Assignment2_Group2

Mercy Cheptoo

2025-03-22

GROUP MEMBERS

1. Mercy Cheptoo 21/05823

2. Edwin Mucheru 21/06264

3. Celine Salesa 21/08374

4. JoyComfort Wangari 21/05738

```
library(readxl)
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(ggplot2)
library(gridExtra)
```

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

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

```
library(corrplot)
```

```
## corrplot 0.95 loaded
```

IMPORTING THE DATASET

```
data <- read.csv("/Users/cheptoo/Downloads/ 4.2/ STATISTICAL PROGRAMMING/ ASSIGNMENTS/epidemiology_data.csv")
View(data)
head(data)
```

```
##   population     time  region  exposure cases
## 1     150913 4.182455 Central 54.019296   204
## 2     396270 4.297973   South 88.159767   809
## 3     210399 3.583901    East  3.753984    43
## 4     442679 1.719756   North 88.143396   614
## 5     470829 4.289088 Central 72.234967   434
## 6      32323 2.602124    East 76.981030   130
```

```
summary(data)
```

```
##    population          time          region             exposure
##  Min.   : 10228   Min.   :1.011   Length:500         Min.   : 0.3535
##  1st Qu.:130538   1st Qu.:3.340   Class :character   1st Qu.:23.4257
##  Median :243512   Median :5.613   Mode  :character   Median :49.4872
##  Mean   :252689   Mean   :5.493                      Mean   :49.4344
##  3rd Qu.:369120   3rd Qu.:7.797                      3rd Qu.:74.3038
##  Max.   :499708   Max.   :9.954                      Max.   :99.9274
##      cases
##  Min.   :   0.00
##  1st Qu.:  63.25
##  Median : 206.00
##  Mean   : 286.70
##  3rd Qu.: 410.25
##  Max.   :1981.00
```

```
colnames(data)
```

```
## [1] "population" "time"       "region"     "exposure"   "cases"
```

```
sum(duplicated(data))
```

```
## [1] 0
```

```
sum(is.na(data))
```

```
## [1] 0
```

- There are no duplicates and no missing values

1. EDA

a. Summarize the distribution of cases

```
# Summarize the cases variable
summary(data$cases)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   63.25  206.00  286.70  410.25 1981.00
```

```
# Calculate additional statistics
mean_cases <- mean(data$cases, na.rm = TRUE)
sd_cases <- sd(data$cases, na.rm = TRUE)
range_cases <- range(data$cases, na.rm = TRUE)

mean_cases
```

```
## [1] 286.702
```

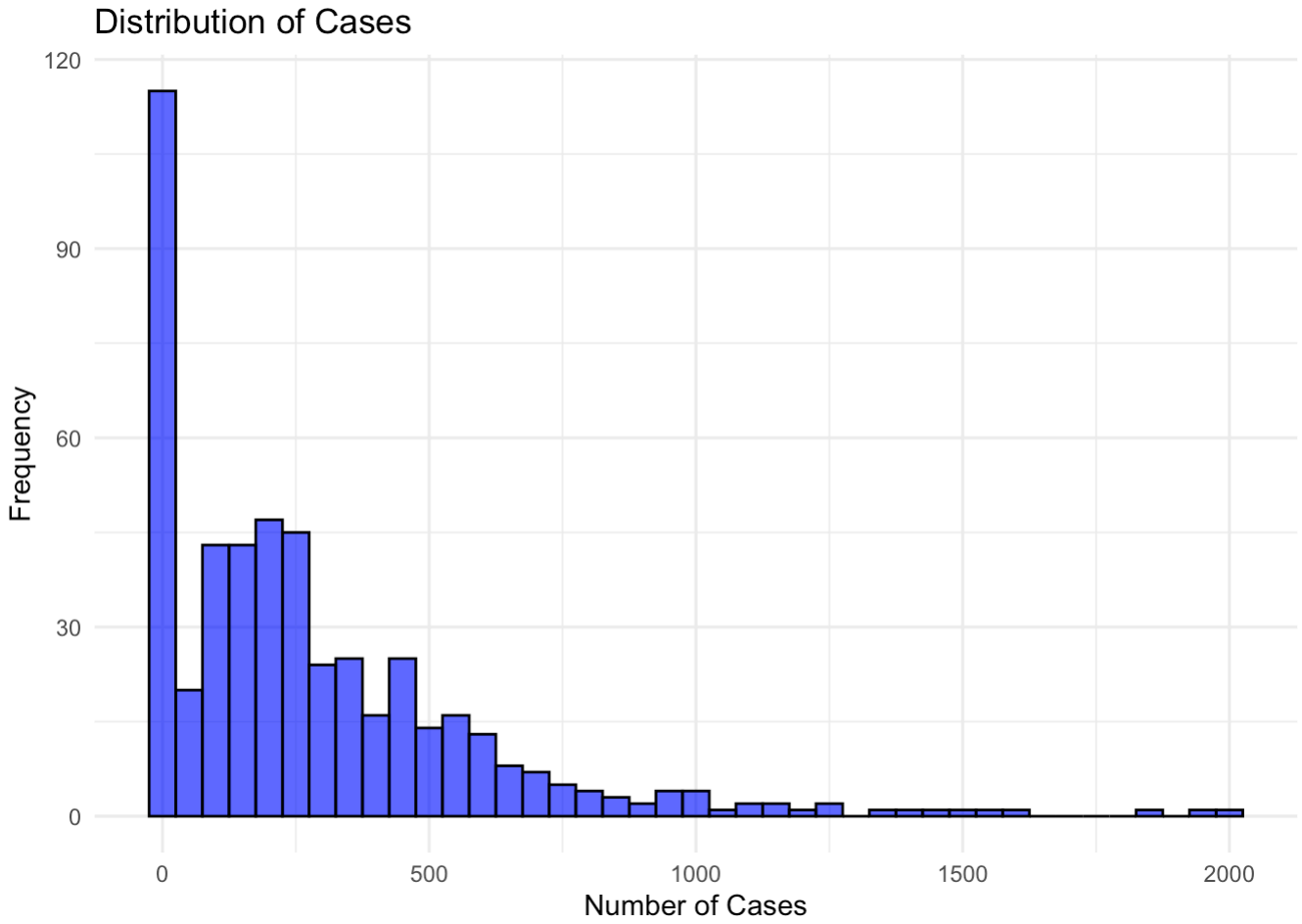
```
sd_cases
```

```
## [1] 316.2083
```

```
range_cases
```

```
## [1]    0 1981
```

```
ggplot(data, aes(x = cases)) +
  geom_histogram(binwidth = 50, fill = "blue", color = "black", alpha = 0.7) +
  labs(title = "Distribution of Cases", x = "Number of Cases", y = "Frequency") +
  theme_minimal()
```



