Stats 762 Assignment 1

Stephen Wang, ID: 173417367

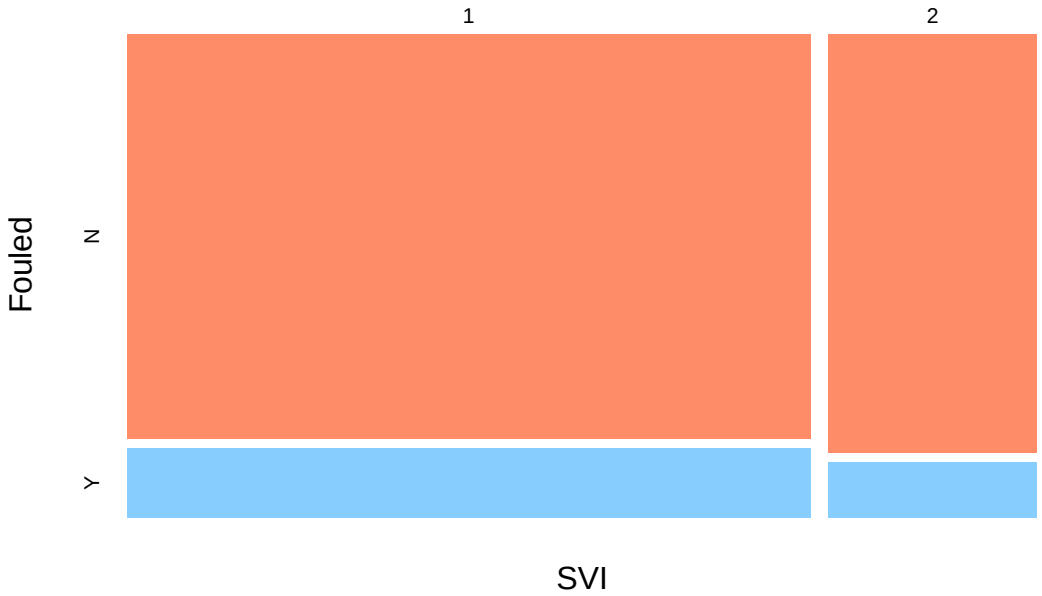
Due: 18 March 2021

Question 1

```
sludge.df<-read.table("sludge.data",header=T)
sludge.df[,1:3]<-lapply(sludge.df[,1:3], as.factor)
sludge.df$fouled<-factor(sludge.df$fouled)
```

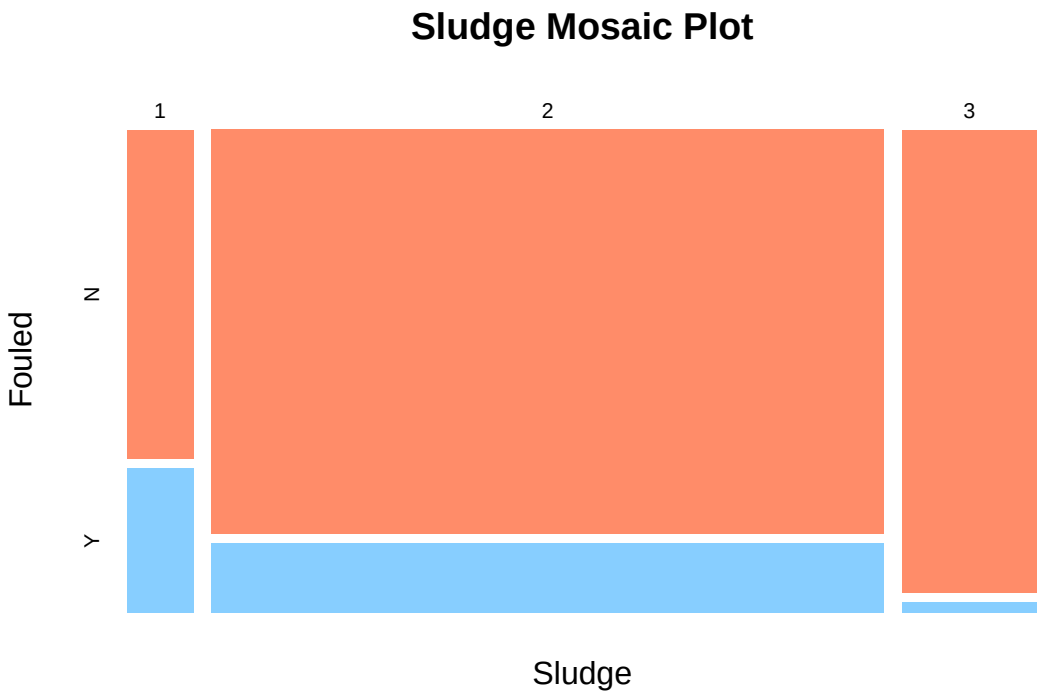
Question 2

```
mosaicplot(~SVI+fouled, data=sludge.df,
           main="SVI Mosaic Plot",
           ylab="Fouled",
           xlab="SVI",
           color=c("salmon1","skyblue1"),
           border=FALSE)
```

