SDS 4392 HW3

Washington University, SP 2025

Aidan Kardan SDS 4392 HW 3

1.[1 pt] In this example we will practice generating data according to the assuming of Poisson distribution with some covariates:

a) Create a simulated data frame named `df0` with 100 rows and 4 columns that fits the following criteria:

- has 2 numerical variables, each randomly generated with range [0,1.5] and [-2,2] respectively, let's call them `x1` and `x2`
- has 1 categorical / factor variables, each with levels `yes`,`no` respectively, let's call it `x3`
- has a variable named `y` that follows the Poisson distribution with the following mean: $g(\mu) = 2 + 0.7x_1 - 1.3x_2$, where $g(\cdot)$ is the log link function.

```r
set.seed(123)
# Define the number of observations
n <- 100

# Generate the covariates:
# x1: Uniformly distributed in [0, 1.5]
x1 <- runif(n, min = 0, max = 1.5)

# x2: Uniformly distributed in [-2, 2]
x2 <- runif(n, min = -2, max = 2)

# x3: A categorical factor with levels "yes" and "no"
x3 <- factor(sample(c("yes", "no"), n, replace = TRUE))


mu <- exp(2 + 0.7*x1 - 1.3*x2)

# Generate the response variable y from the Poisson distribution
y <- rpois(n, lambda = mu)

# Combine into a data frame
df0 <- data.frame(x1 = x1, x2 = x2, x3 = x3, y = y)

head(df0)
```

```
##           x1          x2  x3  y
## 1 0.43136628  0.39995584  no  8
## 2 1.18245770 -0.66870584  no 25
## 3 0.61346538 -0.04554787  no 13
## 4 1.32452611  1.81789531 yes  0
## 5 1.41070093 -0.06839041  no 24
## 6 0.06833475  1.56140089  no  1
```

b) Then fit a generalized linear model with Poisson family for all the variables and comment on the model effects for `x1`, `x2` and `x3` as well as goodness-of-fit. Fit another model by dropping the non-significant variable, comment on the difference.

```
model_full <- glm(y ~ x1 + x2 + x3, data = df0, family = poisson(link = "log"))
summary(model_full)
```

```
## 
## Call:
## glm(formula = y ~ x1 + x2 + x3, family = poisson(link = "log"), 
##     data = df0)
## 
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)    
## (Intercept)  1.954627   0.063992  30.545   <2e-16 ***
## x1           0.747518   0.054346  13.755   <2e-16 ***
## x2          -1.291496   0.028889 -44.706   <2e-16 ***
## x3yes        0.005141   0.041186   0.125    0.901    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for poisson family taken to be 1)
## 
##     Null deviance: 3385.571  on 99  degrees of freedom
## Residual deviance:   94.229  on 96  degrees of freedom
## AIC: 520.68
## 
## Number of Fisher Scoring iterations: 4
```

```
model_reduced <- glm(y ~ x1 + x2, data = df0, family = poisson(link = "log"))
summary(model_reduced)
```

```
## 
## Call:
## glm(formula = y ~ x1 + x2, family = poisson(link = "log"), data = df0)
## 
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept)  1.95744    0.05987   32.69   <2e-16 ***
## x1           0.74706    0.05422   13.78   <2e-16 ***
## x2          -1.29229    0.02819  -45.84   <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: 3385.571  on 99  degrees of freedom
## Residual deviance:   94.245  on 97  degrees of freedom
## AIC: 518.69
## 
## Number of Fisher Scoring iterations: 4
```

```
anova(model_full, model_reduced)
```

```
## Analysis of Deviance Table
## 
```

```
## Model 1: y ~ x1 + x2 + x3
## Model 2: y ~ x1 + x2
##   Resid. Df Resid. Dev Df  Deviance Pr(>Chi)
## 1        96     94.229
## 2        97     94.245 -1 -0.015588   0.9006
```

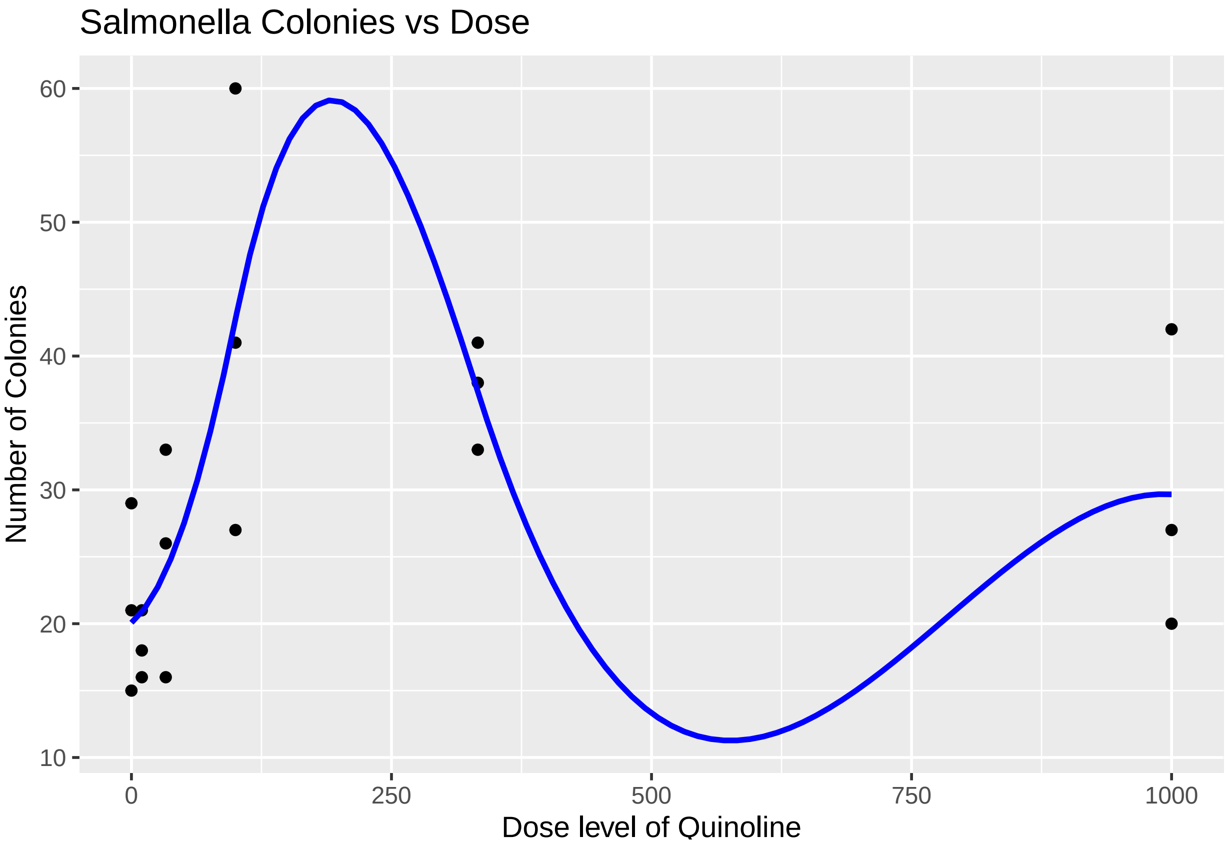
Based on the AIC values (lower AIC for the reduced model) and the anova test (high p value), it is clear that the second model is better, which is intuitive because it removes a variable that is random and unrelated to the response. However, the deviance suggests that the change in the two models is negligible.

2.[2pts] The salmonella data was collected in a salmonella reverse mutagenicity assay. The predictor is the dose level of quinoline and the response is the numbers of revertant colonies of TA98 salmonella observed on each of three replicate plates.

