General Insurance

To Reinsure Claims Seperately or per Policy?

A STATISTICAL ANALYSIS ON CLAIMS HISTORY TO EVALUATE THE BETTER REINSURANCE COVER

PRESENTED BY Edmund Liman (1073769)

Agenda

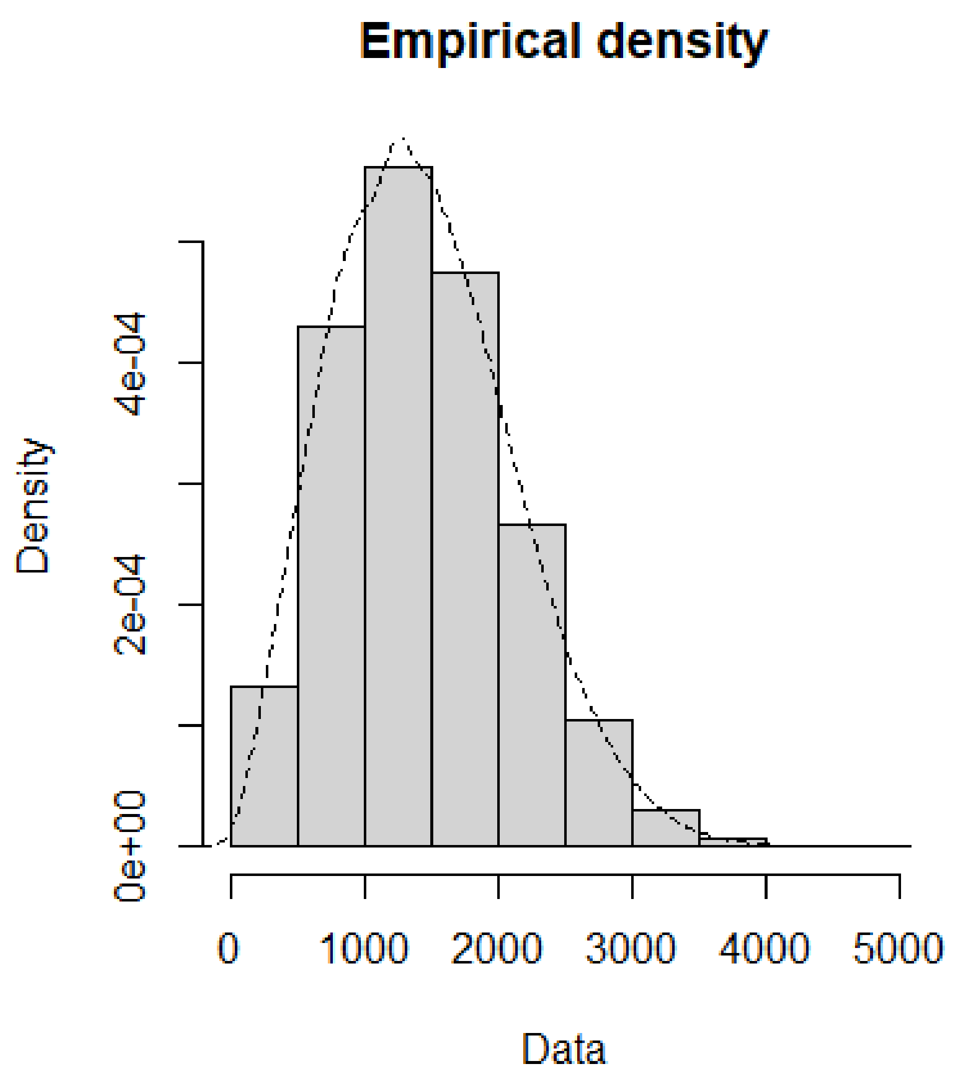
General Insurance

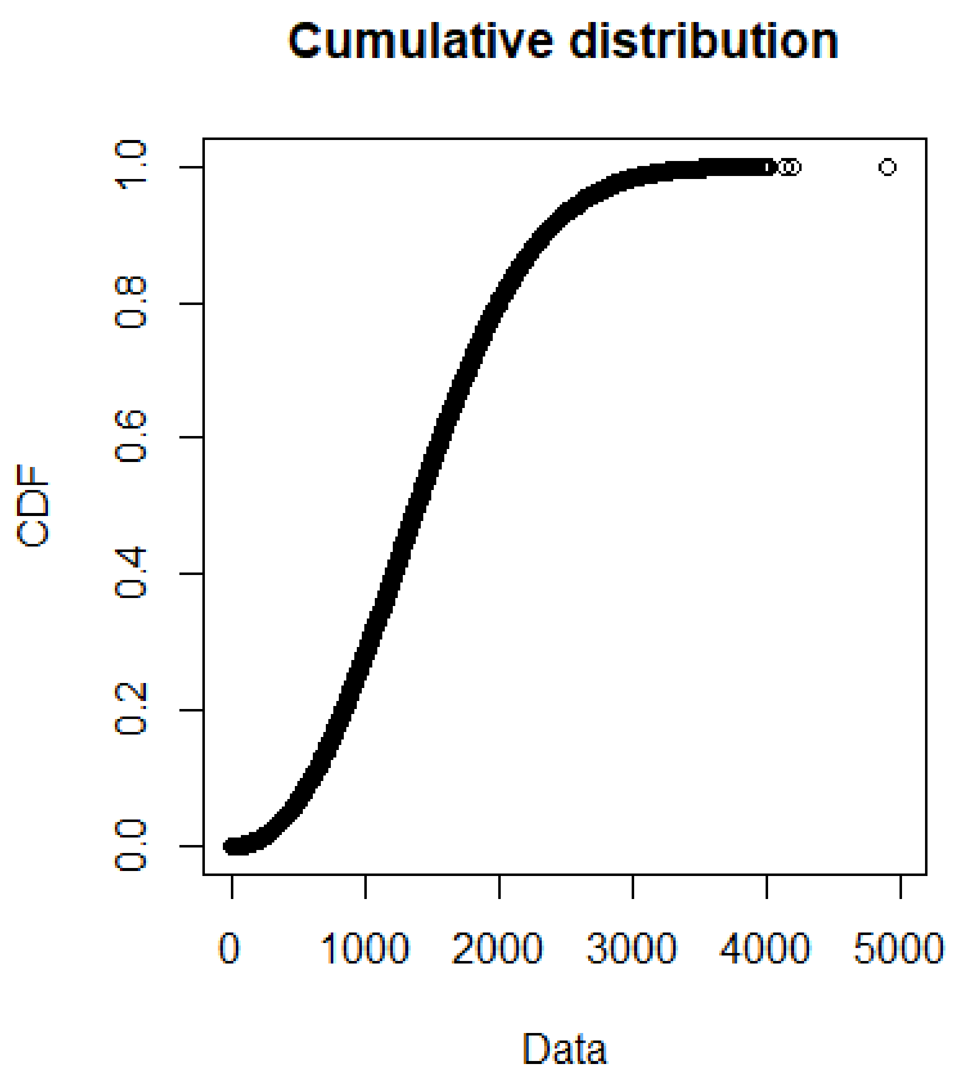
Introduction & Problem Statement

- To compare between two reinsurance covers:
    - Type 1: Excess of loss on each claim, covering the excess 1.5 times the expected claim amount.
    - Type 2: Excess of loss on each policy aggregate cost, covering the excess 1.5 times the expected amount.
- based on past data given by General Insurance.
- We will look on claims severity, claims frequency and costs to determine the best reinsurance cover for the company.

