Homework #8

Zhijian Liu

2. handout R08_PoissonLoglinear_2019.R.

(a) What is the Deviance for GOF and its df in this case?

```
tumorfit <- glm(count ~ type+site+type*site, family=poisson(link=log), data=tumordata)
tumorfit2 <- glm(count ~ type+site, family=poisson(link=log), data=tumordata)
tumorfit2$dev # Deviance for GOF
```

```
## [1] 51.79501
```

The deviance for GOF is 51.79501 and its df = 6.

(b) How to test if type and site are independent?

```
# option 1: contingency table chi2 test
tumortable
```

```
##      aHNK TNK EXT
## aHMF   22   2  10
## SSM    16  54 115
## NOD    19  33  73
## IND    11  17  28
```

```
chisq.test(tumortable) # df = 6
```

```
##
##     Pearson's Chi-squared test
##
## data:  tumortable
## X-squared = 65.813, df = 6, p-value = 2.943e-12
```

```
# option 2: Deviance GOF
c("Deviance (LRT) GOF" = tumorfit2$dev, "p-value" = 1- pchisq(tumorfit2$dev, 6))
```

```
## Deviance (LRT) GOF            p-value
##       5.179501e+01       2.050453e-09
```

```
# option 3: Pearson GOF
pear<-residuals(tumorfit2, type="pearson")
PGOF<-sum(pear^2)
c("Pearson GOF" = PGOF, "p-value" = 1- pchisq(PGOF, 6))
```

```
##  Pearson GOF      p-value
## 6.581293e+01 2.943201e-12
```

The p-values of the tests are closed to 0, so we reject the null hypotheses and conclude that type and site are not independent.

(c) What is the test statistic for global test of model significance and its df?

```
G <- tumorfit2$null.dev-tumorfit2$dev
c("Chi-square for model signficance"=G, "p-value"=1-pchisq(G, 5)) # df = 6-1
```

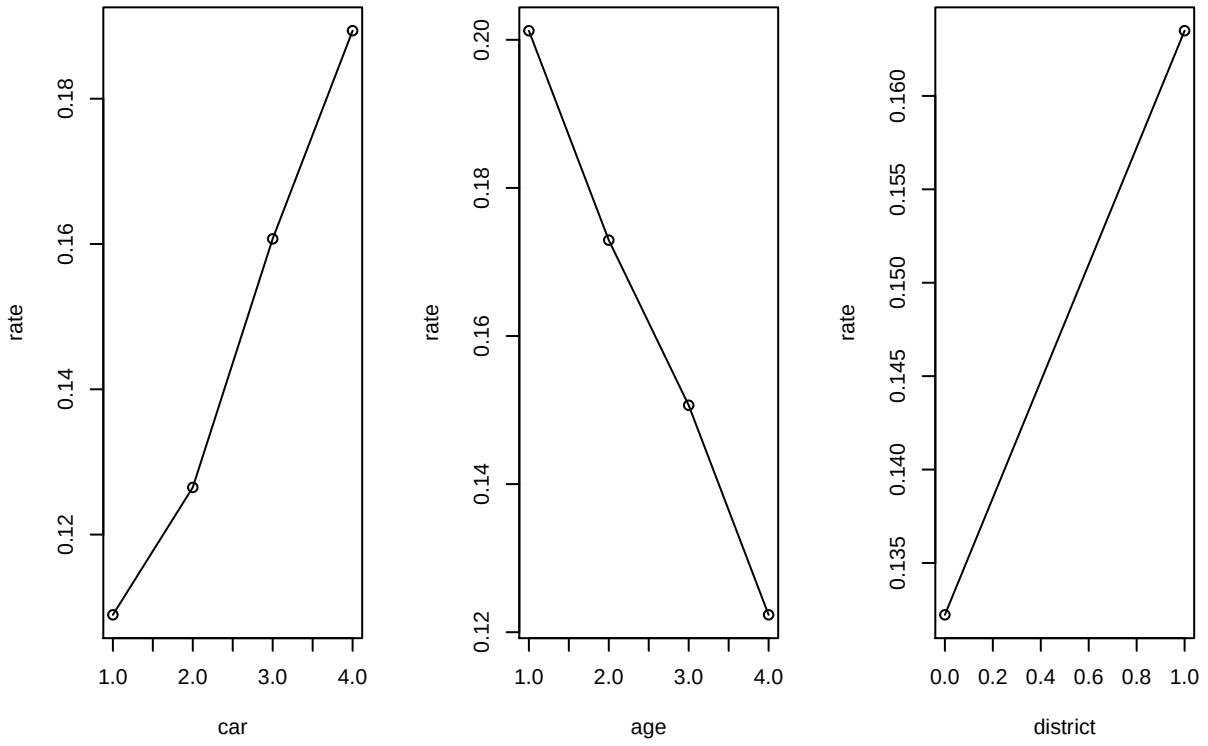
```
## Chi-square for model signficance                    p-value
##                          243.408                      0.000
```

(d) How do we interpret the results?
The p-value of the overall significance LRT is closed to 0, so at least one predictor has relationship with the occurrence rate.

