Problem 1: Horseshoe Crab Data Analysis

We will analyze the Horseshoe crab data to estimate the expected number of satellites using both a Poisson regression model and a linear regression model. We will compare the original satellite values with the estimated values from both models, plot the comparisons, and calculate the sum of squared differences.

Part A: Poisson Regression Model

Step 1: Load the Data

```
library(CatDataAnalysis)

# Load the Horseshoe crab data
data("table_4.3")
x <- table_4.3

# View the structure of the data
str(x)
```

```
## 'data.frame':    173 obs. of  6 variables:
##  $ color : int  3 4 2 4 4 3 2 4 3 4 ...
##  $ spine : int  3 3 1 3 3 3 1 2 1 3 ...
##  $ width : num  28.3 22.5 26 24.8 26 23.8 26.5 24.7 23.7 25.6 ...
##  $ satell: int  8 0 9 0 4 0 0 0 0 0 ...
##  $ weight: int  3050 1550 2300 2100 2600 2100 2350 1900 1950 2150 ...
##  $ y     : int  1 0 1 0 1 0 0 0 0 0 ...
```

Step 2: Fit the Poisson Regression Model
We will use a Poisson regression model with a log link function to estimate the expected number of satellites.

```
# Fit the Poisson regression model
log_model <- glm(satell ~ color + spine + width + weight, data = x, family = poisson(link = "log"))

# Display the summary of the model
summary(log_model)
```

```
##
## Call:
## glm(formula = satell ~ color + spine + width + weight, family = poisson(link = "log"),
##     data = x)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.3435447  0.9684204  -0.355  0.72278
## color       -0.1849325  0.0665236  -2.780  0.00544 **
## spine        0.0399764  0.0568062   0.704  0.48160
## width        0.0275251  0.0479425   0.574  0.56588
## weight       0.0004725  0.0001649   2.865  0.00417 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
```

```
##
##     Null deviance: 632.79  on 172  degrees of freedom
## Residual deviance: 551.85  on 168  degrees of freedom
## AIC: 917.15
##
## Number of Fisher Scoring iterations: 6
```

Step 3: Predict the Estimated Satellite Counts
Using the fitted Poisson regression model, we will predict the estimated satellite counts for each observation in the dataset.

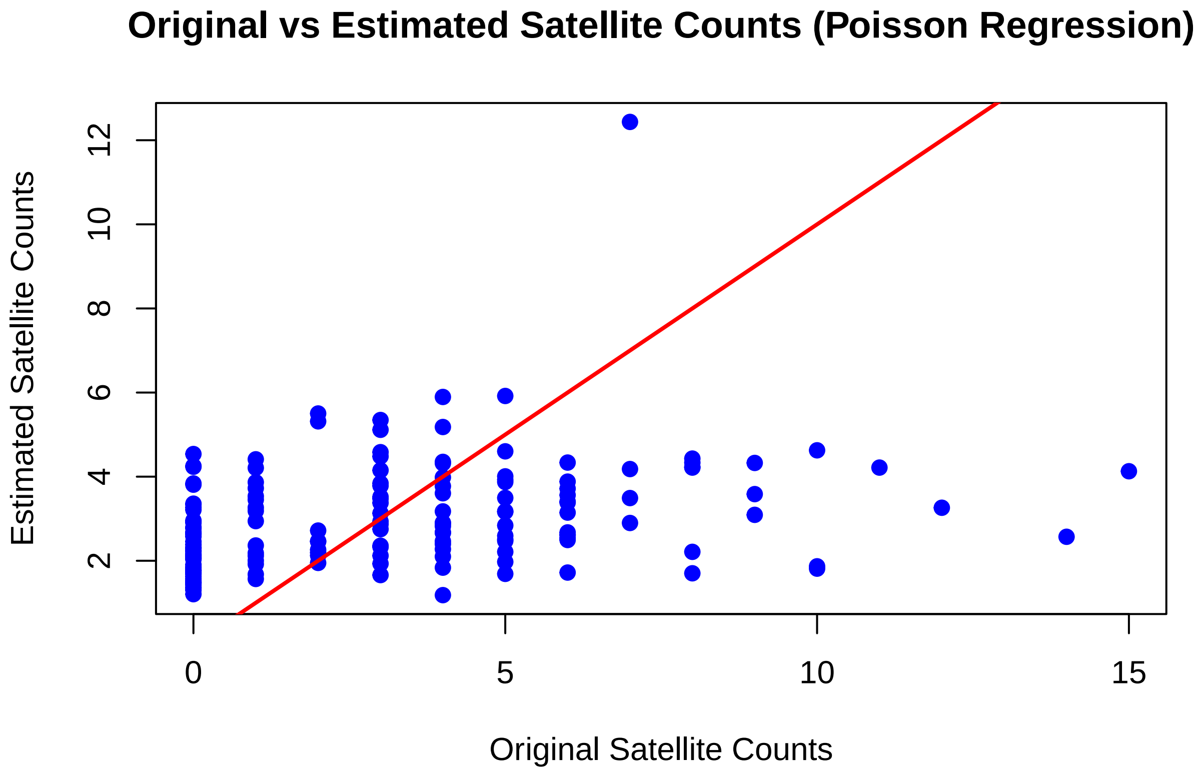
```r
# Predict the estimated satellite counts using the model
x$estimated_satell_poisson <- predict(log_model, type = "response")

# View the first few rows to verify
head(x)
```

```
##   color spine width satell weight y estimated_satell_poisson
## 1     3     3  28.3      8   3050 1                 4.227305
## 2     4     3  22.5      0   1550 0                 1.474457
## 3     2     1  26.0      9   2300 1                 3.092217
## 4     4     3  24.8      0   2100 0                 2.036949
## 5     4     3  26.0      4   2600 1                 2.666363
## 6     3     3  23.8      0   2100 0                 2.384192
```

