R-code—Module-2.R

dland

2022-04-27

```r
### Module 2 - moped data set

## command to indicate where to read the data file and store plots on your computer - please modify the pat
h based on your own needs

setwd('C:/Users/dland/OneDrive - University of Waterloo/Desktop/ACTSC 632 - S22/R code used in class/Module
2')

## to read the moped insurance data and visualize the data

moped<-read.csv("moped.csv",header=TRUE,sep=',')
head(moped)
```

```
##   class age zone duration severity number pure actual  frequency
## 1     1   1    1     62.9    18256     17 4936   2049 0.27027027
## 2     1   1    2    112.9    13632      7  845   1230 0.06200177
## 3     1   1    3    133.1    20877      9 1411    762 0.06761833
## 4     1   1    4    376.6    13045      7  242    396 0.01858736
## 5     1   1    5      9.4        0      0    0    990 0.00000000
## 6     1   1    6     70.8    15000      1  212    594 0.01412429
```

```r
moped
```

```
##    class age zone duration severity number pure actual  frequency
## 1      1   1    1     62.9    18256     17 4936   2049 0.27027027
## 2      1   1    2    112.9    13632      7  845   1230 0.06200177
## 3      1   1    3    133.1    20877      9 1411    762 0.06761833
## 4      1   1    4    376.6    13045      7  242    396 0.01858736
## 5      1   1    5      9.4        0      0    0    990 0.00000000
## 6      1   1    6     70.8    15000      1  212    594 0.01412429
## 7      1   1    7      4.4     8018      1 1829    396 0.22727273
## 8      1   2    1    352.1     8232     52 1216   1229 0.14768532
## 9      1   2    2    840.1     7418     69  609    738 0.08213308
## 10     1   2    3   1378.3     7318     75  398    457 0.05441486
## 11     1   2    4   5505.3     6922    136  171    238 0.02470347
## 12     1   2    5    114.1    11131      2  195    594 0.01752848
## 13     1   2    6    810.9     5970     14  103    356 0.01726477
## 14     1   2    7     62.3     6500      1  104    238 0.01605136
## 15     2   1    1    191.6     7754     43 1740   1024 0.22442589
## 16     2   1    2    237.3     6933     34  993    615 0.14327855
## 17     2   1    3    162.4     4402     11  298    381 0.06773399
## 18     2   1    4    446.5     8214      8  147    198 0.01791713
## 19     2   1    5     13.2        0      0    0    495 0.00000000
## 20     2   1    6     82.8     5830      3  211    297 0.03623188
## 21     2   1    7     14.5        0      0    0    198 0.00000000
## 22     2   2    1    844.8     4728     94  526    614 0.11126894
## 23     2   2    2   1296.0     4252     99  325    369 0.07638889
## 24     2   2    3   1214.9     4212     37  128    229 0.03045518
## 25     2   2    4   3740.7     3846     56   58    119 0.01497046
## 26     2   2    5    109.4     3925      4  144    297 0.03656307
## 27     2   2    6    404.7     5280      5   65    178 0.01235483
## 28     2   2    7     66.3     7795      1  118    119 0.01508296
```

