Group Coursework Submission Form

Specialist Masters Programme

Please list all names of group members: (Surname, first name) 1. Bian, Yu 2. Huang, Ruiqi 3. Liu, Ching-Wei	4. Ou, Jiahua 5. **GROUP NUMBER:** **18**
MSc in: Business Analytics	
Module Code: SMM069	
Module Title: Modelling and Data Analysis	
Lecturer: Zoltan Butt	**Submission Date:** 15.06.2019

Declaration:

By submitting this work, we declare that this work is entirely our own except those parts duly identified and referenced in my submission. It complies with any specified word limits and the requirements and regulations detailed in the coursework instructions and any other relevant programme and module documentation. In submitting this work we acknowledge that we have read and understood the regulations and code regarding academic misconduct, including that relating to plagiarism, as specified in the Programme Handbook. We also acknowledge that this work will be subject to a variety of checks for academic misconduct.

We acknowledge that work submitted late without a granted extension will be subject to penalties, as outlined in the Programme Handbook. Penalties will be applied for a maximum of five days lateness, after which a mark of zero will be awarded.

Marker's Comments (if not being marked on-line):

Deduction for Late Submission of assignment:

For Students:
Once marked please refer to Moodle for your final coursework grade, including your Peer Assessment grade.

SMM069: Modelling and Data Analysis

Group coursework No. 1: Applications in EXCEL

- **Part 1: Parameter Estimation**

1.1 – Sheet 1.1

The k-th raw moment:

$$\hat{m}_k = \frac{1}{n}\sum_{i=1}^{n} x_i^k,$$

The method of moments estimators are given by:

$$\hat{\alpha} = 2\frac{\hat{m}_2 - \hat{m}_1^2}{\hat{m}_2 - 2\hat{m}_1^2},$$

$$\hat{\lambda} = \frac{\hat{m}_1\hat{m}_2}{\hat{m}_2 - 2\hat{m}_1^2},$$

Where $\hat{m}_k$ is the sample k-th raw moment.

Using the formula to first calculate the first and second raw moments of the Pareto Distribution, to get $\hat{m}_1$ = 80.88, $\hat{m}_2$ = 19908.8. Then substitute these values into the method of moments estimator formula and solve the linear equations to get alpha = 3.92, lambda = 235.91.

alpha	3.916818
lambda	235.9134

1.2 – Sheet 1.2

We used the formula below for the log-likelihood for the Pareto Distribution:

$$\log(L(\alpha,\lambda)) = n\,\log(\alpha) + \alpha\, n\,\log(\lambda) - (\alpha+1)\sum_{i=1}^{n}\log(\lambda + x_i)$$

as

$$\log(L(\alpha,\lambda)) = n\,\log(\alpha) + \alpha\, n\,\log(\lambda) - (\alpha+1)\,S(\lambda)$$

First, we calculated log(lambda + xi) for each xi. Then we used solver add-in to find the maximum log-likelihood with the constraints alpha>0, lambda>0. The MLE turned out to be -5353.11, with the corresponding alpha = 4.07, lambda = 247.46.

alpha	4.070594278	Use solver
lambda	247.4552021	
Log-liklihood	-5353.110033	Max

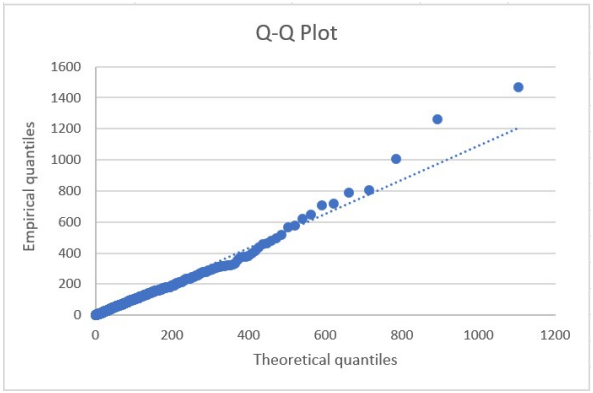
1.3 – Sheet 1.3

In order to produce the Q-Q plot, we took the following steps. First, we sorted the sample claims into ascending order and label them with indices from 1 to 1000. Then to simulate 1000 samples following theoretical Pareto distribution, we set X to be the corresponding uniform cumulative probability, where xi =

i/(1001), and use the formula below and the parameter estimates from 1.2 to get the inverse of Pareto cumulative distribution function (CDF) in column F.

$$F^{-1}(x) = \lambda\left\{(1-x)^{-1/\alpha} - 1\right\}$$

We then produced the Q-Q plot by setting F8:F1007 as the x-axis (theoretical quantiles) and B8:B1007 as the y-axis (actual quantiles).



