# ActSc 632 Moped Example

Spring 2023

Department of Statistics and Actuarial Science, University of Waterloo

# Load and Visualizing Data

```
library(insuranceData)
library(foreach)
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(tibble)

options(tibble.print_max = Inf)

data(dataOhlsson)
moped <- read.csv("/Users/xintong0079/Documents/GitHub/ActSc632/lectures/example/moped.csv", header=TRUE, sep
=',')
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
```

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

levels(moped$class)
```

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

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

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

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

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

There are 3 levels for class, 3 levels for age, and 7 levels for zone.

```
basecell <- moped[which.max(moped[, 4]),1:3]
moped$class <- relevel(moped$class, as.character(basecell$class))
moped$age <- relevel(moped$age, as.character(basecell$age))
moped$zone <- relevel(moped$zone, as.character(basecell$zone))
```

The base cell is usually the cell with the highest duration. We relevel the factors so that the base cell is the reference cell.

## dataOhlsson

1. In dataOhlsson, we will need to following the same steps as above to make the categorical variable into factors.
2. Don't forget to relevel the factors so that the base cell is the reference cell.
3. Check the tariff table on the second page for how the age variable "fordald" is grouped into 2 levels.
4. Check how the "bonuskl" variable is grouped into 3 levels.

# Getting the tariff table

Different from the moped dataset, we will need to combine the information from the dataOhlsson dataset to get the tariff table.

- For each rating factor, sum up the durations, n.claims, and total cost premium.
- Calculate the average claim cost for each rating factor from the total cost premium and n.claims.
- Create binary variables for each rating factor (using contrasts and contract treatment coding).

# Fitting the GLMs

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

```
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
```

What do you think about the model based on the p-value?

## dataOhlsson

In the moped dataset, everything is already in the right format. We will need to reformat the data from the dataOhlsson dataset.

1. Create a new dataframe using “groupby” and “summarize”.
    - “groupby” takes in the rating factors, it groups the data by different values of rating factor combinations.
    - “summarize” takes in the variables that we want to summarize, it calculates the sum of the variables for each group.
        - more specifically, it calculates the sum of the variables for each combination of rating factors (duration, n.claims, and claim cost).
2. Fit the GLM using the new dataframe.
3. Examine the fitted model using deviance statistics and the corresponding p-value.
4. Get the confidence intervals for the coefficients and the tariff cells.

```
P.class <- contrasts(moped$class)
P.age <- contrasts(moped$age)
P.zone <- contrasts(moped$zone)

mat.freq <- summary(freq)$coefficients[, 1:2]

table <- matrix(0, nrow = nlevels(moped$class) + nlevels(moped$age) + nlevels(moped$zone) + 1, ncol = 5)
colnames(table) <- c("Rating Factor", "Level", "Multiplier", "Lower Bound", "Upper Bound")

print(P.class)
```

```
##   2
## 1 0
## 2 1
```

```
print(P.age)
```

```
##   1
## 2 0
## 1 1
```

```
print(P.zone)
```

```
##   1 2 3 5 6 7
## 4 0 0 0 0 0 0
## 1 1 0 0 0 0 0
## 2 0 1 0 0 0 0
## 3 0 0 1 0 0 0
## 5 0 0 0 1 0 0
## 6 0 0 0 0 1 0
## 7 0 0 0 0 0 1
```

You can also rank the classes by duration, the above matrix is the reference matrix to the base cell what we have set previously.

Let's now fill in the table.

### Intercept

```
estimate = mat.freq[1, 1]
std.error = mat.freq[1, 2]
table[1, ] = c("Intercept", "0", exp(estimate), exp(estimate - 1.96 * std.error), exp(estimate + 1.96 * std.erro
r))
```

### Class

```
estimate = P.class %*% mat.freq[2, 1]
std.error = P.class %*% mat.freq[2, 2]

table[2:3, ] = cbind(rep("Class", 2), seq(2), exp(estimate), exp(estimate - 1.96 * std.error), exp(estimate + 1.9
6 * std.error))
```

### Age

```
estimate = P.age %*% mat.freq[3, 1]
std.error = P.age %*% mat.freq[3, 2]

table[4:5, ] = cbind(rep("Age", 2), seq(2), exp(estimate), exp(estimate - 1.96 * std.error), exp(estimate + 1.96
* std.error))
```

### Zone

```
estimate = P.zone %*% mat.freq[4:9, 1]
std.error = P.zone %*% mat.freq[4:9, 2]

table[6:12, ] = cbind(rep("Zone", 7), seq(7), exp(estimate), exp(estimate - 1.96 * std.error), exp(estimate + 1.9
6 * std.error))
```

Here is our final tariff table.

