

ST346 Generalized Linear Models for Regression and Classification

- Coursework 1

2207969

2024-10-21

Contents

1	Data Loading and Exploration	1
2	Question 1: Weighted Regression	1
2.1	Part a: Data Visualization	1
2.2	Part b: Linear Model Fitting	2
2.3	Part c: Residual Analysis	3
2.4	Part d: Variance Estimation Table	4
2.5	Part e: Weight Calculation	4
2.6	Part f: Weighted Linear Model	5
2.7	Part g: Transformed Model	6
2.8	Part h: Residual Comparison	7
3	Question 2: Logistic Regression	8
3.1	Part a: Logistic Regression Model	8
3.2	Part b: Odds Ratio Calculation	8
3.3	Part c: Empirical Relationship Plot	9
3.4	Part d: Quadratic Logistic Regression	10
3.5	Part e: Probability Threshold Analysis	12

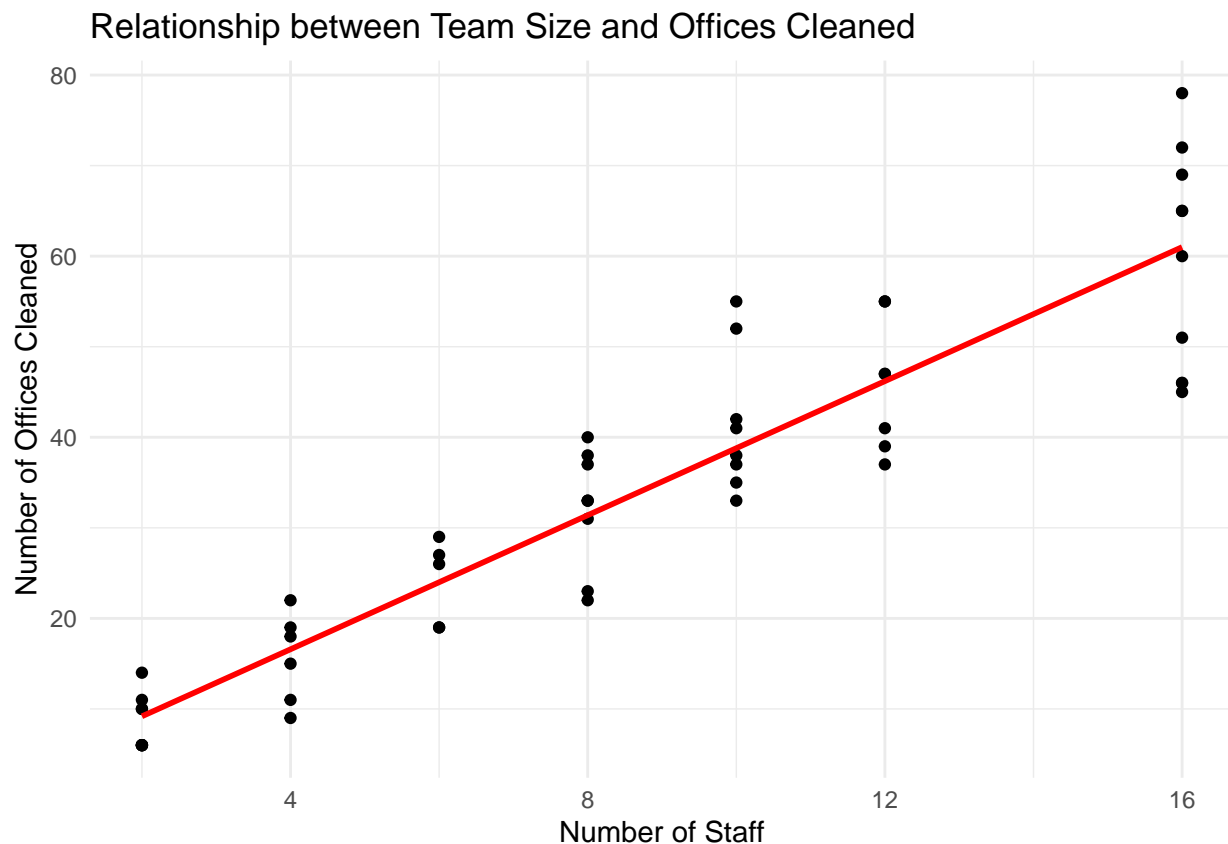
1 Data Loading and Exploration

```
# Load the dataset
load("CourseworkData1.rda")
```