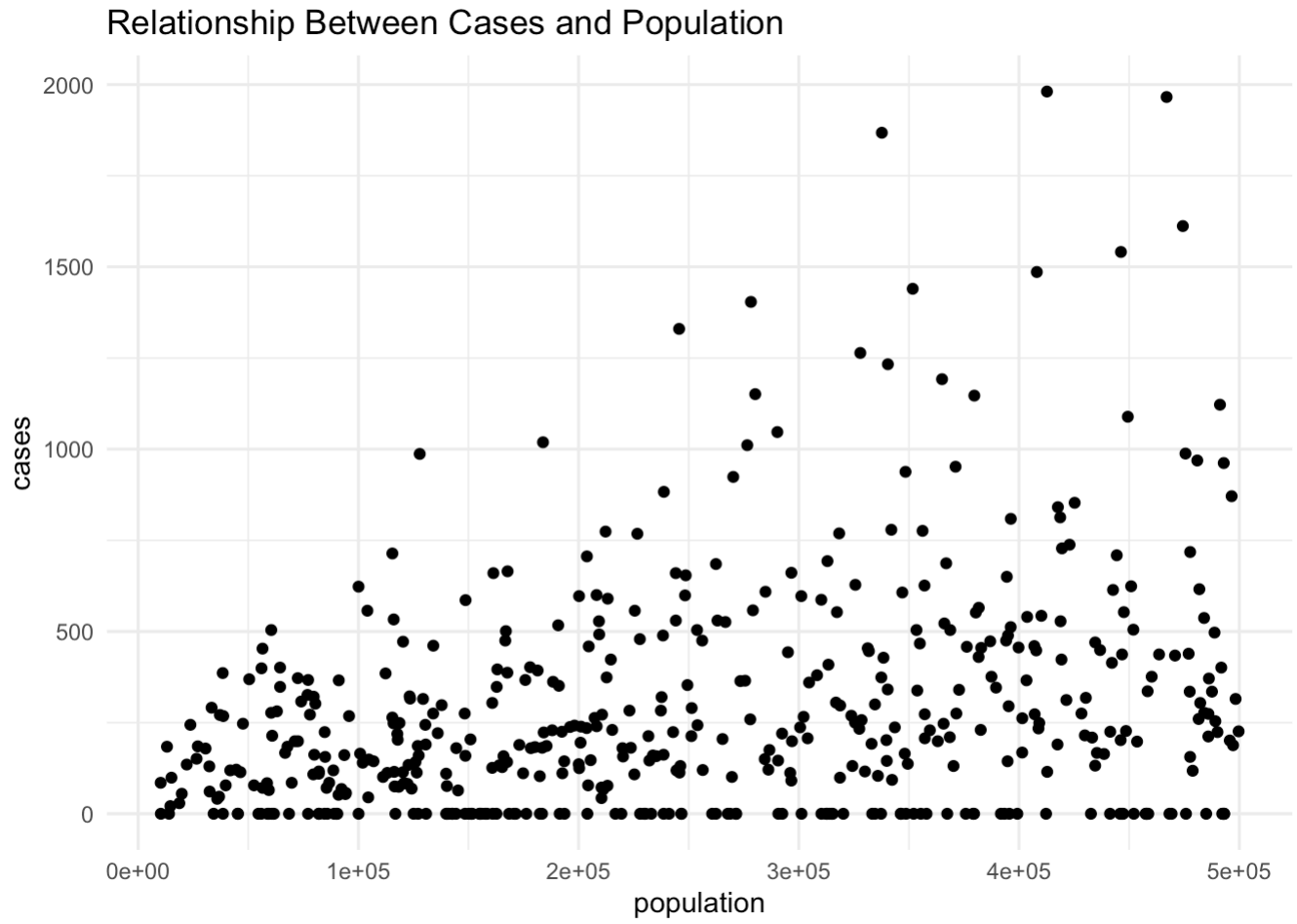
Interpretation

- Excess Zeros: The large spike at 0 indicates that many observations have zero cases. This suggests there might be a significant portion of the data where no cases were reported. This could represent instances where no incidents were recorded, which is important for further analysis, particularly when considering the zero-inflated nature of the data.
- Skewed Distribution: After the initial spike at 0, the distribution shows a rapid decline and tapers off slowly. This suggests a positive skew, where the majority of observations have low counts of cases, and only a few observations have high counts. This is typical for disease-related or rare event data, where a small number of cases may represent the majority of occurrences, but most locations or times report relatively few cases.
- Possible Outliers: There are a few bars far from the bulk of the data, especially on the right side, indicating some high case counts. These outliers may warrant closer examination to ensure they are not data entry errors or represent significant events requiring further analysis.
- Spread of Data: Most of the cases fall in the lower range of the x-axis (from 0 to about 500), with fewer cases as the number increases. This suggests that the occurrence of large numbers of cases is less frequent.

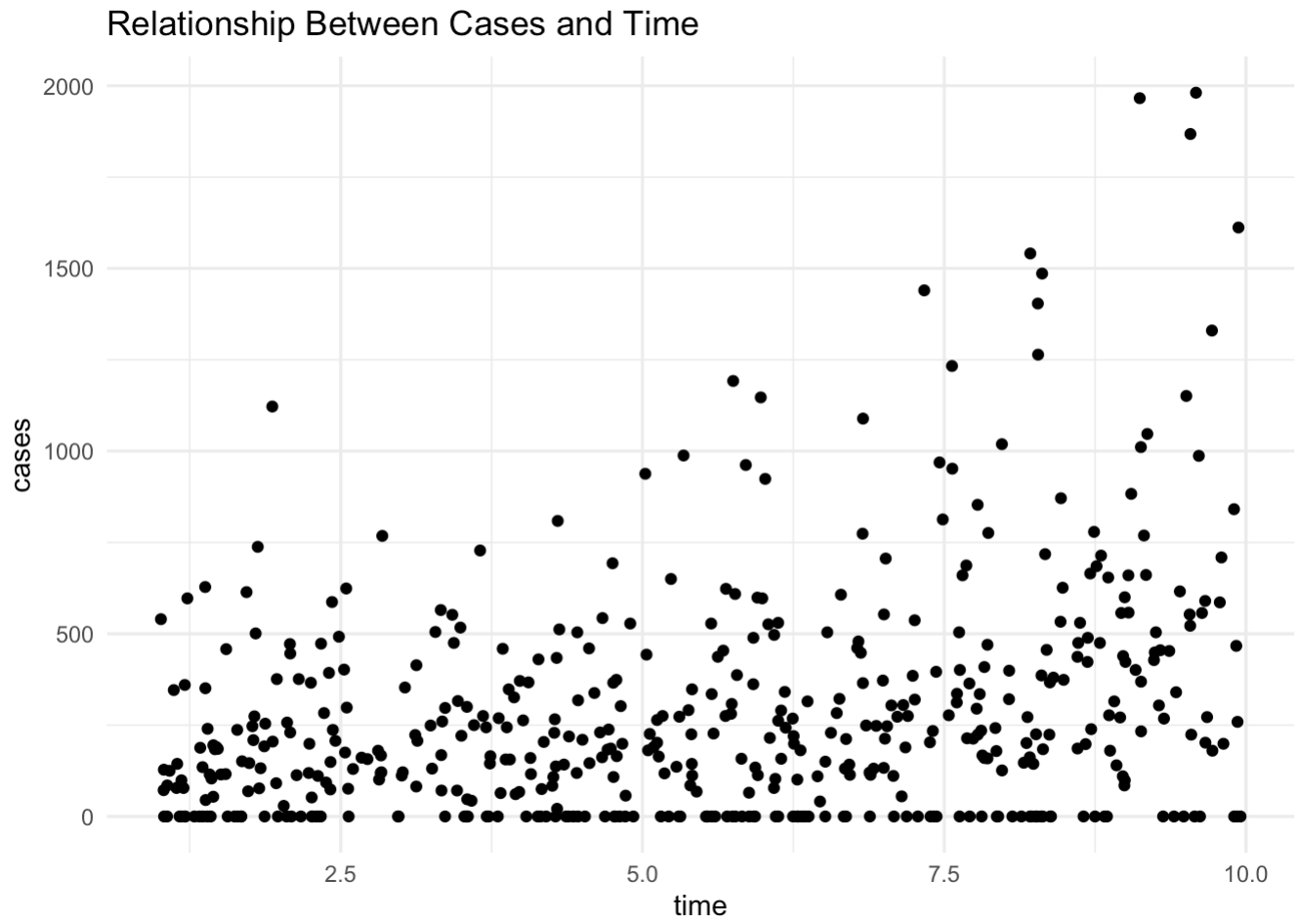
b. Visually and otherwise explore relationships

between cases and each other variable

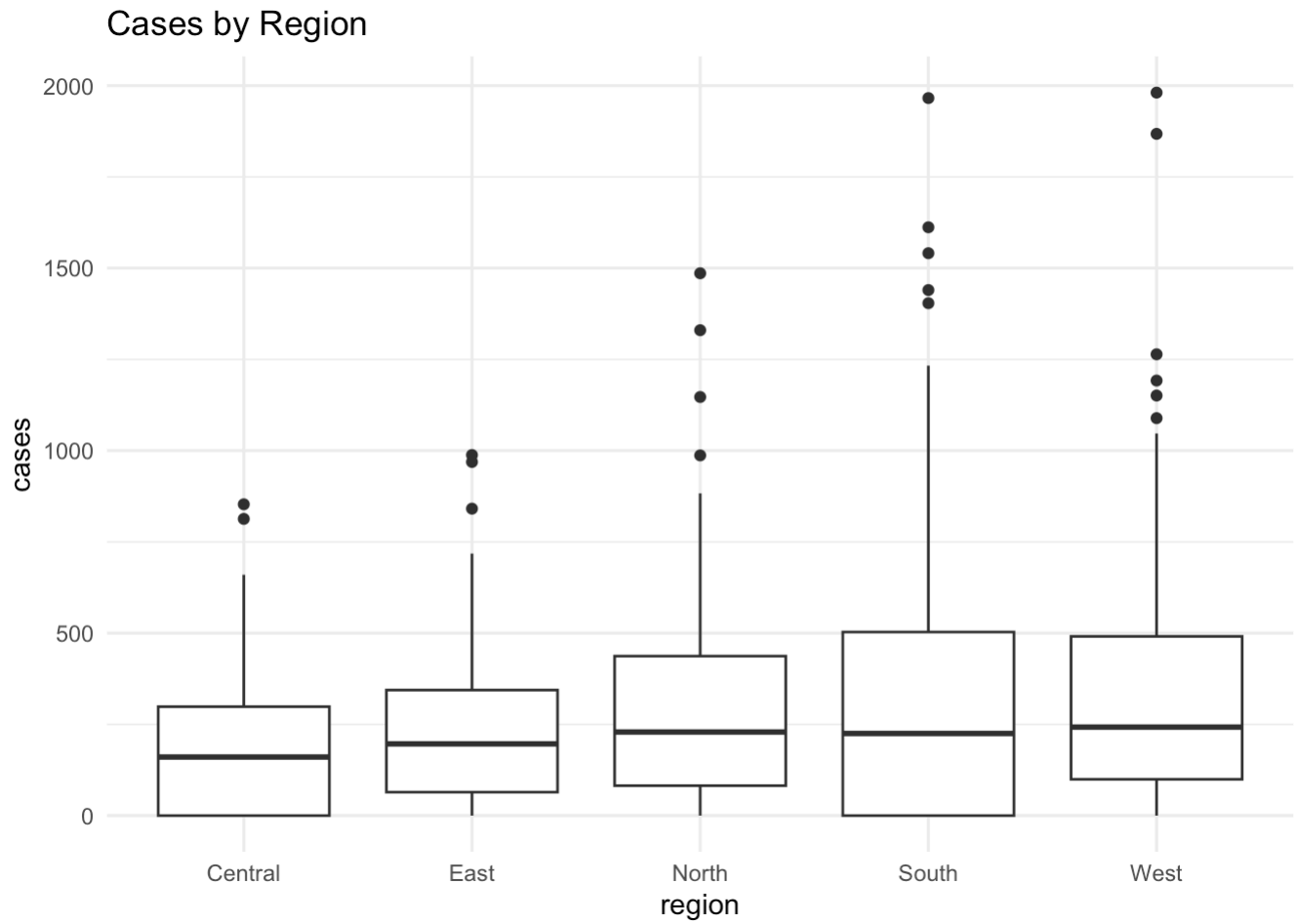
```
# Visualizing the relationship between 'cases' and 'population'
ggplot(data, aes(x = population, y = cases)) +
  geom_point() +
  labs(title = "Relationship Between Cases and Population", x = "population", y = "ca
ses") +
  theme_minimal()
```



