Untitled

2023-10-01

(a) Exploratory Data Analysis

The dataset under consideration comprises 287 observations and 5 variables: Swim (Swimming Frequency), Loc (Location), Age, Sex, and NumInfec (Number of Infections). The dataset provides insights into ear infections and factors that may influence them.

In terms of swimming frequency (Swim), the most common category observed among individuals is "Occas" (occasional swimming), with 144 occurrences. Additionally, there are 143 instances of "Frequent" swimming. Moving on to swimming location (Loc), two primary locations are identified: "Beach" and "NonBeach." Within the dataset, there are 147 observations related to swimming at the beach and 140 observations associated with non-beach locations.

Age groups (Age) of individuals are categorized into "15-19," "20-24," and "25-29." Notably, the "15-19" age group exhibits the highest representation in the dataset, with 140 observations. The "20-24" age group follows with 79 observations, and the "25-29" age group has 68 observations.

Regarding gender distribution (Sex), the dataset encompasses two categories: "Male" and "Female." There is an observable imbalance in gender representation, with 188 observations (65.6%) classified as males and 99 observations (34.4%) classified as females.

Among females, 48 individuals swim occasionally, while 51 individuals swim frequently. For males, 95 individuals swim occasionally, while 93 individuals swim frequently. A chi-squared test yielded a p-value of 0.8372, indicating no significant association between swimming frequency and gender concerning ear infections.

Among females, 62 individuals swim at the beach, while 37 individuals prefer non-beach locations. For males, 85 individuals choose the beach, while 103 opt for non-beach locations. The chi-squared test resulted in a p-value of 0.007335, indicating a significant association between swimming location and gender concerning ear infections.

```
# Load necessary libraries
library(dplyr)      # For data manipulation
```

```
##
## 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(ggplot2)    # For data visualization
library(tidyr)      # For data tidying
library(gridExtra)  # For arranging multiple plots
```

```
##
## Attaching package: 'gridExtra'
```

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

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

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

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
# Load the dataset
EAR_INFECTION <- read.csv("Reference/DataSets/EAR_INFECTION.csv")
```

```
# Step 1: Data Exploration

# View the first few rows of the dataset
head(EAR_INFECTION)
```

```
##   ID  Swim      Loc   Age  Sex NumInfec
## 1  1 Occas NonBeach 15-19 Male        0
## 2  2 Occas NonBeach 15-19 Male        0
## 3  3 Occas NonBeach 15-19 Male        0
## 4  4 Occas NonBeach 15-19 Male        0
## 5  5 Occas NonBeach 15-19 Male        0
## 6  6 Occas NonBeach 15-19 Male        0
```

