# Claim Prediction Analysis

Ben Zhang

2021/12/23

## Contents



## Introduction

In this notebook I will use the `swautoins` dataset from the `CASdatasets` R package and estimate claim amounts and frequency. I will be using SQL and Dplyr to store and manipulate data. Then I will be using regression models (multiple linear regression and generalized linear regression) to model the data and estimate expected pure premiums for each tariff cell.

I'm applying what I have learned from a similar project in ACTSC 431 - Property and Casualty, Pricing at the University of Waterloo. This time I will be dealling with a less tidy dataset so I can practice SQL and dpylr. I will also experiment with the ggplot2 library instead of base R graphics

## Preparing the Data

### SQLite Database

Although it is not necessary to use a database here, lets pretend we will do so for the convenience of adding new data down the line. I mostly just wanted to practice working with SQL in a project.

We create a SQLite database and add a dataset called 'autoclaims' into the db.

```r
swautoins = read.csv("C:/Users/bben555/Desktop/spring 2021/actsc 432/swautoins.csv")

swautoins = swautoins %>%
    select(-"X")

conn = dbConnect(RSQLite::SQLite(), "claim_data.db")

dbWriteTable(conn, "swautoins", value = swautoins, overwrite = TRUE)
dbListTables(conn)
```

```
## [1] "mostpayment" "swautoins"
```

This dataset includes the auto insurance data collected in 1977 in Sweden by the Swedish Committee on the Analysis of Risk Premium. There are 5 variables and total of 1703 tariff cells. Here we get a list of columns included in the dataframe:

```r
swautoins %>%
    colnames()
```

```
## [1] "Kilometres" "Zone"       "Bonus"      "Make"       "Insured"
## [6] "Claims"     "Payment"
```

There are 4 rating factors: Kilometres, Zone, Bonus and Make. Kilometres has 5 categories, while Zone, Bonus and Make all have 7 categories.

Let's create a example view in the database that includes the top 100 tariff cells with the most payments. xtract it as a dataframe.

```r
dbExecute(conn, "CREATE VIEW IF NOT EXISTS mostpayment AS SELECT * FROM swautoins ORDER BY Payment DESC
```

```
## [1] 0
```

```r
mostpayment = dbGetQuery(conn, "SELECT * FROM mostpayment")

head(mostpayment)
```

```
##   Kilometres Zone Bonus Make   Insured Claims  Payment
## 1          3    4     7    7 127781.49   3522 19262635
## 2          2    4     7    7 131708.80   3003 16017692
## 3          4    4     7    7  84665.15   2718 14261028
## 4          3    3     7    7  63261.02   2149 11002188
## 5          3    1     7    7  46275.83   2215 10692912
## 6          3    2     7    7  56954.58   2188 10503115
```

Insert and removing dummy rows:

```r
# Inserting rows
dbExecute(conn, "INSERT INTO swautoins (Kilometres, Zone, Bonus) VALUES (?, ?, ?)",
    params = list(c("6", "6", "6"), c("1", "1", "2"), c("2",
        "5", "5")))
```

```
## [1] 3
```

```r
# Deleting the added rows
dbExecute(conn, "DELETE FROM swautoins WHERE rowid = ?", params = 1704)
```

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

```r
dbExecute(conn, "DELETE FROM swautoins WHERE rowid = ?", params = 1705)
```

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

```r
dbExecute(conn, "DELETE FROM swautoins WHERE rowid = ?", params = 1706)
```

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

Above are the basic CRUD (create, read, update, delete) operations using SQL.

## Exploratory Analysis

```r
summary(swautoins)
```

```
##    Kilometres         Zone           Bonus            Make
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1
##  1st Qu.:2.000   1st Qu.:2.000   1st Qu.:2.000   1st Qu.:2
##  Median :3.000   Median :4.000   Median :4.000   Median :4
##  Mean   :3.009   Mean   :3.981   Mean   :4.011   Mean   :4
##  3rd Qu.:4.000   3rd Qu.:6.000   3rd Qu.:6.000   3rd Qu.:6
##  Max.   :5.000   Max.   :7.000   Max.   :7.000   Max.   :7
##     Insured              Claims           Payment
##  Min.   :     0.01   Min.   :   0.00   Min.   :       0
##  1st Qu.:    27.98   1st Qu.:   1.00   1st Qu.:    4072
##  Median :   114.70   Median :   7.00   Median :   36493
##  Mean   :  1399.40   Mean   :  66.45   Mean   :  329390
##  3rd Qu.:   532.88   3rd Qu.:  30.00   3rd Qu.:  150386
##  Max.   :131708.80   Max.   :3522.00   Max.   :19262635
```

There are no NAs in this dataset.

