General Insurance Case Study 1

Edward W. (Jed) Frees *

May 30, 2018

Contents

Case Summary. In this case study, we show the details of modeling the frequency and severity of fire insurance experience, summarized in Table 1. The data are from the Data Center Management Board of the National Insurance (BPPDAN), see the website by PT. REASURANSI INTERNASIONAL INDONESIA at http://www.reindo.co.id/bppdan/default.asp.

After you have read the case, your assignment will be to replicate this analysis using different data.

- You may use fire insurance experience but for a different year.
- You may analyze another line of business, BPPDAN also provides data by industrial all risk, lost of profit, and earthquake..
- You may use all lines of business but focus a specific province, BPPDAN subdivides Indonesian experience into 32 provinces.
- Alternatively, you may consider a group by occupation code such as mining, drilling, and so forth, BPPDAN provides data by 14 occupation codes.

*Keywords: Indonesian Fire Insurance

1 Data

The data are summarized in Table 1.

Class of Business	: **FIRE**		Data Position from	:	01/01/2009
Underwriting Year	: 2009		to	:	15/06/2011
Occupation Code	: All Code		Processing Date	:	15/10/2011

Class	From	**Sum Insured** (In Million Rp) To	**Number of Policies**	**Original Premium** (In Million Rp)	**Claim Frequency**	**Claim Severity** (In Million Rp)	**Loss Ratio**
1	0	50	161,015	64,850.58	668	38,243.84	58.97
2	50	100	147,879	31,499.72	280	8,297.62	26.34
3	100	200	192,417	73,241.06	461	15,813.84	21.59
4	200	300	120,484	71,680.01	314	14,515.05	20.25
5	300	500	131,621	121,468.45	381	21,907.68	18.04
6	500	750	65,260	90,817.97	210	13,030.42	14.35
7	750	1,000	44,413	87,321.01	215	21,361.32	24.46
8	1,000	1,500	32,665	85,188.86	206	16,853.43	19.78
9	1,500	2,000	16,922	58,965.73	83	13,492.44	22.88
10	2,000	2,500	8,860	39,307.82	78	10,901.01	27.73
11	2,500	3,000	6,673	35,803.31	77	9,007.83	25.16
12	3,000	4,000	7,495	43,755.51	88	5,577.63	12.75
13	4,000	5,000	4,660	34,239.52	77	13,154.63	38.42
14	5,000	7,500	5,819	50,291.28	145	12,752.13	25.36
15	7,500	10,000	3,100	33,666.94	85	5,118.05	15.20
16	10,000	20,000	4,821	67,854.71	354	43,203.70	63.67
17	20,000	50,000	3,989	54,993.99	377	93,502.17	170.02
18	50,000	100,000	1,867	25,606.71	145	27,008.69	105.48
19	100,000	500,000	2,349	26,276.96	280	156,418.90	595.27
20	500,000	above	1,089	73,162.42	222	4,093.45	5.60

Source: PT. Reasuransi Internasional Indonesia
Website : http://www.reindo.co.id/bppdan/default.asp

Table 1: Indonesian Risk and Loss Profile 2009

For this case study, we used the statistical package "R" for the analysis. You may replicate the analysis using this package and the command syntax given in the following. (Of course, there are several other languages that will do similar analyses.) For an introduction to "R" in the context regression modeling (which will be used for much of the following analysis), one source is the web site for the book *Regression Modeling with Actuarial and Financial Applications*, Frees (2010), at http://research.bus.wisc.edu/RegActuaries.

Here are the "R" Commands used to import the data and create important variables.

```
#  "R" Commands to Import Data
Fire2009  <- read.csv("FireRisk2009.csv", header=TRUE)#, sep="\t")
#View(Fire2009)
Fire2009$LossRatio <- 100*Fire2009$Claim/Fire2009$Premium
Fire2009$NumClmPol <- Fire2009$NumClaim/Fire2009$NumPol
#summary(Fire2009)
#attach(Fire2009)
```

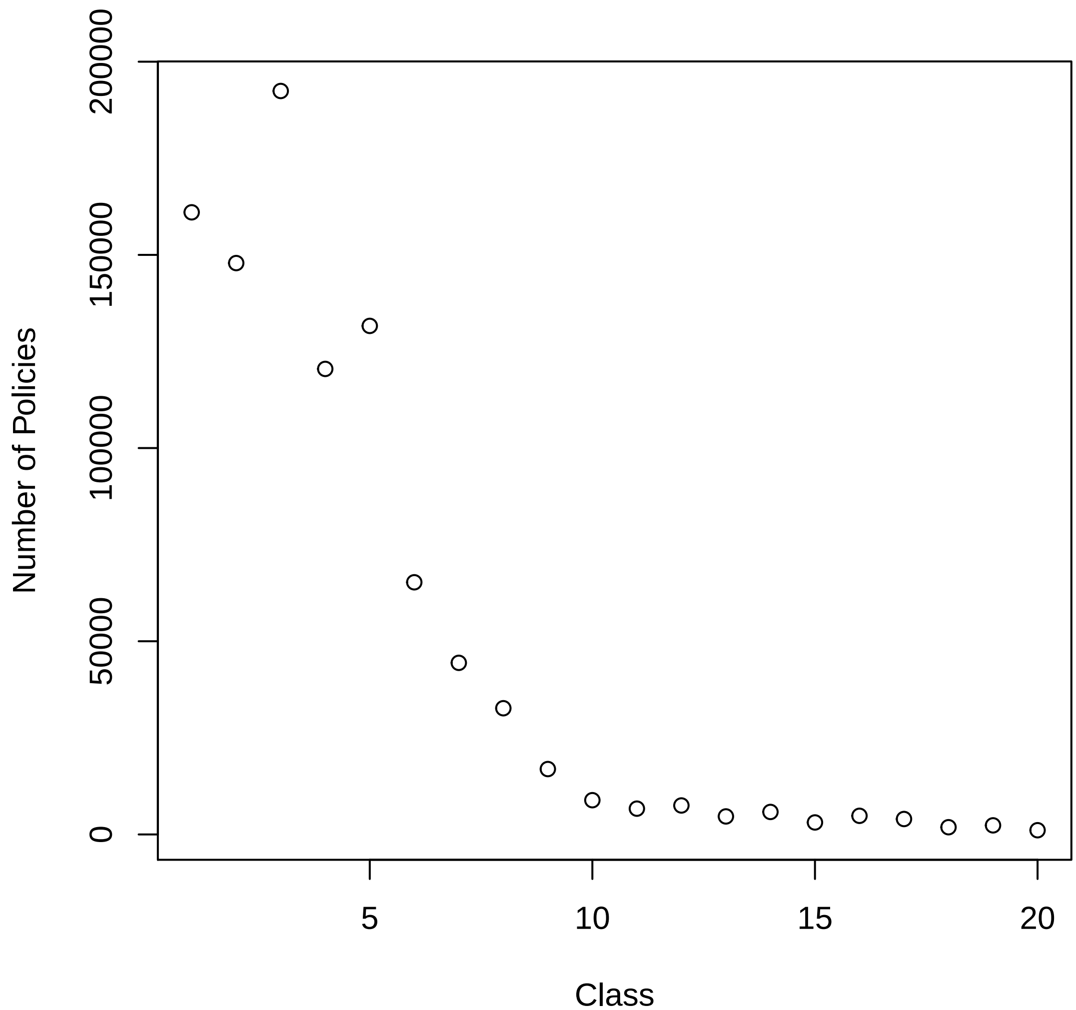
2 Frequency Modeling

2.1 Graphical Approaches

To understand patterns in the frequency of claims, we first examine several graphs.

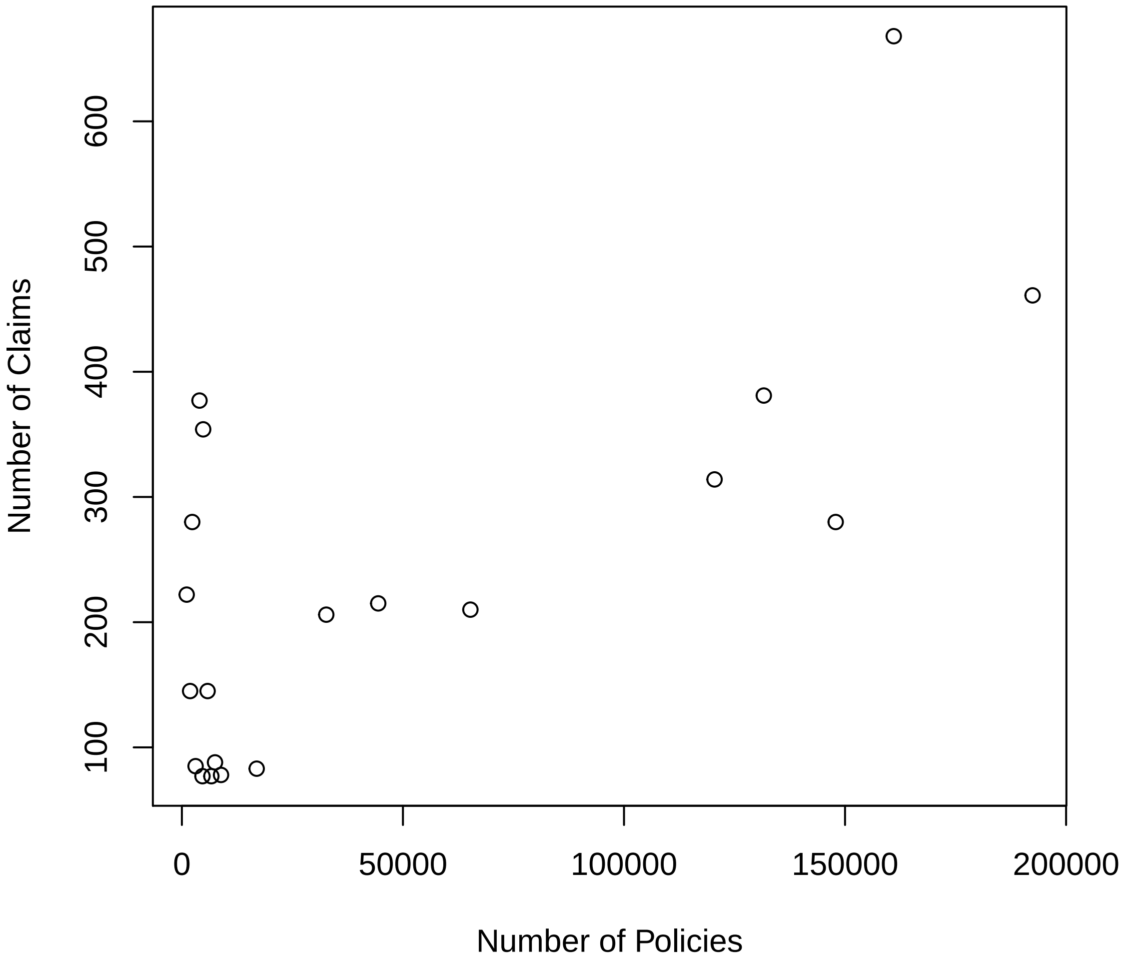
We first plot the number of policies by class. Not surprisingly, we see that the number of policies decreases as class (a measure of size of the policy) increases.

```
plot(Fire2009$Class,Fire2009$NumPol, xlab="Class", ylab="Number of Policies")  #  FEWER POLICIES WITH L
```