```
as_tibble(table)
```

```
## # A tibble: 12 × 5
##    `Rating Factor` Level Multiplier         `Lower Bound`      `Upper Bound`
##    <chr>           <chr> <chr>              <chr>              <chr>
##  1 Intercept       0     0.0217174424233878 0.0187486458870647 0.02515633973003…
##  2 Class           1     1                  1                  1
##  3 Class           2     0.776747073591671  0.672170749647974  0.897593381814369
##  4 Age             1     1                  1                  1
##  5 Age             2     1.54907948992836   1.28854199152314   1.86229651955708
##  6 Zone            1     1                  1                  1
##  7 Zone            2     7.09843974252673   5.81843728256187   8.66003092089653
##  8 Zone            3     4.17114433996427   3.43293349349061   5.06809850461891
##  9 Zone            4     2.23166210897123   1.79359175329665   2.77672762459128
## 10 Zone            5     1.20370897505158   0.534537249458572  2.71059743373793
## 11 Zone            6     0.793566577664374  0.515743891698537  1.22104773962897
## 12 Zone            7     1.00055416920598   0.319999968104965  3.12846482905617
```

For the severity table, follow the same approach as above. The only difference is the model that we are fitting. I will skip the steps here.

# Comparing two models

- After the new model is fitted, you will need to use the deviance test to compare the two models using the p-value.
- The degree of freedom of the chi-square distribution is the difference between df of the two models.
- To access the df of the model, use the following code:

```
freq$df.residual
```

```
## [1] 19
```

# Combine your multiplier estimates from the frequency and severity data

To get multipliers (and associated 95% confidence intervals) for the premium overall, we can combine the multipliers from the frequency and severity data.

```
table <- matrix(0, nrow = nlevels(moped$class) + nlevels(moped$age) + nlevels(moped$zone) + 1, ncol = 5)
colnames(table) <- c("Rating Factor", "Level", "Multiplier", "Lower Bound", "Upper Bound")

estimate.freq <- mat.freq[1, 1]
std.error.freq <- mat.freq[1, 2]
estimate.sev = mat.sev[1, 1]
std.error.sev = mat.sev[1, 2]

table[1, ] <- c("Intercept", "0", exp(estimate.freq + estimate.sev),
                exp(estimate.freq + estimate.sev - 1.96 * sqrt(std.error.freq^2 + std.error.sev^2)),
                exp(estimate.freq + estimate.sev + 1.96 * sqrt(std.error.freq^2 + std.error.sev^2)))

estimate.freq = P.class %*% mat.freq[2, 1]
std.error.freq = P.class %*% mat.freq[2, 2]
estimate.sev = P.class %*% mat.sev[2, 1]
std.error.sev = P.class %*% mat.sev[2, 2]
table.q6[2:3, ] = cbind(rep("Vehicle class", 2), seq(2), exp(estimate.freq + estimate.sev),
                   exp(estimate.freq + estimate.sev - 1.96 * sqrt(std.error.freq^2 + std.error.sev^2)),
                   exp(estimate.freq + estimate.sev + 1.96 * sqrt(std.error.freq^2 + std.error.sev^2)))

# Other details are omitted here
```

To get the estimate and confidence interval for the overall tariff, we combine the multiplier estimates from the frequency and severity data.

1. Get the point estimates and standard errors from the frequency and severity data.
2. Combine the point estimates by adding them together.
   - adding in the exponent scale is equivalent to multiplying in the original scale.
3. Combine the standard errors from independence assumptions.

   $$\text{Variance}(Z) = \text{Variance}(Z_1) + \text{Variance}(Z_2)$$

   Thus, we have

   $$\sigma(Z) = \sqrt{\sigma(Z_1)^2 + \sigma(Z_2)^2}$$

4. Derive the point estimates and the confidence intervals for the overall tariff by exponentiating the combined point estimates and standard errors.

# Good Luck on Your Assignment!