(a) Plot the data and comment on the relationship between dose and colonies.

```r
library(faraway)
data(salmonella)

library(ggplot2)

ggplot(salmonella, aes(x = dose, y = colonies)) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE, color = "blue") +
  labs(title = "Salmonella Colonies vs Dose",
       x = "Dose level of Quinoline",
       y = "Number of Colonies")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



The relationship seems to be clearly nonlinear. The number of colonies initially increases sharply with an increase in dose level up to around 200, then decreases significantly, showing a clear peak and subsequent decline. Therefore, a simple linear model will not suffice for this data.

(b) Compute the mean and variance within each set of observations with the same dose. Plot the variance against the mean and comment on what this says about overdispersion.

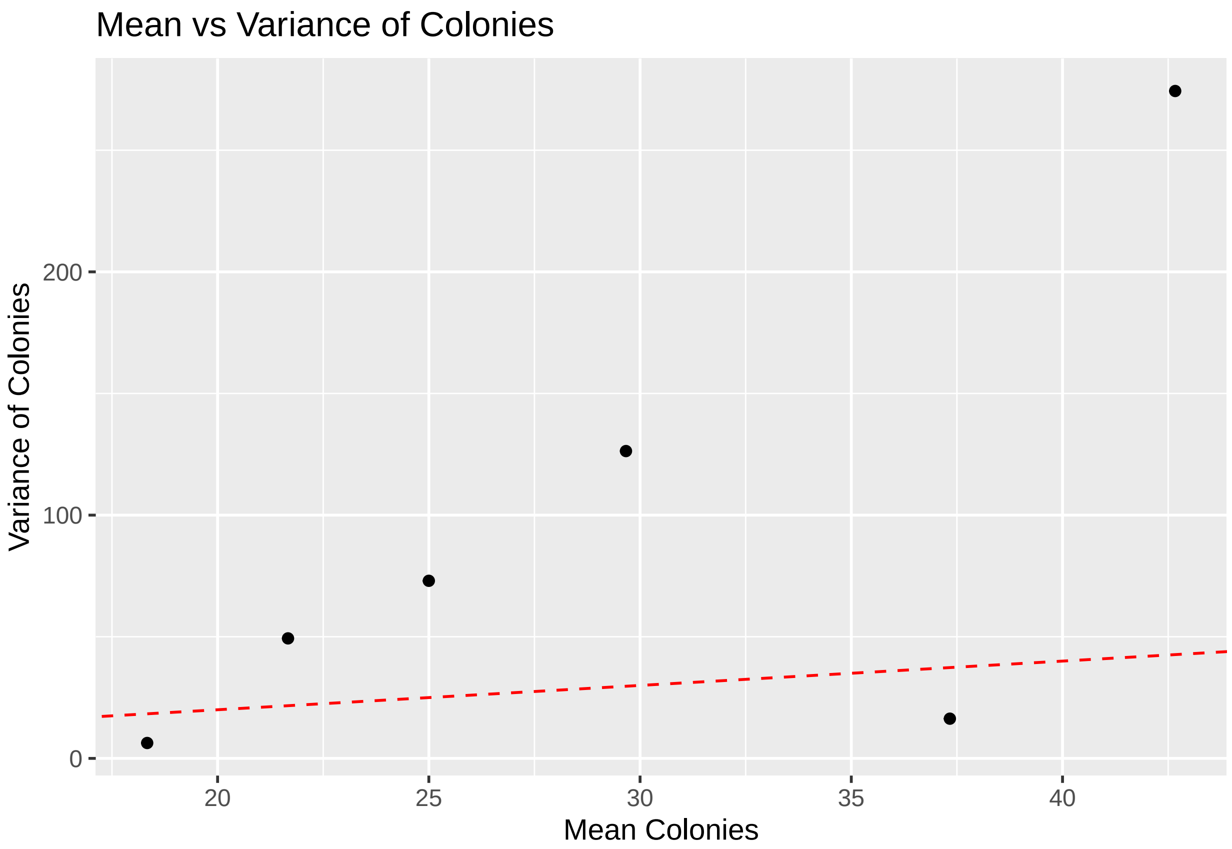
```
library(dplyr)
```

```
##
## 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
```

```
# Compute mean and variance within each dose
dose_stats <- salmonella %>%
  group_by(dose) %>%
  summarise(mean_colonies = mean(colonies),
            var_colonies = var(colonies))
dose_stats
```

```
## # A tibble: 6 x 3
##    dose mean_colonies var_colonies
##   <int>         <dbl>        <dbl>
## 1     0          21.7        49.3
## 2    10          18.3         6.33
## 3    33          25          73
## 4   100          42.7       274.
## 5   333          37.3        16.3
## 6  1000          29.7       126.
```

```
# Plot Variance vs. Mean
ggplot(dose_stats, aes(x = mean_colonies, y = var_colonies)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, col = "red", linetype = "dashed") +
  labs(title = "Mean vs Variance of Colonies",
       x = "Mean Colonies", y = "Variance of Colonies")
```



Most points lie clearly above the red line, which represents the equality of mean and variance (variance = mean, the assumption of a standard Poisson Model)

The observed variance substantially exceeds the mean for most dose groups, indicating clear evidence of over-dispersion.

Over-dispersion suggests that a simple Poisson Model may not be sufficient and a model accounting for dispersion, quasi-poisson or negative binomial would be better.

(c) Fit a model with dose treated as a six-level factor. Check the deviance to determine whether this model fits the data. Do you think it is possible to find a transformation of the dose predictor that results in a Poisson model that does fit the data?

```r
# Fit Poisson GLM with dose as categorical factor (6 levels)
model_dose_factor <- glm(colonies ~ factor(dose), data = salmonella, family = poisson(link = "log"))

# Summary of the model
summary(model_dose_factor)
```

```
##
## Call:
## glm(formula = colonies ~ factor(dose), family = poisson(link = "log"),
##     data = salmonella)
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)          3.0758     0.1240  24.798  < 2e-16 ***
## factor(dose)10      -0.1671     0.1832  -0.912 0.361869
## factor(dose)33       0.1431     0.1695   0.844 0.398427
## factor(dose)100    0.6776     0.1523   4.449 8.62e-06 ***
## factor(dose)333    0.5441     0.1559   3.490 0.000484 ***
## factor(dose)1000   0.3142     0.1632   1.926 0.054099 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 78.358  on 17  degrees of freedom
## Residual deviance: 33.496  on 12  degrees of freedom
## AIC: 138.03
##
## Number of Fisher Scoring iterations: 4
```

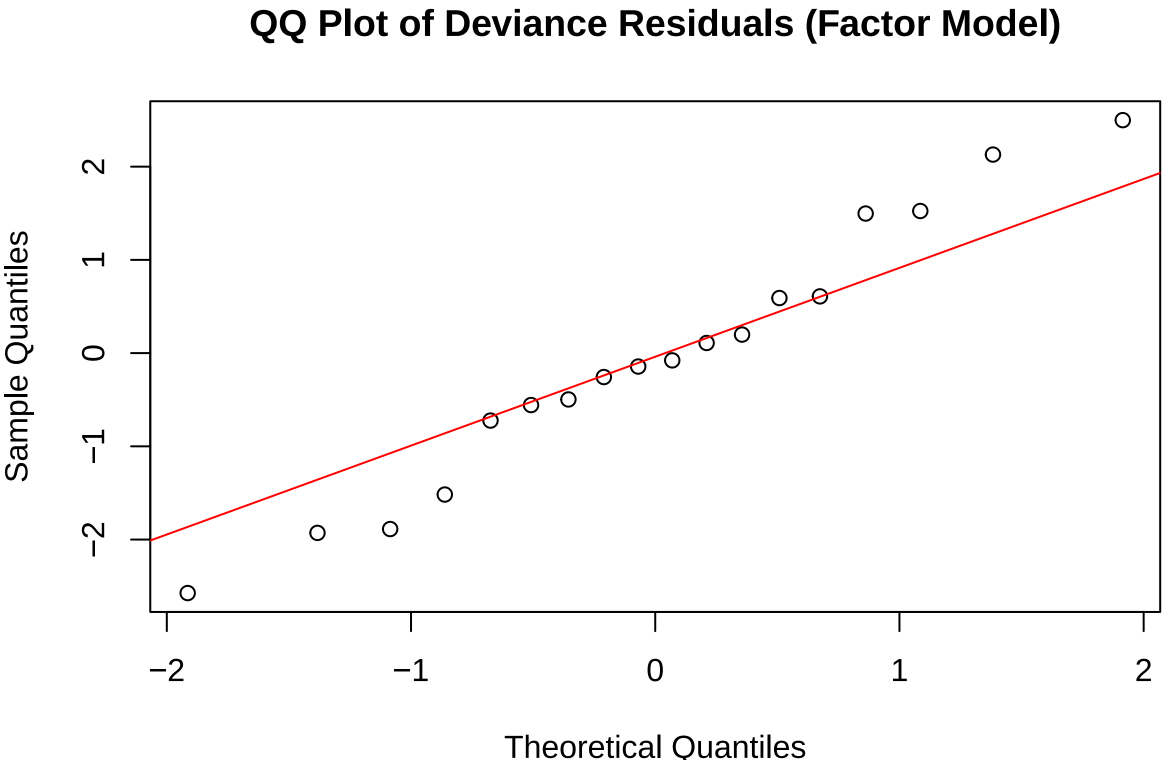
```
model_dose_factor$deviance / model_dose_factor$df.residual
```