Step 4: Compare Original vs Estimated Satellite Counts
We will plot the original satellite counts against the estimated values from the Poisson regression model to visually assess the model's accuracy.

```r
# Plot original vs estimated satellite counts
plot(x$satell, x$estimated_satell_poisson,
     main = "Original vs Estimated Satellite Counts (Poisson Regression)",
     xlab = "Original Satellite Counts",
     ylab = "Estimated Satellite Counts",
     pch = 19, col = "blue")
abline(a = 0, b = 1, col = "red", lwd = 2)
```



Step 5: Calculate the Sum of Squared Differences
To quantify the accuracy of the Poisson regression model, we will calculate the sum of squared differences between the original and estimated satellite counts.

```
# Calculate the sum of squared differences
ssd_poisson <- sum((x$satell - x$estimated_satell_poisson)^2)
cat("Sum of Squared Differences (Poisson Regression):", ssd_poisson, "\n")
```

```
## Sum of Squared Differences (Poisson Regression): 1508.47
```

Part B: Linear Regression Model

Step 1: Fit the Linear Regression Model
We will now fit a linear regression model to estimate the number of satellites.

```
# Fit the linear regression model
linear_model <- lm(satell ~ color + spine + width + weight, data = x)

# Display the summary of the model
summary(linear_model)
```

```
##
## Call:
## lm(formula = satell ~ color + spine + width + weight, data = x)
```

```
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -4.5184 -2.1476 -0.7021  1.5308 11.1840 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)  
## (Intercept) -1.2974108  4.6469810  -0.279   0.7804  
## color       -0.4137137  0.3111604  -1.330   0.1855  
## spine        0.0238591  0.2959169   0.081   0.9358  
## width        0.0547600  0.2323295   0.236   0.8140  
## weight       0.0016987  0.0008485   2.002   0.0469 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## Residual standard error: 2.942 on 168 degrees of freedom
## Multiple R-squared:  0.147,  Adjusted R-squared:  0.1267 
## F-statistic: 7.236 on 4 and 168 DF,  p-value: 2.12e-05
```

Step 2: Predict the Estimated Satellite Counts
Using the fitted linear regression model, we will predict the estimated satellite counts for each observation in the dataset.