2 Question 1: Weighted Regression

2.1 Part a: Data Visualization

```
# Visualize the relationship between team size and offices cleaned
plot <- ggplot(data=cleaners, aes(x = Staff, y = Offices)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(title = "Relationship between Team Size and Offices Cleaned",
       x = "Number of Staff",
       y = "Number of Offices Cleaned") +
  theme_minimal()
print(plot)
```



```
# Calculate correlation
correlation <- cor(cleaners$Staff, cleaners$Offices)
print(paste("Correlation coefficient:", round(correlation, 3)))
```

```
## [1] "Correlation coefficient: 0.926"
```

```
summary(cleaners)
```

```
##      Staff      Offices
##  Min.   : 2.000   Min.   : 6.00
##  1st Qu.: 4.000   1st Qu.:19.00
##  Median : 8.000   Median :35.00
##  Mean   : 8.679   Mean   :33.91
##  3rd Qu.:12.000   3rd Qu.:46.00
##  Max.   :16.000   Max.   :78.00
```

The scatter plot shows a positive correlation between team size and offices cleaned. As the number of staff increases, the number of offices cleaned also increases. The correlation coefficient is 0.926, very close to 1, showing a strong positive relation. It is reasonable to assume a linear relationship.

2.2 Part b: Linear Model Fitting

```
# Fit a normal linear model and interpret the slope coefficient
model <- lm(Offices ~ Staff, data = cleaners)
summary(model)
```

```
##
```

```
## Call:
## lm(formula = Offices ~ Staff, data = cleaners)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.9990  -4.9901   0.8046   4.0010  17.0010
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.7847     2.0965   0.851   0.399
## Staff         3.7009     0.2118  17.472 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.336 on 51 degrees of freedom
## Multiple R-squared:  0.8569, Adjusted R-squared:  0.854
## F-statistic: 305.3 on 1 and 51 DF,  p-value: < 2.2e-16
```

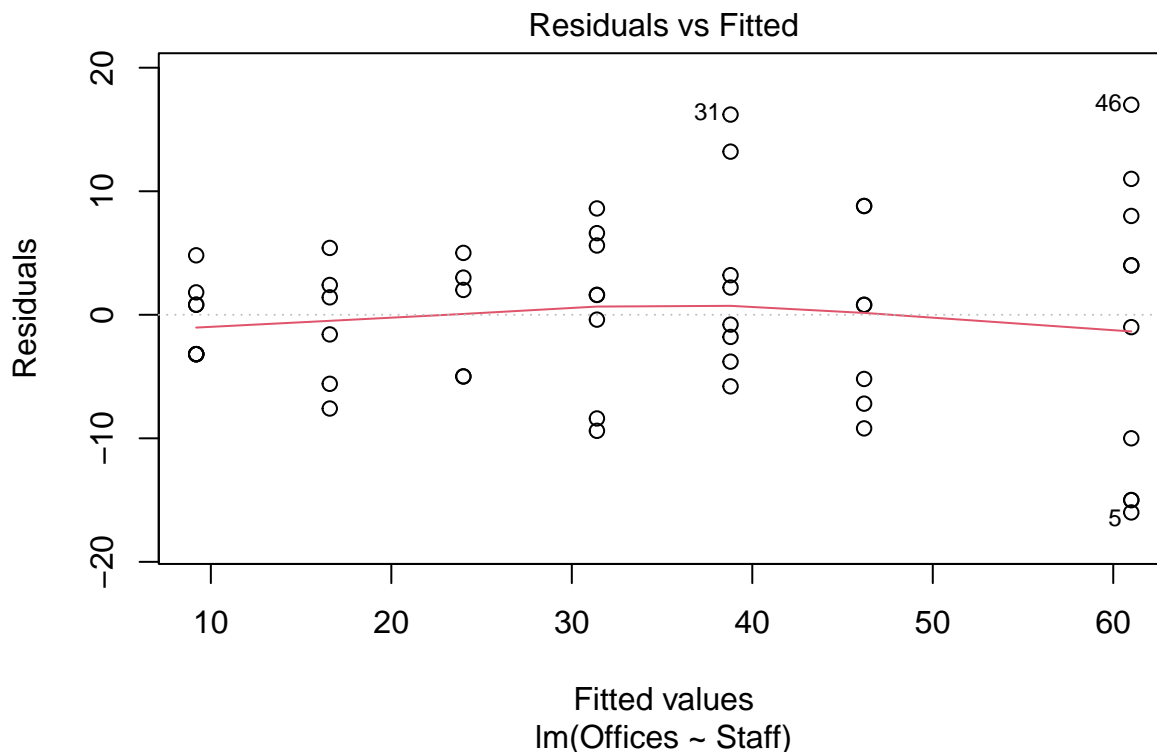
```
slope <- coef(model)["Staff"]
print(paste("Slope coefficient:", round(slope, 3)))
```

```
## [1] "Slope coefficient: 3.701"
```

The slope coefficient is 3.7, meaning on average for each additional cleaner, 3.7 offices can be cleaned.

