Assignment part 2

Nguyen Phuc Thai

2020-11-19

Contents

1 Data import

The data was imported using `read.csv`. And a summary of the variables presented in the data is as follow

Table 1: Summary of variables

	year	exposure	business.type	driver.age	driver.gender	marital.status	yrs.licensed
	Min. :2015	Min. :0.083	NB:27253	Min. :18	Female: 4512	Divorced: 886	Min. : 1.0
	1st Qu.:2016	1st Qu.:0.250	RB:13368	1st Qu.:36	Male :36109	Married :37729	1st Qu.: 2.0
	Median :2017	Median :0.500		Median :44		Single : 1525	Median : 3.0
	Mean :2017	Mean :0.510		Mean :45		Widow : 481	Mean : 3.2
	3rd Qu.:2018	3rd Qu.:0.750		3rd Qu.:52			3rd Qu.: 5.0
	Max. :2018	Max. :1.000		Max. :93			Max. :10.0

	ncd.level	region	body.code	vehicle.age	vehicle.value	no.seats	cubic.cent
	Min. :1.0	Min. : 1	A :17418	Min. : 0.0	Min. : 4.5	Min. :2	Min. : 970
	1st Qu.:1.0	1st Qu.: 7	D :10056	1st Qu.: 1.0	1st Qu.: 17.0	1st Qu.:2	1st Qu.:1398
	Median :3.0	Median :17	E : 8304	Median : 3.0	Median : 22.1	Median :5	Median :1560
	Mean :3.4	Mean :16	C : 2106	Mean : 3.3	Mean : 23.5	Mean :4	Mean :1670
	3rd Qu.:5.0	3rd Qu.:23	G : 1374	3rd Qu.: 5.0	3rd Qu.: 28.8	3rd Qu.:5	3rd Qu.:1896
	Max. :6.0	Max. :38	B : 711	Max. :18.0	Max. :132.6	Max. :9	Max. :3198
			(Other): 652				

	horse.power	weight	length	width	height	fuel.type	prior.claims
	Min. : 42	Min. : 860	Min. :1.8	Min. :1.5	Min. :1.4	Diesel :39798	Min. : 0.00
	1st Qu.: 70	1st Qu.:1190	1st Qu.:4.0	1st Qu.:1.7	1st Qu.:1.8	Gasoline: 568	1st Qu.: 0.00
	Median : 75	Median :1320	Median :4.3	Median :1.7	Median :1.8	LPG : 255	Median : 0.00
	Mean : 86	Mean :1364	Mean :4.3	Mean :1.8	Mean :1.8		Mean : 0.83
	3rd Qu.:100	3rd Qu.:1475	3rd Qu.:4.4	3rd Qu.:1.8	3rd Qu.:1.8		3rd Qu.: 1.00
	Max. :200	Max. :2275	Max. :6.9	Max. :2.1	Max. :2.5		Max. :21.00

	claim.count	claim.incurred
	Min. :0.000	Min. : 0
	1st Qu.:0.000	1st Qu.: 0
	Median :0.000	Median : 0
	Mean :0.084	Mean : 67
	3rd Qu.:0.000	3rd Qu.: 0
	Max. :5.000	Max. :11684

2 Data preparation

The data was properly cleaned,hence only data preparation is needed.

From table 1 and according to the provided data dictionary, I will change the class of `ncd.level` and `region` variables to factor as they are currently recognized as numerical data.

Furthermore, `year` column will be removed from this dataset and not used for modelling. This is because it informs us the time that these observations were recorded. Hence the model built will make prediction on future data, and thus have different years from what we have.

Finally, I will create 3 new variables for this dataset for frequency, severity and total loss (denoted as freq, sev and tot.loss respectively) they are calculated as follow:

```
gen.data$region<-as.factor(gen.data$region)
gen.data$ncd.level<-as.factor(gen.data$ncd.level)

gen.data1<-gen.data%>%
  dplyr::select(-year)%>%
  mutate(freq=claim.count/exposure,
         sev=ifelse(claim.incurred>0,
                    claim.incurred/claim.count,0),
         tot.loss=claim.incurred/exposure)

cat.col<-which(lapply(gen.data1, class)=='factor')
num.col<-which(lapply(gen.data1, class)!='factor')
```

2.1 Data splitting

This step is done beforehand. And every exploration will be from the training data to avoid information leakage. All members from my group will also split the data similarly

```
set.seed(1)
index <- createDataPartition(gen.data$year,p = 0.75,list = FALSE)
train.data <- gen.data1[index,]
test.data <- gen.data1[-index,]
```

3 Exploratory analysis

3.1 Multicollinearity

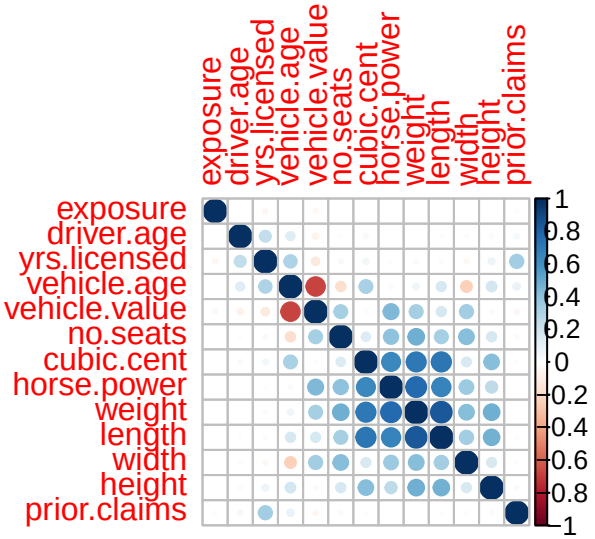

Figure 1: Correlation plot

Figure 1 shows several strong correlation between the variables. These include the negative correlation between `vehicle.age` and `vehicle value`, which is expected as the older a vehicle is, the cheaper it becomes. Moreover, we can also observe some strong correlation between a vehicle features (`no.seat`,`cubic.cent`,`horse.power`,`weight`, `height`, `length` and `width`). Hence these variable need to be consider carefully when doing fitting as inclusion of them in the model may negatively affect its performance.

Hence to account for them, I will use principal components (PC) to combine these correlated variable together

```
#Vehicle size
gen.size.pca1 <- prcomp(train.data[, 12:18],  scale = TRUE)
gen.size_colsnames <- paste0("PCsize", c(1:7))
colnames(gen.size.pca1$x)<-gen.size_colsnames


gen.pca<-rep(NA,nrow(gen.data1)*3)
dim(gen.pca)<-c(nrow(gen.data1),3)
for(j in 1:3){
for(i in 1:nrow(gen.data1)){

 gen.pca[i,j]<-sum(gen.size.pca1$rotation[,j]* (gen.data1[i,12:18]-
     gen.size.pca1$center)/gen.size.pca1$scale )
}
}
colnames(gen.pca)<-gen.size_colsnames[1:3]
gen.data3<-cbind(gen.data1[,-c(12:18)],gen.pca)

#vehicle age/value
gen.age.pca1 <- prcomp(train.data[, 10:11],  scale = TRUE)
gen.age_colsnames <- paste0("PCage", c(1:2))
colnames(gen.age.pca1$x)<-gen.age_colsnames
gen.pca.age<-rep(NA,nrow(gen.data1))
  for(i in 1:nrow(gen.data1)){
    gen.pca.age[i]<-sum(gen.age.pca1$rotation[,1]* (gen.data1[i,10:11]-
                          gen.age.pca1$center)/gen.age.pca1$scale )
  }
gen.data3<-cbind(gen.data3[,-c(10:11)],PCage1=gen.pca.age)

train.data1 <- gen.data3[index, ]
test.data1 <- gen.data3[-index, ]
```

And then to decide the number of PCs to be used I will look at the proportion of variance explained by these components

Vehicle size components

```
## Importance of components:
##                            PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.9644 1.0719 0.8705 0.75988 0.54267 0.49502 0.34241
## Proportion of Variance 0.5513 0.1641 0.1082 0.08249 0.04207 0.03501 0.01675
## Cumulative Proportion  0.5513 0.7154 0.8237 0.90617 0.94824 0.98325 1.00000
```

Vehicle value and age components

```
## Importance of components:
##                            PC1    PC2
## Standard deviation     1.2934 0.5718
## Proportion of Variance 0.8365 0.1635
## Cumulative Proportion  0.8365 1.0000
```

For the Vehicle size, only the first 3 components explain more than 10% of the variance, they will be chosen. For vehicle age and value, as there are only 2 components, and the first one explain more data 80% of the variance, it will be the only one to be used for modelling.

3.2 Distribution of response

The distribution of claim count and severity are as follows

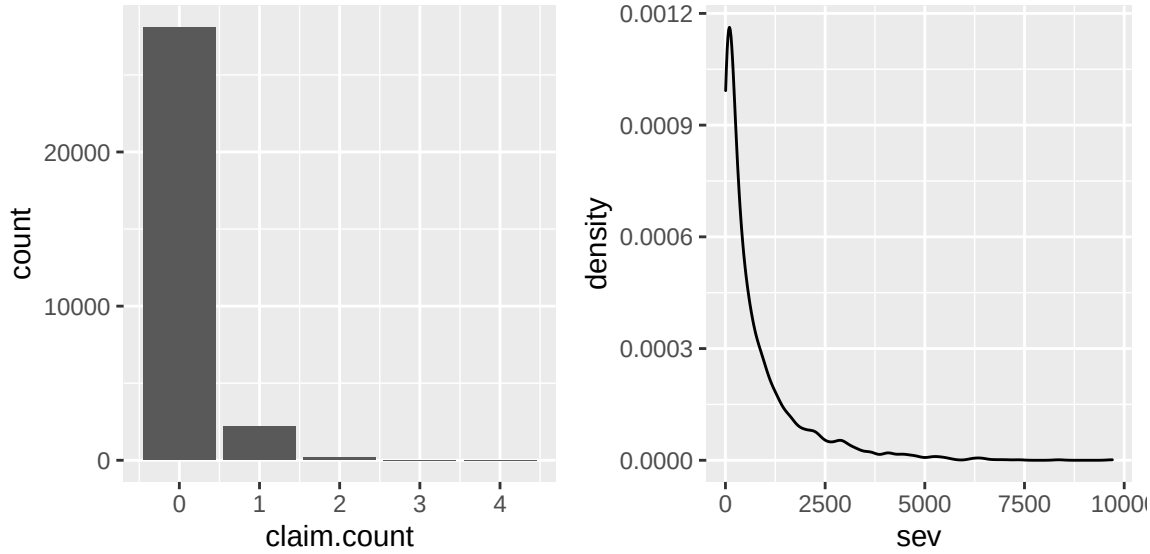

Figure 2: Claim count distribution

From figure 2, most policyholders lodge no claim in the data set. Meanwhile most claims have severity between 0 and 1250, but the distribution also have a very large tail, with some have severity of nearly 10,000

Some exploration of the relationship between the response variables and the predictors are shown in Appendix 6.1. This exploration reveals no standout relationship between predictors and the responses

4 Modelling

4.1 Methodology

Evaluating the pure premium by estimating severity and frequency independently.

Severity and Frequency will be modelled using GLMs

For Frequency: I will model claim count then the predicted frequency will be predicted claim count divided by exposure. Modelled using GLM with Negative Binomial distribution, with log link and Exposure being considered as the offset in this model.

For Severity: Modelled using GLM with Gamma distribution with log link

The modelling steps involve:

First fit a raw models using all the predictors. However, two sets of predictors is tested, 1 with all highly correlated variables remained untouched, and 1 with highly correlated variables combined via Principal components Analysis. If an offset is used, add two more models in this initial steps

Then feature selection is applied. Method 1 is using backward selection with `stepAIC()`. This function start from the full model, then trim responses until the minimal AIC is obtained. Method 2 is just removing the unimportant variables with high coefficient (as they are the ones that will have more impact on the model, as discussed by guest lecturer, Dr. Hugh Miller). 2 methods are applied because there are many variables, hence many insignificant variables with low coefficient may have significant overall impact on predictions.

Collapsing categorical data's level is also considered. Collapsing methods: levels that are similar of mean and median with respect to the response and appearance frequency in the data are combined. And this is done on the training data to avoid information leakage.

Finally, interaction terms will be tested and added to the model. Tested on targeted groups of features, with the 2 groups being:

- Personal information of policyholders (marital.status, driver.gender, driver.age)
- Vehicle information (vehicle.age, vehicle.value, no.seats, cubic.cent, horse.power, weight, fuel.type, length, width, height or their corresponding Principle components)

4.2 Frequency

For claim.count, it is important to notice that each observation have different exposure, which is the period during the year that they are covered by insurance. And hence the number of claim can be expected to vary proportionally with the exposure (people with longer exposure have more time for claim and hence may make more claim). Therefore, this variable may be an offset for the claim.count variable.