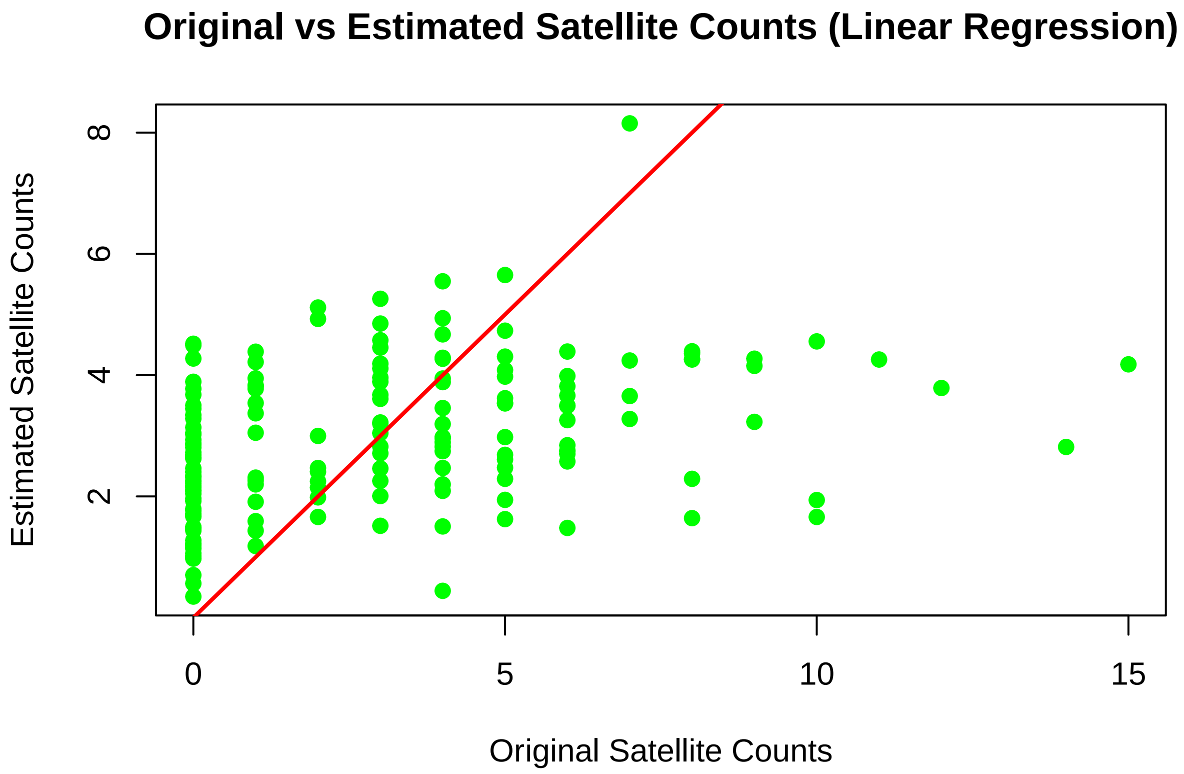
```r
# Predict the estimated satellite counts using the linear model
x$estimated_satell_linear <- predict(linear_model)

# View the first few rows to verify
head(x)
```

```
##   color spine width satell weight y estimated_satell_poisson
## 1     3     3  28.3      8   3050 1                 4.227305
## 2     4     3  22.5      0   1550 0                 1.474457
## 3     2     1  26.0      9   2300 1                 3.092217
## 4     4     3  24.8      0   2100 0                 2.036949
## 5     4     3  26.0      4   2600 1                 2.666363
## 6     3     3  23.8      0   2100 0                 2.384192
##   estimated_satell_linear
## 1               4.2636312
## 2               0.9843273
## 3               3.2296875
## 4               2.0445355
## 5               2.9595749
## 6               2.4034891
```

