

# Análise Experimentos Surgem os Territórios

Jorge H C Fernandes

08/11/2021

## Introdução

Este documento apresenta análises preliminares realizadas na simulação surgem os territórios, que são comparadas com os dados analisados de forma empírica, obtidos do IBGE. O objetivo é avaliar que ajustes de parâmetros melhor sugerem a evolução da população brasileira, nos últimos cem anos, considerando-se a data base de 1919.

O primeiro passo foi ler o arquivo com os dados do modelo gerados na última simulação

```
library(readr)
setwd("~/Área de Trabalho/ExpComp20211/Comput-Experim-20202")
model_data <- read_csv("experiments/jhcf/ProducaoDaCiencia/Python/experimentos/model_data_iter_10_steps.csv")

## New names:
## * `` -> ...1

## Rows: 12960 Columns: 29

## -- Column specification -----
## Delimiter: ","
## dbl (28): ...1, gamma, trading_revigoration, num_nodes, beta, alpha, initial...
## lgl (1): visualizing

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
agent_data <- read_csv("experiments/jhcf/ProducaoDaCiencia/Python/experimentos/agent_data_iter_10_steps.csv")

## New names:
## * `` -> ...1

## Rows: 486000 Columns: 21

## -- Column specification -----
## Delimiter: ","
## chr (1): Trading state
## dbl (19): ...1, gamma, trading_revigoration, num_nodes, beta, alpha, initial...
## lgl (1): visualizing

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
#View(model_data)
#View(agent_data)
```

Os dados em `model_data` foram gerados com 10 passos de simulação para cada registro coletado, e foram executadas 10 simulações diferentes para cada configuração de parâmetros. O dataset contém 12.960 registros totais, de modo que foram exploradas 1296 configurações de parâmetros.

```
nrow(model_data)
```

```
## [1] 12960
```

```
nrow(agent_data)
```

```
## [1] 486000
```

O dataset principal possui as seguintes variáveis para cada registro:

```
names(model_data)
```

```
## [1] "...1"                                "gamma"
## [3] "trading_revigoration"                 "num_nodes"
## [5] "beta"                                "alpha"
## [7] "initial_trading_perc"                 "Run"
## [9] "AlphaAjuste"                         "Available"
## [11] "BetaAjuste"                          "ChannelsDeceasedRate"
## [13] "FinalPopulation"                     "GammaAjuste"
## [15] "InitialPopulation"                   "Resistant"
## [17] "StrongComponentsGT1"                 "Trading"
## [19] "Transitivity"                        "Weakcomponents"
## [21] "fraction_of_brazilian_population"    "delta_in"
## [23] "delta_out"                           "trading_spread_chance"
## [25] "trading_control_frequency"           "trading_recovery_chance"
## [27] "trading_resistance_chance"          "non_trading_decay"
## [29] "visualizing"
```

O dataset detalhe possui as seguintes variáveis para cada registro:

```
names(agent_data)
```

```
## [1] "...1"                                "gamma"
## [3] "trading_revigoration"                 "num_nodes"
## [5] "beta"                                "alpha"
## [7] "initial_trading_perc"                 "Run"
## [9] "AgentId"                             "Economic complexity"
## [11] "Population"                          "Trading state"
## [13] "fraction_of_brazilian_population"    "delta_in"
## [15] "delta_out"                           "trading_spread_chance"
## [17] "trading_control_frequency"           "trading_recovery_chance"
## [19] "trading_resistance_chance"          "non_trading_decay"
## [21] "visualizing"
```

Com base no descrito no relatório da simulação, as seguintes variáveis dependentes foram manipuladas, com os valores listados a seguir:

```
alpha <- c(0.31, 0.41, 0.51)
beta <- c(0.31, 0.54, 0.61)
gamma <- c(0.02, 0.05, 0.1)
initial_trading_perc <- c(0.1, 0.2, 0.4, 0.8)
trading_revigoration <- c(20, 40, 80)
```

Uma simples observação inicial dos registros mostra que há aparentemente um bug no framework MESA, pois as colunas `gamma`, `trading_revigoration`, `num_nodes`, `beta`, `alpha` e `initial_trade_perc` estão com valores

trocados entre si. Uma observação mais detalhada das demais colunas também sugere que apenas esses valores foram trocados. Por outro lado, através do estudo do código fonte de model.py observa-se que os valores de beta, alpha e gamma que foram efetivamente usados para geração do grafo livre de escala são os que foram ajustados automaticamente conforme informado na apresentação do código do construtor da classe Python do modelo, que estão nas variáveis AlphaAjuste, BetaAjuste e GammaAjuste. Assim sendo, as colunas alpha, beta e gamma reais serão desprezadas.

Desse modo, as seguintes operações vão gerar um dataset com apenas os valores a serem utilizados nessa primeira análise:

```
names(model_data)[names(model_data)=="gamma"]<- "num_nodes_"
names(model_data)[names(model_data)=="initial_trading_perc"]<- "trading_revigoration_"
names(model_data)[names(model_data)=="alpha"]<- "initial_trading_perc_"
names(model_data)[names(model_data)=="trading_revigoration"]<- "alpha"
names(model_data)[names(model_data)=="beta"]<- "gamma"
names(model_data)[names(model_data)=="num_nodes"]<- "beta"

model_data_clean <- model_data[,c("Run", "num_nodes_", "alpha", "AlphaAjuste", "beta", "BetaAjuste", "gamma", "GammaAjuste", "initial_trading_perc_", "trading_revigoration_")]
nrow(model_data_clean)

## [1] 12960

#View(model_data_clean)

names(agent_data)[names(agent_data)=="gamma"]<- "num_nodes_"
names(agent_data)[names(agent_data)=="initial_trading_perc"]<- "trading_revigoration_"
names(agent_data)[names(agent_data)=="alpha"]<- "initial_trading_perc_"
names(agent_data)[names(agent_data)=="trading_revigoration"]<- "alpha"
names(agent_data)[names(agent_data)=="beta"]<- "gamma"
names(agent_data)[names(agent_data)=="num_nodes"]<- "beta"
agent_data_clean <- agent_data[,c("Run", "num_nodes_", "alpha", "beta", "gamma", "initial_trading_perc_", "trading_revigoration_")]
nrow(agent_data_clean)

## [1] 486000

#View(agent_data_clean)
```

De outra forma, as seguintes variáveis foram controladas, com os valores indicados: \* “fraction\_of\_brazilian\_population”: 30/211, \* “delta\_in”: 0.02, \* “delta\_out”: 0.02, \* “trading\_spread\_chance”: 0.5, \* “trading\_control\_frequency”: 0.2, \* “trading\_recovery\_chance”: 0.25, \* “trading\_resistance\_chance”: 0.2, \* “non\_trading\_decay”: 20,

A explicação para cada uma das variáveis pode ser vista no começo do relatório descritivo da simulação.

Com base nas variáveis acima indicadas, cabe então a formulação de algumas questões que podem ser inicialmente respondidas pelos dados coletados: \* Qual o número de nós “num\_nodes” na simulação que melhor produz dados estatisticamente bem dispersos : [10, 20, 40, 80], para modelar uma rede similar à empiricamente investigada?? \* Qual a combinação de alpha, beta e gamma, que melhor modela uma rede similar à empiricamente investigada? \* Qual a percentagem inicial de territórios que se encontram comercializando (“initial\_trading\_perc”), que melhor modela uma rede similar à empiricamente investigada: [0.1, 0.2, 0.4, 0.8]? \* Qual a taxa de revigoração de canais de comércio (“trading\_revigoration”): [20, 40, 80], que melhor modela a rede empiricamente investigada? Tudo isso, considerando fixas as variáveis de controle.

Ainda antes de avançar para a questão dos graficos, cabe perguntar quais as variáveis dependentes da simulação (resultados, agregados) melhor representam as condições dos dados empíricos analisados. Essas variáveis dependentes são: \* [final\_population] População do território; \* [number\_trading] quantidade de territórios que estão ativamente tentando comercializar; \* [number\_resistant] quantidade de territórios que estão resistentes ao comércio; \* [number\_available] quantidade de territórios que estão

disponíveis para comercializar; \* [network\_number\_strongly\_connected\_components\_not\_unitary] quantidade de componentes fortemente conectados no grafo, com tamanho acima de 1 vértice; \* [network\_number\_weakly\_connected\_components\_not\_unitary] quantidade de componentes fracamente conectados no grafo; \* [network\_transitivity] Transitividade da rede \* [network\_channels\_deceased\_rate] Taxa média de canais de troca que morrem anualmente;

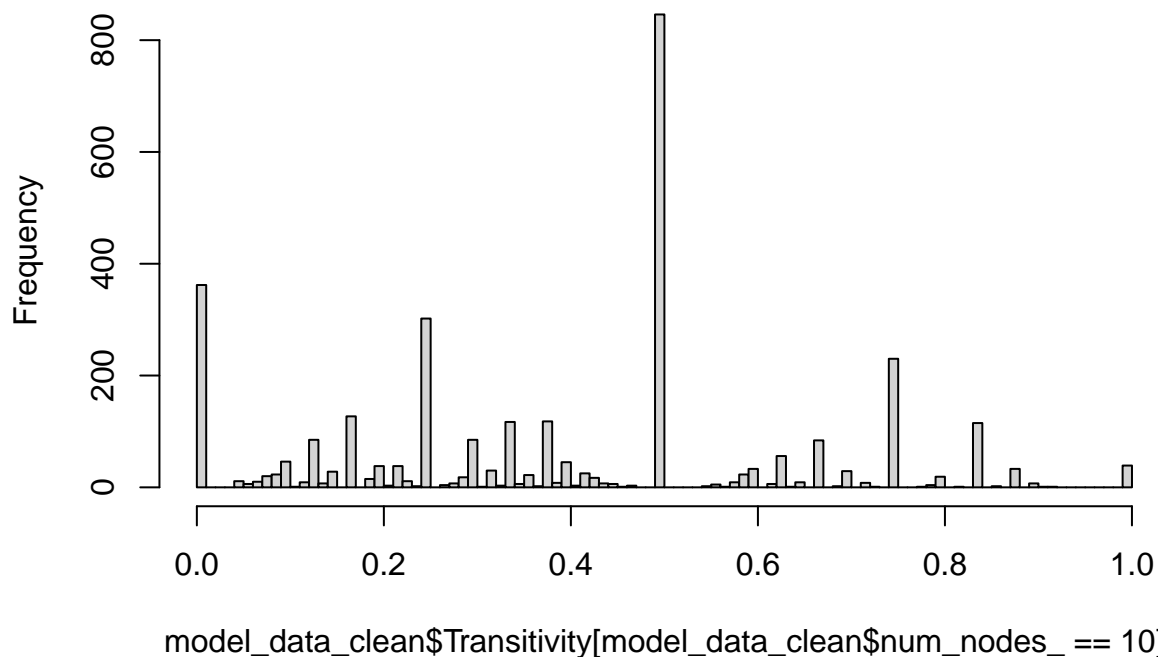
Embora a simulação tenha trabalhado com a situação do comércio, e que os dados empíricos tenham elementos suficientes para estudo das relações de influência entre os municípios, na análise a seguir serão inicialmente consideradas apenas as duas variáveis dependentes, que evidenciam propriedades estruturais básicas da rede, e que são: \* População final do território (está crescendo? Cresceu durante a simulação de dez anos, entre 32 e 36 milhões de habitantes?) \* Transitividade da rede (Apresenta valores de transitividade próximos aos obtidos nos dados empíricos?)

Os parâmetros para variáveis independentes que apresentarem respostas afirmativas a essas duas questões serão usados em um novo ciclo de simulação, para considerar um período mais longo de simulação, no aprofundamento futuro desse trabalho.

O primeiro parâmetro avaliado foi o da transitividade, a fim de se avaliar as diferenças na variação de transitividade nos grafos com 10, 20, 40 e 80 vértices.

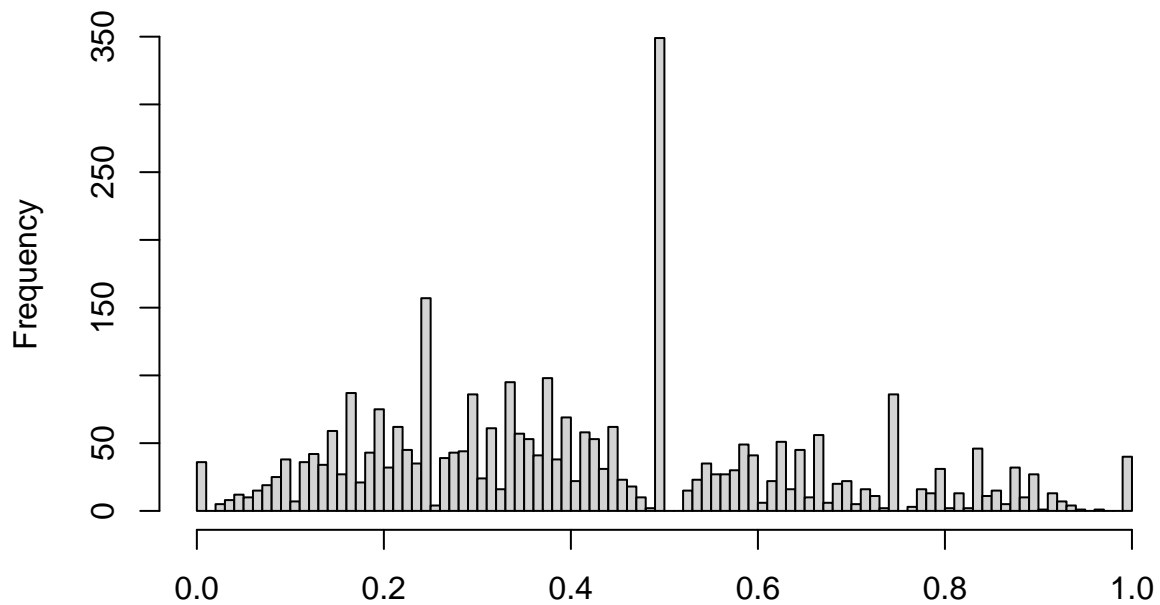
```
hist(model_data_clean$Transitivity[model_data_clean$num_nodes_==10],breaks=100)
```

**ogram of model\_data\_clean\$Transitivity[model\_data\_clean\$num\_node**



```
hist(model_data_clean$Transitivity[model_data_clean$num_nodes_==20],breaks=100)
```

