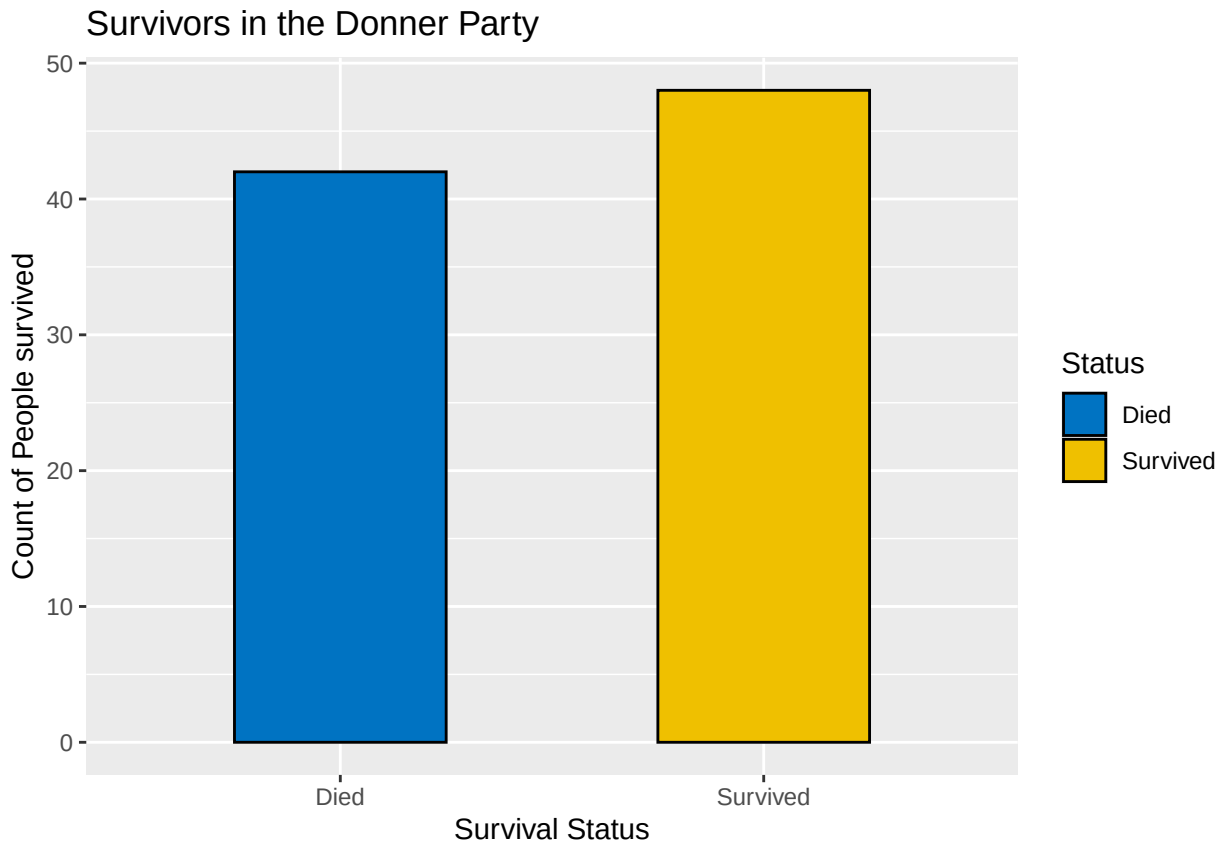
```
##     Status Count
## 1     Died    42
## 2 Survived    48
```

line 142 calculates the number of instances where the 'survived' column in the Donner dataset equals 0,
line 143calculates the number of instances where the 'survived' column equals 1, indicating those who su
data.frame() creates a new data frame named survival_summary.
Status is a factor variable with two levels, "Died" and "Survived".
Count contains the counts of individuals who died and survived, respectively.
factor(c("Died", "Survived"), levels = c("Died", "Survived")) ensures that the levels are ordered as "Di

As we can see, the above Dataframe has attributes Status with Died and survived values and their corresp

#PROBLEM5**********

```
# Load the required library
library(ggpubr)

ggbarplot(survival_summary, x = "Status", y = "Count",
          fill = "Status",
          palette = "jco",
          title = "Survivors in the Donner Party",
          title.pos = "center",
          xlab = "Survival Status",
          ylab = "Count of People survived",
          width = 0.5)+
  theme_gray()
```

