

# MAE399 - Lista 7

André Vinícius - nUSP = 10737290

June 23, 2019

Esta é a sétima e última lista de exercícios do curso “Análise de Dados e Simulação”. Desta vez, faremos simulações do fluxo de caixa de uma seguradora de carros hipotética. Temos definido que teremos clientes que pagarão um valor mensal pelo seguro, cujo valor será sorteado segundo uma distribuição uniforme com valores entre \$100 e \$200 unidades monetárias. Para cada um desses clientes, temos a probabilidade de sinistro a uma taxa de 1 para cada 24 meses, cujo custo será sorteado segundo uma distribuição exponencial de média igual a \$1.200 unidades monetárias. Por fim, temos que durante o funcionamento da nossa seguradora as vendas de apólices obedeçam a uma distribuição exponencial de taxa mensal igual à 1.

A partir das simulações, identificaremos valores para percentis extremos dos dados em determinados pontos do tempo e apresentaremos esses dados numa tabela. Os valores identificados com o saldo do nosso fluxo de caixa serão plotados num gráfico, juntamente com a curva da sua esperança e 5 simulações dos seus valores. Este processo será repetido com modificações: inicialmente, será determinado um valor de franquia mínimo para que a seguradora arque com o custo do sinistro; depois, faremos uma alteração na distribuição do valor das mensalidades. Ao final, teremos 4 gráficos e 4 tabelas para serem feitas comparações e obtidas conclusões.

Definidos os objetivos deste relatório e os parâmetros para nossas simulações, iniciaremos codificando a função que simula cada um dos clientes. A descrição dos scripts será feita pelos comentários no corpo das funções.

```
cliente <- function(contratacao, # data de início do contrato do cliente
                    duracao, # duração do contrato
                    franquia = 0 # franquia do sinistro
){

  #esta função simula dados de um cliente de uma seguradora de carros,
  #a partir de uma data inicial de contratação até seu prazo final de duração
  #os dados gerados se referem a recebimentos mensais e sinistros esperados
  #as distribuições referentes às simulações estão descritas no relatório

  #gerando dados de datas e respectivos recebimentos
  mensalidade <- runif(1, 100, 200) #sorteia valor mensal
  datarecebs <- contratacao + 0:duracao #define datas dos recebimentos
  recebs <- c(rep(mensalidade, duracao), 0) #cria vetor dos recebimentos

  #gerando dados de datas e respectivos custos de sinistros
  datasinis <- contratacao #registra inicio do contrato
  while(tail(datasinis, 1) < (contratacao + duracao)) #sorteia datas dos sinistros
    datasinis <- c(datasinis, (tail(datasinis, 1) + rexp(1, 1/24)))
  siniscusto <- 0 #inicializa vetor de custo de cada sinistro
  for(s in 0:length(datasinis)){ #sorteia custos de sinistro
    custo <- rexp(1, 1/1200)
    if(custo < franquia) custo <- 0
    siniscusto <- c(siniscusto, custo)
  }

  #organizando os dados em tabelas

  #gera tabela com datas e recebimentos respectivos
```

```

recebs <- cbind(datarecebs, recebs)
#gera tabela com datas e custos respectivos
#com o cuidado de nao simular sinistros fora do prazo do contrato
if(tail(datasinis, 1) > tail(datarecebs, 1)){
  sinis <- cbind(datasinis[1:(length(datasinis) - 1)],
                 siniscusto[1:(length(datasinis) - 1)])
}else{
  sinis <- cbind(datasinis, siniscusto)
}

#salva as tabelas numa lista do cliente e retorna os dados da simulacao
return(list(entradas = recebs, saidas = sinis))
}

```

Conforme explicitado, o código acima retorna os dados de entradas e saídas de valores em caixa, mas sem somá-los. A soma será feita mais adiante, quando tivermos todos os clientes já simulados.

Em seguida, simulamos o funcionamento da seguradora por um período de tempo determinado, gerando os dados dos clientes que contratam seus serviços.

```

simula <- function(tempo, franquia = 0){

  #gera datas de novas entradas de clientes
  contratos <- numeric() # inicializa o vetor de novos contratos
  novocliente <- rexp(1) # gera a primeira data de contratação
  while(novocliente < tempo){
    # gera novas datas até atingir o tempo total de atividade simulado
    contratos <- c(contratos, novocliente)
    novocliente <- novocliente + rexp(1)
  }

  N <- length(contratos) # obtém o tamanho do vetor de datas de contratação
  producao <- vector("list", N) # inicializa o vetor de simulações de clientes
  for(c in 1:N){
    #gera uma simulação de cliente para cada data de início de contrato
    #a duração do contrato é delimitada pelo tempo total de atividade simulado
    producao[[c]] <- cliente(contratos[c],
                             ceiling(tempo - contratos[c]),
                             franquia)
  }
  return(producao)
}