```
## [1] 2.79133
```

Based on the deviance results, this model suffers from overdispersion. Deviance / df is not close to 1 so the model does not fit well. A transformation alone will not suffice but a model with a more flexible variance structure like quasi-poisson or negative binomial should work.

(d) Make a QQ plot of the residuals from the previous model. Interpret the plot.

```
# QQ plot of deviance residuals from the categorical factor model
qqnorm(residuals(model_dose_factor, type = "deviance"),
       main = "QQ Plot of Deviance Residuals (Factor Model)")
qqline(residuals(model_dose_factor, type = "deviance"), col = "red")
```



Deviance residuals line explicitly at both extremes, forming an S curve. This clearly indicates the distribution of residuals is heavier or lighter tailed than the theoretical distribution used. This means the model does not accurately describe the data, confirming the poor fit we suspected from the deviance test (overdispersion).

(e) Fit a Poisson model that includes an overdispersion parameter and is quadratic in the dose. Can we determine from the deviance of this model whether the fit is adequate?

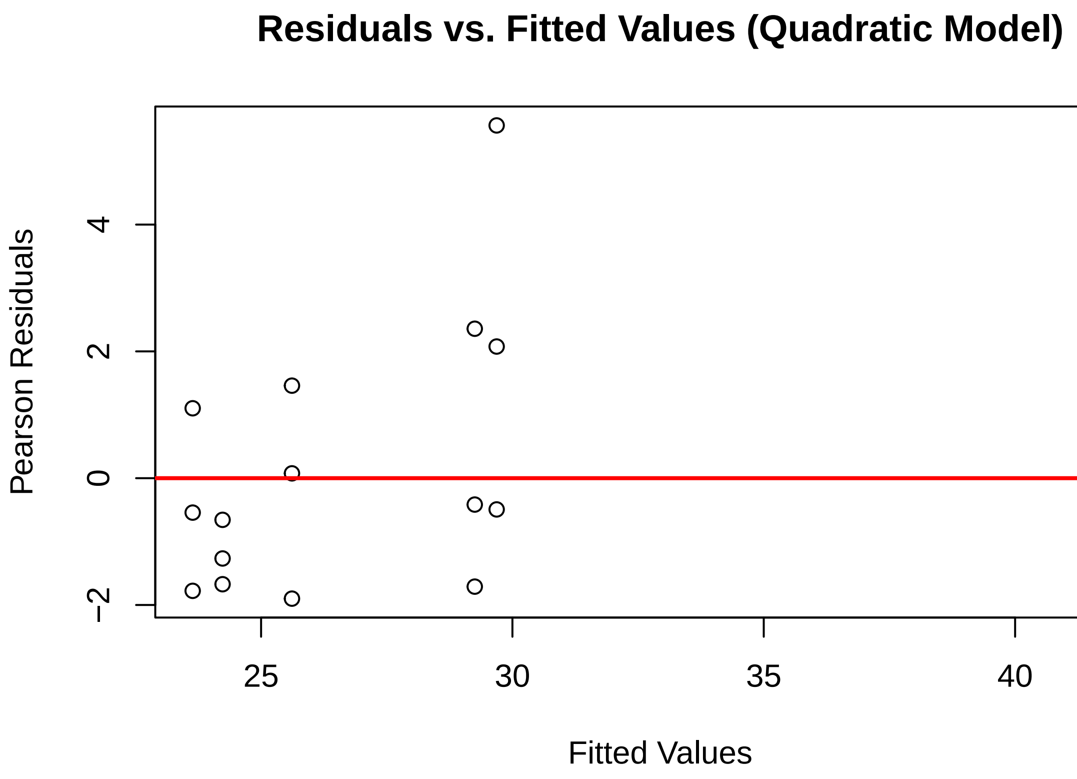
```
# Fit quadratic quasipoisson model (includes overdispersion)
model_quad <- glm(colonies ~ dose + I(dose^2), data = salmonella, family = quasipoisson(link = "log"))
summary(model_quad)
```

```
##
## Call:
## glm(formula = colonies ~ dose + I(dose^2), family = quasipoisson(link = "log"),
##     data = salmonella)
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.163e+00  1.361e-01  23.237 3.55e-13 ***
## dose         2.507e-03  1.040e-03   2.410   0.0293 *
## I(dose^2)   -2.294e-06  1.003e-06  -2.288   0.0371 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 4.126227)
##
##     Null deviance: 78.358  on 17  degrees of freedom
## Residual deviance: 55.535  on 15  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 4
```

In a quasi-poisson model, the deviance alone no longer tells you everything about the model's fit. Deviance / df being close to 1 means good mean-structure fit, high deviance / df indicates while quasi-poisson model adjusted for variance correctly, it missed nonlinear effects or structural terms important to the response.

(f) Plot the residuals against the fitted values for the previous model. Interpret the plot.

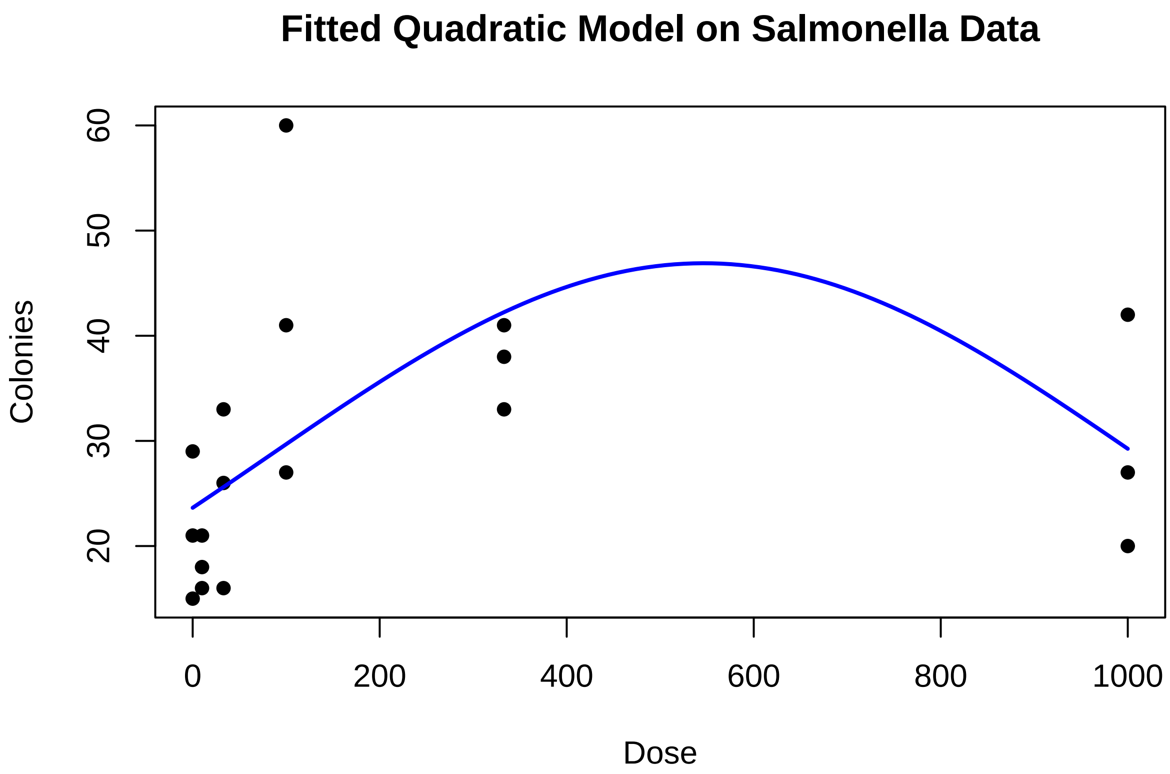
```
# Residual plot for quadratic quasipoisson model
plot(model_quad$fitted.values, residuals(model_quad, type = "pearson"),
     xlab = "Fitted Values", ylab = "Pearson Residuals",
     main = "Residuals vs. Fitted Values (Quadratic Model)")
abline(h = 0, col = "red", lwd = 2)
```



The residuals seem to be randomly dispersed however a pattern of clustering also seems to emerge, where residuals are clustered around fitted values of 25, 30 and 43.

(g) Plot the fitted mean response of this model on top of the data.

