

Lista 3º Unidade

GEISIANNY KELLY, JESSICA EVELIN, LUCAS ALLAN, MICAELA CARNEIRO

Março de 2020

Base

Pacotes:

```
#Pacotes necessários  
library(readr)  
library(fitdistrplus)  
library(MASS)  
library(actuar)
```

Dados:

```
# Importação dos dados:  
grupo4 <- read_csv("D:/Google Drive/6 - R/5 - Estágio 2 - Prova/Dados/grupo  
4.csv", col_types = cols(n = col_number(), x = col_number()))  
  
# separando apenas o x do data frame  
sin <- grupo4$x #sin = sinistros(x)
```

Respostas - Questão 1

1. Considere os dados referente ao seu nome contidos na pasta. Por exemplo, Grupo 1 utilizará o arquivo denominado "grupo1.csv".

a) Elabora uma tabela com as estatísticas descritivas da variável "x" que se refere aos valores dos sinistros.

```
# estatísticas descritivas  
summary(sin)
```

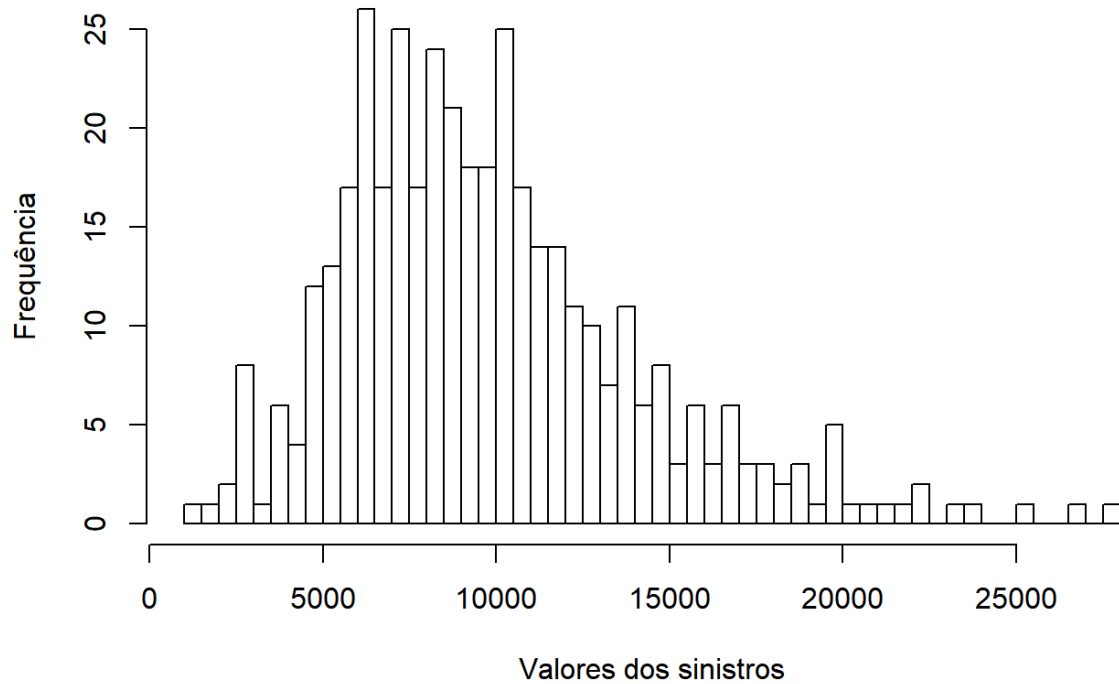
##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
##	1258	6769	9137	9857	11976	27769

Comentários - 1.a):

b) Plote um histograma para a variável valor de 1 sinistro ('x').

```
#histograma  
hist(sin, main = "Histograma dos sinistros", xlab = "Valores dos sinistros",  
ylab = "Frequência", breaks = 90)
```