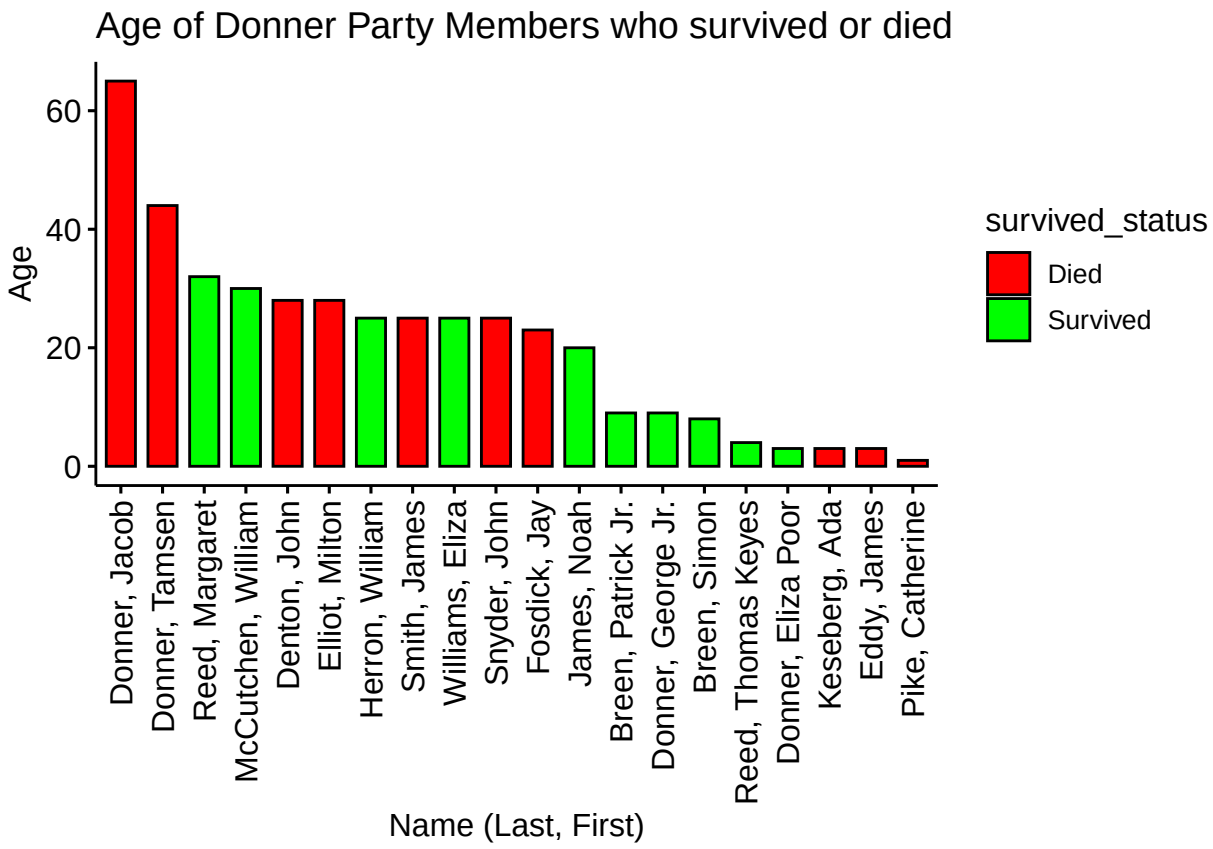



```
ggbarplot() creates a bar plot using the ggplot2 package.
survival_summary is the data frame containing the summary information.
x = "Status" specifies the variable for the x-axis, which is the survival status.
y = "Count" specifies the variable for the y-axis, which is the count of people.
fill = "Status" assigns colors based on the different categories of the "Status" variable.
palette = "jco" specifies the color palette to be used, in this case, the jco palette.
title, title.pos, xlab, ylab, and width are various parameters to customize the appearance and layout o
Here, theme_gray() sets the theme of the plot to use a gray background with white grid lines. It modifi
```

##PROBLEM6**********

```r
# Take a random sample of 20 records from the Donner dataset
sample_data <- Donner[sample(nrow(Donner), 20), ]
#order sample data by age
sample_data <- sample_data[order(sample_data$age, decreasing = TRUE), ]

# Add row numbers as IDs
sample_data$ID <- seq_len(nrow(sample_data))
sample_data <- cbind(rownames(sample_data), sample_data)
rownames(sample_data) <- NULL
colnames(sample_data) <- c("name","family","age","sex","survived_status","death","rowid")
# Convert "survived" to a factor with labels "Survived" and "Died"
sample_data$survived_status <- factor(sample_data$survived_status, levels = c(0, 1), labels = c("Died",

# Create the bar plot with labels, aligned names, and bars arranged by age
```

```r
ggbarplot(sample_data, x = "name", y = "age",
          fill = "survived_status",
          ylab = "Age", xlab = "Name (Last, First)",
          main = "Age of Donner Party Members who survived or died ",
          position = position_dodge(),
          legend = "right",
          palette = c("red", "green")) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))
```



For the above visulaization, we start by taking sample of data with 20 record and order it based on age
y = "age" specifies the variable for the y-axis as the age of the individuals.
fill = "survived_status" fills the bars based on whether the individual survived or died.
ylab and xlab specify the labels for the y-axis and x-axis, respectively.
main sets the main title of the plot.
position = position_dodge() ensures that bars are placed adjacent to each other for better visualizatio
legend = "right" positions the legend on the right side of the plot.
palette = c("red", "green") defines the color palette for the plot.
theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1)) adjusts the angle, vertical and h