Let's plot the distribution of payments, claims counts, and exposure/duration with respect to the four rating factors.

```r
payment_by_kilo = swautoins %>%
    group_by(Kilometres) %>%
    summarise(Payment = sum(Payment))
payment_by_zone = swautoins %>%
    group_by(Zone) %>%
    summarise(Payment = sum(Payment))
payment_by_bonus = swautoins %>%
    group_by(Bonus) %>%
    summarise(Payment = sum(Payment))
payment_by_Make = swautoins %>%
```

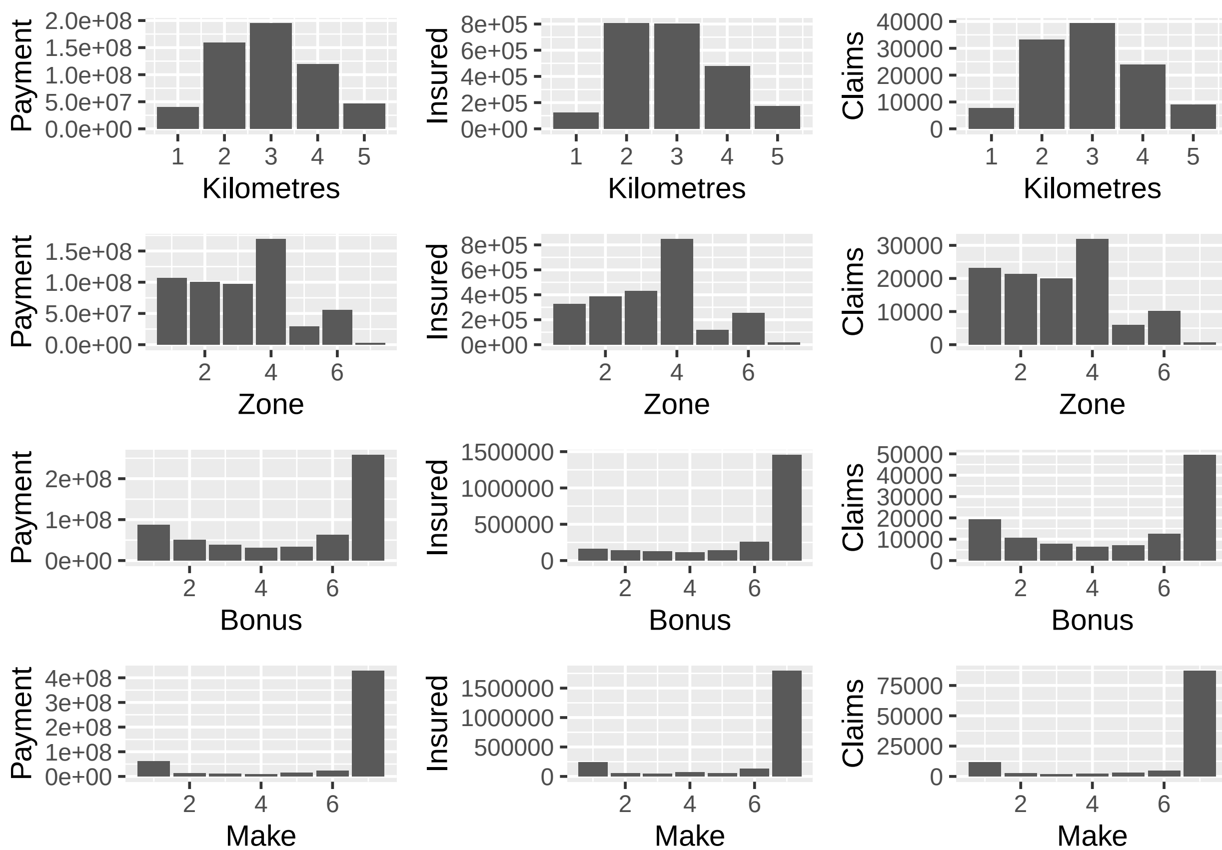
```
    group_by(Make) %>%
    summarise(Payment = sum(Payment))

p1 = ggplot(payment_by_kilo, aes(x = Kilometres, y = Payment)) +
    geom_bar(stat = "identity")
p2 = ggplot(payment_by_zone, aes(x = Zone, y = Payment)) + geom_bar(stat = "identity")
p3 = ggplot(payment_by_bonus, aes(x = Bonus, y = Payment)) +
    geom_bar(stat = "identity")
p4 = ggplot(payment_by_Make, aes(x = Make, y = Payment)) + geom_bar(stat = "identity")

p1.1 = ggplot(swautoins, aes(Kilometres, Insured)) + geom_bar(stat = "identity")
p2.1 = ggplot(swautoins, aes(Zone, Insured)) + geom_bar(stat = "identity")
p3.1 = ggplot(swautoins, aes(Bonus, Insured)) + geom_bar(stat = "identity")
p4.1 = ggplot(swautoins, aes(Make, Insured)) + geom_bar(stat = "identity")

p1.2 = ggplot(swautoins, aes(Kilometres, Claims)) + geom_bar(stat = "identity")
p2.2 = ggplot(swautoins, aes(Zone, Claims)) + geom_bar(stat = "identity")
p3.2 = ggplot(swautoins, aes(Bonus, Claims)) + geom_bar(stat = "identity")
p4.2 = ggplot(swautoins, aes(Make, Claims)) + geom_bar(stat = "identity")

grid.arrange(p1, p1.1, p1.2, p2, p2.1, p2.2, p3, p3.1, p3.2,
    p4, p4.1, p4.2, ncol = 3)
```