```

Gerados os dados dos clientes, podemos fazer as somas que definem o fluxo de caixa. Acumulamos recebimentos, despesas, e podemos prever o saldo do caixa.

```

organiza <- function(producao){

  #organizando dados de recebimentos para serem plotados
  recebimentos <- matrix(nrow = 0, ncol = 2)
  for(c in 1:length(producao)){
    recebimentos <- rbind(recebimentos, producao[[c]]$entradas)
  }
}

```

```

recebimentos <- recebimentos[order(recebimentos[,1]),]
for(i in 2:nrow(recebimentos)){
  recebimentos[i, 2] <- recebimentos[i-1, 2] + recebimentos[i, 2]
}

#organizando dados de despesas para serem plotados
despesas <- matrix(nrow = 0, ncol = 2)
for(c in 1:length(producao)){
  despesas <- rbind(despesas, producao[[c]]$saidas)
}
despesas <- despesas[order(despesas[,1]),]
for(i in 2:nrow(despesas)){
  despesas[i, 2] <- despesas[i-1, 2] + despesas[i, 2]
}

#organizando dados do saldo para ser plotado
saldo <- matrix(nrow = 0, ncol = 2)
for(c in 1:length(producao)){
  saldo <- rbind(saldo, producao[[c]]$entradas)
  saidastmp <- producao[[c]]$saidas
  saidastmp[,2] <- saidastmp[,2]*-1
  saldo <- rbind(saldo, saidastmp)
}
saldo <- saldo[order(saldo[,1]),]
for(i in 2:nrow(saldo)){
  saldo[i, 2] <- saldo[i-1, 2] + saldo[i, 2]
}

#retorna os dados em uma lista
return(list(recebimentos = recebimentos,
            despesas = despesas,
            saldo = saldo))
}

```

Agora que temos como gerar os dados do fluxo de caixa, podemos gerar simulações do funcionamento da seguradora.

```

geradados <- function(nsimul, tempo, franquia = 0){

  #gera 10 x "nsimul" simulações da atividade da seguradora
  #para o período de tempo fixado pelo parâmetro "tempo"
  #e coleta os dados para os pontos t = 50, 100, 150, 200, 250,... , 500.

  #inicializando as matrizes de dados das simulações
  Rsimulado <- matrix(nrow = 0, ncol = 10)
  Dsimulado <- matrix(nrow = 0, ncol = 10)
  Ssimulado <- matrix(nrow = 0, ncol = 10)

  #executa o laço 10 vezes
  for(v in 1:10){

    meusdados <- vector("list", nsimul) # inicializa a lista de simulações
    #print(paste("Simulação ", v, " de 10", sep = ""))
    #pb <- txtProgressBar(min = 1, max = nsimul, style = 3) #barra de progresso
  }
}

```

```

for(s in 1:nsimul){

  meusdados[[s]] <- organiza(simula(tempo, franquia)) #gera os dados de cada simulação
  #setTxtProgressBar(pb, s) #atualiza barra de progresso

}

#close(pb) #fecha barra de progresso

#inicializa matrizes de dados da atual bateria de simulações
Rmatriz <- matrix(nrow = nsimul, ncol = 0)
Dmatriz <- matrix(nrow = nsimul, ncol = 0)
Smatriz <- matrix(nrow = nsimul, ncol = 0)

for(t in seq(50, 500, 50)){

  #coleta os dados de "recebimentos" para o t atual
  R <- numeric(nsimul)
  for(s in 1:nsimul){

    dados <- meusdados[[s]]$recebimentos
    z <- length(dados[dados[,1] <= t,1])
    R[s] <- dados[z,2]

  }
  Rmatriz <- cbind(Rmatriz, R) #adiciona os dados à respectiva coluna

  #coleta os dados de "despesas" para o t atual
  D <- numeric(nsimul)
  for(s in 1:nsimul){

    dados <- meusdados[[s]]$despesas
    z <- length(dados[dados[,1] <= t,1])
    D[s] <- dados[z,2]

  }
  Dmatriz <- cbind(Dmatriz, D) #adiciona os dados à respectiva coluna

  #coleta os dados de "saldo" para o "t" atual
  S <- numeric(nsimul)
  for(s in 1:nsimul){

    dados <- meusdados[[s]]$saldo
    z <- length(dados[dados[,1] <= t,1])
    S[s] <- dados[z,2]

  }
  Smatriz <- cbind(Smatriz, S) #adiciona os dados à respectiva coluna

}

# adiciona os dados da atual bateria de simulações às demais simulações
Rsimulado <- rbind(Rsimulado, Rmatriz)