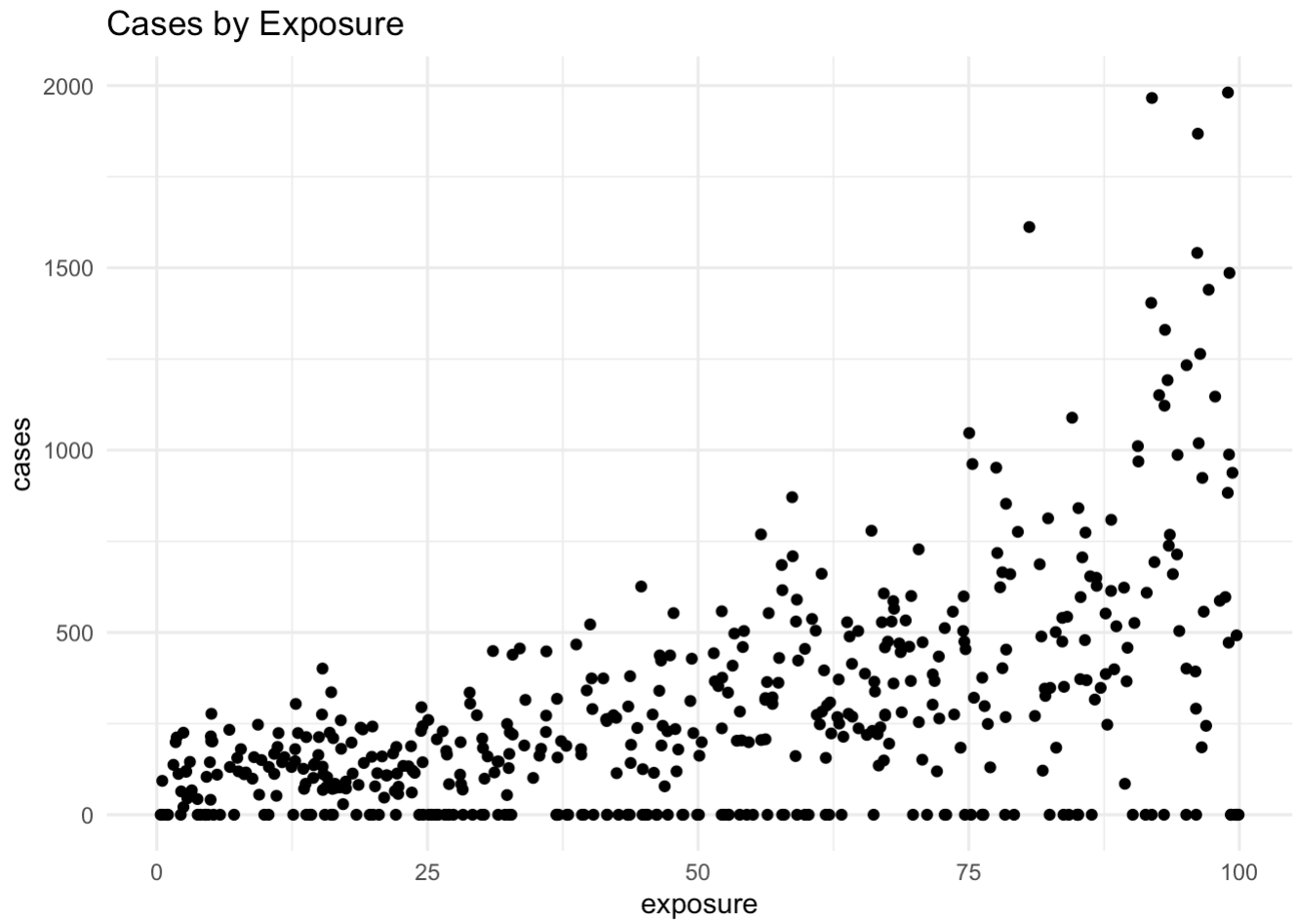

```
# Visualizing the relationship between 'cases' and 'time'
ggplot(data, aes(x = time, y = cases)) +
  geom_point() +
  labs(title = "Relationship Between Cases and Time", x = "time", y = "cases") +
  theme_minimal()
```




```
# Visualizing the relationship between 'cases' and 'region' using boxplot
ggplot(data, aes(x = region, y = cases)) +
  geom_boxplot() +
  labs(title = "Cases by Region", x = "region", y = "cases") +
  theme_minimal()
```




```
# Visualizing the relationship between 'cases' and 'exposure'
ggplot(data, aes(x = exposure, y = cases)) +
  geom_point() +
  labs(title = "Cases by Exposure", x = "exposure", y = "cases") +
  theme_minimal()
```



Interpretation

1. Relationship Between Cases and Population (Scatter Plot):

- The scatter plot between population and the number of cases shows a weak positive relationship.Most points are clustered at the lower end of the population size, with a few data points showing a large number of cases, even in areas with a smaller population.
- This indicates that while larger populations tend to have more cases, the relationship is not very strong. It's possible that factors other than population size are affecting the case numbers.

2. Relationship Between Cases and Time (Scatter Plot):

- The scatter plot between cases and time shows a weak positive relationship, with some data points showing significant spikes in cases at specific time points.
- There is no clear trend indicating that cases increase significantly over time, though the outliers suggest that specific time points saw large outbreaks of cases.

3. Cases by Region (Boxplot):

- The boxplot shows that the regions vary considerably in the number of cases.
- Some regions (such as the South and West) show higher variation and outliers, while others (like East and North) have relatively lower numbers of cases.

- This suggests that cases are more concentrated in certain regions, with some regions having a significantly higher number of cases than others.

4.Cases by Exposure:

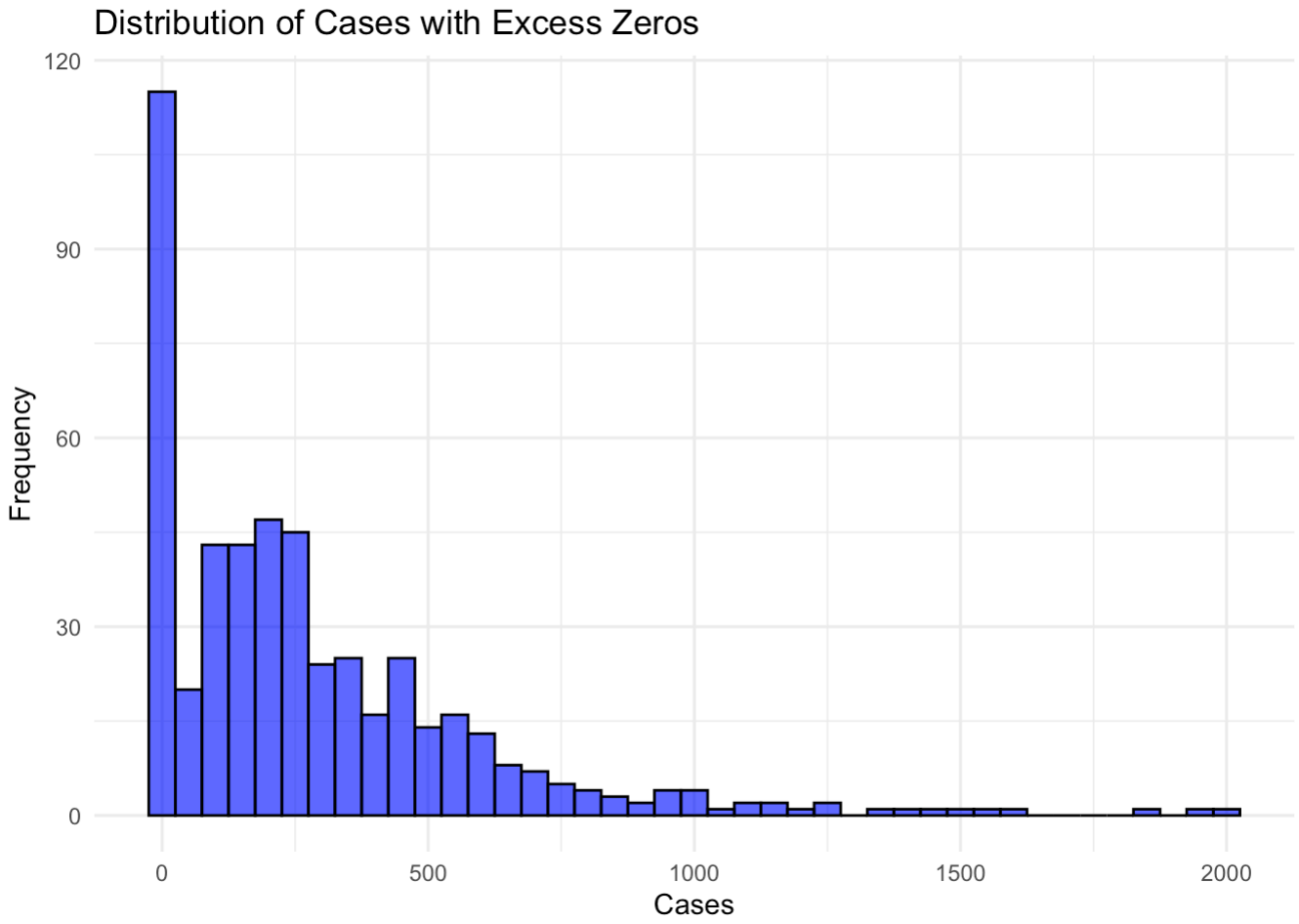
- The scatter plot suggests that exposure is positively related to the number of cases, with a sharp increase in the number of cases as exposure increases. However, the plot also shows a high degree of variability.