In the kilometre rating factor, class 2 and 3 has the most exposure/duration therefore the most sum of claims and payments.

In the zone rating factor, zone 4 has the most exposure and claims and payments. Notice that zone 1 has one of the least exposures but the claims and payments stands out as some of the highest.

In the bonus rating factor, majority of exposures is in class 7. Similarly, class 1 has one of the least exposures but the claims and payments stands out as one of the highest.

In the make rating factor, majority of exposures is in class 7. It has the most sum of claims and payments.

## Regression Analysis

Lets factorize the rating factors and set the tariff cell with the longest duration as the base:

```
swautoins = within(swautoins, {
    Kilometres = factor(Kilometres)
    Zone = factor(Zone)
    Bonus = factor(Bonus)
    Make = factor(Make)
})

basecell = swautoins[which.max(swautoins$Insured), ]

basecell
```

```
##     Kilometres Zone Bonus Make  Insured Claims  Payment
## 530          2    4     7    7 131708.8   3003 16017692
```

```
swautoins$Kilometres = relevel(swautoins$Kilometres, as.character(basecell$Kilometres))
swautoins$Zone = relevel(swautoins$Zone, as.character(basecell$Zone))
swautoins$Bonus = relevel(swautoins$Bonus, as.character(basecell$Bonus))
swautoins$Make = relevel(swautoins$Make, as.character(basecell$Make))
```

### Claim Frequency

We will use a poisson GLM model with insured as an offset and a canonical log link to model claim frequency. We will first fit a crude model with all the rating factors

```
freq = glm(Claims ~ Kilometres + Zone + Bonus + Make + offset(log(Insured)),
    family = poisson("log"), data = swautoins[swautoins$Insured >
        0, ])

freq %>%
    summary()
```

```
##
## Call:
## glm(formula = Claims ~ Kilometres + Zone + Bonus + Make + offset(log(Insured)),
##     family = poisson("log"), data = swautoins[swautoins$Insured >
##         0, ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -6.9849  -0.8875  -0.1749   0.6042   6.2196
##
## Coefficients:
##              Estimate Std. Error  z value Pr(>|z|)
## (Intercept) -3.789635   0.008278 -457.771  < 2e-16 ***
## Kilometres1  0.576668   0.012798   45.061  < 2e-16 ***
## Kilometres3  0.212768   0.007521   28.289  < 2e-16 ***
## Kilometres4  0.320543   0.008654   37.038  < 2e-16 ***
## Kilometres5  0.405129   0.012042   33.643  < 2e-16 ***
## Zone1        0.581742   0.008652   67.238  < 2e-16 ***
## Zone2        0.343639   0.008856   38.801  < 2e-16 ***
## Zone3        0.195477   0.009031   21.646  < 2e-16 ***
## Zone5        0.255835   0.014117   18.122  < 2e-16 ***
## Zone6        0.055825   0.011350    4.919 8.71e-07 ***
## Zone7       -0.149261   0.040552   -3.681 0.000233 ***
## Bonus1       1.327071   0.008677  152.943  < 2e-16 ***
## Bonus2       0.848089   0.010731   79.030  < 2e-16 ***
## Bonus3       0.633966   0.012249   51.755  < 2e-16 ***
## Bonus4       0.499811   0.013381   37.352  < 2e-16 ***
## Bonus5       0.401647   0.012666   31.712  < 2e-16 ***
## Bonus6       0.333900   0.009996   33.404  < 2e-16 ***
## Make1        0.067380   0.009928    6.787 1.15e-11 ***
## Make2        0.143561   0.019451    7.381 1.57e-13 ***
## Make3       -0.180133   0.023610   -7.629 2.36e-14 ***
## Make4       -0.585852   0.022444  -26.102  < 2e-16 ***
## Make5        0.222330   0.018307   12.144  < 2e-16 ***
## Make6       -0.268124   0.015043  -17.824  < 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: 33549.7  on 1702  degrees of freedom
## Residual deviance:  2446.2  on 1680  degrees of freedom
## AIC: 8884
##
## Number of Fisher Scoring iterations: 4
```