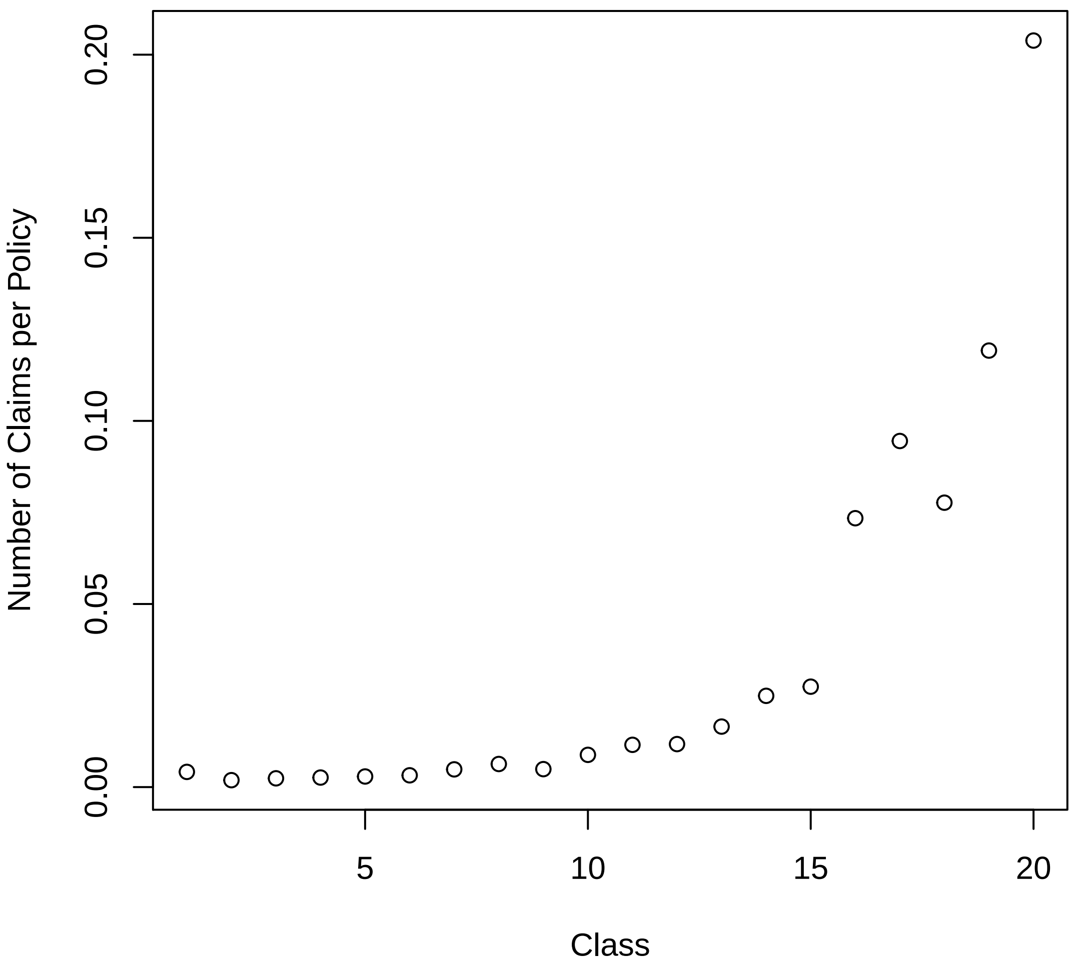
We next examine the number of claims and the number of policies by class size. The figure shows that classes with a large number of policies tend to have a large number of claims and similarly for classes with small numbers. However, the pattern does not appear to be linear.

```
plot(Fire2009$NumPol,Fire2009$NumClaim, xlab="Number of Policies", ylab="Number of Claims") #  MORE CLA
```



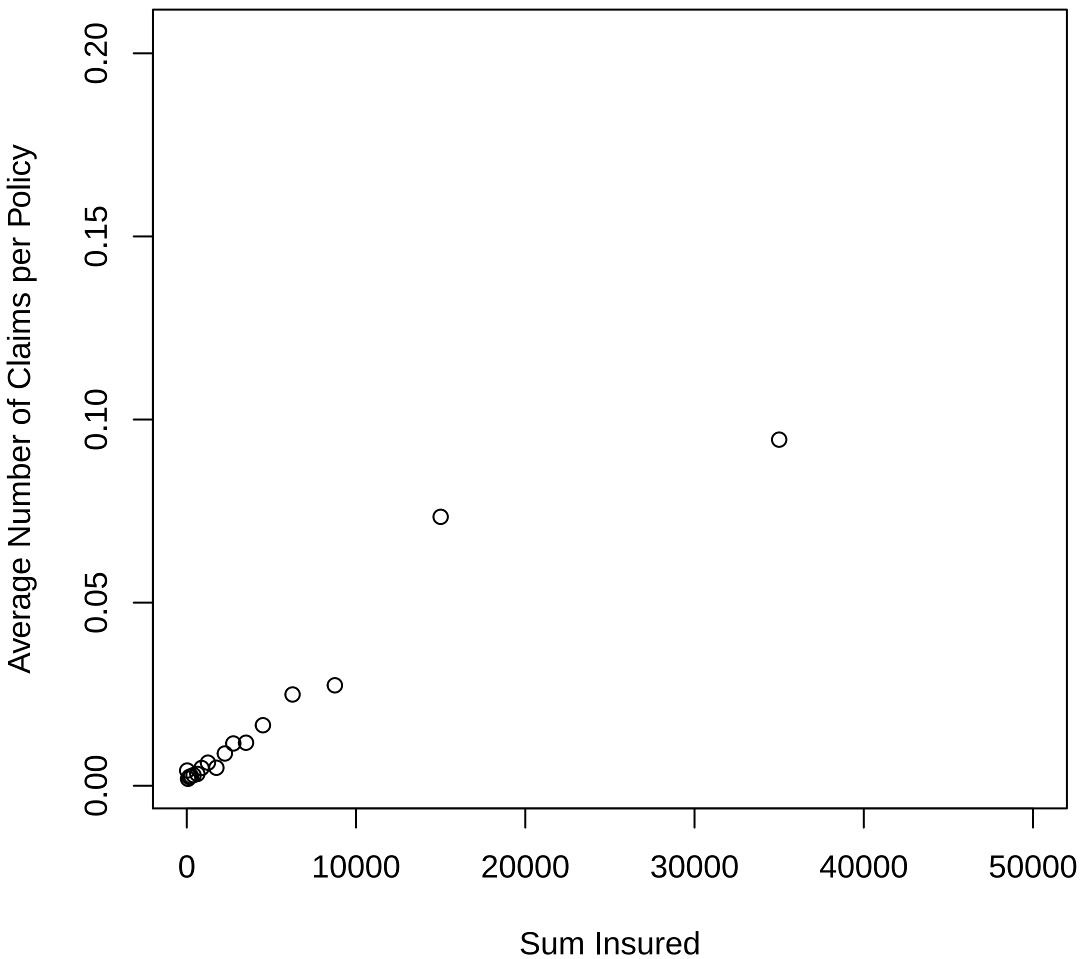
The number of policies is commonly used as an exposure measure for claim frequency. Thus, we rescale claim frequency and examine the average number of claims per policy.
The figure shows number of claims per policy versus class, which is a measure of the size of insurance. Interestingly, the number of claims per policy increases as the class (sum insured) increases.

```
Fire2009$NumClmPol <- Fire2009$NumClaim/Fire2009$NumPol
plot(Fire2009$Class,Fire2009$NumClmPol, xlab="Class",
                    ylab="Number of Claims per Policy") #  MORE CLAIMS PER POLICIES AS SUM INSURED INCR
```



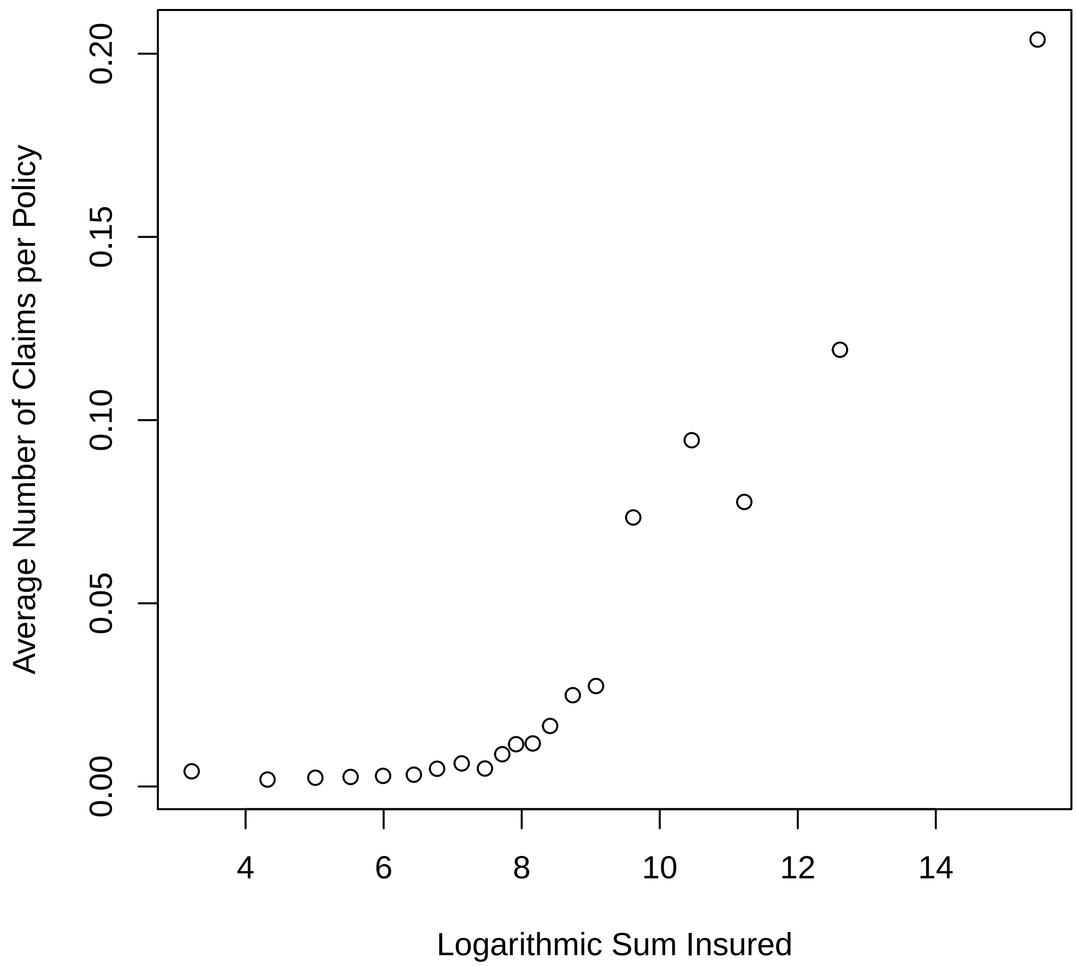
Here is another way to see that the the frequency, as measured by the number of claims per policy, grows with the sum insured. Here, sum insured is defined to be the average of the upper and lower endpoints of the interval defining the class, or band. This figure shows a linear relationship between average number of claims per policy and sum insured for small values of the sum insured. Although not displayed, this linear pattern does not hold for larger values of sum insured.

```
Fire2009$SumIns <- (Fire2009$SumFrom+Fire2009$SumTo)/2
plot(Fire2009$SumIns,Fire2009$NumClmPol,xlab="Sum Insured",
      ylab="Average Number of Claims per Policy",xlim=c(0,50000)) #  MORE CLAIMS PER POLICIES AS SUM IN
```