4.2.1 Initial models

```
#count model with no offset and no PCA variable
countmodel.nb<-glm.nb(claim.count~.-freq-sev-tot.loss-claim.incurred,
                      data=train.data )
#count model with no offset and with PCA variable
countmodel.nb1<-glm.nb(claim.count~.-freq-sev-tot.loss-claim.incurred,
                       data=train.data1 )

#count model with offset and no PCA variables
countmodel.nb.off<-glm.nb(claim.count~.+offset(log(exposure))-exposure-freq-
                          sev-tot.loss-claim.incurred, data=train.data )

#count model with offset and PCA variables
countmodel.nb.off1<-glm.nb(claim.count~.+offset(log(exposure))-exposure-freq-
                          sev-tot.loss-claim.incurred, data=train.data1 )

tab1<-cbind(rbind(AIC(countmodel.nb),AIC(countmodel.nb1),
                          AIC(countmodel.nb.off),AIC(countmodel.nb.off1)),
                  rbind(BIC(countmodel.nb),BIC(countmodel.nb1),
                          BIC(countmodel.nb.off),BIC(countmodel.nb.off1)))
colnames(tab1)<-c('AIC','BIC')
rownames(tab1)<-c('Negative binomial with no PC','Negative binomial with PC',
                  'Negative binomial with no PC (with offset)',
                  'Negative binomial with PC (with offset)')
k=5
set.seed(1)
P1<-cv.glm(train.data,countmodel.nb,K=k)
P2<-cv.glm(train.data1,countmodel.nb1,K=k)
P3<-cv.glm(train.data,countmodel.nb.off,K=k)
P4<-cv.glm(train.data1,countmodel.nb.off1,K=k)
cv.error1<-c(P1$delta[2],P2$delta[2],P3$delta[2],P4$delta[2])
pred.nb<-predict(countmodel.nb,test.data,type='response')
pred.nb1<-predict(countmodel.nb1,test.data1,type='response')
pred.nb.off<-predict(countmodel.nb.off,test.data,type='response')
pred.nb.off1<-predict(countmodel.nb.off1,test.data1,type='response')
test.MSE<-c(RMSE(test.data$claim.count,pred.nb),
            RMSE(test.data$claim.count,pred.nb1),
            RMSE(test.data$claim.count,pred.nb.off),
            RMSE(test.data$claim.count,pred.nb.off1))
tab1<-cbind(tab1,cv.error1,test.MSE)
```

Table 2: performance of initial claim count models

	AIC	BIC	cv.error1	test.MSE
Negative binomial with no PC	16843.31	17434.34	0.0869683	0.2969109
Negative binomial with PC	16861.95	17411.36	0.0869554	0.2969686
Negative binomial with no PC (with offset)	16772.74	17355.45	0.0865882	0.2965233
Negative binomial with PC (with offset)	16795.78	17336.86	0.0868574	0.2966126

From table 2 all of the criteria used seem to agree that the model with the offset is better comparing to the one that does not have the offset. However, regarding the use of principal components, it is suprising to find that models with no PC outperforms the one with PC in all 3 out of 4 criteria, because the latter use less features and its feature is not highly correlated. The only area that the models with PC win is BIC, where the number of features used is penalized harder compared to AIC. **Yet because of this result, I will move forward with the model with offset that did not utilize PCA.**

Before that it is worth looking at the residual plots from these models

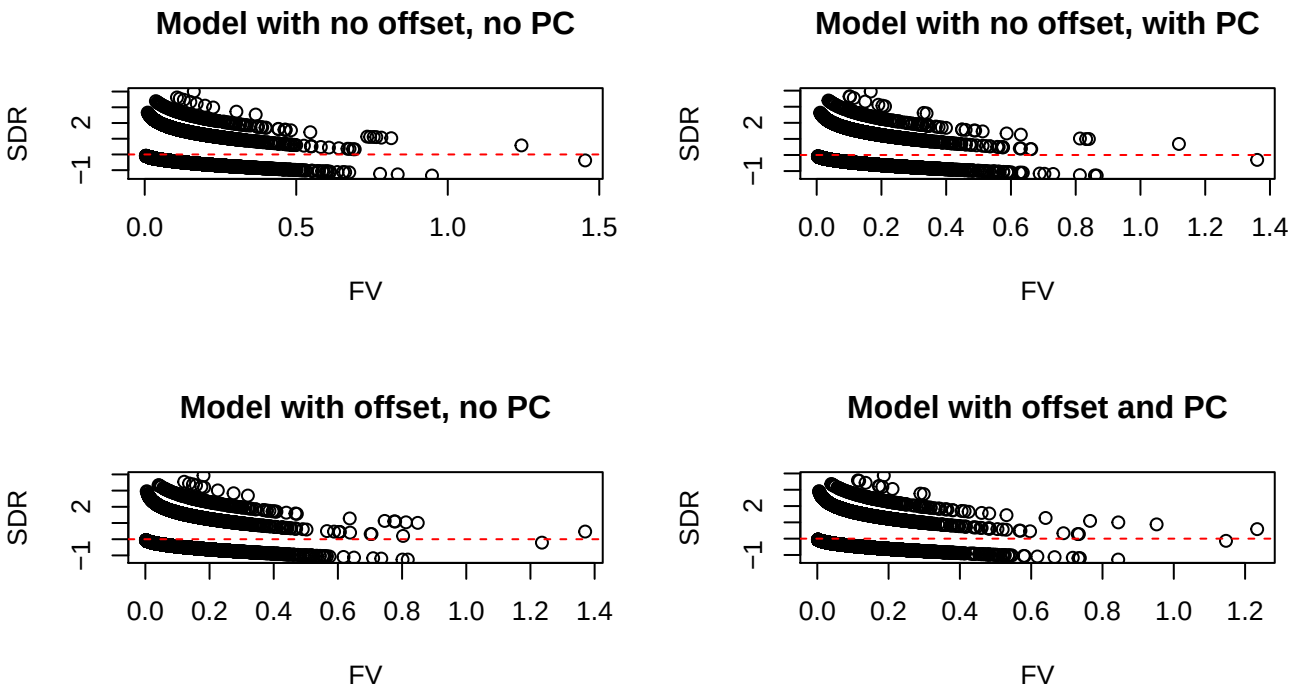

Figure 3: Fitted value vs standardized deviance residuals.

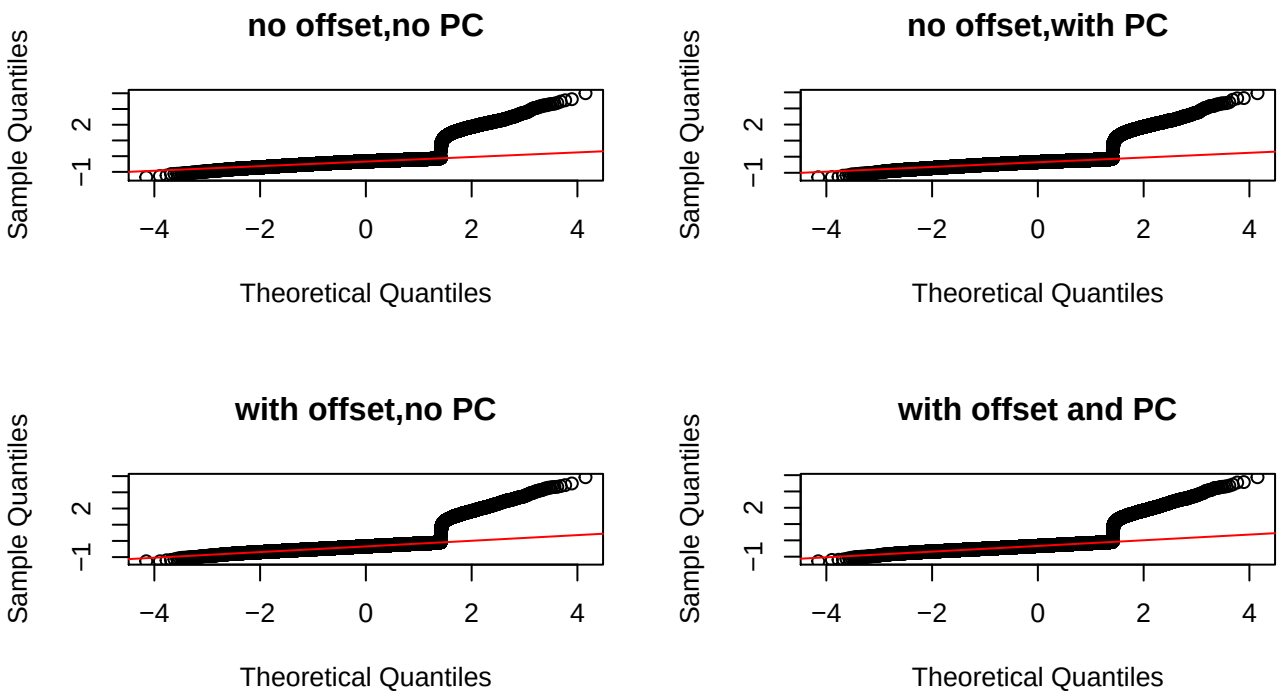

Figure 4: QQ plots

Both figure 3 and 4 show that using Negative binomials assumption is quite inadequate for the claim count. For it to be an appropriate fit, the errors shown in 3 should have no particular trend around the red line. Also, in the QQ plots 4 the points fit the line very poorly at the right tails indicating that the models underfit when the number of claims grow bigger. These plot indicate another model may be more adequate. However, as one goal of this report is to explore GLM with Negative Binomial for modelling claim count, we will carrying on refining this models. These plots will not be looked at again because plots from following models will exhibit very similar pattern.

4.2.2 Refining models with feature selection

Unimportant variables with high coefficients for the models are length, height. fuel.type is also removed as it is a categorical feature that is important on all of its levels. *These result and the resulted model from stepAIC() can be found in Appendix 6.2.1*

For categorical feature collapsing, I found that only the region variable have levels that can be combined. It is done as follow. *And please refer to Appendix 6.4.1 for the tables that were used for the analysis of levels that can be combined.*

```
region1<-ifelse(gen.data1$region%in%c(1,32),'group1',
          ifelse(gen.data1$region %in% c(4,3),'group2',
          ifelse(gen.data1$region %in% c(16,34,31),'group3',
          ifelse(gen.data1$region %in%  c(30,11),'group4',
          ifelse(gen.data1$region %in%  c(21,38,33), 'group5',
```

```
          ifelse(gen.data1$region %in%  c(5,13),'group6',
          ifelse(gen.data1$region %in%  c(14,12,23,37,19), 'group7',
          ifelse(gen.data1$region %in%  c(36,6),'group8',gen.data1$region
                        ))))))))
region1<-as.factor(region1)

gen.data11<-cbind(gen.data1[,-8],region1)
gen.data31<-cbind(gen.data3[,-8],region1 )

train.data11 <- gen.data11[index, ]
test.data11 <- gen.data11[-index, ]
train.data31 <- gen.data31[index, ]
test.data31 <- gen.data31[-index, ]
```

4 new models are considered at this point, they are done as follow:

```
#model with unimportant variable with |coef|>1 removed (model 2)
countmodel.nb.off2.1<-glm.nb(claim.count~.+offset(log(exposure))
                             -exposure-freq-sev-
                               tot.loss-claim.incurred-length-
                               height-fuel.type,data=train.data)

#model with stepAIC variable (model 3)
countmodel.nb.off2.2<-glm.nb(claim.count ~ business.type + driver.age + driver.gender +
  marital.status + yrs.licensed + ncd.level + region + body.code +
  vehicle.age + vehicle.value + no.seats + horse.power + weight +
  width + prior.claims + offset(log(exposure)),data=train.data)

#model with unimportant variable with |coef|>1 removed (model 4)
#   and collapsed categorical variable
countmodel.nb.off2.3<-glm.nb(claim.count~.+offset(log(exposure))-exposure-
                               freq-sev-tot.loss-claim.incurred-length-height-
                           fuel.type,data=train.data11)

#model with stepAIC variable and collapsed categorical variable
#(model 5)
countmodel.nb.off2.4<-glm.nb(claim.count~business.type+driver.age+
                               driver.gender+yrs.licensed+ncd.level+
                               region1+vehicle.age+vehicle.value
                               +no.seats+horse.power+weight+width+prior.claims
                             + offset(log(exposure)),data=train.data11)
```

```
pred.nb.off2.1<-predict(countmodel.nb.off2.1,test.data,type='response')
pred.nb.off2.2<-predict(countmodel.nb.off2.2,test.data,type='response')
pred.nb.off2.3<-predict(countmodel.nb.off2.3,test.data11,type='response')
pred.nb.off2.4<-predict(countmodel.nb.off2.4,test.data11,type='response')
test.MSE1<-c(RMSE(test.data1$claim.count,pred.nb.off2.1),
             RMSE(test.data1$claim.count,pred.nb.off2.2),
             RMSE(test.data11$claim.count,pred.nb.off2.3),
             RMSE(test.data11$claim.count,pred.nb.off2.4))
k=5
set.seed(1)
P5<-cv.glm(train.data,countmodel.nb.off2.1,K=k)
P6<-cv.glm(train.data,countmodel.nb.off2.2,K=k)
P7<-cv.glm(train.data11,countmodel.nb.off2.3,K=k)
P8<-cv.glm(train.data11,countmodel.nb.off2.4,K=k)
cv.error2<-c(P5$delta[2],P6$delta[2],P7$delta[2],P8$delta[2])


tab2<-cbind(c(AIC(countmodel.nb.off),
      AIC(countmodel.nb.off2.1),
      AIC(countmodel.nb.off2.2),
      AIC(countmodel.nb.off2.3),
      AIC(countmodel.nb.off2.4)),
      c(BIC(countmodel.nb.off),
      BIC(countmodel.nb.off2.1),
      BIC(countmodel.nb.off2.2),
      BIC(countmodel.nb.off2.3),
      BIC(countmodel.nb.off2.4)))
colnames(tab2)<-c('AIC','BIC')
rownames(tab2)<-c('original model (1)',
                  'unimportant variables with |coef|>0.1 removed (2)',
                  'variables chosen via stepAIC (3)',
                  'model (2) with collapsed categorical variables (4)',
                  'model (3) with collapsed categorical variable (5)')
tab2<-cbind(tab2,cv.error=c(cv.error1[3],cv.error2),
            test.RMSE=c(test.MSE[3],test.MSE1) )
```