From the summary object we see that all of the predictors are significant as the p-values for them are all below the significance level. The effects on claim frequency is also quite strong for a lot of the predictors. We will use a deviance test to see the fit of the model.

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

```
##      scaled.deviance   df p
## [1,]        2446.175 1680 0
```

The p-value of the deviance test is virtually 0. Therefore this is not a good fitting model.

We will now try splitting the data into 2 sets and check the fit of poisson glm.

```r
# tariff cells in Set 1.
swautoins = read.csv("C:/Users/bben555/Desktop/spring 2021/actsc 432/swautoins.csv")
swautoins_set1 = swautoins[(swautoins$Bonus <= 5) | (swautoins$Zone <=
    4), ]

# turn the rating factors into categorical variables
swautoins_set1 = within(swautoins_set1, {
    Kilometres = factor(Kilometres)
    Zone = factor(Zone)
    Bonus = factor(Bonus)
    Make = factor(Make)
})

# change the base cell
basecell = swautoins_set1[which.max(swautoins_set1$Insured),
    ]
swautoins_set1$Kilometres = relevel(swautoins_set1$Kilometres,
    as.character(basecell$Kilometres))
swautoins_set1$Zone = relevel(swautoins_set1$Zone, as.character(basecell$Zone))
swautoins_set1$Bonus = relevel(swautoins_set1$Bonus, as.character(basecell$Bonus))
swautoins_set1$Make = relevel(swautoins_set1$Make, as.character(basecell$Make))


# relative Poisson glm model
freq.set1 = glm(Claims ~ Kilometres + Zone + Bonus + Make + offset(log(Insured)),
    family = poisson("log"), data = swautoins_set1[swautoins_set1$Insured >
        0, ])

freq.set1 %>%
    summary()
```

```
##
## Call:
## glm(formula = Claims ~ Kilometres + Zone + Bonus + Make + offset(log(Insured)),
##     family = poisson("log"), data = swautoins_set1[swautoins_set1$Insured >
##         0, ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -6.9051  -0.8817  -0.1631   0.6221   6.0792
##
## Coefficients:
##              Estimate Std. Error  z value Pr(>|z|)
## (Intercept) -3.803756   0.008538 -445.532  < 2e-16 ***
## Kilometres1  0.584275   0.013455   43.426  < 2e-16 ***
## Kilometres3  0.222750   0.007834   28.433  < 2e-16 ***
## Kilometres4  0.328763   0.009062   36.279  < 2e-16 ***
## Kilometres5  0.413122   0.012681   32.579  < 2e-16 ***
## Zone1        0.580868   0.008655   67.113  < 2e-16 ***
## Zone2        0.343081   0.008858   38.732  < 2e-16 ***
## Zone3        0.195242   0.009031   21.619  < 2e-16 ***
## Zone5        0.221677   0.020576   10.773  < 2e-16 ***
## Zone6        0.019742   0.016559    1.192  0.23317
```

```
## Zone7       -0.184040   0.061228   -3.006  0.00265 **
## Bonus1       1.340851   0.009084  147.607  < 2e-16 ***
## Bonus2       0.861126   0.011068   77.806  < 2e-16 ***
## Bonus3       0.646469   0.012540   51.552  < 2e-16 ***
## Bonus4       0.511761   0.013639   37.522  < 2e-16 ***
## Bonus5       0.413313   0.012935   31.953  < 2e-16 ***
## Bonus6       0.346691   0.010800   32.100  < 2e-16 ***
## Make1        0.075888   0.010433    7.274 3.49e-13 ***
## Make2        0.141125   0.020545    6.869 6.47e-12 ***
## Make3       -0.182036   0.025007   -7.279 3.35e-13 ***
## Make4       -0.588557   0.022948  -25.647  < 2e-16 ***
## Make5        0.227913   0.019256   11.836  < 2e-16 ***
## Make6       -0.268968   0.015746  -17.081  < 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: 31510.4  on 1492  degrees of freedom
## Residual deviance:  2179.2  on 1470  degrees of freedom
## AIC: 7892.4
##
## Number of Fisher Scoring iterations: 4
```

```r
# deviance
cbind(scaled.deviance = freq.set1$deviance, df = freq.set1$df.residual,
    p = 1 - pchisq(freq.set1$deviance, freq.set1$df.residual))
```

```
##      scaled.deviance   df p
## [1,]        2179.184 1470 0
```

