MTH5126 - Statistics for Insurance

Worksheet 3 - Solutions

Q1. Compound distribution

S has a compound distribution with Poisson parameter 4. The individual claim amounts are either 1, with probability 0.3, or 3, with probability 0.7.
Calculate the probability that $S = 4$.

We need to consider how we could get to an aggregate claim amount of 4.
This could happen in two ways:

2 claims, one for 1 and one for 3.
4 claims, all for an amount of 1.

The probability of this happening is therefore:

$$P(S=4) = P(N=2)P(X_1=1)P(X_2=3) + P(N=2)P(X_1=3)P(X_2=1)$$
$$+P(N=4)P(X_1=1)P(X_2=1)P(X_3=1)P(X_4=1)$$

Since the X_i's are identical this simplifies to:

$$P(S=4) = 2P(N=2)P(X=1)P(X=3) + P(N=4)[P(X=1)]^4$$

$$= 2 \times \frac{e^{-4}4^2}{2!} \times 0.3 \times 0.7 + \frac{e^{-4}4^4}{4!} \times 0.3^4 = 0.06312$$

Q2. Moments of compound distributions

An insurance portfolio contains policies for three categories of policyholder: A, B and C. The number of claims in a year, N, on an individual policy follows a Poisson distribution with mean λ. Individual claim sizes are assumed to be exponentially distributed with mean 4 and are independent from claim to claim. The distribution of λ, depending on the category of the policyholder, is:

Category	Value of λ	Proportion of policyholders
A	2	20%
B	3	60%
C	4	20%

Denote by S the total amount claimed by a policyholder in one year.

1. Prove that $E(S)=E\,[E(S|\lambda)]$
2. Show that $E(S|\lambda)=4\lambda$ and $Var(S|\lambda)=32\lambda$
3. Calculate $E(S)$
4. Calculate $Var(S)$

1. Let $f(s)$ denote the marginal probability density for S and let $f(s|\lambda)$ denote the conditional probability density for $S|\lambda$.

Starting with the RHS of the equation:

$E\,[E(S|\lambda)]$

$= E\,[\int_0^\infty sf(s|\lambda)ds]$

$= \sum_{i=1}^{3} p(\lambda_i) \int_0^\infty sf(s|\lambda_i)ds$

$= \int_0^\infty s \sum_{i=1}^{3} p(\lambda_i) f(s|\lambda_i)ds$, switching integration and summation

But $\sum_{i=1}^{3} p(\lambda_i) f(s|\lambda_i) = f(s)$ by definition.
and so:

$$E(E(S|\lambda)) = \int_0^\infty sf(s)ds = E(S)$$

2. Using the results for compound distributions we get:

$E(S|\lambda)=E(N|\lambda)E(X|\lambda)$, using formula for the mean of compound distributions

$=E(N|\lambda)E(X)$, since X is independent of λ

$=\lambda.\ 4 = 4\lambda$

$Var(S|\lambda) = E(N|\lambda)Var(X|\lambda) + Var(N|\lambda)[E(X|\lambda)]^2$, using formula for the variance of compound distributions

$= E(N|\lambda)Var(X) + Var(N|\lambda)[E(X)]^2$, since X is independent of λ

$= \lambda \times 16 + \lambda \times 4^2$

$= 32\lambda$

3.

$E(S) = E\,[E\,(S|\lambda)]$, using results from part 1(by the law of total expectation)

$= E\,(4\lambda)$, using results from part 2

$= 4\,E(\lambda)$

$= 4 \times (0.2 \times 2 + 0.6 \times 3 + 0.2 \times 4)$

$= 4 \times 3$

$= 12$

4. First note that $E(\lambda) = 3$ and

$Var(\lambda) = E(\lambda^2) - [E(\lambda)]^2 = 0.2 \times 2^2 + 0.6 \times 3^2 + 0.2 \times 4^2 - 3^2 = 0.4$

$Var(S) = Var[E(S|\lambda)] + E[Var(S|\lambda)]$, by the law of total variance

$= Var(4\lambda) + E(32\lambda)$, using results from part 2

$= 16 \times Var(\lambda) + 32E(\lambda)$

$= 16 \times 0.4 + 32 \times 3$

$= 102.4$

Q3. R

Before answering this question, generate the vector, X, in R using the following code:

```
set.seed(1027); X = rexp(n=1000, rate=0.01)
```

The vector X represents the gross claim sizes of 1,000 claims. The payments are to be split between an insurance company and its reinsurer under an Excess of Loss reinsurance arrangement with a retention level $M = 400$.

(i) Determine the proportion of the claims that are fully covered by the insurer. [2]

Hint: The following code might help.

```
length(X[X<=M])/length(X)
```

(ii) Generate an additional vector, *Y*, which is of the same length as *X*, such that *Y* represents the amounts to be paid by the insurer for each component of *X*. [1]

Hint: Use the `pmin` function.

(iii) Generate an additional vector, *Z*, which is of the same length as *X*, such that *Z* represents the amounts to be paid by the reinsurer for each component of *X*. [1]

An actuary assumes that the underlying gross claims distribution follows an exponential distribution of some unknown rate λ. The actuary needs to estimate λ using only the claim amounts recorded in vector *Y*.

(iv) Construct R code that calculates the log-likelihood, as a function of the parameter λ, given the claim amounts data in vector *Y*. [10]

Hint: This is estimation when sample is censored, see lecture slides.