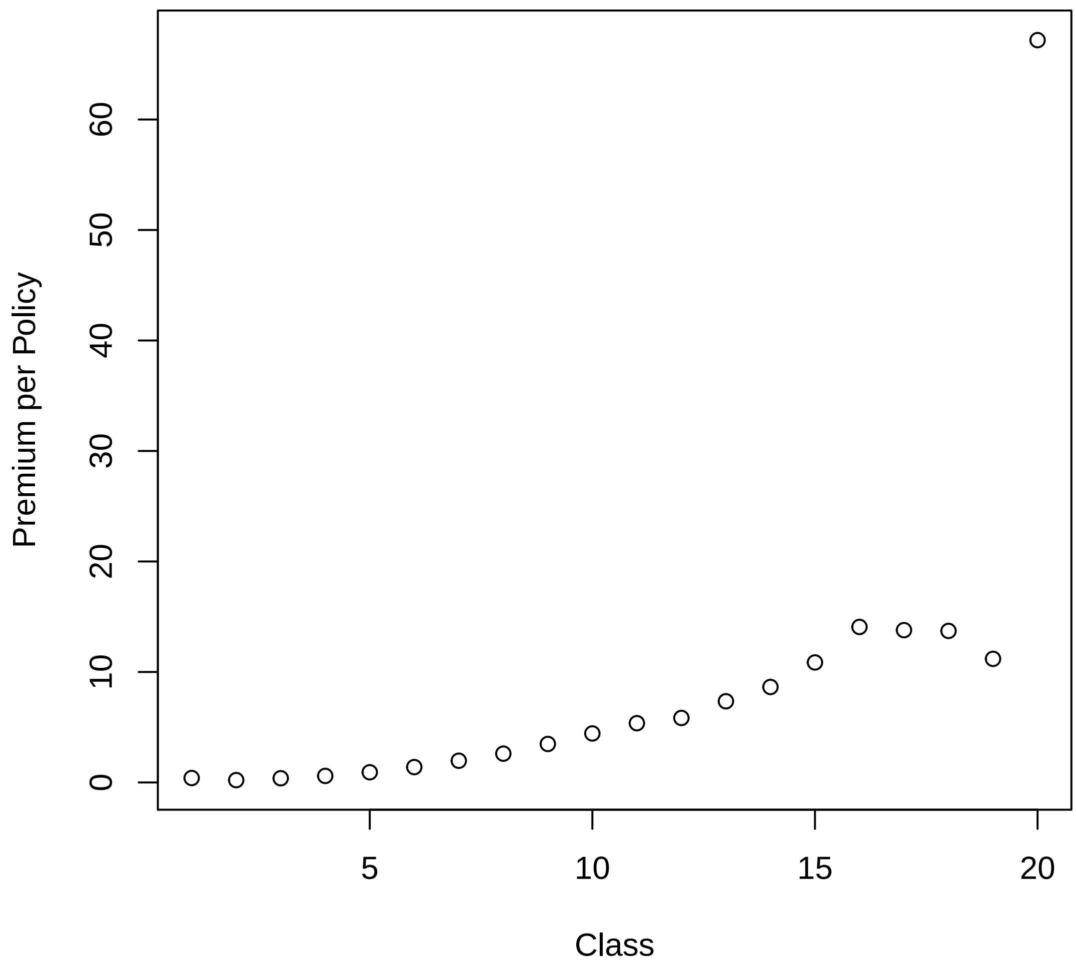
Here is another way to see the relationship between the average number claims per policy and the sum insured. This figure plots the logarithmic sum insured versus claims frequency. This figure also shows that the claims frequency increases with sum insured.

```
plot(log(Fire2009$SumIns),Fire2009$NumClmPol,xlab="Logarithmic Sum Insured",
                                             ylab="Average Number of Claims per Policy") #  MORE CLAIMS
```



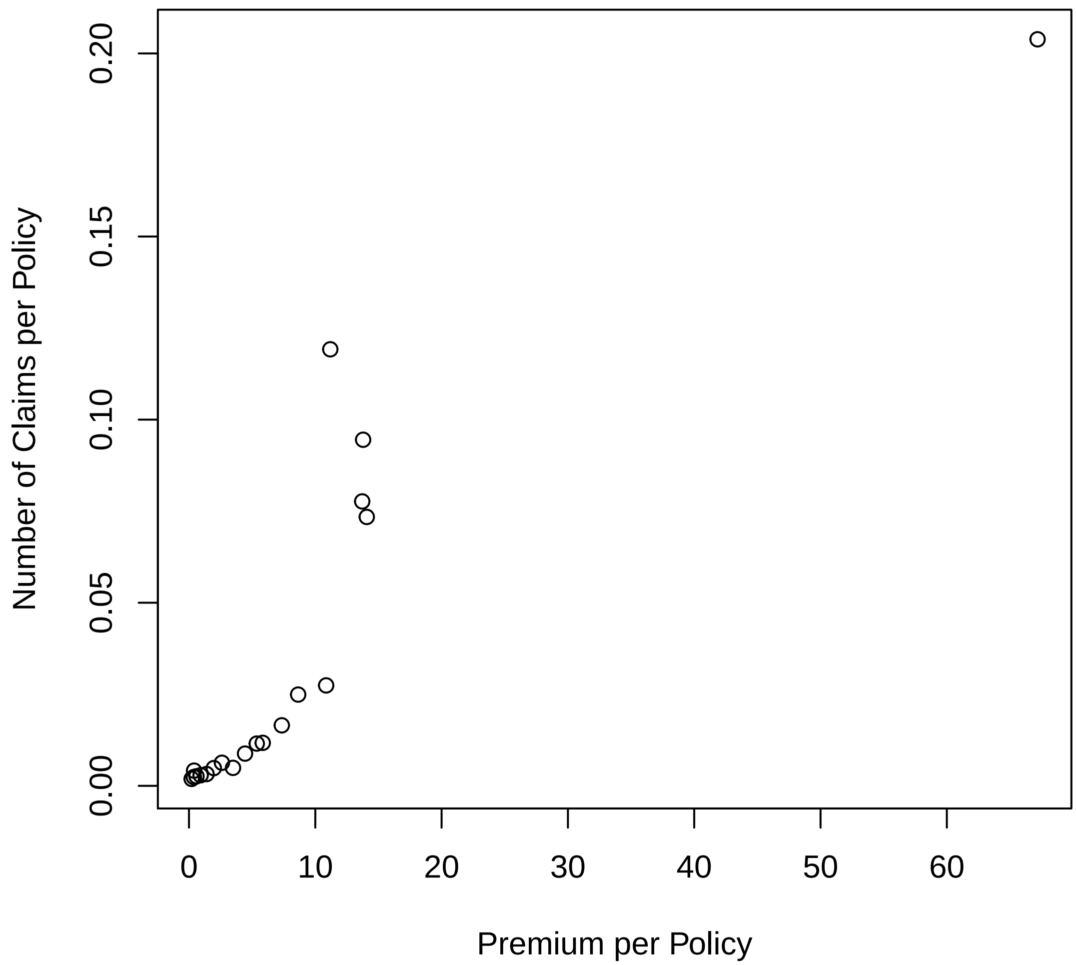
Another exposure measure is premium. Here is a plot that shows that the premium (per policy) is related to policy size as measured by sum insured or class.

```
Fire2009$PremPol <- Fire2009$Premium/Fire2009$NumPol
plot(Fire2009$Class,Fire2009$PremPol, ylab="Premium per Policy",
                                      xlab="Class") # PREMIUM PER POLICY ALSO INCREASES AS SUM INSURED
```



Here is a plot of premium per policy versus number of claims per policy. This suggests that premium might also be a useful exposure measure. From the figure, we see that premium is positively relatied to the number of claims.

```
Fire2009$NumClmPol <- Fire2009$NumClaim/Fire2009$NumPol
plot(Fire2009$PremPol,Fire2009$NumClmPol, xlab="Premium per Policy", ylab="Number of Claims per Policy")
```



2.2 Fitting Claims Number Models

The graphical analysis section suggests a number of variables that may influence the number of claims per policy. In this section, we fit several frequency models that are suggested by the graphical analysis. We use regression and generalized linear model techniques for this fitting. For an introduction or review of these techniques, one source is Frees (2010), *Regression Modeling with Actuarial and Financial Applications*, Cambridge University Press.

2.2.1 Model without Explanatory Variables

As a benchmark, we fit models that do not use any information from potential explanatory variables. To begin, we calculate the average number of claims per policy to be:

```
#  CLAIMS NUMBER MODELS
#  MODEL 1 - IGNORE SUM INSURED, FIT NUMBER OF CLAIMS USING ONLY NUMBER OF POLICIES.
(ModFreq.1.Estimate <- sum(Fire2009$NumClaim)/sum(Fire2009$NumPol))
```

```
## [1] 0.004926313
```

As is well-known, this is the maximum likelihood estimate of a Poisson model. Here is the "R" code that verifies this:

```
ModFreq.1 <- glm(NumClaim ~ 1, offset=log(NumPol),poisson(link=log), data=Fire2009)
summary(ModFreq.1)
```

```
##
## Call:
## glm(formula = NumClaim ~ 1, family = poisson(link = log), data = Fire2009,
##     offset = log(NumPol))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -19.014   -5.089    5.610   17.302   38.893
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.31316    0.01452    -366   <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: 7380  on 19  degrees of freedom
## Residual deviance: 7380  on 19  degrees of freedom
## AIC: 7524
##
## Number of Fisher Scoring iterations: 6
```

```
exp(ModFreq.1$coefficients) # SAME AS THE MEAN
```

```
## (Intercept)
## 0.004926313
```

From this, we can estimate the number of claims as the overall average times the number of policies in each class. One way to assess the fit of a model that is easy to understand and explain is through a chi-square goodness of fit statistic. Here is the calculation of this statistic.

```
ModFreq.1.Summary <- cbind (Fire2009$NumClaim,
   ModFreq.1.Estimate*Fire2009$NumPol)
   ModFreq.1.Summary  #  THIS IS A POOR FITTING MODEL
```

```
##       [,1]       [,2]
##  [1,]  668 793.210272
##  [2,]  280 728.498226
##  [3,]  461 947.906350
##  [4,]  314 593.541884
##  [5,]  381 648.406231
##  [6,]  210 321.491180
##  [7,]  215 218.792335
##  [8,]  206 160.918011
##  [9,]   83  83.363067
## [10,]   78  43.647132
## [11,]   77  32.873286
## [12,]   88  36.922715
## [13,]   77  22.956618
## [14,]  145  28.666215
## [15,]   85  15.271570
## [16,]  354  23.749755
## [17,]  377  19.651062
## [18,]  145   9.197426
## [19,]  280  11.571909
## [20,]  222   5.364755
```

```
(SM.ModFreq.1 <- sum((Fire2009$NumClaim - ModFreq.1.Estimate*Fire2009$NumPol)^2/(ModFreq.1.Estimate*Fire
```

```
## [1] 29984.23
```

Here is a function to make the calculation of these statistics more routine:

```
#  MAKE THESE STATISTICS ROUTINE TO SAVE WORK
ModelSummary1 <- function(ModEstimate){
   ModFreq.Summary <- cbind (Fire2009$NumClaim,ModEstimate)
   ModFreq.Summary }
   ModelSummary2 <- function(ModEstimate){sum((Fire2009$NumClaim - ModEstimate)^2/(ModEstimate))}

ModelSummary1( ModFreq.1.Estimate*Fire2009$NumPol);ModelSummary2( ModFreq.1.Estimate*Fire2009$NumPol)
```

```
##           ModEstimate
##  [1,] 668  793.210272
##  [2,] 280  728.498226
##  [3,] 461  947.906350
##  [4,] 314  593.541884
##  [5,] 381  648.406231
##  [6,] 210  321.491180
##  [7,] 215  218.792335
##  [8,] 206  160.918011
##  [9,]  83   83.363067
## [10,]  78   43.647132
## [11,]  77   32.873286
## [12,]  88   36.922715
## [13,]  77   22.956618
## [14,] 145   28.666215
## [15,]  85   15.271570
```

```
## [16,] 354   23.749755
## [17,] 377   19.651062
## [18,] 145    9.197426
## [19,] 280   11.571909
## [20,] 222    5.364755

## [1] 29984.23
```

2.2.2 Model with Number of Policies

We next consider a Poisson model where the number of policies is an explanatory variable in a Poisson regression. This is a slight extension of prior work in the sense that, in the previous model, we used logarithm number of policies as an offset. Recall, in GLM terminology, that an offset is simply an explanatory variable where the coefficient is pre-specified to be 1, regardless of the data.