Table 3: performance of initial claim count models

	AIC	BIC	cv.error	test.RMSE
original model (1)	16772.74	17355.45	0.0865882	0.2965233
unimportant variables with |coef|>0.1 removed (2)	16770.26	17319.67	0.0867128	0.2964848
variables chosen via stepAIC (3)	16769.25	17310.33	0.0865121	0.2964425
model (2) with collapsed categorical variables (4)	16749.13	17190.32	0.0864774	0.2964787
model (3) with collapsed categorical variable (5)	16779.34	17128.96	0.0867151	0.2963585

From table 3 the new models outperform the original one in most criteria. AIC and BIC heavily favour the models with collapsed categorical variables as they have much less categorical levels. And cross validation and MSE show they perform as good as or even slightly better, indicating that combining levels may actually work. But I will continue using all 4 since as I could not come to any conclusion at this stage.

4.2.3 Modelling claim count with interaction

I look for interaction terms using the follow procedure. Important interacting variables from these two will be added to my model.

```
count.personal<-glm.nb(claim.count~(marital.status+driver.gender+driver.age)^2,
                       data=train.data1)
count.vehicle<-glm.nb(claim.count~(vehicle.age+vehicle.value+no.seats+cubic.cent
                                     +horse.power+weight+fuel.type
                                   +length+width+height)^2,
                      data=train.data)
```

For policyholders personal information, there's no noticeable interaction. But for vehicle information there are interaction between *width and height* of the vehicle and between *cubic.cent and weight* (*Please refer to Appendix 6.3.1 for the result*) These two interaction will be added to the 4 models above resulting in 8 models to consider

```
# model 2 with interaction (model 6)
countmodel.nb.off3.1<-glm.nb(claim.count~.+offset(log(exposure))-
      exposure-freq-sev-tot.loss-claim.incurred-length-height-
      fuel.type+weight:height+cubic.cent:weight,
                           data=train.data )
# model 3 with interaction (model 7)
countmodel.nb.off3.2<-glm.nb(claim.count~business.type + driver.age +
      driver.gender + marital.status +yrs.licensed + ncd.level+region +
      body.code + vehicle.age + vehicle.value + no.seats +
      horse.power+weight + width +prior.claims + offset(log(exposure))+
      weight:height+cubic.cent:weight,
      data=train.data)
# model 4 with interaction (model 8)
countmodel.nb.off3.3<-glm.nb(claim.count~.+offset(log(exposure))-
      exposure-freq-sev-tot.loss-claim.incurred-length-height-
      fuel.type+height:width+cubic.cent:weight,
                                data=train.data11)
# model 5 with interaction (model 9)
countmodel.nb.off3.4<-glm.nb(claim.count~business.type + driver.age + driver.gender +
                                  yrs.licensed + ncd.level + region1+ vehicle.age +
                                  vehicle.value + no.seats + horse.power + weight + width +
                                  prior.claims + offset(log(exposure))+
```

```
                                +height:width+cubic.cent:weight,
                            data=train.data11)

pred.nb.off3.1<-predict(countmodel.nb.off3.1,test.data,type='response')
pred.nb.off3.2<-predict(countmodel.nb.off3.2,test.data,type='response')
pred.nb.off3.3<-predict(countmodel.nb.off3.3,test.data11,type='response')
pred.nb.off3.4<-predict(countmodel.nb.off3.4,test.data11,type='response')
test.MSE2<-c(RMSE(test.data$claim.count,pred.nb.off3.1),
             RMSE(test.data$claim.count,pred.nb.off3.2),
             RMSE(test.data11$claim.count,pred.nb.off3.3),
             RMSE(test.data11$claim.count,pred.nb.off3.4))

k=5
set.seed(1)
P9<-cv.glm(train.data,countmodel.nb.off3.1,K=k)
P10<-cv.glm(train.data,countmodel.nb.off3.2,K=k)
P11<-cv.glm(train.data11,countmodel.nb.off3.3,K=k)
P12<-cv.glm(train.data11,countmodel.nb.off3.4,K=k)
cv.error3<-c(P9$delta[2],P10$delta[2],P11$delta[2],P12$delta[2])

tab3<-cbind(c(AIC(countmodel.nb.off3.1),
              AIC(countmodel.nb.off3.2),
              AIC(countmodel.nb.off3.3),
              AIC(countmodel.nb.off3.4)),
            c(BIC(countmodel.nb.off3.1),
              BIC(countmodel.nb.off3.2),
              BIC(countmodel.nb.off3.3),
              BIC(countmodel.nb.off3.4)))
colnames(tab3)<-c('AIC','BIC')
rownames(tab3)<-c('model (2) with interaction',
                  'model (3) with interaction',
                  'model (4) with interaction',
                  'model (5) with interaction')
tab3<-cbind(tab3,cv.error=cv.error3, test.RMSE=test.MSE2)
tab4<-rbind(tab2,tab3)
```

Table 4: performance of chosen variables claim count models

	AIC	BIC	cv.error	test.RMSE
original model (1)	16772.74	17355.45	0.0865882	0.2965233
unimportant variables with |coef|>0.1 removed (2)	16770.26	17319.67	0.0867128	0.2964848
variables chosen via stepAIC (3)	16769.25	17310.33	0.0865121	0.2964425
model (2) with collapsed categorical variables (4)	16749.13	17190.32	0.0864774	0.2964787
model (3) with collapsed categorical variable (5)	16779.34	17128.96	0.0867151	0.2963585
model (2) with interaction	16774.06	17340.12	0.0867225	0.2964830
model (3) with interaction	16772.19	17329.93	0.0865473	0.2964680
model (4) with interaction	16752.49	17210.33	0.0865216	0.2964692
model (5) with interaction	16778.09	17144.37	0.0867297	0.2963249

Looking at the table, it can be seen that the added interaction term do not improve the models, as they generally have higher AIC, BIC and lower cv errors. Although the for test error, we got mixed results when interaction terms are added, but this can also be caused by sampling errors. Consequently, the identified interaction may not be useful.

4.2.4 Chosen model

For modelling claim count and consequently the frequency of claim, I will choose the model with no PC, with unimportant variable with high coefficient removed, collapsed categorical variable and no interaction. This is because it performs the best in AIC and cross validation error, third best for BIC while its test error is not too high comparing to others model.

4.3 Severity

For severity, it is important to first extract from the train and test data the observation that did have a claim. The data for modelling is prepared as follow:

```
train.sev<-train.data%>%
  filter(sev>0 )

test.sev<-test.data%>%
  filter(sev>0)
train.sev1<-train.data1%>%
  filter(sev>0)

test.sev1<-test.data1%>%
  filter(sev>0)
```

4.3.1 Initial models

Since no offset in this model was identified, there are only 2 initial models.

```r
#Model with no PC
sev.modelgam<-glm(sev~.-claim.incurred-freq-claim.count-tot.loss,
                  data=train.sev,
                family=Gamma(link='log') )
#Model with PC
sev.modelgam1<-glm(sev~.-claim.incurred-freq-claim.count-tot.loss,
                   data=train.sev1,
                family=Gamma(link='log'))

set.seed(1)
P13<-cv.glm(train.sev,sev.modelgam,K=k)
P14<-cv.glm(train.sev1,sev.modelgam1,K=k)
cv.err.sev<-c(P13$delta[2],P14$delta[2])
pred.sevgam<-predict(sev.modelgam,test.sev,type='response')
pred.sevgam1<-predict(sev.modelgam1,test.sev1,type='response')
test.sev.MSE<-c(RMSE(test.sev$sev,pred.sevgam),
                RMSE(test.sev$sev,pred.sevgam1))
tab5<-cbind(c(AIC(sev.modelgam),
              AIC(sev.modelgam1)),
            c(BIC(sev.modelgam),
              BIC(sev.modelgam1)))
colnames(tab5)<-c('AIC','BIC')
rownames(tab5)<-c('model with no PC','model with PC')
tab5<-cbind(tab5,cv.error=cv.err.sev,test.RMSE=test.sev.MSE)
```

Table 5: performance initial severity models

	AIC	BIC	cv.error	test.RMSE
model with no PC	36136.07	36546.00	1225755	1101.726
model with PC	36134.39	36515.44	1207070	1097.571

Table 5 shows that, unlike with claim count, this time model with PC outperform the one with no PC in all criteria. Hence the second one will continue to be used for refining

But before moving on, it is worth checking the residuals plot resulted from these models.

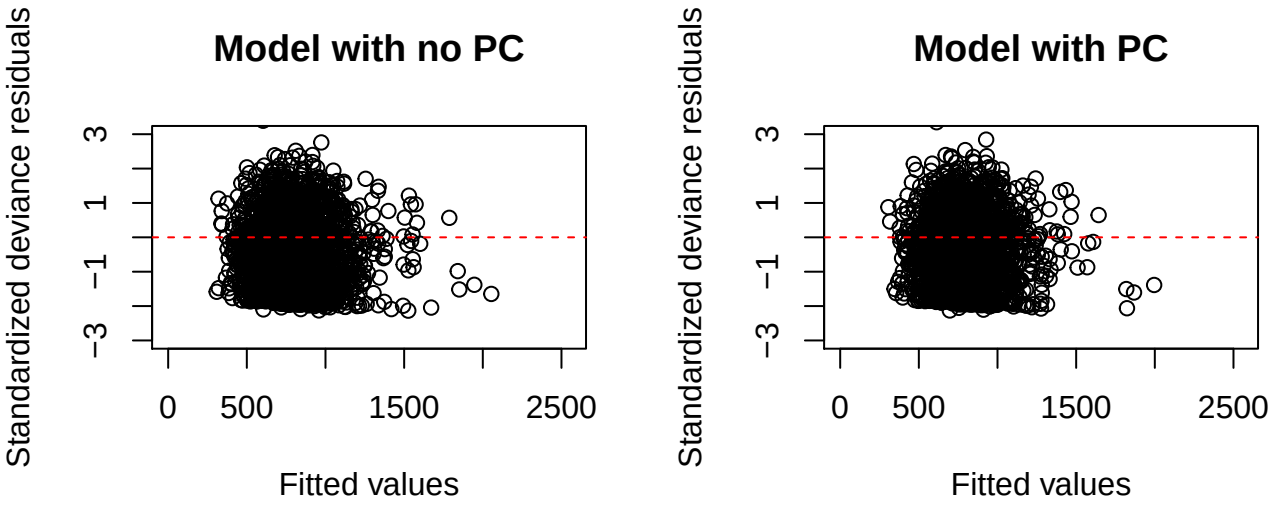

Figure 5: Fitted value vs Standardized deviance residuals

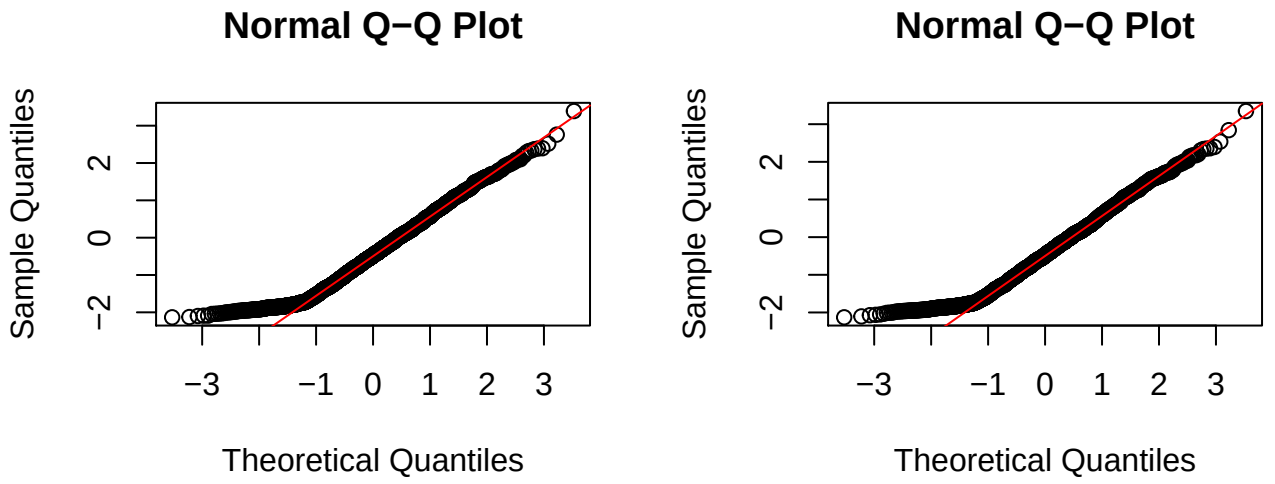

Figure 6: QQ plots

The random residuals pattern appear around the red line in figure 5 indicates that the underlying distribution, Gamma, is indeed a good one to be used. However, this distribution still has some flaw as shown in 6 where there is some deviation around the red lines at the left tails. This indicate that the distribution does not fit the data well at smaller claim amount.