(v) Using the function `nlm`, determine the value of λ at which the log-likelihood function reaches its maximum. [6]

Hint: The `nlm` function performs minimisation, not maximisation. However, maximising the log-likelihood function is the same as minimising the negative log-likelihood. So, we first define the function that we want to hand to `nlm` to be minimised.

Solution:

```
#Q(i) Proportion of claims fully covered by the insurer

set.seed(1027)

X=rexp(1000,0.01)

M=400

> length(X[X<=M])/ length(X)

[1] 0.987
```

So the proportion of claims fully covered by the insurer is 98.7%.

```
#Q(ii) Vector Y, same length as X, represents the amounts to
be paid by the insurer for each component of X.
```

```
Y=pmin(X, M)
```

The following code and output show that Y is indeed the same length as X, i.e. the length of Y is also 1000.

```
> length(Y)

[1] 1000
```

```
#Q(iii) Vector Z represents the amounts to be paid by the reinsurer for each component of X.

Z=X-Y
```

The following code and output show that Z is indeed the same length as X, i.e. the length of Z is also 1000.

```
> length(Z)

[1] 1000
```

```
#OR

Z=pmax(0,X-M)
```

The following code and output show that Z is indeed the same length as X, i.e. the length of Z is also 1000.

```
> length(Z)

[1] 1000
```

```
#Q(iv) Sample is censored. See lecture notes on how the complete likelihood function is made up of two parts.

#The first part relates to the 987 claims, the second part relates to the 13 claims.

#We assume all claims are independent.
```

```
#So the likelihood function for 987 claims =
(lambda^987)*exp(-lambda*sum_of_987_claims)
```

```
#And the likelihood function for 13 claims = [P(X>M)]^13 =
[exp(-lambda*M*13)

#And the complete likelihood function, L = Product of the two
likelihood functions above
```

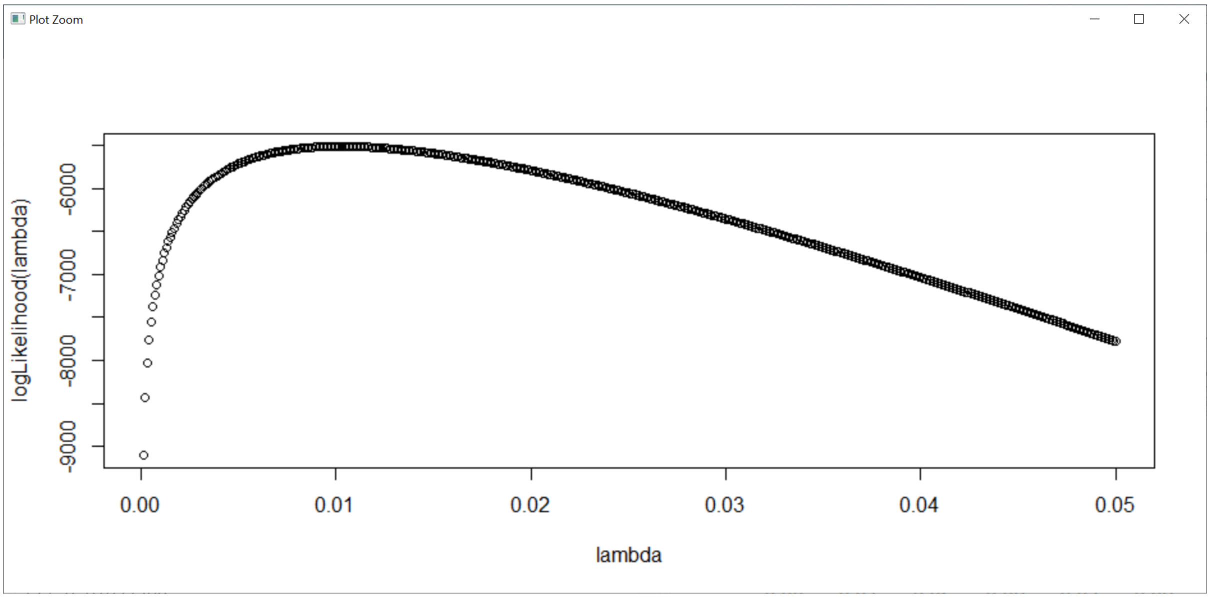
```
#Note that sum_of_987_claims + 13*M is simply the sum of all
the components in vector Y.
```

Final answer is:

```
S = sum(Y)

logLikelihood<-function(lambda){

  987*log(lambda)-lambda*S
```

```
}
```

```
#Q(v) Find the value of lambda at which the logLikelihood is at its maximum.

#The following graph plotting is not required by the question but it helps us to think about where the maximum is.

#Plot the logLikelihood just to get an idea of how it looks like.

#If necessary, adjust the graph so that we can roughly see where the max is.

#We see this happens at around lambda = 0.01.

lambda=seq(0.0001,0.05,by=0.0001)

plot(lambda,logLikelihood(lambda))
```




```
#We find lambda using numerical algorithm such as nlm.

#Note that nlm performs minimisation, not maximisation.

#However, maximising logLikelihood is the same as minimising the -logLikelihood.

#So we define the function that we want to hand to nlm to be minimised.

Function = function(lambda){

  -logLikelihood(lambda)
```

```
}

#To find out more about nlm, we run the following and look at the notes under Help.

?nlm

#p is our starting value for the iterative algorithm. From the graph we know the max is around 0.01, so set p=0.01.

nlm(f=Function,p=0.01)

> nlm(f=Function,p=0.01)$estimate

[1] 0.01023209
```

So, the estimate for lambda is 0.01023209.