```
dim(EAR_INFECTION)
```

```
## [1] 287   6
```

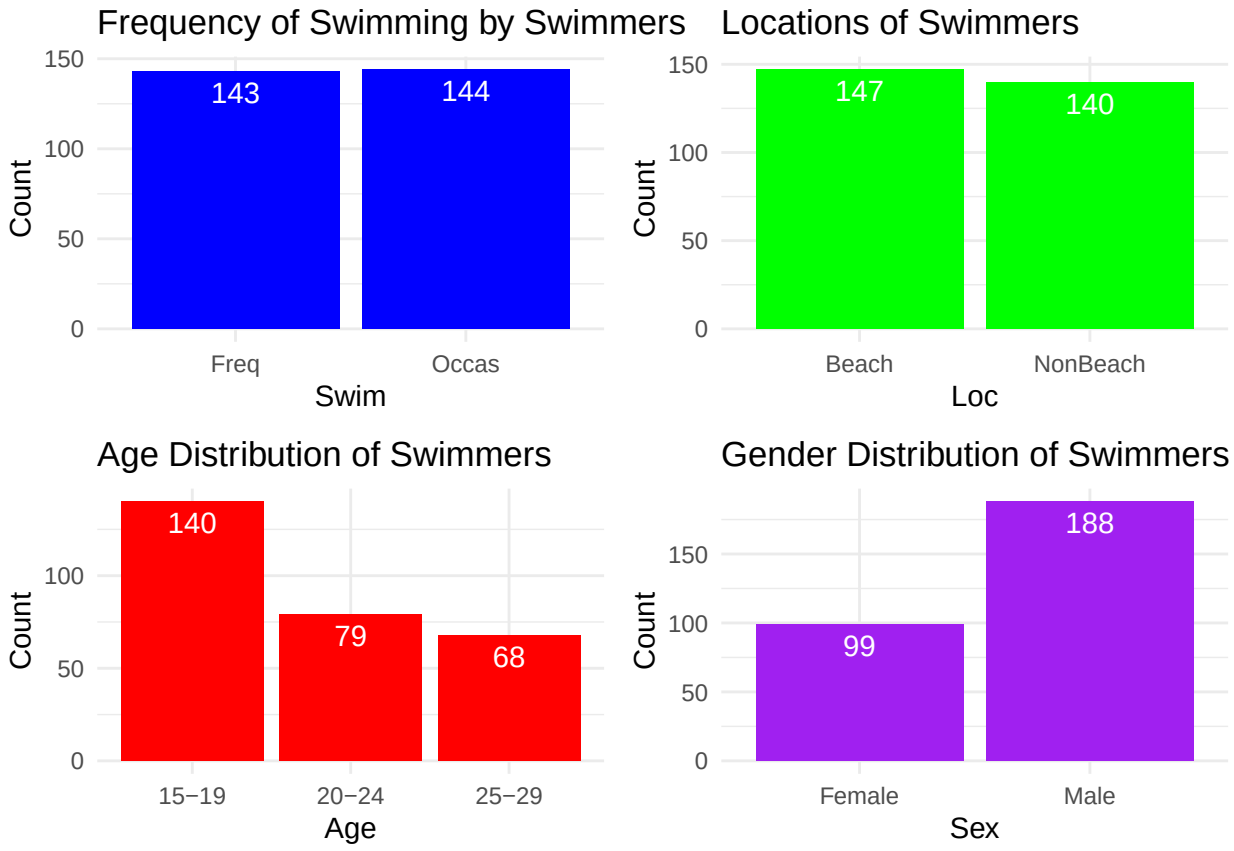
```
# Remove the "ID" column using dplyr::select
EAR_INFECTION <- dplyr::select(EAR_INFECTION, -ID)
```

```
# Function to create a bar plot with count labels and custom colors
create_bar_plot <- function(data, variable, title, bar_color, text_color) {
  plot <- ggplot(data, aes_string(x = variable)) +
    geom_bar(fill = bar_color) +  # Set bar fill color
    geom_text(stat = 'count', aes(label = after_stat(count)), color = text_color, vjust = 1.5) +  # Set
    labs(title = title) +
    xlab(variable) +
    ylab('Count') +
    theme_minimal() +  # Use a minimal theme for a cleaner appearance
    theme(legend.position = 'none')  # Remove the legend

  return(plot)
}
```

```
# Define custom colors for bars and text
bar_colors <- c("Swim" = "blue", "Loc" = "green", "Age" = "red", "Sex" = "purple")
text_colors <- c("Swim" = "white", "Loc" = "white", "Age" = "white", "Sex" = "white")

# Create bar plots for categorical variables with custom colors
swim_plot <- create_bar_plot(EAR_INFECTION, "Swim", 'Frequency of Swimming by Swimmers', bar_colors["Sw
```

```
## 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.
```

```
loc_plot <- create_bar_plot(EAR_INFECTION, "Loc", 'Locations of Swimmers', bar_colors["Loc"], text_colo
age_plot <- create_bar_plot(EAR_INFECTION, "Age", 'Age Distribution of Swimmers', bar_colors["Age"], te
sex_plot <- create_bar_plot(EAR_INFECTION, "Sex", 'Gender Distribution of Swimmers', bar_colors["Sex"],

# Arrange all the bar plots in a grid
grid.arrange(swim_plot, loc_plot, age_plot, sex_plot, ncol = 2)
```




```
# Save the combined plot as an image
ggsave("categorical_bar_plots.png", width = 10, height = 8)
```