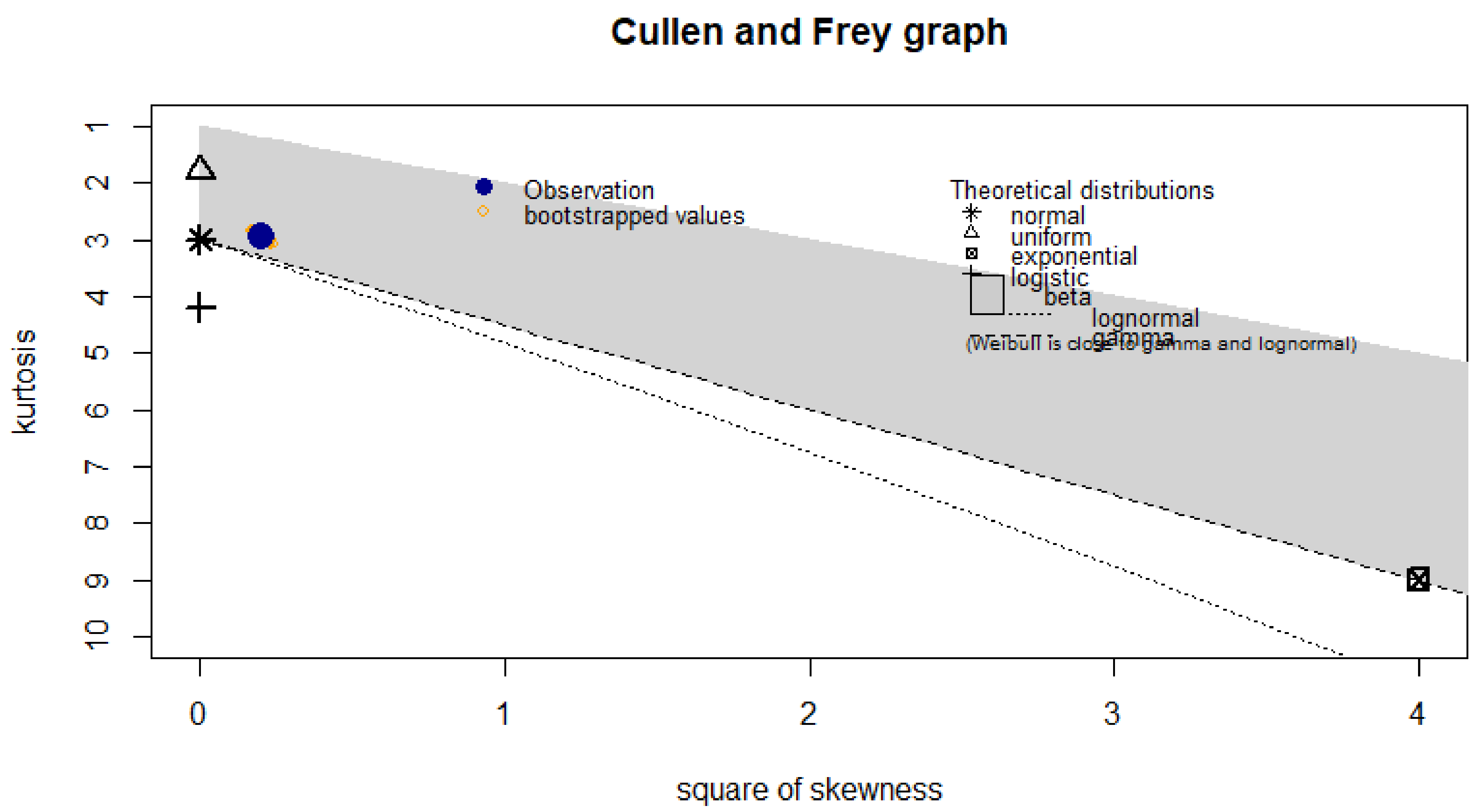
Claims Severity Analysis

- 25,041 rows of data
- 9,099 unique policy numbers
- 593 policies with no claims
- 24,448 claims
- Minimum claim amount = 11.91567
- Maximum claim amount = 4907.379
- Mean claim amount = 1443.124
- Coefficient of variation = 0.4633643