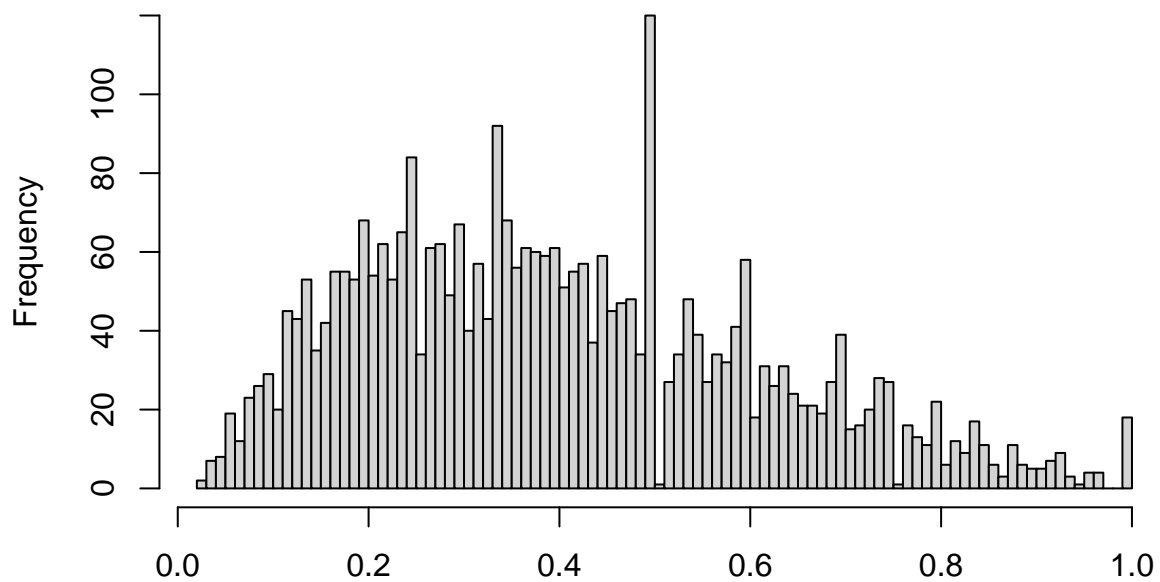
ogram of model\_data\_clean\$Transitivity[model\_data\_clean\$num\_node



model\_data\_clean\$Transitivity[model\_data\_clean\$num\_nodes\_ == 20]

```
hist(model_data_clean$Transitivity[model_data_clean$num_nodes_==40],breaks=100)
```

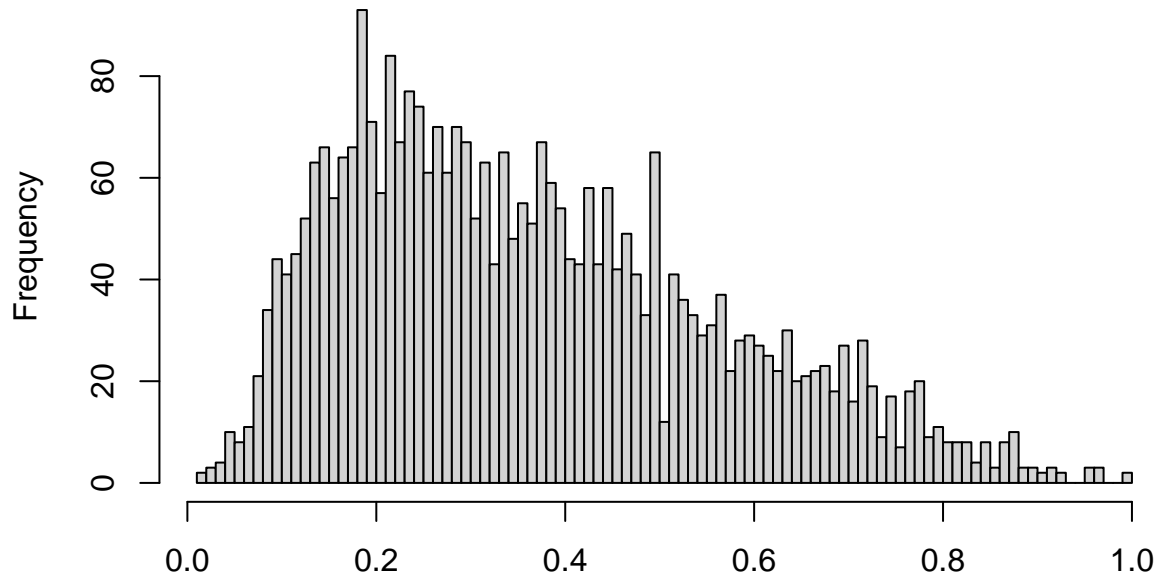
ogram of model\_data\_clean\$Transitivity[model\_data\_clean\$num\_node



model\_data\_clean\$Transitivity[model\_data\_clean\$num\_nodes\_ == 40]

```
hist(model_data_clean$Transitivity[model_data_clean$num_nodes_==80],breaks=100)
```

## Histogram of model\_data\_clean\$Transitivity[model\_data\_clean\$num\_node



```
model_data_clean$Transitivity[model_data_clean$num_nodes_ == 80]
```

Observa-se que os grafos com 10 e 20 vértices não apresentaram normalidade no histograma do número de triângulos, o que faz supor que essas quantidades são pequenas demais para modelar um processo estatisticamente estável, e desse modo foram eliminados os registros de simulações que usavam 10 e 20 vértices.

```
model_data_clean <- model_data_clean[model_data_clean$num_nodes_ %in% c(40,80),]
nrow(model_data_clean)
```

```
## [1] 6480
```

Também serão removidos os registros de dados individuais dos agentes simulados em redes com esses números de vértices de 10 e 20.

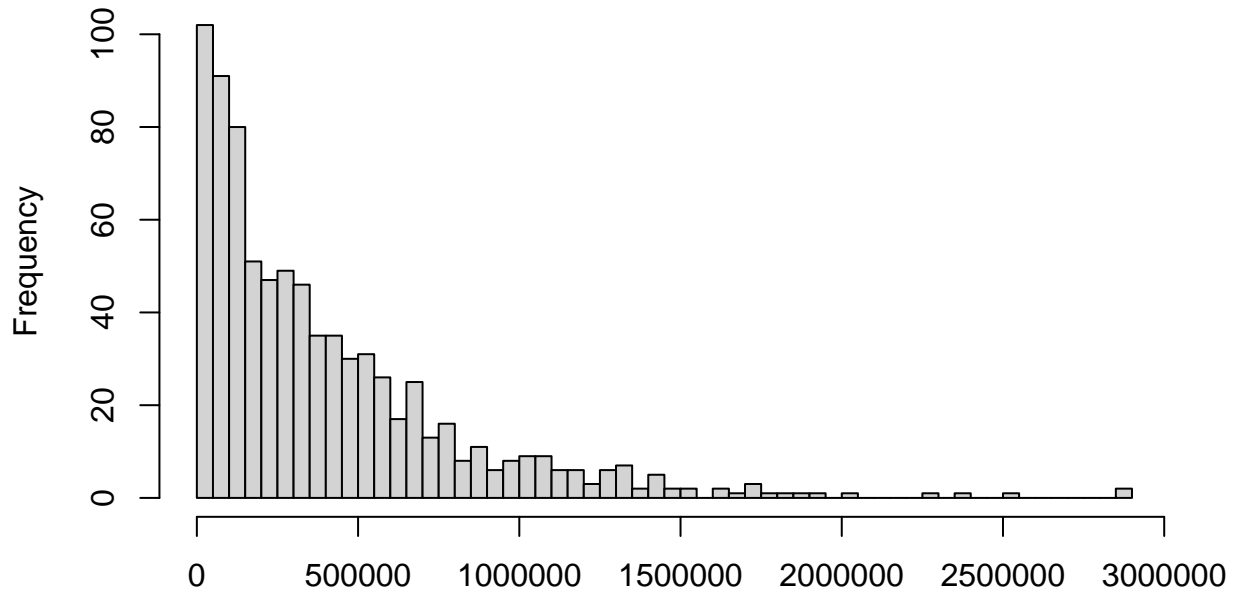
```
agent_data_clean <- agent_data_clean[agent_data_clean$num_nodes_ %in% c(40,80),]
nrow(agent_data_clean)
```

```
## [1] 388800
```

A fim de se obter os parâmetros que produzem as melhores distribuições de frequência da população, as linhas a seguir comparam a distribuição obtida com uma rede de 80 nós, em comparação a uma que tem 40 nós, mantendo-se todos os demais parâmetros fixados. ou seja, em cada uma das 10 simulações, totalizando 800 dados populacionais de territórios, relacionados em cada um dos histogramas.

```
hist(agent_data_clean[agent_data_clean$num_nodes_==80&
  agent_data_clean$alpha==0.51&
  agent_data_clean$beta==0.61&
  agent_data_clean$gamma==0.02&
  agent_data_clean$initial_trading_perc_==0.8&
  agent_data_clean$trading_revigoration_==20,]$Population,breaks=100, main = "Frequên
```

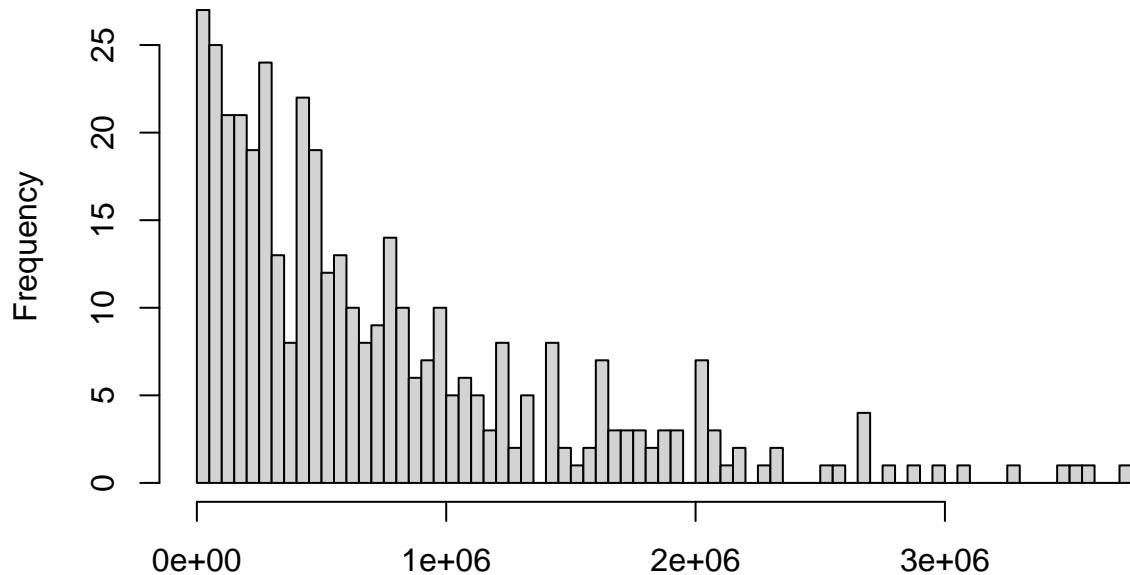
## Frequência da população de territórios: Simulações com 80 vértice



1 & agent\_data\_clean\$beta == 0.61 & agent\_data\_clean\$gamma == 0.02 & agent\_data\_

```
hist(agent_data_clean[agent_data_clean$num_nodes_==40&
  agent_data_clean$alpha==0.51&
  agent_data_clean$beta==0.61&
  agent_data_clean$gamma==0.02&
  agent_data_clean$initial_trading_perc_==0.8&
  agent_data_clean$trading_revigoration_==20,]$Population,breaks=100,main = "Frequên
```

## Frequência da população de territórios: Simulações com 40 vértice



```
1 & agent_data_clean$beta == 0.61 & agent_data_clean$gamma == 0.02 & agent_data_
```

Tendo em vista que a distribuição de frequência com o uso de 40 nós, também não apresenta aparente uniformidade na distribuição de frequência, decidiu-se por eliminar as configurações de simulação com 40 nós, e focar apenas das de 80. Nota-se, por outro lado, que a curva ainda apresenta uma forma bem menos acentuada que a que ocorre na curva populacional das cidades brasileiras, conforme os dados do IBGE. Entretanto, foram feitas simulações com apenas 10 anos, e nesse caso deve-se avaliar se a curva vai se acentuar com 100 anos de simulação.

```
model_data_clean_2 <- model_data_clean[model_data_clean$num_nodes_ == 80,]  
nrow(model_data_clean_2)
```

```
## [1] 3240
```

```
agent_data_clean_2 <- agent_data_clean[agent_data_clean$num_nodes_==80,]  
nrow(agent_data_clean_2)
```

```
## [1] 259200
```

Agora, será feita a geração de uma série de gráficos que exploram a variação populacional conforme as variações de configuração.

### Explorando alpha, beta e gamma

Sabendo-se que para cada uma das 27 combinações de valores distintos para alpha, beta e gamma, foram geradas 12 simulações para as variações de `initial_trading_perc` e `trading_revigoration`, chega-se a um número de  $12 * 80 = 960$  territórios simulados, para cada combinação de alpha, beta e gamma.

```
alpha <- c(0.31, 0.41, 0.51)  
beta <- c(0.31, 0.54, 0.61)  
gamma <- c(0.02, 0.05, 0.1)  
length(alpha)*length(beta)*length(gamma)
```

```
## [1] 27
```



```
initial_trading_perc <- c(0.1, 0.2, 0.4, 0.8)
trading_revigoration <- c(20, 40, 80)
length(initial_trading_perc)* length(trading_revigoration)
```

```
## [1] 12
```

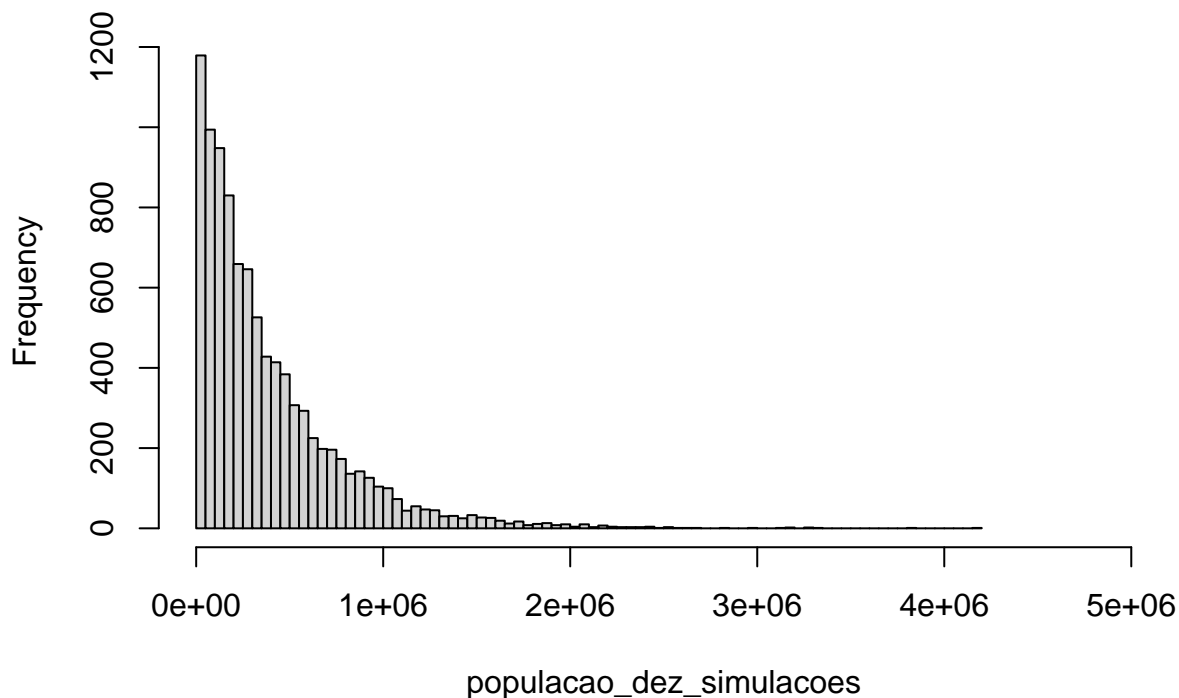
```
length(initial_trading_perc)* length(trading_revigoration)*80
```

```
## [1] 960
```