```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
```

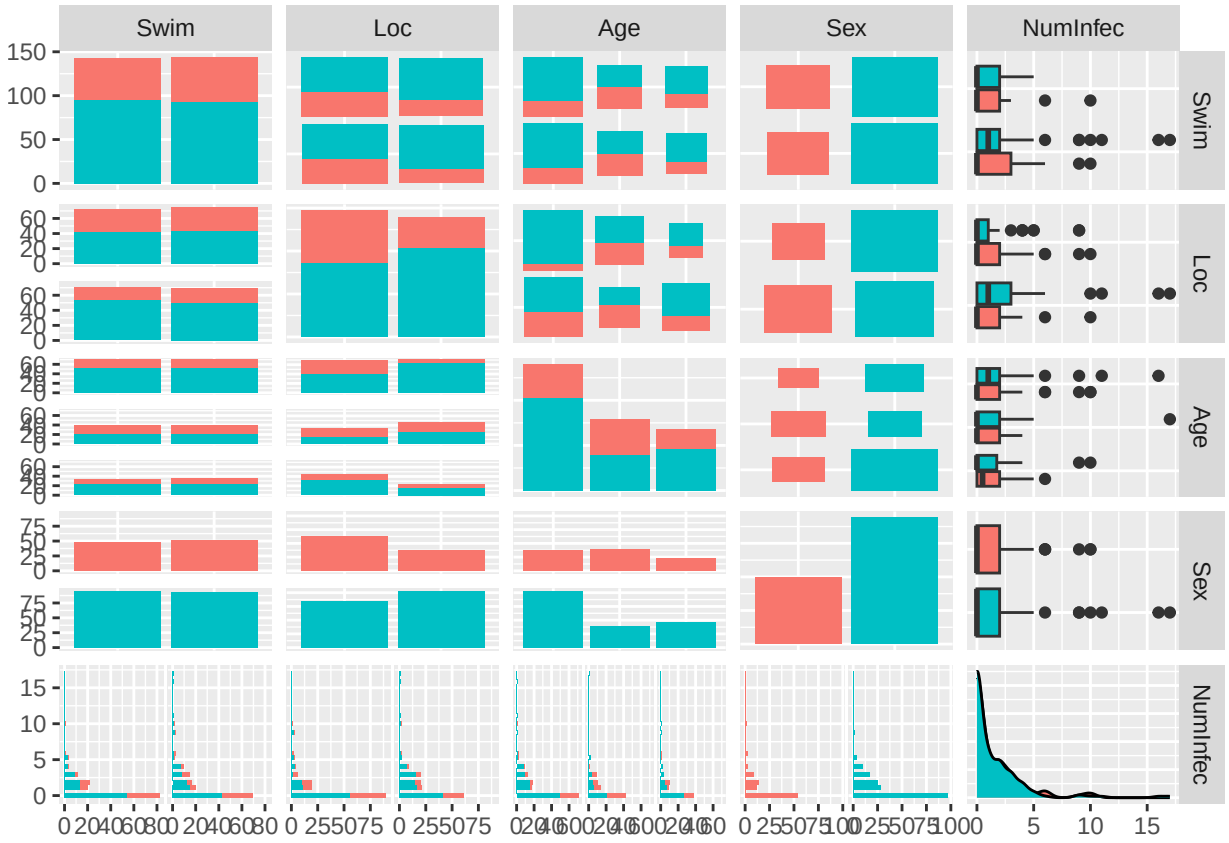
```
##   +.gg   ggplot2
```

```
# Create pair plot with correlations
ggpairs(EAR_INFECTION,
        aes(color = Sex),  # Optional: Color by Sex or other categorical variable
        lower = list(continuous = "cor"),  # Display correlations in the lower diagonal
        diag = list(continuous = "density")  # Display density plots on the diagonal
)
```