```
# Observation: some tariff cells have very low duration and some have no claims over the duration period



## the predictors class, age and zone are categorical variables
## make use of the function "factor" to turn each variable into a categorical variable (otherwise, they are treated as quantitative)

moped <- within(moped, {
    class <- factor(class)
    age <- factor(age)
    zone <- factor(zone)
})



## the function "levels" enumerate the different outcomes/categories of each predictor

levels(moped$class)
```

```
## [1] "1" "2"
```

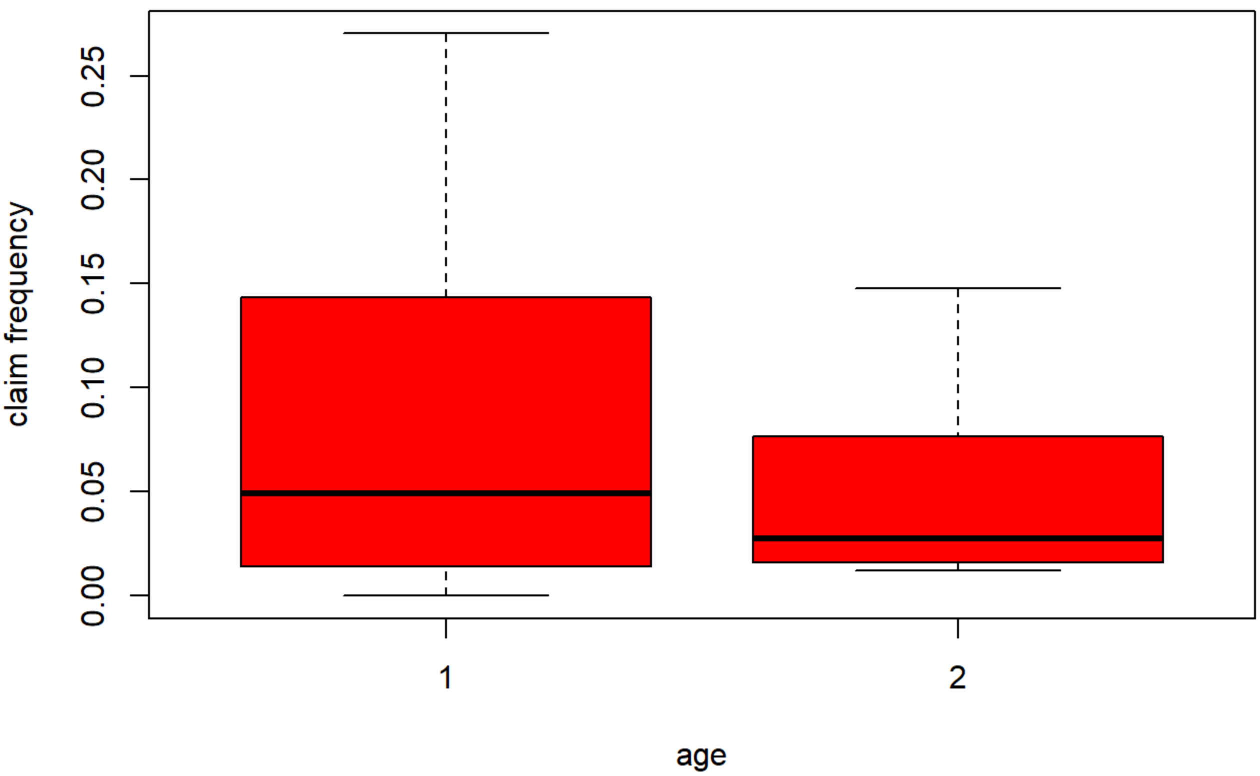
```
levels(moped$age)
```

```
## [1] "1" "2"
```

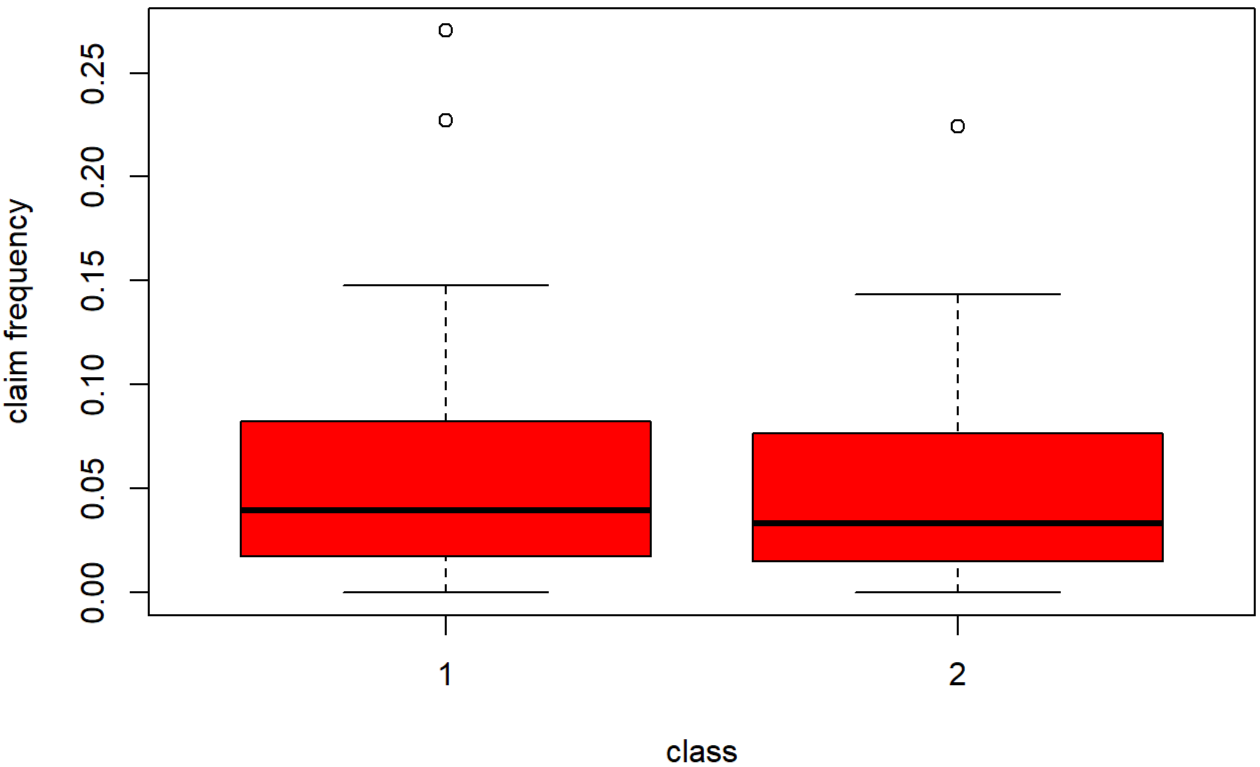
```
levels(moped$zone)
```

```
## [1] "1" "2" "3" "4" "5" "6" "7"
```

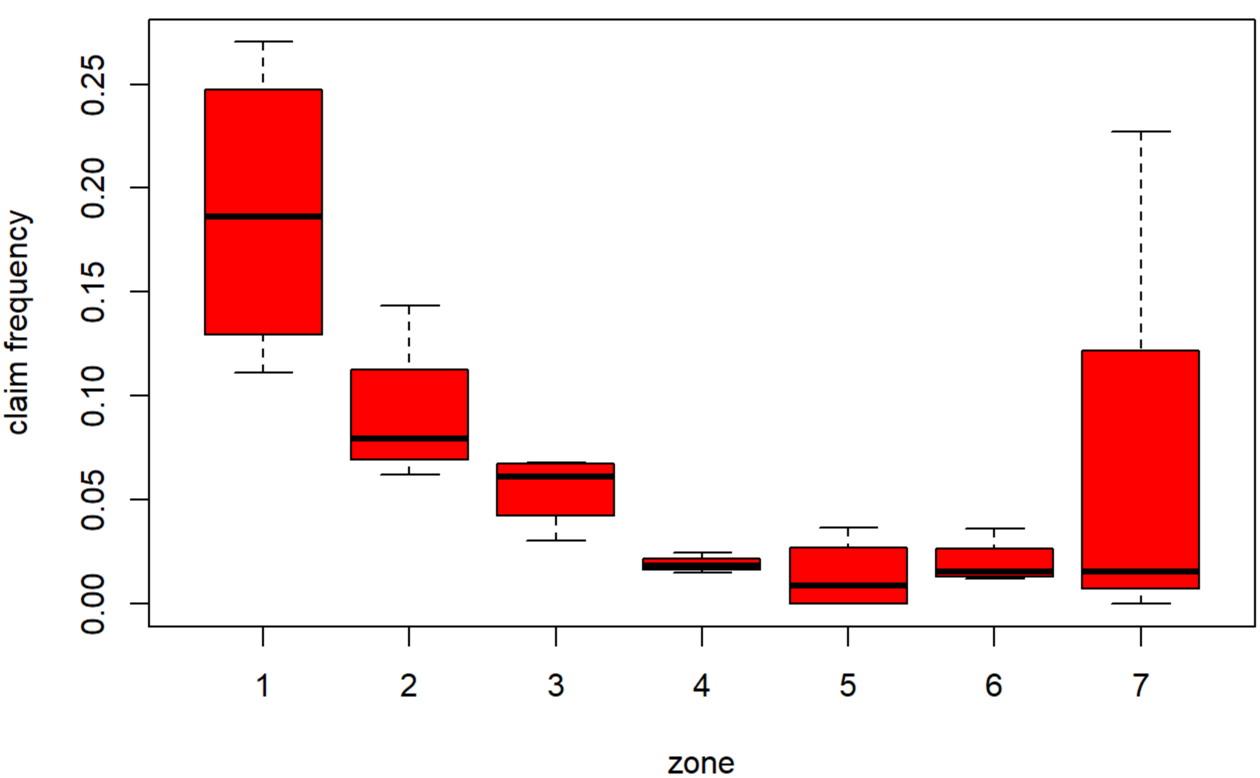
```
# Observation: class and age have only two categories, while zone has a total of 7 categories


## to better understand the relationship between predictors and claim frequency


plot(moped$age, moped$frequency, col="red",xlab="age",ylab="claim frequency")
```




```
plot(moped$class, moped$frequency, col="red",xlab="class",ylab="claim frequency")
```




```
plot(moped$zone, moped$frequency, col="red",xlab="zone",ylab="claim frequency")
```



```
# Observations: claim frequency seems to vary quite a bit between the different "zone" categories, this see
ms to be less pronounced for age and even less so for class


## GLM fitting

# if nothing is done, (1,1,1) is the base tariff cell
# we usually want the base tariff cell to be the one with the largest exposure (e.g., largest duration) so
 we pick tariff cell (1,2,4) to be the base tariff cell
# this is because all tariff cells can be easily compared to the base tariff cell (which preferably should
 be a tariff cell well known by the insurer)
# You can reorder the levels of each categorical variable to achieve this

print(basecell<- moped[which.max(moped[,4]),1:3])
```

```
##    class age zone
## 11     1   2    4
```

```
print(moped$class<- relevel(moped$class, as.character(basecell$class)))
```

```
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## Levels: 1 2
```

```
print(moped$age<- relevel(moped$age, as.character(basecell$age)))
```

```
##  [1] 1 1 1 1 1 1 1 2 2 2 2 2 2 2 1 1 1 1 1 1 1 2 2 2 2 2 2 2
## Levels: 2 1
```

```