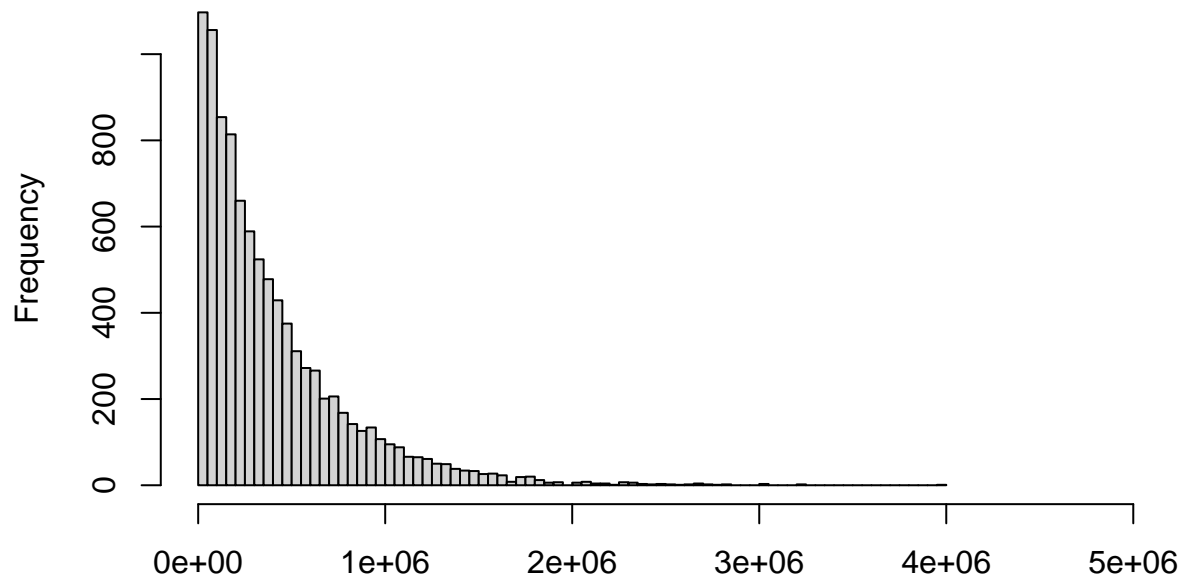
O código a seguir gera os histogramas para cada uma dessas combinações de alpha, beta e gamma

```
for (gam_ in gamma) {
for (alp_ in alpha) {
  for (bet_ in beta) {
    populacao_dez_simulacoes <- agent_data_clean_2[
      agent_data_clean_2$alpha==alp_ &
      agent_data_clean_2$beta==bet_ &
      agent_data_clean_2$gamma==gam_,]$Population
    hist(populacao_dez_simulacoes,breaks=100,xlim=range(0,5000000),
      main = paste("Pop: alp:",alp_," bet:",bet_," gam:",gam_," max:",max(as.integer(populacao_dez_simulacoes))
    )
  }
}
}
```