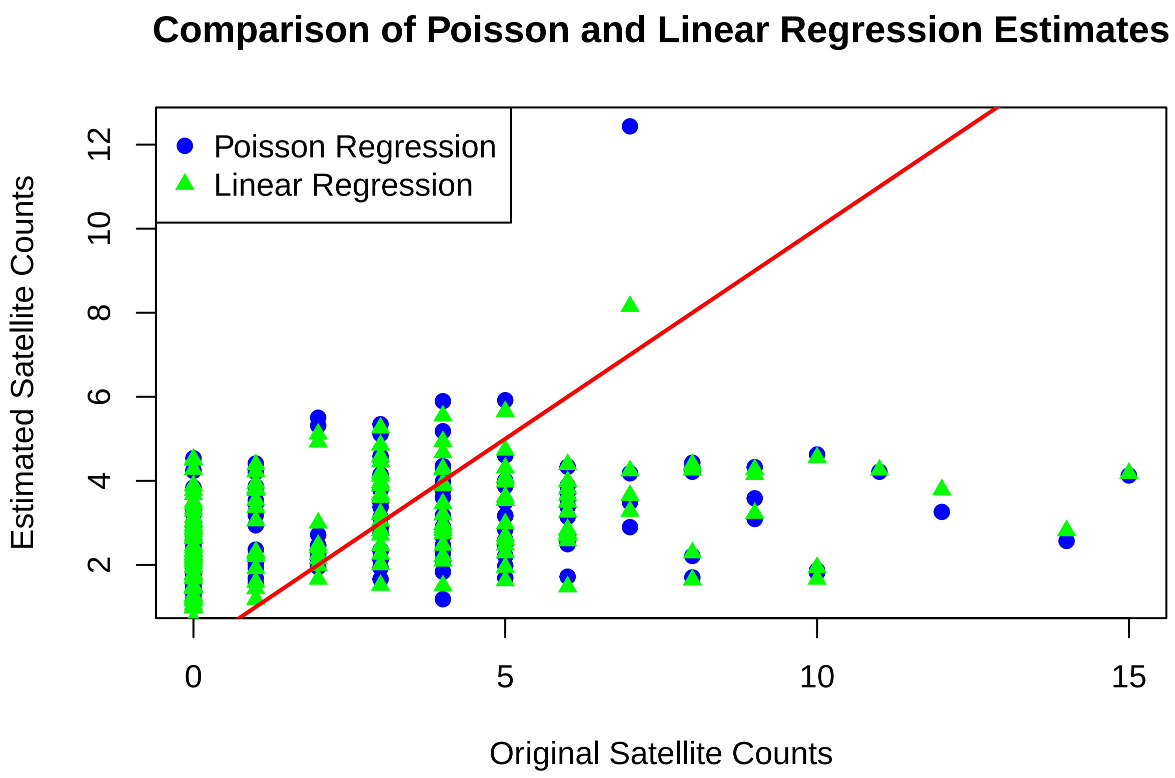
Step 3: Compare Original vs Estimated Satellite Counts (Linear Regression)
We will plot the original satellite counts against the estimated values from the linear regression model to visually assess the model’s accuracy.

```r
# Plot original vs estimated satellite counts for linear regression
plot(x$satell, x$estimated_satell_linear,
     main = "Original vs Estimated Satellite Counts (Linear Regression)",
     xlab = "Original Satellite Counts",
     ylab = "Estimated Satellite Counts",
     pch = 19, col = "green")
abline(a = 0, b = 1, col = "red", lwd = 2)
```



Step 4: Plot Both Models for Comparison
To evaluate the performance of both models, we will create a comparative plot of the original satellite counts against the estimated values from both the Poisson and linear regression models.

```
# Plot original vs estimated satellite counts for both models
plot(x$satell, x$estimated_satell_poisson,
     main = "Comparison of Poisson and Linear Regression Estimates",
     xlab = "Original Satellite Counts",
     ylab = "Estimated Satellite Counts",
     pch = 19, col = "blue")
points(x$satell, x$estimated_satell_linear, pch = 17, col = "green")
abline(a = 0, b = 1, col = "red", lwd = 2)
legend("topleft", legend = c("Poisson Regression", "Linear Regression"),
       col = c("blue", "green"), pch = c(19, 17))
```



Step 5: Calculate the Sum of Squared Differences
We will calculate the sum of squared differences between the original and estimated satellite counts for the linear regression model. This metric will allow us to quantitatively compare the accuracy of both models.

```
# Sum of squared differences for linear regression
ssd_linear <- sum((x$satell - x$estimated_satell_linear)^2)
cat("Sum of Squared Differences (Linear Regression):", ssd_linear, "\n")
```

```
## Sum of Squared Differences (Linear Regression): 1454.306
```

Conclusion

Poisson Regression Model: The sum of squared differences is 1508.47.