The goodness of fit statistic shows that the extra flexibility of allowing number of policies to be an explanatory variable improves the fit.

```r
#  MODEL 1A - FIT NUMBER OF CLAIMS USING NUMBER OF POLICIES AS AN EXPLANATORY VARIABLE IN A POISSON REG
ModFreq.1A <- glm(NumClaim ~ log(NumPol),poisson(link=log), data=Fire2009)
summary(ModFreq.1A)
```

```
##
## Call:
## glm(formula = NumClaim ~ log(NumPol), family = poisson(link = log),
##     data = Fire2009)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -11.098   -6.982   -3.267    4.853   14.099
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 3.393299   0.091487   37.09   <2e-16 ***
## log(NumPol) 0.209441   0.008857   23.65   <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: 1838.1  on 19  degrees of freedom
## Residual deviance: 1268.7  on 18  degrees of freedom
## AIC: 1414.6
##
## Number of Fisher Scoring iterations: 5
```

```r
ModelSummary1(ModFreq.1A$fitted.values);ModelSummary2(ModFreq.1A$fitted.values) #  NOT GREAT BUT BETTER
```

```
##        ModEstimate
## 1  668    366.6272
## 2  280    360.1502
## 3  461    380.5665
## 4  314    345.0231
## 5  381    351.4713
## 6  210    303.4434
## 7  215    279.9447
```

```
## 8  206    262.4986
## 9   83    228.7199
## 10  78    199.7318
## 11  77    188.2185
## 12  88    192.8541
## 13  77    174.5834
## 14 145    182.8969
## 15  85    160.2975
## 16 354    175.8298
## 17 377    168.9901
## 18 145    144.1465
## 19 280    151.2494
## 20 222    128.7569
```

```
## [1] 1345.079
```

2.2.3 Model with Additonal Explanatory Variables

Adding logarithmic sum insured, a measure of the policy size, helps improve our fits.

```
#  MODEL 2 - INCLUDE CLASS AND log(NumPol) AS EXPLANATORY VARIABLEs IN A POISSON REGRESSION
ModFreq.2 <- glm(NumClaim ~ log(SumIns)+log(NumPol),poisson(link=log), data=Fire2009)
ModelSummary2(ModFreq.2$fitted.values) #  NOT GREAT BUT BETTER THAN MODEL 1
```

```
## [1] 1282.855
```

Replace logarithmic sum insured with Class, another measure of the policy size, helps improve our fits.

```
#  MODEL 2 - INCLUDE CLASS AND log(NumPol) AS EXPLANATORY VARIABLEs IN A POISSON REGRESSION
ModFreq.2A <- glm(NumClaim ~ Class+log(NumPol),poisson(link=log), data=Fire2009)
summary(ModFreq.2A)
```

```
##
## Call:
## glm(formula = NumClaim ~ Class + log(NumPol), family = poisson(link = log),
##     data = Fire2009)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -7.4277  -4.0410  -3.1887   0.4856  21.9564
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.37242    0.51845  -12.29   <2e-16 ***
## Class        0.22658    0.01173   19.32   <2e-16 ***
## log(NumPol)  0.97265    0.04074   23.88   <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: 1838.13  on 19  degrees of freedom
## Residual deviance:  884.57  on 17  degrees of freedom
## AIC: 1032.5
##
## Number of Fisher Scoring iterations: 5
```

```r
ModelSummary2(ModFreq.2A$fitted.values) #  NOT GREAT BUT BETTER THAN MODEL 1
```

```
## [1] 1097.349
```

For this model, the coefficient associated with logarithmic number of policies is nearly one. Thus, we call it one and go back to using logarithmic number of policies as an offset. The following model is a our preferred fitted model. This model essentially treats claims per policy as the dependent variable and "class" as an explanatory variable.

```r
#  MODEL 2B - INCLUDE CLASS AS AN EXPLANATORY VARIABLE, NUMPOL AS AN OFFSET, IN A POISSON REGRESSION
ModFreq.2B <- glm(NumClaim ~ Class, offset=log(NumPol),poisson(link=log), data=Fire2009)
summary(ModFreq.2B)
```

```
## 
## Call:
## glm(formula = NumClaim ~ Class, family = poisson(link = log), 
##     data = Fire2009, offset = log(NumPol))
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -7.2857  -3.9006  -3.1038   0.3561  22.1868  
## 
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -6.720151   0.026406  -254.5   <2e-16 ***
## Class        0.234280   0.002471    94.8   <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: 7380.00  on 19  degrees of freedom
## Residual deviance:  885.02  on 18  degrees of freedom
## AIC: 1031
## 
## Number of Fisher Scoring iterations: 4
```

```r
ModelSummary1(ModFreq.2B$fitted.values);ModelSummary2(ModFreq.2B$fitted.values) #  THIS IS THE BEST
```

```
##        ModEstimate
## 1  668    245.5209
## 2  280    285.0199
## 3  461    468.7688
## 4  314    371.0148
## 5  381    512.3110
## 6  210    321.0718
## 7  215    276.1924
## 8  206    256.7621
## 9   83    168.1305
## 10  78    111.2693
## 11  77    105.9276
## 12  88    150.3856
## 13  77    118.1863
## 14 145    186.5418
## 15  85    125.6135
## 16 354    246.9211
```

```
## 17 377    258.2449
## 18 145    152.7772
## 19 280    242.9651
## 20 222    142.3755
```

```
## [1] 1106.816
```

```
#  AS EITHER THE SUM INSURED OR THE NUMBER OF POLICIES INCREASE, THE EXPECTED NUMBER OF CLAIMS INCREASE
```

We tried a few other models. They were not bad but also did not provide a significant improvement.

```
#  A FEW OTHER MODELS TRIED BUT NOT ADOPTED
ModFreq.3 <- glm(NumClaim ~ log(SumIns),offset=log(NumPol),poisson(link=log), data=Fire2009)
ModelSummary2(ModFreq.3$fitted.values)
```

```
## [1] 2255.955
```

```
ModFreq.4 <- glm(NumClaim ~ log(SumIns)+log(PremPol),offset=log(NumPol),poisson(link=log), data=Fire200
ModelSummary2(ModFreq.4$fitted.values)
```

```
## [1] 1539.403
```

```
ModFreq.5 <- glm(NumClaim ~ Class+log(PremPol),offset=log(NumPol),poisson(link=log), data=Fire2009)
ModelSummary2(ModFreq.5$fitted.values)
```

```
## [1] 1119.328
```

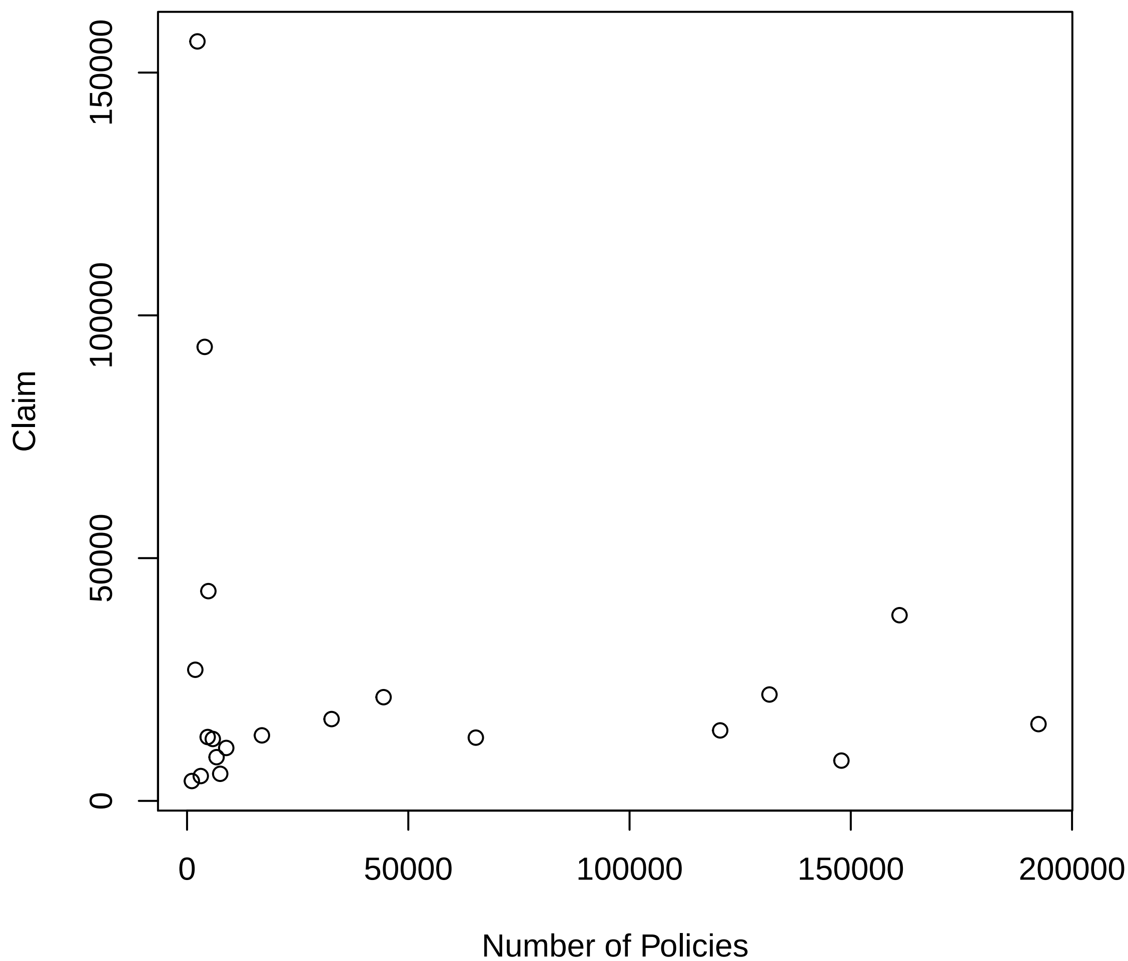
3 Severity Modeling

3.1 Graphical Approaches

To understand patterns in the claim severity, we again begin by examining several graphs.

Somewhat surprisingly, the relationship between number of policies and total claims is not clear. One would expect that for bands with more polices that we can observe greater claims. However, the figure shows that the relationship is not clear.