**Pop: alp: 0.31 bet: 0.31 gam: 0.02 max: 4185818 avg: 383905**

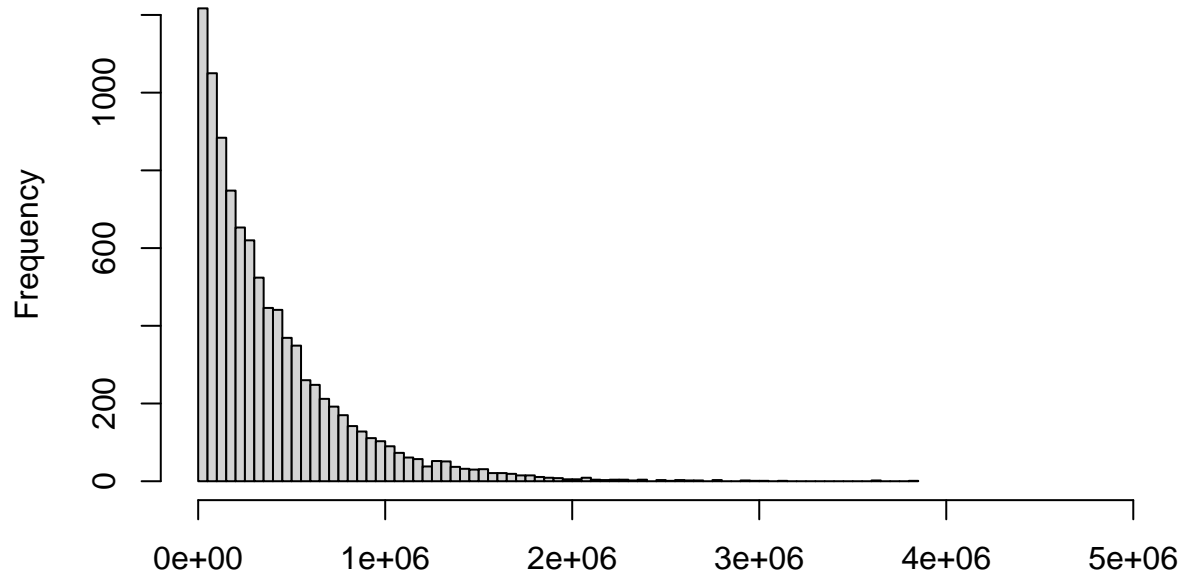


**Pop: alp: 0.31 bet: 0.54 gam: 0.02 max: 3963719 avg: 398480**



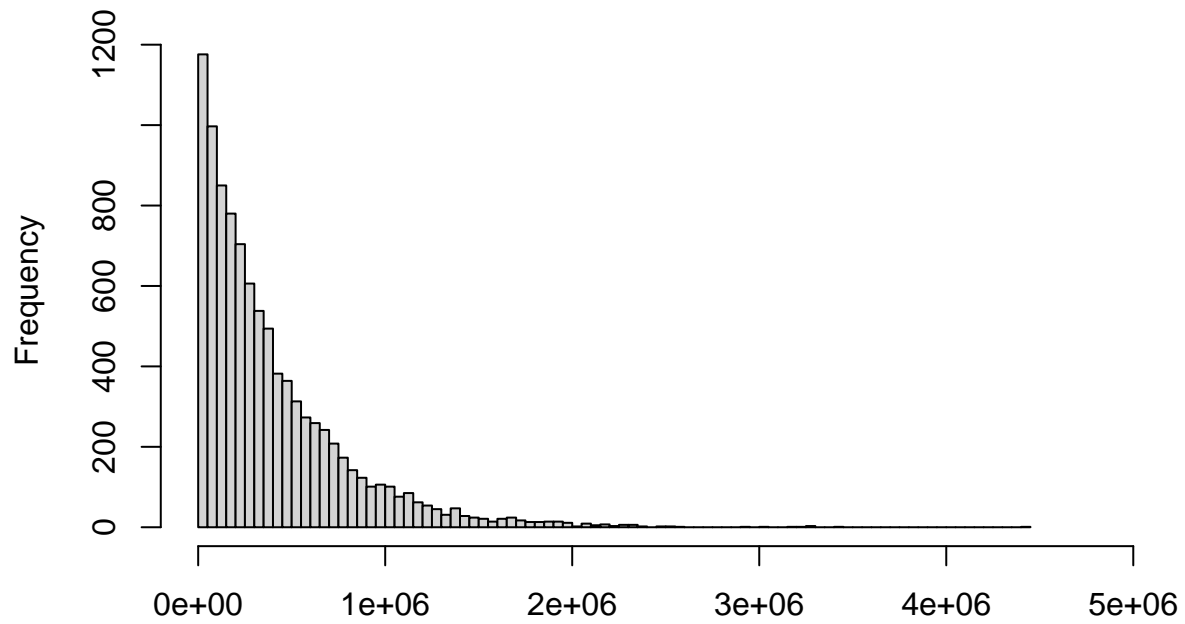
populacao\_dez\_simulacoes

**Pop: alp: 0.31 bet: 0.61 gam: 0.02 max: 3841940 avg: 387951**



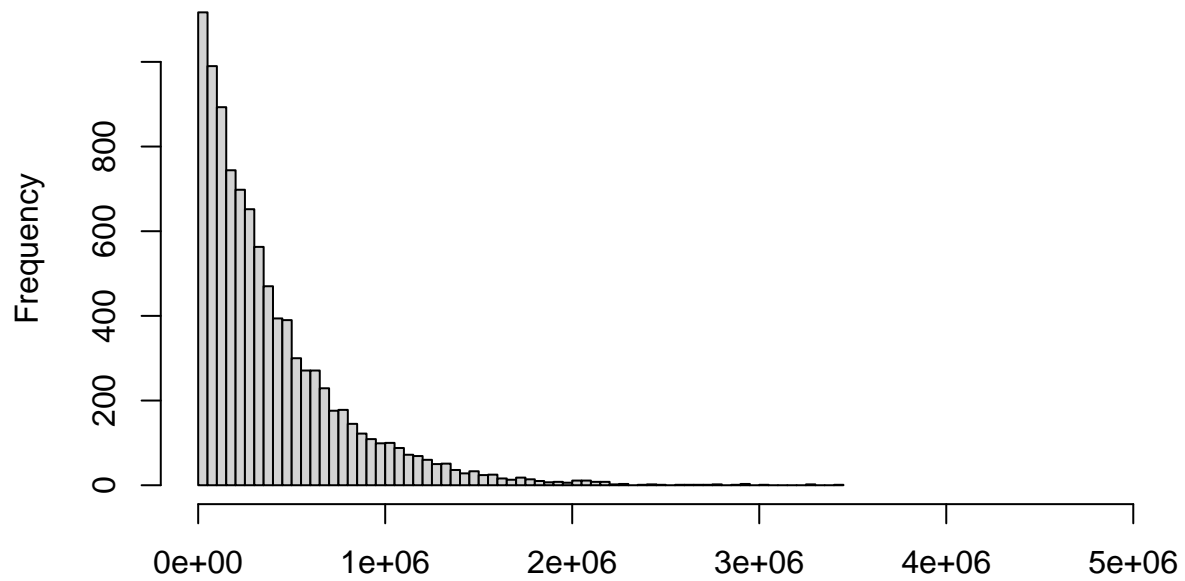
populacao\_dez\_simulacoes

**Pop: alp: 0.41 bet: 0.31 gam: 0.02 max: 4424574 avg: 394149**



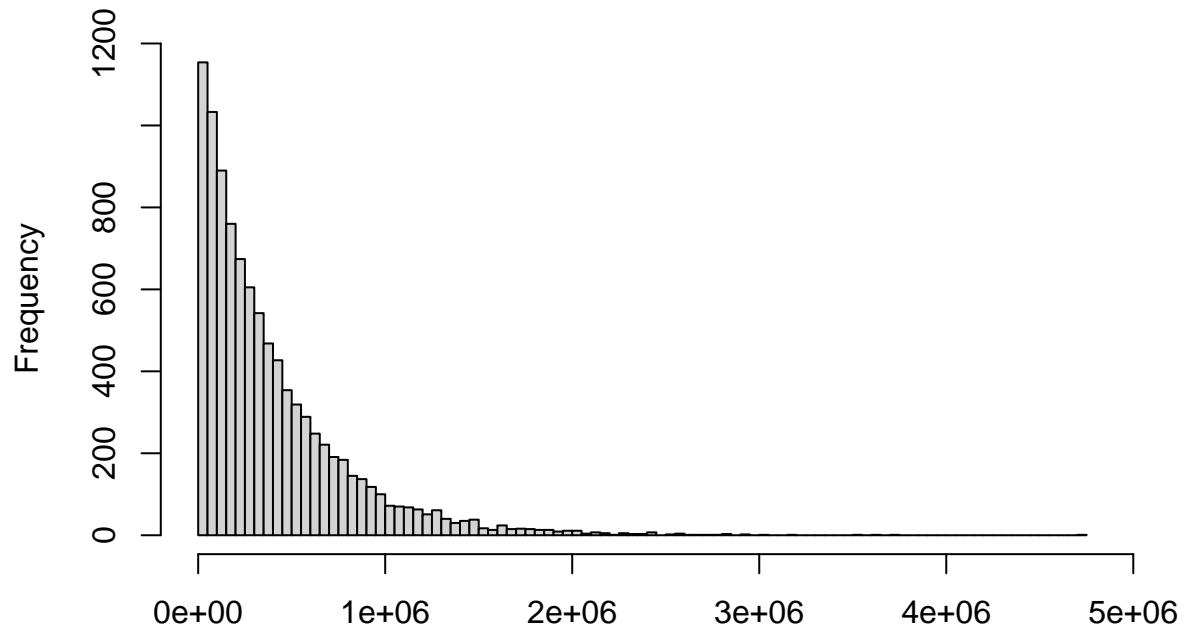
populacao\_dez\_simulacoes

**Pop: alp: 0.41 bet: 0.54 gam: 0.02 max: 3431333 avg: 395236**



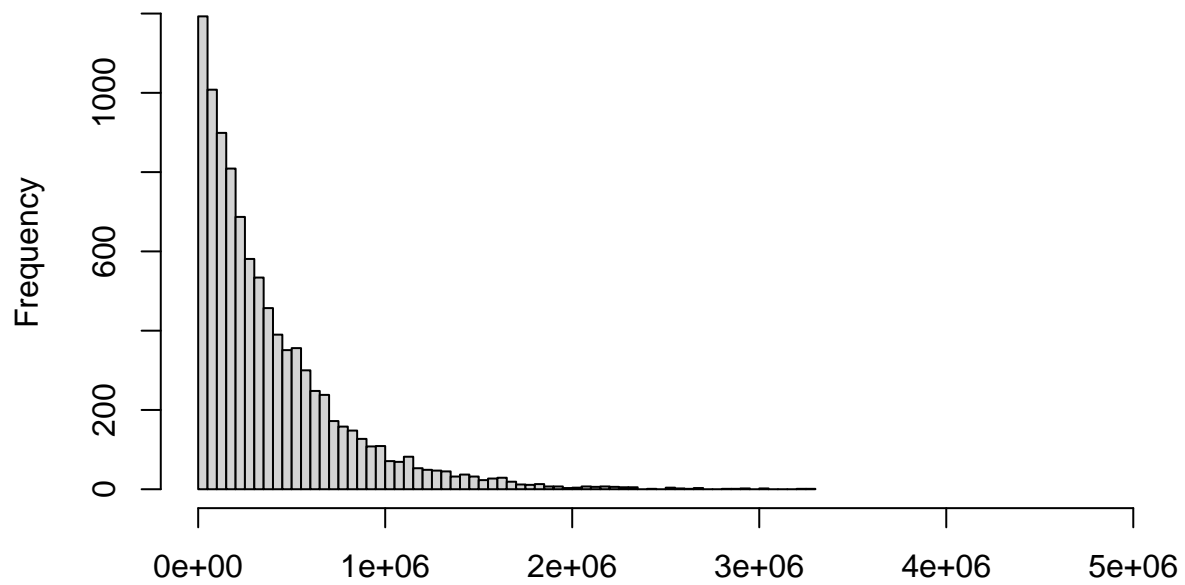
populacao\_dez\_simulacoes

**Pop: alp: 0.41 bet: 0.61 gam: 0.02 max: 4712368 avg: 394647**



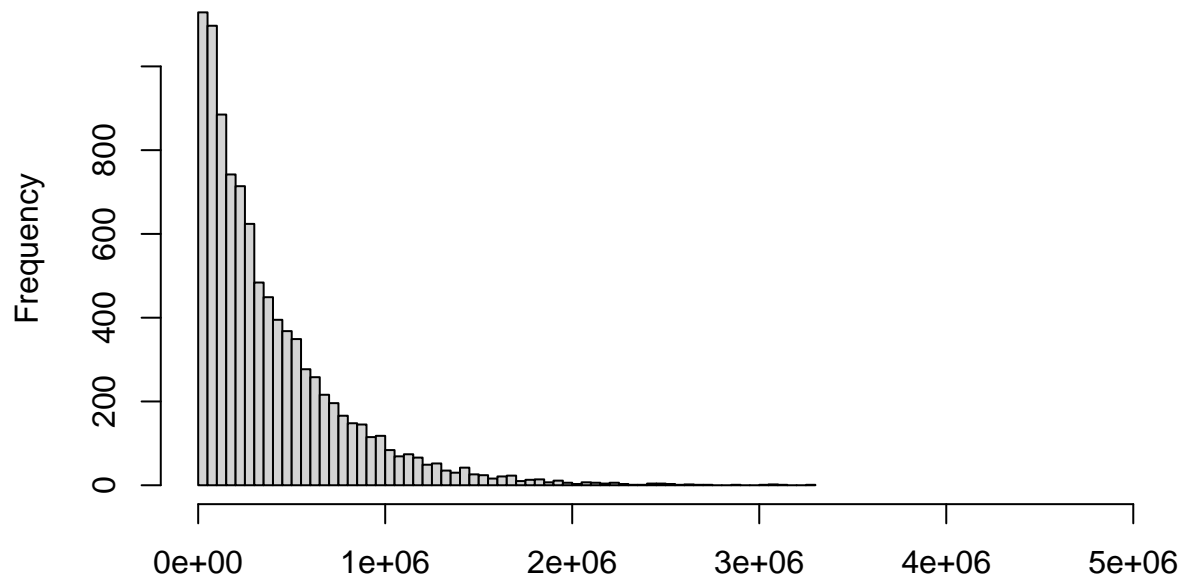
populacao\_dez\_simulacoes

**Pop: alp: 0.51 bet: 0.31 gam: 0.02 max: 3272531 avg: 388212**



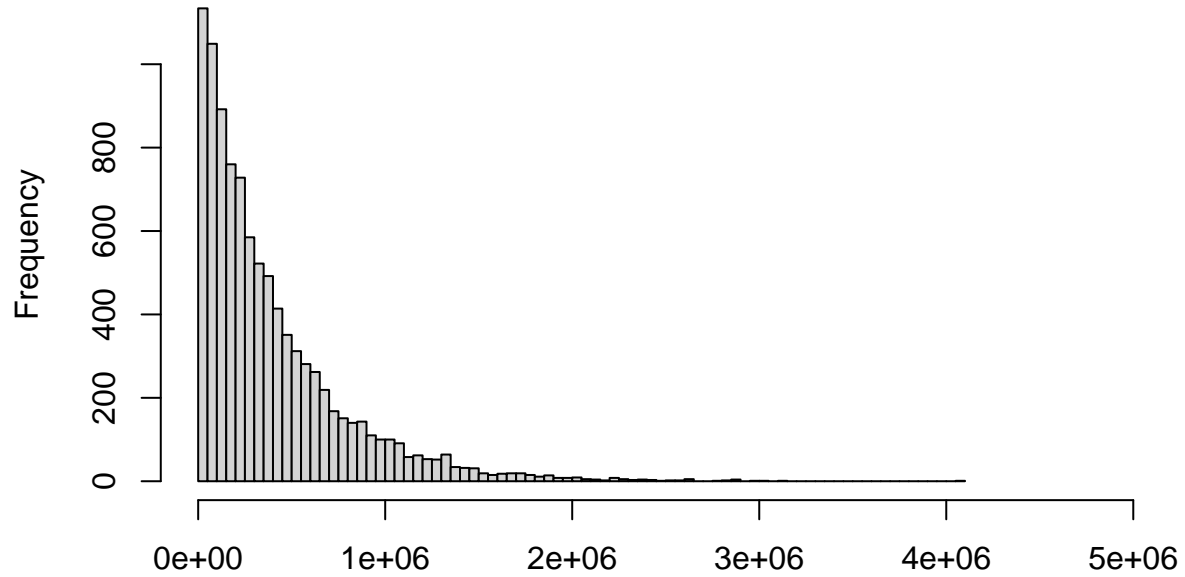
populacao\_dez\_simulacoes