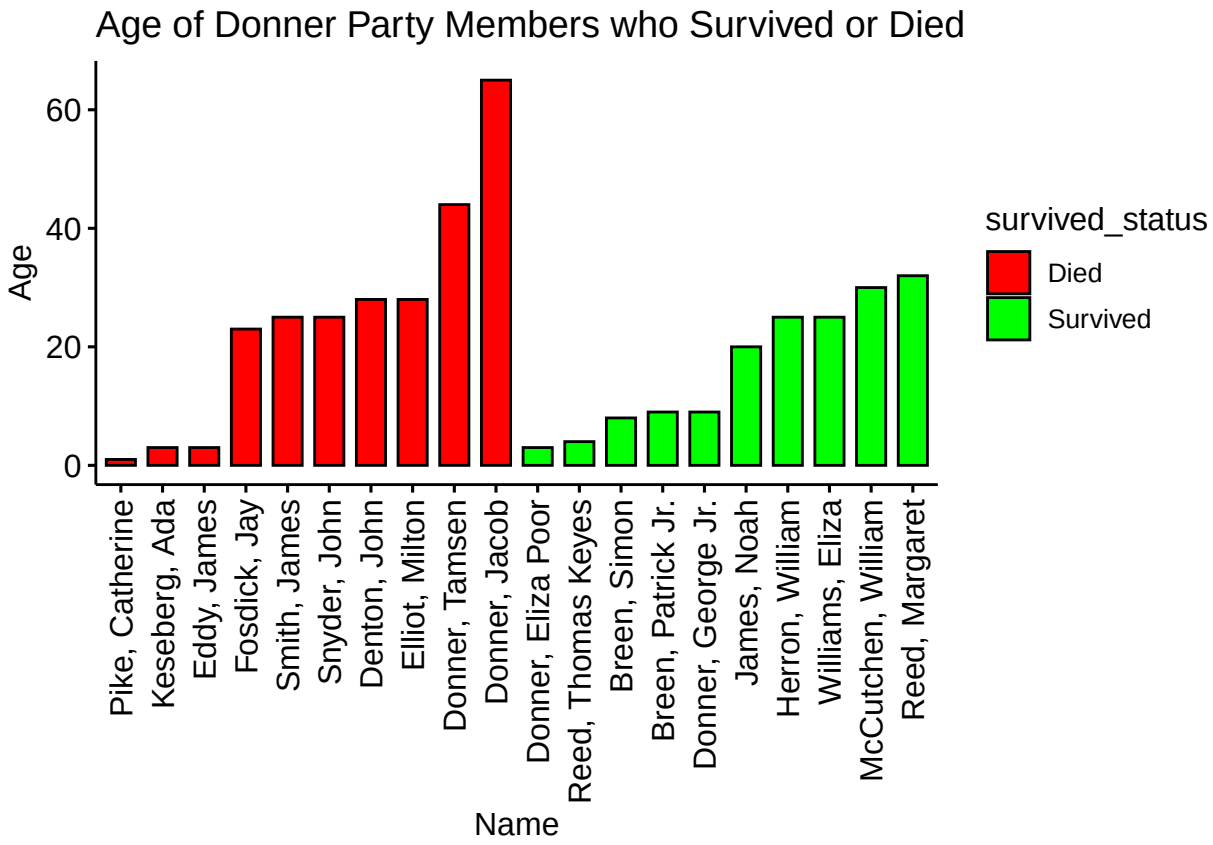
As we can observe,The bars are organised in decreasing order of the age, with each bar associated with

PROBLEM 7**********

```r
#Reorder the levels of "name" based on survival status
sample_data <- sample_data[order(sample_data$survived_status, sample_data$age), ]

sample_data
```

```
##                  name    family age    sex survived_status      death rowid
## 20    Pike, Catherine MurFosPik   1 Female            Died 1847-02-20    20
## 18      Keseberg, Ada  Keseberg   3 Female            Died 1847-02-25    18
## 19        Eddy, James      Eddy   3   Male            Died 1847-03-13    19
## 11       Fosdick, Jay  FosdWolf  23   Male            Died 1847-01-18    11
## 8        Smith, James     Other  25   Male            Died 1846-12-21     8
## 10       Snyder, John     Other  25   Male            Died 1846-10-05    10
## 5        Denton, John     Other  28   Male            Died 1847-02-26     5
## 6      Elliot, Milton     Other  28   Male            Died 1847-02-09     6
## 2      Donner, Tamsen    Donner  44 Female            Died 1847-03-28     2
## 1       Donner, Jacob    Donner  65   Male            Died 1846-12-21     1
## 17 Donner, Eliza Poor    Donner   3 Female        Survived       <NA>    17
## 16 Reed, Thomas Keyes      Reed   4   Male        Survived       <NA>    16
## 15       Breen, Simon     Breen   8   Male        Survived       <NA>    15
## 13 Breen, Patrick Jr.     Breen   9   Male        Survived       <NA>    13
## 14 Donner, George Jr.    Donner   9   Male        Survived       <NA>    14
## 12        James, Noah     Other  20   Male        Survived       <NA>    12
## 7     Herron, William     Other  25   Male        Survived       <NA>     7
## 9     Williams, Eliza     Other  25 Female        Survived       <NA>     9
## 4  McCutchen, William McCutchen  30   Male        Survived       <NA>     4
## 3      Reed, Margaret      Reed  32 Female        Survived       <NA>     3
```

```r
# Create the bar plot with labels, aligned names, and bars arranged by survival status and then age
ggbarplot(sample_data, x = "name", y = "age",
          fill = "survived_status",
          ylab = "Age", xlab = "Name",
          title  = "Age of Donner Party Members who Survived or Died",
          position = position_dodge(),
          legend = "right",
          palette = c( "red","green")) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))
```