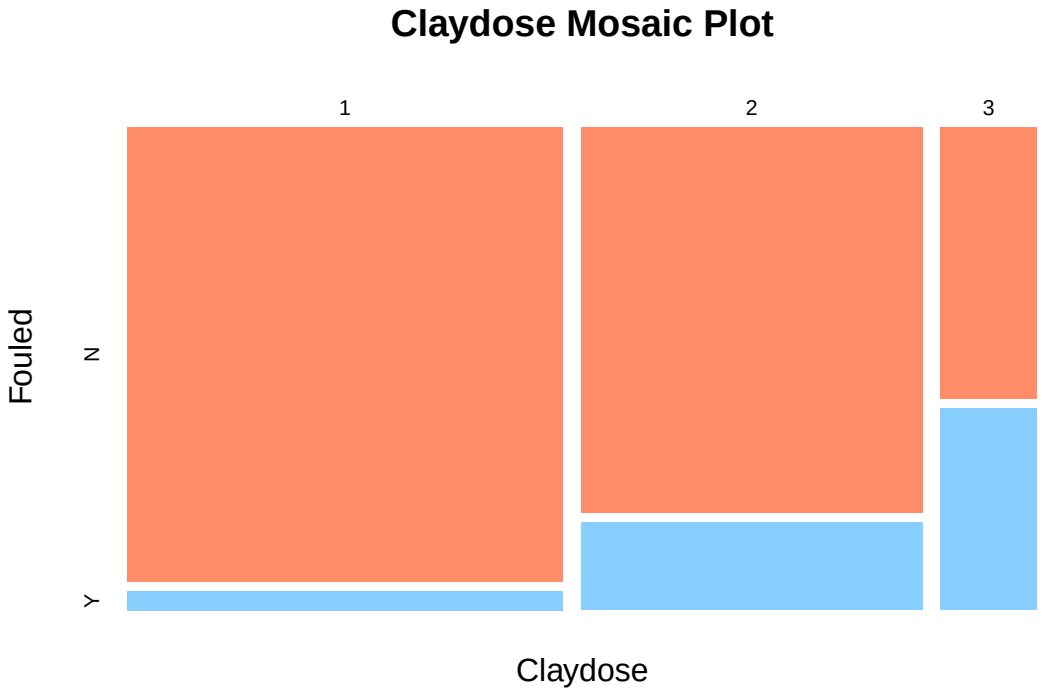



```
mosaicplot(~sludgefd+fouled, data=sludge.df,
           main="Sludge Mosaic Plot",
           ylab="Fouled",
           xlab="Sludge",
           color=c("salmon1","skyblue1"),
```

```
          border=FALSE)
```




```
mosaicplot(~claydose+fouled, data=sludge.df,
           main="Claydose Mosaic Plot",
           ylab="Fouled",
           xlab="Claydose",
           color=c("salmon1","skyblue1"),
           border=FALSE)
```



- The proportion of fouling is similar for sludge volume indexes <60ml/g and >60ml/g.

- The addition of flotation sludge reduces the proportion of fouling compared to no addition of floating sludge. However, the addition of polyaluminumchloride greatly reduces the proportion of fouling compared to no addition of flotation sludge and compared to addition of flotation sludge. - As the level of clay dosing increases from low to medium to high, the proportion of fouling increases respectively.