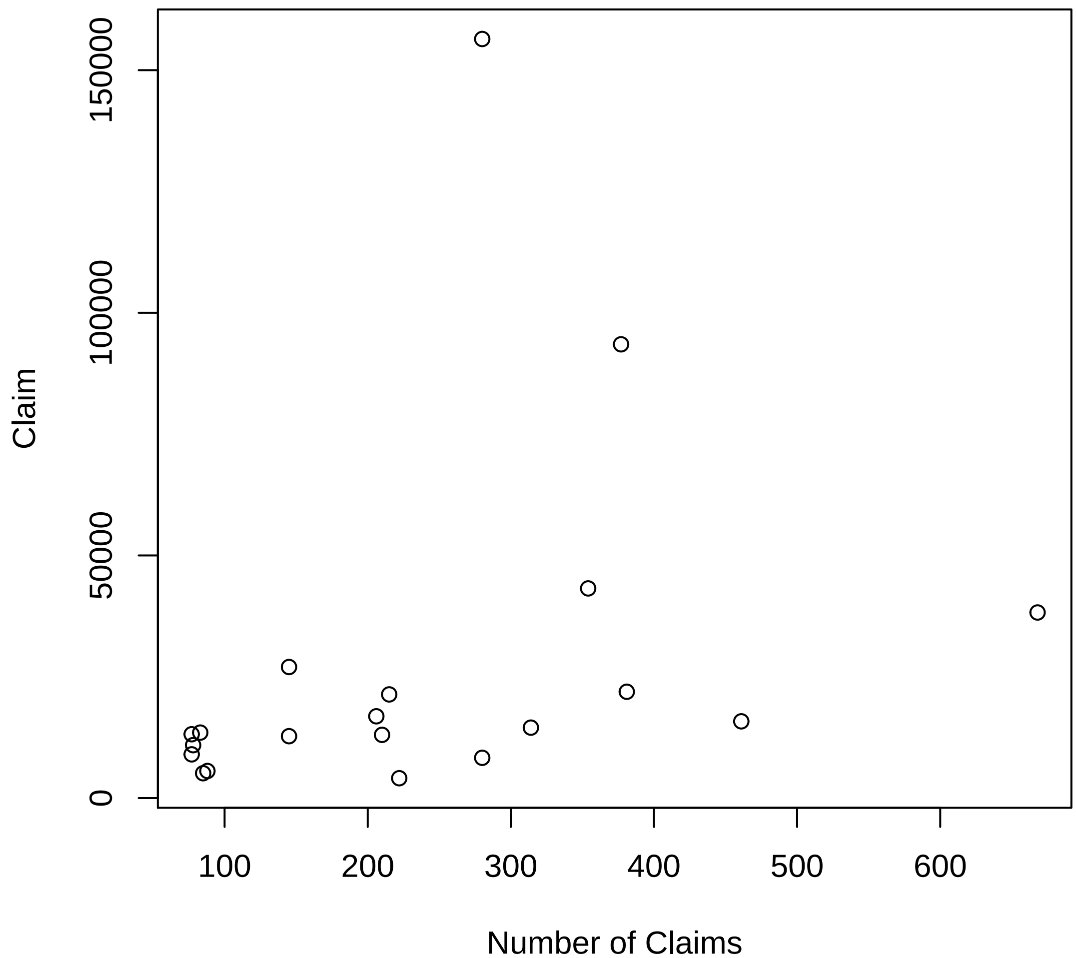
```
plot(Fire2009$NumPol, Fire2009$Claim,xlab="Number of Policies",ylab="Claim")  # RELATIONSHIP BETWEEN TO
```



Similar plots (not displayed here) of (1) total claims versus sum insured and (2) total claims versus total premiums do not reveal clear patterns.

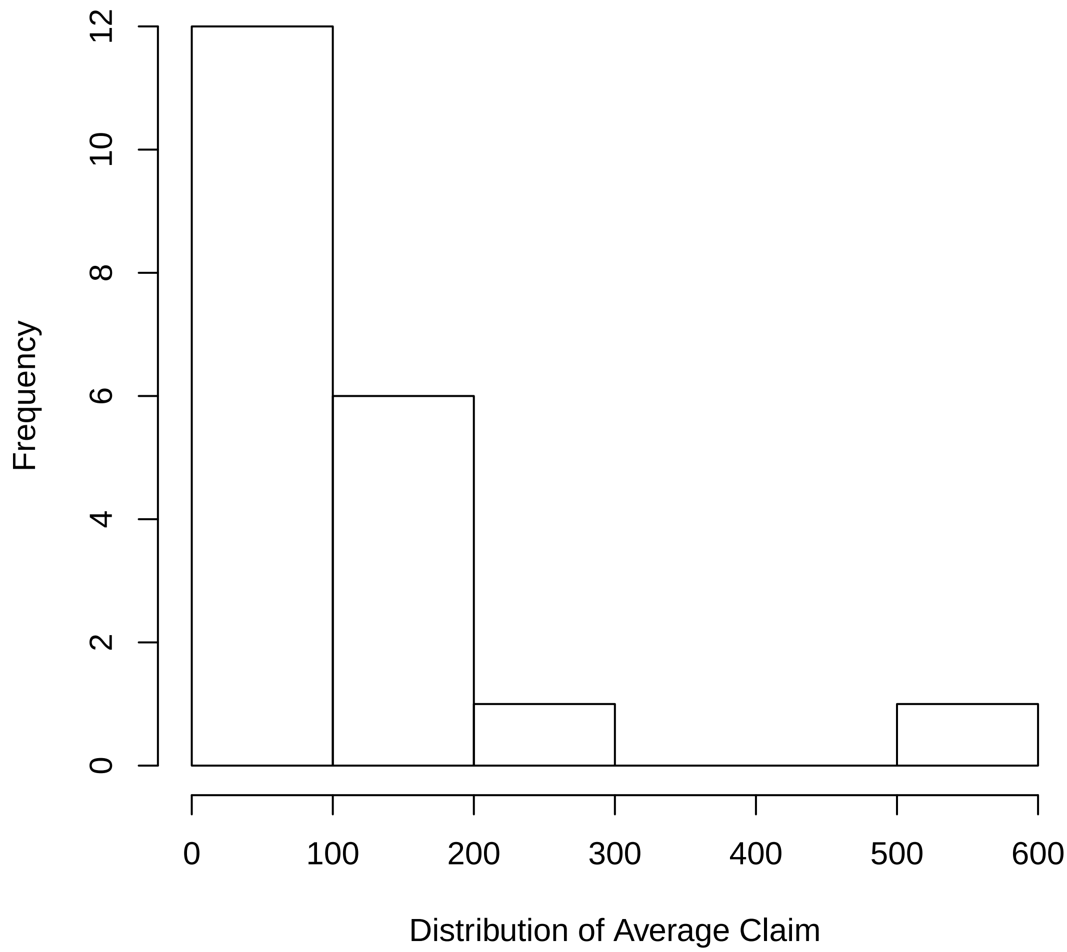
For another approach, the following figure shows a plot of total claims versus number of claims (for each class, or band). Here, we see that as the total amount of claims increases as the number of claims increases although the relationship is not linear.

```
#Fire2009$SumIns <- (Fire2009$SumFrom+Fire2009$SumTo)/2
#plot(Fire2009$SumIns,Fire2009$Claim,xlab="Sum Insured",ylab="Claim") #  SIZE OF POLICY
#plot(Fire2009$Premium,Fire2009$Claim,xlab="Premium",ylab="Claim")     # SAME WITH PREMIUMS
plot(Fire2009$NumClaim, Fire2009$Claim,xlab="Number of Claims",ylab="Claim")
```



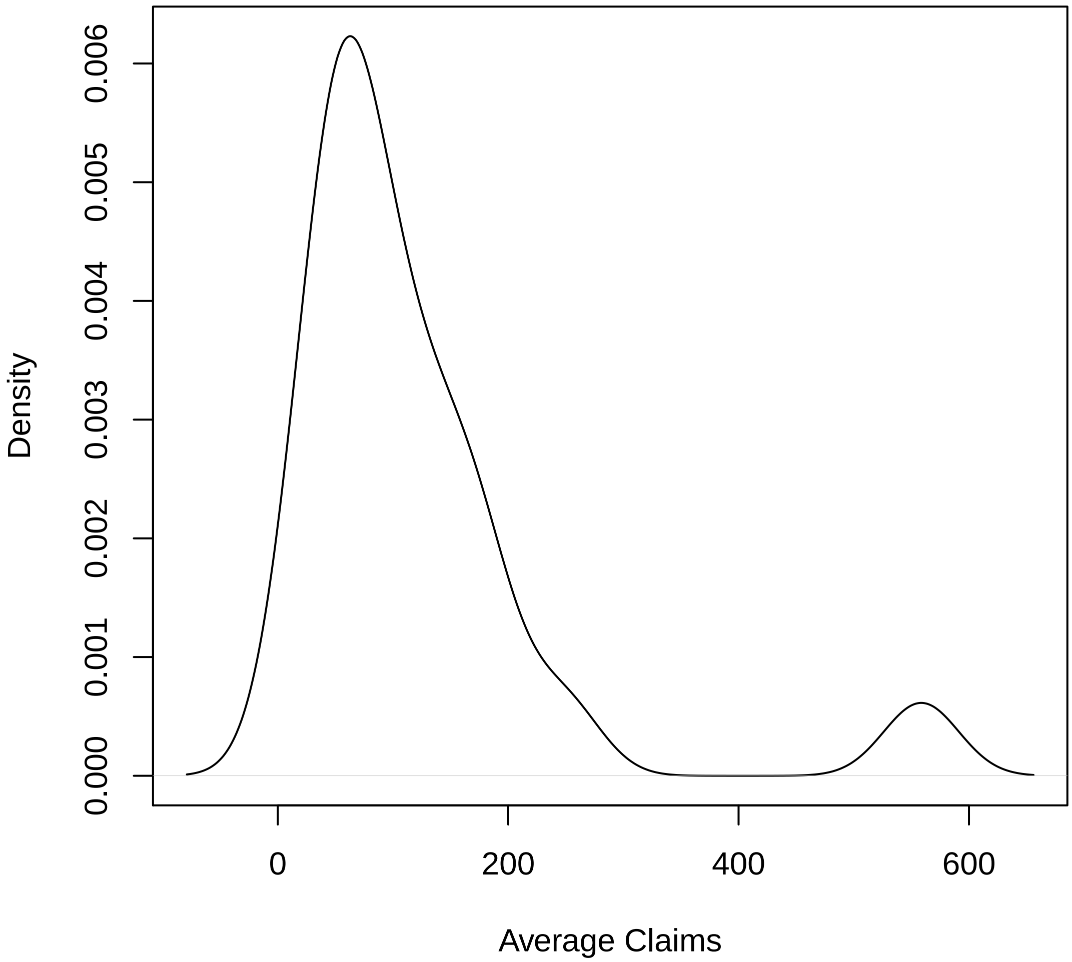
Let us instead examine claims on a per claim basis. Here is the histogram of the claims distribution. The figure shows that the distribution is right skewed with a few outlying large observations.

```
Fire2009$AvgClaim <- Fire2009$Claim/Fire2009$NumClaim
hist(Fire2009$AvgClaim, main="", xlab="Distribution of Average Claim")
```