(e) What's the benefit of using Poisson (log-linear) over just using Chi-square for contingency table? The advantage of log- linear modelling over the conventional chi-squared test for independence is that it provides a method for analyzing more complicated cross-tabulated data.

3. Exercise 9.2

(a)



(b)
```
for (i in 1:3){df[,i] <- df[,i] %>% unlist() %>% as.factor()}
full <- glm(y ~ car*age*district,family=poisson(link=log), offset = log(n),data=df)
summary(full)
```

```
##
## Call:
## glm(formula = y ~ car * age * district, family = poisson(link = log),
##     data = df, offset = log(n))
##
## Deviance Residuals:
##  [1]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## [24]  0  0  0  0  0  0  0  0  0
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -1.585e+00  1.240e-01 -12.775  < 2e-16 ***
```

```
## car2                -1.673e-02  1.600e-01  -0.105 0.916721
## car3                -1.091e-01  1.994e-01  -0.547 0.584401
## car4                 2.935e-01  3.260e-01   0.900 0.367947
## age2                -4.065e-01  1.754e-01  -2.317 0.020477 *
## age3                -6.504e-01  1.860e-01  -3.496 0.000472 ***
## age4                -7.681e-01  1.364e-01  -5.630  1.8e-08 ***
## district1           -7.181e-01  7.179e-01  -1.000 0.317198
## car2:age2            1.649e-01  2.174e-01   0.759 0.448106
## car3:age2            5.726e-01  2.524e-01   2.269 0.023298 *
## car4:age2            2.556e-01  3.876e-01   0.660 0.509574
## car2:age3            2.049e-01  2.248e-01   0.912 0.361989
## car3:age3            7.173e-01  2.575e-01   2.785 0.005345 **
## car4:age3            2.390e-01  3.888e-01   0.615 0.538711
## car2:age4            2.021e-01  1.731e-01   1.168 0.242996
## car3:age4            4.854e-01  2.124e-01   2.286 0.022271 *
## car4:age4            2.505e-01  3.399e-01   0.737 0.461104
## car2:district1       8.312e-01  8.176e-01   1.017 0.309299
## car3:district1       1.131e+00  8.601e-01   1.315 0.188626
## car4:district1      -2.139e+01  4.225e+04  -0.001 0.999596
## age2:district1       8.220e-01  8.548e-01   0.962 0.336246
## age3:district1       6.504e-01  8.858e-01   0.734 0.462753
## age4:district1       8.984e-01  7.392e-01   1.215 0.224188
## car2:age2:district1 -1.184e+00  9.950e-01  -1.190 0.234003
## car3:age2:district1 -1.425e+00  1.052e+00  -1.354 0.175588
## car4:age2:district1  2.175e+01  4.225e+04   0.001 0.999589
## car2:age3:district1 -4.298e-01  9.944e-01  -0.432 0.665567
## car3:age3:district1 -8.832e-01  1.039e+00  -0.850 0.395125
## car4:age3:district1  2.202e+01  4.225e+04   0.001 0.999584
## car2:age4:district1 -8.042e-01  8.428e-01  -0.954 0.340026
## car3:age4:district1 -1.032e+00  8.881e-01  -1.162 0.245079
## car4:age4:district1  2.178e+01  4.225e+04   0.001 0.999589
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 2.0783e+02  on 31  degrees of freedom
## Residual deviance: 4.1219e-10  on  0  degrees of freedom
## AIC: 232.36
##
## Number of Fisher Scoring iterations: 20
```

Based on the result, all 3-way interaction terms and the 2-way interaction terms between district and two other predictors are not significant. So I drop all these insignificant interaction terms.