Based on the Q-Q plot, the Pareto Distribution appears to be a fairly safe assumption for our sample data. Most of the data lies on a 45-degree straight line, with only a few data points (potential outliers) on the larger end drifting apart from the line.

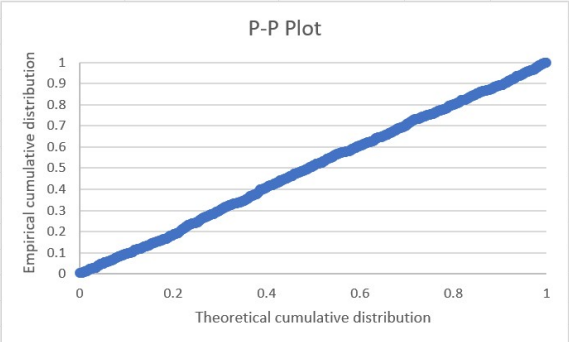
To produce the P-P plot, which compares the empirical cumulative distribution function of a data set with a specified theoretical cumulative distribution function, we used the formula below to find the CDF for the Pareto Distribution in column H, and we want to compare it with the uniform CDF in column D.

The density and distribution functions of a Pareto variate are given by:

$$f(x) = \frac{\alpha\lambda^{\alpha}}{(\lambda+x)^{\alpha+1}}, \qquad x > 0,$$

$$F(x) = 1 - \left(\frac{\lambda}{\lambda+x}\right)^{\alpha}, \qquad x > 0,$$

Then by setting D8:D1007 as the x-axis and H8:H1007 as the y-axis, we have obtained the desired P-P plot.

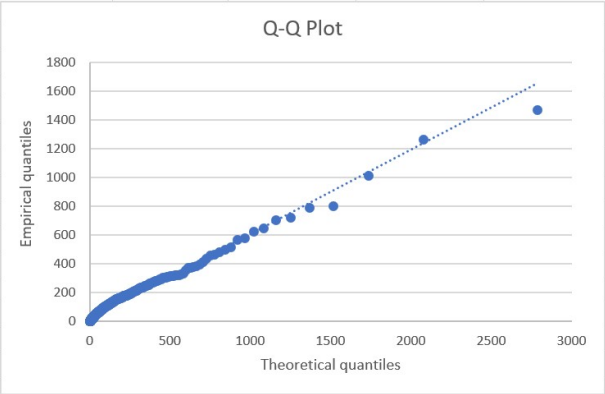


From this P-P plot, we can see clearly that it follows the 45-degree straight line almost perfectly, with extremely minor deviation. This again confirms that the Pareto distribution is a very good match for our sample data.

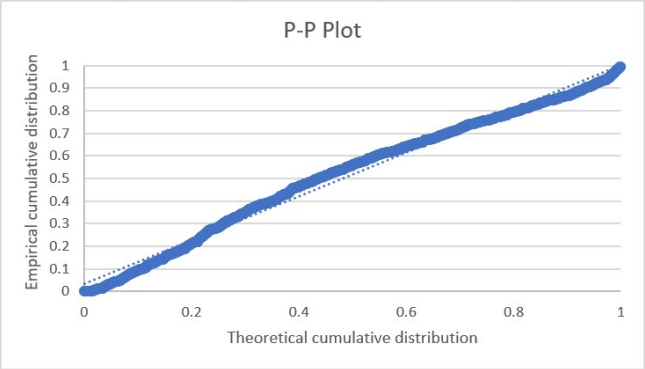
1.4 – Sheet 1.4

Similar to 1.3, we first sort both the claims and the log of claims in ascending orders and label the latter with indices from 1 to 1000, and then produce the same uniform CDF as in 1.3. We can also calculate the mean and standard deviation of log claim to be 3.65 and 1.38, respectively.