```

```

Dsimulado <- rbind(Dsimulado, Dmatriz)
Ssimulado <- rbind(Ssimulado, Smatriz)

}

# coloca nomes nas colunas das matrizes
colnames(Rsimulado) <- paste0("R", seq(50, 500, 50))
colnames(Dsimulado) <- paste0("D", seq(50, 500, 50))
colnames(Ssimulado) <- paste0("S", seq(50, 500, 50))

# retorna lista com as matrizes
return(list(recebimentos = Rsimulado,
            despesas = Dsimulado,
            saldos = Ssimulado))

}

```

De agora em diante, geraremos simulações e visualizações dos dados. Conforme previsto, a primeira simulação gerará dados de contratos sem franquia. Depois da tabela, apresentaremos o primeiro gráfico, que relaciona os dados de saldo da tabela, sua esperança, e algumas simulações. Quanto à curva da Esperança, como sabemos que esta é linear, temos  $E[R(t) - D(t)] = E[R(t)] - E[D(t)]$ , com as esperanças para um cliente isolado sendo  $E[R(t)] = \lambda_R t E(Y_R)$  e  $E[D(t)] = \lambda_D t E(Y_D)$ . Logo,

$$E[R(t) - D(t)] = \lambda_R t E(Y_R) - \lambda_D t E(Y_D) = t(\lambda_R E(Y_R) - \lambda_D E(Y_D))$$

Para uma sequência onde são adicionados clientes a uma taxa de 1 por mês, a esperança de quantidade de clientes para o mês  $t$  será também  $t$ . Assim, a esperança do valor acumulado para esta sequência será:

$$\sum_{i=1}^t i(\lambda_R E(Y_R) - \lambda_D E(Y_D)) = \frac{t(t+1)}{2}(\lambda_R E(Y_R) - \lambda_D E(Y_D))$$

Repetiremos a mesma curva para a Esperança quando apresentarmos os dados das simulações com franquia, para que possa haver comparação.

```

dados <- geradados(100, 500)

Rtabela <- matrix(nrow = 0, ncol = 2)
Dtabela <- matrix(nrow = 0, ncol = 2)
Stabela <- matrix(nrow = 0, ncol = 2)
for(t in seq(50, 500, 50)){

  Rtabela <- rbind(Rtabela, quantile(dados$recebimentos[,paste0("R", t)],
                                     probs = c(0.005, 0.995)))
  Dtabela <- rbind(Dtabela, quantile(dados$despesas[,paste0("D", t)],
                                     probs = c(0.005, 0.995)))
  Stabela <- rbind(Stabela, quantile(dados$saldos[,paste0("S", t)],
                                     probs = c(0.005, 0.995)))

}

tabela <- data.frame(R = Rtabela, D = Dtabela, S = Stabela)
rownames(tabela) <- paste0("t", seq(50, 500, 50))
colnames(tabela) <- c("R(t) 5%", "R(t) 95%",
                     "D(t) 5%", "D(t) 95%",

```

```
"R(t)-D(t) 5%", "R(t)-D(t) 95%")
```

```
tabela
```

##	R(t) 5%	R(t) 95%	D(t) 5%	D(t) 95%	R(t)-D(t) 5%	R(t)-D(t) 95%
## t50	110899.7	278337.7	28194.91	107449.4	63813.82	200435.2
## t100	542303.3	1012214.0	165102.72	348601.3	337273.92	694067.0
## t150	1301521.0	2135739.6	410915.15	749696.4	837598.60	1450734.4
## t200	2403470.4	3688085.3	773337.08	1271526.8	1575905.34	2487700.3
## t250	3824154.2	5672993.9	1269412.83	1906186.3	2522574.01	3737316.0
## t300	5657670.2	8050291.7	1861669.71	2676772.2	3749054.76	5344962.5
## t350	7862279.9	10777277.4	2584169.57	3617638.9	5190108.98	7213884.4
## t400	10356321.7	13936675.2	3379524.03	4738594.5	6881250.28	9335975.3
## t450	13276299.2	17483971.0	4299963.40	5854149.2	8784433.95	11763741.3
## t500	16492406.4	21357041.4	5347912.19	7175447.4	10931229.00	14352132.6

```
E <- function(x) (x*(x+1)/2)*(1*(200+100)/2 - (1/24)*1200)
```

```
curve(E, 0, 500,
      lwd = 2,
      ylim = c(0, 1.5e+07),
      main = "R(t) - D(t): Esperança, 5 Simulações e Percentis",
      sub = "(sem franquia)",
      xlab = "Tempo(meses)",
      ylab = "",
      las = 1)
```

```
cores <- rainbow(5)
```