```r
# Plotting fitted quadratic curve on data
dose_seq <- seq(min(salmonella$dose), max(salmonella$dose), length.out = 100)
predicted_mu <- predict(model_quad, newdata = data.frame(dose = dose_seq), type = "response")

plot(salmonella$dose, salmonella$colonies, pch = 16,
     xlab = "Dose", ylab = "Colonies",
     main = "Fitted Quadratic Model on Salmonella Data")
lines(dose_seq, predicted_mu, col = "blue", lwd = 2)
```



(h) Give the predicted mean response for a dose of 500. Compute a 95% confidence interval.

```
# Predict at dose = 500 and calculate confidence interval
pred_500 <- predict(model_quad, newdata = data.frame(dose = 500), se.fit = TRUE, type = "link")

# Predicted mean (response scale)
predicted_mean <- exp(pred_500$fit)

# 95% CI (response scale)
ci_lower <- exp(pred_500$fit - 1.96 * pred_500$se.fit)
ci_upper <- exp(pred_500$fit + 1.96 * pred_500$se.fit)

cat("Predicted mean response at dose 500:", predicted_mean, "\n")
```

```
## Predicted mean response at dose 500: 46.66347
```

```
cat("95% CI: [", ci_lower, ",", ci_upper, "]\n")
```

```
## 95% CI: [ 31.14967 , 69.90377 ]
```

95% CI : [31.14967, 69.90377]

(i) At what dose does the maximum predicted response occur?

```
# Extract quadratic model coefficients
coef_quad <- coef(model_quad)

# Calculate the dose at vertex (maximum response)
optimal_dose <- -coef_quad["dose"] / (2 * coef_quad["I(dose^2)"])

cat("Dose at maximum predicted response:", optimal_dose, "\n")
```

```
## Dose at maximum predicted response: 546.4212
```

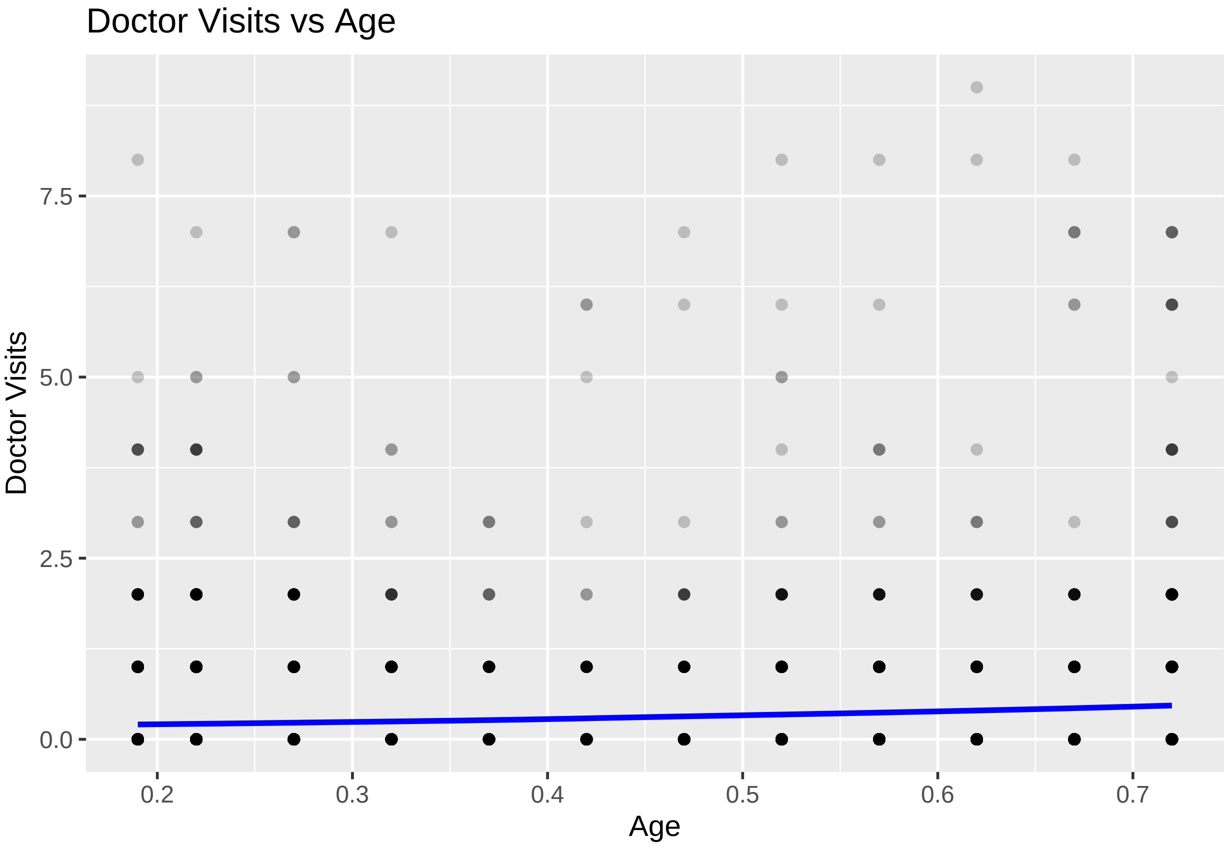
Dose at maximum predicted response is 546.4212.

3.[4pts] The dvisits data comes from the Australian Health Survey of 1977–1978 and consist of 5190 single adults where young and old have been oversampled.

(a) Make plots which show the relationship between the response variable, doctorco, and the potential predictors, age and illness.

```
library(faraway)
data("dvisits")
# doctorco vs age
ggplot(dvisits, aes(x = age, y = doctorco)) +
  geom_point(alpha=0.2) +
  geom_smooth(method = "loess", se=FALSE, col="blue") +
  labs(title="Doctor Visits vs Age", x="Age", y="Doctor Visits")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