```
## Warning in check_and_set_ggpairs_defaults("diag", diag, continuous =
## "densityDiag", : Changing diag$continuous from 'density' to 'densityDiag'
```

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

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

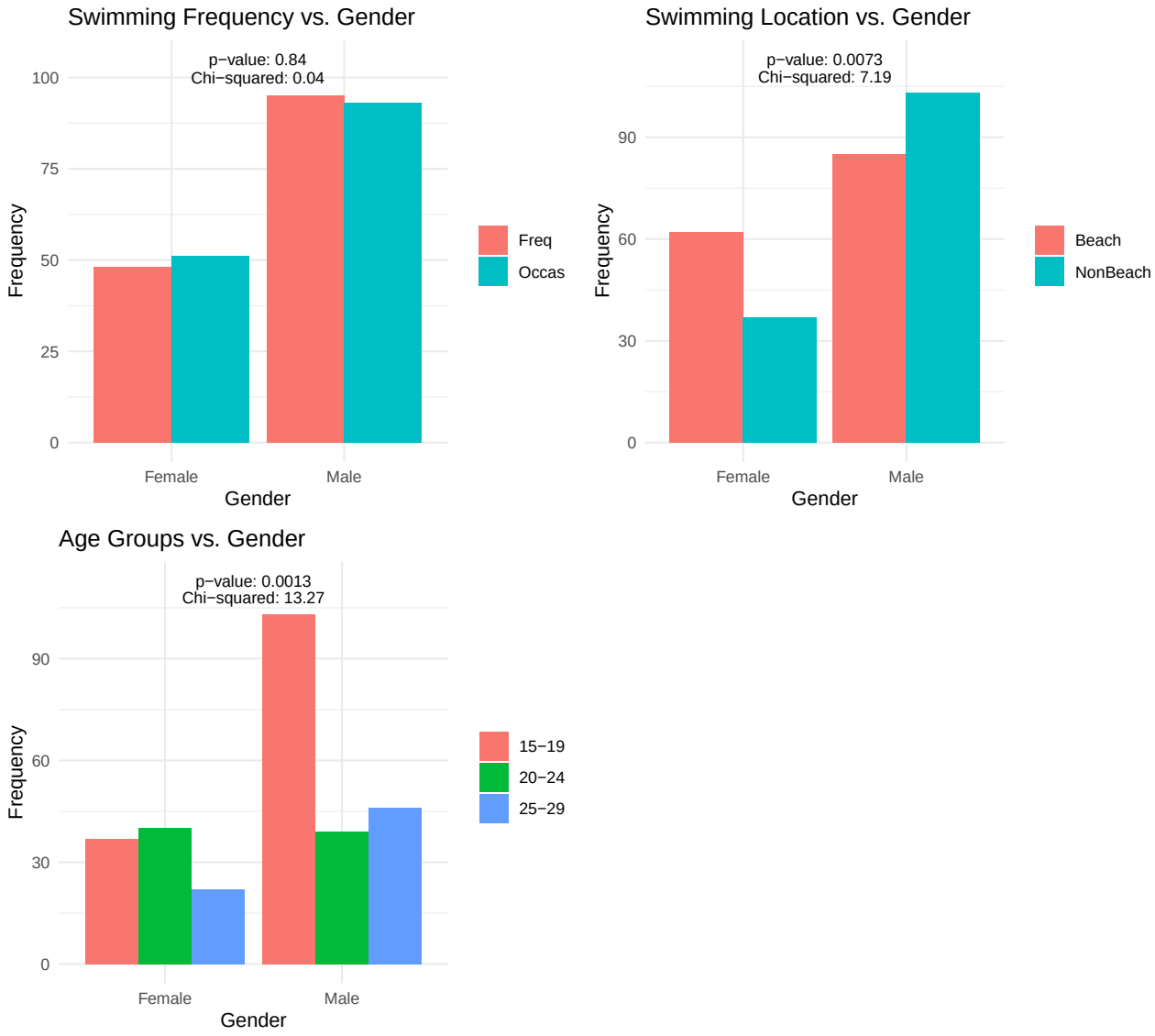


There are variations in the distribution of ear infections across age categories. Among females aged 15-19, 37 individuals experienced ear infections, with 22 individuals not affected. In the 20-24 age group, 40 females had ear infections, while 39 did not. In the 25-29 age group, 22 females had ear infections, and 46 did not. In contrast, among males aged 15-19, 103 individuals experienced ear infections, while 46 did not. For the 20-24 age group, 39 males had ear infections, and 40 did not. In the 25-29 age group, 46 males had ear infections, while 22 did not. A Pearson's chi-squared test for age groups yielded a p-value of 0.00131, indicating a significant association between age groups and gender concerning ear infections.