Linear Regression Model: The sum of squared differences is 1454.31.

By comparing the sum of squared differences and the plots, we can assess which model provides a better fit to the data.

```
print(paste("Poisson:", round(ssd_poisson, 2)))
```

```
## [1] "Poisson: 1508.47"
```

```r
print(paste("Linear:", round(ssd_linear, 2)))
```

```
## [1] "Linear: 1454.31"
```

```r
# Compare AIC values
aic_poisson <- AIC(log_model)
aic_linear <- AIC(linear_model)
cat("AIC (Poisson Regression):", aic_poisson, "\n")
```

```
## AIC (Poisson Regression): 917.151
```

```r
cat("AIC (Linear Regression):", aic_linear, "\n")
```

```
## AIC (Linear Regression): 871.2685
```

# Problem: Horseshoe Crab Data Analysis

We will analyze the Horseshoe crab data to perform the following tasks:

- Produce a table of horseshoe crab counts based on grouped width values.
- Run a Poisson Generalized Linear Model (GLM) using a log link function with an offset of "t" and the explanatory variable as "New width values" to find the expected mean of total satellites.
- Interpret the parameter estimates from the model summary.
- Determine the 95% confidence interval for the beta parameter.
- Find the expected satellite count for each group in the grouped table.
- Calculate the chi-square statistic and p-value for the model.
- Calculate the Pearson residuals to evaluate the model fit.

Step 1: Produce the Table of Horseshoe Crab Based on Grouped Width Values

First, we will load the data and group it based on the specified width ranges.

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

```
# Load the horseshoe crab data
data("table_4.3")
crab_data <- table_4.3

# Group the data based on the specified width ranges and summarize
grouped_data <- crab_data %>%
  mutate(width_group = cut(width,
                           breaks = c(-Inf, 23.25, 24.25, 25.25, 26.25, 27.25, 28.25, 29.25, Inf),
                           labels = c("<23.25", "23.25-24.25", "24.25-25.25", "25.25-26.25",
                                      "26.25-27.25", "27.25-28.25", "28.25-29.25", ">29.25"),
                           right = TRUE)) %>%
  group_by(width_group) %>%
  summarize(
    t = n(),
    total_satellites = sum(satell),
    new_width = mean(width)
  )

# View the grouped data
print(grouped_data)
```

```
## # A tibble: 8 x 4
##   width_group     t total_satellites new_width
##   <fct>       <int>            <int>     <dbl>
## 1 <23.25         14               14      22.7
## 2 23.25-24.25    14               20      23.8
## 3 24.25-25.25    28               67      24.8
## 4 25.25-26.25    39              105      25.8
## 5 26.25-27.25    22               63      26.8
## 6 27.25-28.25    24               93      27.7
## 7 28.25-29.25    18               71      28.7
## 8 >29.25         14               72      30.4
```

Explanation:

We use the `cut()` function to categorize the width into specified ranges. Then, we group the data by `width_group` and calculate the following:

- **t**: the number of cases in each group.
- **total_satellites**: the total number of satellites in each group.
- **new_width**: the mean width in each group.

Step 2: Run a Poisson GLM with Offset as "t" and Explanatory Variable as "New width values"

We will run a Poisson regression model using the grouped data, where: - `total_satellites` is the response variable, - `new_width` is the explanatory variable, and - `log(t)` is used as the offset in the model.