4.3.2 Refining models with Feature selection

Unimportant variables with high coefficients and categorical features that are important on all of its levels for the severity model are body.code, exposure, marital.status, driver.gender, fuel.type. *These result and the resulted model from stepAIC() can be found in Appendix 6.2.2*

For combining levels, this time aside from region I also found that body.code and ncd.level also have levels that can be combined. (*Please refer to Appendix 6.4.2 for the tables that were used for this.*) They are combined as follow:

```
region2<-ifelse(gen.data1$region %in% c(30,21,34,12),'group1',
                ifelse(gen.data1$region %in% c(20,38,27,16,37),'group2',
                ifelse(gen.data1$region %in% c(5,19),'group3',
                ifelse(gen.data1$region %in% c(28,22),'group4',
                ifelse(gen.data1$region %in% c(1,25),'group5',
                ifelse(gen.data1$region %in% c(32,15,18), 'group6',gen.data1$region
                                                  ))))))

body.code2<-ifelse(gen.data1$body.code %in% c('C','G'),'group1',
                   gen.data1$body.code)
ncd.level2<-ifelse(gen.data1$ncd.level %in% c(4,6),'group1',
                   gen.data1$body.code)
region2<-as.factor(region2)
body.code2<-as.factor(body.code2)
ncd.level2<-as.factor(ncd.level2)
gen.data12<-cbind(gen.data1[,-c(7,8,9)],region2,body.code2,ncd.level2)
gen.data32<-cbind(gen.data3[,-c(7,8,9)],region2,body.code2,ncd.level2)

train.sev2<-gen.data32[index,]%>%
  filter(sev>0 )
test.sev2<-gen.data32[-index,]%>%
  filter(sev>0 )
```

Then the four models are built as follow

```
#model with unimportant variables with high coefficient removed (model 2)
sev.modelgam1.1<-glm(sev~.-claim.incurred-freq-claim.count-tot.loss-body.code
                     -exposure-marital.status-driver.gender-fuel.type,
                   data=train.sev1,
                   family=Gamma(link='log') )
#model with stepAIC variables (model 3)
sev.modelgam1.2<-glm(sev ~ exposure + ncd.level + PCage1,family=Gamma(link='log'),
                     data=train.sev1)
#model 2 with collapsed categorical variables
sev.modelgam1.3<-glm(sev~.-claim.incurred-freq-claim.count-tot.loss
                     -exposure-marital.status-driver.gender-fuel.type,
                     data=train.sev2,
                     family=Gamma(link='log'))
#model 3 with collapsed categorical variables
sev.modelgam1.4<-glm(sev ~ exposure + ncd.level2+ body.code2
                     +region2+ PCage1,family=Gamma(link='log'),
                     data=train.sev2)

k=5
set.seed(1)
```

```
P15<-cv.glm(train.sev1,sev.modelgam1.1,K=k)
P16<-cv.glm(train.sev1,sev.modelgam1.2,K=k)
P17<-cv.glm(train.sev2,sev.modelgam1.3,K=k)
P18<-cv.glm(train.sev2,sev.modelgam1.4,K=k)
cv.err.sev1<-c(P15$delta[2],P16$delta[2],P17$delta[2],P18$delta[2])


pred.sevgam1.1<-predict(sev.modelgam1.1,test.sev1,type='response')
pred.sevgam1.2<-predict(sev.modelgam1.2,test.sev1,type='response')
pred.sevgam1.3<-predict(sev.modelgam1.3,test.sev2,type='response')
pred.sevgam1.4<-predict(sev.modelgam1.4,test.sev2,type='response')
test.sev.MSE1<-c(RMSE(test.sev1$sev,pred.sevgam1.1),
                 RMSE(test.sev1$sev,pred.sevgam1.2),
                 RMSE(test.sev2$sev,pred.sevgam1.3),
                 RMSE(test.sev2$sev,pred.sevgam1.4))

tab6<-cbind(c(AIC(sev.modelgam1),
              AIC(sev.modelgam1.1),
              AIC(sev.modelgam1.2),
              AIC(sev.modelgam1.3),
              AIC(sev.modelgam1.4)),
            c(BIC(sev.modelgam1),
              BIC(sev.modelgam1.1),
              BIC(sev.modelgam1.2),
              BIC(sev.modelgam1.3),
              BIC(sev.modelgam1.4))
            )
colnames(tab6)<-c('AIC','BIC' )
rownames(tab6)<-c('original model (1)',
                  'unimportant variable with |coef|>0.1 removed (2)',
                  'variables chosen using stepAIC (3)',
                  'model 2 with collapsed categorical variables (4)',
                  'model 3 with collapsed categorical variables (5)')
tab6<-cbind(tab6,cv.error=c(P14$delta[2],cv.err.sev1),
            test.RMSE=c(test.sev.MSE[2],test.sev.MSE1 ))
```

Table 6: performance severity models with feature selection

	AIC	BIC	cv.error	test.RMSE
original model (1)	36134.39	36515.44	1207070	1097.571
unimportant variable with |coef|>0.1 removed (2)	36122.88	36423.10	1212597	1095.607
variables chosen using stepAIC (3)	36075.26	36127.23	1179583	1093.633
model 2 with collapsed categorical variables (4)	36114.60	36397.51	1195214	1094.410
model 3 with collapsed categorical variables (5)	36106.38	36354.65	1194551	1092.856

Based on the result in table 6 there are strong indication that collapsing category levels may make models with stepAIC perform worse as the cross validation, AIC and BIC increased when categorical features are collapsed. On the other hand it seems to improve the model with only high coefficient, unimportant variables removed. Yet all 4 will still be considered

4.3.3 With interaction

Finding interaction term will be carried out similarly to the previous model. *And the result can be found in Appendix 6.3.2*

```
sev.personal<-glm(sev~(marital.status+driver.gender+driver.age)^2,
                  data=train.sev1,
                  family=Gamma(link='log') )
sev.vehicle<-glm(sev~(PCage1+ PCsize1+PCsize2+PCsize3+fuel.type)^2,
                     data=train.sev1,
                  family=Gamma(link='log') )
```

For severity, there are no noticeable interaction from these two models. As the interaction between `PCage1` and cars run on gasoline are close to be important, the interaction between `PCage1` and `fuel.type` will be added for testing (I tested more term than supposed to as I suspect they will all be insignificant)

```
#model 2 with interaction
sev.modelgam2.1<-glm(sev~.-claim.incurred-freq-claim.count-tot.loss-body.code
                     -exposure-marital.status-driver.gender-fuel.type
                     +PCage1:fuel.type,data=train.sev1,family=Gamma(link='log') )

#model 3 with interaction
sev.modelgam2.2<-glm(sev~exposure + ncd.level + PCage1+PCage1:fuel.type,
                     data=train.sev1,family=Gamma(link='log') )

#model 4 with interaction
sev.modelgam2.3<-glm(sev~.-claim.incurred-freq-claim.count-tot.loss
                     -exposure-marital.status-driver.gender-fuel.type
 +PCage1:fuel.type,data=train.sev2,family=Gamma(link='log'))

#model 5 with interaction
sev.modelgam2.4<-glm(sev ~ exposure + ncd.level2+ body.code2
                     +region2+ PCage1+PCage1:fuel.type,
                     family=Gamma(link='log'),data=train.sev2)

tab7<-cbind(c(AIC(sev.modelgam2.1),
              AIC(sev.modelgam2.2),
              AIC(sev.modelgam2.3),
              AIC(sev.modelgam2.4)),
```

```
              c(BIC(sev.modelgam2.1),
              BIC(sev.modelgam2.2),
              BIC(sev.modelgam2.3),
              BIC(sev.modelgam2.4)
            )
)
colnames(tab7)<-c('AIC','BIC')
rownames(tab7)<-c('(2) with interaction effect (6)',
                  '(3) with interaction effect (7)',
                  '(4) with interaction effect (8)',
                  '(5) with interaction effect (9)')
set.seed(1)
P19<-cv.glm(train.sev1,sev.modelgam2.1,K=k)
P20<-cv.glm(train.sev1,sev.modelgam2.2,K=k)
P21<-cv.glm(train.sev2,sev.modelgam2.3,K=k)
P22<-cv.glm(train.sev2,sev.modelgam2.4,K=k)


cv.err.sev2<-c(P19$delta[2],P20$delta[2],P21$delta[2],P22$delta[2])

pred.sevgam2.1<-predict(sev.modelgam2.1,test.sev1,type='response')
pred.sevgam2.2<-predict(sev.modelgam2.2,test.sev1,type='response')
pred.sevgam2.3<-predict(sev.modelgam2.3,test.sev2,type='response')
pred.sevgam2.4<-predict(sev.modelgam2.4,test.sev2,type='response')


test.sev.MSE2<-c(RMSE(test.sev1$sev,pred.sevgam2.1),
                 RMSE(test.sev1$sev,pred.sevgam2.2),
                 RMSE(test.sev2$sev,pred.sevgam2.3),
                 RMSE(test.sev2$sev,pred.sevgam2.4))
tab7<-cbind(tab7,cv.error=cv.err.sev2,test.RMSE=test.sev.MSE2)
tab8<-rbind(tab6[,],tab7 )
```

Table 7: performance finals candidate severity models

	AIC	BIC	cv.error	test.RMSE
original model (1)	36134.39	36515.44	1207070	1097.571
unimportant variable with |coef|>0.1 removed (2)	36122.88	36423.10	1212597	1095.607
variables chosen using stepAIC (3)	36075.26	36127.23	1179583	1093.633
model 2 with collapsed categorical variables (4)	36114.60	36397.51	1195214	1094.410
model 3 with collapsed categorical variables (5)	36106.38	36354.65	1194551	1092.856
(2) with interaction effect (6)	36126.41	36438.18	1216495	1096.270
(3) with interaction effect (7)	36078.87	36142.38	1180565	1094.152
(4) with interaction effect (8)	36118.23	36412.68	1196636	1094.959
(5) with interaction effect (9)	36109.97	36369.78	1195234	1093.603

Looking at table 7, we can see that the interaction term have no noticeable impact on the model 2, but it does improve model 3 slightly in cross validation error and test error.

4.3.4 Chosen model

For severity, based on the result, I will choose the model that use PC variables, and stepAIC for variable selection. This is because the model perform the best cross validation error, AIC and BIC. However, the fact that stepAIC removes most variable indicate that very few features have good predictive variable for severity. Hence, despite having adequate underlying distribution, this models may still perform poorly.

5 Final result

```
final.pred.freq<-predict(countmodel.nb.off2.3,test.data11,
                         type='response') / test.data11$exposure
final.pred.sev<-predict(sev.modelgam1.2,test.data1,type='response')

final.pred.prem<-final.pred.freq*final.pred.sev

testMSE.prem<-sqrt(mean((final.pred.prem-test.data1$tot.loss)^2))
postResample(pred=final.pred.prem,obs=test.data$tot.loss )
```

```
##          RMSE      Rsquared           MAE
## 8.964600e+02 7.876947e-03 2.348209e+02
```

The final result indicate that using GLM with negative binomial distribution to model freqency and using GLM with gamma distribution to model severity may not be the best option to predict the pure premium for its high error and close to 0 R squared. However, this is probably due to

negative binomial may not be a good assumption of claim count distribution, and for severity, input may not have predictive power to predict this response. However, looking from another perspective, the total claim incurred on the test data set is 6.6068536×10^5, while the predicted sum is 6.7477799×10^5. And from this perspective, the model seem to perform relatively well.