For the tariff cells in Set 1, we obtain a p-value of virtually 0 which implies that the relative Poisson glm does not fit well the data.

```r
# repeat the same process for tariff cells in Set 2.
swautoins = read.csv("C:/Users/bben555/Desktop/spring 2021/actsc 432/swautoins.csv")
swautoins_set2 = swautoins[(swautoins$Bonus > 5) & (swautoins$Zone >
    4), ]
swautoins_set2 = within(swautoins_set2, {
    Kilometres = factor(Kilometres)
    Zone = factor(Zone)
    Bonus = factor(Bonus)
    Make = factor(Make)
})

basecell = swautoins_set2[which.max(swautoins_set2$Insured),
    ]
swautoins_set2$Kilometres = relevel(swautoins_set2$Kilometres,
    as.character(basecell$Kilometres))
swautoins_set2$Zone = relevel(swautoins_set2$Zone, as.character(basecell$Zone))
swautoins_set2$Bonus = relevel(swautoins_set2$Bonus, as.character(basecell$Bonus))
swautoins_set2$Make = relevel(swautoins_set2$Make, as.character(basecell$Make))
```

```r
freq.set2 = glm(Claims ~ Kilometres + Zone + Bonus + Make + offset(log(Insured)),
    family = poisson("log"), data = swautoins_set2[swautoins_set2$Insured >
        0, ])

summary(freq.set2)
```

```
##
## Call:
## glm(formula = Claims ~ Kilometres + Zone + Bonus + Make + offset(log(Insured)),
##     family = poisson("log"), data = swautoins_set2[swautoins_set2$Insured >
##         0, ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7073  -0.7918  -0.2205   0.5078   2.8562
##
## Coefficients:
##             Estimate Std. Error  z value Pr(>|z|)
## (Intercept) -3.51710    0.02142 -164.183  < 2e-16 ***
## Kilometres1  0.39138    0.04064    9.630  < 2e-16 ***
## Kilometres2 -0.08938    0.02685   -3.329 0.000871 ***
## Kilometres4  0.13255    0.02786    4.757 1.96e-06 ***
## Kilometres5  0.21940    0.03754    5.845 5.06e-09 ***
## Zone5        0.20042    0.02177    9.206  < 2e-16 ***
## Zone7       -0.21226    0.05505   -3.855 0.000116 ***
## Bonus6       0.26110    0.02644    9.877  < 2e-16 ***
## Make1       -0.01179    0.03228   -0.365 0.714863
## Make2        0.16418    0.06041    2.718 0.006572 **
## Make3       -0.16367    0.07166   -2.284 0.022382 *
## Make4       -0.49725    0.10774   -4.615 3.93e-06 ***
## Make5        0.16712    0.05905    2.830 0.004650 **
## Make6       -0.25696    0.05092   -5.047 4.50e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 683.24  on 209  degrees of freedom
## Residual deviance: 207.87  on 196  degrees of freedom
## AIC: 960.49
##
## Number of Fisher Scoring iterations: 4
```

```r
cbind(scaled.deviance = freq.set2$deviance, df = freq.set2$df.residual,
    p = 1 - pchisq(freq.set2$deviance, freq.set2$df.residual))
```

```
##      scaled.deviance  df         p
## [1,]        207.8698 196 0.2671894
```

For the tariff cells in Set 2, we obtain a p-value of 0.2671894 which implies that we do not reject the null hypothesis that the relative Poisson glm fits the data well. Hence, the relative Poisson model is more appropriate for the tariff cells in Set 2 (than those in Set 1).