This model will help estimate the expected mean of total satellites based on the width of the horseshoe crabs.

```
# Run the Poisson GLM with log link and offset
poisson_model <- glm(total_satellites ~ new_width + offset(log(t)),
                     family = poisson(link = "log"),
                     data = grouped_data)

# Display the summary of the model
summary(poisson_model)
```

```
##
## Call:
## glm(formula = total_satellites ~ new_width + offset(log(t)),
##     family = poisson(link = "log"), data = grouped_data)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.54018    0.57658  -6.140 8.26e-10 ***
## new_width    0.17290    0.02125   8.135 4.11e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 72.3772  on 7  degrees of freedom
## Residual deviance:  6.5164  on 6  degrees of freedom
## AIC: 56.961
##
## Number of Fisher Scoring iterations: 4
```

Explanation:

- We use the `glm()` function to fit a Poisson regression model with a log link.
- The response variable is `total_satellites`.
- The explanatory variable is `new_width`.
- We include `offset(log(t))` to adjust for the number of cases in each group.

Step 3: Interpret the Parameter Values from the Summary Function

From the summary output, we interpret the coefficients as follows:

Interpretation:

- **Intercept ((Intercept))**: Represents the log of the expected mean number of satellites when `new_width` is zero. This may not be meaningful in this context but provides a baseline.
- **Slope (new_width)**: The coefficient for `new_width` indicates its effect on the log of the expected number of satellites.
  - A positive coefficient suggests that as `new_width` increases, the expected number of satellites also increases.
  - Specifically, for each unit increase in `new_width`, the log of the expected satellite count increases by the value of this coefficient.

Step 4: Determine 95% Confidence Interval for Beta

We calculate the 95% confidence intervals for the coefficients to assess the range within which the true values of the coefficients likely fall, providing insight into the precision and reliability of the estimated parameters.

```r
# Calculate 95% confidence intervals using standard errors
confint_default <- confint.default(poisson_model)

# Calculate 95% confidence intervals using profile likelihood
confint_profile <- confint(poisson_model)
```

```
## Waiting for profiling to be done...
```

```r
# Display the confidence intervals
cat("95% Confidence Intervals (Default Method):\n")
```

```
## 95% Confidence Intervals (Default Method):
```

```r
print(confint_default)
```

```
##                  2.5 %     97.5 %
## (Intercept) -4.6702579 -2.4100950
## new_width    0.1312427  0.2145525
```

```r
cat("\n95% Confidence Intervals (Profile Likelihood):\n")
```

```
##
## 95% Confidence Intervals (Profile Likelihood):
```

```r
print(confint_profile)
```

```
##                 2.5 %     97.5 %
## (Intercept) -4.674644 -2.4139791
## new_width    0.131246  0.2145746
```

Explanation:

- `confint.default()` calculates the confidence intervals using standard errors, which is a simpler method based on the normal approximation.
- `confint()` calculates the confidence intervals using the profile likelihood method, which can provide more accurate intervals, especially in smaller datasets.

The confidence intervals provide a range where we are 95% confident that the true parameter values lie, giving an indication of the precision of our estimates.

Step 5: Find the Expected Satellite Count for Each Group in the Grouped Table

Using the Poisson regression model, we calculate the expected number of satellites for each group. This involves predicting the satellite count based on the `new_width` values and adjusting for `t`, allowing us to assess how well the model estimates satellite counts across different width groups.

```r
# Add the expected satellites to the grouped data
grouped_data$expected_satellites <- predict(poisson_model, type = "response")

# View the updated grouped data
print(grouped_data)
```

```
## # A tibble: 8 x 5
##   width_group     t total_satellites new_width expected_satellites
##   <fct>       <int>            <int>     <dbl>               <dbl>
## 1 <23.25         14               14      22.7                20.5
## 2 23.25-24.25    14               20      23.8                25.1
## 3 24.25-25.25    28               67      24.8                58.9
## 4 25.25-26.25    39              105      25.8                98.6
## 5 26.25-27.25    22               63      26.8                65.6
## 6 27.25-28.25    24               93      27.7                84.2
## 7 28.25-29.25    18               71      28.7                74.2
## 8 >29.25         14               72      30.4                78.0
```

Explanation:

- We use the `predict()` function with `type = "response"` to get the expected counts. This provides the model's estimates of the total satellites for each group.

These predicted values represent the model's best estimate of the satellite counts based on the `new_width` variable and the grouped data.