Question 3

```
sludge.table<-with(sludge.df, table(claydose,fouled)[,2:1])
sludge.table
```

```
##         fouled
## claydose   Y   N
##        1  35 827
##        2 125 550
##        3  81 109
```

```
probability_high_clay<-sludge.table[3,'Y']/(sludge.table[3,'Y']+sludge.table[3,'N'])
probability_high_clay
```

```
## [1] 0.4263158
```

```
odds_high_clay<-probability_high_clay/(sludge.table[3,'N']/(sludge.table[3,'Y']+sludge.table[3,'N']))
odds_high_clay
```

```
## [1] 0.7431193
```

```
odds_low_clay<-(sludge.table[1,'Y']/(sludge.table[1,'Y']+sludge.table[1,'N']))/(sludge.table[1,'N']/(slu
odds_ratio_low_high<-odds_high_clay/odds_low_clay
odds_ratio_low_high
```

```
## [1] 17.55885
```

Question 4

```
## Probability of fouling~claydose.
sludge1.glm<-glm(fouled~claydose, data=sludge.df, family="binomial")
summary(sludge1.glm)
```

```
##
## Call:
## glm(formula = fouled ~ claydose, family = "binomial", data = sludge.df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0542  -0.6400  -0.2879  -0.2879   2.5314
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -3.1625     0.1725 -18.330   <2e-16 ***
## claydose2     1.6809     0.1990   8.448   <2e-16 ***
## claydose3     2.8656     0.2265  12.653   <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: 1395.9  on 1726  degrees of freedom
## Residual deviance: 1199.0  on 1724  degrees of freedom
## AIC: 1205
##
## Number of Fisher Scoring iterations: 5
```

(a) We could either use the inverse link function formula or the predict function to obtain the probability of fouling.

```
## Inverse link function formula:
exp(-3.1625+2.8656)/(1+exp(-3.1625+2.8656))
```

```
## [1] 0.4263155
```

```
## Predict:
predict(sludge1.glm, data.frame(claydose="3"), type="response", se.fit=TRUE)
```

```
## $fit
##         1
## 0.4263158
##
## $se.fit
##          1
## 0.03587776
##
## $residual.scale
## [1] 1
```

(b/c) We use the following to obtain the odd ratios.

```
exp(cbind(coef(sludge1.glm), confint(sludge1.glm)))
```

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

```
##                                2.5 %     97.5 %
## (Intercept)  0.04232165  0.02962032  0.0583814
## claydose2    5.37012920  3.67645134  8.0387137
## claydose3   17.55884447 11.35871048 27.6567474
```

Since 1 is the base category for claydose level, we can look at the intercept to obtain the odds of fouling when level of dosing is low (0.0423)

And derive the odds of fouling when level of claydose is high compared to the odds of fouling when the level of claydose is low (17.55) by looking at claydose3 intercept.

Question 5

```
sludge2.glm<-glm(fouled~SVI, data=sludge.df, family="binomial")
summary(sludge2.glm)
```

```
##
## Call:
## glm(formula = fouled ~ SVI, family = "binomial", data = sludge.df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.5629  -0.5629  -0.5629  -0.4980   2.0731
##
```

```
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.76211    0.07771 -22.674   <2e-16 ***
## SVI2        -0.26267    0.17356  -1.513     0.13
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1395.9  on 1726  degrees of freedom
## Residual deviance: 1393.5  on 1725  degrees of freedom
## AIC: 1397.5
##
## Number of Fisher Scoring iterations: 4
```

```
sludge3.glm<-glm(fouled~SVI+sludgefd+claydose, data=sludge.df, family="binomial")
summary(sludge3.glm)
```