To produce the Q-Q plot, we use the function LOGNORM.INV(x, mean, std) to find the inverse of LogNormal CDF, as in column H. Then plotting the theoretical quantiles (H8:H1007) against the empirical quantiles (C8:C1007) gives us the Q-Q plot.



Then we need to create the P-P plot. This time we need the CDF of the LogNormal Distribution, which can be achieved by the function LOGNORM.DIST(x, mean, std) as in column I. Again, plotting the LogNormal CDF against the uniform cumulative probability gives us the P-P plot as required.



From the above Q-Q and P-P plots, we could tell that LogNormal Distribution is also a pretty good match to our sample data. Since the majority of points fall on the 45-degree straight line, with only a few extreme large values deviating from the line. In terms of the P-P plot, the points generally lie on the 45-degree straight line, meaning that LogNormal Distribution fits our data quite well. However, it is still noticeable that there are slight deviations from the straight line at the beginning, middle and end of the plot.

1.5

Both distributions are reasonable matches to the claim sizes data. However, the Pareto Distribution does seem to be more appropriate, because of its (almost) perfectly aligned P-P plot, compared to the slighted deviated P-P plot for LogNormal.

- **Part 2: Monte Carlo simulation**

2.1 – Sheet 2.1

According to the mean £2,500 and standard deviation £10,000, we can apply E(X)= 2500 and Var(X)=$10{,}000^2$ to the formulas of Pareto distribution: $E(X) = \frac{\lambda}{\alpha-1} = 2500$ and $Var(X) = \frac{\lambda^2\alpha}{(\alpha-1)^2(\alpha-2)} = 10{,}000^2$ with $\alpha > 2$, then calculate the $\alpha = 2.13$, λ=2833.33 in cell F2 and F3.

Enter in cells B8:B5007 the integers from 1 to 5000 and label them SimNr. Then we produce 5000 simulations from one independent random variables U (C7:C5007) using RAND(), which is uniformly distributed between 0 and 1. Similar as 1.3, with applying U to the inverse of the Pareto CDF, with $\alpha = 2.13$, λ=2833.33, column F8:F5007 named "Inverse of CDF" is the sample of 5,000 losses with no insurance (X) using the inverse transform method.

The probability of sample losses $\hat{p}$ (with no insurance) above £10,000 is 0.0386 (cell I8) using COUNTIF(E8:E5007,">="&10000)/5000, calculating by the number of losses above £10,000 divided by the total number of the losses.

Sample probability of losses (p_hat) above £10,000:
0.0386

F8:F5007 show the sorted sample losses from smallest to largest, and G8:G5007 indicate the cumulated probability of theoretical losses generated by SimNr/5001. According to these theoretical cumulative probabilities corresponding to each sorted sample loss value, therefore we can find the cumulative probabilities corresponding to 10,000 and subtract it from 1 as we only need the probability for value bigger than 10,000 rather than smaller than 10,000. As a result, the actual value of p from the theoretical distribution can be calculated to be 0.0388 (cell I12) using 1-VLOOKUP(10000,F8:G5007,2,TRUE).

Theoretical probability of losses (p) above £10,000:
0.0388

The 90% confidence interval for p is:

$$p \pm z_{0.05}\sqrt{\frac{p(1-p)}{n}}$$

p = E(p) should be the theoretical probabilities 0.0388 and the variance could be calculated as Var(p) = p(1-p)/n = 0.000007. Then we calculate the Z confidence level using NORM.S.INV((1+90%)/2) = 1.644854. So a 90% confidence interval for p is $p \pm z_{0.05}\sqrt{\frac{p(1-p)}{n}}$ = 0.0388$\pm$1.644854*$\sqrt{0.000007}$ = (0.0343, 0.0433).

90% CI (p)	
E(p) = P(Losses>=10,000)	0.0388
Var(p) = p(1-p)/n	0.000007
Conf. level	90%
Z - Conf. level	1.644854
Conf. intv const	0.004492
Lower bound	0.0343
Upper bound	0.0433

2.2 – Sheet 2.2

Our group have different understandings of "within ±3% of the actual value", so we make two solutions by different understandings. In the first version, the 3% is the absolute range, so the confidence interval is [actual value-3%, actual value+3%]; second one considers the 3% is relative to the actual value, so the confidence interval should be within [actual value*(1-3%), actual value*(1+3%)].