c. Comment on the presence of excess zeros in cases

```
# Check for excess zeros in the 'cases' column
zero_count <- sum(data$cases == 0, na.rm = TRUE)
total_count <- nrow(data)

zero_proportion <- zero_count / total_count

# Visualizing the distribution of 'cases'
ggplot(data, aes(x = cases)) +
  geom_histogram(binwidth = 50, fill = "blue", color = "black", alpha = 0.7) +
  labs(title = "Distribution of Cases with Excess Zeros", x = "Cases", y = "Frequenc
y") +
  theme_minimal()
```




```
zero_proportion
```

```
## [1] 0.228
```

Interpretation

- The histogram highlighting excess zeros shows that around 22.8% of the observations are zero cases, which is a significant proportion. This suggests that the cases data may require a model that accounts for excess zeros, like zero-inflated models.

d. Is there overdispersion in the cases variable?

```
# Calculate the mean and variance of the 'cases' variable
mean_cases <- mean(data$cases, na.rm = TRUE)
var_cases <- var(data$cases, na.rm = TRUE)

# Check for overdispersion
overdispersion_ratio <- var_cases / mean_cases

overdispersion_ratio
```

```
## [1] 348.7512
```

Interpretation

- The overdispersion ratio of 348.75 indicates that there is overdispersion in the cases variable. This suggests that the variance is much higher than the mean, which is common in count data that is highly variable.

2. Poisson Regression

a. Fit Poisson regression model glm(data=data, cases ~ time + region + exposure + offset(log(population)), family = poisson)

```
glm(data = data, cases ~ time + region + exposure + offset(log(population)), family =
poisson)
```

```
##
## Call:  glm(formula = cases ~ time + region + exposure + offset(log(population)),
##     family = poisson, data = data)
##
## Coefficients:
## (Intercept)         time   regionEast  regionNorth  regionSouth   regionWest
##   -8.914255     0.108080    -0.002018     0.387658     0.395296     0.355812
##    exposure
##    0.021488
##
## Degrees of Freedom: 499 Total (i.e. Null);  493 Residual
## Null Deviance:       150400
## Residual Deviance: 76430     AIC: 79320
```

i. Explain the term offset(log(population)) and describe its purpose in the model.

- The offset term in a Poisson regression, using log(population), adjusts for varying population sizes when modeling cases. It ensures that the number of cases is compared relative to the population, providing a “rate” of cases per unit of population. This adjustment allows for a fair comparison between regions or time periods by accounting for the population size’s influence on the number of cases.

ii. Exponentiate coefficients to report incidence rate ratios (IRRs) and explain the effect of each predictor on the number of cases.

```
model <- glm(cases ~ time + region + exposure + offset(log(population)), data = data,
family = poisson)
summary(model)
```

```
##
## Call:
## glm(formula = cases ~ time + region + exposure + offset(log(population)),
##     family = poisson, data = data)
##
## Coefficients:
##               Estimate Std. Error  z value Pr(>|z|)
## (Intercept) -8.9142552  0.0108631 -820.600   <2e-16 ***
## time         0.1080796  0.0010423  103.692   <2e-16 ***
## regionEast  -0.0020176  0.0095066   -0.212    0.832
## regionNorth  0.3876584  0.0091035   42.584   <2e-16 ***
## regionSouth  0.3952964  0.0089553   44.141   <2e-16 ***
## regionWest   0.3558117  0.0089461   39.773   <2e-16 ***
## exposure     0.0214877  0.0001004  213.951   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 150384  on 499  degrees of freedom
## Residual deviance:  76430  on 493  degrees of freedom
## AIC: 79323
##
## Number of Fisher Scoring iterations: 5
```

```
# Exponentiate the coefficients to get the IRRs
exp(coef(model))
```

```
##  (Intercept)         time   regionEast  regionNorth  regionSouth   regionWest
## 0.0001344585 1.1141364269 0.9979844771 1.4735263373 1.4848241682 1.4273387119
##     exposure
## 1.0217202055
```

interpretation

Time:

- The IRR of 1.1141 suggests that for each unit increase in time, the rate of cases increases by approximately 11.4%, holding other factors constant.

RegionEast:

- The IRR of 0.998 suggests that being in the East region results in a slight decrease in the rate of cases compared to the reference region (holding other variables constant). The change is very minimal and not statistically significant.

RegionNorth:

- The IRR of 1.4735 indicates that being in the North region increases the rate of cases by 47.35% compared to the reference region, holding other variables constant.

RegionSouth:

- The IRR of 1.4848 shows that being in the South region increases the rate of cases by 48.48%, compared to the reference region, after adjusting for other factors.

RegionWest:

- The IRR of 1.4273 suggests that being in the West region increases the rate of cases by 42.73% compared to the reference region, adjusting for other factors.

Exposure:

- The IRR of 1.0217 means that for each one-unit increase in exposure, the rate of cases increases by approximately 2.17%, holding other factors constant.

b. Assess the model fit

i. Test goodness of fit with a chi-squared test.

```
anova(model, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: cases
##
## Terms added sequentially (first to last)
##
##
##          Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                       499     150384
## time      1    15231       498     135153 < 2.2e-16 ***
## region    4     7752       494     127400 < 2.2e-16 ***
## exposure  1    50970       493      76430 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

interpretation

- The model has a strong fit to the data as the p-values for each term are extremely low, indicating that the model components (time, region, and exposure) significantly contribute to explaining the variance in cases.

ii. Check for overdispersion.

```
# Calculate the residual deviance and the degrees of freedom
deviance <- model$deviance
df <- model$df.residual

# Calculate the overdispersion ratio
overdispersion_ratio <- deviance / df
overdispersion_ratio
```

```
## [1] 155.0311
```

interpretation

- since the overdispersion ratio is significantly larger than 1, it indicates overdispersion in the cases variable.

3. Zero-inflated Poisson (ZIP) model

a. Explain why ZIP model is necessary for cases data.

- The Zero-Inflated Poisson (ZIP) model is necessary for cases data when there is a large number of zeros in the data that cannot be explained by the standard Poisson regression model. This happens when:
- Excess zeros: The data includes more zeros than what would be expected under a Poisson distribution, which assumes that the data only follows a count distribution. For example, in epidemiological data like cases, there may be locations or time periods where no cases were observed, but this does not mean that these locations or times should follow a Poisson process. Instead, there may be a separate process responsible for the occurrence of these excess zeros.
- Overdispersion: The variance in the data might be larger than the mean, which is another indicator that the data doesn't follow a standard Poisson distribution. In such cases, the ZIP model helps model the excess zeros and overdispersion separately, treating them as coming from different processes.

b. Fit a ZIP model using pscl package: Logistic regression predicting excess zeros (use exposure and region as predictors).