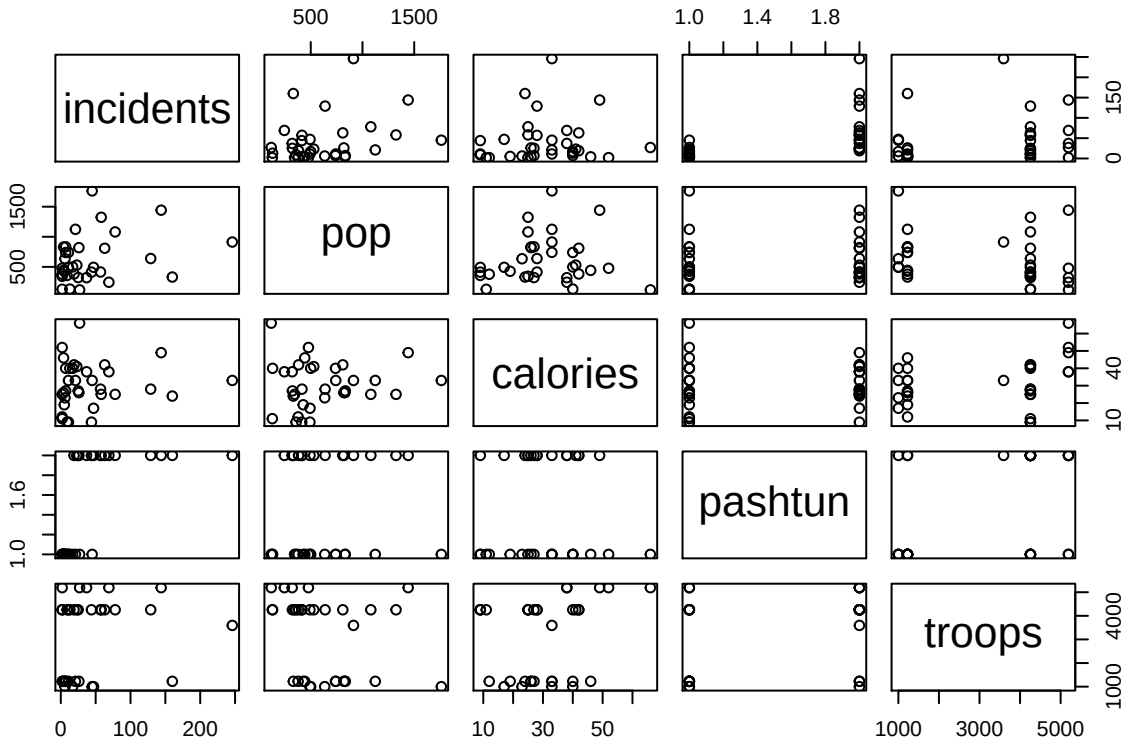
```
##
## Call:
## glm(formula = fouled ~ SVI + sludgefd + claydose, family = "binomial",
##     data = sludge.df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9121  -0.5374  -0.2952  -0.1629   2.9424
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.8622     0.2489  -7.481 7.39e-14 ***
## SVI2         -1.5449     0.2066  -7.477 7.62e-14 ***
## sludgefd2    -1.2491     0.2387  -5.234 1.66e-07 ***
## sludgefd3    -2.4532     0.4741  -5.174 2.29e-07 ***
## claydose2     1.9862     0.2120   9.369  < 2e-16 ***
## claydose3     3.5149     0.2644  13.292  < 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: 1395.9  on 1726  degrees of freedom
## Residual deviance: 1094.2  on 1721  degrees of freedom
## AIC: 1106.2
##
## Number of Fisher Scoring iterations: 6
```

When we fit the GLM to explain fouling with only one explanatory variable SVI, there is weak statistical significance (p-value=0.13) for when SVI >60ml/g. This suggests that the level of SVI doesn't have any significant influence on fouling which we observed from the mosaic plots.

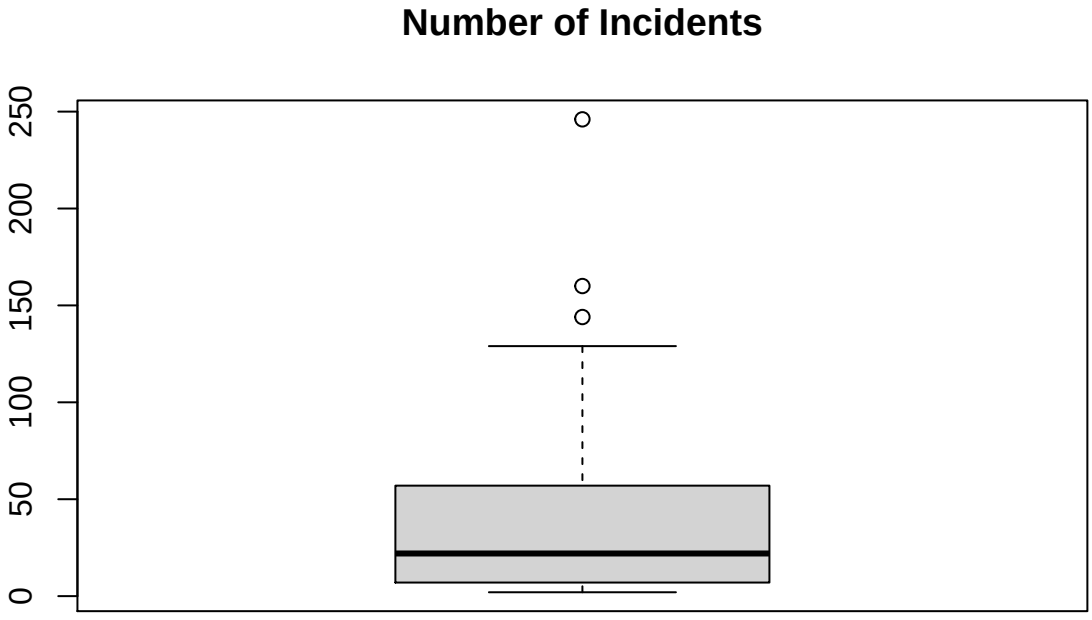
However, once we add the other explanatory variables into the model, SVI becomes statistically significant (p-value<0.05) which indicates a correlation between SVI and the newly added regressors. Therefore, only when we take into account the other explanatory variables, will SVI be relevant in predicting fouling.