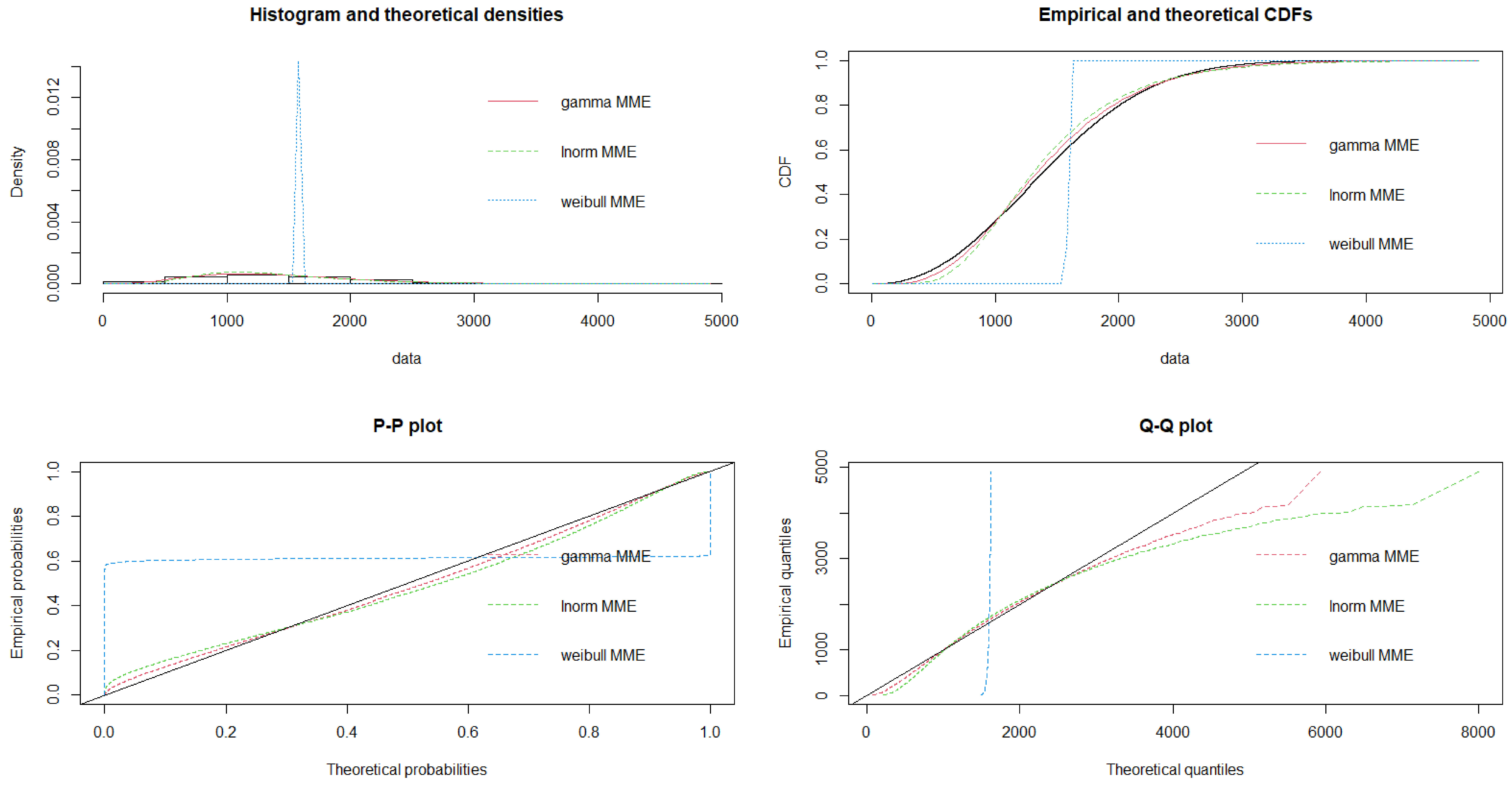
Cullen and Frey graph



We will be trying Gamma, Lognormal and Weibull

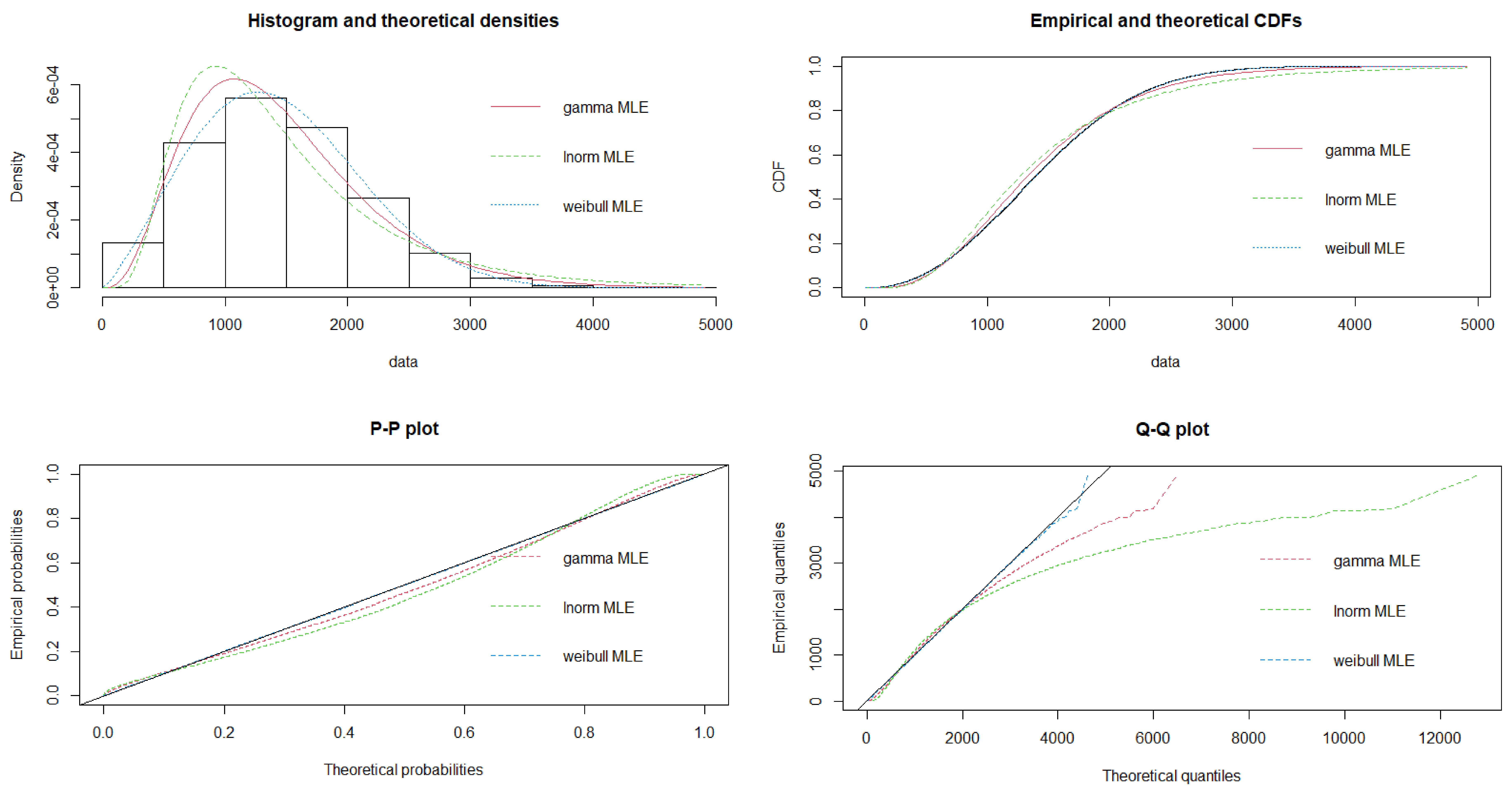
MME Estimation

Distribution	Parameter 1	Parameter 2	Loglikelihood
Gamma	Shape 4.657711953	Rate 0.003227521	-193741.2
Lognormal	Meanlog 7.1773177	Sdlog 0.4410161	-197121.3
Weibull	Shape 157.8534	Scale 1596.2348	-inf


Histogram and theoretical densities
Density
data
gamma MME
lnorm MME
weibull MME
Empirical and theoretical CDFs
CDF
data
gamma MME
lnorm MME
weibull MME
P-P plot
Empirical probabilities
Theoretical probabilities
gamma MME
lnorm MME
weibull MME
Q-Q plot
Empirical quantiles
Theoretical quantiles
gamma MME
lnorm MME
weibull MME

MLE Estimation

Distribution	Parameter 1	Parameter 2	Loglikelihood
Gamma	Shape 3.948323712	Rate 0.002736376	-193551.3
Lognormal	Meanlog 7.1426474	Sdlog 0.5636995	-195299.2
Weibull	Shape 2.286708	Scale 1628.721687	-192994.1


Histogram and theoretical densities
Density
data
gamma MLE
lnorm MLE
weibull MLE
Empirical and theoretical CDFs
CDF
data
P-P plot
Empirical probabilities
Theoretical probabilities
Q-Q plot
Empirical quantiles
Theoretical quantiles

Which distribution?

```
## Goodness-of-fit statistics
##                              gamma MLE    lnorm MLE weibull MLE    gamma MME
## Kolmogorov-Smirnov statistic  0.03943959   0.07520581 0.004271387   0.03328785
## Cramer-von Mises statistic   13.68946898  50.44231939 0.039274503  11.52012644
## Anderson-Darling statistic   81.68017684 309.01849236 0.218687383 102.63166339
##                                lnorm MME  weibull MME
## Kolmogorov-Smirnov statistic   0.06013151    0.5808157
## Cramer-von Mises statistic    41.12659777 2252.3165817
## Anderson-Darling statistic   473.20968512          Inf
##
## Goodness-of-fit criteria
##                                gamma MLE lnorm MLE weibull MLE gamma MME
## Akaike's Information Criterion  387106.6  390602.5    385992.3  387486.4
## Bayesian Information Criterion  387122.8  390618.7    386008.5  387502.6
##                                lnorm MME weibull MME
## Akaike's Information Criterion  394246.5         Inf
## Bayesian Information Criterion  394262.7         Inf
```

```
claimgof$adtest
```

```
##      gamma MLE      lnorm MLE    weibull MLE      gamma MME      lnorm MME
##     "rejected" "not computed" "not rejected" "not computed" "not computed"
##    weibull MME
## "not computed"
```

```
claimgof$kstest
```