```
# Load necessary library
library(pscl)
```

```
## Classes and Methods for R originally developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University (2002-2015),
## by and under the direction of Simon Jackman.
## hurdle and zeroinfl functions by Achim Zeileis.
```

```
# Fit a ZIP model using exposure and region as predictors for excess zeros
zip_model <- zeroinfl(cases ~ time + region + exposure | region,
                      data = data,
                      dist = "poisson")

# View model summary
summary(zip_model)
```

```
##
## Call:
## zeroinfl(formula = cases ~ time + region + exposure | region, data = data,
##     dist = "poisson")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -2.1411 -1.0589  0.2289  0.9539  2.8014
##
## Count model coefficients (poisson with log link):
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.9622535  0.0109693 361.212   <2e-16 ***
## time         0.1044541  0.0010282 101.587   <2e-16 ***
## regionEast  -0.0795344  0.0095057  -8.367   <2e-16 ***
## regionNorth  0.1339762  0.0091086  14.709   <2e-16 ***
## regionSouth  0.4093358  0.0089039  45.973   <2e-16 ***
## regionWest   0.2962574  0.0088752  33.380   <2e-16 ***
## exposure     0.0197791  0.0001008 196.138   <2e-16 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.8954     0.2204  -4.063 4.85e-05 ***
## regionEast   -0.4444     0.3255  -1.365   0.1721
## regionNorth  -0.6331     0.3408  -1.857   0.0632 .
## regionSouth  -0.1342     0.3207  -0.419   0.6755
## regionWest   -0.4656     0.3337  -1.395   0.1630
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 12
## Log-likelihood: -1.107e+04 on 12 Df
```

c. Interpret the coefficients for zero-inflation components.

interpretation

Intercept: -0.8954

- The negative sign suggests that, when all other predictors are held constant, the likelihood of a zero case occurring is lower compared to the baseline (which means non-zero cases are more likely to occur).

RegionEast: -0.4445

- The negative coefficient indicates that, compared to the baseline region (likely the reference region), the East region is less likely to have a zero count, meaning that there are fewer excess zeros in the East region.

RegionNorth: -0.6331

- A negative coefficient suggests that, compared to the reference region, the North region has a significantly lower probability of zero counts. The further from zero the coefficient is (in the negative direction), the lower the likelihood of observing a zero in that region.

RegionSouth: -0.1342

- The coefficient is small and negative, which suggests a slight reduction in the probability of zero cases in the South region compared to the reference region. This effect, however, is less pronounced.

RegionWest: -0.4656

- Similar to the other regions, this negative coefficient indicates a lower likelihood of excess zeros in the West region compared to the reference region. The magnitude of the coefficient shows that the reduction is somewhat significant but not as large as in the North region.

d. Compare the ZIP model to the Poisson model in 2a. above using AIC and a Vuong test.

Poisson model

```
poisson_model <- glm(cases ~ time + region + exposure + offset(log(population)),
                     family = poisson, data = data)

# AIC comparison
AIC(poisson_model, zip_model)
```

```
##               df      AIC
## poisson_model  7 79322.74
## zip_model     12 22160.73
```

```
# Vuong test
vuong_test <- vuong(poisson_model, zip_model)
```

```
## NA or numerical zeros or ones encountered in fitted probabilities
## dropping these 5 cases, but proceed with caution
## Vuong Non-Nested Hypothesis Test-Statistic:
## (test-statistic is asymptotically distributed N(0,1) under the
##  null that the models are indistinguishible)
## -------------------------------------------------------------
##               Vuong z-statistic             H_A    p-value
## Raw                  -8.741111 model2 > model1 < 2.22e-16
## AIC-corrected        -8.739159 model2 > model1 < 2.22e-16
## BIC-corrected        -8.735055 model2 > model1 < 2.22e-16
```

interpretation

1. AIC Comparison:

- The ZIP model has a significantly lower AIC than the Poisson model, which suggests that the ZIP model is a better fit for the data. This result indicates that the ZIP model accounts for both the count of cases and the excess zeros more effectively than the Poisson model.

2. Vuong Test:

- The p-value is very small, which indicates that the ZIP model fits the data significantly better than the Poisson model. The negative z-statistic suggests that the ZIP model is the preferred model for explaining the data, particularly due to its ability to handle excess zeros.

conclusion

- The ZIP model is the better model as indicated by both the lower AIC and the significant Vuong test p-value. It effectively handles both the count data and excess zeros, making it a more appropriate choice than the Poisson model for this data.

4. Negative Binomial Model

a. Fit a negative binomial regression model using glm.nb() from the MASS package to address overdispersion (use offset(log(population)).Interpret the results.

```
library(MASS)
```

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

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

```
# Fit the negative binomial model
nb_model <- glm.nb(cases ~ time + region + exposure + offset(log(population)), data =
data)
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
## Warning in sqrt(1/i): NaNs produced
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
## Warning in sqrt(1/i): NaNs produced
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
## Warning in sqrt(1/i): NaNs produced
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
## Warning in sqrt(1/i): NaNs produced
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
## Warning in sqrt(1/i): NaNs produced
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
## Warning in sqrt(1/i): NaNs produced
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
## Warning in sqrt(1/i): NaNs produced
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
## Warning in sqrt(1/i): NaNs produced
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
## Warning in sqrt(1/i): NaNs produced
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
## Warning in sqrt(1/i): NaNs produced
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
## Warning in sqrt(1/i): NaNs produced
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
## Warning in sqrt(1/i): NaNs produced
```

```
## Warning in glm.nb(cases ~ time + region + exposure + offset(log(population)), :
## alternation limit reached
```

```
# Display the results
summary(nb_model)
```

```
##
## Call:
## glm.nb(formula = cases ~ time + region + exposure + offset(log(population)),
##     data = data, init.theta = 0.3861435578, link = log)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.724744   0.245137 -35.591  < 2e-16 ***
## time         0.100681   0.027652   3.641 0.000272 ***
## regionEast  -0.051131   0.224861  -0.227 0.820121
## regionNorth  0.462639   0.227709   2.032 0.042183 *
## regionSouth  0.466490   0.231361   2.016 0.043770 *
## regionWest   0.474011   0.229696   2.064 0.039052 *
## exposure     0.022299   0.002485   8.973  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.3861) family taken to be 1)
##
##     Null deviance: 694.8  on 499  degrees of freedom
## Residual deviance: 586.3  on 493  degrees of freedom
## AIC: 6165.1
##
## Number of Fisher Scoring iterations: 4
##
##
##               Theta:  0.3861
##           Std. Err.:  0.0243
## Warning while fitting theta: alternation limit reached
##
##  2 x log-likelihood:  -6149.0720
```

interpretation

Intercept:

- Estimate = -8.724744, Standard Error = 0.245137, Z value = -35.591, p-value < 2e-16.
- This is the baseline value of the number of cases when all predictors (time, region, exposure) are at zero. The negative value indicates a lower baseline number of cases.

Time:

- Estimate = 0.100681, Standard Error = 0.027652, Z value = 3.641, p-value = 0.000272.
- The positive coefficient indicates that as time increases, the number of cases increases. This is statistically significant ($p < 0.001$).

RegionEast:

- Estimate = -0.051131, Standard Error = 0.224861, Z value = -0.227, p-value = 0.820121.
- The coefficient is not statistically significant ($p > 0.05$), meaning there is no significant effect of being in the East region on the number of cases compared to the reference region.

RegionNorth:

- Estimate = 0.462639, Standard Error = 0.227709, Z value = 2.032, p-value = 0.042183.
- The positive coefficient suggests that the number of cases in the North region is higher compared to the reference region, and it is statistically significant ($p < 0.05$).

RegionSouth:

- Estimate = 0.466490, Standard Error = 0.231361, Z value = 2.016, p-value = 0.043770.
- Similar to the North region, the positive coefficient indicates more cases in the South region compared to the reference region, and this is statistically significant ($p < 0.05$).

RegionWest:

- Estimate = 0.474011, Standard Error = 0.229696, Z value = 2.064, p-value = 0.039052.
- The positive coefficient for the West region shows that it has more cases compared to the reference region, and it is statistically significant ($p < 0.05$).

Exposure:

- Estimate = 0.022299, Standard Error = 0.002485, Z value = 8.973, p-value < 2e-16.
- The positive and highly significant coefficient suggests that as exposure increases, the number of cases also increases. This relationship is very strong.

Dispersion Parameter (Theta):

- Theta = 0.3861 with Standard Error = 0.0243.
- Theta represents the dispersion parameter of the negative binomial model. A value close to 1 suggests that the data may still have some degree of overdispersion, but it is not as extreme as in the Poisson model. Since theta is far from 0, it indicates that the negative binomial model is appropriate for handling overdispersion, where the variance exceeds the mean.

b. Compare it to the Poisson and ZIP models:

i. Use AIC and BIC to evaluate model fit.