5.1 Limitation

Negative binomials does not seem to be an appropriate distribution for claim count. Furthermore, another problem is that the dataset does not seem to contain many individual variables with strong predictive powers for both frequency and severity. Furthermore, predictors that are likely to have interaction effect was not identified in my models. This is because limited pairwise interaction can be tested (because of the lack of computational power and numerous variables at hand).

Hence other to improve this GLM I recommend using another underlying distribution for claim count, which is worked on by my teammate. And more interaction effect should be tested.

6 Appendix

6.1 Exploration

6.1.1 Frequency

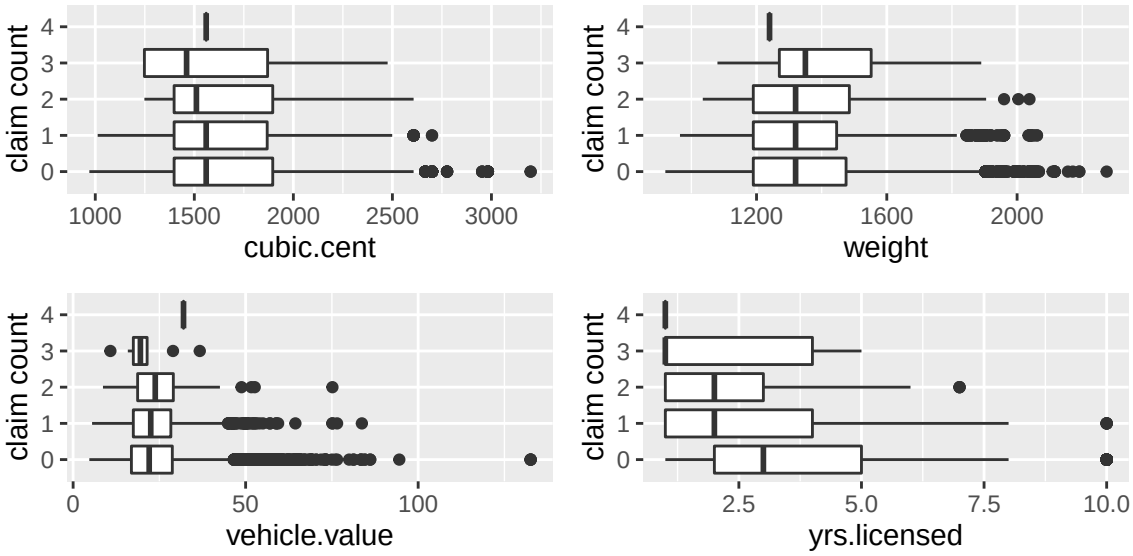

Figure 7: claim count against some numerical feature

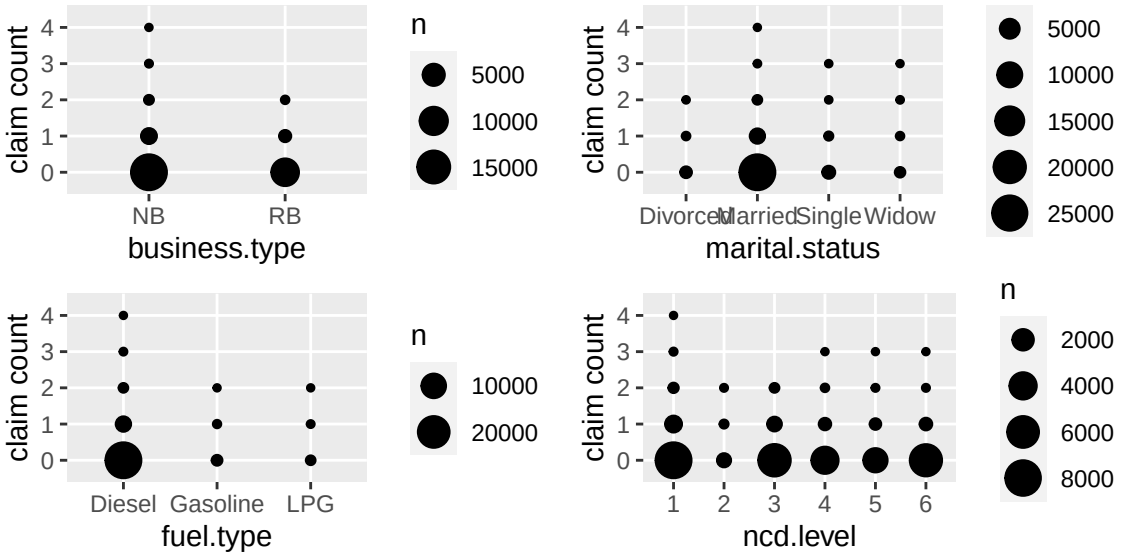

Figure 8: claim count against some categorical feature

6.1.2 Severity

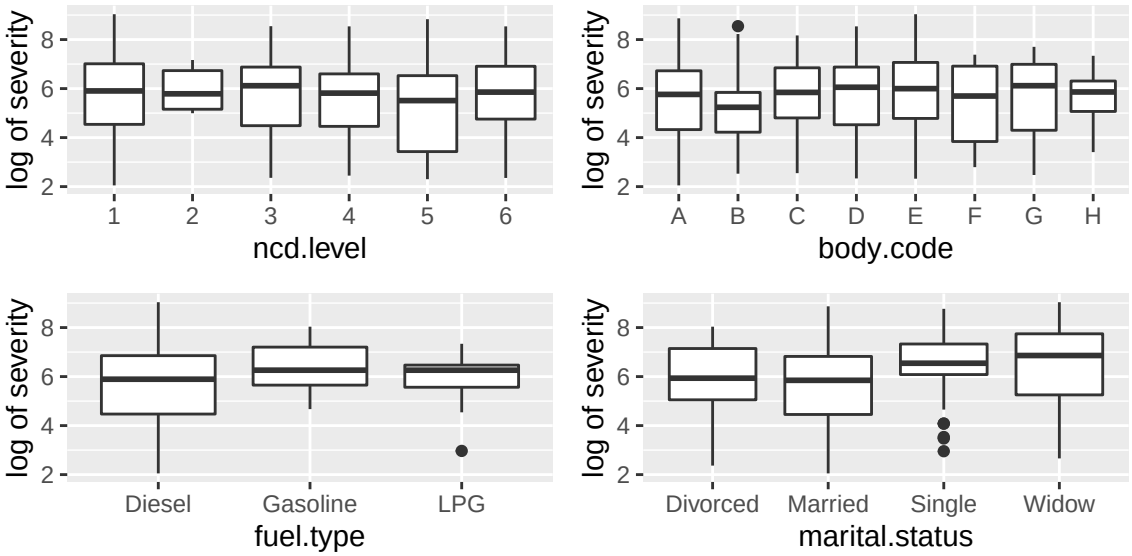

Figure 9: severity against some categorical feature

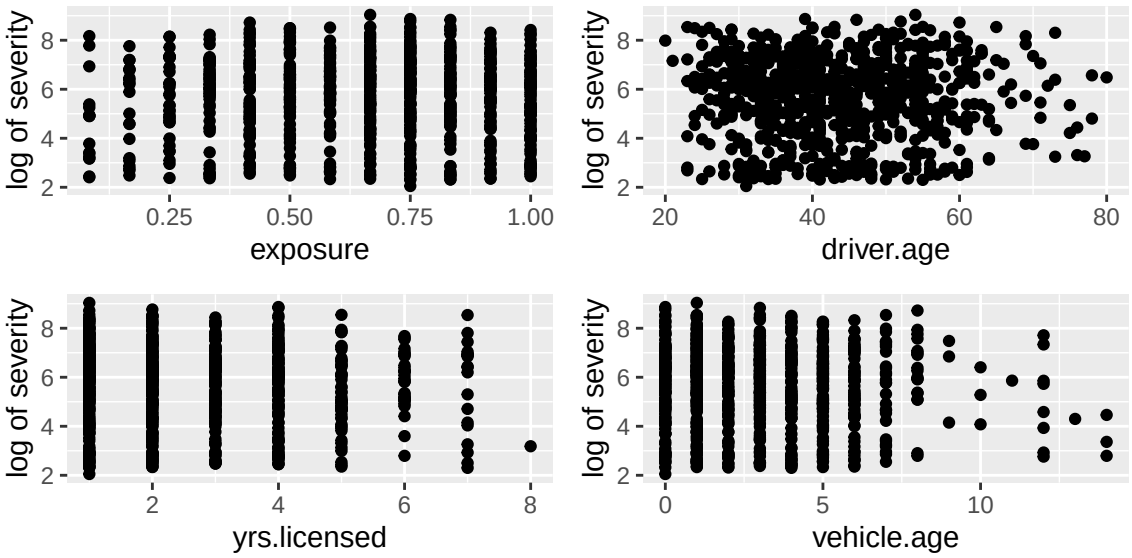

Figure 10: severity against some numerical features

6.2 Feature selection

6.2.1 For frequency

```
#Looking for unimportant variables with high coefficient
summary(countmodel.nb.off)
```

```
##
## Call:
## glm.nb(formula = claim.count ~ . + offset(log(exposure)) - exposure -
##     freq - sev - tot.loss - claim.incurred, data = train.data,
##     init.theta = 7.309262387, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2451  -0.4572  -0.3393  -0.2223   3.9289
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -1.889e+00  6.114e-01  -3.089 0.002006 **
## business.typeRB       -2.209e-01  5.365e-02  -4.119 3.81e-05 ***
## driver.age            -5.081e-03  2.039e-03  -2.492 0.012713 *
## driver.genderMale     -1.702e-01  6.018e-02  -2.828 0.004678 **
## marital.statusMarried  1.022e-01  1.369e-01   0.746 0.455387
## marital.statusSingle   7.359e-02  1.663e-01   0.443 0.658037
## marital.statusWidow    6.680e-01  1.955e-01   3.416 0.000634 ***
## yrs.licensed          -6.446e-02  1.634e-02  -3.946 7.96e-05 ***
## ncd.level2            -4.976e-01  1.688e-01  -2.948 0.003200 **
## ncd.level3            -1.472e-01  5.256e-02  -2.800 0.005110 **
## ncd.level4            -3.772e-01  7.318e-02  -5.154 2.55e-07 ***
## ncd.level5            -3.938e-01  8.843e-02  -4.453 8.48e-06 ***
## ncd.level6            -5.751e-01  8.389e-02  -6.856 7.10e-12 ***
## region2               -5.133e-01  2.463e-01  -2.084 0.037180 *
## region3               -3.987e-02  1.209e-01  -0.330 0.741622
## region4               -1.475e-01  1.271e-01  -1.160 0.246005
## region5               -6.810e-02  1.909e-01  -0.357 0.721207
## region6                2.056e-01  1.907e-01   1.078 0.280909
## region7               -1.149e+00  3.917e-01  -2.935 0.003339 **
## region8                3.990e-01  1.206e-01   3.310 0.000934 ***
## region9                4.252e-01  1.896e-01   2.242 0.024964 *
## region10               1.619e-02  1.656e-01   0.098 0.922101
## region11              -5.096e-02  2.636e-01  -0.193 0.846694
## region12              -1.154e-01  2.012e-01  -0.574 0.566249
## region13              -1.313e-01  1.796e-01  -0.731 0.464599
## region14              -1.739e-02  1.709e-01  -0.102 0.918951
## region15              -5.411e-02  1.311e-01  -0.413 0.679899
## region16               7.510e-02  2.153e-01   0.349 0.727201
## region17               2.171e-01  1.067e-01   2.034 0.041991 *
## region18               2.925e-01  1.305e-01   2.241 0.025046 *
## region19              -1.460e-01  2.121e-01  -0.688 0.491225
## region20              -5.315e-01  2.701e-01  -1.968 0.049067 *
## region21               9.995e-02  2.091e-01   0.478 0.632641
```

```
## region22             -3.366e-01  2.961e-01  -1.137 0.255674
## region23             -2.365e-01  2.238e-01  -1.057 0.290668
## region24              1.134e-01  2.290e-01   0.495 0.620346
## region25              2.320e-01  1.421e-01   1.632 0.102687
## region26              1.925e-01  1.980e-01   0.973 0.330762
## region27             -3.893e-01  2.172e-01  -1.792 0.073089 .
## region28             -3.777e-01  2.633e-01  -1.435 0.151407
## region29             -1.107e-01  2.060e-01  -0.537 0.591183
## region30             -4.360e-02  2.261e-01  -0.193 0.847100
## region31              6.158e-02  2.637e-01   0.234 0.815365
## region32              1.515e-01  1.367e-01   1.108 0.267905
## region33             -6.607e-02  2.367e-01  -0.279 0.780143
## region34             -2.137e-01  2.412e-01  -0.886 0.375589
## region35             -1.751e-01  1.711e-01  -1.024 0.306066
## region36              1.485e-01  1.694e-01   0.877 0.380631
## region37             -3.931e-02  1.992e-01  -0.197 0.843555
## region38              8.801e-02  2.516e-01   0.350 0.726508
## body.codeB           -9.201e-01  2.602e-01  -3.536 0.000406 ***
## body.codeC            1.533e-01  1.397e-01   1.097 0.272515
## body.codeD           -4.311e-02  7.977e-02  -0.540 0.588938
## body.codeE            2.044e-02  6.481e-02   0.315 0.752487
## body.codeF           -3.197e-01  2.350e-01  -1.360 0.173727
## body.codeG            2.825e-01  1.074e-01   2.631 0.008522 **
## body.codeH            4.122e-01  2.127e-01   1.938 0.052617 .
## vehicle.age          -5.277e-02  1.410e-02  -3.743 0.000182 ***
## vehicle.value        -1.482e-02  4.250e-03  -3.487 0.000488 ***
## no.seats             -3.118e-02  2.482e-02  -1.256 0.208993
## cubic.cent            8.457e-05  1.001e-04   0.845 0.398139
## horse.power           3.346e-03  1.960e-03   1.707 0.087755 .
## weight               -4.315e-04  2.328e-04  -1.853 0.063860 .
## length                1.382e-01  1.164e-01   1.187 0.235264
## width                 8.135e-01  2.460e-01   3.306 0.000945 ***
## height               -3.399e-01  3.012e-01  -1.129 0.259001
## fuel.typeGasoline    -2.908e-01  1.795e-01  -1.620 0.105333
## fuel.typeLPG         -2.835e-01  2.937e-01  -0.965 0.334392
## prior.claims          1.392e-01  1.243e-02  11.203  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(7.3093) family taken to be 1)
##
##     Null deviance: 12073  on 30466  degrees of freedom
## Residual deviance: 11428  on 30398  degrees of freedom
## AIC: 16773
##
```

```
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  7.31
##           Std. Err.:  4.43
##
##  2 x log-likelihood:  -16632.74
```

```
#Feature selection with stepAIC
countmodel.none<-glm.nb(claim.count~1,data=train.data)
step.choose<-stepAIC(countmodel.nb.off,
                     direction = "backward",
                     k = 2,
                     scope = list(upper = countmodel.nb.off, lower = countmodel.none)
)
```

```
## Start:  AIC=16770.74
## claim.count ~ exposure + business.type + driver.age + driver.gender +
##     marital.status + yrs.licensed + ncd.level + region + body.code +
##     vehicle.age + vehicle.value + no.seats + cubic.cent + horse.power +
##     weight + length + width + height + fuel.type + prior.claims +
##     claim.incurred + freq + sev + tot.loss + offset(log(exposure)) -
##     exposure - freq - sev - tot.loss - claim.incurred
##
##                   Df   AIC
## - cubic.cent       1 16770
## - height           1 16770
## - length           1 16770
## - no.seats         1 16770
## - fuel.type        2 16770
## <none>               16771
## - horse.power      1 16772
## - weight           1 16772
## - driver.age       1 16775
## - driver.gender    1 16776
## - marital.status   3 16779
## - width            1 16779
## - vehicle.value    1 16781
## - vehicle.age      1 16783
## - yrs.licensed     1 16784
## - business.type    1 16786
## - body.code        7 16788
## - region          37 16814
## - ncd.level        5 16822
## - prior.claims     1 16871
```

```
##
## Step:  AIC=16769.45
## claim.count ~ business.type + driver.age + driver.gender + marital.status +
##     yrs.licensed + ncd.level + region + body.code + vehicle.age +
##     vehicle.value + no.seats + horse.power + weight + length +
##     width + height + fuel.type + prior.claims + offset(log(exposure))
##
##                   Df   AIC
## - height           1 16768
## - length           1 16769
## <none>               16770
## - fuel.type        2 16770
## - no.seats         1 16770
## - weight           1 16770
## - horse.power      1 16772
## - driver.age       1 16774
## - driver.gender    1 16775
## - marital.status   3 16778
## - width            1 16778
## - vehicle.age      1 16781
## - vehicle.value    1 16781
## - yrs.licensed     1 16783
## - business.type    1 16785
## - body.code        7 16786
## - region          37 16812
## - ncd.level        5 16821
## - prior.claims     1 16869
##
## Step:  AIC=16768.41
## claim.count ~ business.type + driver.age + driver.gender + marital.status +
##     yrs.licensed + ncd.level + region + body.code + vehicle.age +
##     vehicle.value + no.seats + horse.power + weight + length +
##     width + fuel.type + prior.claims + offset(log(exposure))
##
##                   Df   AIC
## - length           1 16768
## - fuel.type        2 16768
## <none>               16768
## - no.seats         1 16769
## - horse.power      1 16770
## - weight           1 16771
## - driver.age       1 16773
## - driver.gender    1 16774
## - marital.status   3 16776
## - width            1 16777
```

```
## - vehicle.age     1 16781
## - vehicle.value   1 16781
## - yrs.licensed    1 16782
## - business.type   1 16784
## - body.code       7 16792
## - region         37 16811
## - ncd.level       5 16820
## - prior.claims    1 16868
##
## Step:  AIC=16767.8
## claim.count ~ business.type + driver.age + driver.gender + marital.status +
##     yrs.licensed + ncd.level + region + body.code + vehicle.age +
##     vehicle.value + no.seats + horse.power + weight + width +
##     fuel.type + prior.claims + offset(log(exposure))
##
##                  Df   AIC
## - fuel.type       2 16767
## <none>              16768
## - no.seats        1 16768
## - weight          1 16769
## - horse.power     1 16770
## - driver.age      1 16772
## - driver.gender   1 16774
## - marital.status  3 16776
## - width           1 16778
## - vehicle.age     1 16780
## - vehicle.value   1 16781
## - yrs.licensed    1 16782
## - business.type   1 16783
## - body.code       7 16791
## - region         37 16811
## - ncd.level       5 16820
## - prior.claims    1 16868
##
## Step:  AIC=16767.25
## claim.count ~ business.type + driver.age + driver.gender + marital.status +
##     yrs.licensed + ncd.level + region + body.code + vehicle.age +
##     vehicle.value + no.seats + horse.power + weight + width +
##     prior.claims + offset(log(exposure))
##
##                  Df   AIC
## <none>              16767
## - weight          1 16767
## - no.seats        1 16768
## - horse.power     1 16769
```

```
## - driver.age      1 16771
## - driver.gender   1 16773
## - marital.status  3 16776
## - width           1 16777
## - vehicle.value   1 16779
## - vehicle.age     1 16779
## - yrs.licensed    1 16781
## - business.type   1 16782
## - body.code       7 16790
## - region         37 16810
## - ncd.level       5 16819
## - prior.claims    1 16868
```