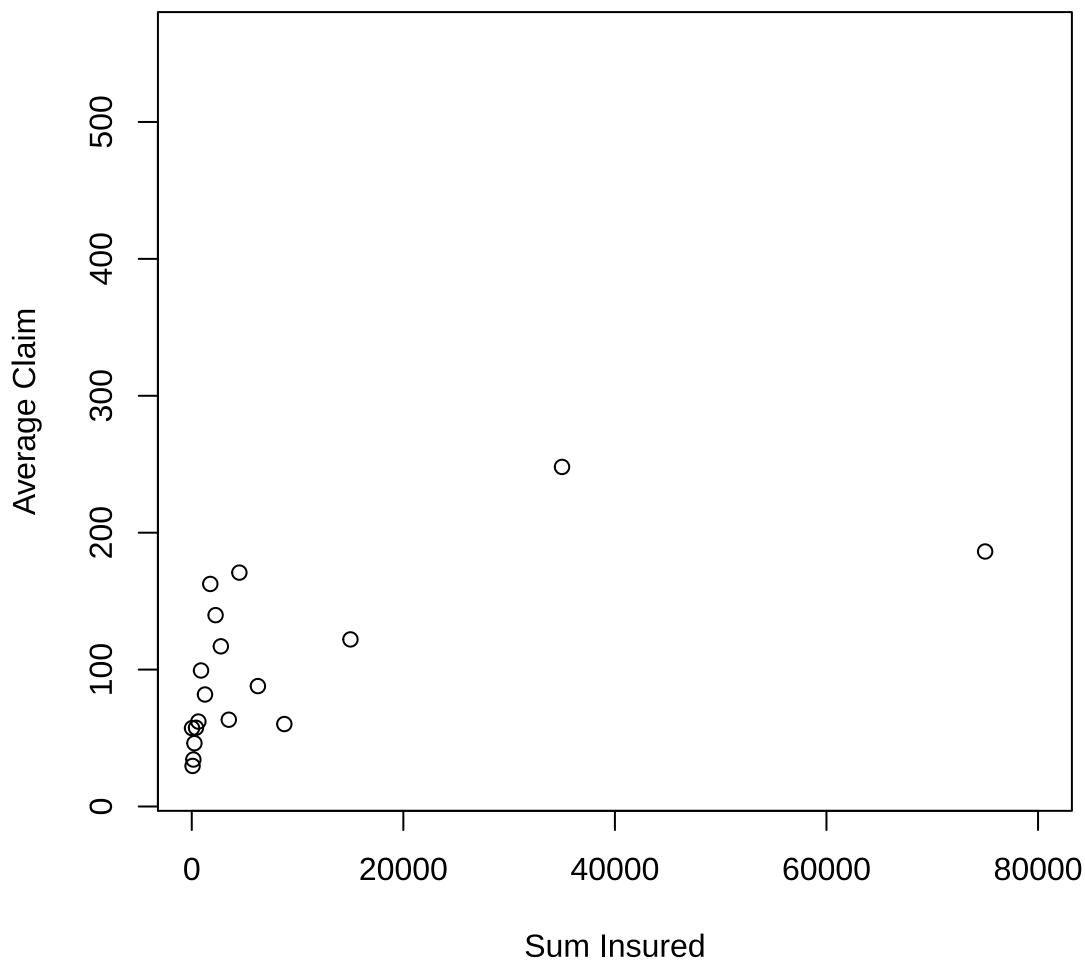
For another way of looking at the claims distribution, here is a smooth histogram of claims per policy. Like the unsmooth version, the figure shows that the distribution is right skewed with a few outlying large observations.

```r
plot(density(Fire2009$AvgClaim), main="", xlab="Average Claims")#Gaussian kernel
```



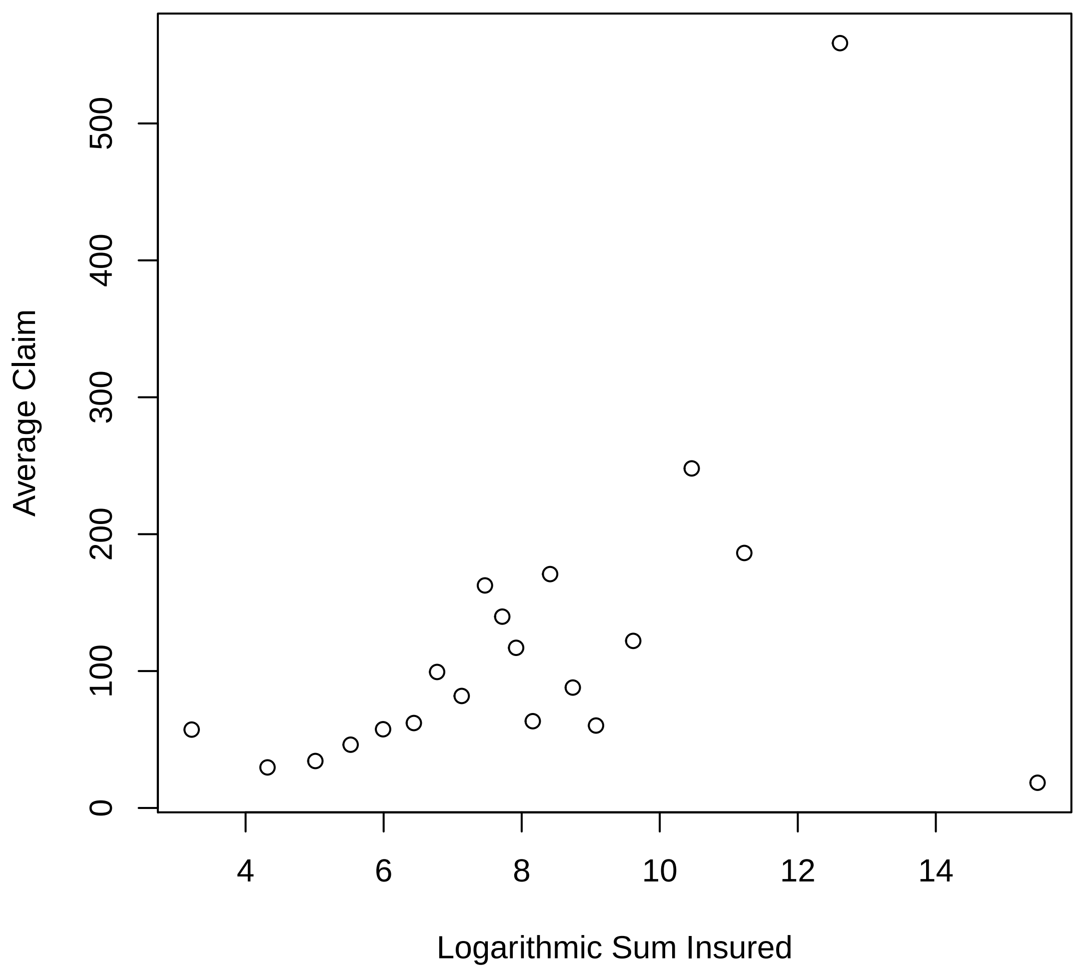
Let us know examine the claim in terms of policy size. This figure shows the average claim by sum insured. Recall that sum insured is defined to be the average of the upper and lower endpoints of the interval defining the class, or band. This figure shows that the average claim increases as sum insured increases, although the relationship is not linear. The largest class was omitted from this graph to allow a viewer to see this nonlinear pattern.

```r
#  CLAIM SEVERITY BY SUM INSURED
Fire2009$SumIns <- (Fire2009$SumFrom+Fire2009$SumTo)/2
plot(Fire2009$SumIns,Fire2009$AvgClaim,xlab="Sum Insured",ylab="Average Claim",xlim=c(0,80000))
```



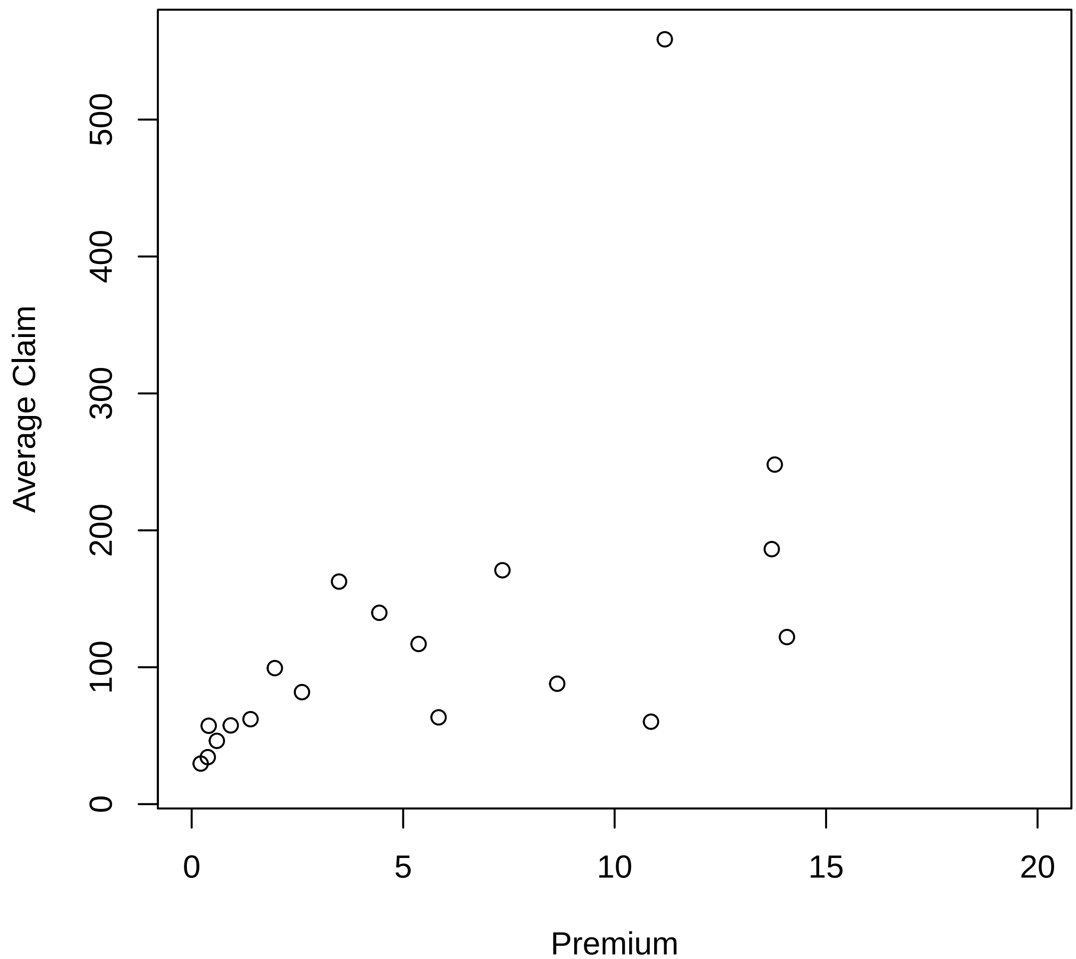
Here is a plot of average claim by logarithmic sum insured. The pattern is now clearer on this scale with the effect of the unusual highest class apparant.

```
#  CLAIM SEVERITY BY SUM INSURED
plot(log(Fire2009$SumIns),Fire2009$AvgClaim,xlab="Logarithmic Sum Insured",ylab="Average Claim")
```