```
# Compare AIC and BIC for the three models
AIC(poisson_model, zip_model, nb_model)
```

```
##               df       AIC
## poisson_model  7 79322.738
## zip_model     12 22160.731
## nb_model       8  6165.072
```

```
BIC(poisson_model, zip_model, nb_model)
```

```
##                df       BIC
## poisson_model   7 79352.240
## zip_model      12 22211.307
## nb_model        8  6198.789
```

interpretation

- Negative Binomial model has the lowest AIC (6165.072) and BIC (6198.789) compared to both the Poisson model (AIC = 79322.738, BIC = 79352.240) and ZIP model (AIC = 22160.731, BIC = 22211.307).
- This suggests that the Negative Binomial model provides the best fit to the data among the three models.

ii. Discuss how the negative binomial handles overdispersion versus the ZIP's focus on excess zeros.

Negative Binomial Model:

- The negative binomial model is useful when the data shows overdispersion (variance is greater than the mean), which is common in many real-world count datasets. It introduces an additional parameter to account for the extra variability, making it more flexible than the Poisson model.
- It assumes the data follows a Poisson distribution with an additional gamma-distributed random effect to account for the overdispersion.

ZIP Model:

- The ZIP (Zero-Inflated Poisson) model is designed for count data with excess zeros. It combines a Poisson count model with a logistic regression model to account for the extra zeros in the data.
- The ZIP model assumes that some of the zeros come from a structural zero process (where no events can occur) and the rest of the zeros follow a Poisson distribution.

Key Difference:

- The negative binomial model is focused on handling overdispersion, whereas the ZIP model is designed to handle excess zeros by incorporating both a Poisson count model and a logistic regression for zero inflation.

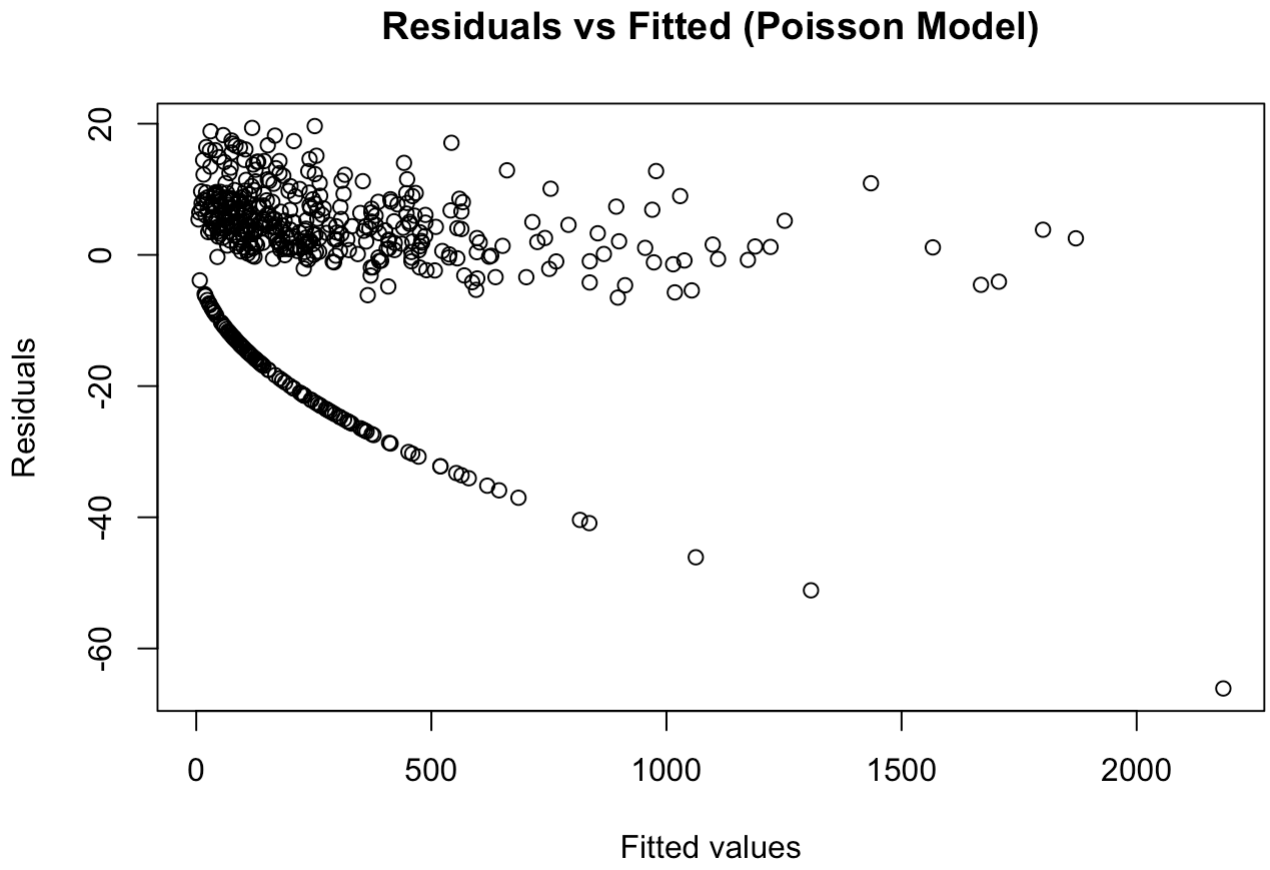
5. Diagnostics

For both Poisson and ZIP models:

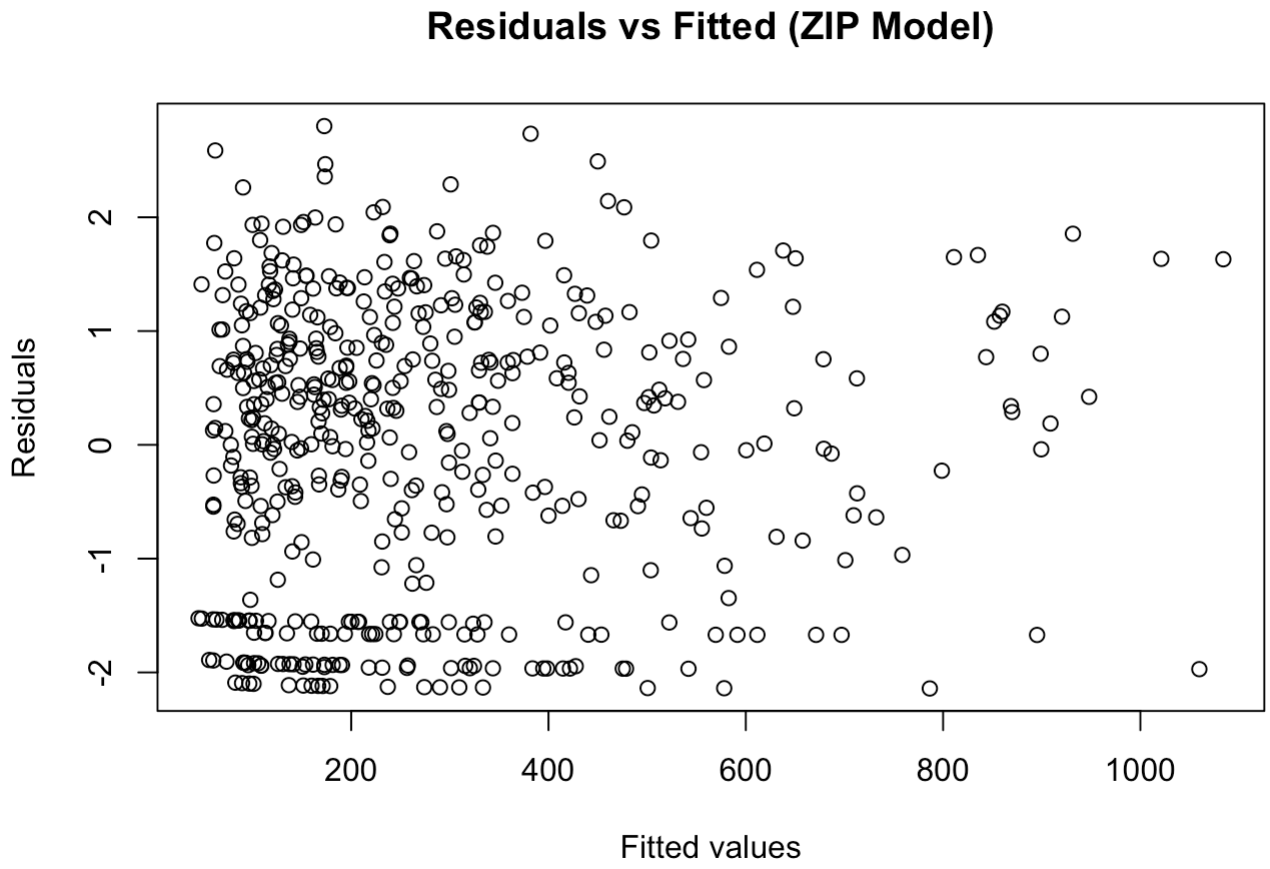
a. Plot residuals (e.g., Pearson or deviance

residuals) against fitted values and predictors.

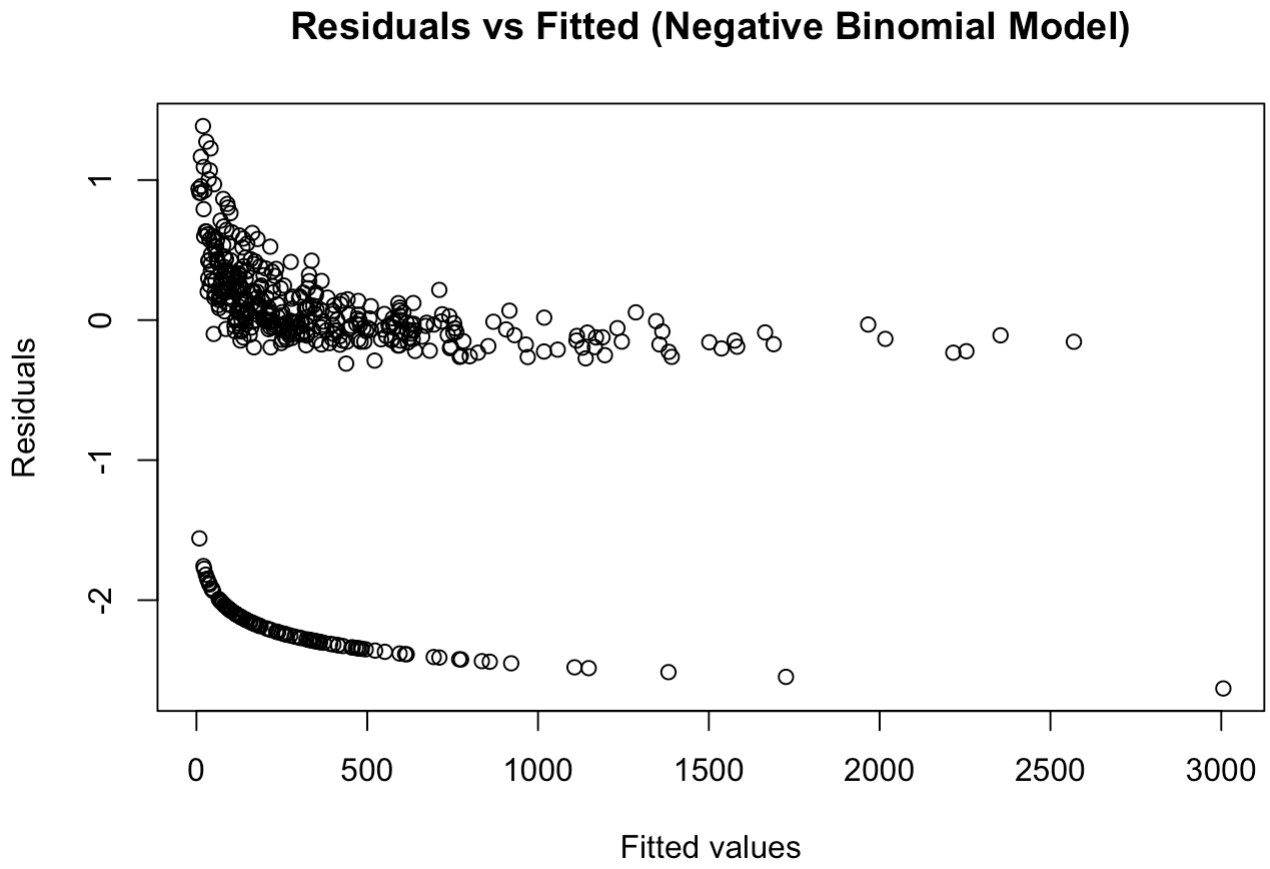
```
# Residuals vs Fitted for Poisson Model
plot(fitted(poisson_model), residuals(poisson_model),
     xlab = "Fitted values", ylab = "Residuals",
     main = "Residuals vs Fitted (Poisson Model)")
```