6.2.2 For severity

```
#Feature selection with stepAIC
sev.model.none<-glm(sev~1,
                    data=train.sev1,
                    family=Gamma(link='log'))
step.choose1<-stepAIC(sev.modelgam1,
                      direction = "backward",
                      k = 2,
                      scope = list(upper = sev.modelgam1,
                                   lower = sev.model.none)
)
```

```
## Start:  AIC=36134.39
## sev ~ (exposure + business.type + driver.age + driver.gender +
##     marital.status + yrs.licensed + ncd.level + region + body.code +
##     fuel.type + prior.claims + claim.count + claim.incurred +
##     freq + tot.loss + PCsize1 + PCsize2 + PCsize3 + PCage1) -
##     claim.incurred - freq - claim.count - tot.loss
##
##                  Df Deviance   AIC
## - region         37   4763.9 36093
## - body.code       7   4713.7 36125
## - marital.status  3   4711.9 36132
## - business.type   1   4705.3 36132
## - prior.claims    1   4705.4 36132
## - yrs.licensed    1   4705.4 36132
## - PCsize1         1   4705.6 36133
## - fuel.type       2   4709.4 36133
## - PCsize3         1   4706.0 36133
## - PCsize2         1   4706.1 36133
```

```
## - driver.age      1   4706.9 36133
## - ncd.level       5   4721.5 36133
## <none>                4705.3 36134
## - exposure        1   4709.2 36135
## - driver.gender   1   4709.6 36135
## - PCage1          1   4709.8 36135
##
## Step:  AIC=36097.75
## sev ~ exposure + business.type + driver.age + driver.gender +
##     marital.status + yrs.licensed + ncd.level + body.code + fuel.type +
##     prior.claims + PCsize1 + PCsize2 + PCsize3 + PCage1
##
##                  Df Deviance   AIC
## - body.code       7   4772.1 36088
## - marital.status  3   4770.2 36095
## - business.type   1   4764.0 36096
## - prior.claims    1   4764.0 36096
## - PCsize1         1   4764.1 36096
## - yrs.licensed    1   4764.2 36096
## - PCsize3         1   4764.5 36096
## - PCsize2         1   4765.1 36096
## - driver.age      1   4765.2 36096
## - fuel.type       2   4769.0 36096
## - ncd.level       5   4780.2 36097
## <none>                4763.9 36098
## - exposure        1   4767.7 36098
## - driver.gender   1   4767.7 36098
## - PCage1          1   4768.4 36098
##
## Step:  AIC=36088.94
## sev ~ exposure + business.type + driver.age + driver.gender +
##     marital.status + yrs.licensed + ncd.level + fuel.type + prior.claims +
##     PCsize1 + PCsize2 + PCsize3 + PCage1
##
##                  Df Deviance   AIC
## - marital.status  3   4778.4 36086
## - ncd.level       5   4786.9 36087
## - prior.claims    1   4772.3 36087
## - yrs.licensed    1   4772.3 36087
## - business.type   1   4772.4 36087
## - PCsize3         1   4772.4 36087
## - PCsize1         1   4772.5 36087
## - driver.age      1   4773.0 36087
## - fuel.type       2   4777.0 36088
## - PCsize2         1   4773.4 36088
```

```
## - driver.gender   1   4775.4 36089
## <none>                4772.1 36089
## - exposure        1   4775.8 36089
## - PCage1          1   4776.1 36089
##
## Step:  AIC=36086.93
## sev ~ exposure + business.type + driver.age + driver.gender +
##     yrs.licensed + ncd.level + fuel.type + prior.claims + PCsize1 +
##     PCsize2 + PCsize3 + PCage1
##
##                 Df Deviance   AIC
## - prior.claims   1   4778.6 36085
## - ncd.level      5   4793.4 36085
## - PCsize3        1   4778.6 36085
## - business.type  1   4778.7 36085
## - yrs.licensed   1   4778.7 36085
## - PCsize1        1   4778.9 36085
## - fuel.type      2   4783.1 36086
## - PCsize2        1   4779.5 36086
## - driver.gender  1   4780.5 36086
## - driver.age     1   4781.2 36086
## - exposure       1   4781.9 36087
## <none>               4778.4 36087
## - PCage1         1   4782.2 36087
##
## Step:  AIC=36085.04
## sev ~ exposure + business.type + driver.age + driver.gender +
##     yrs.licensed + ncd.level + fuel.type + PCsize1 + PCsize2 +
##     PCsize3 + PCage1
##
##                 Df Deviance   AIC
## - yrs.licensed   1   4778.7 36083
## - ncd.level      5   4793.6 36083
## - PCsize3        1   4778.8 36083
## - business.type  1   4778.9 36083
## - PCsize1        1   4779.1 36083
## - fuel.type      2   4783.3 36084
## - PCsize2        1   4779.6 36084
## - driver.gender  1   4780.7 36084
## - driver.age     1   4781.3 36085
## - exposure       1   4782.1 36085
## <none>               4778.6 36085
## - PCage1         1   4782.4 36085
##
## Step:  AIC=36083.16
```

```
## sev ~ exposure + business.type + driver.age + driver.gender +
##     ncd.level + fuel.type + PCsize1 + PCsize2 + PCsize3 + PCage1
##
##                 Df Deviance   AIC
## - PCsize3        1   4779.0 36081
## - business.type  1   4779.1 36081
## - PCsize1        1   4779.3 36081
## - fuel.type      2   4783.4 36082
## - PCsize2        1   4779.8 36082
## - driver.gender  1   4780.9 36082
## - driver.age     1   4781.3 36083
## - ncd.level      5   4797.0 36083
## - exposure       1   4782.4 36083
## <none>               4778.7 36083
## - PCage1         1   4782.7 36083
##
## Step:  AIC=36081.34
## sev ~ exposure + business.type + driver.age + driver.gender +
##     ncd.level + fuel.type + PCsize1 + PCsize2 + PCage1
##
##                 Df Deviance   AIC
## - business.type  1   4779.4 36080
## - PCsize1        1   4779.6 36080
## - PCsize2        1   4780.0 36080
## - fuel.type      2   4784.1 36080
## - driver.gender  1   4781.1 36080
## - driver.age     1   4781.5 36081
## - ncd.level      5   4797.1 36081
## - exposure       1   4782.6 36081
## <none>               4779.0 36081
## - PCage1         1   4783.3 36082
##
## Step:  AIC=36079.59
## sev ~ exposure + driver.age + driver.gender + ncd.level + fuel.type +
##     PCsize1 + PCsize2 + PCage1
##
##                 Df Deviance   AIC
## - PCsize1        1   4780.0 36078
## - PCsize2        1   4780.4 36078
## - fuel.type      2   4784.4 36078
## - driver.gender  1   4781.5 36079
## - driver.age     1   4782.2 36079
## - exposure       1   4782.9 36079
## <none>               4779.4 36080
## - PCage1         1   4783.5 36080
```

```
## - ncd.level      5   4802.3 36082
##
## Step:  AIC=36077.95
## sev ~ exposure + driver.age + driver.gender + ncd.level + fuel.type +
##     PCsize2 + PCage1
##
##                 Df Deviance   AIC
## - PCsize2        1   4781.0 36076
## - fuel.type      2   4785.3 36077
## - driver.gender  1   4782.0 36077
## - driver.age     1   4782.7 36077
## - exposure       1   4783.3 36078
## <none>               4780.0 36078
## - PCage1         1   4783.7 36078
## - ncd.level      5   4802.5 36080
##
## Step:  AIC=36076.58
## sev ~ exposure + driver.age + driver.gender + ncd.level + fuel.type +
##     PCage1
##
##                 Df Deviance   AIC
## - driver.gender  1   4783.0 36076
## - fuel.type      2   4786.8 36076
## - driver.age     1   4783.8 36076
## - exposure       1   4784.4 36076
## <none>               4781.0 36077
## - PCage1         1   4787.0 36078
## - ncd.level      5   4803.5 36079
##
## Step:  AIC=36075.87
## sev ~ exposure + driver.age + ncd.level + fuel.type + PCage1
##
##              Df Deviance   AIC
## - fuel.type   2   4789.0 36075
## - driver.age  1   4785.9 36075
## - exposure    1   4786.6 36076
## <none>            4783.0 36076
## - PCage1      1   4789.5 36077
## - ncd.level   5   4806.3 36078
##
## Step:  AIC=36075.61
## sev ~ exposure + driver.age + ncd.level + PCage1
##
##              Df Deviance   AIC
## - driver.age  1   4791.6 36075
```

```
## <none>             4789.0 36076
## - exposure     1   4792.8 36076
## - PCage1       1   4794.5 36077
## - ncd.level    5   4811.7 36078
##
## Step:  AIC=36075.26
## sev ~ exposure + ncd.level + PCage1
##
##             Df Deviance   AIC
## - exposure   1   4795.2 36075
## <none>           4791.6 36075
## - PCage1     1   4797.0 36076
## - ncd.level  5   4818.1 36079
##
## Step:  AIC=36075.55
## sev ~ ncd.level + PCage1
```

```
#Looking for unimportant variables with high coefficient
summary(sev.modelgam1 )
```

```
##
## Call:
## glm(formula = sev ~ . - claim.incurred - freq - claim.count -
##     tot.loss, family = Gamma(link = "log"), data = train.sev1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8000  -1.6101  -0.6367   0.2866   4.4765
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)             6.878415   0.271346  25.349  < 2e-16 ***
## exposure               -0.182101   0.118221  -1.540  0.12361
## business.typeRB         0.004235   0.075480   0.056  0.95526
## driver.age             -0.002624   0.002804  -0.936  0.34954
## driver.genderMale      -0.133060   0.083874  -1.586  0.11278
## marital.statusMarried   0.170043   0.191911   0.886  0.37568
## marital.statusSingle    0.289106   0.231166   1.251  0.21119
## marital.statusWidow    -0.158949   0.278362  -0.571  0.56804
## yrs.licensed            0.006326   0.022473   0.281  0.77836
## ncd.level2             -0.263807   0.244345  -1.080  0.28041
## ncd.level3             -0.119505   0.074835  -1.597  0.11042
## ncd.level4             -0.223360   0.102772  -2.173  0.02985 *
## ncd.level5             -0.241187   0.124861  -1.932  0.05353 .
## ncd.level6             -0.311461   0.118110  -2.637  0.00842 **
```

```