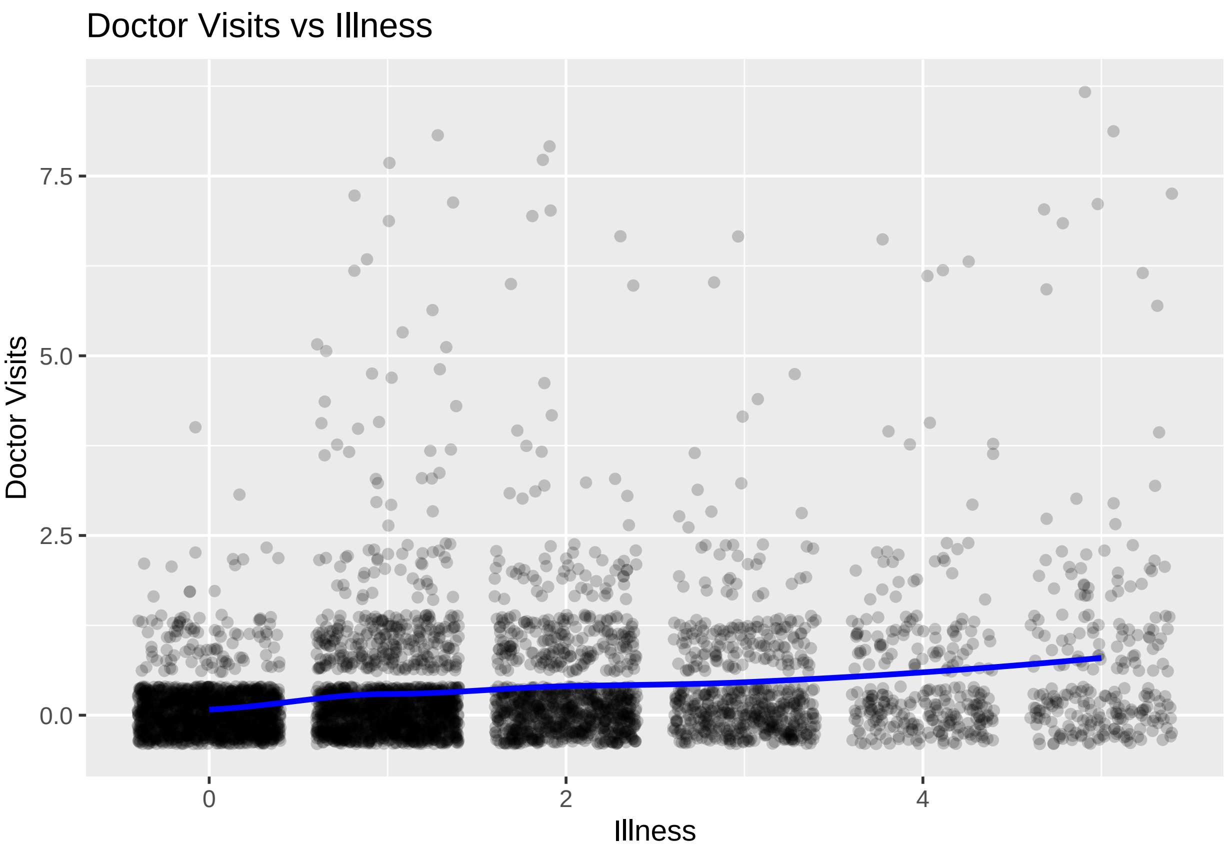


```
# doctorco vs illness
ggplot(dvisits, aes(x = illness, y = doctorco)) +
  geom_jitter(alpha=0.2) +
  geom_smooth(method = "loess", se=FALSE, col="blue") +
  labs(title="Doctor Visits vs Illness", x="Illness", y="Doctor Visits")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : pseudoinverse used at -0.025
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : neighborhood radius 2.025

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : reciprocal condition number 7.482e-15

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : There are other near singularities as well. 1
```

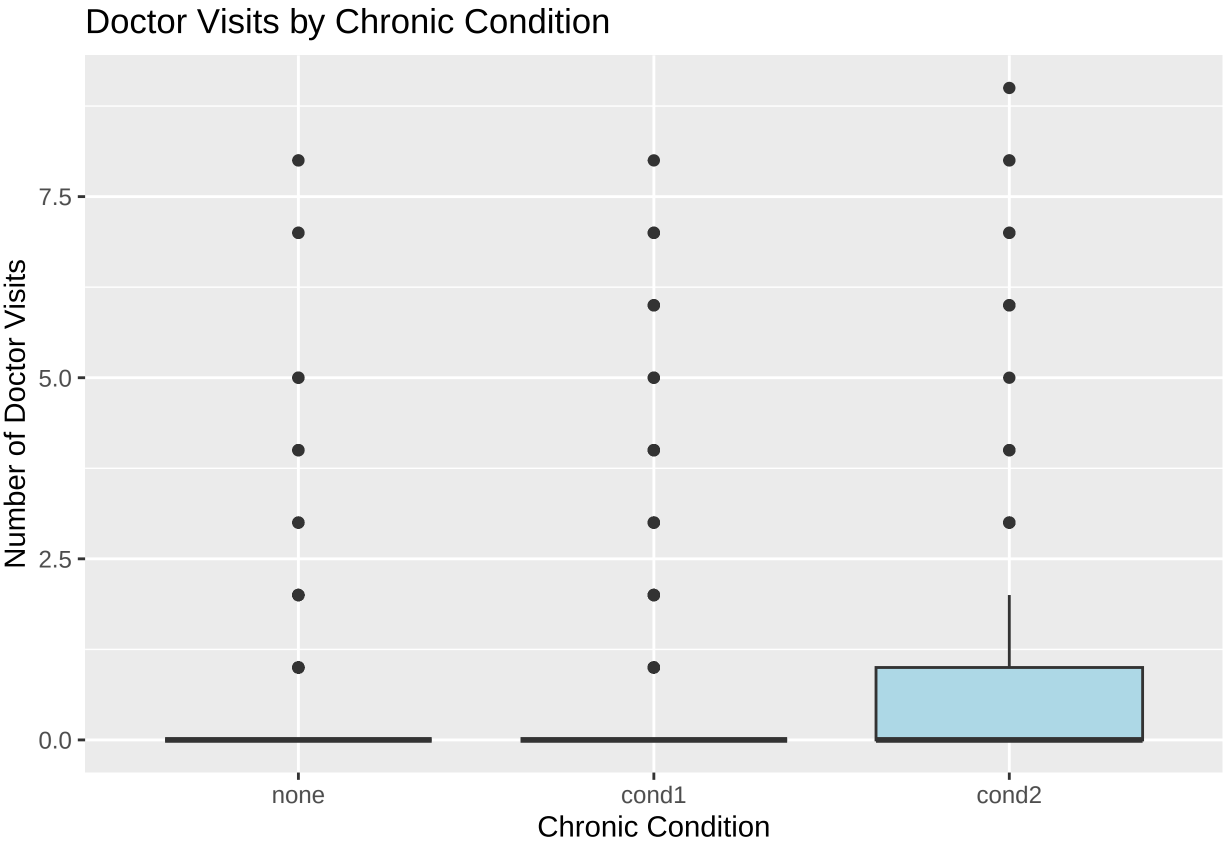


(b) Combine the predictors `chcond1` and `chcond2` into a single three-level factor. Make an appropriate plot showing the relationship between this factor and the response. Comment.

```r
library(dplyr)

# Explicit numeric coding: 0 = none, 1 = condition1 only, 2 = condition2 only
dvisits <- dvisits %>%
  mutate(chronic = case_when(
    chcond1 == 1 ~ 1,
    chcond2 == 1 ~ 2,
    TRUE ~ 0
  ))

# Convert to factor explicitly for analysis clarity
dvisits$chronic <- factor(dvisits$chronic,
                          levels = c(0, 1, 2),
                          labels = c("none", "cond1", "cond2"))

# Verify results
table(dvisits$chronic)
```

```
##
##  none cond1 cond2
##  2493  2092   605
```

```
library(ggplot2)

ggplot(dvisits, aes(x = chronic, y = doctorco)) +
  geom_boxplot(fill = "lightblue") +
  labs(title = "Doctor Visits by Chronic Condition",
       x = "Chronic Condition",
       y = "Number of Doctor Visits")
```



It seems that cond2 is the strongest predictor of numerous doctors visits. There is a higher median number of doctors visits and more variability based on this predictor which is shown in the boxplot.