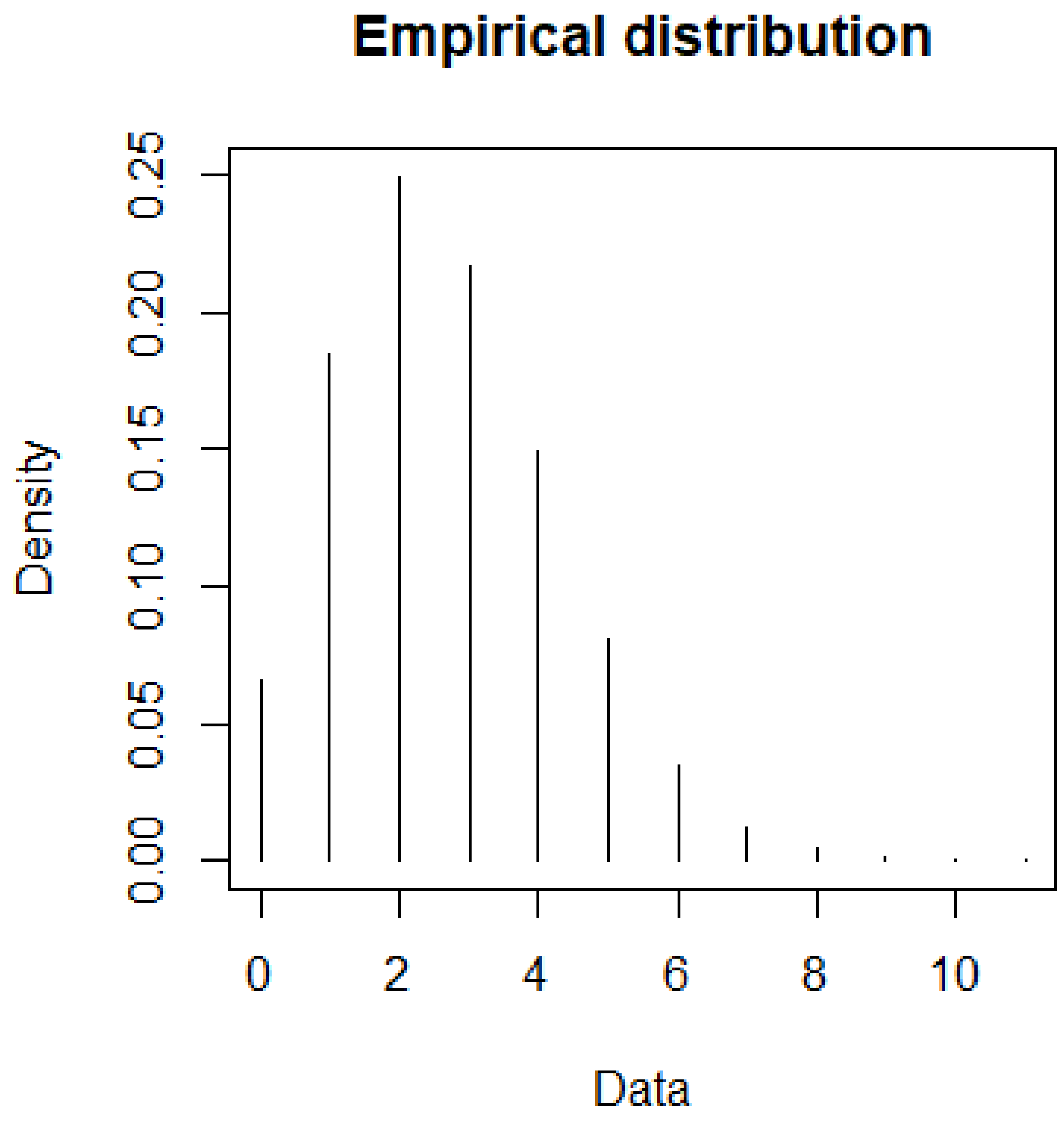
```
##      gamma MLE      lnorm MLE    weibull MLE      gamma MME      lnorm MME
##     "rejected"     "rejected" "not rejected"     "rejected"     "rejected"
##    weibull MME
##     "rejected"
```

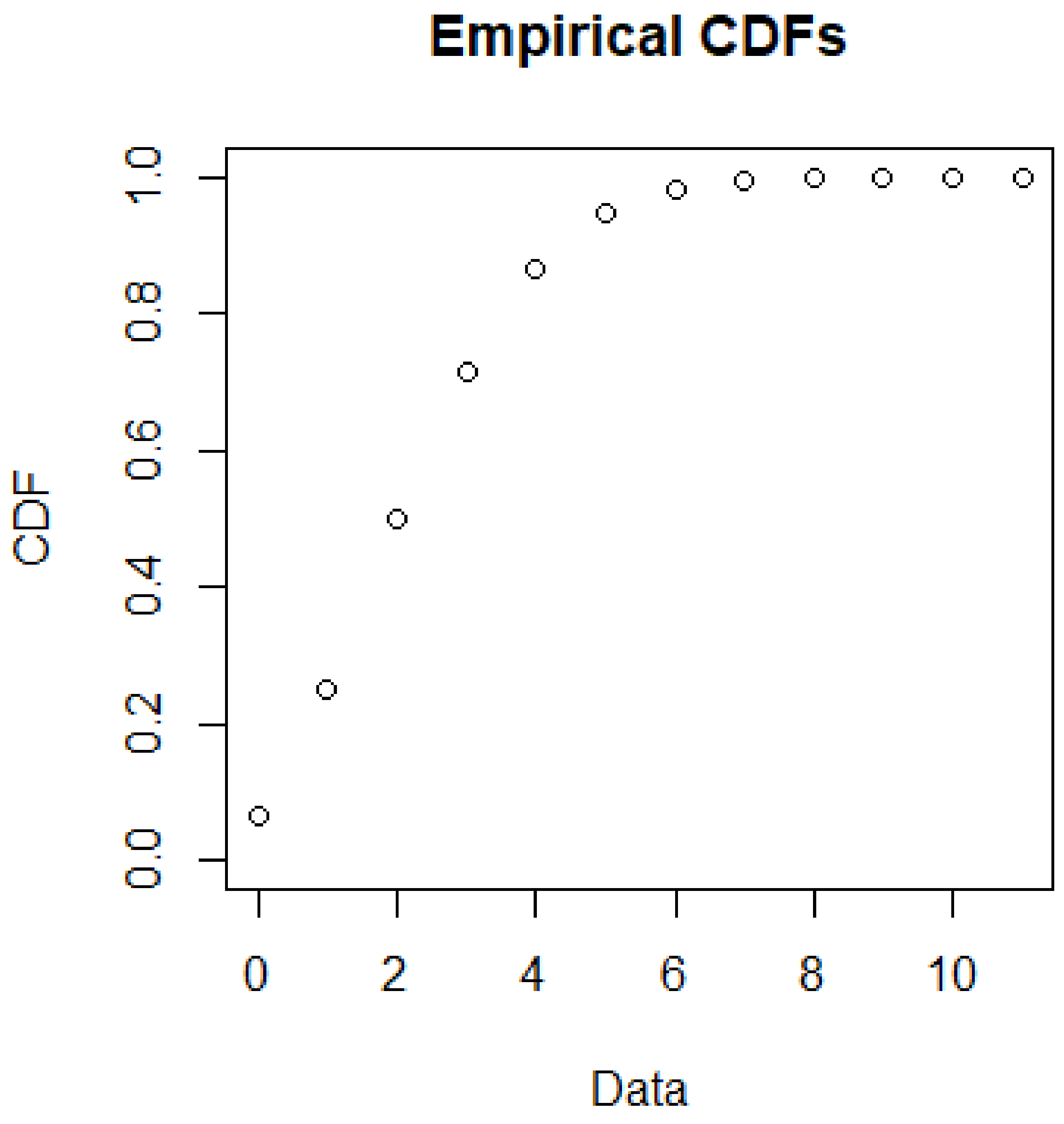
Conclusion:

We will be using Weibull MLE to model claim severity

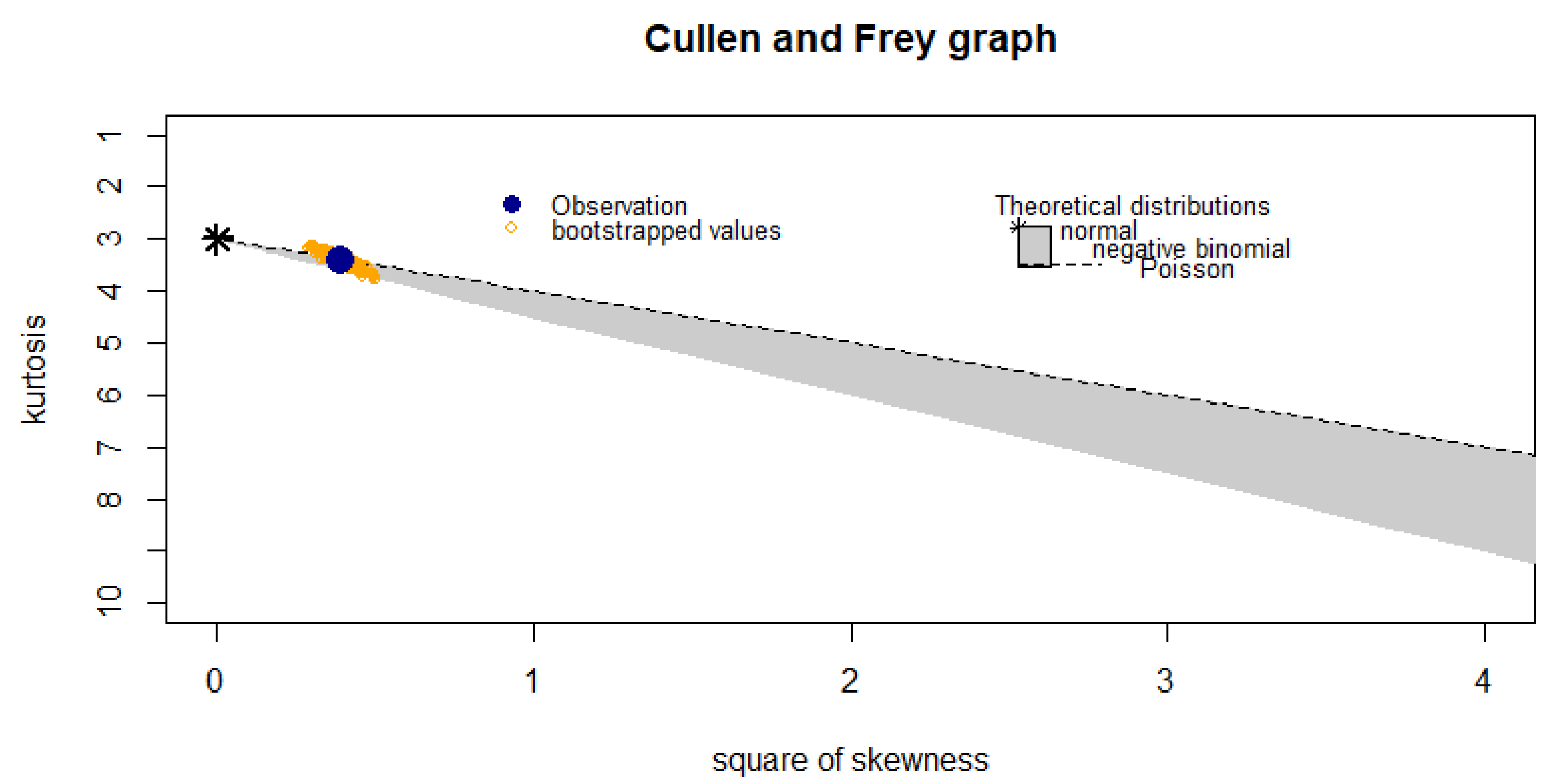
Claim Severity ~ Weibull(Shape = 2.286708, Scale = 1628.721687)