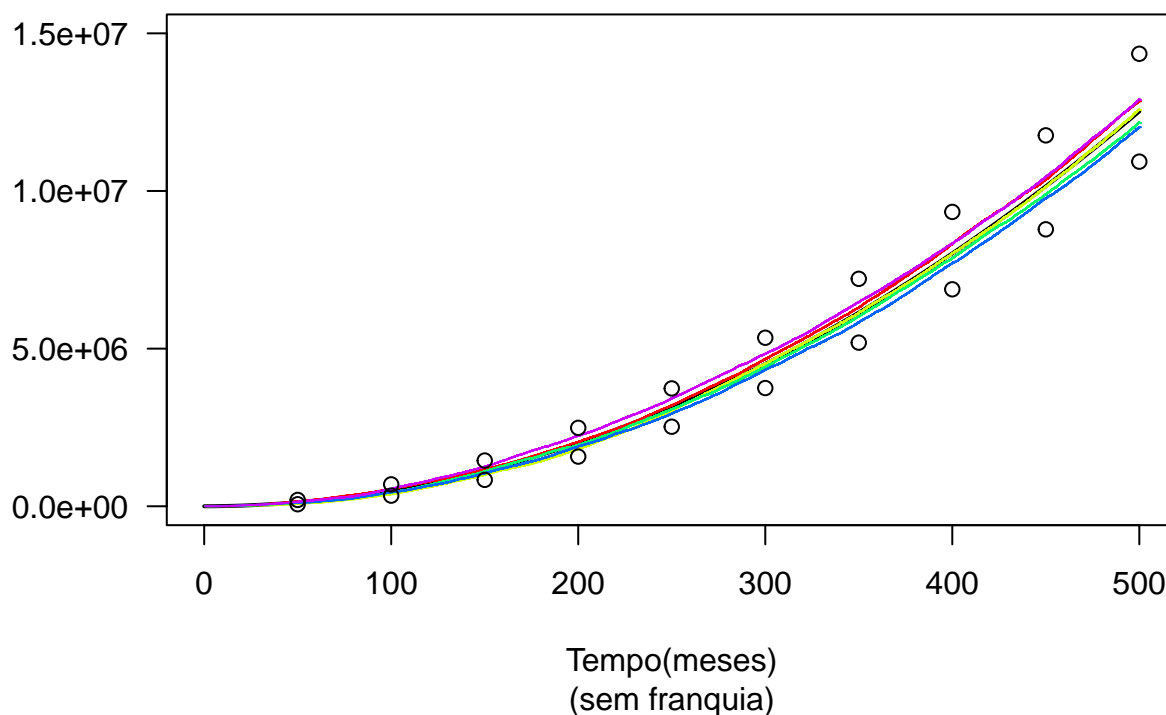
```
for(s in 1:5){
```

```
  simulacao <- organiza(simula(500))
  lines(simulacao$saldo[,1],
        simulacao$saldo[,2],
        col = cores[s])
}
```

```
for(i in seq(50, 500, 50)){
```

```
  #plotando os pontos da tabela
  points(c(i, i), tabela[paste0("t", i),5:6])
}
```

## R(t) – D(t): Esperança, 5 Simulações e Percentis



Abaixo, simulamos delimitando os gastos de sinistro pela franquia.

```
dados <- geradados(100, 500, 600)

Rtabela <- matrix(nrow = 0, ncol = 2)
Dtabela <- matrix(nrow = 0, ncol = 2)
Stabela <- matrix(nrow = 0, ncol = 2)
for(t in seq(50, 500, 50)){

  Rtabela <- rbind(Rtabela, quantile(dados$recebimentos[,paste0("R", t)],
                                     probs = c(0.005, 0.995)))
  Dtabela <- rbind(Dtabela, quantile(dados$despesas[,paste0("D", t)],
                                     probs = c(0.005, 0.995)))
  Stabela <- rbind(Stabela, quantile(dados$saldos[,paste0("S", t)],
                                     probs = c(0.005, 0.995)))

}

tabela <- data.frame(R = Rtabela, D = Dtabela, S = Stabela)
rownames(tabela) <- paste0("t", seq(50, 500, 50))
colnames(tabela) <- c("R(t) 5%", "R(t) 95%",
                     "D(t) 5%", "D(t) 95%",
                     "R(t)-D(t) 5%", "R(t)-D(t) 95%")

tabela
```