print(moped$zone<- relevel(moped$zone, as.character(basecell$zone)))
```

```
##  [1] 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7
## Levels: 4 1 2 3 5 6 7
```

```
# fit relative Poisson glm (with phi=1) for nb of claims that uses an offset

summary(freq<-glm(number ~ class + age + zone + offset(log(duration)), data = moped[moped$duration>0,], fam
ily=poisson("log")))
```

```
##
## Call:
## glm(formula = number ~ class + age + zone + offset(log(duration)),
##     family = poisson("log"), data = moped[moped$duration > 0,
##         ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5001  -0.8712  -0.3153   0.8260   1.5251
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.829639   0.074997 -51.064  < 2e-16 ***
## class2      -0.252640   0.073777  -3.424 0.000616 ***
## age1         0.437661   0.093954   4.658 3.19e-06 ***
## zone1        1.959875   0.101451  19.319  < 2e-16 ***
## zone2        1.428190   0.099375  14.372  < 2e-16 ***
## zone3        0.802747   0.111493   7.200 6.02e-13 ***
## zone5        0.185408   0.414164   0.448 0.654393
## zone6       -0.231218   0.219861  -1.052 0.292958
## zone7        0.000554   0.581627   0.001 0.999240
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 520.352  on 27  degrees of freedom
## Residual deviance:  30.077  on 19  degrees of freedom
## AIC: 157.34
##
## Number of Fisher Scoring iterations: 5
```

```
# IF the glm function could admit "relative.poisson" as a family (which is not the case), this is how we wo
uld code it
# summary(freq <- glm(frequency ~ class + age + zone, data = moped[moped$duration > 0, ], family = relativ
e.poisson("log"), weights = duration))


# fits a gamma glm on claim severity, using only the classes that have more than one claim

summary(sev <- glm(severity ~ class + age + zone, data = moped[moped$number > 0, ], family = Gamma("log"),
 weights = number))
```

```
##
## Call:
## glm(formula = severity ~ class + age + zone, family = Gamma("log"),
##     data = moped[moped$number > 0, ], weights = number)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.55662  -0.25644   0.01745   0.32310   1.42683
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.85756    0.05301 167.089  < 2e-16 ***
## class2      -0.60677    0.05494 -11.044 6.78e-09 ***
## age1         0.58397    0.06943   8.411 2.88e-07 ***
## zone1        0.19400    0.07472   2.596   0.0195 *
## zone2        0.07206    0.07328   0.983   0.3401
## zone3        0.06416    0.08066   0.795   0.4380
## zone5        0.19151    0.29983   0.639   0.5320
## zone6       -0.02100    0.15890  -0.132   0.8965
## zone7        0.18126    0.42039   0.431   0.6721
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.521651)