2.3 Part c: Residual Analysis

```
# Produce and evaluate a residuals versus fitted values plot
plot(model, which=1)
```



The smoother resembles a horizontal line at zero. However, the residual plot resembles a ‘right-opening

megaphone'. The variation of the residuals is increasing with fitted value. It violates the assumption of homoscedasticity.

2.4 Part d: Variance Estimation Table

```
variance_table <- aggregate(Offices ~ Staff, data = cleaners, FUN = var)
names(variance_table) <- c("Team_size_x", "Variance")
variance_table$Variance <- round(variance_table$Variance, 2)
print(variance_table)
```

```
##   Team_size_x Variance
## 1           2     9.00
## 2           4    24.67
## 3           6    22.00
## 4           8    44.12
## 5          10    62.84
## 6          12    53.14
## 7          16   144.01
```

This produces the same variance estimates as given in the question.

2.5 Part e: Weight Calculation

Derivation of weight expression:

Since we assume that $w_i = 1$ for $x_i = 2$ and $\text{Var}(Y_i|x_i) = \text{Var}(Y_j|x_j)$ if $x_i = x_j$,

$$\begin{aligned}\text{Var}(Y_i|x_i) &= \phi/w_i \\ w_i &= \phi/\text{Var}(Y_i|x_i) \\ \text{Var}(Y_i|x=2) &= \phi/1 = \phi \\ w_i &= \frac{\phi/\text{Var}(Y_i|x_i)}{\phi/\text{Var}(Y|x=2)} = \frac{\text{Var}(Y|x=2)}{\text{Var}(Y_i|x_i)}\end{aligned}$$

```
# Use the derived expression to compute estimates
base_variance <- variance_table$Variance[variance_table$Team_size_x == 2]
variance_table$Weight <- base_variance / variance_table$Variance
print(variance_table)
```

```
##   Team_size_x Variance   Weight
## 1           2     9.00 1.00000000
## 2           4    24.67 0.36481557
## 3           6    22.00 0.40909091
## 4           8    44.12 0.20398912
## 5          10    62.84 0.14322088
## 6          12    53.14 0.16936394
## 7          16   144.01 0.06249566
```

```
assign_weight <- function(staff_size) {
  weight <- variance_table$Weight[variance_table$Team_size_x == staff_size]
  return(weight)
}
```

```
cleaners$Weight <- sapply(cleaners$Staff, assign_weight)
head(cleaners)
```

```
##   Staff Offices      Weight
## 1    16         51 0.06249566
## 2    10         37 0.14322088
## 3    12         37 0.16936394
## 4    16         46 0.06249566
## 5    16         45 0.06249566
## 6     4         11 0.36481557
```

```
summary(cleaners$Weight)
```

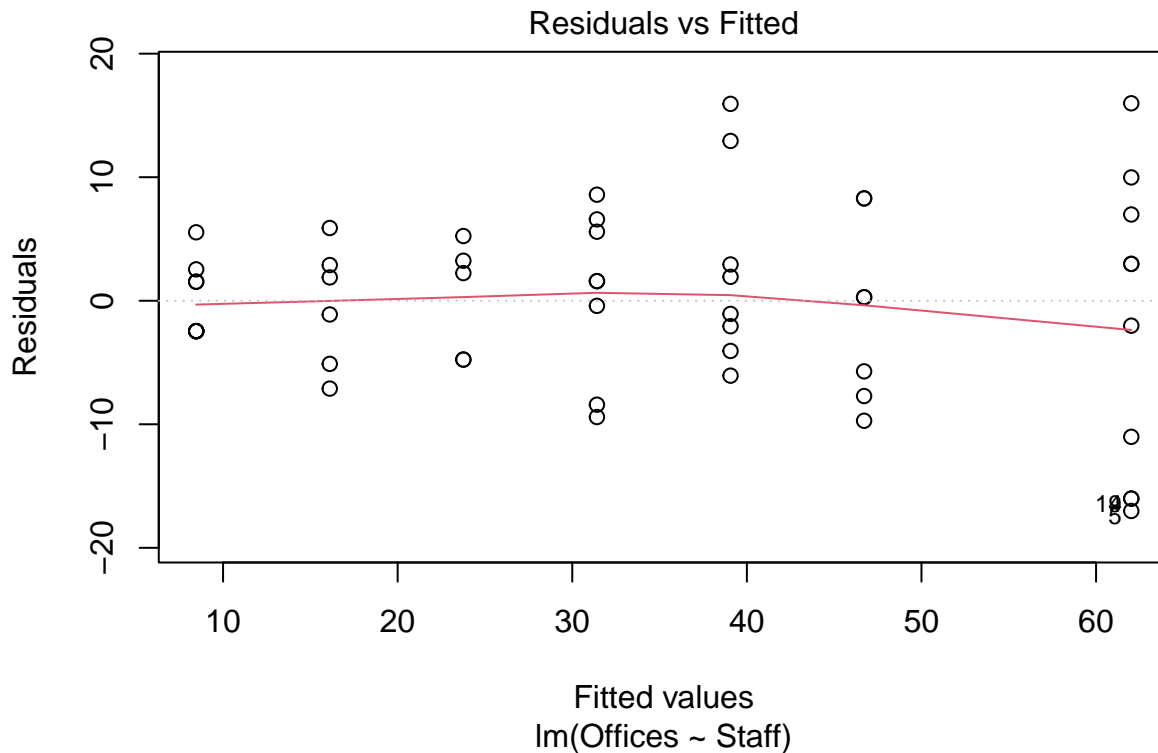
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.0625  0.1432  0.2040  0.3363  0.4091  1.0000
```

2.6 Part f: Weighted Linear Model