Histograma dos sinistros



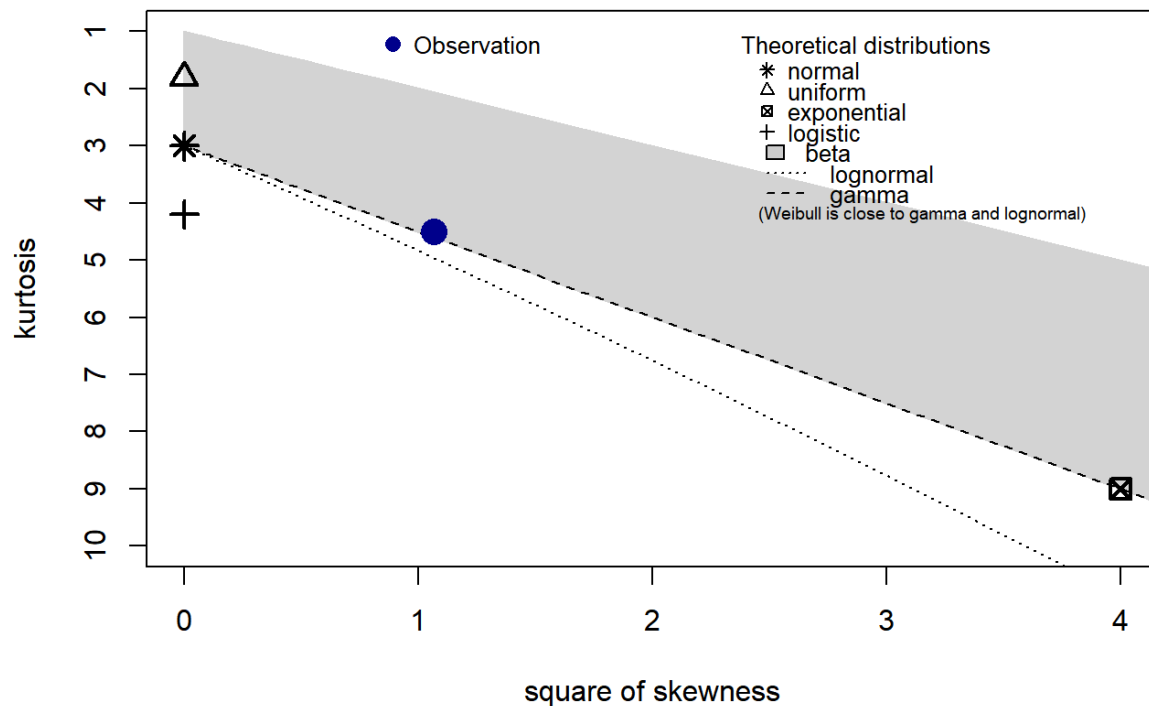
Comentários - 1.b):

.

c) Estime os parâmetros pelo método da máxima verossimilhança para as distribuições: "normal", "gamma", "lognormal", "weibull", "pareto".

```
# ajustes da distribuição:  
descdist(sin)
```

Cullen and Frey graph



```
## summary statistics
## -----
## min: 1257.953    max: 27768.73
## median: 9137.406
## mean: 9856.514
## estimated sd: 4358.057
## estimated skewness: 1.033744
## estimated kurtosis: 4.504158
```

```
# Estimando os parâmetros para normal,gamma,lognormal,weibull e pareto:
# Gamma e Pareto precisam do pacote actuar
fit.normal <- fitdist(sin, "norm")
fit.gamma <- fitdist(sin, distr = "gamma", method = "mle", lower = c(0, 0),
  start = list(scale = 1, shape = 500))
fit.lognormal <- fitdist(sin, "lnorm")
fit.weibull <- fitdist(sin, "weibull")
fit.pareto <- fitdist(sin, "pareto", start = list(shape = 1, scale = 500))

# Parâmetros de cada distribuição:
summary(fit.normal)
```

```
## Fitting of the distribution ' norm ' by maximum likelihood
## Parameters :
##      estimate Std. Error
## mean 9856.514    217.4645
## sd   4352.606    153.5644
## Loglikelihood: -3918.987    AIC: 7841.975    BIC: 7849.958
```

```
## Correlation matrix:
##      mean sd
## mean    1  0
## sd      0  1
```

```
summary(fit.gamma)
```

```
## Fitting of the distribution ' gamma ' by maximum likelihood
## Parameters :
##      estimate Std. Error
## scale 1874.456150      NA
## shape  5.258522      NA
## Loglikelihood: -3887.371  AIC:  7778.742  BIC:  7786.725
## Correlation matrix:
## [1] NA
```

```
summary(fit.lognormal)
```

```
## Fitting of the distribution ' lnorm ' by maximum likelihood
## Parameters :
##      estimate Std. Error
## meanlog 9.0978026 0.02287142
## sdlog  0.4574284 0.01617219
## Loglikelihood: -3893.842  AIC:  7791.685  BIC:  7799.668
## Correlation matrix:
##      meanlog sdlog
## meanlog      1    0
## sdlog        0    1
```

```
summary(fit.weibull)
```

```
## Fitting of the distribution ' weibull ' by maximum likelihood
## Parameters :
##      estimate Std. Error
## shape  2.38808  0.08743466
## scale 11133.85816 247.55457727
## Loglikelihood: -3901.033  AIC:  7806.067  BIC:  7814.049
## Correlation matrix:
##      shape      scale
## shape 1.0000000 0.3293913
## scale 0.3293913 1.0000000
```

```
summary(fit.pareto)
```