```
# Create a data frame for Swimming Frequency vs. Gender
swim_gender_data <- table(EAR_INFECTION$Sex, EAR_INFECTION$Swim)
chisq_swim_gender <- chisq.test(swim_gender_data)
```

```r
# Create a grouped bar plot for Swimming Frequency vs. Gender
swim_gender_plot <- ggplot(data = as.data.frame(swim_gender_data), aes(x = Var1, y = Freq, fill = Var2))
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Swimming Frequency vs. Gender",
       x = "Gender",
       y = "Frequency") +
  annotate("text", x = 1.5, y = max(swim_gender_data) + 5,
           label = paste("Chi-squared:", round(chisq_swim_gender$statistic, 2)),
           hjust = 0.5, size = 3) +
  annotate("text", x = 1.5, y = max(swim_gender_data) + 10,
           label = paste("p-value:", format.pval(chisq_swim_gender$p.value, digits = 2)),
           hjust = 0.5, size = 3) +
  theme_minimal() +
  theme(legend.title = element_blank())

# Create a data frame for Swimming Location vs. Gender
loc_gender_data <- table(EAR_INFECTION$Sex, EAR_INFECTION$Loc)
chisq_loc_gender <- chisq.test(loc_gender_data)

# Create a grouped bar plot for Swimming Location vs. Gender
loc_gender_plot <- ggplot(data = as.data.frame(loc_gender_data), aes(x = Var1, y = Freq, fill = Var2))
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Swimming Location vs. Gender",
       x = "Gender",
       y = "Frequency") +
  annotate("text", x = 1.5, y = max(loc_gender_data) + 5,
           label = paste("Chi-squared:", round(chisq_loc_gender$statistic, 2)),
           hjust = 0.5, size = 3) +
  annotate("text", x = 1.5, y = max(loc_gender_data) + 10,
           label = paste("p-value:", format.pval(chisq_loc_gender$p.value, digits = 2)),
           hjust = 0.5, size = 3) +
  theme_minimal() +
  theme(legend.title = element_blank())

# Create a data frame for Age Groups vs. Gender
age_gender_data <- table(EAR_INFECTION$Sex, EAR_INFECTION$Age)
chisq_age_gender <- chisq.test(age_gender_data)

# Create a grouped bar plot for Age Groups vs. Gender
age_gender_plot <- ggplot(data = as.data.frame(age_gender_data), aes(x = Var1, y = Freq, fill = Var2))
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Age Groups vs. Gender",
       x = "Gender",
       y = "Frequency") +
  annotate("text", x = 1.5, y = max(age_gender_data) + 5,
           label = paste("Chi-squared:", round(chisq_age_gender$statistic, 2)),
           hjust = 0.5, size = 3) +
  annotate("text", x = 1.5, y = max(age_gender_data) + 10,
           label = paste("p-value:", format.pval(chisq_age_gender$p.value, digits = 2)),
           hjust = 0.5, size = 3) +
  theme_minimal() +
  theme(legend.title = element_blank())
```

```
# Arrange the plots using gridExtra
grid.arrange(swim_gender_plot, loc_gender_plot, age_gender_plot, ncol = 2)
```




```
# Define a function to calculate the percentages and round to 2 decimal places
calculate_percentages <- function(data) {
  freq_percent <- prop.table(data) * 100
  freq_percent <- round(freq_percent, 2)
  return(freq_percent)
}

# Print combined contingency table and percentages for swimming frequency and gender
cat("Contingency Table for Swimming Frequency and Gender:\n")
```

```
## Contingency Table for Swimming Frequency and Gender:
```

```
combined_table_freq_percent <- calculate_percentages(swim_gender_data)
print(swim_gender_data)
```

```
##
```

```
##          Freq Occas
##   Female   48    51
##   Male     95    93
```

```
print(combined_table_freq_percent)
```

```
##
##           Freq Occas
##   Female 16.72 17.77
##   Male   33.10 32.40
```

```
cat("\n")
```

```
# Print combined contingency table and percentages for swimming location and gender
cat("Contingency Table for Swimming Location and Gender:\n")
```

```
## Contingency Table for Swimming Location and Gender:
```

```
combined_table_loc_percent <- calculate_percentages(loc_gender_data)
print(loc_gender_data)
```

```
##
##          Beach NonBeach
##   Female    62       37
##   Male      85      103
```

```
print(combined_table_loc_percent)
```

```
##
##          Beach NonBeach
##   Female 21.60    12.89
##   Male   29.62    35.89
```

```
cat("\n")
```

```
# Print combined contingency table and percentages for age groups and gender
cat("Contingency Table for Age Groups and Gender:\n")
```