Explanation 7: The sample_data dataframe is reordered based on the survival status and then the age of t

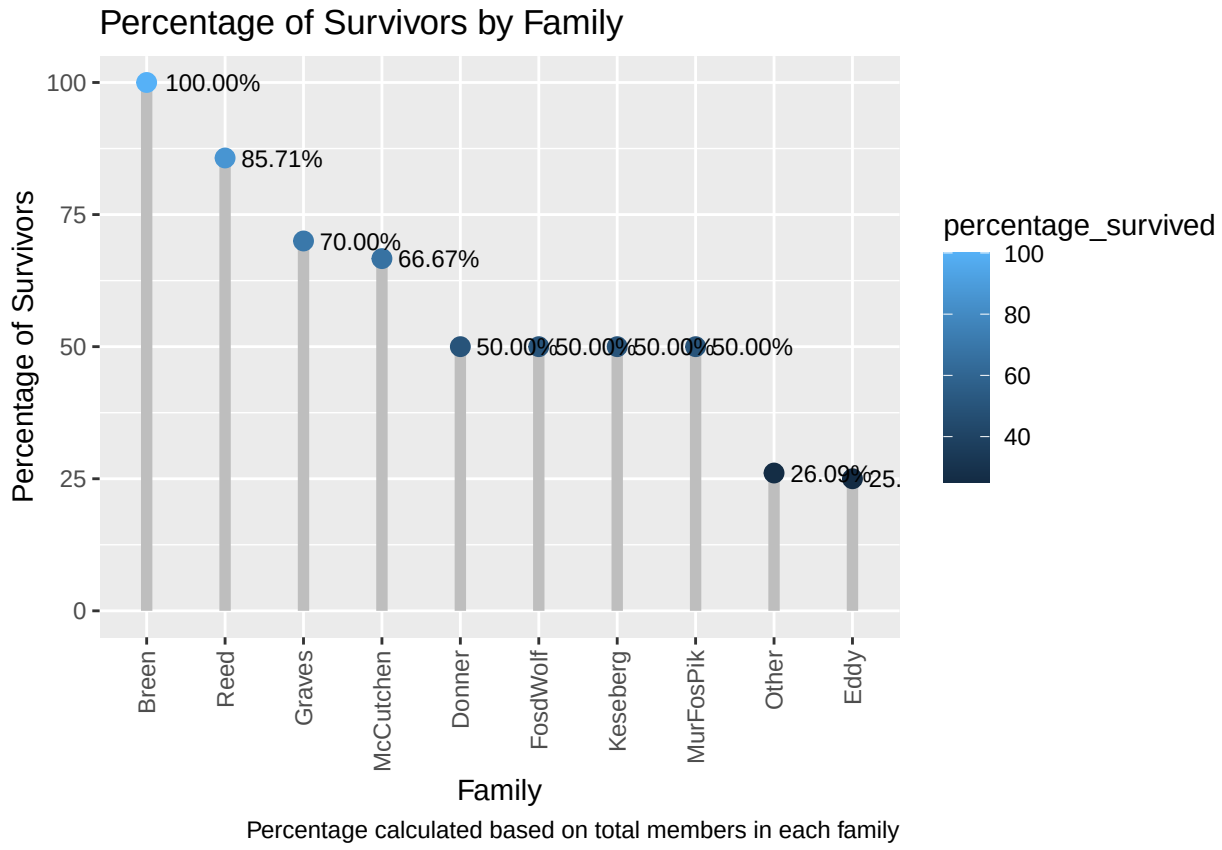
The ggbarplot has the following attributes
x = "name" specifies the variable for the x-axis as the names of the individuals.
y = "age" specifies the variable for the y-axis as the age of the individuals.
fill = "survived_status" fills the bars based on whether the individual survived or died.
ylab and xlab specify the labels for the y-axis and x-axis, respectively.
title sets the main title of the plot.
position = position_dodge() ensures that bars are placed adjacent to each other for better visualizatio
legend = "right" positions the legend on the right side of the plot.
palette = c("red", "green") defines the color palette for the plot.
theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1)) adjusts the angle, vertical, and h

As we can observe, In Died Category, Catherine Pike is Youngest and Augustus Spitzer is eldest. In survi