Question 6

```
afghan.df = read.table("afghan.data", row.names=1, header=T)
afghan.df$pashtun<-factor(afghan.df$pashtun)
plot(afghan.df)
```




```
boxplot(afghan.df$incidents,
        main="Number of Incidents")
```




```
afghan.df[afghan.df$incidents>140,]
```

```
##          incidents    pop calories pashtun troops
```

```
## Helmand        144 1441.8       49       1 5193.7
## Kabul          160  331.4       24       1 1225.0
## Kandahar       246  913.0       33       1 3593.7
```

Although there are three events (Helmand, Kabul and Kandahar) where there is seemingly higher incident count, we assume these are valid observations (due to unlikeliness of miscounting terror incidents) and don't have reason to remove these points. From the pairs plot, we can briefly observe the following: - The majority of incidents are within the 0-60 count range. - There is a higher number of incidents when observing regions with Pushtun majority. - There are loose clusters when observing number of incidents to population and number of incidents to calories. - There doesn't seem to be any distinguishable relationship between number of incidents and troops.

Question 7

```
model1<-glm(incidents~calories+pashtun+troops+offset(log(pop/1000)), family=poisson, data=afghan.df)
summary(model1)
```

```
##
## Call:
## glm(formula = incidents ~ calories + pashtun + troops + offset(log(pop/1000)),
##     family = poisson, data = afghan.df)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -9.688  -2.447  -1.280   1.621  13.460
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.810e+00  1.075e-01  26.136  < 2e-16 ***
## calories     7.825e-03  2.939e-03   2.663  0.00776 **
## pashtun1     1.978e+00  9.108e-02  21.722  < 2e-16 ***
## troops      -7.204e-05  2.323e-05  -3.101  0.00193 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1661.64  on 33  degrees of freedom
## Residual deviance:  819.54  on 30  degrees of freedom
## AIC: 991.12
##
## Number of Fisher Scoring iterations: 5
```

Question 8

```
model2<-glm(incidents~pashtun+troops+calories+offset(log(pop/1000)), family=quasipoisson, data=afghan.d
summary(model2)
```

```
##
## Call:
## glm(formula = incidents ~ pashtun + troops + calories + offset(log(pop/1000)),
##     family = quasipoisson, data = afghan.df)
##
## Deviance Residuals:
```

```
##    Min      1Q  Median      3Q     Max
## -9.688  -2.447  -1.280   1.621  13.460
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.810e+00  6.691e-01   4.200  0.00022 ***
## pashtun1     1.978e+00  5.667e-01   3.491  0.00151 **
## troops      -7.204e-05  1.445e-04  -0.498  0.62185
## calories     7.825e-03  1.829e-02   0.428  0.67180
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 38.72083)
##
##     Null deviance: 1661.64  on 33  degrees of freedom
## Residual deviance:  819.54  on 30  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

There's weak statistical significance for two of the explanatory variables, calories and troops, so we can re-fit the model.

```
model2refitted.glm<-glm(incidents~pashtun+offset(log(pop/1000)), family=quasipoisson, data=afghan.df)
summary(model2refitted.glm)
```

```
##
## Call:
## glm(formula = incidents ~ pashtun + offset(log(pop/1000)), family = quasipoisson,
##     data = afghan.df)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -8.929  -2.386  -1.286   1.103  14.539
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.9252     0.4425   6.610 1.88e-07 ***
## pashtun1      1.8359     0.4765   3.853 0.000528 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 38.1884)
##
##     Null deviance: 1661.64  on 33  degrees of freedom
## Residual deviance:  831.14  on 32  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

```
anova(model2refitted.glm, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: quasipoisson, link: log
##
```

```
## Response: incidents
##
## Terms added sequentially (first to last)
##
##
##         Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                       33    1661.64
## pashtun  1    830.5        32     831.14 3.11e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

By looking at the summary, we can check the dispersion parameter. The dispersion parameter for both Quasipoisson models is estimated to be >38 which indicates how much larger the variance is than the mean. Since this dispersion parameter is significantly larger than 1, the Poisson distribution has serious overdispersion.

Question 9