```
## Contingency Table for Age Groups and Gender:
```

```
combined_table_age_percent <- calculate_percentages(age_gender_data)
print(age_gender_data)
```

```
##
##          15-19 20-24 25-29
##   Female    37    40    22
##   Male     103    39    46
```

```
print(combined_table_age_percent)
```

```
##
##          15-19 20-24 25-29
##   Female 12.89 13.94  7.67
##   Male   35.89 13.59 16.03
```

(b) Model for Number of Infections

The response variable of interest is the Number of Infections (NumInfec), which represents counts of ear infections. Given that we are working with count data, it is appropriate to model these data using a Generalized Linear Model (GLM) with a Poisson error distribution. The first model includes all available predictors.

The residual deviance, indicating the model's performance with the predictors included, was 755.43 on 281 degrees of freedom. The AIC (Akaike Information Criterion) for this model was 1139.8.

```
# Fit the Poisson GLM model with all predictors
modP <- glm(NumInfec ~ ., family = poisson, data = EAR_INFECTION)

# Print summary of the Poisson GLM model
summary(modP)
```

```
##
## Call:
## glm(formula = NumInfec ~ ., family = poisson, data = EAR_INFECTION)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.12261    0.13706  -0.895  0.37100
## SwimOccas    0.61149    0.10500   5.823 5.77e-09 ***
## LocNonBeach  0.53454    0.10668   5.011 5.43e-07 ***
## Age20-24    -0.37442    0.12836  -2.917  0.00354 **
## Age25-29    -0.18973    0.13009  -1.458  0.14473
## SexMale     -0.08985    0.11231  -0.800  0.42371
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 824.51  on 286  degrees of freedom
## Residual deviance: 755.43  on 281  degrees of freedom
## AIC: 1139.8
##
## Number of Fisher Scoring iterations: 6
```

To address the issue of overdispersion, a negative binomial model was refitted to the data using the glm.nb command in R. The resulting negative binomial model provided a better fit to the data, with a dispersion parameter (Theta) estimated at 0.5760.

The null deviance for the negative binomial model was 289.90 on 286 degrees of freedom, and the residual deviance was 269.13 on 281 degrees of freedom. The AIC for this model improved to 904.69, indicating a better fit compared to the initial Poisson model.

Table 1: Negative binomial models for Number of Infections (NumInfec)

Model Description	Model Formula
Model 1: (mod_nb1)	`glm.nb(NumInfec ~ ., data = EAR_INFECTION)`
Model 2: (mod_nb2)	`glm.nb(NumInfec ~ Swim + Loc + Age, data = EAR_INFECTION)`
Model 3: (mod_nb3)	`glm.nb(NumInfec ~ Swim * Sex + Loc + Age, data = EAR_INFECTION)`
Model 4: (mod_nb4)	`glm.nb(NumInfec ~ Swim + Loc * Sex + Age, data = EAR_INFECTION)`

```
# Fit negative binomial models
mod_nb1 <- glm.nb(NumInfec ~ ., data = EAR_INFECTION)
mod_nb2 <- glm.nb(NumInfec ~ Swim + Loc + Age, data = EAR_INFECTION)
mod_nb3 <- glm.nb(NumInfec ~ Swim * Sex  + Loc+ Age, data = EAR_INFECTION)
mod_nb4 <- glm.nb(NumInfec ~ Swim + Loc * Sex + Age, data = EAR_INFECTION)
```

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

```
##
## Attaching package: 'AICcmodavg'
```

```
## The following object is masked from 'package:lme4':
##
##     checkConv
```

```r
# Create a list of your models
model_list <- list(
  mod.nb1 = mod_nb2,
  mod.nb2 = mod_nb4,
  mod.nb3 = mod_nb1,
  mod.nb4 = mod_nb3
)

# Initialize empty vectors to store model information
model_names <- character(length(model_list))
num_parameters <- numeric(length(model_list))
aic_values <- numeric(length(model_list))
delta_aicc <- numeric(length(model_list))

# Calculate AICc values and other information for each model
for (i in 1:length(model_list)) {
  model_names[i] <- names(model_list)[i]
  num_parameters[i] <- df.residual(model_list[[i]])
  aic_values[i] <- AICc(model_list[[i]], k = 2)
}