Claims Frequency Analysis





Cullen and Frey graph

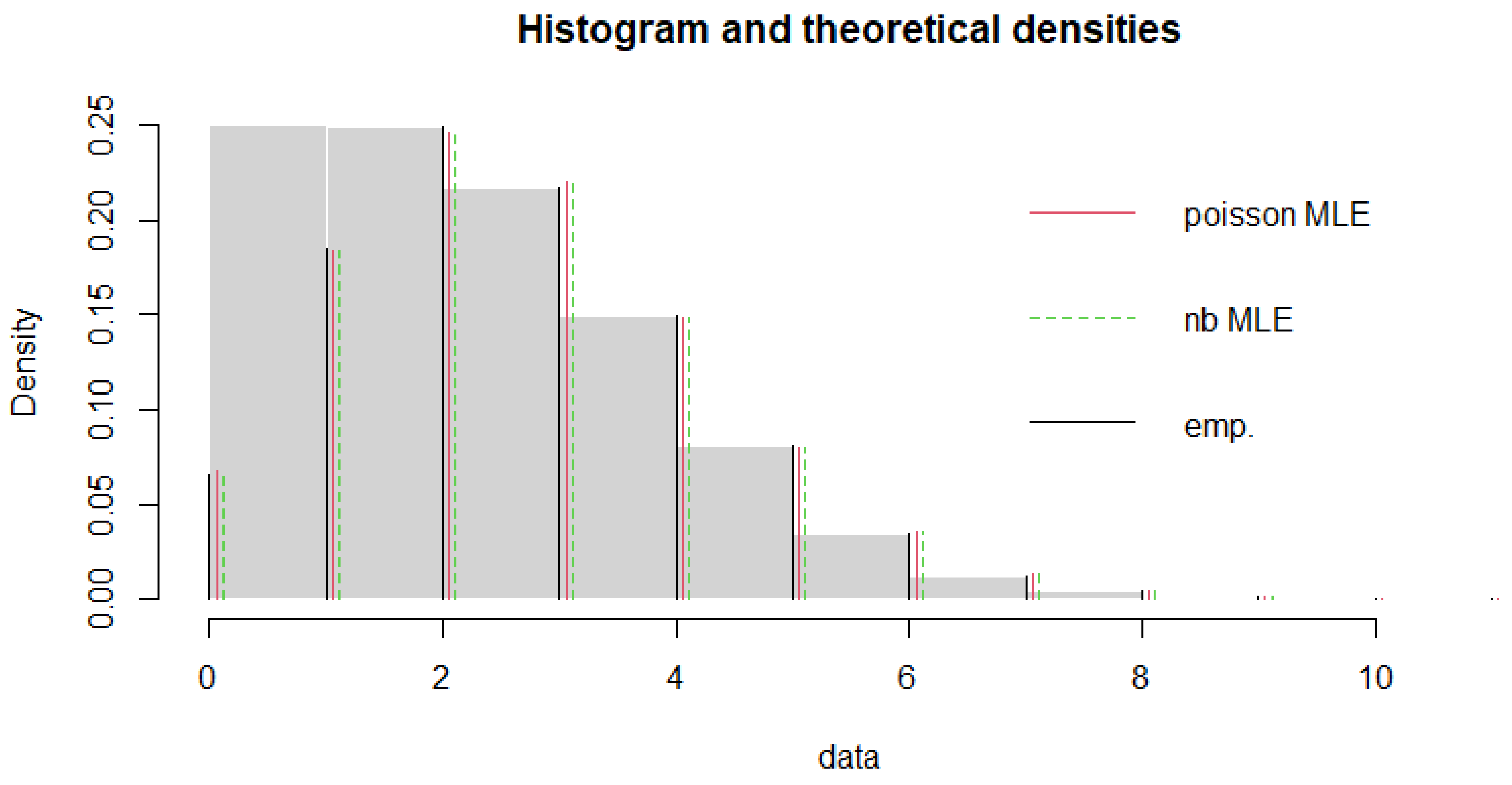


We will be trying Negative Binomial and Poisson

Parameter Estimation

Distribution	Parameter 1	Parameter 2	Loglikelihood
Negative Binomial MME	Size NaN	Mu 2.686889	NaN
Negative Binomial MLE	Size 38080.966228	Mu 2.687037	-16980.86
Poisson MLE = MME	Lambda 2.686889	-	-16980.86

Parameter Estimation



Which Distribution?

```
## Goodness-of-fit criteria
##                                poisson MLE   nb MLE
## Akaike's Information Criterion    33963.71 33965.72
## Bayesian Information Criterion    33970.83 33979.95
```

Conclusion:

We will be using Poisson MLE to model claims frequency

Claims Frequency ~ Poisson(Lambda = 2.686889)

Reinsurance Analysis

Gross Profit = Revenue - Insurer Loss - Reinsurance Price

Will be the same for all

0 for no reinsurance,
P_1 for first type of reinsurance
P_2 for second type of reinsurance

E[Claim] *
Number of Policies *
E[Claim Frequency per Policy]

E[Min(0, 1.5E[Claim amount])] *
Number of Policies *
E[Claim Frequency per Policy]

E[Min(0, 1.5E[Aggregate Claim Amount per Policy])] *
Number of Policies

No Reinsurance

Insurer Loss = E[Claim] * Number of Policies * E[Claim Frequency per Policy]

= E[Weibull(Shape = 2.286708, Scale = 1628.721687)] * 9,099 * E[Pois(Lambda = 2.686889)]

= $35, 273, 993

Type 1 Reinsurance

Insurer Loss = E[Min(0, 1.5E[Claim amount])] * Number of Policies * E[Claim Frequency per Policy]

E[Min(0, 1.5E[Claim amount])] = integrate(1 - F(x)) from 0 to 1.5E[Claim amount]

= $1,383.099

Where F(x) is the CDF of Weibull(Shape = 2.286708, Scale = 1628.721687)

Insurer Loss = $1,383.099 * 9,099 * 2.686889

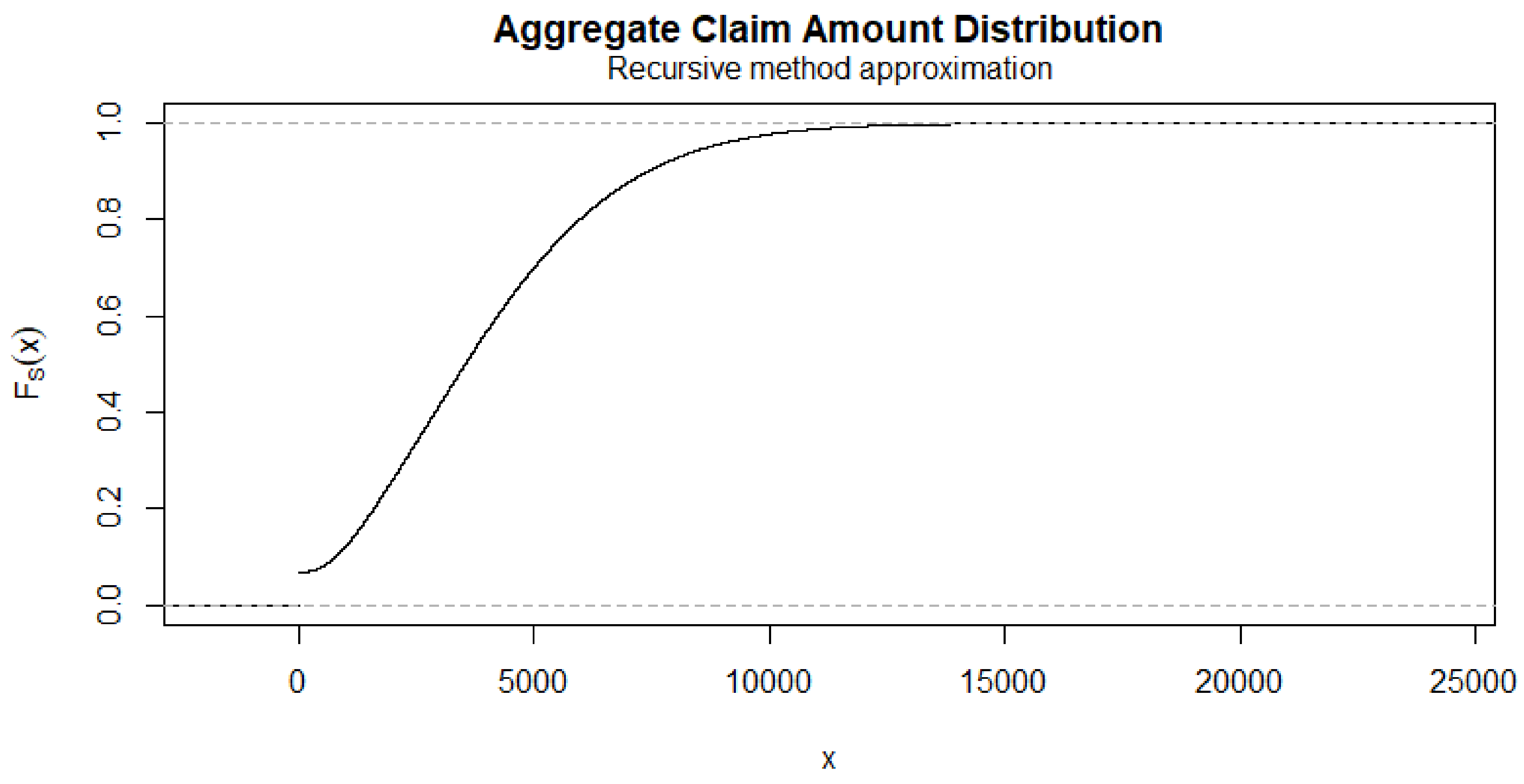
= $33, 814, 016

Type 2 Reinsurance

Insurer Loss =E[Min(0, 1.5E[Aggregate Claim Amount per Policy])] * Number of Policies