##
##     Null deviance: 109.7707  on 24  degrees of freedom
## Residual deviance:   7.9998  on 16  degrees of freedom
## AIC: 12377
##
## Number of Fisher Scoring iterations: 5
```

```
# Observation: Check beta coefficients obtained via MLE under the column "Estimate" of the output



############### everything above covers material up to and including Section 2.6 in the lecture notes




## Deviance

# Frequency model: fit of the relative Poisson model
# Use the residual deviance statistic
# This computes the p-value corresponding to the residual deviance provided in R
# The deviance test indicates there is not enough evidence to reject the fitted model at a 95% confidence l
evel (just barely though) as the p-value is slightly superior to 5%

cbind(scaled.deviance=freq$deviance,df=freq$df.residual, p=1-pchisq(freq$deviance, freq$df.residual))
```

```
##      scaled.deviance df          p
## [1,]        30.07667 19 0.05083071
```

```
# Gamma severity model
# Need to compute the scaled deviance by first extracting the phi parameter
# seems to indicate a good fit using the deviance statistic as the p-value is slightly above 50%

sev.phi<-summary(sev)$dispersion
cbind(scaled.deviance = sev$deviance/sev.phi, df = sev$df.residual, p = 1-pchisq(sev$deviance/sev.phi, sev
$df.residual))
```

```
##      scaled.deviance df         p
## [1,]        15.33558 16 0.5002111
```

```
## Pearson's goodness of fit

# Frequency model: fit of the relative Poisson model
# This time, the goodness of fit test rejects the null hypothesis that the relative Poisson provides a good
fit as the p-value of the test is < 5%

chifreq<-sum(residuals(freq,type="pearson")^2)
cbind(scaled.pearson = chifreq, df = freq$df.residual, p = 1-pchisq(chifreq, freq$df.residual))
```

```
##      scaled.pearson df          p
## [1,]        30.3629 19 0.04735772
```

```
# Severity model: fit of the gamma model
# The goodness of fit test is consistent with the conclusion we reached with the deviance test. No evidence
to reject the fitted gamma model

chisev<-sum(residuals(sev,type="pearson")^2)
cbind(scaled.pearson = chisev/sev.phi, df = sev$df.residual, p = 1-pchisq(chisev/sev.phi, sev$df.residual))
```

```
##      scaled.pearson df         p
## [1,]             16 16 0.4529608
```

```
## Estimation of phi for the gamma severity model
# Check to verify that R uses Method 1 in the notes to estimate phi

chisev/sev$df.residual
```

```
## [1] 0.521651
```

```
print(sev.phi)
```

```
## [1] 0.521651
```

```
## Hierarchical model

summary(freq)
```

```