**Pop: alp: 0.51 bet: 0.54 gam: 0.02 max: 3284474 avg: 389734**



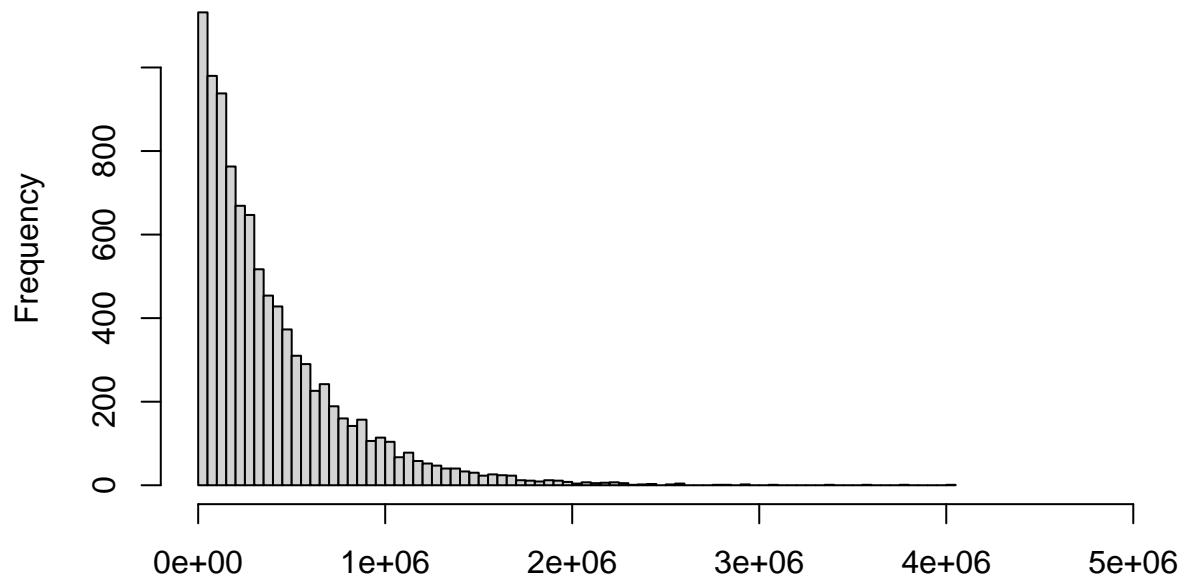
populacao\_dez\_simulacoes

**Pop: alp: 0.51 bet: 0.61 gam: 0.02 max: 4081766 avg: 393535**



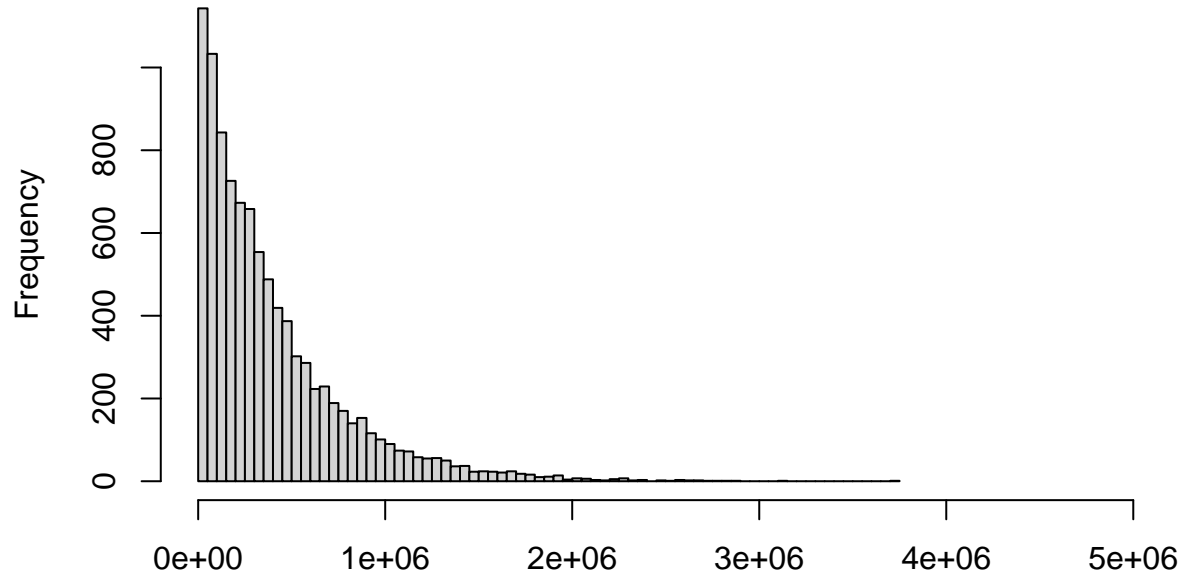
populacao\_dez\_simulacoes

**Pop: alp: 0.31 bet: 0.31 gam: 0.05 max: 4021148 avg: 394891**



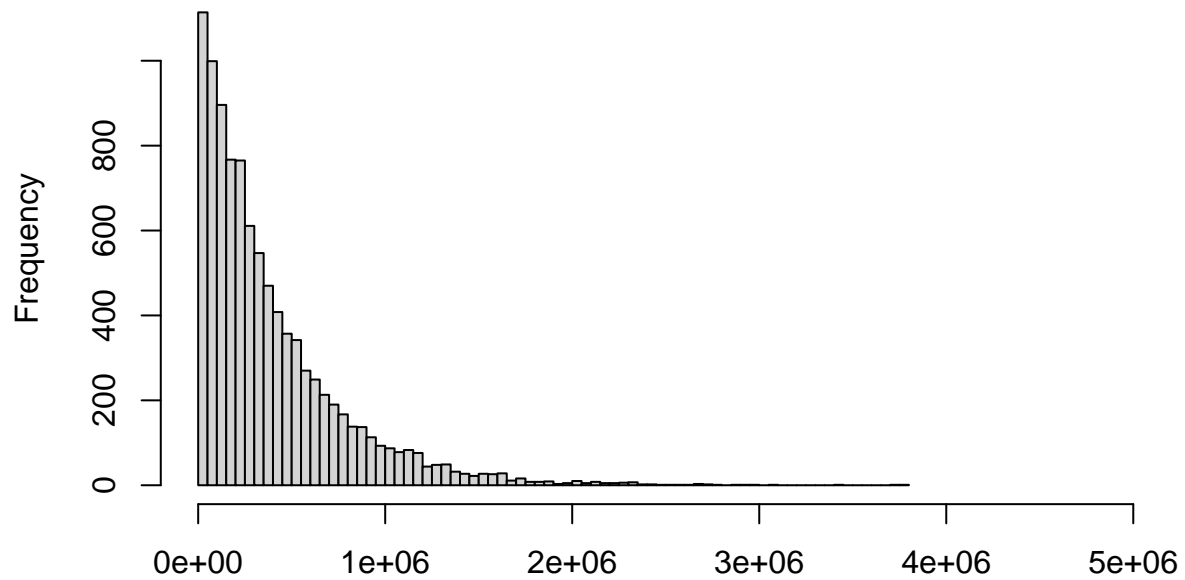
populacao\_dez\_simulacoes

**Pop: alp: 0.31 bet: 0.54 gam: 0.05 max: 3742525 avg: 395285**



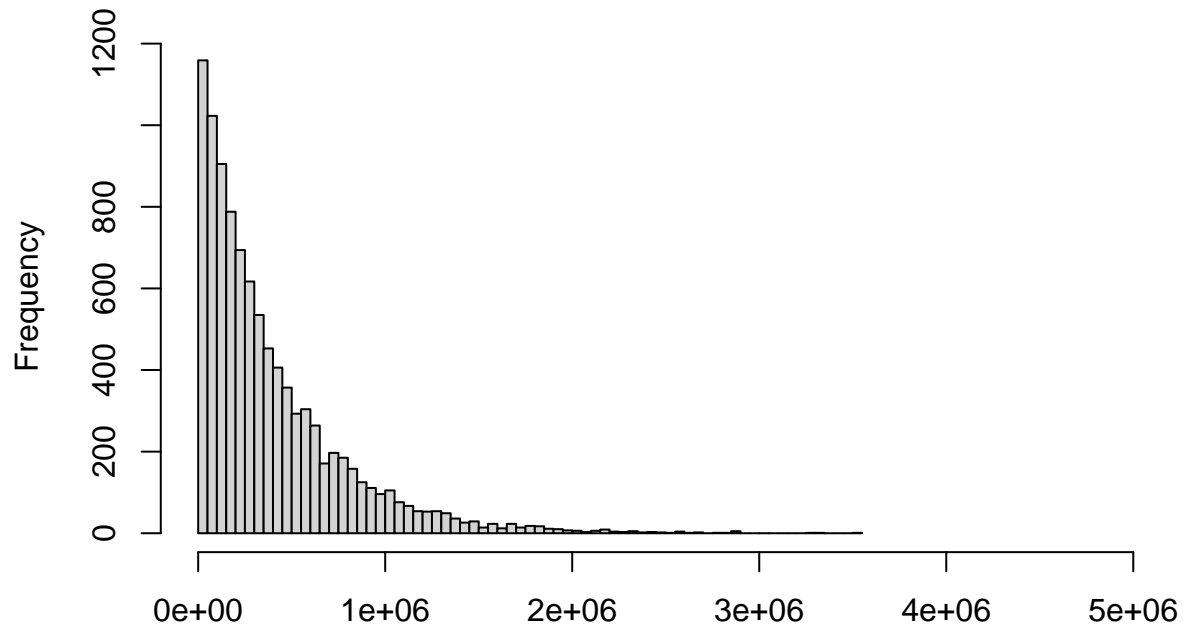
populacao\_dez\_simulacoes

**Pop: alp: 0.31 bet: 0.61 gam: 0.05 max: 3766304 avg: 391459**



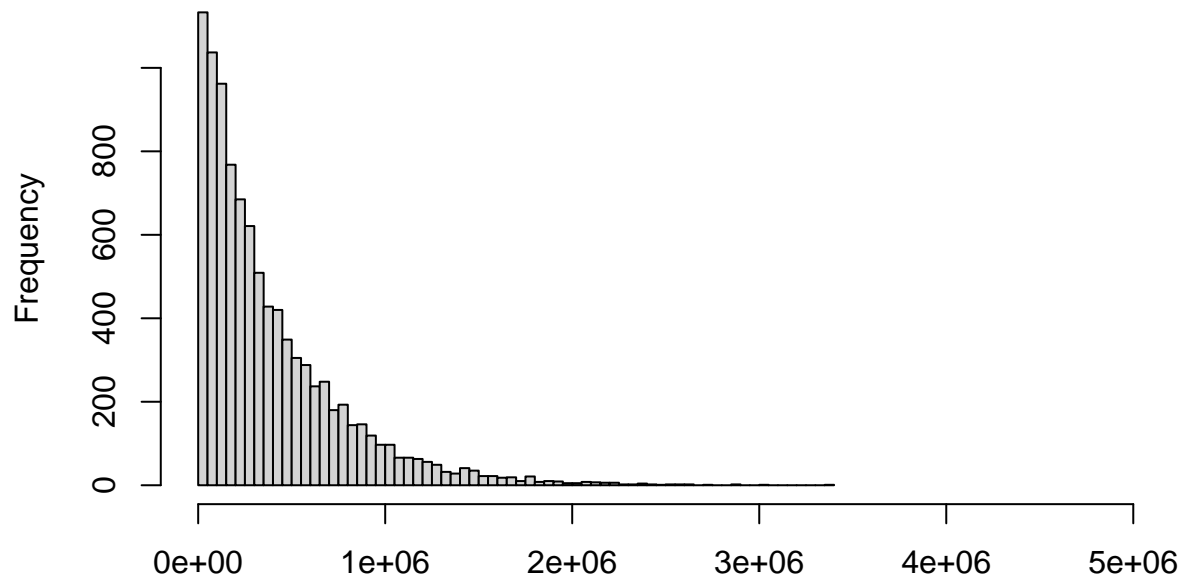
populacao\_dez\_simulacoes

**Pop: alp: 0.41 bet: 0.31 gam: 0.05 max: 3517968 avg: 391018**



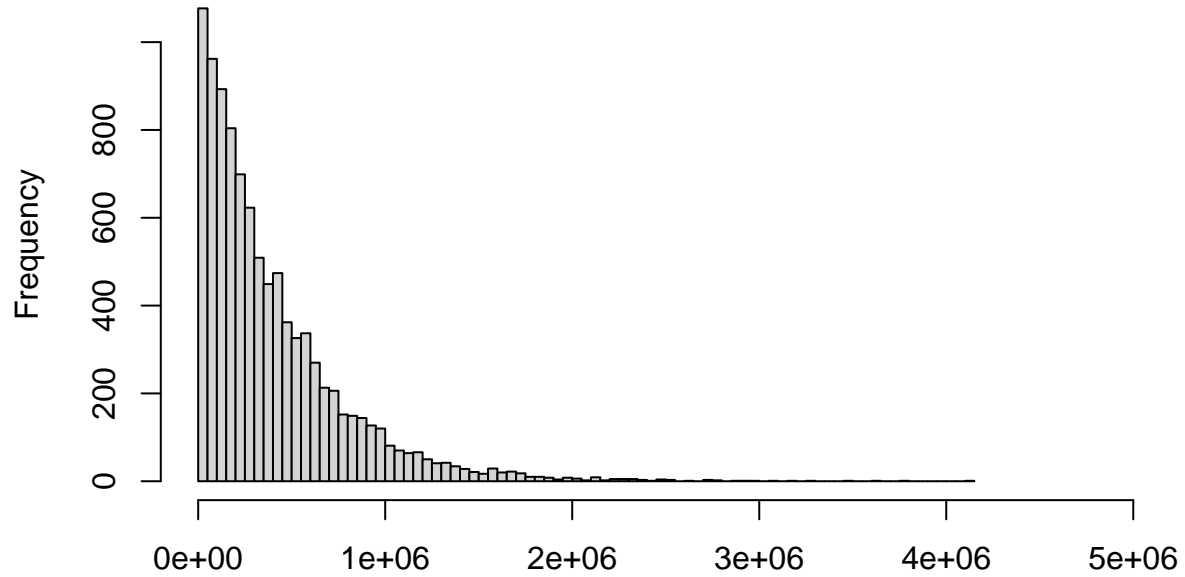
populacao\_dez\_simulacoes

**Pop: alp: 0.41 bet: 0.54 gam: 0.05 max: 3392887 avg: 388883**



populacao\_dez\_simulacoes

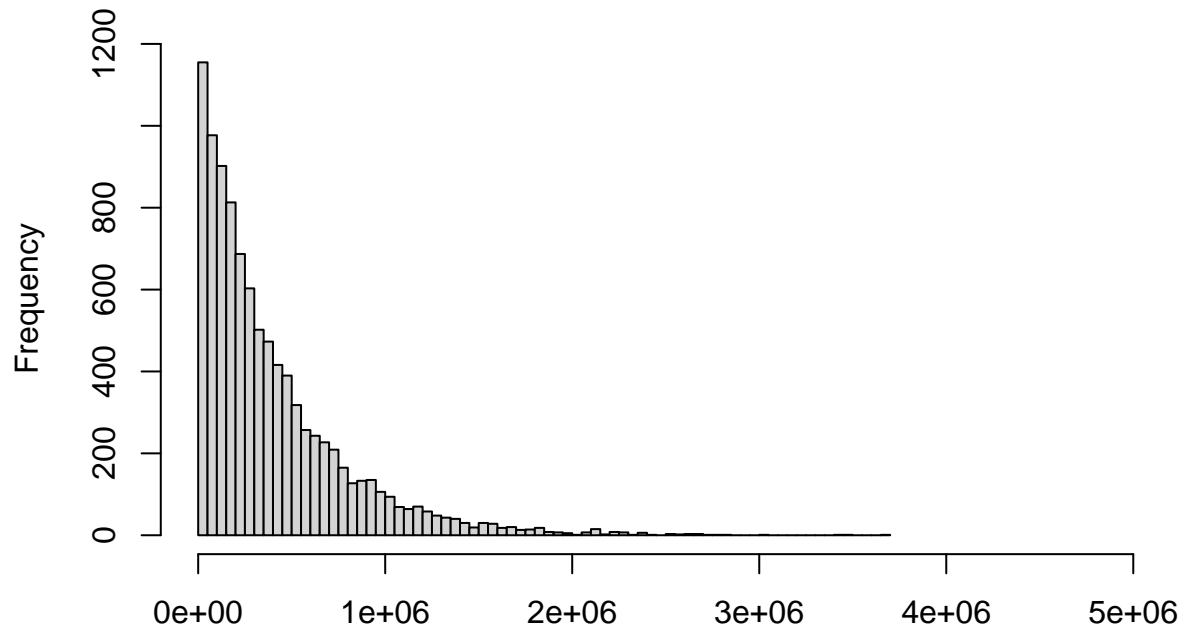
**Pop: alp: 0.41 bet: 0.61 gam: 0.05 max: 4141333 avg: 395847**