(c)
```
reg.b <- glm(y ~ car*age + district,family=poisson(link=log), offset = log(n), data=df)
for (i in 1:2){df[,i] <- df[,i] %>% unlist() %>% as.numeric()}
reg <- glm(y ~ car+age+district,family=poisson(link=log),offset = log(n), data=df)
summary(reg); summary(reg.b)
```

```
##
## Call:
```

```
## glm(formula = y ~ car + age + district, family = poisson(link = log),
##     data = df, offset = log(n))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7248  -0.5681  -0.1679   0.3384   1.9126
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.85253    0.07990 -23.185  < 2e-16 ***
## car          0.19777    0.02080   9.507  < 2e-16 ***
## age         -0.17674    0.01849  -9.559  < 2e-16 ***
## district1    0.21865    0.05853   3.736 0.000187 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 207.833  on 31  degrees of freedom
## Residual deviance:  24.685  on 28  degrees of freedom
## AIC: 201.05
##
## Number of Fisher Scoring iterations: 4
```

```
##
## Call:
## glm(formula = y ~ car * age + district, family = poisson(link = log),
##     data = df, offset = log(n))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4804  -0.2481  -0.0349   0.3179   1.6404
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.62983    0.12224 -13.333  < 2e-16 ***
## car2         0.02116    0.15636   0.135 0.892356
## car3        -0.04446    0.19148  -0.232 0.816377
## car4         0.24957    0.32532   0.767 0.442991
## age2        -0.36989    0.17091  -2.164 0.030450 *
## age3        -0.62858    0.18106  -3.472 0.000517 ***
## age4        -0.72688    0.13349  -5.445 5.17e-08 ***
## district1    0.21916    0.05854   3.744 0.000181 ***
## car2:age2    0.10099    0.21131   0.478 0.632705
## car3:age2    0.48849    0.24293   2.011 0.044346 *
## car4:age2    0.34022    0.38015   0.895 0.370800
## car2:age3    0.20265    0.21768   0.931 0.351867
## car3:age3    0.67189    0.24710   2.719 0.006546 **
## car4:age3    0.35722    0.38075   0.938 0.348145
## car2:age4    0.16706    0.16841   0.992 0.321190
## car3:age4    0.43166    0.20349   2.121 0.033897 *
```

```
## car4:age4    0.34847    0.33723   1.033 0.301444
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 207.833  on 31  degrees of freedom
## Residual deviance:  13.192  on 15  degrees of freedom
## AIC: 215.55
##
## Number of Fisher Scoring iterations: 4
```

This model is better than (b) based on its lower AIC, 201.05. Also this model is simpler, and therefore easier to interpret.

4. Exercise 9.3 (a, b)

(a)
```
# conventional
df.mat <- matrix(df$frequency, ncol=3, byrow=T)
colnames(df.mat)<-c("small", "moderate", "large")
rownames(df.mat)<-c("placebo","vaccine")
(df.mat.tm <- df.mat %>% rowSums())
```

```
## placebo vaccine
##      38      35
```

```
chisq.test(df.mat.tm)
```

```
##
##     Chi-squared test for given probabilities
##
## data:  df.mat.tm
## X-squared = 0.12329, df = 1, p-value = 0.7255
```

```
# poisson
reg <- glm(frequency ~ treatment + response, family=poisson(link=log), data=df)
summary(reg)
```

```
##
## Call:
## glm(formula = frequency ~ treatment + response, family = poisson(link = log),
##     data = df)
##
## Deviance Residuals:
##      1       2       3       4       5       6
##  2.040  -1.630  -1.247  -2.615   1.469   1.128
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        2.11972    0.27408   7.734 1.04e-14 ***
## treatmentvaccine -0.08224    0.23428  -0.351   0.7256
## responsemoderate  0.48551    0.31774   1.528   0.1265
## responsesmall     0.66140    0.30783   2.149   0.0317 *
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 23.807  on 5  degrees of freedom
## Residual deviance: 18.643  on 2  degrees of freedom
## AIC: 51.771
##
## Number of Fisher Scoring iterations: 5
```

The p-value for the test is $0.7256 > 0.05$, so the distribution of responses is the same for the placebo and vaccine groups.

(b)
```
full <- glm(frequency ~ treatment * response, family=poisson(link=log), data=df)
full$fitted.values # fitted value
```

```
##  1  2  3  4  5  6
## 25  8  5  6 18 11
```

```
(dev <- full$deviance) # deviance residual & D
```

```
## [1] 1.776357e-15
```

```
(pear<-residuals(full, type="pearson")) # Pearson residual
```

```
##             1             2             3             4             5
## -5.684342e-15 -3.140185e-15  3.972055e-16 -3.263376e-15 -3.349531e-15
##             6
## -2.142367e-15
```

```
(X2<-sum(pear^2)) # X2
```

```
## [1] 6.878899e-29
```

```
df.mat
```

```
##         small moderate large
## placebo    25        8     5
## vaccine     6       18    11
```

```
which.max(pear^2)
```

```
## 1
## 1
```

```
c("X^2" = X2, "p-value" = 1- pchisq(X2, 2))
```

```
##          X^2      p-value
## 6.878899e-29 1.000000e+00
```

The cell of small response with placebo contribute most to X^2. The small X^2 indicates the homogeneity of response. distributions.