##
## Call:
## glm(formula = number ~ class + age + zone + offset(log(duration)),
##     family = poisson("log"), data = moped[moped$duration > 0,
##         ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5001  -0.8712  -0.3153   0.8260   1.5251
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.829639   0.074997 -51.064  < 2e-16 ***
## class2      -0.252640   0.073777  -3.424 0.000616 ***
## age1         0.437661   0.093954   4.658 3.19e-06 ***
## zone1        1.959875   0.101451  19.319  < 2e-16 ***
## zone2        1.428190   0.099375  14.372  < 2e-16 ***
## zone3        0.802747   0.111493   7.200 6.02e-13 ***
## zone5        0.185408   0.414164   0.448 0.654393
## zone6       -0.231218   0.219861  -1.052 0.292958
## zone7        0.000554   0.581627   0.001 0.999240
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 520.352  on 27  degrees of freedom
## Residual deviance:  30.077  on 19  degrees of freedom
## AIC: 157.34
##
## Number of Fisher Scoring iterations: 5
```

```
# from the fitted frequency model, we see that zones 5, 6 and 7 do not seem to be statistically different t
han zone 4 so we want to combine all 4 zones into one using the function "recode" in R under the library "d
plyr"
install.packages("dplyr")
```

```
## Warning: package 'dplyr' is in use and will not be installed
```

```
library(dplyr)

levels(moped$zone)<-recode(levels(moped$zone), "4"="4+")
levels(moped$zone)<-recode(levels(moped$zone), "5"="4+")
levels(moped$zone)<-recode(levels(moped$zone), "6"="4+")
levels(moped$zone)<-recode(levels(moped$zone), "7"="4+")

# run the simplified poisson glm, fit is improved relative to the number of parameters
# the residual deviance statistic improves

summary(freq.new<-glm(number ~ class + age + zone + offset(log(duration)), data = moped[moped$duration>0,],
family=poisson("log")))
```

```
##
## Call:
## glm(formula = number ~ class + age + zone + offset(log(duration)),
##     family = poisson("log"), data = moped[moped$duration > 0,
##         ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4906  -0.8207  -0.3083   0.6254   1.7057
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.85073    0.07056 -54.575  < 2e-16 ***
## class2      -0.25053    0.07377  -3.396 0.000683 ***
## age1         0.43581    0.09393   4.640 3.49e-06 ***
## zone1        1.98002    0.09820  20.164  < 2e-16 ***
## zone2        1.44847    0.09606  15.079  < 2e-16 ***
## zone3        0.82323    0.10855   7.584 3.35e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 520.352  on 27  degrees of freedom
## Residual deviance:  31.498  on 22  degrees of freedom
## AIC: 152.76
##
## Number of Fisher Scoring iterations: 4
```