```
# Residuals vs Fitted for ZIP Model
plot(fitted(zip_model), residuals(zip_model),
     xlab = "Fitted values", ylab = "Residuals",
     main = "Residuals vs Fitted (ZIP Model)")
```




```
# Residuals vs Fitted for Negative Binomial Model
plot(fitted(nb_model), residuals(nb_model),
     xlab = "Fitted values", ylab = "Residuals",
     main = "Residuals vs Fitted (Negative Binomial Model)")
```



b. Identify patterns suggesting misspecification (e.g., poor handling of zeros in Poisson).

Poisson Model:

- Residuals vs Fitted Plot: The plot shows a clear curvature, with residuals increasing as the fitted values increase, particularly for the higher values.
- This indicates a poor fit for overdispersed data and inadequate handling of the zeros. In a Poisson model, overdispersion occurs when the variance exceeds the mean, leading to inflated residuals for high count values and showing that the model is not well-suited to handle the variation in the data.
- Zero-inflation issue: There are large numbers of residuals that are far from zero, particularly in the lower fitted values, suggesting poor handling of zeros in the Poisson model, which doesn't account for the excess zeros effectively.

ZIP Model:

- Residuals vs Fitted Plot: The ZIP model plot looks more evenly spread out compared to the Poisson model. Residuals are generally distributed around zero across all fitted values, with fewer large residuals.
- This suggests that the ZIP model handles the excess zeros better as the residuals behave more randomly and symmetrically, without any strong patterns or skew. However, there is still some dispersion seen in the data.

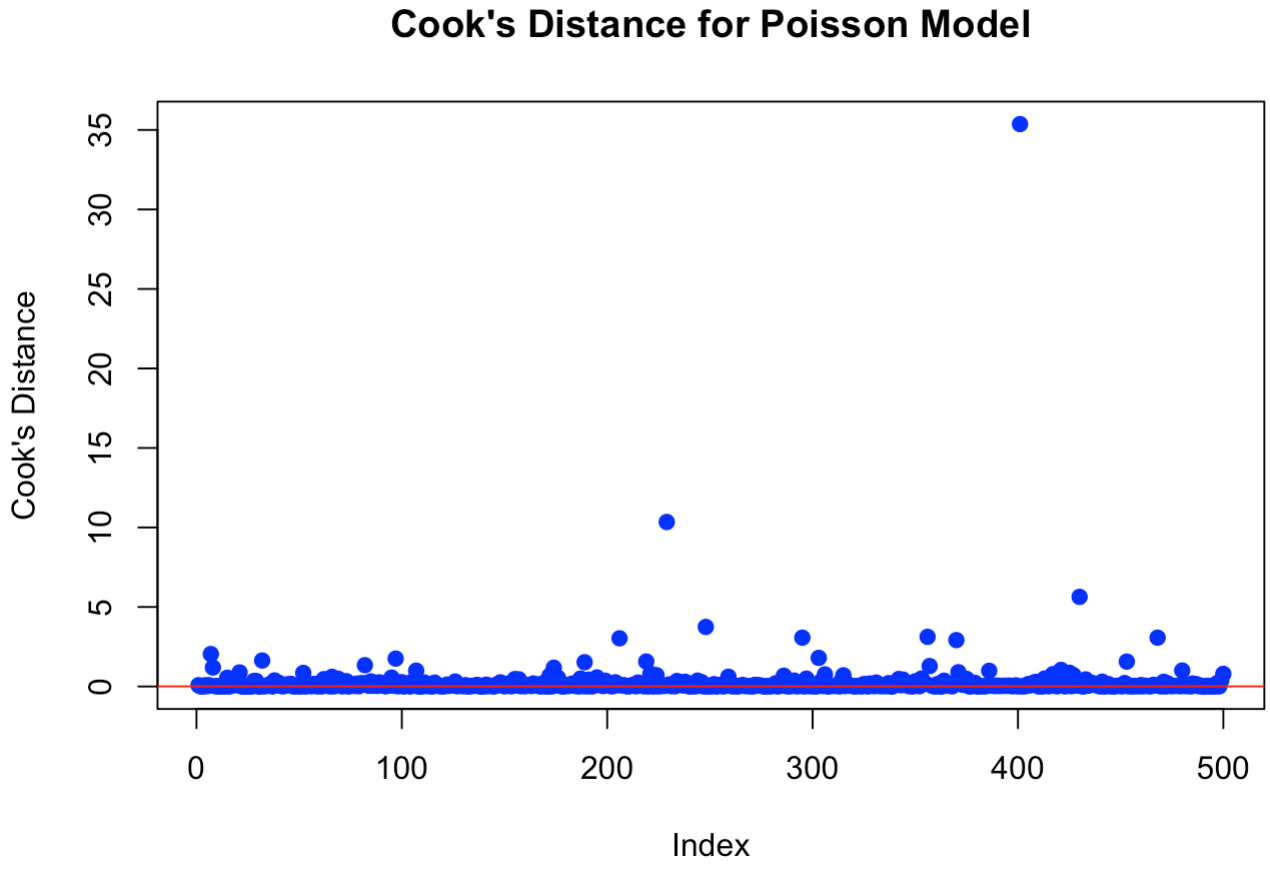
Negative Binomial Model:

- Residuals vs Fitted Plot: The residuals are more spread around zero, but there is still some curvature at higher fitted values, though not as pronounced as in the Poisson model.
- This suggests that the Negative Binomial model is a better fit for overdispersed data than the Poisson model, but it still doesn't perfectly handle all the nuances of the data, especially when it comes to very high count values.

c. Check for influential observations using Cook's distance or leverage.