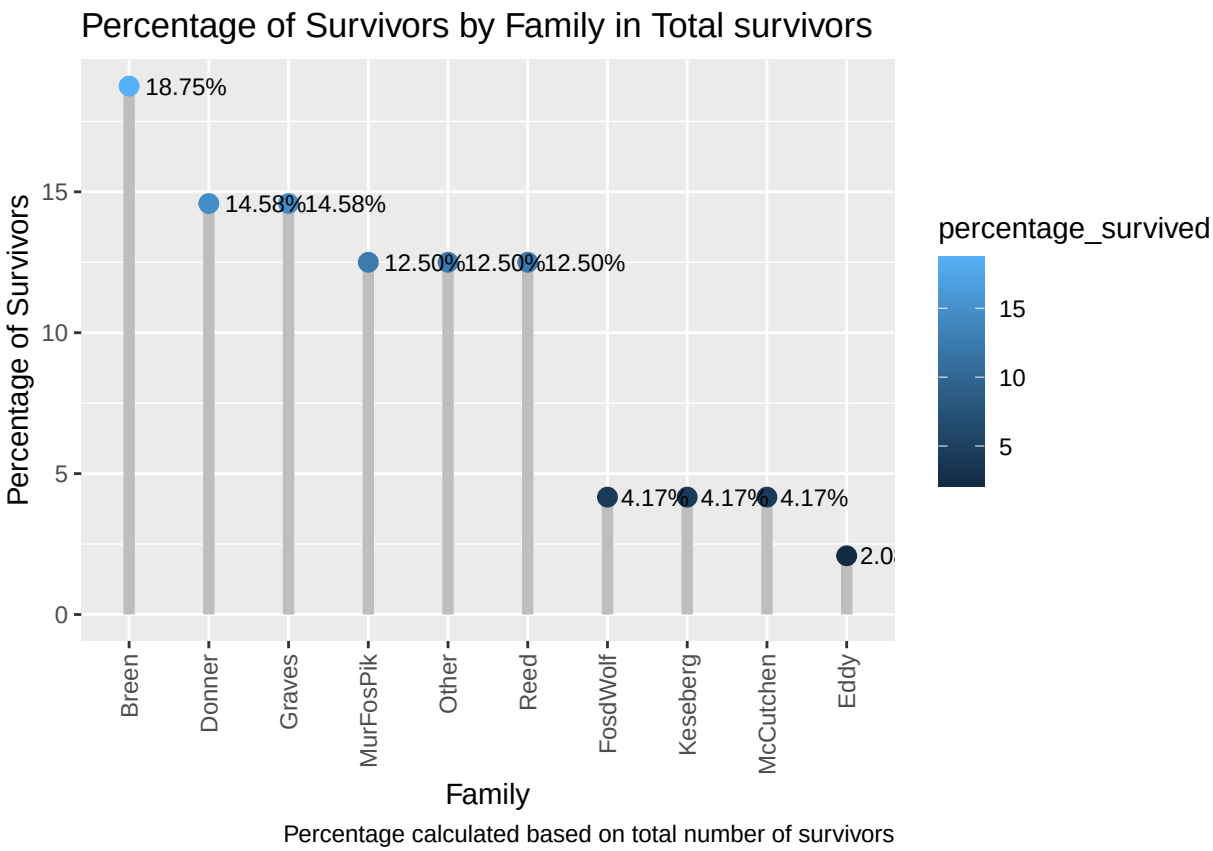
#PROBLEM8**********

```
# Calculate the total number of survivors
total_survivors <- sum(Donner$survived)

# Calculate the percentage of survivors by family based on total survivors
survival_percentage_by_family <- Donner %>%
  group_by(family) %>%
  summarize(total_members = n(),
            total_survivors_in_family = sum(survived),
```

```
            percentage_survived = (total_survivors_in_family / total_survivors) * 100) %>%
  arrange(desc(percentage_survived))

# Create the dot chart for survivor percent of each family
ggplot(family_survival_summary, aes(x = reorder(family, -percentage_survived), y = percentage_survived,
  geom_segment(aes(x = reorder(family, -percentage_survived), xend = family, y = 0, yend = percentage_su
  geom_point(aes(color = percentage_survived), size = 3) +
  geom_text(size = 3, hjust = -0.2) +
  labs(title = "Percentage of Survivors by Family",
       x = "Family",
       y = "Percentage of Survivors",
       caption = "Percentage calculated based on total members in each family") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))
```




```
# Create the dot chart
ggplot(survival_percentage_by_family, aes(x = reorder(family, -percentage_survived), y = percentage_surv
  geom_segment(aes(x = reorder(family, -percentage_survived), xend = family, y = 0, yend = percentage_su
  geom_point(size = 3,aes(color = percentage_survived)) +
  geom_text(size = 3, hjust = -0.2) +
  labs(title = "Percentage of Survivors by Family in Total survivors", title.align="center",
       x = "Family",
       y = "Percentage of Survivors",
       caption = "Percentage calculated based on total number of survivors") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))
```




```
The survival_percentage_by_family block code calculates the percentage of survivors in each family base
It groups the data by family, calculates the total members in each family, the total survivors in each

The ggplot code creates a dot chart to visualize the percentage of survivors by family.
geom_segment, geom_point, and geom_text are used to represent the data points and corresponding colors
The x-axis represents each family, reordered by the percentage of survivors.
The y-axis represents the percentage of survivors in each family.
The chart is labeled with appropriate titles, axis labels, and captions.

The above solution has 2 dot chart
1. first dot chart depicting the percent of survivors based on total members in each family
2. second dot chart depicting the percent of survivors when considering total survivors of all families
As we can observe, Breen family has the highest survival percentage, followed by Reed family. Eddy Fami
```