```
# Model summary
weighted_model <- lm(Offices ~ Staff, data = cleaners, weights = Weight)
summary(weighted_model)
```

```
##
## Call:
## lm(formula = Offices ~ Staff, data = cleaners, weights = Weight)
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -4.2952 -2.4604  0.1173  2.0709  6.0309
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.8095     1.1158   0.725   0.471
## Staff         3.8255     0.1788  21.400 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.894 on 51 degrees of freedom
## Multiple R-squared:  0.8998, Adjusted R-squared:  0.8978
## F-statistic:  458 on 1 and 51 DF,  p-value: < 2.2e-16
```

```
# Weighted residual plot
plot(weighted_model, which=1)
```



The smoother resembles a horizontal line at zero. No clear ‘right-opening megaphone’ pattern. The variation of the residuals is rather stable with fitted value. It follows the assumption of homoscedasticity and linearity. The slope coefficient is 3.826, meaning on average for each additional cleaner, 3.826 offices can be cleaned. It is higher than that of the unweighted model, suggesting the unweighted model underestimates the impact of staff number. Overall, the weighted model is more suitable for this dataset.

2.7 Part g: Transformed Model

```
# Transform the model
cleaners$Y_star <- cleaners$Offices * sqrt(cleaners$Weight)
cleaners$x_star <- cleaners$Staff * sqrt(cleaners$Weight)
cleaners$intercept_star <- sqrt(cleaners$Weight)
transformed_model <- lm(Y_star ~ 0 + intercept_star + x_star, data = cleaners)

# Report model summary
summary(transformed_model)
```

```
##
## Call:
## lm(formula = Y_star ~ 0 + intercept_star + x_star, data = cleaners)
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
	-4.2952	-2.4604	0.1173	2.0709	6.0309

```
##
## Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
intercept_star	0.8095	1.1158	0.725	0.471
x_star	3.8255	0.1788	21.400	<2e-16 ***

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.894 on 51 degrees of freedom
## Multiple R-squared:  0.9617, Adjusted R-squared:  0.9602
## F-statistic: 639.6 on 2 and 51 DF,  p-value: < 2.2e-16
```

```
cat("Transformed model coefficients:\n")
```

```
## Transformed model coefficients:
```

```
print(coef(transformed_model))
```

```
## intercept_star      x_star
##      0.8094971      3.8254605
```

```
cat("\nWeighted model coefficients:\n")
```

```
##
```

```
## Weighted model coefficients:
```

```
print(coef(weighted_model))
```

```
## (Intercept)      Staff
##      0.8094971      3.8254605
```

```
cat("\nTransformed model standard errors:\n")
```

```
##
```

```
## Transformed model standard errors:
```

```
print(sqrt(diag(vcov(transformed_model))))
```

```
## intercept_star      x_star
##      1.1157924      0.1787589
```

```
cat("\nWeighted model standard errors:\n")
```

```
##
```

```
## Weighted model standard errors:
```

```
print(sqrt(diag(vcov(weighted_model))))
```

```
## (Intercept)      Staff
##      1.1157924      0.1787589
```

2.8 Part h: Residual Comparison

```
# Verify that the weighted and transformed model residuals are the same
```

```
weighted_residuals <- residuals(weighted_model, type = "response") * sqrt(cleaners$Weight)
```

```
transformed_residuals <- residuals(transformed_model)
```

```
residual_difference <- weighted_residuals - transformed_residuals
```

```
cat("Summary of residual differences:\n")
```

```
## Summary of residual differences:
```

```
print(summary(residual_difference))
```