We need to first create a new CDF for Aggregate Claim Amount per Policy

By using Panjer's recursion algorithm, with unbiased discretisation of the Weibull distribution for model severity and Poisson for model frequency:



Type 2 Reinsurance

E[Min(0, 1.5E[Aggregate Claim Amount per Policy])] = integrate(1 - F(x)) from 0 to 1.5E[Aggregate Claim Amount per Policy] = $3,463.925

Where F(x) is the CDF of Aggregate Claim Amount per Policy

Insurer Loss = $3,463.925 * 9,099

= $31, 518, 257

Conclusion

Outflow from no reinsurance = $35, 273, 993

Outflow from Type 1 reinsurance = $33, 814, 016 + $P_1

Outflow from Type 2 reinsurance = $31, 518, 257 + $P_2

Where $P_i is the total price of Type i reinsurance cover

Recommendation:

Choose the type which evaluates to the lowest outflow to maximise gross profit

Limitations and Moving Forward

- Too few data
- Overfitting of the data
- Assumptions may not reflect company's assumptions

Appendix

General Insurance

````
```{r}
# Chunk setting up the workspace
set.seed(30007)
setwd("C:\\Users\\edmun\\Desktop\\Code Folder\\ACTL30007\\Assignment")
library("readxl")
library("writexl")
library("actuar")
library("fitdistrplus")
library("extRemes")
library("evir")
raw_data <- read_excel("1073769.xlsx")
```
````

````
```{r}
sum(is.na(raw_data)) # Check how many policy id have zero claim
cleaned_data <- raw_data
cleaned_data$claim[is.na(cleaned_data$claim)] <- 0
write_xlsx(data.frame(raw_data, cleaned_data),"C:\\Users\\edmun\\Desktop\\Code Folder\\ACTL30007\\Assignment\\output.xlsx")
attach(cleaned_data)
```
````

There are 593 policies with no claim.

````
```{r}
plotdist(claim[claim>0], hist = TRUE, demp = TRUE)
min(claim[claim > 0])
max(claim[claim > 0])
format(actuar::emm(claim[claim>0], order = 1:3), scientific = FALSE)
sd(claim[claim>0])/mean(claim[claim>0])
```

Data looks nice.

```{r}
descdist(claim[claim>0], boot = 1000)
```

We will check Gamma, Lognormal and Weibull
```{r}
claim.nz.lif <- cumsum(sort(claim[claim>0]))/sum(claim[claim>0])
plot(1:length(claim[claim>0])/length(claim[claim>0]), claim.nz.lif,
  xlab = "number of claims (in 100%)", ylab = "empirical loss size index function")
abline(h = 0.2, v = 0.8)
```

Not heavy-tailed distribution
````

```{r}
mef <- function(x, u) {
  mefvector <- c()
  for (i in u) {
    mefvector <- c(mefvector, sum(pmax(sort(x) - i, 0))/length(x[x >
      i]))
  }
  return(mefvector)
}

mef(claim, c(0, 100, 1000, 2000, 4000, 4200, 4300))
length(claim[claim>4000])
length(claim[claim>4200])
mrlplot(claim)
```

The mean excess function tells us the average added value loss Y would take above a threshold. Decreasing mean excess function indicates a light-tailed distribution.

One exception in my data is a high-values claim at the tail which leads to an increase in mean excess function. However, there are only 5 claims above 4000 and 1 claim above 4200 out of 25041 claims so it can be an outlier or a very rare event.

Indicates that log-normal would not be a good fit as it is a heavy-tailed distribution. Weibull Shape parameter (tao in lectures) would not be between 0 and 1 or else it will be heavy-tailed. Gamma is light-tailed

~~~
```{r}
emplot(claim[claim  > 0], alog = "xy", labels = TRUE)
```
~~~

The log-log is concave which supports that distribution is not heavy-tailed. 1-F(x) rapidly diminishes to 0 at the tail.

~~~
```{r}
fit.gamma.mme <- fitdist(claim[claim > 0], "gamma", method = "mme", order = 1:2)
fit.gamma.mme$estimate
fit.gamma.mme$loglik

fit.lnorm.mme <- fitdist(claim[claim > 0], "lnorm", method = "mme", order = 1:2)
fit.lnorm.mme$estimate
fit.lnorm.mme$loglik

memp <- function(x, order) mean(x^order)
fit.weibull.mme <- fitdist(claim[claim > 0], "weibull", method = "mme", memp = memp, order = 1:2)
fit.weibull.mme$estimate
fit.weibull.mme$loglik
```
~~~

````
```{r}
plot.legend <- c("gamma MME", "lnorm MME", "weibull MME")
fitdistrplus::denscomp(list(fit.gamma.mme, fit.lnorm.mme, fit.weibull.mme),
  legendtext = plot.legend, fitlwd = 1)
fitdistrplus::cdfcomp(list(fit.gamma.mme, fit.lnorm.mme, fit.weibull.mme),
  legendtext = plot.legend, fitlwd = 1, datapch = 10)
fitdistrplus::ppcomp(list(fit.gamma.mme, fit.lnorm.mme, fit.weibull.mme),
  legendtext = plot.legend, fitpch = 20)
fitdistrplus::qqcomp(list(fit.gamma.mme, fit.lnorm.mme, fit.weibull.mme),
  legendtext = plot.legend, fitpch = 20)
```
````

````
```{r}
fit.gamma.mle <- fitdist(claim[claim > 0], "gamma", method = "mle")
fit.gamma.mle$estimate
fit.gamma.mle$loglik

fit.lnorm.mle <- fitdist(claim[claim > 0], "lnorm", method = "mle")
fit.lnorm.mle$estimate
fit.lnorm.mle$loglik

fit.weibull.mle <- fitdist(claim[claim > 0], "weibull", method = "mle")
fit.weibull.mle$estimate
fit.weibull.mle$loglik