```
cbind(scaled.deviance = freq.new$deviance, df = freq.new$df.residual, p = 1-pchisq(freq.new$deviance, freq.
new$df.residual))
```

```
##      scaled.deviance df          p
## [1,]        31.49797 22 0.08636702
```

```
# show that we are statistically justified to choose the simplified model (over the more complicated model)

cbind(diff.scaled.deviance=freq.new$deviance-freq$deviance,df=freq.new$df.residual-freq$df.residual,p = 1-p
chisq(freq.new$deviance-freq$deviance, freq.new$df.residual-freq$df.residual))
```

```
##      diff.scaled.deviance df         p
## [1,]             1.421295  3 0.7005506
```

```
# or equivalently

anova(freq.new,freq)
```

```
## Analysis of Deviance Table
##
## Model 1: number ~ class + age + zone + offset(log(duration))
## Model 2: number ~ class + age + zone + offset(log(duration))
##   Resid. Df Resid. Dev Df Deviance
## 1        22     31.498
## 2        19     30.077  3   1.4213
```

```
# with p-value for the test of

1-pchisq(anova(freq.new,freq)[2,]$Deviance,anova(freq.new,freq)[2,]$Df)
```

```
## [1] 0.7005506
```

```
# Move on to severity

summary(sev)
```

```
##
## Call:
## glm(formula = severity ~ class + age + zone, family = Gamma("log"),
##     data = moped[moped$number > 0, ], weights = number)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.55662  -0.25644   0.01745   0.32310   1.42683
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.85756    0.05301 167.089  < 2e-16 ***
## class2      -0.60677    0.05494 -11.044 6.78e-09 ***
## age1         0.58397    0.06943   8.411 2.88e-07 ***
## zone1        0.19400    0.07472   2.596   0.0195 *
## zone2        0.07206    0.07328   0.983   0.3401
## zone3        0.06416    0.08066   0.795   0.4380
## zone5        0.19151    0.29983   0.639   0.5320
## zone6       -0.02100    0.15890  -0.132   0.8965
## zone7        0.18126    0.42039   0.431   0.6721
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.521651)
##
##     Null deviance: 109.7707  on 24  degrees of freedom
## Residual deviance:   7.9998  on 16  degrees of freedom
## AIC: 12377
##
## Number of Fisher Scoring iterations: 5
```

```
## see if we can drop zone from the severity analysis - the result shows that we are justified to do so as
 the p-value of the test is 25.4%

summary(sev.new <- glm(severity ~ class + age, family = Gamma("log"), data = moped[moped$number > 0, ], wei
ghts = number))
```

```
##
## Call:
## glm(formula = severity ~ class + age, family = Gamma("log"),
##     data = moped[moped$number > 0, ], weights = number)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7323  -0.4808  -0.0401   0.5229   1.3237
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.92016    0.03851 231.651  < 2e-16 ***