In this model, Make1 is deemed as not significant. Lets try dropping the rating factor `Make` to see if it is a better fitting model.

```
# relative Poisson glm model for Set 2 without the rating
# factor Make
freq.set2.wMake = glm(Claims ~ Kilometres + Zone + Bonus + offset(log(Insured)),
    family = poisson("log"), data = swautoins_set2[swautoins_set2$Insured >
        0, ])

freq.set2.wMake %>%
    summary()
```

```
##
## Call:
## glm(formula = Claims ~ Kilometres + Zone + Bonus + offset(log(Insured)),
##     family = poisson("log"), data = swautoins_set2[swautoins_set2$Insured >
##         0, ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.2703  -0.8785  -0.3180   0.5706   3.0603
##
## Coefficients:
##             Estimate Std. Error  z value Pr(>|z|)
## (Intercept) -3.52947    0.02042 -172.855  < 2e-16 ***
## Kilometres1  0.39529    0.04058    9.741  < 2e-16 ***
## Kilometres2 -0.10235    0.02671   -3.832 0.000127 ***
## Kilometres4  0.13549    0.02785    4.865 1.15e-06 ***
## Kilometres5  0.22303    0.03752    5.945 2.77e-09 ***
## Zone5        0.20152    0.02177    9.258  < 2e-16 ***
## Zone7       -0.21181    0.05503   -3.849 0.000119 ***
## Bonus6       0.26126    0.02643    9.884  < 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: 683.24  on 209  degrees of freedom
## Residual deviance: 281.84  on 202  degrees of freedom
## AIC: 1022.5
##
## Number of Fisher Scoring iterations: 4
```

```
# likelihood ratio test H0: all betas of the rating factor
# 'Make' are 0, Ha: at least one beta of the rating factor
# 'Make' is different than 0

anova(freq.set2.wMake, freq.set2, test = "LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: Claims ~ Kilometres + Zone + Bonus + offset(log(Insured))
## Model 2: Claims ~ Kilometres + Zone + Bonus + Make + offset(log(Insured))
```

```
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       202     281.84
## 2       196     207.87  6   73.973 6.244e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Since the p-value is 6.239453e-14, we are not statistically justified to simplify the model by dropping the variable Make. Lets instead try grouping Make 1 and 3 into the base Make level 7.

```r
# merge categories 1, 3 and 7 of Make into one category

levels(swautoins_set2$Make) = recode(levels(swautoins_set2$Make),
    `1` = "1&3&7")
levels(swautoins_set2$Make) = recode(levels(swautoins_set2$Make),
    `3` = "1&3&7")
levels(swautoins_set2$Make) = recode(levels(swautoins_set2$Make),
    `7` = "1&3&7")

# merge categories 2 and 5 of Make into one category

levels(swautoins_set2$Make) = recode(levels(swautoins_set2$Make),
    `2` = "2&5")
levels(swautoins_set2$Make) = recode(levels(swautoins_set2$Make),
    `5` = "2&5")

# relative Poisson glm model

freq.set2.mMake = glm(Claims ~ Kilometres + Zone + Bonus + Make +
    offset(log(Insured)), family = poisson, data = swautoins_set2[swautoins_set2$Insured >
    0, ])

freq.set2.mMake %>%
    summary()
```

```
##
## Call:
## glm(formula = Claims ~ Kilometres + Zone + Bonus + Make + offset(log(Insured)),
##     family = poisson, data = swautoins_set2[swautoins_set2$Insured >
##         0, ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7136  -0.8309  -0.2682   0.4841   2.8508
##
## Coefficients:
##             Estimate Std. Error  z value Pr(>|z|)
## (Intercept) -3.52304    0.02079 -169.482  < 2e-16 ***
## Kilometres1  0.38863    0.04059    9.574  < 2e-16 ***
## Kilometres2 -0.08703    0.02680   -3.247 0.001166 **
## Kilometres4  0.13160    0.02786    4.724 2.31e-06 ***
## Kilometres5  0.21767    0.03752    5.801 6.58e-09 ***
## Zone5        0.20026    0.02177    9.199  < 2e-16 ***
## Zone7       -0.21000    0.05504   -3.815 0.000136 ***
```

```
## Bonus6       0.26184    0.02643    9.906  < 2e-16 ***
## Make2&5      0.17168    0.04278    4.013 6.01e-05 ***
## Make4       -0.49308    0.10767   -4.580 4.66e-06 ***
## Make6       -0.25134    0.05070   -4.957 7.16e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 683.24  on 209  degrees of freedom
## Residual deviance: 213.41  on 199  degrees of freedom
## AIC: 960.03
##
## Number of Fisher Scoring iterations: 4
```

```r
# likelihood ratio test

anova(freq.set2.mMake, freq.set2, test = "LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: Claims ~ Kilometres + Zone + Bonus + Make + offset(log(Insured))
## Model 2: Claims ~ Kilometres + Zone + Bonus + Make + offset(log(Insured))
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       199     213.41
## 2       196     207.87  3   5.5384   0.1364
```

For this alternative simplified model, the p-value of the likelihood ratio test is 0.1363605. We are statistically justified to simplify `freq.set2` to `freq.set2.mMake`.

## Claim Severity

Let's focus on only data in set 2 while modeling claim severity.