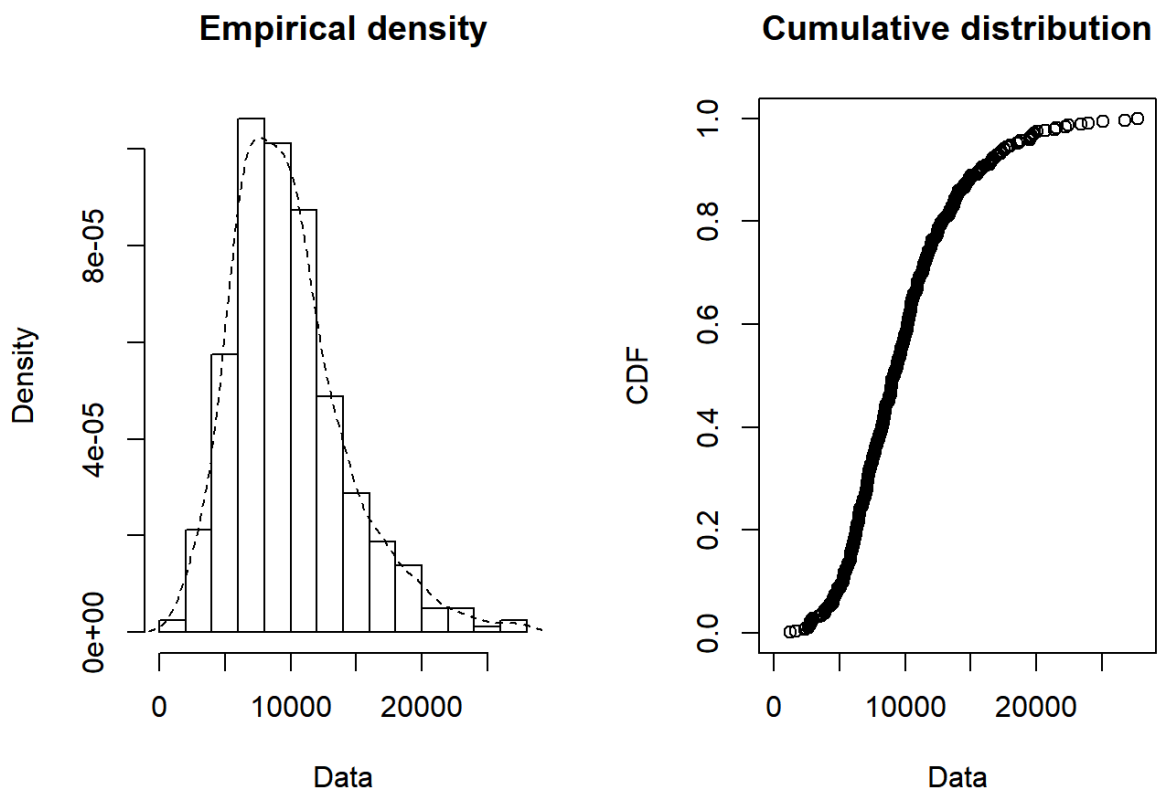
```
## Fitting of the distribution ' pareto ' by maximum likelihood
## Parameters :
##      estimate Std. Error
## shape  4928152      NA
```

```
## scale 48554100412          NA
## Loglikelihood: -4078.355    AIC: 8160.71    BIC: 8168.693
## Correlation matrix:
## [1] NA
```