Version 1:

Based on the result of the theoretical probability p = 0.0388 calculated in part 2.1, we use solver to find how many samples should be simulated by setting the objective number by changing variable cell J24. The confidence interval constant should be within 0.03 so the subject to the constrains can be J20 = 0.03, then we can get the result that n = 112.09, so 113 losses need to be simulated to be 90% confident that the estimate for p in part 2.1 is within ±3% of the actual value.

Version 2:

If the 90% confidence interval need to be within the 3% of actual value, the maximum shift from the actual value should be 3%*0. 0.0388(theoretical probability) = 0.001164, and with other steps maintaining the same in version 1, produce the result n = 74489.

90% CI (p)				
E(p) = P(Losses>=10,000)	0.0388	0.0388		
Var(p) = p(1-p)/n	0.000333	0.000001		
Conf. level	90%	90%		
Z - Conf. level	1.644854	1.644854		
Conf. intv const	0.03	0.001164	0.001164	=0.03*J16
Lower bound	0.008792	0.037628		
Upper bound	0.068792	0.039956		
n	112.0919	74488.02	Using solver	

2.3 – Sheet 2.3

To calculate the individual payments by the insurer (Y) to claimant, before recovery of reinsurance, first we classify the payments into 3 types in column "Type of payments" according to the losses (X) range, using IF("X"<=2000,1,IF(AND("X">2000, "X"<=127000),2,IF("X">127000,3,0))). Then in the next column, we calculate the payments based on types using IF("type"=1,0,IF("type"=2,0.8*("X" -2000),IF("type"=3,100000,"Error"))). Then with AVERAGE() and STDEV.S(), mean E(Y) and standa rd deviation SD(Y) of the payments can be found out as 1072.75 and 3944.84. Finally, we can cal culate the 95% confidence interval for the expected value E[Y], showing the following result (963.4 042, 1182.0911).

95% CI	
E(Y)	1072.7476
SD(Y)	3944.843146
Conf. level	95%
Z - Conf. level	1.959964
Conf. intv const	109.343464
Lower bound	963.4042
Upper bound	1182.0911

2.4 – Sheet 2.4

To reduce the 95% confidence interval to a quarter of the range estimated in part 2.3, the confiden ce interval constant should be a quarter of the confidence interval constant of part2.3, which is 10 9.34/4= 27.34. Again, we use solver to set the objective n and with the constrains H11 = 27.34, th en we can get the result that n = 80000, so about 80000 losses need to be simulated to reduce th e 95% confidence interval to a quarter of the range estimated in part 2.3.

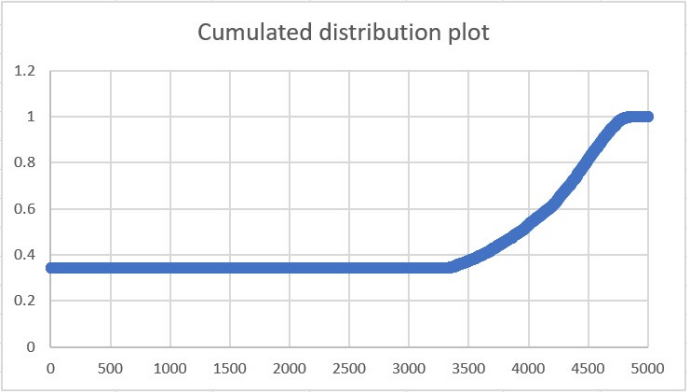
95% CI		
E(Y)	1072.7476	
SD(Y)	3944.843146	
Conf. level	95%	
Z - Conf. level	1.959964	
Conf. intv const	27.3358643	27.33587
Lower bound	1045.41177	
Upper bound	1100.0835	
n	80000.0098	Using solver