# Calculate Delta AICc relative to the best model
best_model_index <- which.min(aic_values)
delta_aicc <- aic_values - aic_values[best_model_index]

# Create the model selection table
model_selection_table <- data.frame(
  Model = model_names,
  K = num_parameters,
  AICc = aic_values,
  Delta_AICc = delta_aicc
)

# Print the model selection table
knitr::kable(model_selection_table,
             caption = "Table 2: Model selection results")
```

Table 2: Table 2: Model selection results

Model	K	AICc	Delta_AICc
mod.nb1	282	903.4356	0.000000
mod.nb2	280	904.6208	1.185223
mod.nb3	281	905.0914	1.655852
mod.nb4	280	907.0962	3.660604

We examined the residual deviance for the preferred model, which is 269.21 on 282 degrees of freedom, indicating a non-significant fit ($p = 0.48232$). This suggests that the model fits the data adequately.

The final model, which describes the relationship between NumInfec and its predictors, is as follows:

$$\text{NumInfec} \sim \text{Negative Binomial}(\mu_i)$$

$$\log(\mu_i) = \beta_0 + \beta_1 \text{SwimOccas} + \beta_2 \text{LocNonBeach} + \beta_3 \text{Age20-24} + \beta_4 \text{Age25-29}$$

Where: - μ_i is the mean number of NumInfec.

The coefficients for each predictor are as follows:

- $\beta_0 = -0.1393$
- β_1 for SwimOccas is 0.6041 ($p < 0.001$)
- β_2 for LocNonBeach is 0.5050 ($p = 0.00858$)
- β_3 for Age20-24 is -0.4021 ($p = 0.07753$)
- β_4 for Age25-29 is -0.2597 ($p = 0.27986$)

Interpreting the coefficients: - The coefficient for SwimOccas (0.6041) is significant ($p < 0.001$), suggesting that as the frequency of swimming occasions increases, the expected number of NumInfec also increases. - The coefficient for LocNonBeach (0.5050) is significant ($p = 0.00858$), indicating that the location being Non-Beach is associated with a higher expected number of NumInfec. - The coefficient for Age20-24 (-0.4021) is marginally significant ($p = 0.07753$), suggesting that the age group $20 - 24$ may have a slightly lower expected number of NumInfec compared to other age groups. - The coefficient for Age25-29 (-0.2597) is not significant ($p = 0.27986$), indicating that age group $25 - 29$ is not significantly different from the reference group.

This model provides valuable insights into the factors influencing NumInfec, considering the significant predictors and their respective coefficients.