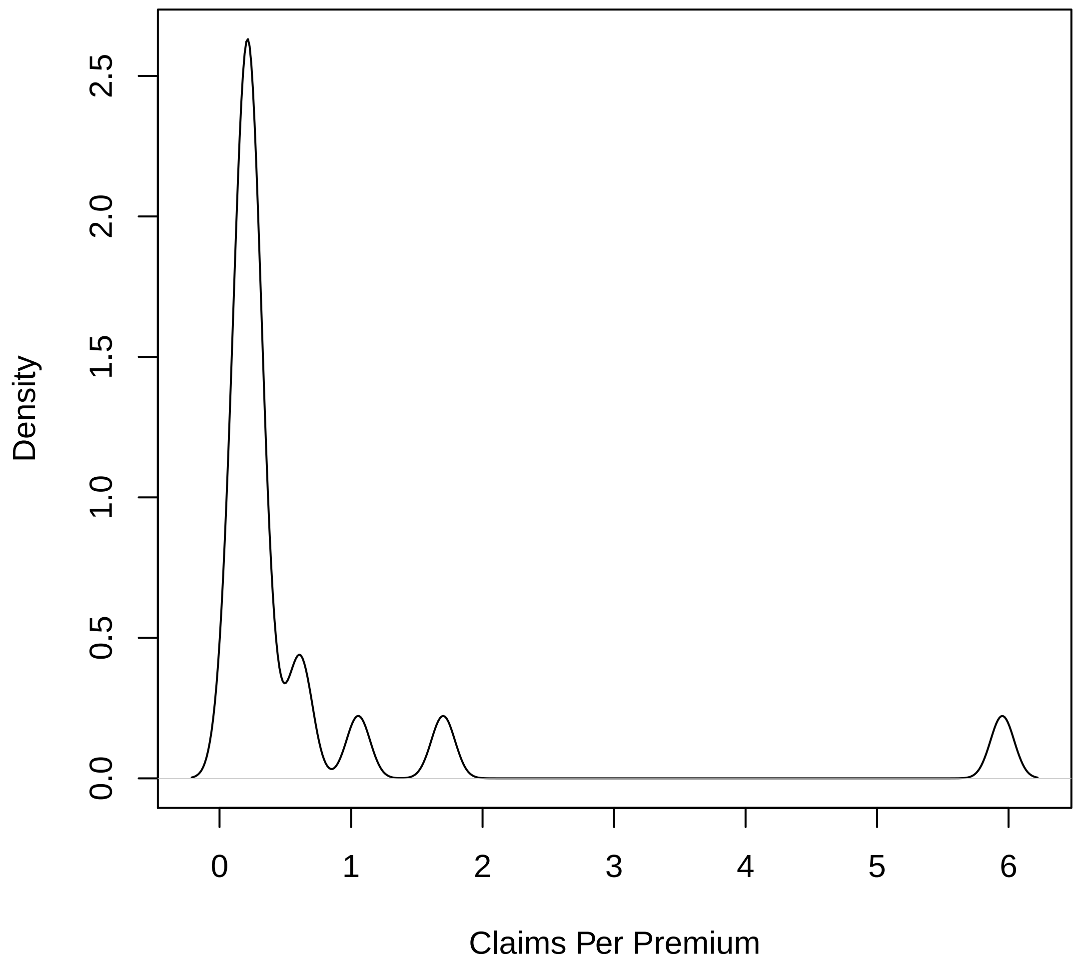
As one way of incorporating premiums, we might also examine premium on a per policy basis Here is a plot of average claim by premium per policy. This plot shows that larger claim sizes are associated with larger premiums.

```
Fire2009$PremPol <- Fire2009$Premium/Fire2009$NumPol
plot(Fire2009$PremPol,Fire2009$AvgClaim,xlab="Premium",ylab="Average Claim",xlim=c(0,20))
```



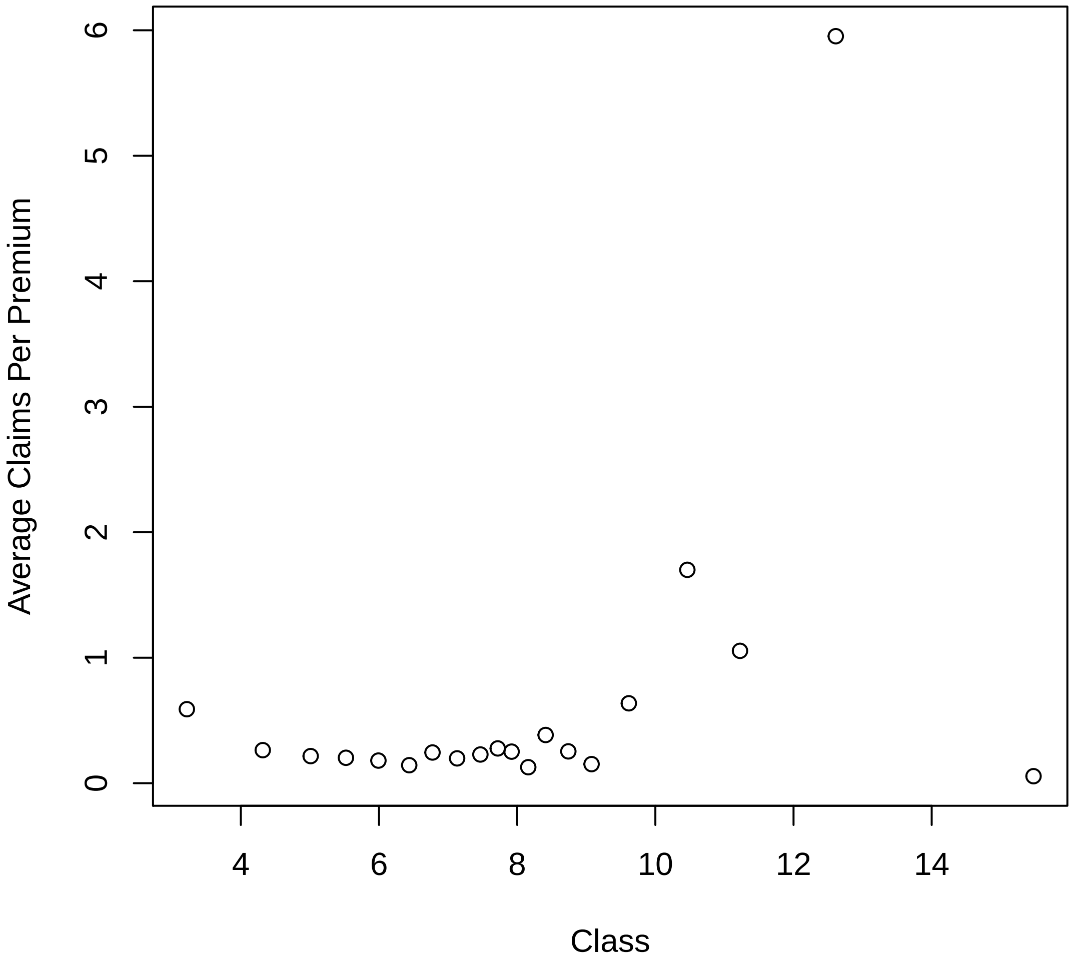
For an alternative basis, we might also consider average claims per premium (the loss ratio). Here is the smooth histogram of average claims per premium. The distribution is similar to the distribution of claims per policy, the figure shows that the distribution is right skewed with a few outlying large observations.

```
Fire2009$ClaimPrem <- Fire2009$Claim/Fire2009$Premium
plot(density(Fire2009$ClaimPrem), main="", xlab="Claims Per Premium")
```



This is a plot of average claims per premium in terms of logarithmic sum insured. There appears to be a slight "U shape" pattern, indicating large average claims for small and large sums insured when compared to intermediate sums insured.

```
plot(log(Fire2009$SumIns),Fire2009$ClaimPrem, xlab="Class", ylab="Average Claims Per Premium")
```



3.2 Fitting Claims Severity Models

As seen in Table 1 and graphical summaries, there are some classes with unusually large average claims. In particular, for class 19, corresponding to sum insured between 100,000 and 500,000 million Rupiahs, the average claim per policy is $156,418.90/280 = 558.64$ which is far in excess of the overall average claim, $544,253.84/4,746 = 114.68$.

```
Fire2009$Claim[19]/Fire2009$NumClaim[19] #  average claim for band 19
```

```
## [1] 558.6389
```

```
sum(Fire2009$Claim)/sum(Fire2009$NumClaim) #  average claim
```

```
## [1] 114.6763
```

We can start by fitting a linear model of average claims in terms of logarithmic sum insured. The following output shows that logarithmic sum insured is a statistically significant variable with a goodness of fit $R^2 = 21.54\%$.

```
summary(lm(AvgClaim ~ log(SumIns),data=Fire2009))
```

```
##
## Call:
## lm(formula = AvgClaim ~ log(SumIns), data = Fire2009)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -243.63  -27.49  -19.62   27.14  351.39
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -34.290     73.597  -0.466   0.6469
## log(SumIns)   19.152      8.616   2.223   0.0393 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 108.5 on 18 degrees of freedom
## Multiple R-squared:  0.2154, Adjusted R-squared:  0.1718
## F-statistic: 4.941 on 1 and 18 DF,  p-value: 0.03927
```

By omitting the largest class, the significance of logarithmic sum insured increases and the goodness of fit increases to $R^2 = 56.15\%$.

```
Fire2009small <- subset(Fire2009,Class<20)
summary(lm(AvgClaim ~ log(SumIns),data=Fire2009small))
```

```
##
## Call:
## lm(formula = AvgClaim ~ log(SumIns), data = Fire2009small)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -118.589  -48.609    2.515   16.923  245.531
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -166.083     65.249  -2.545 0.020909 *
## log(SumIns)   37.996      8.143   4.666 0.000222 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 81.77 on 17 degrees of freedom
## Multiple R-squared:  0.5615, Adjusted R-squared:  0.5357
## F-statistic: 21.77 on 1 and 17 DF,  p-value: 0.0002217
```

By incorporating premiums per policy, the goodness of fit increases to $R^2 = 64.12\%$. The new variable is "somewhat" statistically signficant (with a p-value of 7.766%).

```
summary(lm(AvgClaim ~ log(SumIns)+PremPol,data=Fire2009small))
```

```
##
## Call:
## lm(formula = AvgClaim ~ log(SumIns) + PremPol, data = Fire2009small)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -91.89 -49.83 -12.17  19.80 185.99
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -304.582     95.379  -3.193  0.00566 **
## log(SumIns)   67.404     17.347   3.886  0.00131 **
## PremPol      -15.451      8.195  -1.885  0.07766 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 76.24 on 16 degrees of freedom
## Multiple R-squared:  0.6412, Adjusted R-squared:  0.5964
## F-statistic:  14.3 on 2 and 16 DF,  p-value: 0.0002744
```

Another approach is to use a generalized linear model (GLM). Here is the result from a gamma regression with a logarithmic link. Note that we were able to fit the entire data set with this model (on your own, trying fitting the model without the class corresponding to the largest sum insured. There is not that much difference.)

```
GLM.model <- glm(AvgClaim ~ log(SumIns)+PremPol, data=Fire2009,
      control = glm.control(maxit = 50),
    family=Gamma(link="log"))
summary(GLM.model)
```

```
##
## Call:
## glm(formula = AvgClaim ~ log(SumIns) + PremPol, family = Gamma(link = "log"),
##     data = Fire2009, control = glm.control(maxit = 50))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.6006  -0.3280  -0.1027   0.2277   0.8146
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.14318    0.39678   5.401 4.77e-05 ***
## log(SumIns)  0.37801    0.05712   6.618 4.35e-06 ***
## PremPol     -0.07451    0.01130  -6.591 4.58e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.1793246)
##
##     Null deviance: 12.6922  on 19  degrees of freedom
## Residual deviance:  2.8126  on 17  degrees of freedom
## AIC: 204.85
##
## Number of Fisher Scoring iterations: 8
```