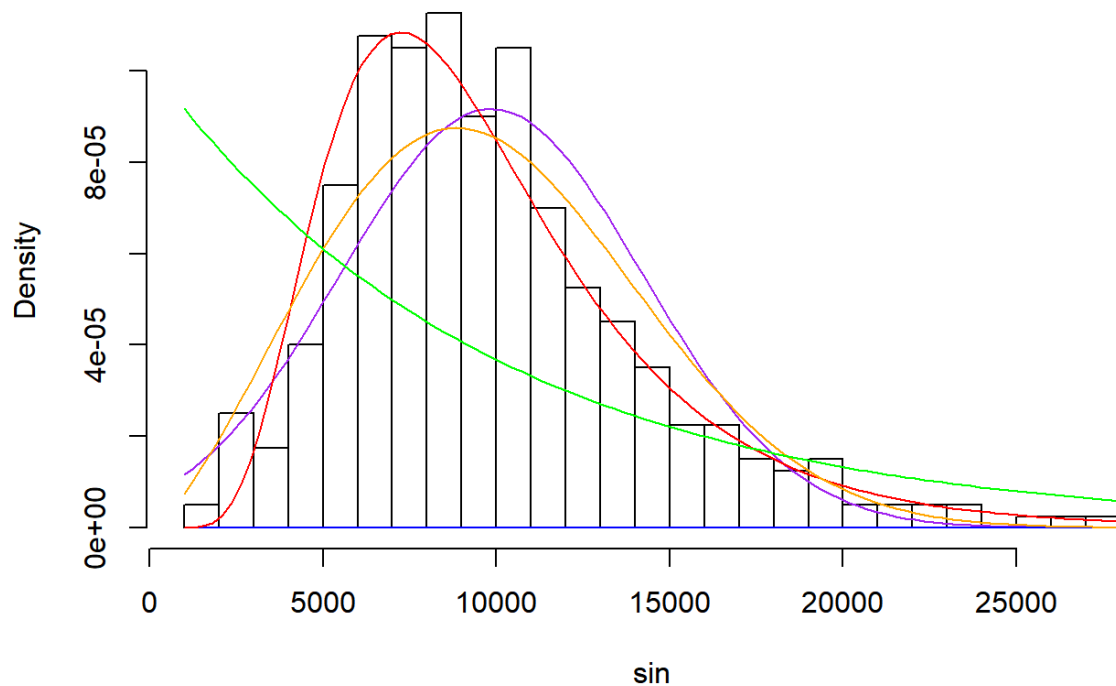
Comentários - 1.c):

d) Apresente os gráficos dos modelos probabilísticos estimados em relação à distribuição empírica.

```
# dados empiricos
plotdist(sin, histo = TRUE, demp = TRUE)
```



```
# dados estimados
hist(sin, pch=20, breaks = 20, prob = TRUE, main = "")
curve(dnorm(x, fit.normal$estimate[1], fit.normal$estimate[2]), add = TRUE,
      col = "purple")
curve(dgamma(x, fit.gamma$estimate[1], fit.gamma$estimate[2]), add = TRUE, col = "blue")
curve(dlnorm(x, fit.lognormal$estimate[1], fit.lognormal$estimate[2]), add = TRUE, col = "red")
curve(dweibull(x, fit.weibull$estimate[1], fit.weibull$estimate[2]), add = TRUE, col = "orange")
curve(dpareto(x, fit.pareto$estimate[1], fit.pareto$estimate[2]), add = TRUE, col = "green")
```



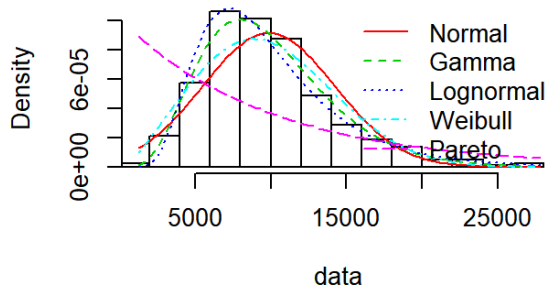
Comentários - 1.d):

.

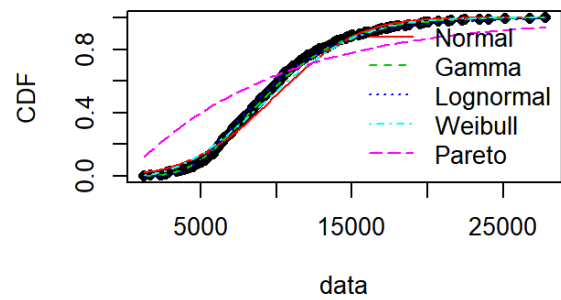
e) Indique qual o modelo que melhor se ajusta aos dados dos sinistros e explique o porquê.

```
#analisar qual distribuição se parece com o histograma dos dados:
par(mfrow=c(2,2))
plot.legend <- c("Normal", "Gamma", "Lognormal", "Weibull", "Pareto")
denscomp(list(fit.normal, fit.gamma, fit.lognormal, fit.weibull, fit.paret
o), legendtext = plot.legend)
cdfcomp (list(fit.normal, fit.gamma, fit.lognormal, fit.weibull, fit.paret
o), legendtext = plot.legend)
qqcomp (list(fit.normal, fit.gamma, fit.lognormal, fit.weibull, fit.paret
o), legendtext = plot.legend)
ppcomp (list(fit.normal, fit.gamma, fit.lognormal, fit.weibull, fit.paret
o), legendtext = plot.legend)
```