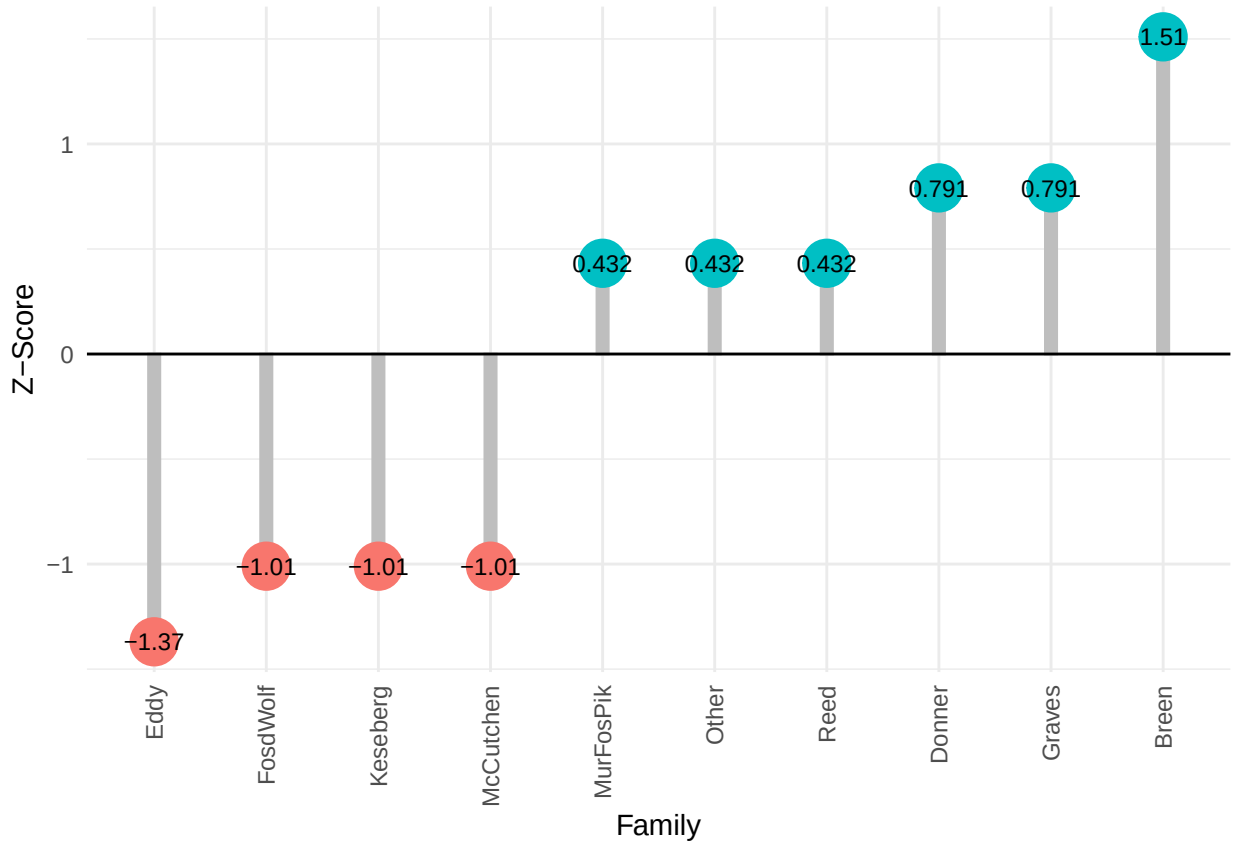
#PROBLEM9**********

```
# Filter the data for survivors and calculate the count of survivors per family
survivors_data <- Donner %>% filter(survived == 1) %>% count(family)
# Calculate the total count of individuals per family
total_data <- Donner %>% count(family)
# Calculate the number of survivors
num_survivors <- survivors_data$n
# Calculate the mean and standard deviation of the number of survivors
mean_survivors <- mean(num_survivors)
sd_survivors <- sd(num_survivors)
```

```r
# Calculate the z-score for each family based on the number of survivors and round off to 3 digit decim
z_scores <- signif((num_survivors - mean_survivors) / sd_survivors, digits = 3)

# Combine the data into a single dataframe
family_data <- data.frame(
  Family = survivors_data$family,
  NumberSurvived = survivors_data$n,
  ZScore = z_scores
)

# Order the dataframe by z-score
family_data <- family_data[order(family_data$ZScore), ]

# Print the formatted dataframe
print(family_data)
```

```
##       Family NumberSurvived ZScore
## 3       Eddy              1 -1.370
## 4   FosdWolf              2 -1.010
## 6   Keseberg              2 -1.010
## 7  McCutchen              2 -1.010
## 8  MurFosPik              6  0.432
## 9      Other              6  0.432
## 10      Reed              6  0.432
## 2     Donner              7  0.791
## 5     Graves              7  0.791
## 1      Breen              9  1.510
```

```r
# Create the plot
ggplot(family_data, aes(reorder(Family, ZScore), ZScore, color = ZScore > 0)) +
  # Add columns representing z-scores
  geom_col(width = 0.1, color = "grey", fill = "grey") +
  # Add points for z-scores
  geom_point(size = 8, show.legend = FALSE) +
  # Set x-axis label
  xlab("Family") +
  # Set y-axis label
  ylab("Z-Score") +
  # Add horizontal line at y = 0
  geom_hline(yintercept = 0) +
  # Add text labels for z-scores
  geom_text(aes(label = ZScore), color = "black", size = 3)+
  theme_minimal()+ #remove any background color
  # Rotate x-axis labels vertically
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))
```