## region2              0.816098   0.339071   2.407  0.01617 *
## region3             -0.061039   0.166909  -0.366  0.71462
## region4              0.218806   0.174819   1.252  0.21084
## region5              0.082183   0.271298   0.303  0.76197
## region6              0.249380   0.265490   0.939  0.34766
## region7              0.304371   0.529041   0.575  0.56513
## region8              0.095902   0.167303   0.573  0.56655
## region9              0.260615   0.260765   0.999  0.31769
## region10            -0.007181   0.227034  -0.032  0.97477
## region11            -0.480132   0.355109  -1.352  0.17649
## region12             0.009687   0.278256   0.035  0.97223
## region13             0.341896   0.246939   1.385  0.16633
## region14            -0.276263   0.240744  -1.148  0.25128
## region15             0.112266   0.180039   0.624  0.53297
## region16            -0.089335   0.292835  -0.305  0.76034
## region17             0.022293   0.147354   0.151  0.87976
## region18             0.182476   0.180611   1.010  0.31245
## region19             0.234608   0.286809   0.818  0.41345
## region20             0.139746   0.368081   0.380  0.70423
## region21            -0.322491   0.284728  -1.133  0.25749
## region22             0.360905   0.414216   0.871  0.38368
## region23             0.067634   0.312570   0.216  0.82871
## region24            -0.070014   0.331412  -0.211  0.83270
## region25             0.003460   0.197398   0.018  0.98602
## region26             0.090994   0.282055   0.323  0.74702
## region27            -0.070949   0.302332  -0.235  0.81448
## region28             0.142078   0.363758   0.391  0.69614
## region29            -0.097228   0.284470  -0.342  0.73254
## region30            -0.153613   0.331934  -0.463  0.64356
## region31            -0.408150   0.365783  -1.116  0.26461
## region32             0.248199   0.189119   1.312  0.18952
## region33             0.012307   0.332354   0.037  0.97047
## region34             0.063031   0.346538   0.182  0.85569
## region35            -0.073563   0.236049  -0.312  0.75534
## region36            -0.043443   0.235999  -0.184  0.85397
## region37             0.060440   0.272318   0.222  0.82438
## region38            -0.192238   0.347354  -0.553  0.58002
## body.codeB          -0.150237   0.343948  -0.437  0.66230
## body.codeC           0.073234   0.192509   0.380  0.70367
## body.codeD           0.013485   0.096187   0.140  0.88852
## body.codeE           0.059407   0.085255   0.697  0.48599
## body.codeF          -0.550953   0.310666  -1.773  0.07628 .
## body.codeG           0.022568   0.154699   0.146  0.88403
## body.codeH           0.227561   0.296660   0.767  0.44311
## fuel.typeGasoline   -0.348401   0.244444  -1.425  0.15421
```

```
## fuel.typeLPG          -0.330634   0.411809  -0.803  0.42213
## prior.claims          -0.004954   0.017662  -0.280  0.77915
## PCsize1               -0.007864   0.018792  -0.418  0.67565
## PCsize2                0.025890   0.037387   0.692  0.48870
## PCsize3               -0.026664   0.042096  -0.633  0.52652
## PCage1                 0.043519   0.026887   1.619  0.10567
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 1.809112)
##
##     Null deviance: 4822.9  on 2376  degrees of freedom
## Residual deviance: 4705.3  on 2312  degrees of freedom
## AIC: 36134
##
## Number of Fisher Scoring iterations: 13
```

6.3 Interaction identification

6.3.1 For claim count

```
summary(count.personal)
```

```
##
## Call:
## glm.nb(formula = claim.count ~ (marital.status + driver.gender +
##     driver.age)^2, data = train.data1, init.theta = 1.156316916,
##     link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.5657  -0.4197  -0.3966  -0.3724   3.8157
##
## Coefficients:
##                                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)                             -1.314758   0.687565  -1.912   0.0559 .
## marital.statusMarried                   -0.741162   0.671773  -1.103   0.2699
## marital.statusSingle                     0.028480   0.852853   0.033   0.9734
## marital.statusWidow                     -0.397176   1.105095  -0.359   0.7193
## driver.genderMale                        0.136550   0.381641   0.358   0.7205
## driver.age                              -0.025008   0.016086  -1.555   0.1200
## marital.statusMarried:driver.genderMale -0.016378   0.291092  -0.056   0.9551
## marital.statusSingle:driver.genderMale  -0.282776   0.386976  -0.731   0.4649
```

```
## marital.statusWidow:driver.genderMale    0.096259   0.426740   0.226   0.8215
## marital.statusMarried:driver.age         0.019176   0.015568   1.232   0.2181
## marital.statusSingle:driver.age          0.002259   0.020928   0.108   0.9141
## marital.statusWidow:driver.age           0.023774   0.021718   1.095   0.2737
## driver.genderMale:driver.age            -0.007187   0.006406  -1.122   0.2619
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(1.1563) family taken to be 1)
##
##     Null deviance: 11488  on 30466  degrees of freedom
## Residual deviance: 11413  on 30454  degrees of freedom
## AIC: 18023
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  1.156
##           Std. Err.:  0.184
##
##  2 x log-likelihood:  -17995.392
```

```
summary(count.vehicle)
```

```
##
## Call:
## glm.nb(formula = claim.count ~ (vehicle.age + vehicle.value +
##     no.seats + cubic.cent + horse.power + weight + fuel.type +
##     length + width + height)^2, data = train.data, init.theta = 1.29852926,
##     link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3481  -0.4311  -0.3913  -0.3495   3.8562
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     5.219e+00  1.183e+01   0.441   0.6592
## vehicle.age                    -5.556e-01  3.984e-01  -1.394   0.1632
## vehicle.value                  -3.298e-02  1.198e-01  -0.275   0.7831
## no.seats                        6.889e-01  7.320e-01   0.941   0.3466
## cubic.cent                     -3.553e-03  4.116e-03  -0.863   0.3881
## horse.power                    -7.803e-03  5.345e-02  -0.146   0.8839
## weight                         -1.191e-02  7.055e-03  -1.688   0.0914 .
## fuel.typeGasoline              -4.408e+00  5.385e+00  -0.818   0.4131
```

```
## fuel.typeLPG                      1.514e+01  8.861e+00   1.709   0.0874 .
## length                            8.379e+00  3.798e+00   2.206   0.0274 *
## width                            -7.315e+00  7.284e+00  -1.004   0.3153
## height                           -8.789e+00  6.390e+00  -1.375   0.1690
## vehicle.age:vehicle.value        -2.077e-03  1.550e-03  -1.340   0.1804
## vehicle.age:no.seats             -7.418e-03  1.175e-02  -0.631   0.5279
## vehicle.age:cubic.cent            7.778e-05  5.523e-05   1.408   0.1590
## vehicle.age:horse.power          -1.224e-03  1.069e-03  -1.146   0.2519
## vehicle.age:weight                1.572e-04  1.374e-04   1.144   0.2527
## vehicle.age:fuel.typeGasoline    -7.939e-02  1.271e-01  -0.624   0.5323
## vehicle.age:fuel.typeLPG         -1.363e-01  2.556e-01  -0.533   0.5938
## vehicle.age:length               -9.184e-03  7.673e-02  -0.120   0.9047
## vehicle.age:width                 2.833e-01  1.853e-01   1.529   0.1264
## vehicle.age:height               -1.297e-01  1.490e-01  -0.870   0.3842
## vehicle.value:no.seats            1.512e-03  4.032e-03   0.375   0.7077
## vehicle.value:cubic.cent          2.361e-05  2.043e-05   1.155   0.2480
## vehicle.value:horse.power        -7.821e-06  2.304e-04  -0.034   0.9729
## vehicle.value:weight              2.516e-05  4.062e-05   0.619   0.5357
## vehicle.value:fuel.typeGasoline  3.559e-02  5.502e-02   0.647   0.5177
## vehicle.value:fuel.typeLPG       -2.493e-02  9.781e-02  -0.255   0.7988
## vehicle.value:length             -7.082e-03  2.171e-02  -0.326   0.7443
## vehicle.value:width               5.644e-02  6.043e-02   0.934   0.3503
## vehicle.value:height             -8.291e-02  5.555e-02  -1.493   0.1355
## no.seats:cubic.cent               9.508e-05  8.373e-05   1.136   0.2561
## no.seats:horse.power              3.723e-03  1.980e-03   1.881   0.0600 .
## no.seats:weight                  -3.197e-04  2.474e-04  -1.293   0.1962
## no.seats:fuel.typeGasoline       -3.037e-01  3.044e-01  -0.998   0.3184
## no.seats:fuel.typeLPG             6.099e-01  4.038e-01   1.510   0.1310
## no.seats:length                   1.535e-01  1.316e-01   1.166   0.2435
## no.seats:width                   -4.017e-01  3.568e-01  -1.126   0.2603
## no.seats:height                  -3.883e-01  2.860e-01  -1.358   0.1745
## cubic.cent:horse.power            1.651e-05  1.067e-05   1.547   0.1218
## cubic.cent:weight                -2.739e-06  1.196e-06  -2.290   0.0220 *
## cubic.cent:fuel.typeGasoline     -7.911e-04  2.635e-03  -0.300   0.7640
## cubic.cent:fuel.typeLPG          -3.998e-04  3.267e-03  -0.122   0.9026
## cubic.cent:length                -3.522e-04  5.684e-04  -0.620   0.5355
## cubic.cent:width                  3.379e-04  1.785e-03   0.189   0.8499
## cubic.cent:height                 3.099e-03  1.834e-03   1.690   0.0911 .
## horse.power:weight               -3.968e-06  2.038e-05  -0.195   0.8456
## horse.power:fuel.typeGasoline     3.281e-02  3.373e-02   0.973   0.3308
## horse.power:fuel.typeLPG          7.341e-03  4.968e-02   0.148   0.8825
## horse.power:length               -7.750e-03  1.276e-02  -0.608   0.5435
## horse.power:width                -4.212e-02  2.981e-02  -1.413   0.1576
## horse.power:height                4.696e-02  2.732e-02   1.719   0.0856 .
## weight:fuel.typeGasoline         -2.772e-03  3.342e-03  -0.830   0.4068
```

```
## weight:fuel.typeLPG              7.199e-04  6.747e-03   0.107   0.9150
## weight:length                    1.979e-03  1.014e-03   1.953   0.0508 .
## weight:width                     5.625e-03  3.561e-03   1.579   0.1142
## weight:height                   -1.080e-03  2.841e-03  -0.380   0.7039
## fuel.typeGasoline:length         1.183e+00  1.106e+00   1.069   0.2851
## fuel.typeLPG:length             -5.692e+00  3.732e+00  -1.525   0.1272
## fuel.typeGasoline:width          8.603e-01  2.837e+00   0.303   0.7617
## fuel.typeLPG:width               6.212e+00  5.262e+00   1.180   0.2378
## fuel.typeGasoline:height         6.968e-02  2.189e+00   0.032   0.9746
## fuel.typeLPG:height             -2.990e+00  5.400e+00  -0.554   0.5798
## length:width                    -2.875e+00  1.786e+00  -1.610   0.1074
## length:height                   -2.587e+00  1.562e+00  -1.656   0.0977 .
## width:height                     8.515e+00  3.634e+00   2.343   0.0191 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(1.2985) family taken to be 1)
##
##     Null deviance: 11642  on 30466  degrees of freedom
## Residual deviance: 11420  on 30401  degrees of freedom
## AIC: 17984
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  1.299
##           Std. Err.:  0.220
##
##  2 x log-likelihood:  -17850.160
```