2.5 – Sheet 2.5

To stimulate the net costs Z, using the formulas given by the question, we use the 5000 simulation s from part 2.1. First we classify the net costs Z into 3 types in column “Type of net costs” by u sing IF(“losses”<=2000,1,IF(AND(“losses”>2000,<=17000),2,IF(“losses”>17000,3,0))). Then based on t hese types, the net costs (Z) can be calculated in column E using IF(“type”=1,0,IF(“type”=2,0.8*(“l osses”-2000),IF(type=3,12000,"Error"))). The mean and standard deviation of net costs therefore can be calculated as following:

mean	881.3775
SD	2189.959

After calculating the mean and standard deviance, we can calculate the cumulative distribution funct ion of the net costs using the formula “NORM.DIST(“net costs”,881.377,2189.96,TRUE)” and sort t he result in the next column. Then use the column “SimNr” and “CDF (Sorted)” to plot the cumul ated distribution plot as follow:



The median of the net costs can be calculated by MEDIAN() as 0 and the third quartile can be ca lculated by QUARTILE.INC() as 557.777.

median	0
3rd quartile	557.7768

2.6 – Sheet 2.6.1 to 2.6.5

1) Sheet 2.6.1

The first step is again to generate 5,000 simulations of independent numbers U using RAND() functions which returns pseudo-random numbers from the interval [0, 1]. As now we assume individual loss to follow a logNormal distribution but with the same mean of 2,500 and same standard deviation of 10,000, we now use LOGNORM.INV() function to calculate the sample value of X. The syntax within LOGNORM.INV() function requires the mean and standard deviation of ln(X), therefore we use the following formula to calculate the required parameter of ln(X):

The parameters μ and σ can be obtained if the arithmetic mean and the arithmetic variance are known:

$$\mu = \ln\left(\frac{\mathrm{E}[X]^2}{\sqrt{\mathrm{E}[X^2]}}\right) = \ln\left(\frac{\mathrm{E}[X]^2}{\sqrt{\mathrm{Var}[X] + \mathrm{E}[X]^2}}\right),$$

$$\sigma^2 = \ln\left(\frac{\mathrm{E}[X^2]}{\mathrm{E}[X]^2}\right) = \ln\left(1 + \frac{\mathrm{Var}[X]}{\mathrm{E}[X]^2}\right).$$

which results 6.41 for mean and 2.83 for standard deviation:

mu	6.407439339
sigma^2	2.833213344

Now the sample value of X can be calculated in column D, and we sort the sample losses, and calculate the theoretical cumulative probabilities corresponding to each of the sample loss value in column F.

Now the sample probability for insurance loss bigger than 10,000 can be calculated using COUNTIF() function to count the number of sample value which are larger than 10,000 and divided by the total sample size of 5,000. This results a sample probability of 0.0474.

Sample probability of losses (p_hat) above £10,000:
0.0474

Since we have already calculated the theoretical cumulative probabilities corresponding to each sample loss value sorted from smallest to largest, therefore we just need to find the cumulative probabilities corresponding to 10,000 using VLOOKUP() and subtract it from 1 as we only need the probability for value bigger than 10,000 rather than smaller than 10,000.

Theoretical probability of losses (p) above £10,000:	
0.0476	

The theoretical probability therefore is calculated to be 0.0476, very close to the sample probability of 0.0474. The reason for the slight difference might be due to as long as the sample size is not infinity, the probability we random generated can never be completely continuous therefore the discrete probabilities could cause the difference.

To calculate the 90% confidence interval, E(p) should be the theoretical probabilities and the variance could be calculated as p(1-p)/n. We use NORM.S.INV() function to calculate the Z value and multiply by the standard deviation which is the square root of the variance and results in the confidence interval [0.0426,0.0525].

90% CI (p)	
E(p) = P(Losses>=10,000)	0.0476
Var(p) = p(1-p)/n	0.000009
Conf. level	90%
Z - Conf. level	1.644854
Conf. intv const	0.004952
Lower bound	0.0426
Upper bound	0.0525

2) Sheet 2.6.2

As mentioned in 2.2, the understanding of "within ±3% of the actual value" are different within our groups, so we gave two different solutions based on different understandings.

Version 1:

In the first version, the 3% is the absolute range, so the confidence interval is [actual value-3%, actual value+3%]. Therefore in our layout this mean the confidence interval constance should be 0.03. Following the same steps in part 2.2, solver results the n to be 136.25, which could be round up to 137. Therefore at least 137 losses are needed to simulate to be 90% confident that the estimate for p in part 2.6.1 is within ±3% of the actual value.