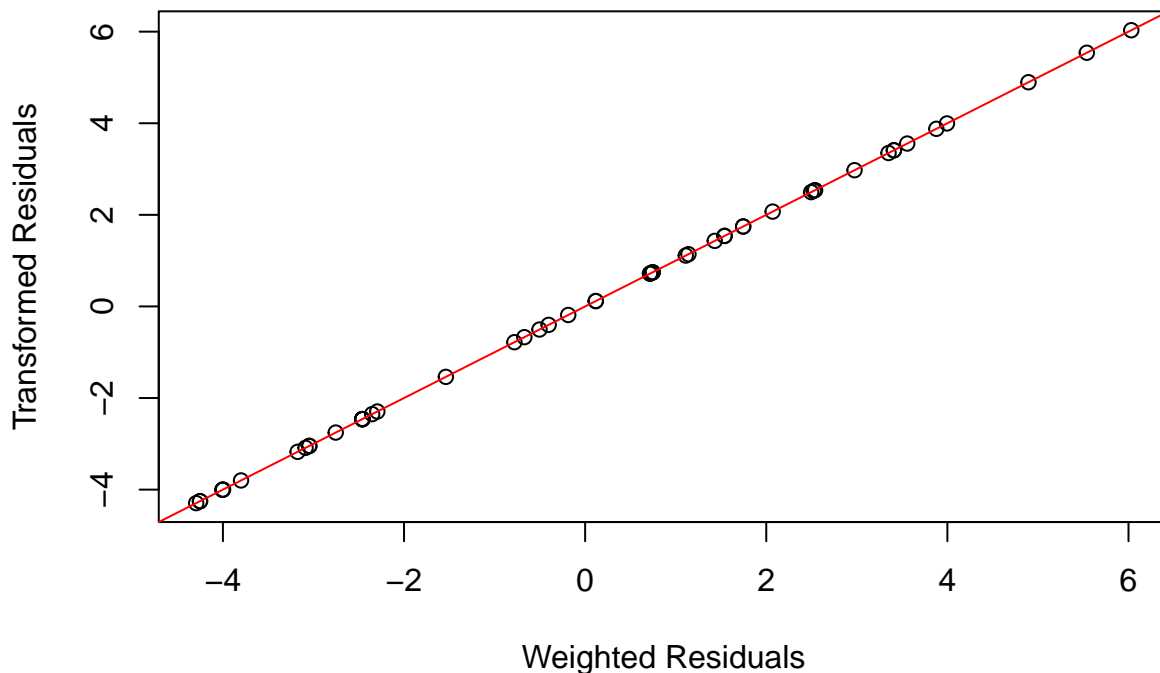
```
##      Min.      1st Qu.      Median      Mean      3rd Qu.      Max.
## -2.220e-16  0.000e+00  0.000e+00  8.903e-18  0.000e+00  4.441e-16
```

```
all_same <- all(abs(residual_difference) < 1e-10)
cat("\nAre all residuals essentially the same? ", all_same)
```

```
##
## Are all residuals essentially the same? TRUE

plot(weighted_residuals, transformed_residuals,
     main = "Weighted vs Transformed Residuals",
     xlab = "Weighted Residuals", ylab = "Transformed Residuals")
abline(0, 1, col = "red")
```

Weighted vs Transformed Residuals



```
residual_correlation <- cor(weighted_residuals, transformed_residuals)
cat("\nCorrelation between residuals: ", residual_correlation)
```

```
##
## Correlation between residuals: 1
```

3 Question 2: Logistic Regression

3.1 Part a: Logistic Regression Model

Use logistic function to turn linear predictor into probability:

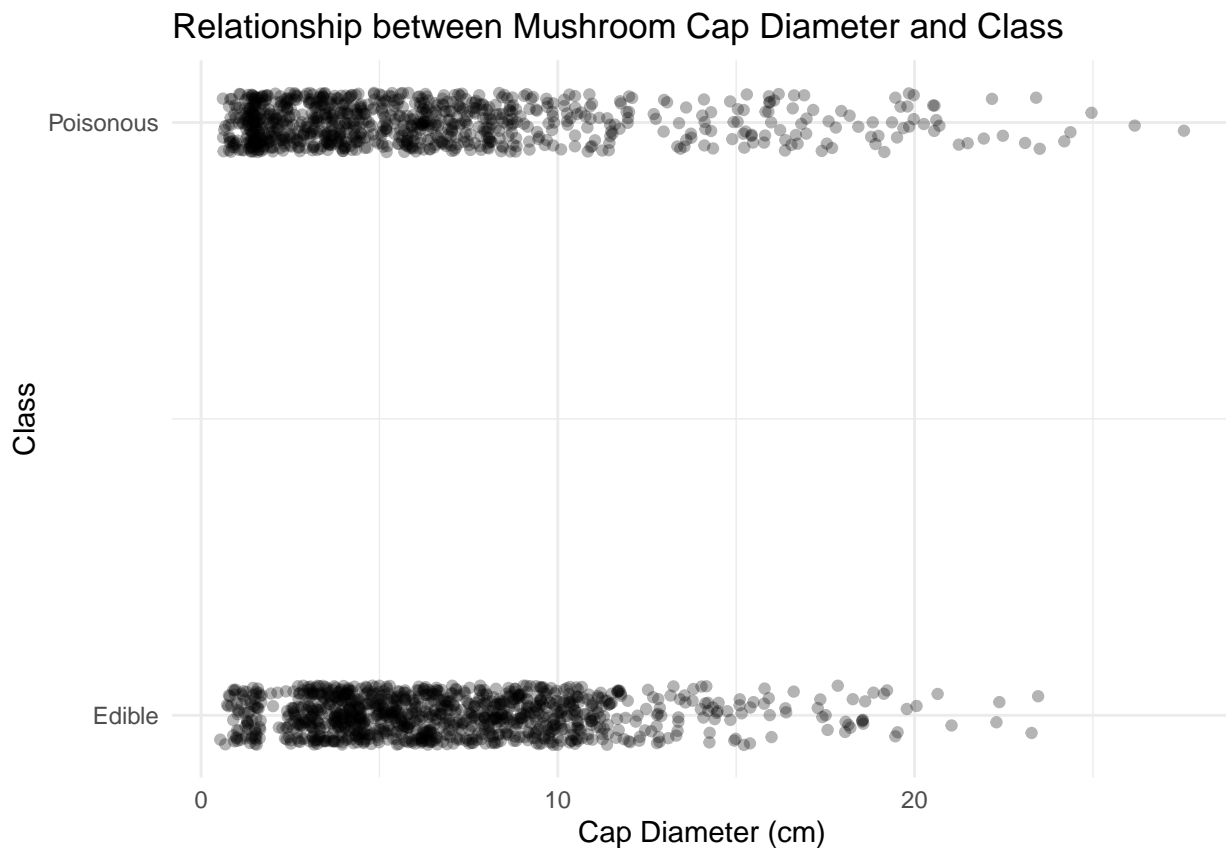
$$P(\text{Poisonous} | \text{Cup Diameter}_i) = h(\eta_i) = \frac{1}{1 + \exp(-\alpha - \beta * \text{Diameter}_i)}$$

3.2 Part b: Odds Ratio Calculation

$$\text{odds ratio} = \text{odds}(\text{Diameter} = 10) / \text{odds}(\text{Diameter} = 5) = \exp(\beta * (10 - 5)) = \exp(5\beta)$$