(c) Build a Poisson regression model with `doctorco` as the response and `sex`, `age`, `agesq`, `income`, `levyplus`, `freepoor`, `freerepa`, `illness`, `actdays`, `hscore` and the three-level condition factor as possible predictor variables. Considering the deviance of this model, does this model fit the data?

```
model_pois <- glm(doctorco ~ sex + age + agesq + income + levyplus + freepoor +
                    freerepa + illness + actdays + hscore + chronic,
                  family = poisson(link = "log"), data = dvisits)

summary(model_pois)
```

```
##
## Call:
## glm(formula = doctorco ~ sex + age + agesq + income + levyplus +
##     freepoor + freerepa + illness + actdays + hscore + chronic,
##     family = poisson(link = "log"), data = dvisits)
## 
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)    
## (Intercept)  -2.223848   0.189816 -11.716   <2e-16 ***
## sex           0.156882   0.056137   2.795   0.0052 ** 
## age           1.056299   1.000780   1.055   0.2912    
## agesq        -0.848704   1.077784  -0.787   0.4310    
## income       -0.205321   0.088379  -2.323   0.0202 *  
## levyplus      0.123185   0.071640   1.720   0.0855 .  
## freepoor     -0.440061   0.179811  -2.447   0.0144 *  
## freerepa      0.079798   0.092060   0.867   0.3860    
## illness       0.186948   0.018281  10.227   <2e-16 ***
## actdays       0.126846   0.005034  25.198   <2e-16 ***
## hscore        0.030081   0.010099   2.979   0.0029 ** 
## chroniccond1  0.114085   0.066640   1.712   0.0869 .  
## chroniccond2  0.141158   0.083145   1.698   0.0896 .  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for poisson family taken to be 1)
## 
##     Null deviance: 5634.8  on 5189  degrees of freedom
## Residual deviance: 4379.5  on 5177  degrees of freedom
## AIC: 6737.1
## 
## Number of Fisher Scoring iterations: 6
```

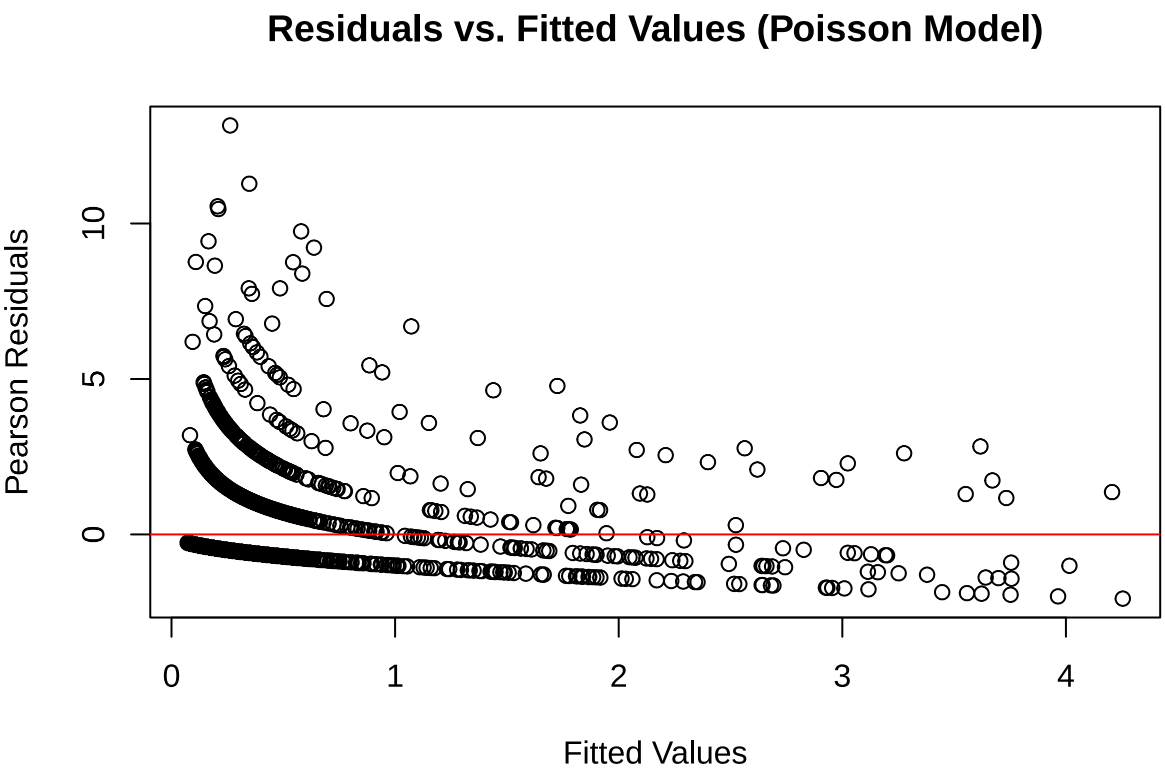
```
model_pois$deviance / model_pois$df.residual
```

```
## [1] 0.8459562
```

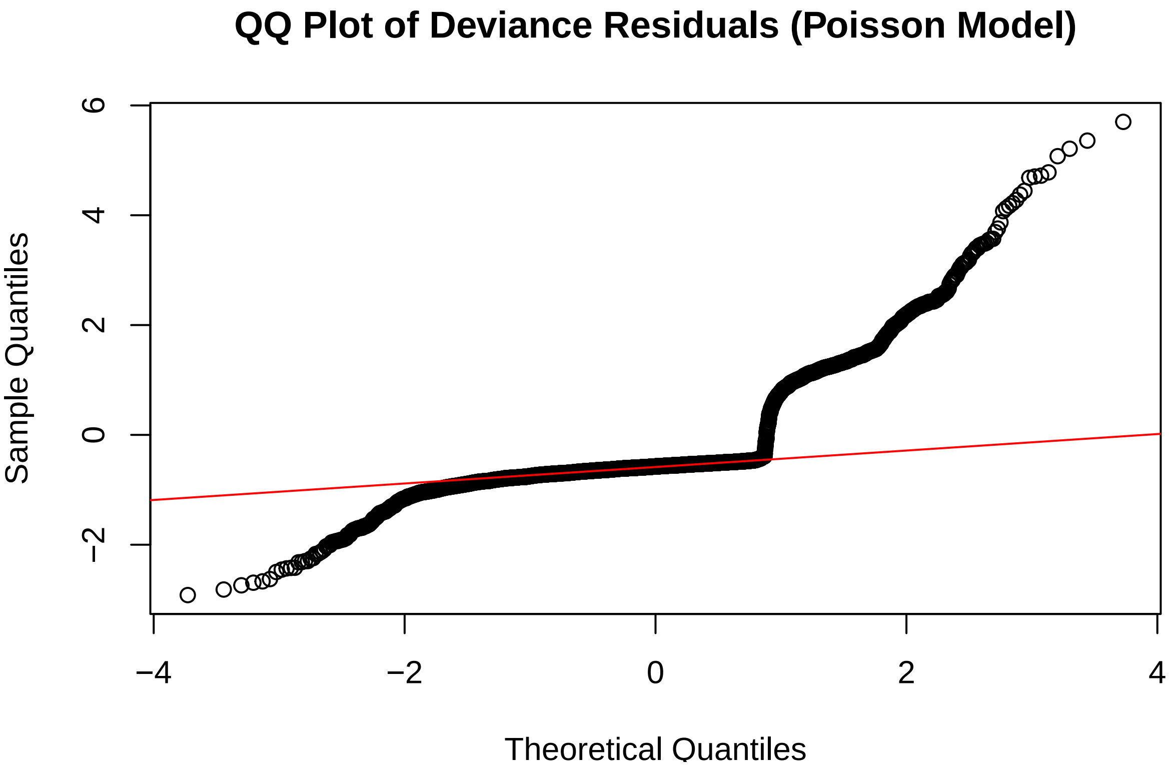
Based on the Deviance, Deviance / df is close to 1, so the model fit to the data is good!

(d) Plot the residuals and the fitted values — why are there lines of observations on the plot? Make a QQ plot of the residuals and comment.

```
# Residual plot (Pearson residuals vs fitted values)
plot(model_pois$fitted.values, residuals(model_pois, type = "pearson"),
     xlab = "Fitted Values",
     ylab = "Pearson Residuals",
     main = "Residuals vs. Fitted Values (Poisson Model)")
abline(h = 0, col = "red")
```



```r
# QQ plot of deviance residuals
qqnorm(residuals(model_pois, type = "deviance"),
       main = "QQ Plot of Deviance Residuals (Poisson Model)")
qqline(residuals(model_pois, type = "deviance"), col = "red")
```



The response variable, number of doctor visits, is a discrete count, so observations with the same actual count will cluster together, creating distinct horizontal lines or bands in the residual plot. The QQ plot shows evidence of overdispersion, this means the model might have balanced the mean-variance but missed the actual data distribution, because deviance/ df looks good but the model is still inadequate.