5. Exercise 9.5

(a)
```
##
## Call:
## glm(formula = frequency ~ contact * satisfaction, family = poisson(link = log),
##     data = df[df$type == unique(df$type)[1], ])
```

```
##
## Deviance Residuals:
## [1]  0  0  0  0  0  0
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   4.605e+00  1.000e-01  46.052  < 2e-16 ***
## contactlow                   -1.338e-16  1.414e-01   0.000   1.0000
## satisfactionlow              -1.079e+00  1.985e-01  -5.434 5.51e-08 ***
## satisfactionmedium           -7.550e-01  1.769e-01  -4.269 1.96e-05 ***
## contactlow:satisfactionlow    6.480e-01  2.546e-01   2.546   0.0109 *
## contactlow:satisfactionmedium 1.388e-01  2.445e-01   0.568   0.5702
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 5.7491e+01  on 5  degrees of freedom
## Residual deviance: 2.3537e-14  on 0  degrees of freedom
## AIC: 47.795
##
## Number of Fisher Scoring iterations: 3
```

```
##
## Call:
## glm(formula = frequency ~ contact * satisfaction, family = poisson(link = log),
##     data = df[df$type == unique(df$type)[2], ])
##
## Deviance Residuals:
## [1]  0  0  0  0  0  0
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    5.25227    0.07236  72.588  < 2e-16 ***
## contactlow                    -0.54274    0.11935  -4.547 5.43e-06 ***
## satisfactionlow               -0.30351    0.11103  -2.734  0.00626 **
## satisfactionmedium            -0.49868    0.11771  -4.236 2.27e-05 ***
## contactlow:satisfactionlow     0.46152    0.17038   2.709  0.00675 **
## contactlow:satisfactionmedium  0.11989    0.18980   0.632  0.52761
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 5.6480e+01  on 5  degrees of freedom
## Residual deviance: 2.4869e-14  on 0  degrees of freedom
## AIC: 51.898
##
## Number of Fisher Scoring iterations: 2
```

```
##
## Call:
```

```
## glm(formula = frequency ~ contact * satisfaction, family = poisson(link = log),
##     data = df[df$type == unique(df$type)[3], ])
##
## Deviance Residuals:
## [1]  0  0  0  0  0  0
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  4.644391   0.098058  47.364  < 2e-16 ***
## contactlow                  -0.517257   0.160451  -3.224  0.00127 **
## satisfactionlow              0.223144   0.131559   1.696  0.08986 .
## satisfactionmedium           0.009569   0.138344   0.069  0.94485
## contactlow:satisfactionlow  -0.145585   0.219914  -0.662  0.50796
## contactlow:satisfactionmedium -0.265503   0.236858  -1.121  0.26231
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance:  5.8866e+01  on 5  degrees of freedom
## Residual deviance: -1.1546e-14  on 0  degrees of freedom
## AIC: 49.409
##
## Number of Fisher Scoring iterations: 2
```

(b)
```
poisson.reg <- glm(frequency ~ type+contact*satisfaction, family=poisson(link=log), data=df)
summary(poisson.reg)
```

```
##
## Call:
## glm(formula = frequency ~ type + contact * satisfaction, family = poisson(link = log),
##     data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.0586  -1.4587  -0.1792   0.9363   4.0247
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    5.19162    0.05696  91.152  < 2e-16 ***
## typehouse                     -0.39377    0.05697  -6.912 4.77e-12 ***
## typetower block               -0.64841    0.06170 -10.509  < 2e-16 ***
## contactlow                    -0.36941    0.07871  -4.694 2.68e-06 ***
## satisfactionlow               -0.25857    0.07623  -3.392 0.000693 ***
## satisfactionmedium            -0.38790    0.07914  -4.901 9.51e-07 ***
## contactlow:satisfactionlow     0.21745    0.11528   1.886 0.059268 .
## contactlow:satisfactionmedium -0.03979    0.12468  -0.319 0.749617
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
```

```
##     Null deviance: 294.477  on 17  degrees of freedom
## Residual deviance:  84.222  on 10  degrees of freedom
## AIC: 213.32
##
## Number of Fisher Scoring iterations: 4
```

(c)
```
reduced <- glm(frequency ~ type+contact+satisfaction, family=poisson(link=log), data=df)
G <- reduced$deviance - poisson.reg$deviance
c("Chi-square for LRT"=G, "p-value"=1-pchisq(G, 2)) # df = 6-1
```

```
## Chi-square for LRT            p-value
##         5.12581817         0.07708018
```

The p-value for the LRT is $0.077 > 0.05$, so we fail to reject the null hypothesis, and remove the interaction term between contact and satisfaction. The result is different from what we obtained using ordinal logistic regression. But the p-values of two cases do not differ a lot, 0.077 and 0.045. So to some extent, they are closed to each other.