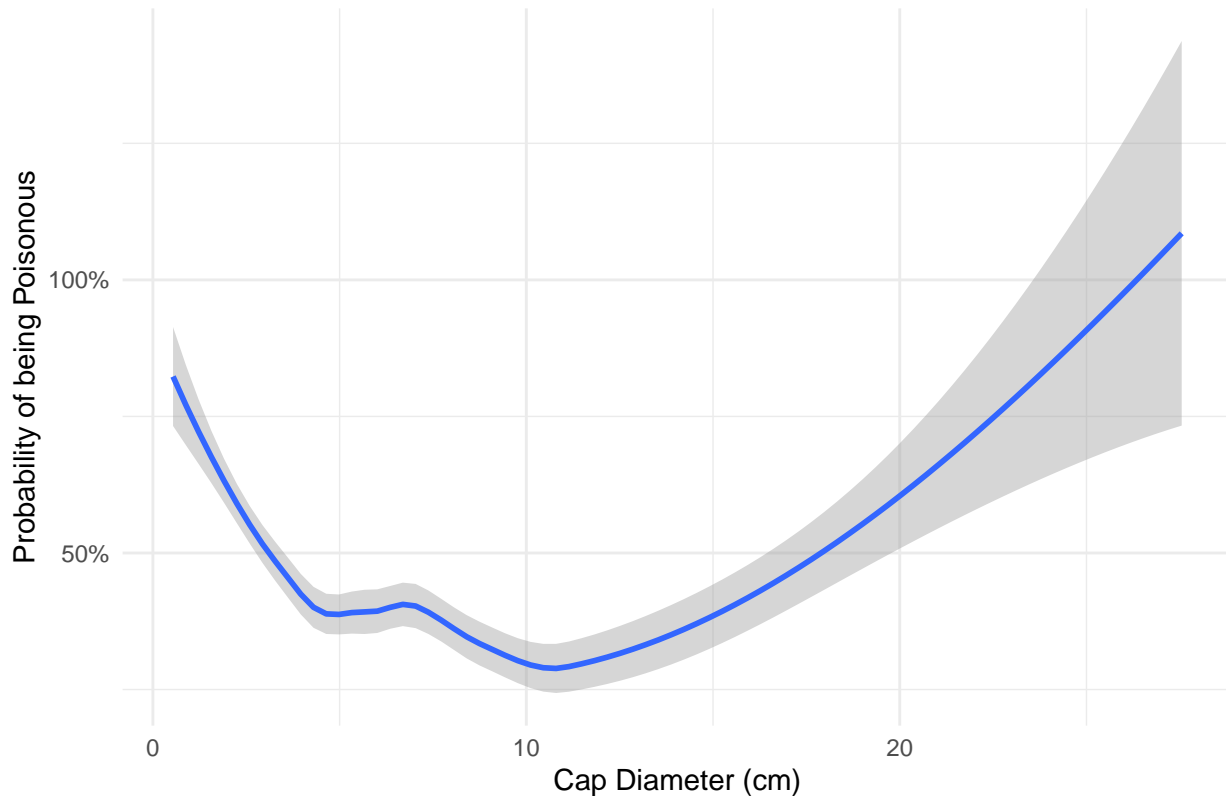
3.3 Part c: Empirical Relationship Plot

```
p1 <- ggplot(mushrooms, aes(x = Diameter, y = as.numeric(Class) - 1)) +  
  geom_jitter(alpha = 0.3, height = 0.05) +  
  scale_y_continuous(breaks = c(0, 1), labels = c("Edible", "Poisonous")) +  
  labs(title = "Relationship between Mushroom Cap Diameter and Class",  
        x = "Cap Diameter (cm)", y = "Class") +  
  theme_minimal()  
  
p2 <- ggplot(mushrooms, aes(x = Diameter, y = as.numeric(Class) - 1)) +  
  geom_smooth(method = "loess", se = TRUE) +  
  scale_y_continuous(labels = scales::percent) +  
  labs(title = "Smoothed Probability of Mushroom being Poisonous",  
        x = "Cap Diameter (cm)", y = "Probability of being Poisonous") +  
  theme_minimal()  
  
print(p1)
```



```
print(p2)
```

Smoothed Probability of Mushroom being Poisonous



```
summary_data <- mushrooms %>%
  group_by(Diameter) %>%
  summarise(prob_poisonous = mean(Class == "poisonous"))
head(summary_data)
```

```
## # A tibble: 6 x 2
##   Diameter prob_poisonous
##   <dbl>         <dbl>
## 1     0.54             0
## 2     0.61             1
## 3     0.62             1
## 4     0.63             1
## 5     0.65             1
## 6     0.68             0.5
```