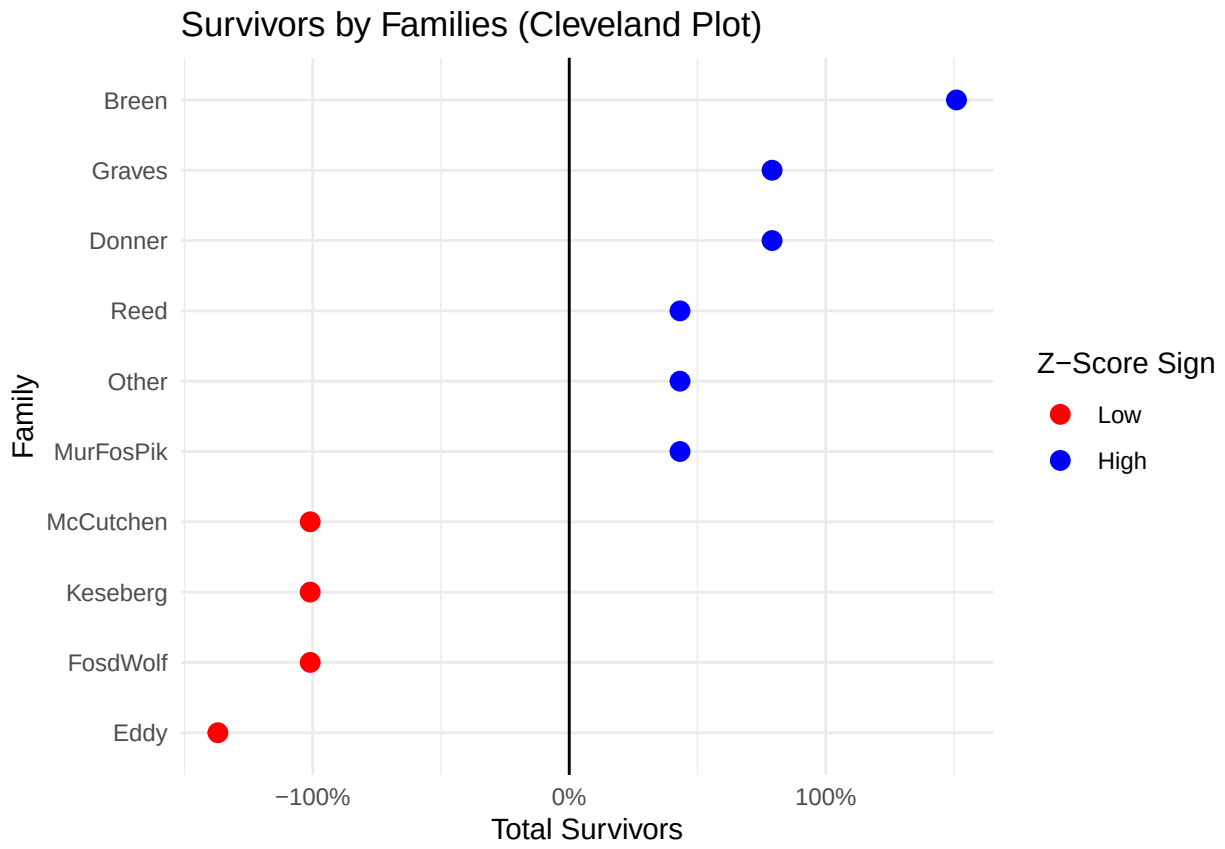
This above code filters the data to include only survivors, calculates the count of survivors per family
geom_col adds columns representing the Z-scores.
geom_point adds points for the Z-scores.
xlab and ylab set the labels for the x-axis and y-axis, respectively.
geom_hline adds a horizontal line at y = 0 to represent the mean.
geom_text adds text labels for the Z-scores.
theme_minimal() removes any background color, providing a clean appearance.
theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1)) rotates the x-axis labels vertical

As we can see, the dots are arranged in ascending order of their Z-score values (increasing from left t