populacao\_dez\_simulacoes

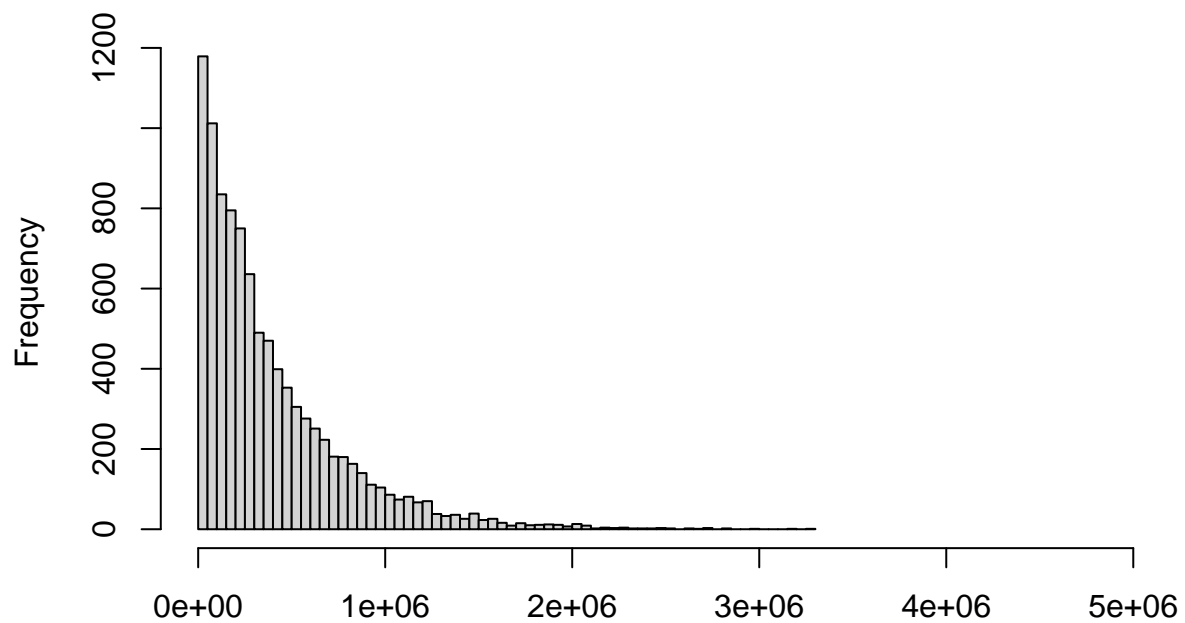


**Pop: alp: 0.51 bet: 0.31 gam: 0.05 max: 3671387 avg: 394873**



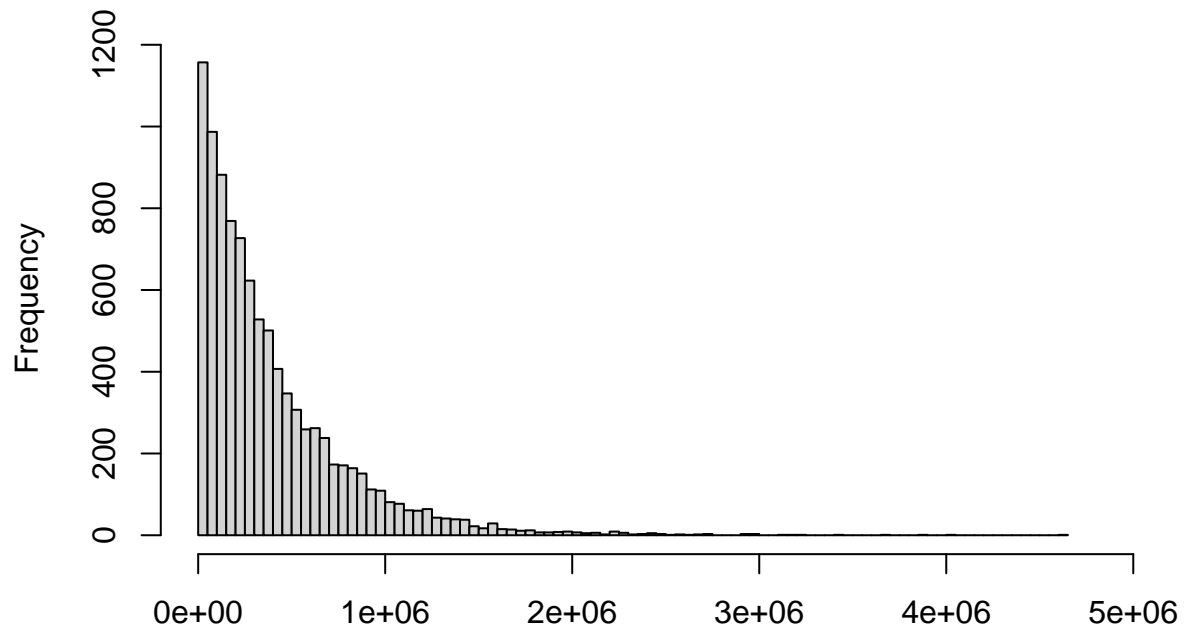
populacao\_dez\_simulacoes

**Pop: alp: 0.51 bet: 0.54 gam: 0.05 max: 3289705 avg: 390421**



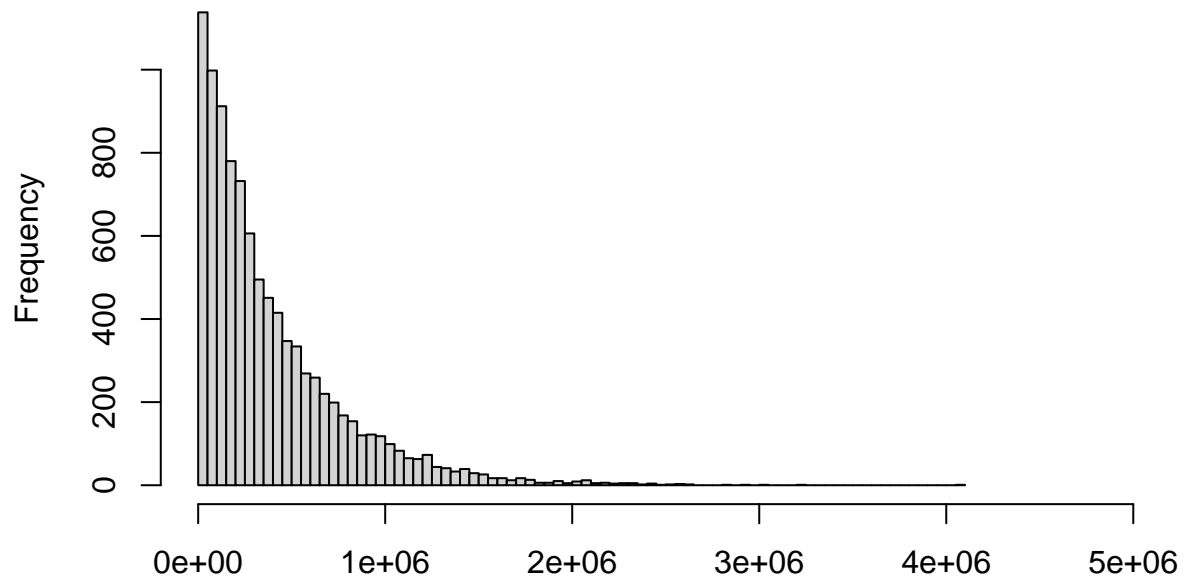
populacao\_dez\_simulacoes

**Pop: alp: 0.51 bet: 0.61 gam: 0.05 max: 4647171 avg: 393285**



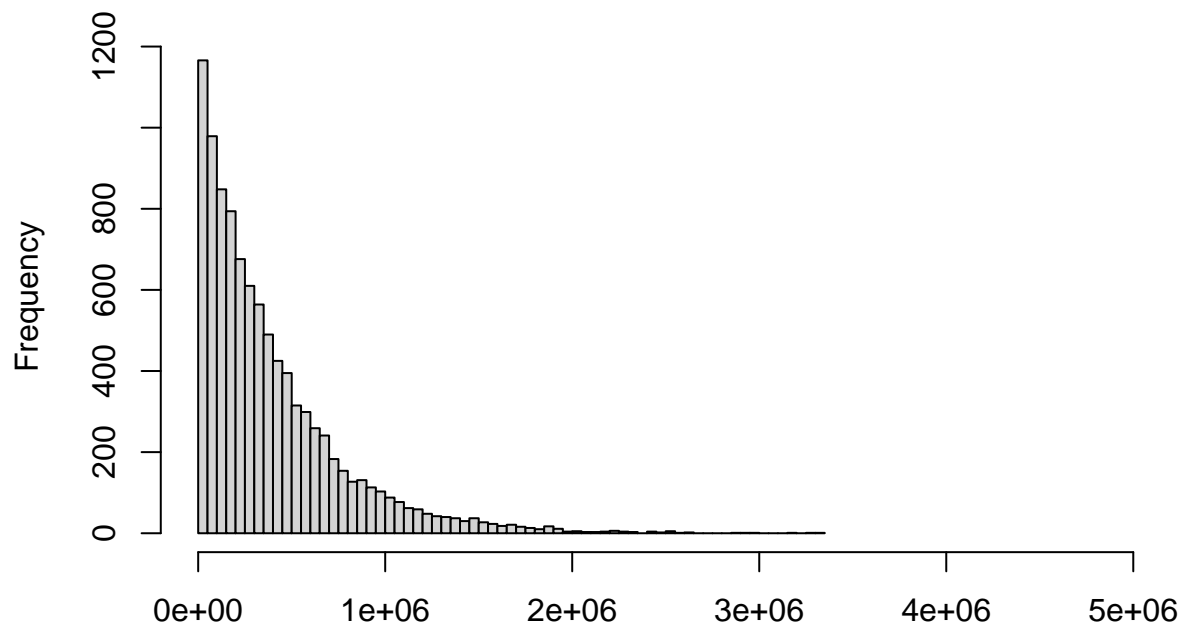
populacao\_dez\_simulacoes

**Pop: alp: 0.31 bet: 0.31 gam: 0.1 max: 4061964 avg: 393264**



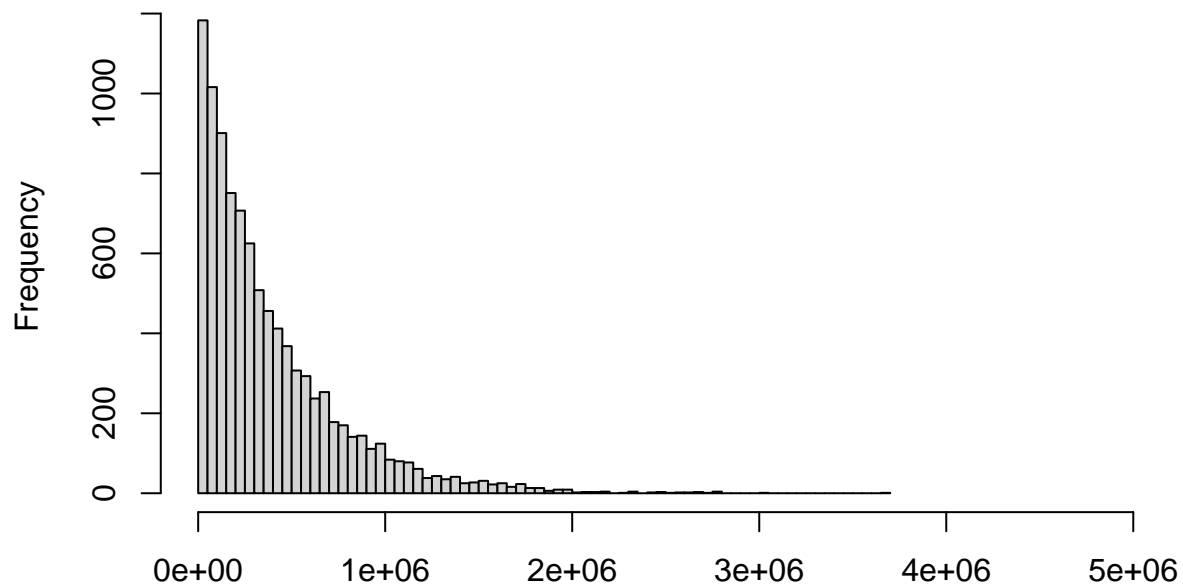
populacao\_dez\_simulacoes

**Pop: alp: 0.31 bet: 0.54 gam: 0.1 max: 3301048 avg: 390366**



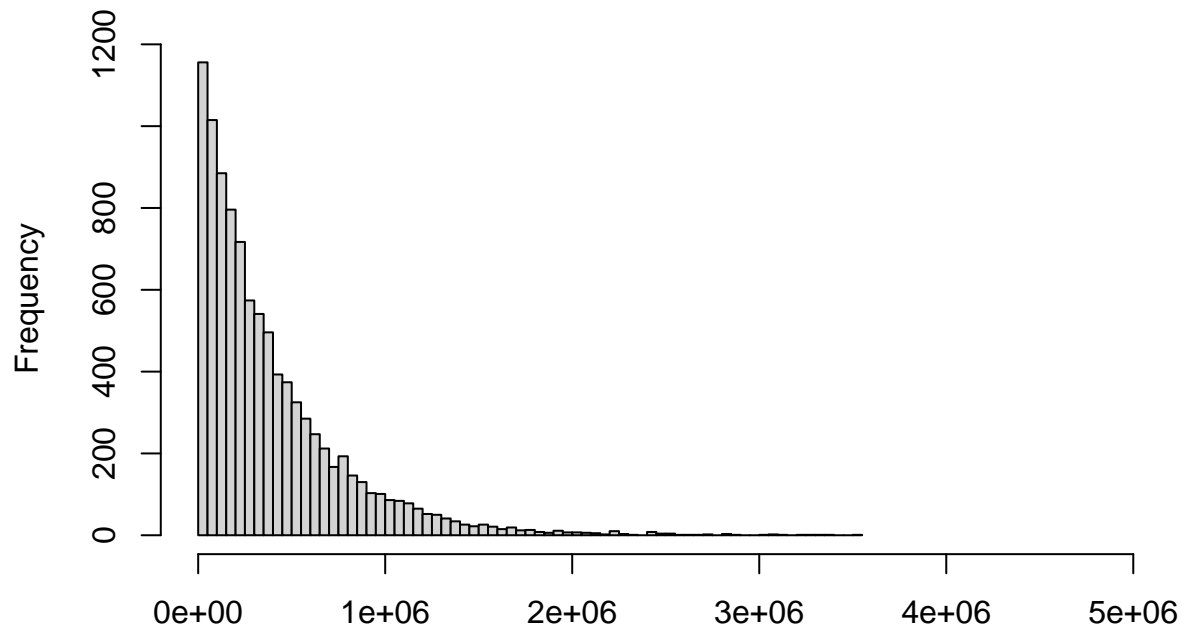
populacao\_dez\_simulacoes

**Pop: alp: 0.31 bet: 0.61 gam: 0.1 max: 3679716 avg: 387429**



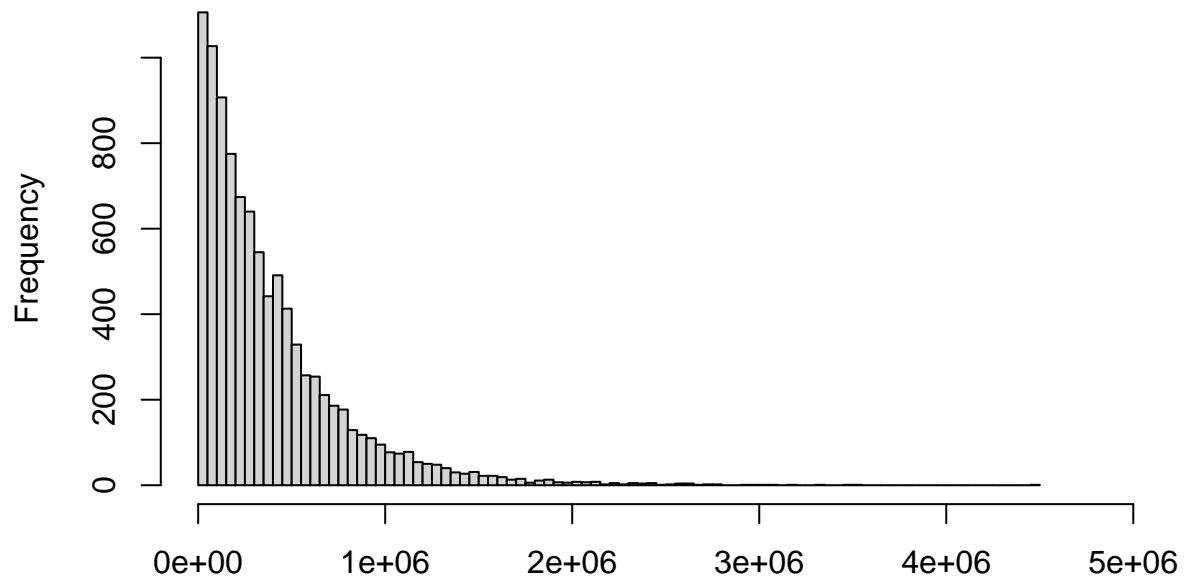
populacao\_dez\_simulacoes

**Pop: alp: 0.41 bet: 0.31 gam: 0.1 max: 3520167 avg: 390107**



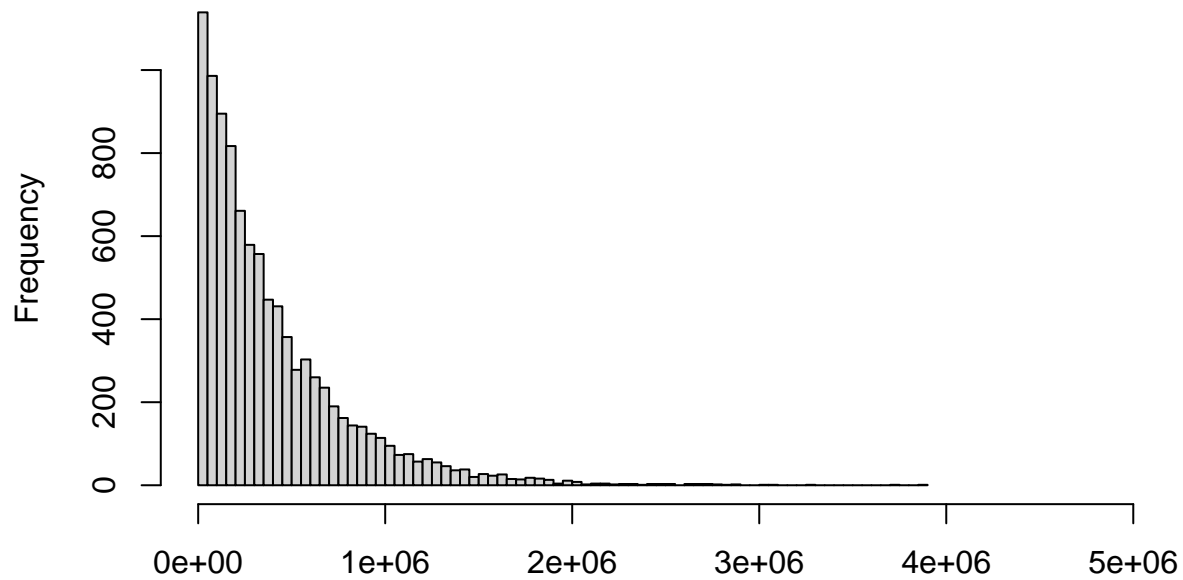
populacao\_dez\_simulacoes

**Pop: alp: 0.41 bet: 0.54 gam: 0.1 max: 4496278 avg: 388550**



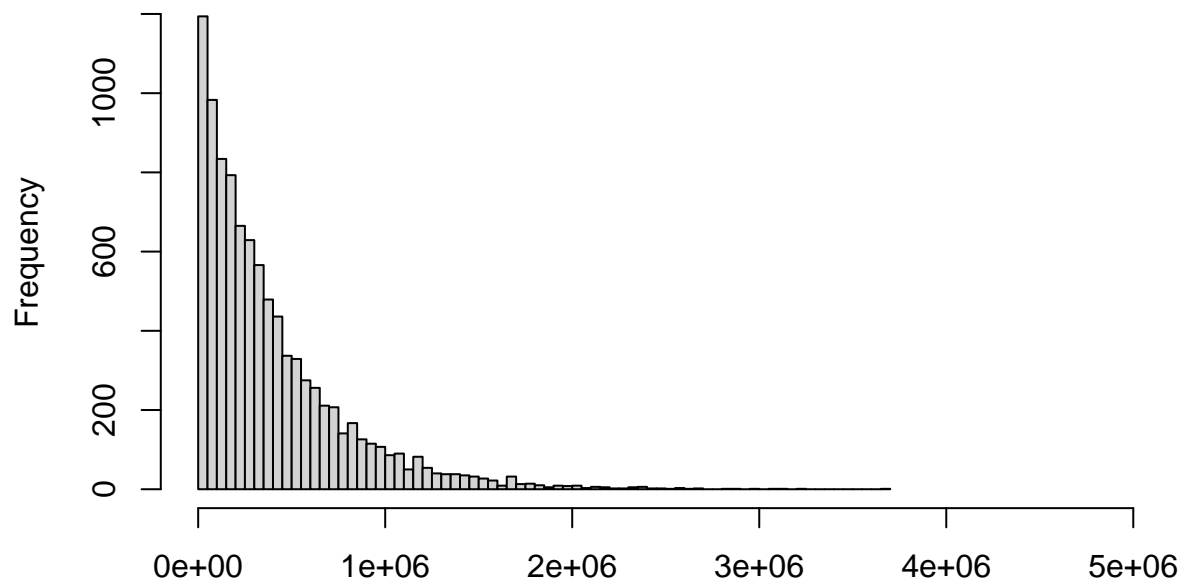
populacao\_dez\_simulacoes

**Pop: alp: 0.41 bet: 0.61 gam: 0.1 max: 3899832 avg: 398088**



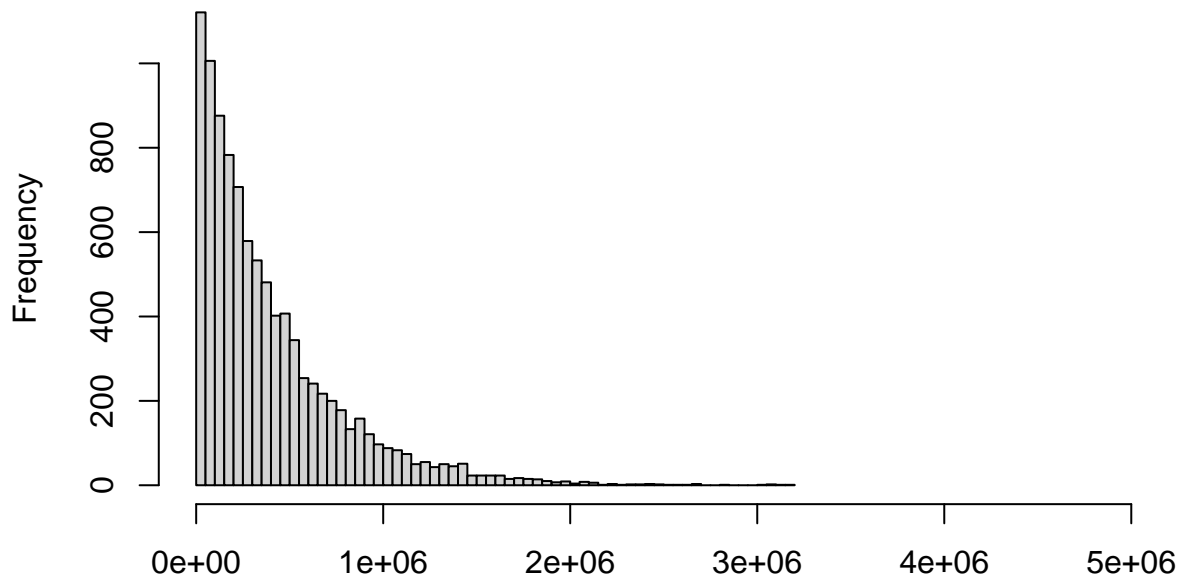
populacao\_dez\_simulacoes

**Pop: alp: 0.51 bet: 0.31 gam: 0.1 max: 3672094 avg: 392191**



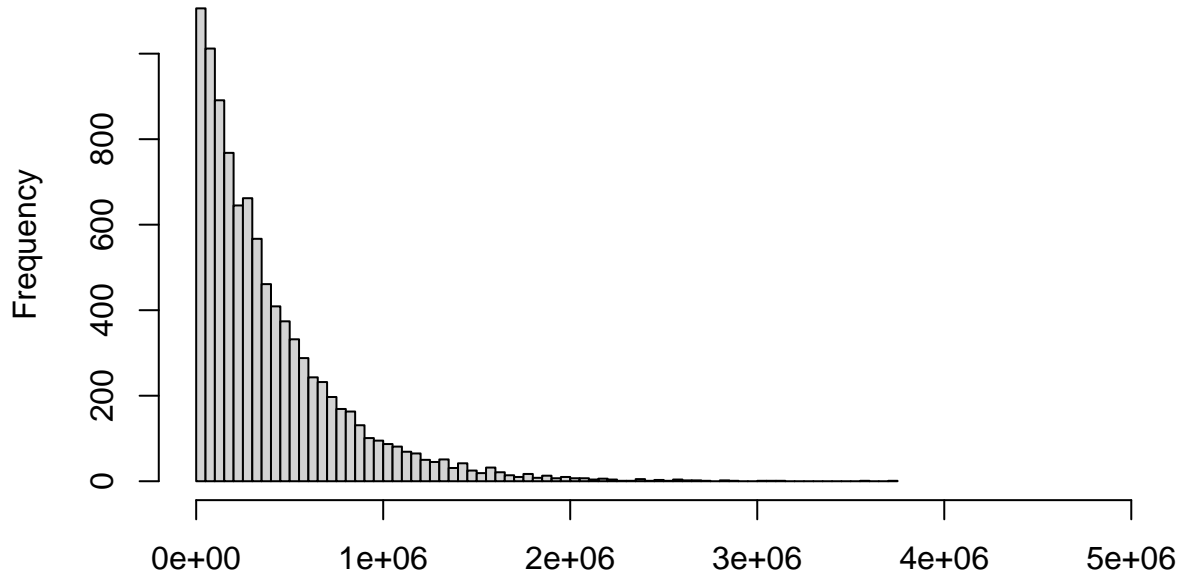
populacao\_dez\_simulacoes