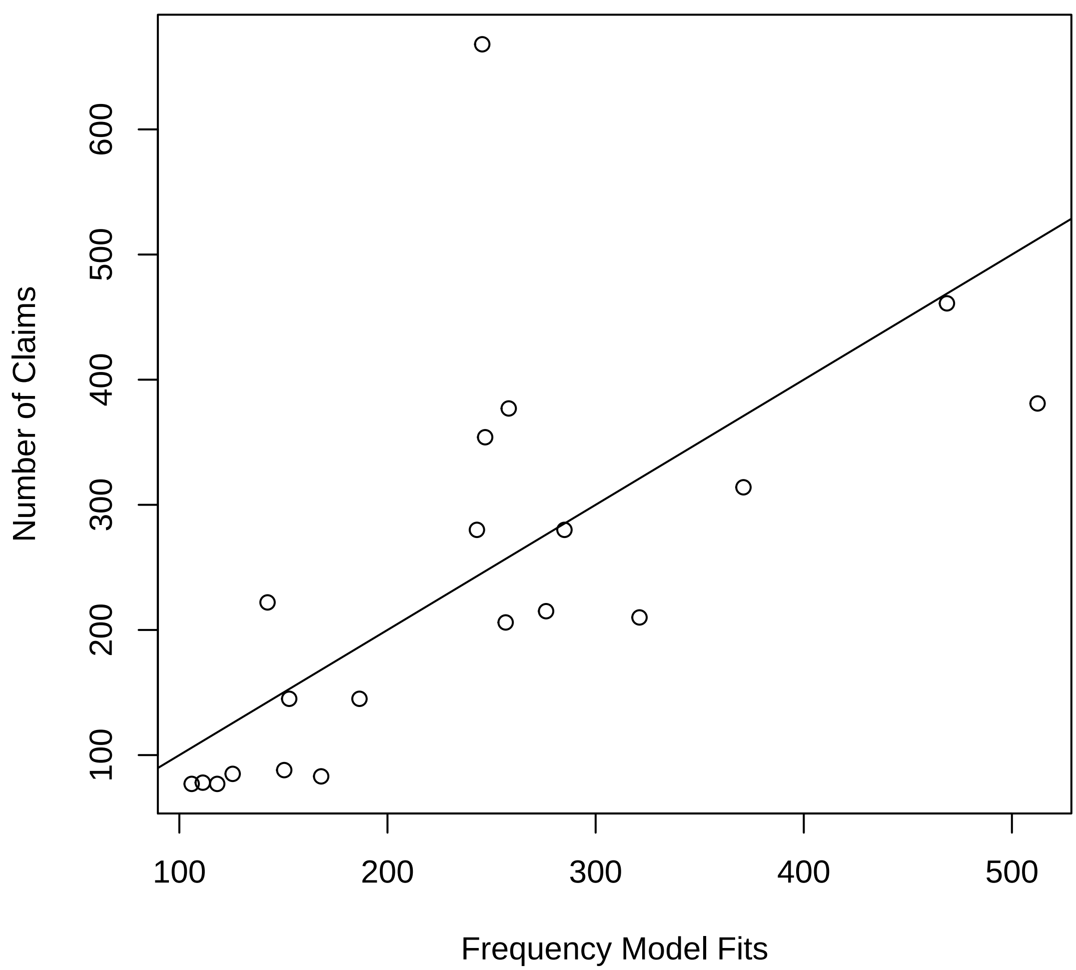
4 Summarizing the Fit

Graphically summarizing the fit is helpful for communication to many users. Here is a plot of the fitted frequency model versus the actually number of claims. Recall that only two coefficients (plus knowledge of the explanatory variables) are needed to produce the fitted values. The nonparametric (Spearman) correlation associated with this plot is 78.13%.

```
cor(ModFreq.2B$fitted.values,Fire2009$NumClaim, method="spearman")
```

```
## [1] 0.7813328
```

```
plot (ModFreq.2B$fitted.values,Fire2009$NumClaim,
      xlab="Frequency Model Fits",ylab="Number of Claims")
abline(0,1)
```

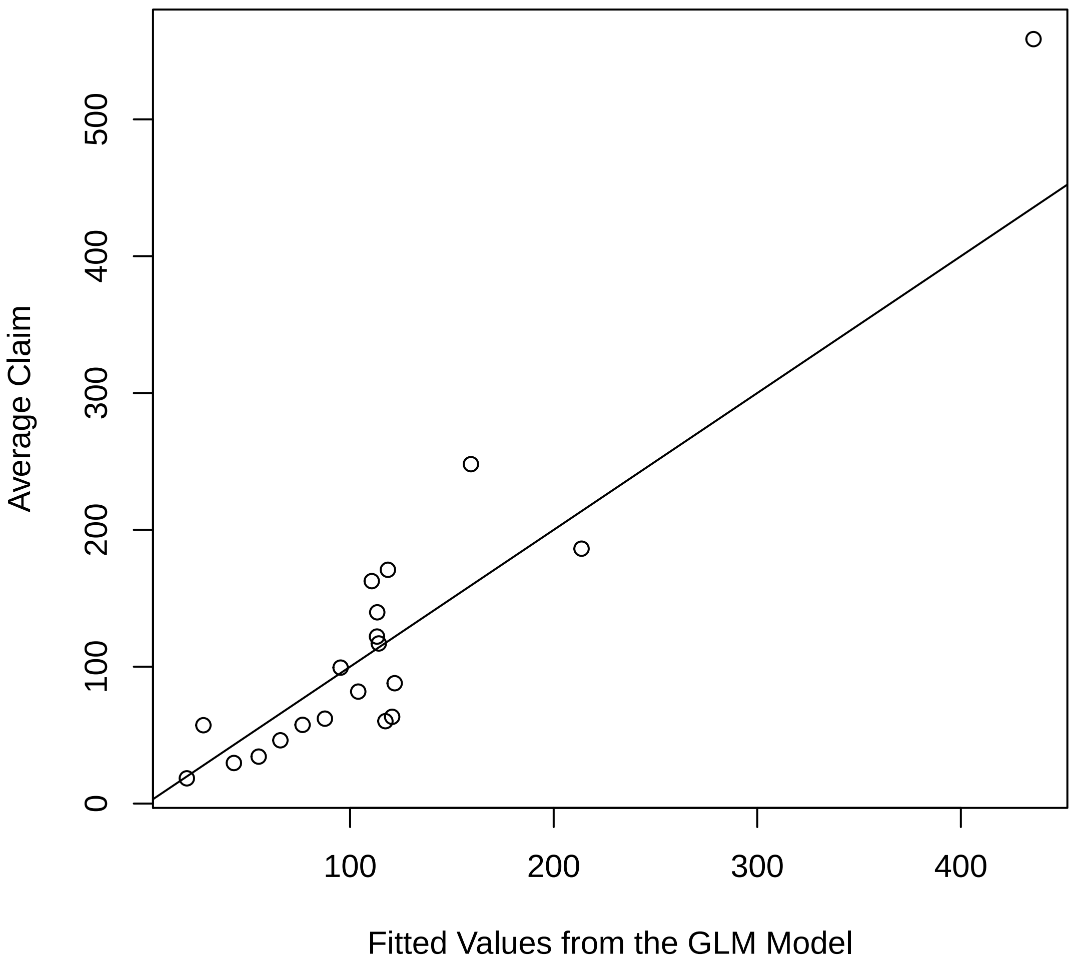


To see how well the GLM model fits the severity, here is a plot of fitted values versus average claims. Recall that only three coefficients (plus knowledge of the explanatory variables) are needed to produce the fitted values. The nonparametric (Spearman) correlation associated with this plot is 83.15%.

```
cor(GLM.model$fitted.values,Fire2009$AvgClaim, method="spearman")
```

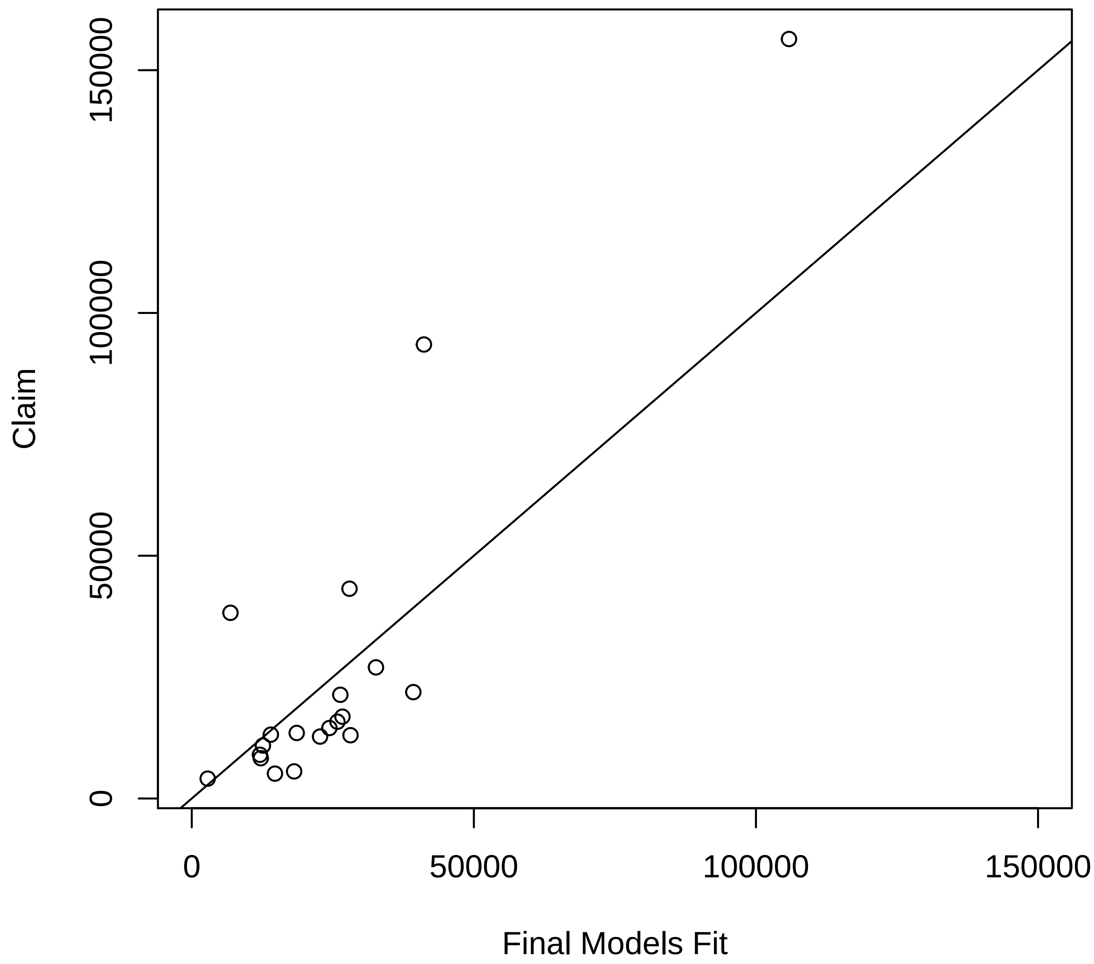
```
## [1] 0.8315789
```

```
plot(GLM.model$fitted.values,Fire2009$AvgClaim,xlab="Fitted Values from the GLM Model",
                                              ylab="Average Claim")
abline(0,1)
```



These plots look good. However, when we multiply fitted frequency times fitted severity and plot the point estimates versus total claims, we see that there is room to improve upon our models. We urge the reader to explore this.

```
FinalModelFit <- GLM.model$fitted.values*ModFreq.2B$fitted.values
plot (FinalModelFit,Fire2009$Claim,
     xlab="Final Models Fit",ylab="Claim",xlim=c(0,150000))
abline(0,1)
```




```r
cor(FinalModelFit,Fire2009$Claim, method="spearman")
```

```
## [1] 0.7112782
```