(e) Use a stepwise AIC-based model selection method. What sort of person would be predicted to visit the doctor the most under your selected model?

```r
# Stepwise selection using AIC criterion
model_step <- step(model_pois, direction = "both", trace = FALSE)
summary(model_step)
```

```
##
## Call:
## glm(formula = doctorco ~ sex + age + income + levyplus + freepoor +
##     illness + actdays + hscore + chronic, family = poisson(link = "log"),
##     data = dvisits)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.089063   0.100811 -20.723  < 2e-16 ***
## sex           0.162000   0.055824   2.902  0.00371 **
## age           0.355131   0.143196   2.480  0.01314 *
## income       -0.199806   0.084328  -2.369  0.01782 *
## levyplus      0.083689   0.053544   1.563  0.11805
## freepoor     -0.469596   0.176360  -2.663  0.00775 **
## illness       0.186101   0.018260  10.191  < 2e-16 ***
## actdays       0.126611   0.005029  25.177  < 2e-16 ***
## hscore        0.031116   0.010065   3.092  0.00199 **
## chroniccond1  0.121100   0.066389   1.824  0.06814 .
## chroniccond2  0.158894   0.081762   1.943  0.05197 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 5634.8  on 5189  degrees of freedom
## Residual deviance: 4381.0  on 5179  degrees of freedom
## AIC: 6734.5
##
## Number of Fisher Scoring iterations: 6
```

Based on the beta coefficients, the most impactful predictors are illness and actdays which means a person with these characteristics are the most likely to go visit the doctor. This makes intuitive sense, if you have an illness you will go visit the doctor.

(f) For the last person in the dataset,compute the predicted probability distribution for their visits to the doctor, i.e., give the probability they visit 0, 1, 2, etc. times.

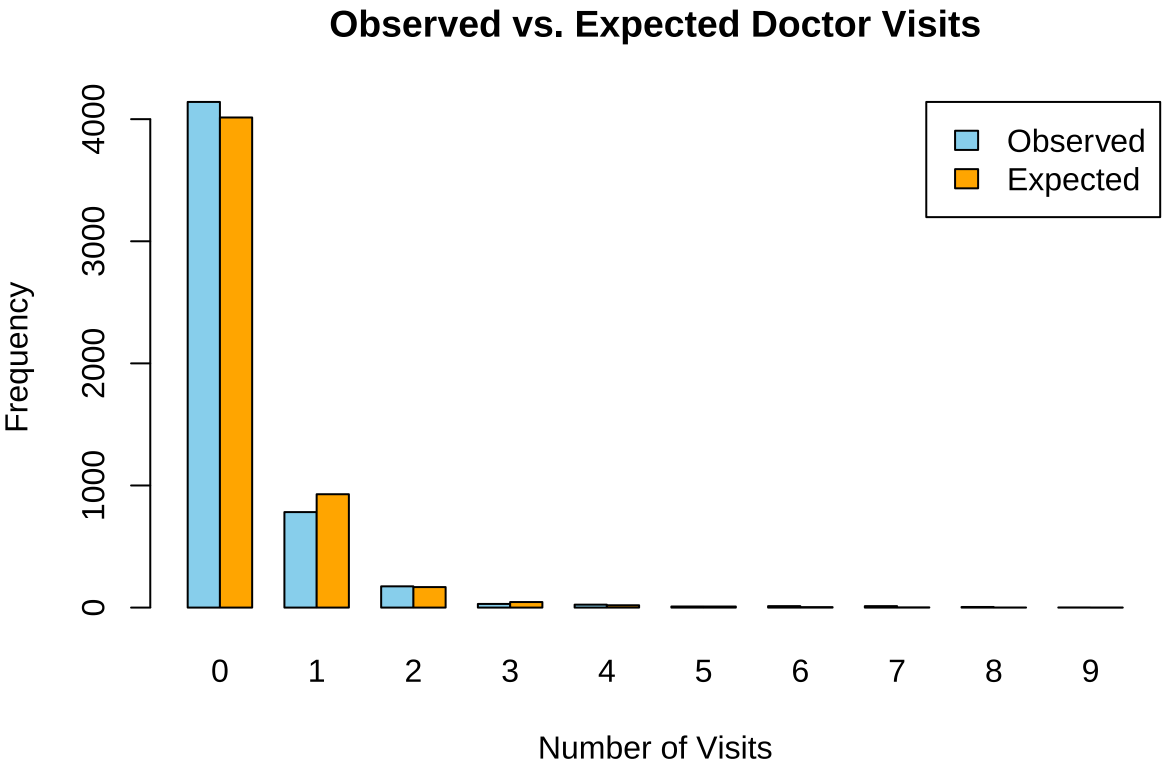
```r
# Last individual in dataset
last_person <- tail(dvisits, 1)

# Predicted mean for last person
pred_mean_last <- predict(model_step, newdata = last_person, type = "response")

# Probabilities of 0-10 doctor visits
visit_counts <- 0:10
visit_probs <- dpois(visit_counts, lambda = pred_mean_last)
```

```r
# Tabulated probabilities
prob_table <- data.frame(Visits = visit_counts, Probability = visit_probs)
prob_table
```

```
##    Visits  Probability
## 1       0 8.589160e-01
## 2       1 1.306275e-01
## 3       2 9.933193e-03
## 4       3 5.035606e-04
## 5       4 1.914590e-05
## 6       5 5.823578e-07
## 7       6 1.476124e-08
## 8       7 3.207073e-10
## 9       8 6.096814e-12
## 10      9 1.030255e-13
## 11     10 1.566854e-15
```

(g) Tabulate the frequencies of the number of doctor visits. Compute the expected frequencies of doctor visits under your most recent model. Compare the observed with the expected frequencies and comment on whether it is worth fitting a zero-inflated count model.

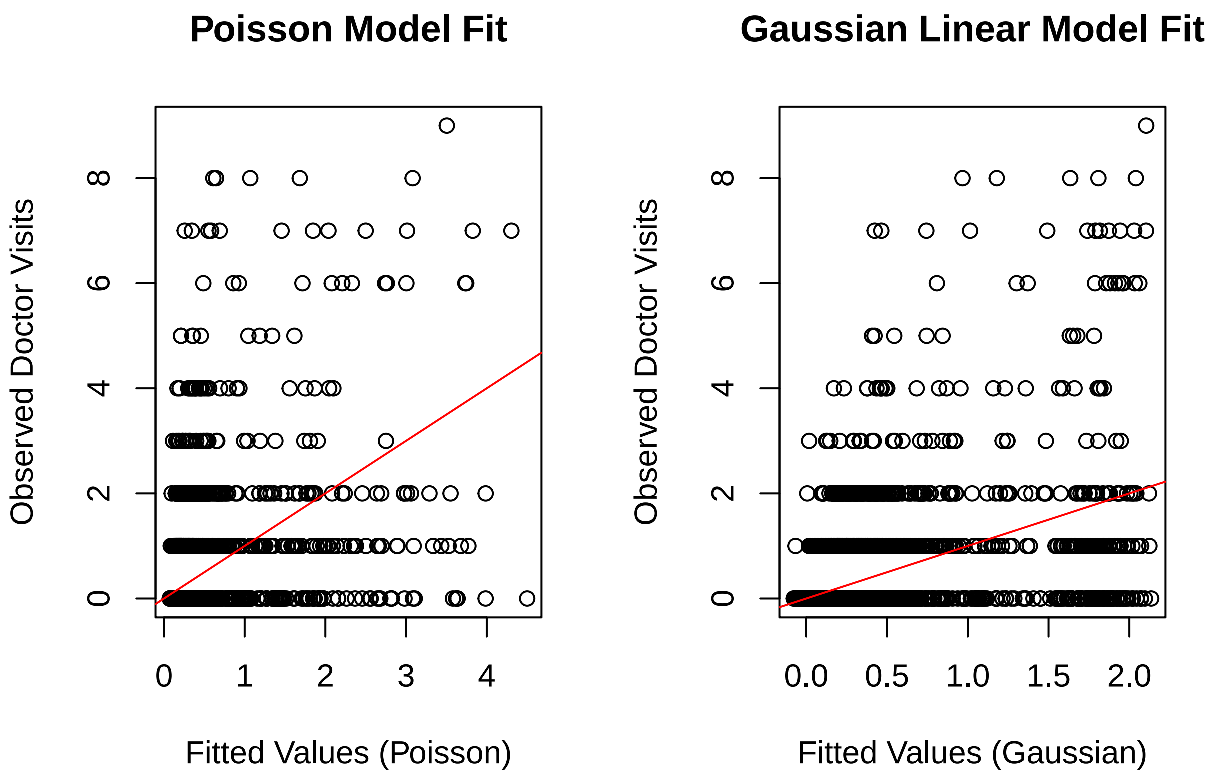
```r
# Observed visit frequencies
obs_freq <- table(dvisits$doctorco)

# Expected frequencies under the fitted stepwise model
predicted_means <- predict(model_step, type = "response")
expected_freq <- sapply(0:max(dvisits$doctorco), function(x) sum(dpois(x, predicted_means)))

# Combine observed vs expected frequencies
freq_df <- data.frame(
  Visits = 0:max(dvisits$doctorco),
  Observed = as.numeric(obs_freq),
  Expected = expected_freq
)

# Plot observed vs expected frequencies
barplot(t(as.matrix(freq_df[, c("Observed", "Expected")])),
        beside = TRUE,
        names.arg = freq_df$Visits,
        col = c("skyblue", "orange"),
        legend.text = c("Observed", "Expected"),
        args.legend = list(x = "topright"),
        main = "Observed vs. Expected Doctor Visits",
        xlab = "Number of Visits",
        ylab = "Frequency")
```