## class2      -0.57354    0.05421 -10.581 4.28e-10 ***
## age1         0.61521    0.07064   8.709 1.40e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.5572562)
##
##     Null deviance: 109.771  on 24  degrees of freedom
## Residual deviance:  12.063  on 22  degrees of freedom
## AIC: 12688
##
## Number of Fisher Scoring iterations: 5
```

```
cbind(scaled.deviance = sev.new$deviance/sev.phi, df = sev.new$df.residual, p = 1-pchisq(sev.new$deviance/s
ev.phi, sev.new$df.residual))
```

```
##      scaled.deviance df         p
## [1,]        23.12493 22 0.3947041
```

```
cbind(diff.deviance=(sev.new$deviance-sev$deviance)/sev.phi,df=sev.new$df.residual-sev$df.residual,p = 1-pc
hisq((sev.new$deviance-sev$deviance)/sev.phi, sev.new$df.residual-sev$df.residual))
```

```
##      diff.deviance df         p
## [1,]      7.789352  6 0.2539457
```

```
# Equivalently

anova(sev.new,sev)
```

```
## Analysis of Deviance Table
##
## Model 1: severity ~ class + age
## Model 2: severity ~ class + age + zone
##   Resid. Df Resid. Dev Df Deviance
## 1        22    12.0631
## 2        16     7.9998  6   4.0633
```

```
# with p-value for the test of

1-pchisq(anova(sev.new,sev)[2,]$Deviance/sev.phi,anova(sev.new,sev)[2,]$Df)
```

```
## [1] 0.2539457
```

```
## Variance covariance matrix of the beta coefficients
# use the vcov function to get the scaled variance-covariance matrix

vcov(freq.new)
```

```
##               (Intercept)        class2          age1         zone1         zone2        zone3
## (Intercept)  0.004978527 -0.0018970548 -0.0009227370 -0.0034926230 -0.0037312028 -0.004048321
## class2      -0.001897055  0.0054417451 -0.0004547458 -0.0016050222 -0.0010628013 -0.000314583
## age1        -0.000922737 -0.0004547458  0.0088222863 -0.0009378107 -0.0005805306 -0.000185831
## zone1       -0.003492623 -0.0016050222 -0.0009378107  0.0096426662  0.0045792465  0.004302768
## zone2       -0.003731203 -0.0010628013 -0.0005805306  0.0045792465  0.0092273570  0.004261701
## zone3       -0.004048321 -0.0003145830 -0.0001858310  0.0043027676  0.0042617006  0.011783160
```

```
vcov(sev.new)
```

```
##               (Intercept)       class2          age1
## (Intercept)  0.0014827836 -0.001348443 -0.0005360043
## class2      -0.0013484433  0.002938331 -0.0007146400
## age1        -0.0005360043 -0.000714640  0.0049899445
```

```
# to produce a R markdown file

#install.packages("rmarkdown")
#library(rmarkdown)
#install.packages("installr")
#library(installr)
#installr::install.pandoc()
#render("R code - Module 2.R")
```