	R(t) 5%	R(t) 95%	D(t) 5%	D(t) 95%	R(t)-D(t) 5%	R(t)-D(t) 95%
## t50	115420.3	270454.9	24774.74	102553.3	73518.26	194825.9
## t100	520757.4	976037.6	144535.26	316386.9	360631.38	697821.9

```
## t150 1287266.6 2096197.2 362760.07 669563.9 883982.40 1497919.7
## t200 2410444.2 3650037.6 689312.10 1149288.2 1648854.44 2568682.5
## t250 3870744.4 5591282.8 1131566.70 1733153.7 2653524.02 3928105.8
## t300 5642369.0 7943208.4 1628513.38 2467228.0 3874214.71 5577858.2
## t350 7807452.0 10666839.1 2303420.75 3285936.8 5415877.80 7469919.7
## t400 10390322.8 13790625.2 3057915.14 4220373.4 7190455.92 9651612.7
## t450 13201973.2 17389752.2 3899922.42 5286535.3 9204498.61 12151889.8
## t500 16556557.0 21189042.6 4883759.73 6447975.7 11460020.83 14890544.6
```

```
E <- function(x) (x*(x+1)/2)*(1*(200+100)/2 - (1/24)*1200)

curve(E, 0, 500,
      lwd = 2,
      ylim = c(0, 1.5e+07),
      main = "R(t) - D(t): Esperança, 5 Simulações e Percentis",
      sub = "(franquia = $600)",
      xlab = "Tempo(meses)",
      ylab = "",
      las = 1)

cores <- rainbow(5)

for(s in 1:5){

  simulacao <- organiza(simula(500, 600))
  lines(simulacao$saldo[,1],
        simulacao$saldo[,2],
        col = cores[s])

}

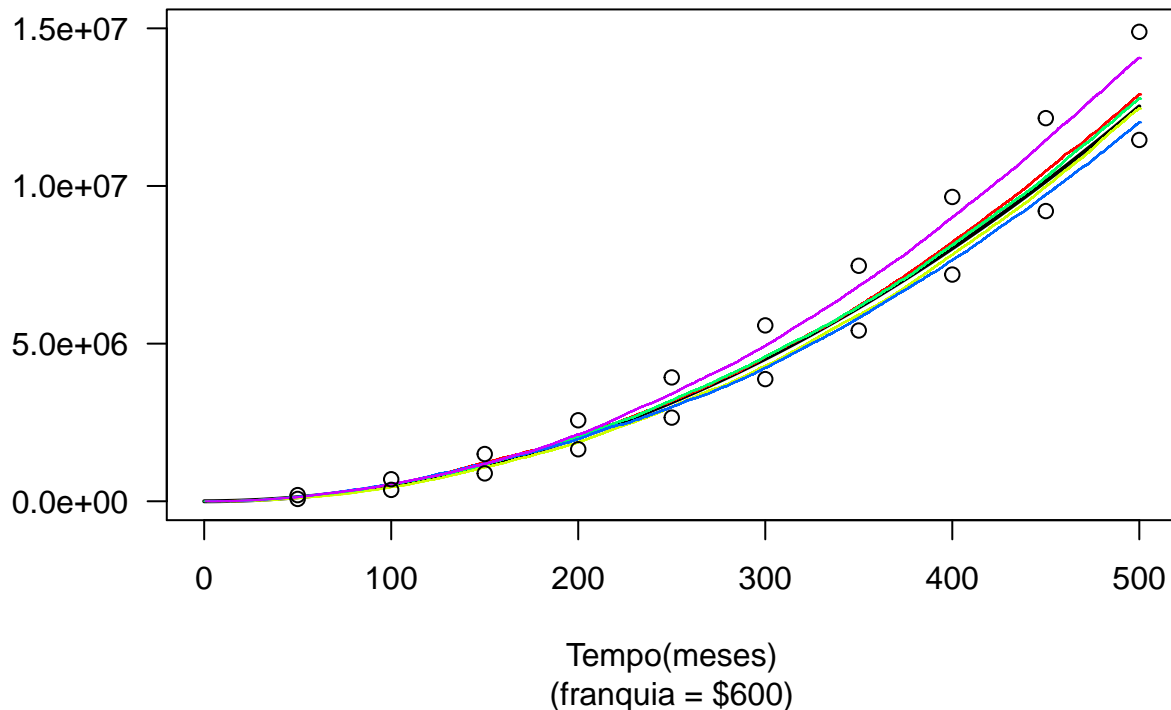
for(i in seq(50, 500, 50)){

  #plotando os pontos da tabela
  points(c(i, i), tabela[paste0("t", i),5:6])

}
```