Version 2:

If the 90% confidence interval need to be within the 3% of actual value, the maximum shift from the actual value should be 3%*0.0476(theoretical probability) = 0.001428. With everything else the same, we just change the condition in solver to be 0.001428 rather than 0.03 and resulting sample size to be 60162.

90% CI (p)				
E(p) = P(Losses>=10,000)	0.0476	0.0476		
Var(p) = p(1-p)/n	0.000333	0.000001		
Conf. level	90%	90%		
Z - Conf. level	1.644854	1.644854		
Conf. intv const	0.03	0.001428	0.001428	=0.03*I16
Lower bound	0.01759	0.046163		
Upper bound	0.07759	0.049018		
			Using solver	
n	136.2562	60161.41		

3) Sheet 2.6.3

We copy the simulated losses from previous and using IF() function to categorise them into $X \leq 2000$, $2000 < X \leq 127000$ and $X > 127000$, then calculating the corresponding payment according to the formula, which follows the same steps in part 2.3. The mean E(Y) and standard deviation SD(Y) of sample payments can be calculated thereafter and as before we use NORM.S.INV() to generate the Z statistics and calculating the 90% confidence interval is [1077.92, 1377.76].

95% CI	
E(Y)	1227.8426
SD(Y)	5408.597606
Conf. level	95%
Z - Conf. level	1.959964
Conf. intv const	149.9159221
Lower bound	1077.9267
Upper bound	1377.7586

4) Sheet 2.6.4

We first calculate the target range by multiply the previous range by 0.25, which result in the absolute shift to the actual value to be 37.48. Again, we use solver to set the n as variable and objective and set conditions to make confidence interval constance equal to 37.48. This result an n = 80000, which means that around 80000 sample losses is needed to reduce the confidence interval to the target level.

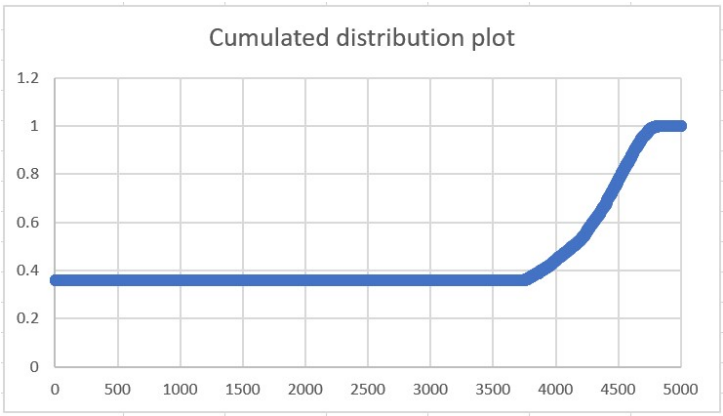
95% CI		
E(Y)	1227.8426	
SD(Y)	5408.597606	
Conf. level	95%	
Z - Conf. level	1.959964	
Conf. intv const	37.47898053	37.47898
Lower bound	1190.363651	
Upper bound	1265.321612	
		Using solver
n	80000.00001	

5) Sheet 2.6.5

Following the same methods in 2.5, we use IF() function to categorise the type of net costs and calculating the corresponding net costs. The mean and standard deviation of net costs therefore can be calculated as following:

mean	881.3775
SD	2189.959

We use NORM.DIST function to calculate the cumulative probabilities of individual net costs and sort them from smallest to the largest. The plotted cumulated distribution function shows as follows:



The median and the 3rd quartile are both calculated to be 0.

6) Comparison of the results from Pareto and LogNormal distribution

By comparing the result from LogNormal assumption with Pareto assumption, the result shows that assumptions do significantly impact the simulation result. For example, the theoretical probability for losses bigger than 10,000 is calculated to be 0.0388 under Pareto assumption while 0.0476 under LogNormal assumption. 74489 sample is needed to obtain a 90% confidence interval within 3% of the actual value under Pareto assumptions while only 60161 sample is needed under LogNormal assumptions, indicating that the LogNormal model converge much faster to the true value by increasing sample size, however in calculating the sample size in reducing the range of confidence interval for payments the n under both assumptions are similar around 80000. The average payment and net costs calculated under both assumptions are different, whether considered significant or not. The distribution plots are looking different as LogNormal's is sharper than Pareto's.