**Pop: alp: 0.51 bet: 0.54 gam: 0.1 max: 3174483 avg: 395186**



populacao\_dez\_simulacoes

**Pop: alp: 0.51 bet: 0.61 gam: 0.1 max: 3710675 avg: 394317**



populacao\_dez\_simulacoes

A comparação visual entre os 27 gráficos mostra uma satisfatória homogeneidade nos dados, em todas as simulações, mas a combinação  $\alpha, \beta, \gamma = 0.41, 0.61, 0.02$  alcançou a maior população máxima, enquanto que  $\alpha, \beta, \gamma = 0.51, 0.54, 0.1$  alcançou a menor população..

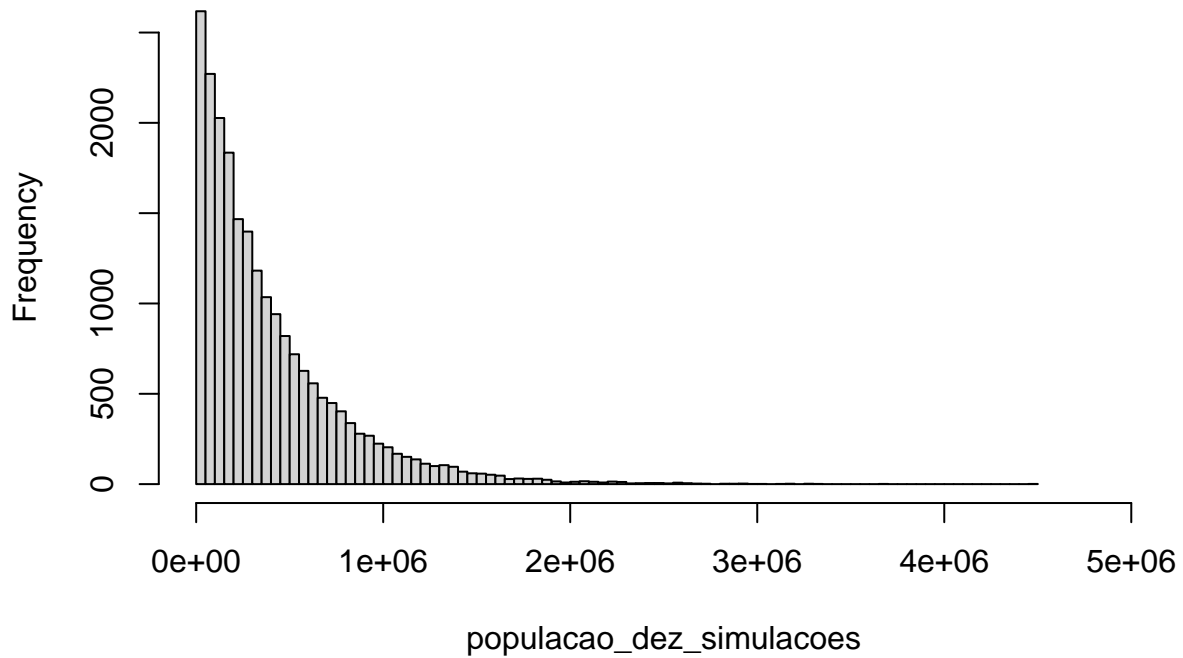
Agora vamos à comparação a variação dos demais parâmetros: `initial_trading_perc` e `trading_revigoration`, sabendo que para cada uma dessas combinações foram gerados  $27 * 80 = 2160$  registros.

```

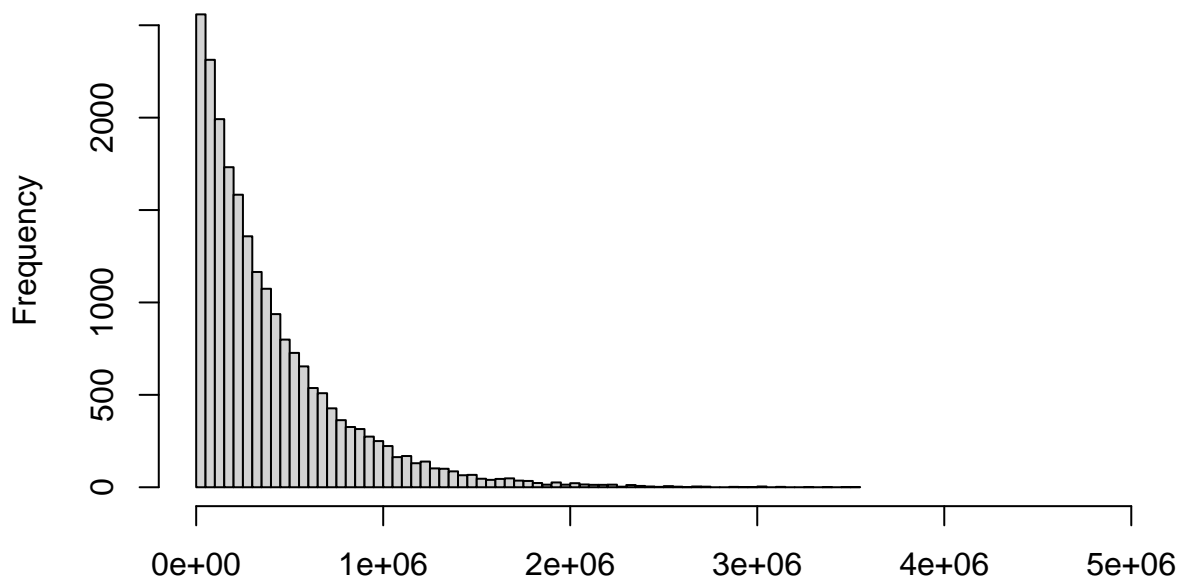
for (per_ in initial_trading_perc) {
for (rev_ in trading_revigoration) {
  populacao_dez_simulacoes <- agent_data_clean_2[
    agent_data_clean_2$initial_trading_perc==per_ &
    agent_data_clean_2$trading_revigoration==rev_,]$Population
  hist(populacao_dez_simulacoes,breaks=100,xlim=range(0,5000000),
    main = paste("Pop: % trade:",per_," revigoration:",rev_," max:",max(as.integer(populacao_dez,
  }
}

```

**Pop: % trade: 0.1 revigoration: 20 max: 4496278 avg: 390109**

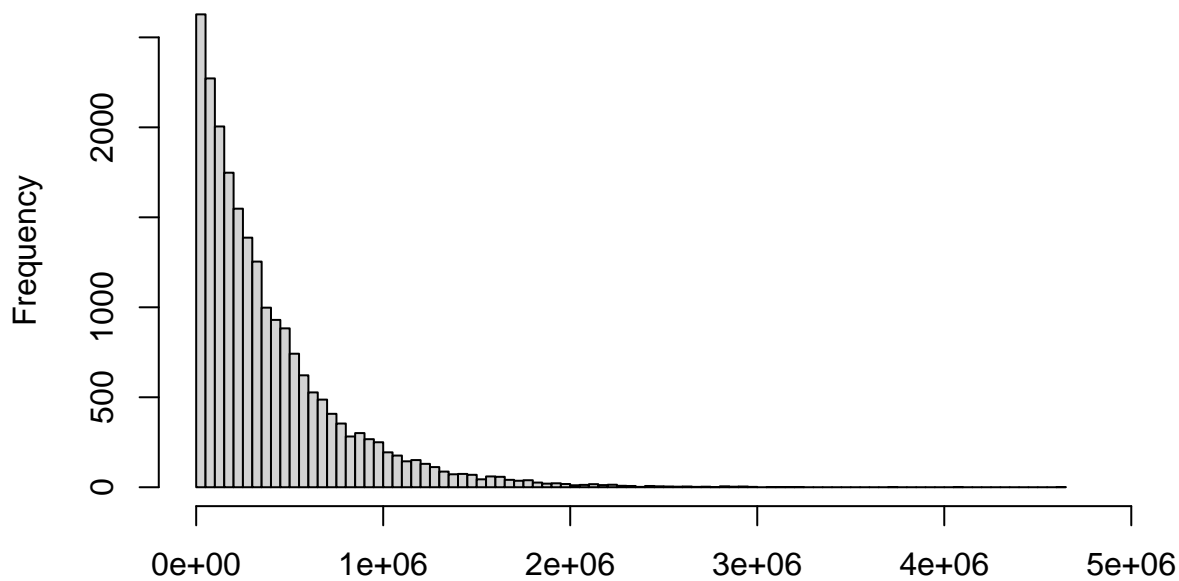


**Pop: % trade: 0.1 revigoratio: 40 max: 3517968 avg: 393469**



populacao\_dez\_simulacoes

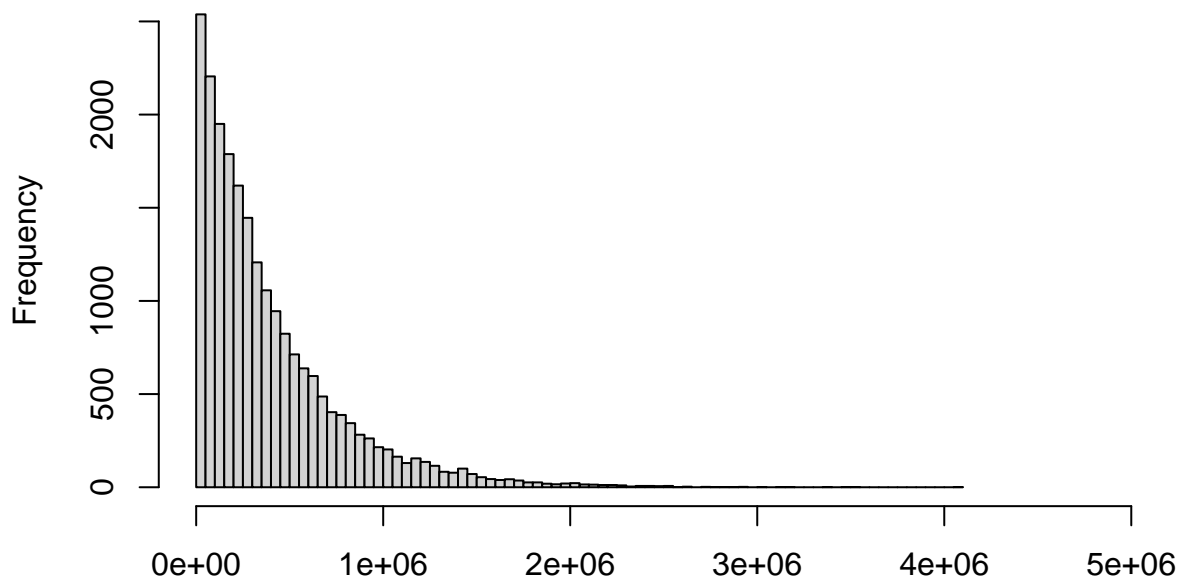
**Pop: % trade: 0.1 revigoratio: 80 max: 4647171 avg: 391561**