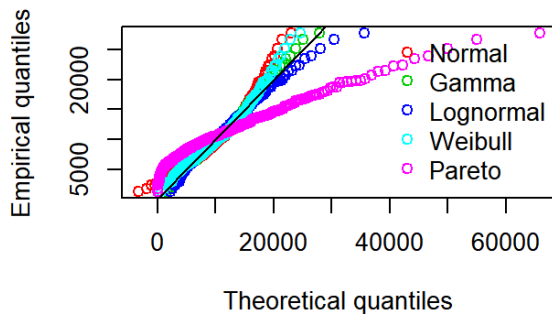
Histogram and theoretical densities



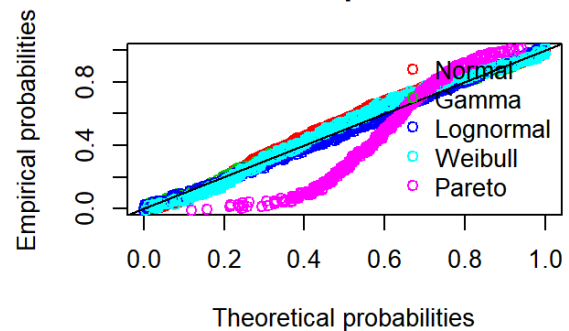
Empirical and theoretical CDFs



Q-Q plot



P-P plot



#Analisar com base o Kolmogorov-Smirnov statistic e Akaike's Information Criterion:

```
gofstat(list(fit.normal, fit.gamma, fit.lognormal, fit.weibull, fit.pareto),
fitnames = c("Normal", "Gamma", "Lognormal", "Weibull", "Pareto"))
```

Goodness-of-fit statistics

	Normal	Gamma	Lognormal	Weibull
## Kolmogorov-Smirnov statistic	0.0873606	0.03181546	0.0353167	0.06589116
## Cramer-von Mises statistic	0.8849215	0.07356169	0.1319229	0.49467775
## Anderson-Darling statistic	5.4351576	0.51865813	0.9779977	3.18313740

Pareto

	Pareto
## Kolmogorov-Smirnov statistic	0.3180183
## Cramer-von Mises statistic	12.3073783
## Anderson-Darling statistic	62.1302840

##

Goodness-of-fit criteria

	Normal	Gamma	Lognormal	Weibull	Pareto
## Akaike's Information Criterion	7841.975	7778.742	7791.685	7806.067	8160.710
## Bayesian Information Criterion	7849.958	7786.725	7799.668	7814.049	8168.693

Comentários - 1.e):

.

Respostas - Questão 2

2. Calcule o Prêmio Puro pelo Princípio do Valor Esperado para a Carteira pelo Modelo de Risco Coletivo. Utilize o carregamento de segurança $\theta = 10\%$. Utilizará uma base de dados que corresponde aos sinistros de uma carteira de seguro de responsabilidade civil. A distribuição referente à frequência de sinistros por apólice é uma Poisson com parâmetro $\lambda = 400$. A quantidade de segurados na carteira é 10000.

$$PrêmioPuro = E(S) * (1 + \theta)$$

```
# Dados
theta = 0.1
lambda = 400
n = 10000

# Severidade "X"
esperanca.x <- mean(sin)
variancia.x <- var(sin)
desvio.x <- sd(sin)
coefvar.x <- desvio.x / esperanca.x

# Frequência "N" ~ Poisson (lambda = 4 )
lambda = 400
esperanca.n = variancia.n = lambda

# Sinistro Agregado
esperanca.s = esperanca.n * esperanca.x
variancia.s = esperanca.n * variancia.x + (esperanca.x) ^ 2 * variancia.n

# Prêmio Puro
premiopuro = esperanca.s * (1 + theta)
premiopuro
```

```
## [1] 4336866
```

Comentários - 2):

Respostas - Questão 3

3. Calcule o Prêmio Comercial com carregamento para despesas $\alpha = 35\%$:

$$PrêmioComercial = \frac{PrêmioPuro}{1 - \alpha}$$

```
alpha = 0.35
premiocomercial = premiopuro / (1 - alpha)
premiocomercial
```

```
## [1] 6672102
```

Comentários - 3):