```
model3<-glm.nb(incidents~calories+pashtun+troops+offset(log(pop/1000)), data=afghan.df)
summary(model3)
```

```
##
## Call:
## glm.nb(formula = incidents ~ calories + pashtun + troops + offset(log(pop/1000)),
##     data = afghan.df, init.theta = 1.590278141, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9969  -1.0560  -0.4750   0.3728   2.6027
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) 2.099e+00  4.273e-01   4.912 9.01e-07 ***
## calories    2.566e-02  1.176e-02   2.181   0.0292 *
## pashtun1    1.806e+00  3.194e-01   5.656 1.55e-08 ***
## troops      8.087e-05  9.911e-05   0.816   0.4145
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(1.5903) family taken to be 1)
##
##     Null deviance: 75.425  on 33  degrees of freedom
## Residual deviance: 36.312  on 30  degrees of freedom
## AIC: 302.88
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  1.590
##           Std. Err.:  0.388
##
##  2 x log-likelihood:  -292.878
```

The fitted model suggests weak statistical significance (p-value=0.4145) for troops to explain number of incidents so we will fit a new model without troops.

```r
model4<-glm.nb(incidents~calories+pashtun+offset(log(pop/1000)), data=afghan.df)
summary(model4)
```

```
##
## Call:
## glm.nb(formula = incidents ~ calories + pashtun + offset(log(pop/1000)),
##     data = afghan.df, init.theta = 1.572865484, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9056  -1.0537  -0.5164   0.3613   2.2633
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.23935    0.41022   5.459 4.79e-08 ***
## calories     0.02871    0.01141   2.516   0.0119 *
## pashtun1     1.84519    0.28737   6.421 1.35e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(1.5729) family taken to be 1)
##
##     Null deviance: 74.663  on 33  degrees of freedom
## Residual deviance: 36.722  on 31  degrees of freedom
## AIC: 301.65
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  1.573
##           Std. Err.:  0.386
##
##  2 x log-likelihood:  -293.651
```

```r
anova(model4, test="Chisq")
```

```
## Warning in anova.negbin(model4, test = "Chisq"): tests made without re-
## estimating 'theta'
```

```
## Analysis of Deviance Table
##
## Model: Negative Binomial(1.5729), link: log
##
## Response: incidents
##
## Terms added sequentially (first to last)
##
##
##          Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                        33     74.663
## calories  1    1.720        32     72.943    0.1897
## pashtun   1   36.221        31     36.722 1.762e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Estimated coefficients
coef(model1)
```

```
##   (Intercept)      calories      pashtun1        troops
##  2.810410e+00  7.824696e-03  1.978416e+00 -7.203525e-05
```

```
coef(model4)
```

```
## (Intercept)    calories    pashtun1
##  2.23935212  0.02870995  1.84518942
```

```
## Estimated standard errors
sqrt(diag(vcov(model1)))
```

```
##  (Intercept)     calories     pashtun1       troops
## 1.075285e-01 2.938839e-03 9.107878e-02 2.322795e-05
```

```
sqrt(diag(vcov(model4)))
```

```
## (Intercept)    calories    pashtun1
##  0.41021652  0.01141011  0.28736923
```

When we compare the Poisson distribution to the Negative Binomial distribution, we can see that the coefficient values of the regressors in the Negative Binomial model have deflated. On the other hand, we've had an inverse effect with the standard error of the coefficients where the Negative Binomial distribution have inflated the standard errors compared to the Poisson distribution.

Question 10

The number of terror incidents in provinces that have Pushtun ethnic majority are likely to have 1.8 more incidents per 1000 population compared to provinces without Pushtun ethnic majority.

Question 11

```
farah=data.frame(pop=493,calories=17,pashtun="1",troops=1000)
predict(model4, farah, type="response")
```

```
##        1
## 47.72067
```

```
## Obtain theta from the model summary
theta<-1.573

## Calculate variance
nb_mean<-predict(model4, farah, type="response")
nb_variance<-nb_mean+(nb_mean^2/theta)
```

We obtain the theta value of 1.573 from the Negative Binomial model summary. For the Poisson distribution, the variance is assumed to be equal to the mean therefore, if the Poisson distribution has the same expected value of 47.7, then the variance is also expected to be 47.7.

On the other hand, the Negative Binomial distribution has one more parameter to adjust the variance independently from the mean and therefore has a variance greater than the mean. From our calculations, we can determine that the Negative Binomial distribution has a variance of 1480 which is 31x larger than the Poisson distribution.