## R(t) – D(t): Esperança, 5 Simulações e Percentis



Repetiremos agora as simulações fazendo uma alteração na distribuição das mensalidades. Usaremos a distribuição exponencial, mas com a mesma média, para que possamos observar a diferença. A escolha da distribuição exponencial se deve ao fato de acreditarmos que essa será a verdadeira distribuição dos valores das apólices: poucas apólices terão valor alto, sendo a maioria de valores menores, com cauda longa. Faremos as simulações para os casos com e sem franquia. Deixamos observado também que a curva da esperança é a mesma, pois foi mantida a média das mensalidades. Antes das simulações, segue a função alterada.

```
cliente <- function(contratacao, # data de início do contrato do cliente
                    duracao, # duração do contrato
                    franquias = 0 # franquias do sinistro
                    ){

  #esta função simula dados de um cliente de uma seguradora de carros,
  #a partir de uma data inicial de contratação até seu prazo final de duração
  #os dados gerados se referem a recebimentos mensais e sinistros esperados
  #as distribuições referentes às simulações estão descritas no relatório

  #gerando dados de datas e respectivos recebimentos
  mensalidade <- rexp(1, 1/150) #sorteia valor mensal ***ALTERADO***
  datarecebs <- contratacao + 0:duracao #define datas dos recebimentos
  recebs <- c(rep(mensalidade, duracao), 0) #cria vetor dos recebimentos

  #gerando dados de datas e respectivos custos de sinistros
  datasinisis <- contratacao #registra inicio do contrato
  while(tail(datasinisis, 1) < (contratacao + duracao)) #sorteia datas dos sinistros
    datasinisis <- c(datasinisis, (tail(datasinisis, 1) + rexp(1, 1/24)))
  siniscusto <- 0 #inicializa vetor de custo de cada sinistro
  for(s in 0:length(datasinisis)){ #sorteia custos de sinistro
```

```

    custo <- rexp(1, 1/1200)
    if(custo < franquia) custo <- 0
    siniscusto <- c(siniscusto, custo)
  }

  #organizando os dados em tabelas

  #gera tabela com datas e recebimentos respectivos
  recebs <- cbind(datarecebs, recebs)
  #gera tabela com datas e custos respectivos
  #com o cuidado de nao simular sinistros fora do prazo do contrato
  if(tail(datasinis, 1) > tail(datarecebs, 1)){
    sinis <- cbind(datasinis[1:(length(datasinis) - 1)],
                  siniscusto[1:(length(datasinis) - 1)])
  }else{
    sinis <- cbind(datasinis, siniscusto)
  }

  #salva as tabelas numa lista do cliente e retorna os dados da simulacao
  return(list(entradas = recebs, saidas = sinis))
}

```

Seguem as tabelas e gráficos.

```

dados <- geradados(100, 500)

Rtabela <- matrix(nrow = 0, ncol = 2)
Dtabela <- matrix(nrow = 0, ncol = 2)
Stabela <- matrix(nrow = 0, ncol = 2)
for(t in seq(50, 500, 50)){

  Rtabela <- rbind(Rtabela, quantile(dados$recebimentos[,paste0("R", t)],
                                     probs = c(0.005, 0.995)))
  Dtabela <- rbind(Dtabela, quantile(dados$despesas[,paste0("D", t)],
                                     probs = c(0.005, 0.995)))
  Stabela <- rbind(Stabela, quantile(dados$saldos[,paste0("S", t)],
                                     probs = c(0.005, 0.995)))
}

tabela <- data.frame(R = Rtabela, D = Dtabela, S = Stabela)
rownames(tabela) <- paste0("t", seq(50, 500, 50))
colnames(tabela) <- c("R(t) 5%", "R(t) 95%",
                    "D(t) 5%", "D(t) 95%",
                    "R(t)-D(t) 5%", "R(t)-D(t) 95%")

tabela

```

##	R(t) 5%	R(t) 95%	D(t) 5%	D(t) 95%	R(t)-D(t) 5%	R(t)-D(t) 95%
## t50	92663.99	307541.6	27755.2	104442.7	41423.41	241495.6
## t100	456253.39	1078035.2	157263.0	357604.5	256664.71	804854.7
## t150	1142877.85	2290378.5	404622.1	743733.6	670578.73	1628223.0
## t200	2114062.84	3978103.6	758364.0	1265969.0	1250471.20	2782821.0
## t250	3468383.97	6006477.7	1216132.8	1905865.6	2087824.76	4236612.6

```
## t300 5205082.24 8443193.6 1796588.1 2711284.1 3145371.54 5906446.3
## t350 7218068.99 11259423.8 2512599.5 3636344.5 4432977.57 7956371.1
## t400 9531255.32 14563965.3 3318810.6 4704153.8 5897341.58 10210037.2
## t450 12189030.67 18246284.4 4241951.0 5965297.2 7600588.46 12722124.8
## t500 15238574.99 22307804.1 5266351.3 7268832.8 9545919.06 15476250.7
```

```
E <- function(x) (x*(x+1)/2)*(1*150 - (1/24)*1200)
```

```
curve(E, 0, 500,
      lwd = 2,
      ylim = c(0, 1.5e+07),
      main = "R(t) - D(t): Esperança, 5 Simulações e Percentis",
      sub = "(sem franquia, mensalidade com distribuição exponencial)",
      xlab = "Tempo(meses)",
      ylab = "",
      las = 1)
```

```
cores <- rainbow(5)
```