No a zero-inflated model does not appear necessary given that the model captures the zero counts well.

(h) Fit a comparable (Gaussian) linear model and graphically compare the fits. Describe how they differ.

```
# Fit Gaussian (linear) model
model_lm <- lm(doctorco ~ sex + age + agesq + income + levyplus + freepoor +
                 freerepa + illness + actdays + hscore + chronic, data = dvisits)

# Graphically compare fitted values
par(mfrow = c(1,2))

# Poisson Model Fit
plot(model_step$fitted.values, dvisits$doctorco,
     xlab = "Fitted Values (Poisson)",
     ylab = "Observed Doctor Visits",
     main = "Poisson Model Fit")
abline(a = 0, b = 1, col = "red")

# Gaussian Model Fit
plot(model_lm$fitted.values, dvisits$doctorco,
     xlab = "Fitted Values (Gaussian)",
     ylab = "Observed Doctor Visits",
     main = "Gaussian Linear Model Fit")
abline(a = 0, b = 1, col = "red")
```



```r
par(mfrow = c(1,1))
```

Fitted values for Poisson are mostly betewen 0 and 4, fitted values for Gaussian are mostly between 0 and 2. More vertical spread for high counts for poisson (4+ doctor visits) than Gaussian (less spread, fitted values compressed). Poisson accounts for discreteness of data, Gaussian assumes a continuous response which isn't ideal for count data. Poisson fits exponential mean structure while Gaussian assumes a linear relationship.

4.[3pts] Data is generated from the exponential distribution with density

$$f(y) = \lambda \exp(-\lambda y), \lambda, y > 0$$

a) Identify the specific form of $\theta, \phi, a(), b()$ and $c()$ for the exponential distribution.

```r
# Done in separate pdf for personal convenience
```

b) What is the canonical link and variance function for a GLM with a response following the exponential distribution?

```r
# Done in separate pdf for personal convenience
```

c) Identify a proactical difficulty that may arise when using the canonical link in this instance.

```r
# Done in separate pdf for personal convenience
```

d) When comparing nested models in this case, should an F or χ^2 test be used? Explain.

```r
# Done in separate pdf for personal convenience
```

e) Express the deviance in this case in terms of the responses y_i and the fitted values $\hat{\mu}_i$.

```r
# Done in separate pdf for personal convenience
```

5.[5pts] Consider the Galapagos data and model analyzed in chapter 8. The purpose of this question is to reproduce the details of the GLM fitting of this data.

(a) Fit a Poisson model to the species response with the five geographic variables as predictors. Do not use the endemics variable. Report the values of the coefficients and the deviance.

```r
library(faraway)
data("gala")
# Fit Poisson GLM without endemics
model_gala <- glm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent,
                  family = poisson(link = "log"), data = gala)

# Display summary of coefficients and deviance
summary(model_gala)
```

```
##
## Call:
## glm(formula = Species ~ Area + Elevation + Nearest + Scruz +
##     Adjacent, family = poisson(link = "log"), data = gala)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.155e+00  5.175e-02  60.963  < 2e-16 ***
## Area        -5.799e-04  2.627e-05 -22.074  < 2e-16 ***
## Elevation    3.541e-03  8.741e-05  40.507  < 2e-16 ***
## Nearest      8.826e-03  1.821e-03   4.846 1.26e-06 ***
## Scruz       -5.709e-03  6.256e-04  -9.126  < 2e-16 ***
## Adjacent    -6.630e-04  2.933e-05 -22.608  < 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: 3510.73  on 29  degrees of freedom
## Residual deviance:  716.85  on 24  degrees of freedom
## AIC: 889.68
##
## Number of Fisher Scoring iterations: 5
```

```r
vif(model_gala)
```

```
##       Area  Elevation    Nearest      Scruz   Adjacent
## 0.01494704 0.03938240 0.01960072 0.05253515 0.01864225
```

The deviance is quite high, even given the range of the response. It is clear that something important is either omitted from the model or the model is suffering from other problems, but multicollinearity is not one of them as VIF confirmed.

(b) For a Poisson GLM, derive $\eta, d\eta/d\mu, V(\mu)$ and the weights to be used in an iteratively fit GLM. What is the form of the adjusted dependent variable here?

```r
# Done in separate pdf for personal convenience
```

(c) Using the observed response as initial values, compute the first stage of the iteration, stopping after the first linear model fit. Compare the coefficients of this linear model to those found in the GLM fit. How close are they?

```r
# Initial values set to observed response y (Species)
mu_init <- gala$Species
eta_init <- log(mu_init)
```

```r
# Calculate derivative and weights explicitly
d_eta_d_mu <- 1 / mu_init
weights <- mu_init

# Compute adjusted dependent variable explicitly
z <- eta_init + (gala$Species - mu_init) * d_eta_d_mu

# Perform first iteration explicitly using weighted regression
iteration1 <- lm(z ~ Area + Elevation + Nearest + Scruz + Adjacent, weights = weights, data = gala)
summary(iteration1)
```

```
##
## Call:
## lm(formula = z ~ Area + Elevation + Nearest + Scruz + Adjacent,
##     data = gala, weights = weights)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -8.5272 -4.2324 -2.2714  0.2647  7.8018
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.5191545  0.2931247  12.006 1.24e-11 ***
## Area        -0.0005298  0.0001379  -3.843 0.000783 ***
## Elevation    0.0031644  0.0004757   6.651 7.03e-07 ***
## Nearest      0.0025189  0.0077627   0.324 0.748382
## Scruz       -0.0037900  0.0029756  -1.274 0.214976
## Adjacent    -0.0006624  0.0001490  -4.444 0.000171 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.902 on 24 degrees of freedom
## Multiple R-squared:  0.7571, Adjusted R-squared:  0.7065
## F-statistic: 14.96 on 5 and 24 DF,  p-value: 1.047e-06
```

```r
# Compare coefficients explicitly to GLM
coefficients(iteration1)
```

```
##   (Intercept)          Area     Elevation       Nearest         Scruz
##  3.5191545412 -0.0005298484  0.0031643557  0.0025188990 -0.0037899780
##      Adjacent
## -0.0006623523
```

```r
coefficients(model_gala)
```

```
##   (Intercept)          Area     Elevation       Nearest         Scruz
##  3.1548078779 -0.0005799429  0.0035405940  0.0088255719 -0.0057094223
##      Adjacent
## -0.0006630311
```

They are very close to each other as expected. All values are almost identical with the intercept being the most different across the two methods.