6.3.2 For severity

```
summary(sev.personal)
```

```
##
## Call:
## glm(formula = sev ~ (marital.status + driver.gender + driver.age)^2,
##     family = Gamma(link = "log"), data = train.sev1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8098  -1.6094  -0.6805   0.2638   4.2316
##
## Coefficients:
```

```
##                                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                             6.5142072  0.9719835   6.702 2.56e-11
## marital.statusMarried                   0.3813352  0.9474327   0.402   0.6874
## marital.statusSingle                    0.5238101  1.1741077   0.446   0.6555
## marital.statusWidow                    -0.1548793  1.4520856  -0.107   0.9151
## driver.genderMale                       0.5676161  0.5248228   1.082   0.2796
## driver.age                             -0.0055374  0.0222213  -0.249   0.8032
## marital.statusMarried:driver.genderMale -0.6840450  0.4003950  -1.708   0.0877
## marital.statusSingle:driver.genderMale  -0.6738388  0.5281719  -1.276   0.2022
## marital.statusWidow:driver.genderMale   -0.9973884  0.5836471  -1.709   0.0876
## marital.statusMarried:driver.age         0.0033346  0.0214897   0.155   0.8767
## marital.statusSingle:driver.age          0.0030837  0.0281805   0.109   0.9129
## marital.statusWidow:driver.age           0.0092627  0.0287794   0.322   0.7476
## driver.genderMale:driver.age            -0.0007195  0.0086689  -0.083   0.9339
##
## (Intercept)                             ***
## marital.statusMarried
## marital.statusSingle
## marital.statusWidow
## driver.genderMale
## driver.age
## marital.statusMarried:driver.genderMale .
## marital.statusSingle:driver.genderMale
## marital.statusWidow:driver.genderMale   .
## marital.statusMarried:driver.age
## marital.statusSingle:driver.age
## marital.statusWidow:driver.age
## driver.genderMale:driver.age
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 1.845557)
##
##     Null deviance: 4822.9  on 2376  degrees of freedom
## Residual deviance: 4801.5  on 2364  degrees of freedom
## AIC: 36092
##
## Number of Fisher Scoring iterations: 13
```

```
summary(sev.vehicle )
```

```
##
## Call:
## glm(formula = sev ~ (PCage1 + PCsize1 + PCsize2 + PCsize3 + fuel.type)^2,
##     family = Gamma(link = "log"), data = train.sev1)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8366  -1.6078  -0.6634   0.2920   4.4602
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  6.690508   0.031981 209.204   <2e-16 ***
## PCage1                       0.021543   0.027861   0.773   0.4395
## PCsize1                     -0.003552   0.016928  -0.210   0.8338
## PCsize2                      0.035812   0.030604   1.170   0.2420
## PCsize3                      0.004716   0.038255   0.123   0.9019
## fuel.typeGasoline           -0.581408   0.362320  -1.605   0.1087
## fuel.typeLPG                -1.133332   0.933895  -1.214   0.2250
## PCage1:PCsize1               0.003595   0.011096   0.324   0.7460
## PCage1:PCsize2              -0.022398   0.023849  -0.939   0.3477
## PCage1:PCsize3              -0.014315   0.023892  -0.599   0.5491
## PCage1:fuel.typeGasoline     0.537206   0.274186   1.959   0.0502 .
## PCage1:fuel.typeLPG         -0.429772   0.501493  -0.857   0.3915
## PCsize1:PCsize2              0.004147   0.019391   0.214   0.8307
## PCsize1:PCsize3              0.018448   0.011832   1.559   0.1191
## PCsize1:fuel.typeGasoline -0.130310   0.257882  -0.505   0.6134
## PCsize1:fuel.typeLPG         0.213305   0.487126   0.438   0.6615
## PCsize2:PCsize3              0.051303   0.036808   1.394   0.1635
## PCsize2:fuel.typeGasoline -0.290145   0.303473  -0.956   0.3391
## PCsize2:fuel.typeLPG        -0.193774   0.581158  -0.333   0.7388
## PCsize3:fuel.typeGasoline -0.036047   0.293825  -0.123   0.9024
## PCsize3:fuel.typeLPG         0.283859   0.525834   0.540   0.5894
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 1.879858)
##
##     Null deviance: 4822.9  on 2376  degrees of freedom
## Residual deviance: 4795.9  on 2356  degrees of freedom
## AIC: 36104
##
## Number of Fisher Scoring iterations: 9
```

6.4 collapsing categorical data

6.4.1 For Frequency

Grouped region for frequency

```
region.analysis<-train.data1%>%
       group_by(region)%>%
       summarize(mean.freq=mean(claim.count),
                  sd.freq=sd(claim.count),
                 count=n())
```

Table 8: grouped region for frequecy

region	mean.freq	sd.freq	count	region	mean.freq	sd.freq	count
1	0.0876649	0.3040792	1289	3	0.0785403	0.2862241	2521
32	0.0858469	0.2963543	1293	4	0.0677308	0.2630834	2318
region	mean.freq	sd.freq	count	region	mean.freq	sd.freq	count
16	0.0762943	0.2759175	367	11	0.0666667	0.2499344	255
31	0.0758929	0.2818089	224	30	0.0844595	0.3338959	296
34	0.0650155	0.2821538	323				
region	mean.freq	sd.freq	count	region	mean.freq	sd.freq	count
21	0.0849858	0.2892507	353	5	0.0729367	0.2949198	521
33	0.0905350	0.3149723	243	13	0.0722311	0.2712055	623
38	0.0896226	0.3024166	212				
region	mean.freq	sd.freq	count	region	mean.freq	sd.freq	count
12	0.0730088	0.2769437	452	6	0.1005291	0.3264628	378
14	0.0791476	0.2970493	657	36	0.0916808	0.3168973	589
19	0.0749354	0.2636279	387				
23	0.0649351	0.2670080	385				
37	0.0779817	0.2768817	436				

Analyzing body codes

```
bodycode.analysis<-train.data1%>%
  group_by(body.code)%>%
  summarize(mean.freq=mean(claim.count),
            sd.freq=sd(claim.count),
            count=n())
```

Table 9: analyzing grouping for Body code

body.code	mean.freq	sd.freq	count
A	0.0846189	0.3025191	13094
B	0.0338983	0.1811381	531
C	0.0960560	0.3255466	1572
D	0.0852311	0.3003843	7509
E	0.0802370	0.2904870	6244
F	0.0769231	0.2792053	299
G	0.0994208	0.3471898	1036
H	0.1538462	0.4184443	182

Marital status

```
marital.analysis<-train.data1%>%
  group_by(marital.status)%>%
  summarize(mean.freq=mean(claim.count),
            sd.freq=sd(claim.count),
            count=n())
```

Table 10: analyzing grouping for Marital status

marital.status	mean.freq	sd.freq	count
Divorced	0.0886850	0.3002201	654
Married	0.0827304	0.2983601	28333
Single	0.1018767	0.3281461	1119
Widow	0.1551247	0.4259402	361

ncd levels

```
ncd.analysis<-train.data1%>%
  group_by(ncd.level)%>%
  summarize(mean.freq=mean(claim.count),
            sd.freq=sd(claim.count),
            count=n())
```

Table 11: analyzing grouping for ncd level

ncd.level	mean.freq	sd.freq	count
1	0.1203824	0.3608904	9204
2	0.0775510	0.3035343	490
3	0.0944373	0.3192489	6957
4	0.0683215	0.2677844	4230
5	0.0614982	0.2509396	3057
6	0.0447235	0.2125579	6529

6.4.2 For severity

Grouped region for severity

```
region.analysis<-train.data1%>%
       group_by(region)%>%
       summarize(mean.freq=mean(claim.count),
                 count=n())
```

Table 12: grouped region for severity

region	mean.freq	count
12	0.0730088	452
21	0.0849858	353
30	0.0844595	296
34	0.0650155	323

region	mean.freq	count
16	0.0762943	367
20	0.0484848	330
27	0.0496324	544
37	0.0779817	436
38	0.0896226	212

region	mean.freq	count
5	0.0729367	521
19	0.0749354	387

region	mean.freq	count
22	0.0520000	250
28	0.0523077	325

region	mean.freq	count
1	0.0876649	1289
25	0.0973896	996

region	mean.freq	count
15	0.0715048	1874
18	0.1097756	1248
32	0.0858469	1293

```
ncd.analysis.sev<-train.data1%>%
  filter(sev>0)%>%
  group_by(ncd.level)%>%
  summarize(mean.sev=mean(sev),
            median.sev=median(sev),
            count=n())
```

Table 13: grouped ncd levels

ncd.level	mean.sev	median.sev	count
4	725.1915	318.61	273
6	674.8106	346.57	285

```
bodycode.analysis.sev<-train.data1%>%
  filter(sev>0)%>%
  group_by(body.code)%>%
  summarize(mean.sev=mean(sev),
            median.sev=median(sev),
            count=n())
```

Table 14: grouped body code levels

body.code	mean.sev	median.sev	count
C	810.1271	350.335	136
G	826.1736	321.675	88

```
marital.analysis.sev<-train.data1%>%
  filter(sev>0)%>%
  group_by(marital.status)%>%
  summarize(mean.sev=mean(sev),
```

```
            median.sev=median(sev),
            count=n())
```

Table 15: analysiing grouping for marital levels

marital.status	mean.sev	median.sev	count
Divorced	795.0845	433.02	55
Married	788.5722	361.25	2167
Single	955.0049	524.67	107
Widow	603.2310	355.35	48