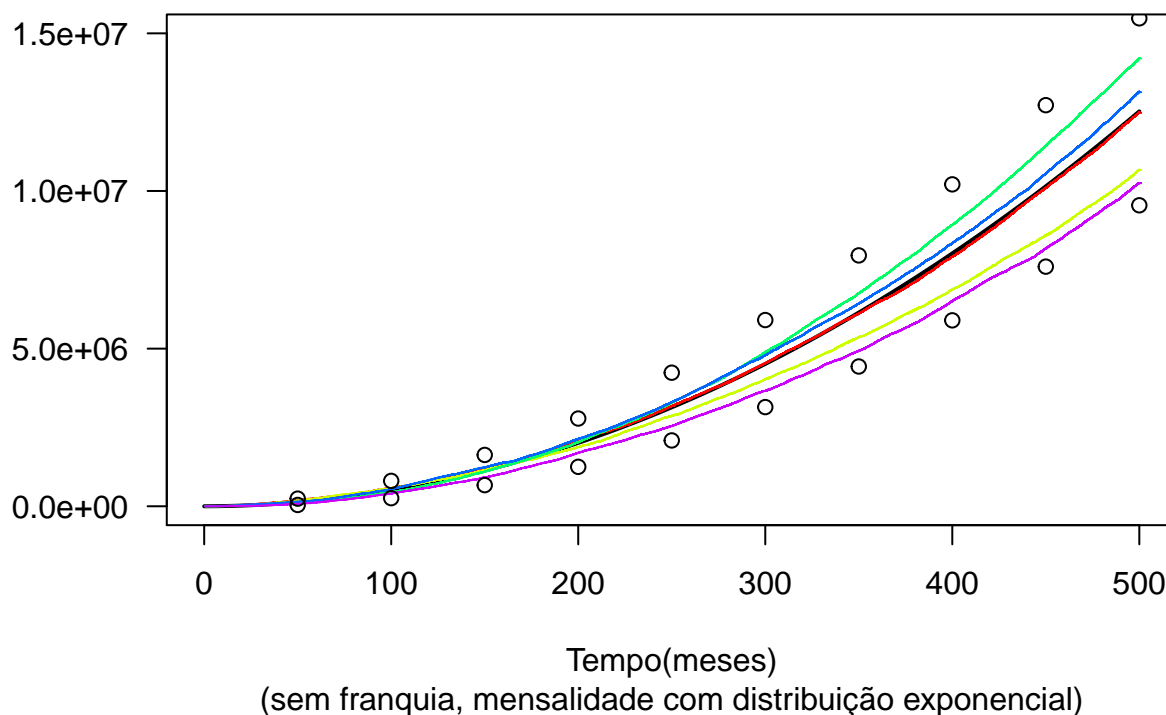
```
for(s in 1:5){
```

```
  simulacao <- organiza(simula(500, 600))
  lines(simulacao$saldo[,1],
        simulacao$saldo[,2],
        col = cores[s])
}
```

```
for(i in seq(50, 500, 50)){
```

```
  #plotando os pontos da tabela
  points(c(i, i), tabela[paste0("t", i),5:6])
}
```

## R(t) – D(t): Esperança, 5 Simulações e Percentis



```
dados <- geradados(100, 500, 600)

Rtabela <- matrix(nrow = 0, ncol = 2)
Dtabela <- matrix(nrow = 0, ncol = 2)
Stabela <- matrix(nrow = 0, ncol = 2)
for(t in seq(50, 500, 50)){

  Rtabela <- rbind(Rtabela, quantile(dados$recebimentos[,paste0("R", t)],
                                     probs = c(0.005, 0.995)))
  Dtabela <- rbind(Dtabela, quantile(dados$despesas[,paste0("D", t)],
                                     probs = c(0.005, 0.995)))
  Stabela <- rbind(Stabela, quantile(dados$saldos[,paste0("S", t)],
                                     probs = c(0.005, 0.995)))

}

tabela <- data.frame(R = Rtabela, D = Dtabela, S = Stabela)
rownames(tabela) <- paste0("t", seq(50, 500, 50))
colnames(tabela) <- c("R(t) 5%", "R(t) 95%",
                     "D(t) 5%", "D(t) 95%",
                     "R(t)-D(t) 5%", "R(t)-D(t) 95%")
```