Step 6: Calculate Chi-Square Statistic and the p-Value

We calculate the chi-square statistic to assess the goodness-of-fit of the model. This test helps determine whether the Poisson regression model adequately fits the data by comparing the observed satellite counts to the model's expected counts. The corresponding p-value will indicate if there is a significant difference between the observed and expected values, with a high p-value suggesting a good model fit.

```r
# Calculate chi-square statistic
chi_square <- sum((grouped_data$total_satellites - grouped_data$expected_satellites)^2 / grouped_data$e

# Degrees of freedom
df <- nrow(grouped_data) - length(coef(poisson_model))

# P-value
p_value <- pchisq(chi_square, df = df, lower.tail = FALSE)

cat("Chi-square statistic:", chi_square, "\n")
```

```
## Chi-square statistic: 6.246493
```

```r
cat("Degrees of freedom:", df, "\n")
```

```
## Degrees of freedom: 6
```

```
cat("P-value:", p_value, "\n")
```

```
## P-value: 0.396152
```

Explanation:

- The chi-square statistic is calculated as the sum over all groups of (observed − expected)^2/expected.
- Degrees of freedom are calculated as the number of groups minus the number of parameters estimated in the model.
- The p-value is obtained from the chi-square distribution with the calculated degrees of freedom.

A high p-value suggests that the model fits the data well, meaning there is no significant difference between the observed and expected satellite counts across the groups.

Step 7: Calculate Pearson Residuals

We calculate the Pearson residuals for each group. The Pearson residuals measure the standardized difference between the observed and expected counts, allowing us to identify any groups where the model may not fit well. These residuals can provide insight into specific observations that might be influencing the model's overall goodness-of-fit.

```
# Calculate Pearson residuals
grouped_data$pearson_residuals <- (grouped_data$total_satellites - grouped_data$expected_satellites) / s

# View the residuals
print(grouped_data[, c("width_group", "pearson_residuals")])
```

```
## # A tibble: 8 x 2
##   width_group pearson_residuals
##   <fct>                   <dbl>
## 1 <23.25                 -1.44
## 2 23.25-24.25            -1.01
## 3 24.25-25.25             1.06
## 4 25.25-26.25             0.647
## 5 26.25-27.25            -0.316
## 6 27.25-28.25             0.955
## 7 28.25-29.25            -0.370
## 8 >29.25                 -0.675
```

Explanation:

- The Pearson residuals measure the standardized difference between the observed and expected counts for each group.
- Residuals close to zero indicate a good fit for that group, suggesting that the model's predictions closely match the observed data.

Conclusion

- We have successfully produced a grouped table based on the width values of the horseshoe crabs.
- The Poisson GLM indicates that the width of the crabs has a significant effect on the number of satellites.
- The positive coefficient for `new_width` suggests that larger crabs tend to have more satellites, with satellite counts increasing as width increases.
- The chi-square goodness-of-fit test helps determine whether the model adequately fits the data, with a high p-value supporting a good fit.
- Pearson residuals provide further insight, helping to identify any groups where the model does not fit well, indicating potential areas for model refinement.

This analysis provides a comprehensive understanding of the relationship between crab width and satellite counts, supporting the use of Poisson regression for count data with grouped predictors.

Final Remarks

The Poisson regression model with a log link function and an offset provides a suitable approach for modeling count data, especially when counts are aggregated over groups of different sizes.

- This analysis indicates a positive relationship between crab width and the number of satellites, with larger crabs tending to have more satellites.
- Further diagnostic checks, such as residual analysis, can be performed to ensure the model's assumptions are satisfied and to identify any potential areas for improvement.

Answer to Specific Questions

Print Sum of Squared Differences:
Although not explicitly required in this problem, if you wish to calculate and print the sum of squared differences between the observed and expected counts (as done in the previous problem), you can do so with the following approach:

```
# Sum of Squared Differences
ssd <- sum((grouped_data$total_satellites - grouped_data$expected_satellites)^2)
cat("Sum of Squared Differences:", round(ssd, 2), "\n")
```

```
## Sum of Squared Differences: 304.61
```

Print Sum of Squared Differences for Each Model:

```
print(paste("Poisson Model Sum of Squared Differences:", round(ssd, 2)))
```

```
## [1] "Poisson Model Sum of Squared Differences: 304.61"
```