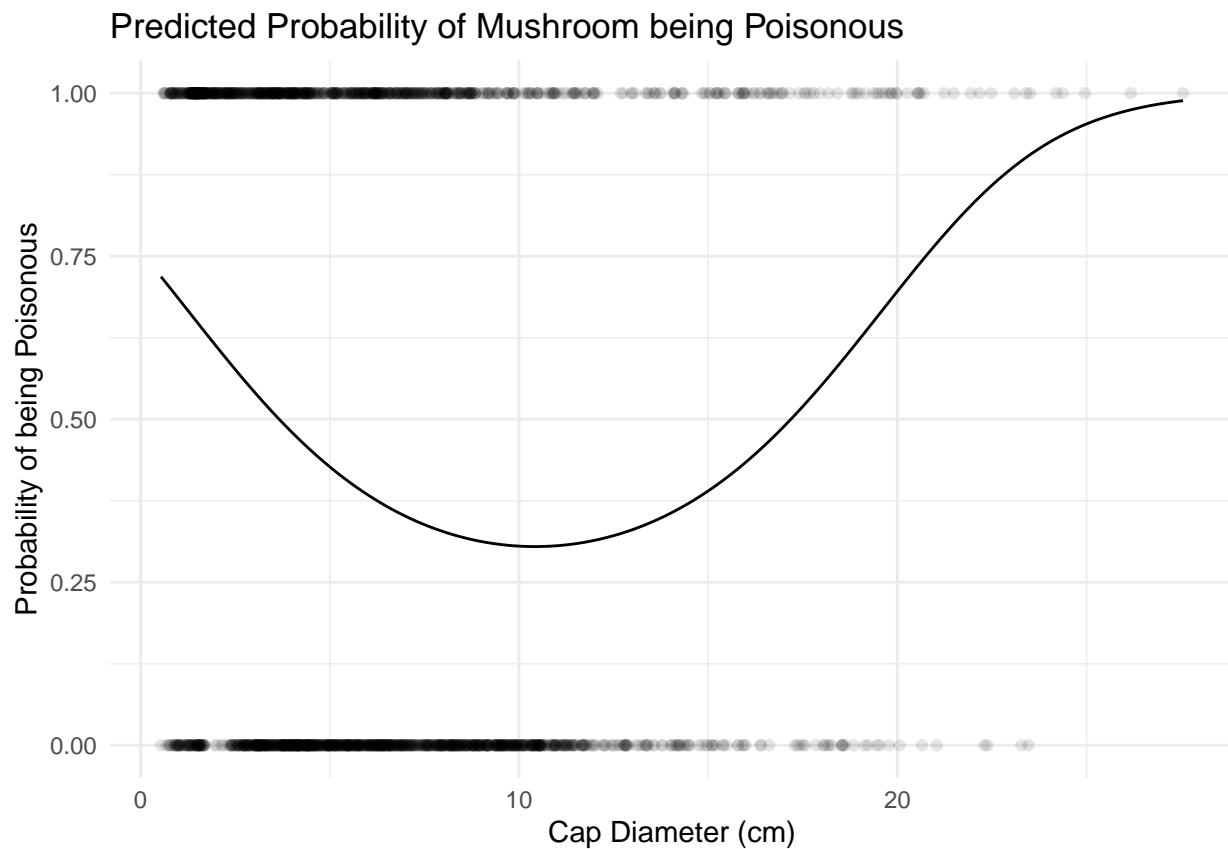
3.4 Part d: Quadratic Logistic Regression

```
quadratic_model <- glm(Class ~ Diameter + I(Diameter^2), data = mushrooms, family = binomial)
new_data <- data.frame(Diameter = seq(min(mushrooms$Diameter), max(mushrooms$Diameter), length.out = 20))
new_data$Probability <- predict(quadratic_model, newdata = new_data, type = "response")

p <- ggplot(new_data, aes(x = Diameter, y = Probability)) +
  geom_line() +
  geom_point(data = mushrooms, aes(y = as.numeric(Class) - 1), alpha = 0.1) +
  labs(title = "Predicted Probability of Mushroom being Poisonous",
       x = "Cap Diameter (cm)",
       y = "Probability of being Poisonous") +
```

```
theme_minimal()
```

```
print(p)
```



```
summary(quadratic_model)
```

```
##
## Call:
## glm(formula = Class ~ Diameter + I(Diameter^2), family = binomial,
##      data = mushrooms)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.135480   0.132419   8.575   <2e-16 ***
## Diameter      -0.376103   0.034742 -10.826   <2e-16 ***
## I(Diameter^2)  0.018040   0.001847   9.770   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 2752.2  on 1999  degrees of freedom
## Residual deviance: 2618.8  on 1997  degrees of freedom
## AIC: 2624.8
##
## Number of Fisher Scoring iterations: 4
```

3.5 Part e: Probability Threshold Analysis

```
diameters <- seq(round(min(mushrooms$Diameter) * 2) / 2,
                 round(max(mushrooms$Diameter) * 2) / 2,
                 by = 0.5)
pred_data <- data.frame(Diameter = diameters)
pred_data$Probability <- predict(quadratic_model, newdata = pred_data, type = "response")
safe_diameters <- pred_data[pred_data$Probability < 0.5, ]

cat("Cap diameters (to nearest 0.5 cm) where predicted probability of being poisonous is below 50%:\n")

## Cap diameters (to nearest 0.5 cm) where predicted probability of being poisonous is below 50%:
print(safe_diameters$Diameter)

## [1] 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0 8.5 9.0 9.5 10.0 10.5 11.0
## [16] 11.5 12.0 12.5 13.0 13.5 14.0 14.5 15.0 15.5 16.0 16.5 17.0

ggplot(pred_data, aes(x = Diameter, y = Probability)) +
  geom_line() +
  geom_point(data = safe_diameters, color = "red", size = 3) +
  geom_hline(yintercept = 0.5, linetype = "dashed", color = "blue") +
  labs(title = "Predicted Probability of Mushroom being Poisonous",
       subtitle = "Red points indicate diameters with <50% probability",
       x = "Cap Diameter (cm)",
       y = "Probability of being Poisonous") +
  theme_minimal()
```