#PROBLEM10**********

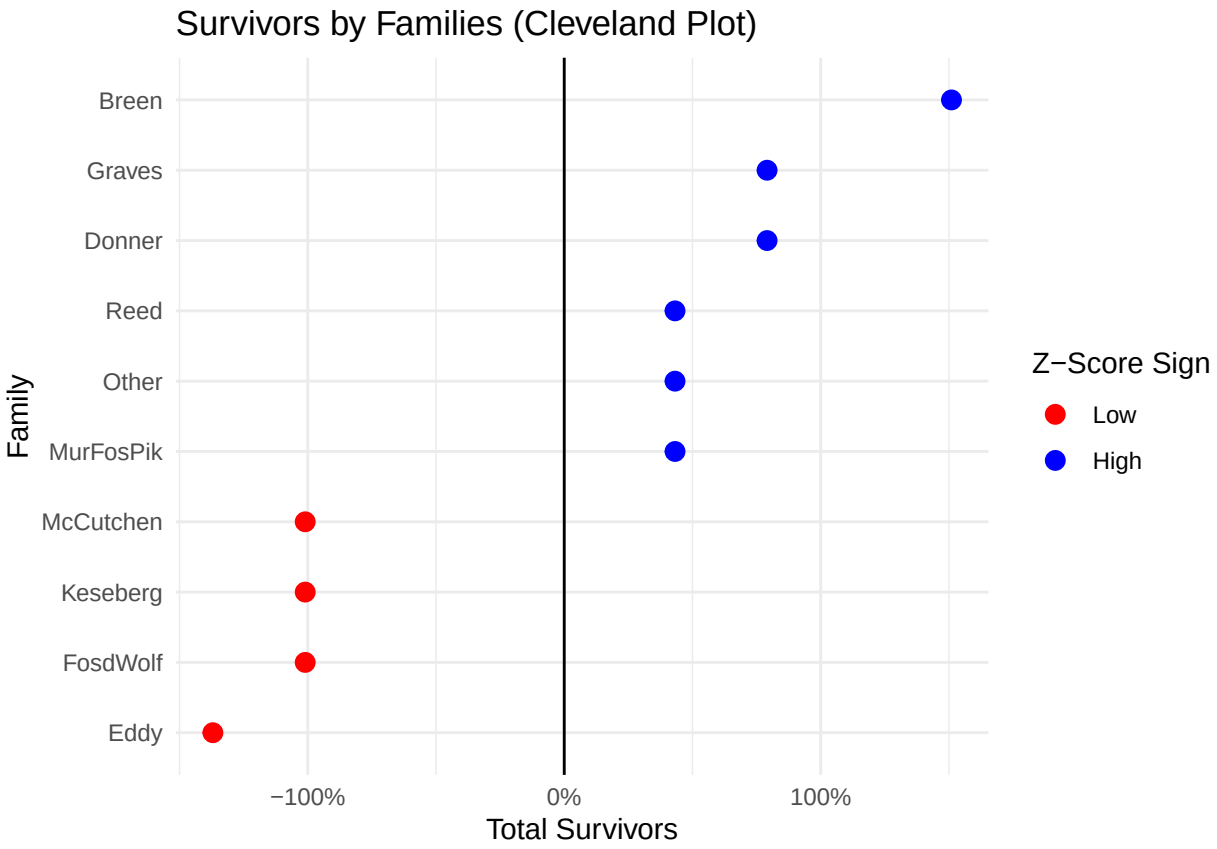
```
# reusing the family data from Problem 9 with attributes Family, ZScore, NumberSurvived to plot the Cle
#a<- ifelse(family_data$ZScore < 0, "red", "lightblue")
# Create the Cleveland plot
ggplot(family_data, aes(x = reorder(Family, ZScore), y = ZScore, color = factor(sign(ZScore)))) +
  geom_point(size = 3) +
  geom_hline(yintercept = 0) +
  scale_y_continuous(labels = scales::percent_format()) +  # Format y-axis as percentage
  scale_color_manual(values = c("red", "blue"), labels = c("Low", "High")) +
  labs(title = "Survivors by Families (Cleveland Plot)",
       x = "Family",
       y = "Total Survivors",
       color = "Z-Score Sign") +
```

```
  theme(plot.title = element_text(hjust = 1))+
  theme_minimal()+
  coord_flip()  # Rotate the plot by 90 degrees
```




```
# Create a vector of colors for x-axis labels based on Z-score
family_label_colors <- ifelse(family_data$ZScore < 0, "red", "blue")
# reusing the family data from Problem 9 with attributes Family, ZScore, NumberSurvived to plot the Cle

# Create the Cleveland plot
ggplot(family_data, aes(x = reorder(Family, ZScore), y = ZScore, color = factor(sign(ZScore)))) +
  geom_point(size = 3) +
  geom_hline(yintercept = 0) +
  scale_y_continuous(labels = scales::percent_format()) +  # Format y-axis as percentage
  scale_color_manual(values = c("red", "blue"), labels = c("Low", "High")) +
  labs(title = "Survivors by Families (Cleveland Plot)",
       x = "Family",
       y = "Total Survivors",
       color = "Z-Score Sign") +
  theme(plot.title = element_text(hjust = 1),
        axis.text.x = element_text(colour = family_label_colors, angle = 90, vjust = 0.5, hjust = 1))+
  theme_minimal()+
  coord_flip()
```

```
## Warning: Vectorized input to `element_text()` is not officially supported.
## i Results may be unexpected or may change in future versions of ggplot2.
```




```
 The code creates a Cleveland plot using ggplot2 to visualize the Z-scores and the number of survivors
aes  which specifies the aesthetic mappings.
geom_point adds points to the plot representing the Z-scores.
geom_hline adds a horizontal line at y = 0 to indicate the reference line.
scale_y_continuous formats the y-axis labels as percentages.
scale_color_manual manually sets the colors for positive and negative Z-scores, with labels "Low" and
labs sets the titles and labels for the plot and legend.
theme(plot.title = element_text(hjust = 1)) adjusts the horizontal alignment of the plot title.
theme_minimal() removes any background color, providing a clean appearance.
coord_flip() rotates the plot by 90 degrees, making the x-axis horizontal for better readability.

The resulting plot visualizes the Z-scores and the number of survivors by family, with different colors
```