summary(fit.gamma.mle)
summary(fit.lnorm.mle)
summary(fit.weibull.mle)
```
````

````
```{r}
plot.legend <- c("gamma MLE", "lnorm MLE", "weibull MLE")
fitdistrplus::denscomp(list(fit.gamma.mle, fit.lnorm.mle, fit.weibull.mle),
  legendtext = plot.legend, fitlwd = 1)
fitdistrplus::cdfcomp(list(fit.gamma.mle, fit.lnorm.mle, fit.weibull.mle),
  legendtext = plot.legend, fitlwd = 1, datapch = 10)
fitdistrplus::ppcomp(list(fit.gamma.mle, fit.lnorm.mle, fit.weibull.mle),
  legendtext = plot.legend, fitpch = 20)
fitdistrplus::qqcomp(list(fit.gamma.mle, fit.lnorm.mle, fit.weibull.mle),
  legendtext = plot.legend, fitpch = 20)
```
````

````
```{r}
gofstat(list(fit.gamma.mle, fit.lnorm.mle, fit.weibull.mle, fit.gamma.mme, fit.lnorm.mme, fit.weibull.mme),
  fitnames = c("gamma MLE", "lnorm MLE", "weibull MLE", "gamma MME", "lnorm MME", "weibull MME"))
claimgof <- gofstat(list(fit.gamma.mle, fit.lnorm.mle, fit.weibull.mle, fit.gamma.mme, fit.lnorm.mme, fit.weibull.mme), fitnames = c("gamma MLE", "lnorm MLE", "weibull MLE", "gamma MME", "lnorm MME", "weibull MME"), chisqbreaks = c(0, 1000, 2000, 3000, 4000, 5000))
claimgof$chisqpvalue
claimgof$adtest
claimgof$kstest
claimgof$chisqtable
```
````

We will be using Weibull distribution with shape = 2.286708 and scale = 1628.721687

```r
weibull.shape = as.numeric(fit.weibull.mle$estimate["shape"])
weibull.shape
weibull.scale = as.numeric(fit.weibull.mle$estimate["scale"])
weibull.scale
mean(claim[claim > 0])
weibull.scale * gamma(1+1/weibull.shape)
var(claim[claim > 0])
(weibull.scale ^ 2) * (gamma(1 + 2/weibull.shape) - (gamma(1 + 1/weibull.shape)) ^ 2)
```

Checking fitness looks good!

Finding the distribution of the claims frequency:

```r
claimfreq <- as.data.frame(table(cleaned_data$polind[claim > 0]))
claimfreq <- rbind(claimfreq, data.frame(Var1 = rep("manual", 593), Freq = rep(0, 593))) # adding 593 polind which had no claim
plotdist(claimfreq$Freq, hist = TRUE, demp = TRUE, discrete = TRUE)
descdist(claimfreq$Freq, boot = 1000, discrete = TRUE)
```

We will try negative binomial and poisson

```{r}
fit.nb.mme <- fitdist(claimfreq$Freq, "nbinom", method = "mme")
fit.nb.mme$estimate
fit.nb.mme$loglik

fit.pois.mle <- fitdist(claimfreq$Freq, "pois", method = "mle")
fit.pois.mle$estimate
fit.pois.mle$loglik

fit.nb.mle <- fitdist(claimfreq$Freq, "nbinom", method = "mle")
fit.nb.mle$estimate
fit.nb.mle$loglik
```

MLE and MME for Poisson distribution is the same.
Since MME for negative binomial is incomplete, we will be comparing MLE for Poisson and Negative Binomial.

```{r}
plot.legend <- c("poisson MLE", "nb MLE")
fitdistrplus::denscomp(list(fit.pois.mle, fit.nb.mle),
  legendtext = plot.legend, fitlwd = 1)
fitdistrplus::cdfcomp(list(fit.pois.mle, fit.nb.mle),
  legendtext = plot.legend, fitlwd = 1, datapch = 10)
fitdistrplus::ppcomp(list(fit.pois.mle, fit.nb.mle),
  legendtext = plot.legend, fitpch = 20)
fitdistrplus::qqcomp(list(fit.pois.mle, fit.nb.mle),
  legendtext = plot.legend, fitpch = 20)

gofstat(list(fit.pois.mle, fit.nb.mle),
  fitnames = c("poisson MLE", "nb MLE"))
claimgof <- gofstat(list(fit.pois.mle, fit.nb.mle), fitnames = c("poisson MLE", "nb MLE"), chisqbreaks = c(2, 4, 6, 8, 10))
```

We will use Poisson distribution with lambda = 2.686889 due to lower AIC and BIC scores.

```{r}
pois.lambda <- as.numeric(fit.pois.mle$estimate)
pois.lambda
mean(claimfreq$Freq)
```

Good Fitting!

Without reinsurance,
Profit = Revenue - Total Loss

```{r}
nopolind <- 9099 # total number of polind
# Loss = Mean loss per claim * # policies * Expected # claim per policy
Total_loss_insurer <- weibull.scale * gamma(1+1/weibull.shape) * nopolind * pois.lambda
Total_loss_insurer
```

Reinsurance Conclusions:
Type 1: Profit = Revenue - min(0, 1.5E[Y]) * claim - Total Price_1
Where Y is Loss
Type 2: Profit = Revenue - min(0, 1.5E[S]) * policy - Total Price_2
Where S = Aggregate per Policyholder
Assumption: Policyholder can only hold 1 Policy - Disadvantage!
We can cross out Revenue since both cases have the same number.

Calculating min(0, 1.5E[Y]) - Type 1 Reinsurance

```{r}
upper_bound1 = 1.5 * weibull.scale * gamma(1+1/weibull.shape)

sweibull <- function(x) {1 - pweibull(x, shape = weibull.shape, scale = weibull.scale)}

minfuncvalue <- integrate(sweibull, 0, upper_bound1)$value
minfuncvalue
total_loss_insurer_1 <- minfuncvalue * nopolind * pois.lambda
total_loss_insurer_1
```

Calculating min(0, 1.5E[S]) - Type 2 Reinsurance
Since per Policyholder Severity ~ Weibull and Frequency ~ Pois, We can model S ~ CompoundPois(lambda * v, Y ~ Weibull) this is per policy

```{r}
expected_S = pois.lambda * weibull.scale * gamma(1+1/weibull.shape)
upper_bound2 = 1.5 * expected_S

stp = 1
final = 7000
# First, we discretise the proposed Weibull distribution
weibull.discr.unbia <- discretise(pweibull(x, shape = weibull.shape, scale = weibull.scale), from = 0,
  to = final, step = stp, method = "unbiased", lev = levweibull(x,
    shape = weibull.shape, scale = weibull.scale))
weibull.discr.unbia.cdf <- cumsum(weibull.discr.unbia)
curve(pweibull(x, shape = weibull.shape, scale = weibull.scale), from = 0, to = final)
lines((0:(final/stp)) * stp, weibull.discr.unbia.cdf, type = "s",
  pch = 20, col = "green")

S.unbia.cdf <- aggregateDist(method = "recursive", model.freq = "poisson",
  lambda = pois.lambda, model.sev = weibull.discr.unbia, x.scale = stp,
  maxit = 100000000)
plot(S.unbia.cdf, pch = 20, col = "black", cex = 0.5)

S.unbia.sdf <- function(x){1 - S.unbia.cdf(x)}

minfuncvalue2 <- integrate(S.unbia.sdf, 0, upper_bound2)$value
minfuncvalue2
total_loss_insurer_2 <- minfuncvalue2 * nopolind
total_loss_insurer_2
```

In conclusion, insurer loss under reinsurance 1 is more than when under reinsurance 2. Without reinsurance, insurer loss is the greatest.