```
# For Poisson model, calculate Cook's distance
cooks_dist_poisson <- cooks.distance(poisson_model)

# Plot Cook's Distance for Poisson model
plot(cooks_dist_poisson,
     main = "Cook's Distance for Poisson Model",
     ylab = "Cook's Distance",
     xlab = "Index",
     pch = 19, col = "blue")
abline(h = 4 / length(cooks_dist_poisson), col = "red")  # Threshold line for Cook's
Distance
```



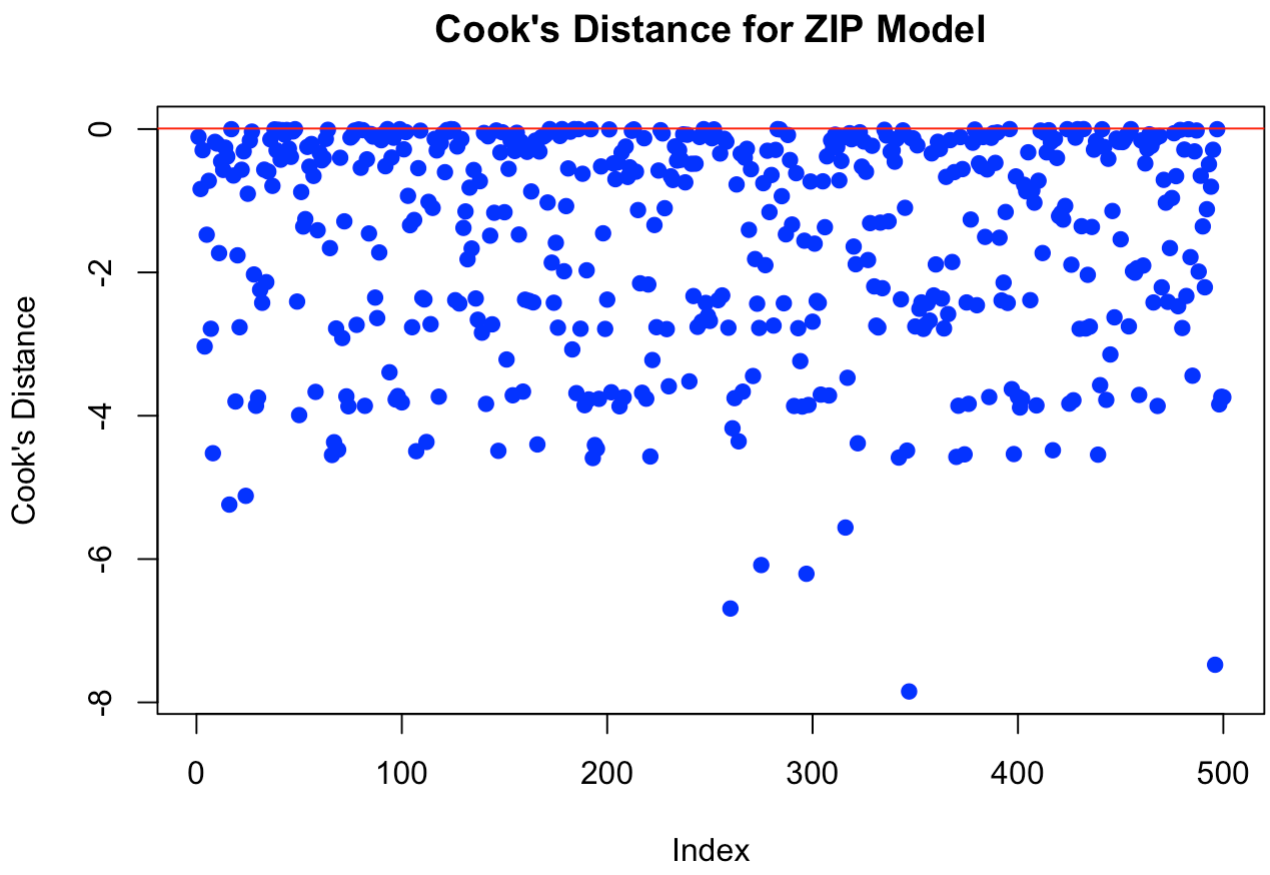
```
# ********************************************************************************
*****

# Obtain fitted values and residuals
fitted_values <- predict(zip_model, type = "response")
residuals_zip <- residuals(zip_model, type = "pearson")

# Compute leverages (using a pseudo leverage computation)
# Leveraging the fitted values and residuals
n <- length(fitted_values)  # Number of observations
hii <- fitted_values * (1 - fitted_values)  # Approximate leverage values

# Calculate Cook's Distance for each observation
cooks_dist_zip <- (residuals_zip^2 * hii) / (1 - hii)

# Plot Cook's Distance
plot(cooks_dist_zip,
     main = "Cook's Distance for ZIP Model",
     ylab = "Cook's Distance",
     xlab = "Index",
     pch = 19, col = "blue")
abline(h = 4 / n, col = "red")  # Threshold line for Cook's Distance
```



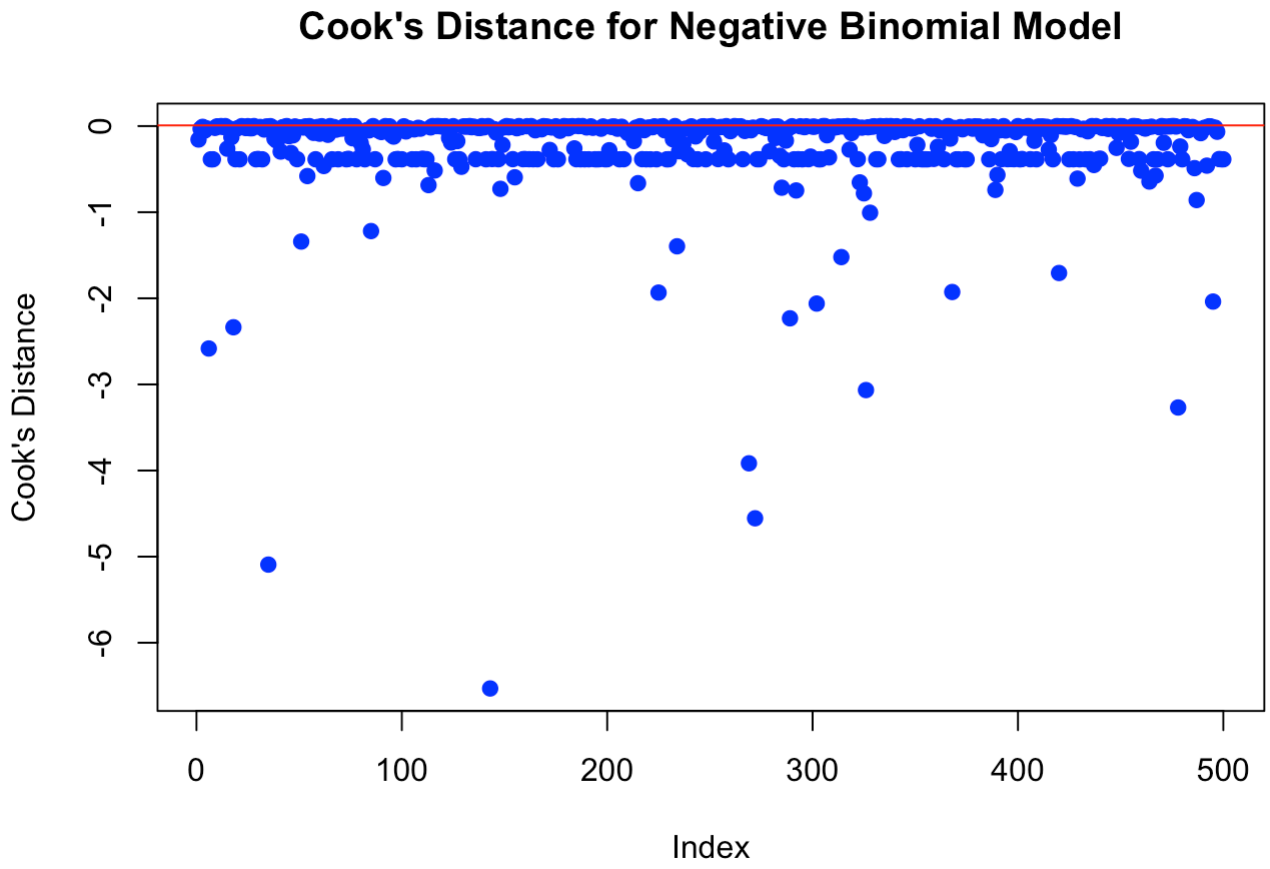
```
# ********************************************************************

# Obtain fitted values and residuals
fitted_values_nb <- fitted(nb_model)
residuals_nb <- residuals(nb_model, type = "pearson")

# Compute leverage (hii) values
hii_nb <- fitted_values_nb * (1 - fitted_values_nb)  # Approximate leverage values fo
r negative binomial

# Calculate Cook's Distance for each observation
n_nb <- length(fitted_values_nb)  # Number of observations
cooks_dist_nb <- (residuals_nb^2 * hii_nb) / (1 - hii_nb)

# Plot Cook's Distance
plot(cooks_dist_nb,
     main = "Cook's Distance for Negative Binomial Model",
     ylab = "Cook's Distance",
     xlab = "Index",
     pch = 19, col = "blue")
abline(h = 4 / n_nb, col = "red")  # Threshold line for Cook's Distance
```



d. If issues are detected, propose model modifications.

Issues detected:

- Influential observations: In all models, there are several influential observations based on the Cook's Distance threshold. These observations might be driving the results.
- Potential misspecification: If the influential points are concentrated in specific regions or time periods, this might indicate that the model is missing some important predictors or interactions.

Proposed model modifications:

- Removing Influential Observations: Investigate these influential points further and assess whether they should be removed from the dataset.
- Including Interaction Terms: Consider adding interaction terms (e.g., between exposure and region or time) to capture potential complex relationships.
- Use Robust Standard Errors: When influential points remain in the model, using robust standard errors can help reduce the impact of these outliers.
- Recheck Model Assumptions: Double-check whether the assumptions (such as the distribution of the residuals) hold. If the assumptions are violated, switching to a more flexible model (e.g., a Generalized Additive Model) might be necessary.