```
# preferred model
summary(mod_nb2)
```

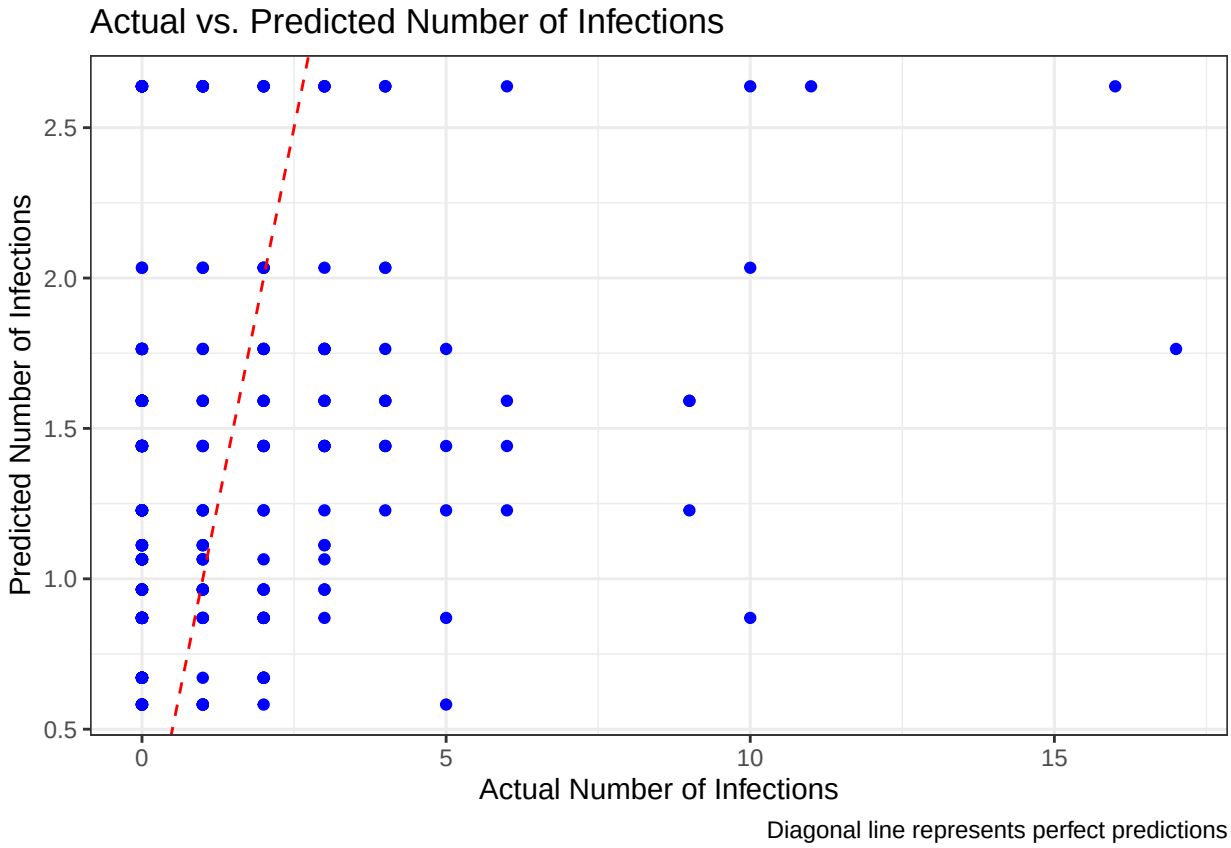
```
##
## Call:
## glm.nb(formula = NumInfec ~ Swim + Loc + Age, data = EAR_INFECTION,
##     init.theta = 0.5744421161, link = log)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.1393     0.1982  -0.703  0.48232
## SwimOccas     0.6041     0.1897   3.185  0.00145 **
## LocNonBeach   0.5050     0.1921   2.628  0.00858 **
## Age20-24     -0.4021     0.2278  -1.765  0.07753 .
## Age25-29     -0.2597     0.2403  -1.081  0.27986
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.5744) family taken to be 1)
##
##     Null deviance: 289.50  on 286  degrees of freedom
## Residual deviance: 269.21  on 282  degrees of freedom
## AIC: 903.14
##
## Number of Fisher Scoring iterations: 1
```

```
##
##
##                Theta:  0.5744
##            Std. Err.:  0.0900
##
##  2 x log-likelihood:  -891.1360
```

(c) Predicted Values for Number of Infections Versus Actual Values

The plot of Predicted versus Actual values for the number of infections (Figure 4) illustrates the model's performance in predicting the Number of Infections using Swimming Frequency, Location, Age, and Gender as predictors. The model, which includes appears to have utility for predicting the Number of Infections in this dataset. However, the widening variance in prediction errors as the actual values increase can be observed. This indicates that while the model may perform reasonably well for some cases, it may struggle to accurately predict the Number of Infections in instances with higher actual values.

```r
# Predict values using the best-fitting negative binomial model (mod.nb2)
predicted_values <- predict(mod_nb2, type = "response")

# Create a dataframe with actual and predicted values
predictions_df <- data.frame(Actual = EAR_INFECTION$NumInfec, Predicted = predicted_values)

# Create a scatterplot of actual vs. predicted values
plot_actual_vs_predicted <- ggplot(predictions_df, aes(x = Actual, y = Predicted)) +
  geom_point(size = 1.5, color = "blue") +  # Scatterplot points
  geom_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") +  # Add a diagonal referen
  labs(
    title = "Actual vs. Predicted Number of Infections",
    x = "Actual Number of Infections",
    y = "Predicted Number of Infections",
    caption = "Diagonal line represents perfect predictions"
  ) +
  theme_bw()  # Use a minimal theme for a cleaner appearance

# Display the scatterplot
print(plot_actual_vs_predicted)
```


Actual vs. Predicted Number of Infections
Predicted Number of Infections
2.5
2.0
1.5
1.0
0.5
0
5
10
15
Actual Number of Infections
Diagonal line represents perfect predictions