```r
# reload the dataset and extract only tariff cells in Set 2

swautoins = read.csv("C:/Users/bben555/Desktop/spring 2021/actsc 432/swautoins.csv")
swautoins_set2 = swautoins[(swautoins$Bonus > 5) & (swautoins$Zone >
    4), ]

# turn the rating factors into categorical variables

swautoins_set2 = within(swautoins_set2, {
    Kilometres = factor(Kilometres)
    Zone = factor(Zone)
    Bonus = factor(Bonus)
    Make = factor(Make)
})

# change the base tariff cell

basecell = swautoins_set2[which.max(swautoins_set2$Insured),
```

```r
    ]
swautoins_set2$Kilometres = relevel(swautoins_set2$Kilometres,
    as.character(basecell$Kilometres))
swautoins_set2$Zone = relevel(swautoins_set2$Zone, as.character(basecell$Zone))
swautoins_set2$Bonus = relevel(swautoins_set2$Bonus, as.character(basecell$Bonus))
swautoins_set2$Make = relevel(swautoins_set2$Make, as.character(basecell$Make))
```

### Multiple Linear Regression

Let's first try fitting a multiple linear regression model. This will assume the response, severity per claim, is normally distributed.

```r
sev.norm = lm(Payment/Claims ~ Kilometres + Zone + Bonus + Make,
    data = swautoins_set2[swautoins_set2$Claims > 0, ])

summary(sev.norm)
```

```
##
## Call:
## lm(formula = Payment/Claims ~ Kilometres + Zone + Bonus + Make,
##     data = swautoins_set2[swautoins_set2$Claims > 0, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7255.9 -2695.2  -800.4   853.2 24975.3
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5561.13    1342.77   4.142 5.63e-05 ***
## Kilometres1  2409.76    1236.21   1.949   0.0531 .
## Kilometres2    22.94    1143.04   0.020   0.9840
## Kilometres4   759.76    1136.72   0.668   0.5049
## Kilometres5   746.56    1197.05   0.624   0.5338
## Zone5        -837.07     862.69  -0.970   0.3334
## Zone7         354.97    1007.35   0.352   0.7250
## Bonus6      -1113.17     763.99  -1.457   0.1471
## Make1         251.02    1291.04   0.194   0.8461
## Make2         189.93    1318.03   0.144   0.8856
## Make3         628.00    1373.71   0.457   0.6482
## Make4         158.97    1551.49   0.102   0.9185
## Make5       -1763.41    1371.82  -1.285   0.2005
## Make6         935.17    1355.20   0.690   0.4912
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4907 on 156 degrees of freedom
## Multiple R-squared:  0.0791, Adjusted R-squared:  0.002363
## F-statistic: 1.031 on 13 and 156 DF,  p-value: 0.4247
```

```r
c(R_Squared = summary(sev.norm)$r.squared)
```

```
##  R_Squared
```

```
## 0.07910477
```

The $R^2$ of the linear regression model is very poor. The predictors are not significant either. Therefore linear regression is not appropriate for this dataset.

### Gamma GLM

Now we will check the fit of a gamma glm model.

```r
# gamma glm model Note that the response variable is
# Payment/Claims, which measures the average claim amount
# per claim.  We use weights because data was a sum of
# gamma variables.
sev = glm(Payment/Claims ~ Kilometres + Zone + Bonus + Make,
    family = Gamma("log"), data = swautoins_set2[swautoins_set2$Claims >
        0, ], weights = Claims)

summary(sev)
```

```
##
## Call:
## glm(formula = Payment/Claims ~ Kilometres + Zone + Bonus + Make,
##     family = Gamma("log"), data = swautoins_set2[swautoins_set2$Claims >
##         0, ], weights = Claims)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3314  -1.2512  -0.4377   0.8988   4.8973
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.60975    0.03735 230.542  < 2e-16 ***
## Kilometres1  0.02134    0.07074   0.302  0.76327
## Kilometres2  0.01205    0.04663   0.258  0.79640
## Kilometres4  0.01823    0.04847   0.376  0.70740
## Kilometres5  0.05561    0.06534   0.851  0.39602
## Zone5       -0.09742    0.03791  -2.570  0.01111 *
## Zone7       -0.05667    0.09575  -0.592  0.55482
## Bonus6      -0.03722    0.04598  -0.810  0.41944
## Make1        0.04718    0.05614   0.840  0.40194
## Make2        0.09200    0.10508   0.876  0.38262
## Make3        0.04411    0.12452   0.354  0.72365
## Make4       -0.02426    0.18720  -0.130  0.89704
## Make5       -0.28798    0.10263  -2.806  0.00566 **
## Make6        0.09076    0.08856   1.025  0.30699
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 3.020134)
##
##     Null deviance: 436.16  on 169  degrees of freedom
## Residual deviance: 378.56  on 156  degrees of freedom
## AIC: 160113
```