tabela

##	R(t) 5%	R(t) 95%	D(t) 5%	D(t) 95%	R(t)-D(t) 5%
## t50	90953.06	326553.3	24690.69	96870.63	46367.84
## t100	475920.85	1084100.7	148965.03	322201.24	282810.81
## t150	1168114.72	2262043.1	370404.72	671828.75	708760.99
## t200	2247074.85	3897810.3	706314.85	1150837.74	1459641.58

```
## t250 3580782.51 5967818.2 1123745.75 1760305.61 2365465.79
## t300 5270139.51 8455010.9 1677150.03 2475390.37 3478958.81
## t350 7257974.78 11376311.7 2344448.80 3320844.42 4808902.32
## t400 9620707.05 14769292.9 3102350.72 4201276.14 6404684.76
## t450 12275332.70 18396756.6 3948996.03 5338574.91 8213517.76
## t500 15280977.49 22337198.5 4852814.83 6560237.71 10352668.00
##      R(t)-D(t) 95%
## t50      253224.6
## t100     850275.6
## t150    1706106.5
## t200    2918946.3
## t250    4369429.2
## t300    6119038.4
## t350    8269884.6
## t400   10585583.2
## t450   13234843.2
## t500   16252242.3
```

```
E <- function(x) (x*(x+1)/2)*(1*150 - (1/24)*1200)
```

```
curve(E, 0, 500,
      lwd = 2,
      ylim = c(0, 1.5e+07),
      main = "R(t) - D(t): Esperança, 5 Simulações e Percentis",
      sub = "(franquia = $600, mensalidade com distribuição exponencial)",
      xlab = "Tempo(meses)",
      ylab = "",
      las = 1)
```

```
cores <- rainbow(5)
```

```
for(s in 1:5){

  simulacao <- organiza(simula(500, 600))
  lines(simulacao$saldo[,1],
        simulacao$saldo[,2],
        col = cores[s])

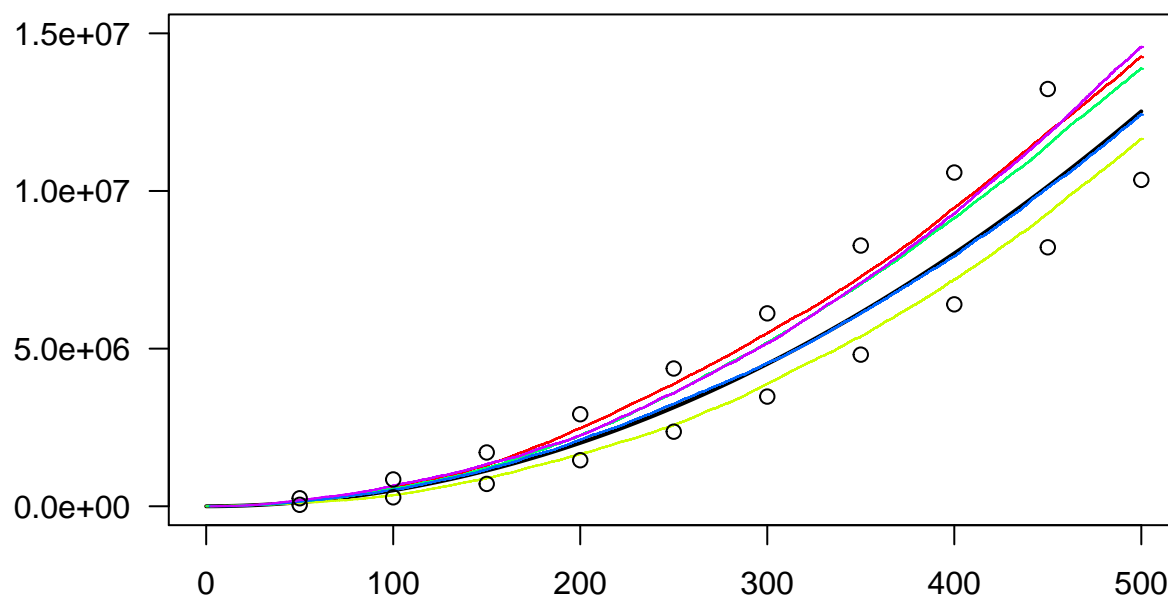
}

for(i in seq(50, 500, 50)){

  #plotando os pontos da tabela
  points(c(i, i), tabela[paste0("t", i),5:6])

}
```

### **R(t) – D(t): Esperança, 5 Simulações e Percentis**



Tempo(meses)

(franquia = \$600, mensalidade com distribuição exponencial)

Finalizando esta análise, concluímos que a hipótese que julgamos ser real, a da distribuição exponencial dos valores das mensalidades, acarreta em maior risco para o início do funcionamento da seguradora, uma vez que a dispersão das possibilidades para o saldo do fluxo de caixa é maior. Sendo assim, seria necessário um maior investimento inicial para garantir que a empresa possa cumprir com suas obrigações mesmo no pior cenário.