```
##
## Number of Fisher Scoring iterations: 5
```

```r
# deviance (deviance/dispersion)
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,]        125.3445 156 0.9660689
```

Except for Make 5 and possibly Zone 5, all the other tariff factors do not seem to be significant to predict the severity key ratio in the tariff cells in Set 2. As measured by the deviance statistic, the gamma glm fit seems to be quite good as the p-value is of 0.9660689.

Let's compare the model above to a model where we group all categories except for zone 5 and make 5 into the base category.

```r
# group all categories (except 5) of Zone into the base
# category of Zone (Zone 6)
swautoins_set2$Zone[swautoins_set2$Zone != "5"] <- 6

# group all categories (except 5) of Make into the base
# category of Make (Make 7)
swautoins_set2$Make[swautoins_set2$Make != "5"] <- 7

# gamma glm severity model
summary(sev.simpl <- glm(Payment/Claims ~ Zone + Make, family = Gamma("log"),
    data = swautoins_set2[swautoins_set2$Claims > 0, ], weights = Claims))
```

```
##
## Call:
## glm(formula = Payment/Claims ~ Zone + Make, family = Gamma("log"),
##     data = swautoins_set2[swautoins_set2$Claims > 0, ], weights = Claims)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2833  -1.2269  -0.4072   0.8419   4.9414
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.62896    0.02195 393.079  < 2e-16 ***
## Zone5       -0.09624    0.03687  -2.610  0.00988 **
## Make5       -0.30332    0.10083  -3.008  0.00303 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 2.944661)
##
##     Null deviance: 436.16  on 169  degrees of freedom
## Residual deviance: 391.25  on 167  degrees of freedom
## AIC: 160408
##
## Number of Fisher Scoring iterations: 5
```

```r
# likelihood ratio test
anova(sev.simpl, sev, test = "LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: Payment/Claims ~ Zone + Make
## Model 2: Payment/Claims ~ Kilometres + Zone + Bonus + Make
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       167     391.25
## 2       156     378.56 11   12.697   0.9636
```

Since the p-value is of 0.9636, we are statistically justified to simplify `sev` to `sev.simpl`.

# Predictions & Conclusion

After fitting the models, we found two statistically significant GLM models that fits the claim frequency and severity of the swautoins dataset.

The expected key ratio is the the ratio of a rating factor level compared to the base level. Multiplying the expected key frequency ratio with key severity ratio will give us the expected pure premium of a tariff cell.

```r
# print the relativities of key frequency ratio of claim
# freq
exp(freq.set2.mMake$coefficients)
```

```
## (Intercept) Kilometres1 Kilometres2 Kilometres4 Kilometres5       Zone5
##   0.0295096   1.4749521   0.9166481   1.1406477   1.2431803   1.2217174
##       Zone7      Bonus6     Make2&5       Make4       Make6
##   0.8105850   1.2993165   1.1872921   0.6107446   0.7777589
```

```r
# print the relativities of key severity ratio of claim sev
exp(sev.simpl$coefficients)
```

```
##  (Intercept)        Zone5        Make5
## 5591.2344401    0.9082442    0.7383632
```

Let's calculated the expected pure premium of the cell (Kilometres=5, Zone=7, Bonus=7 and Make=4).

The expected key frequency ratio of the cell (Kilometres=5, Zone=7, Bonus=7 and Make=4) is:

```r
0.0295096 * 1.2431803 * 0.810585 * 1 * 0.6107446
```

```
## [1] 0.01816166
```

The expected key severity ratio of the cell (Kilometres=5, Zone=7, Bonus=7 and Make=4) is:

```r
5591.2344401 * 1 * 1 * 1 * 1
```

```
## [1] 5591.234
```

The expected pure premium of the cell (Kilometres=5, Zone=7, Bonus=7 and Make=4) is:

```r
0.01816166 * 5591.234
```

```
## [1] 101.5461
```

Compare the result to the actual observed premiums:

```r
swautoins %>%
    filter(Kilometres == 5, Zone == 7, Bonus == 7, Make == 4)
```

```
##      X Kilometres Zone Bonus Make Insured Claims Payment
## 1 1700          5    7     7    4    6.69      0       0
```