populacao\_dez\_simulacoes

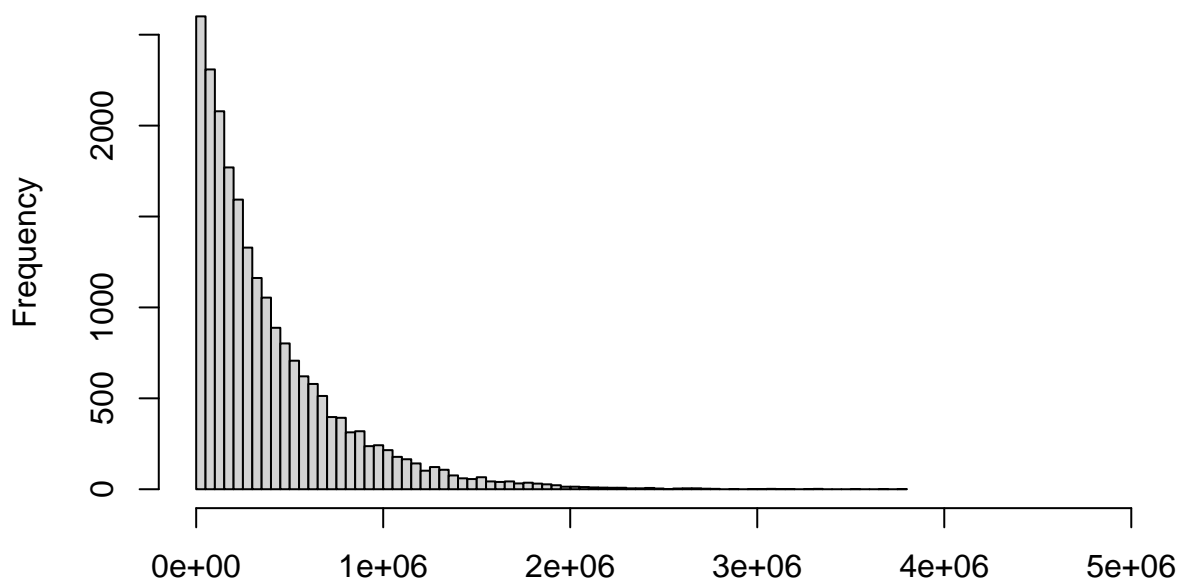


**Pop: % trade: 0.2 revigoratio: 20 max: 4061964 avg: 392834**



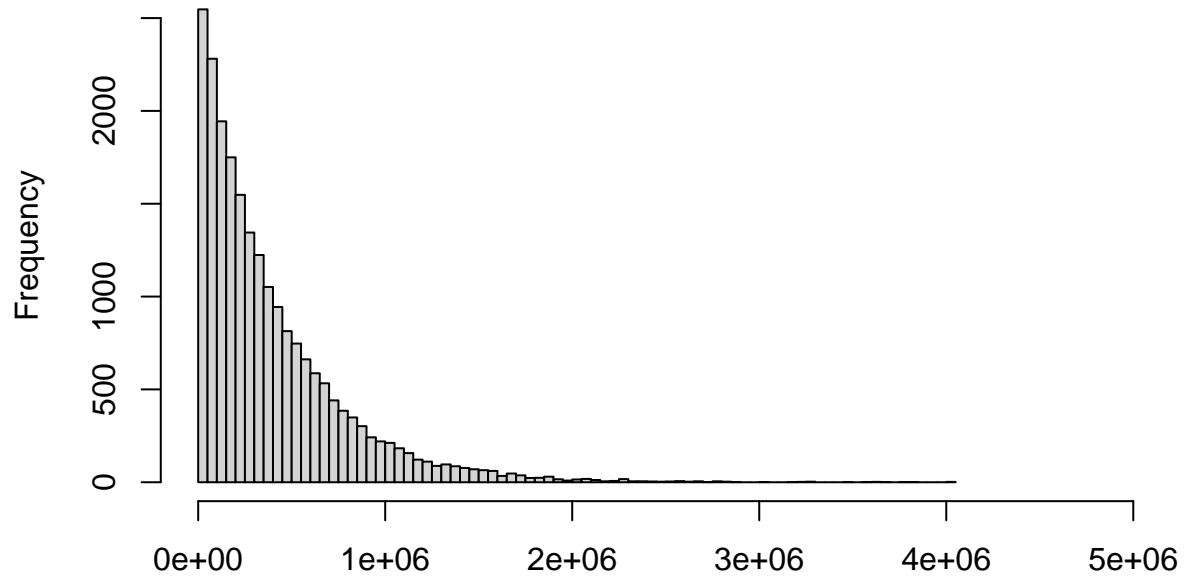
populacao\_dez\_simulacoes

**Pop: % trade: 0.2 revigoratio: 40 max: 3795664 avg: 389407**



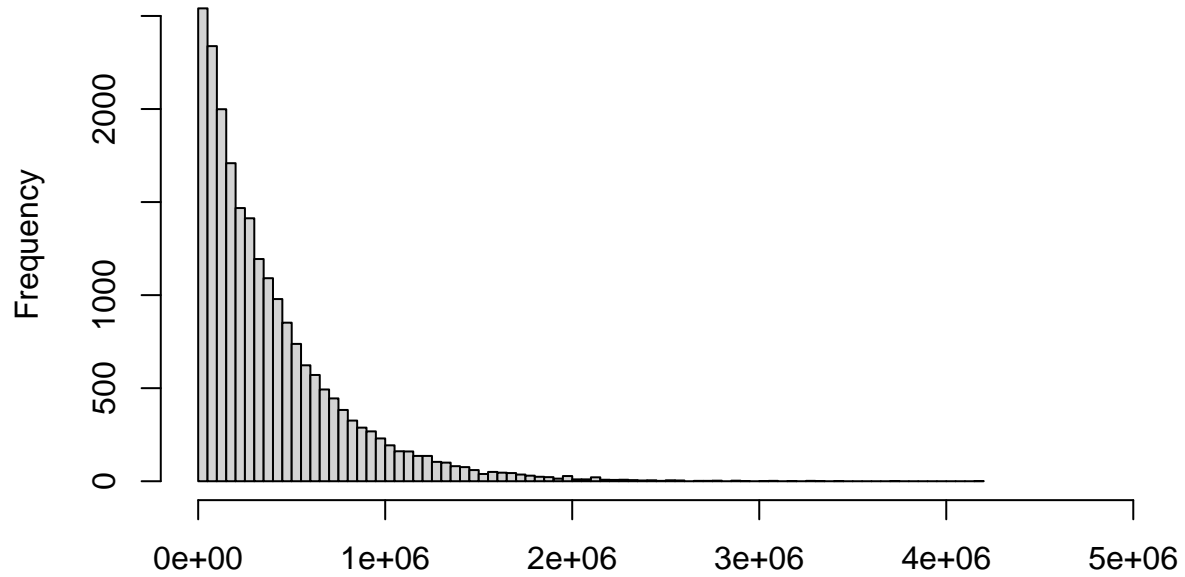
populacao\_dez\_simulacoes

**Pop: % trade: 0.2 revigoratio: 80 max: 4021148 avg: 395019**



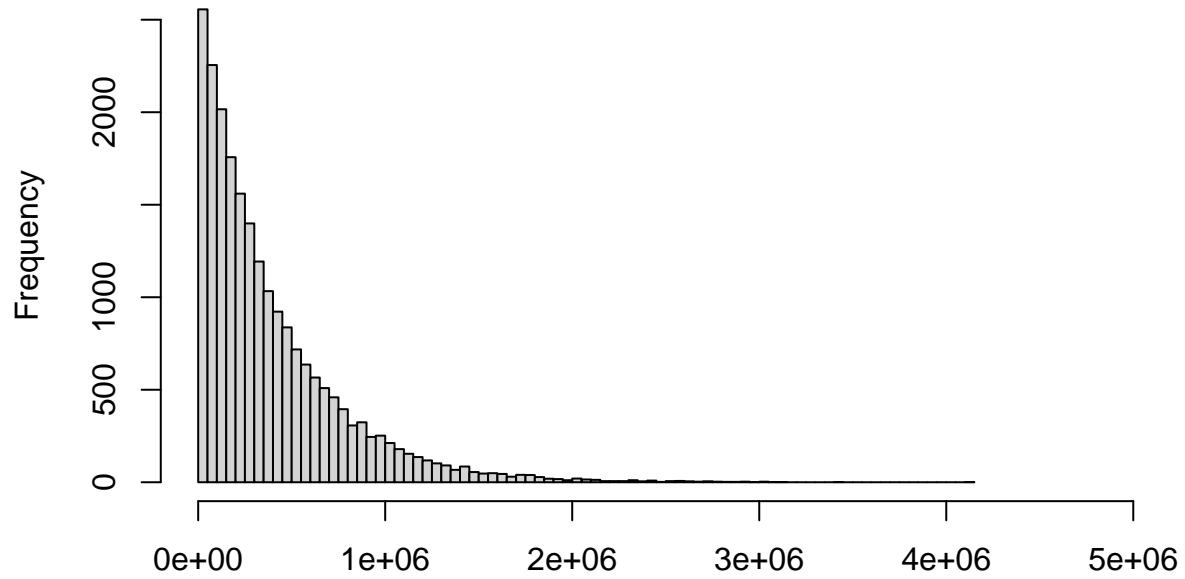
populacao\_dez\_simulacoes

**Pop: % trade: 0.4 revigoratio: 20 max: 4185818 avg: 391047**



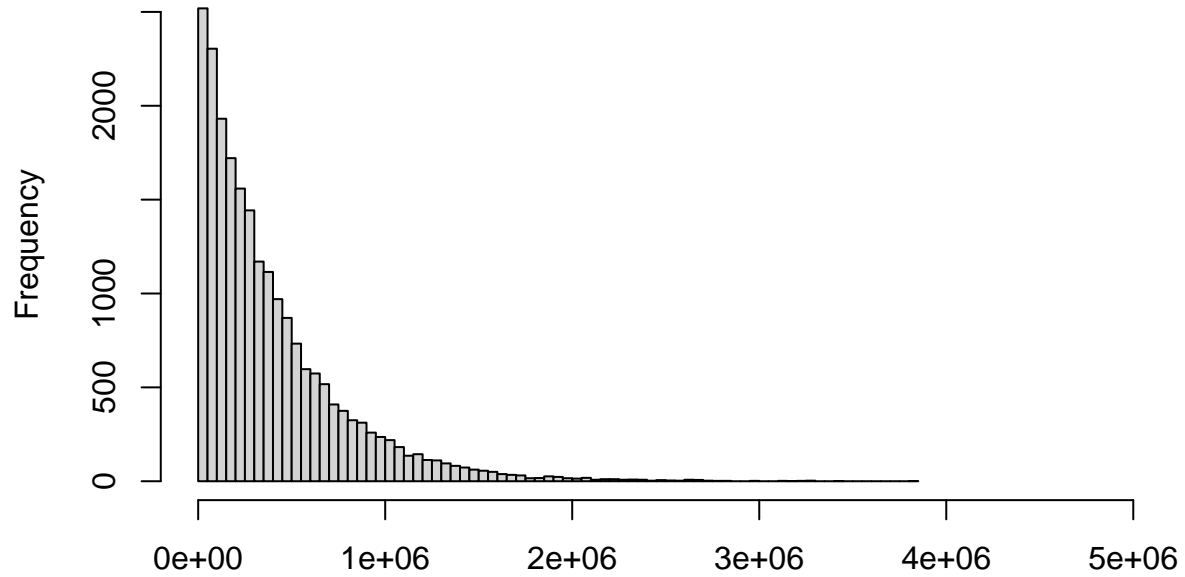
populacao\_dez\_simulacoes

**Pop: % trade: 0.4 revigoratio: 40 max: 4141333 avg: 392601**



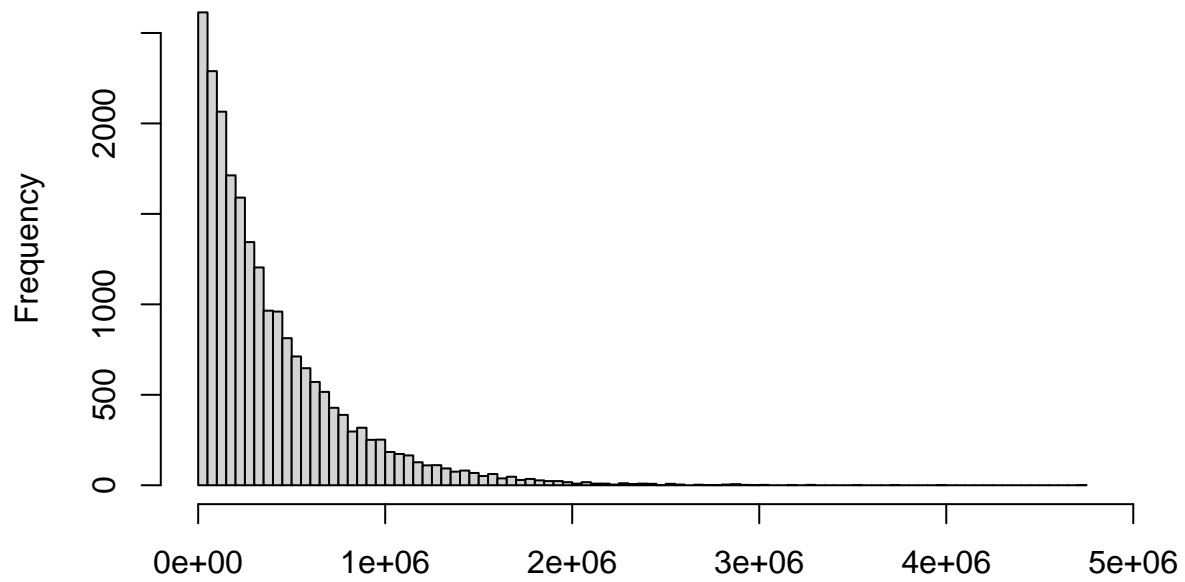
populacao\_dez\_simulacoes

**Pop: % trade: 0.4 revigoratio: 80 max: 3841940 avg: 392507**



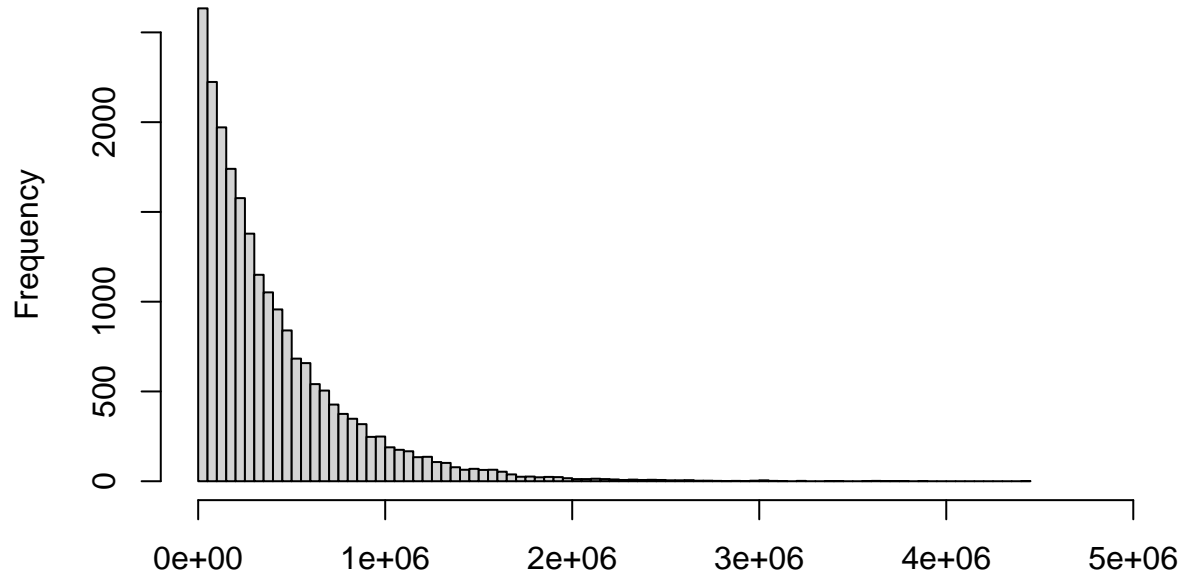
populacao\_dez\_simulacoes

**Pop: % trade: 0.8 revigoratio: 20 max: 4712368 avg: 391050**



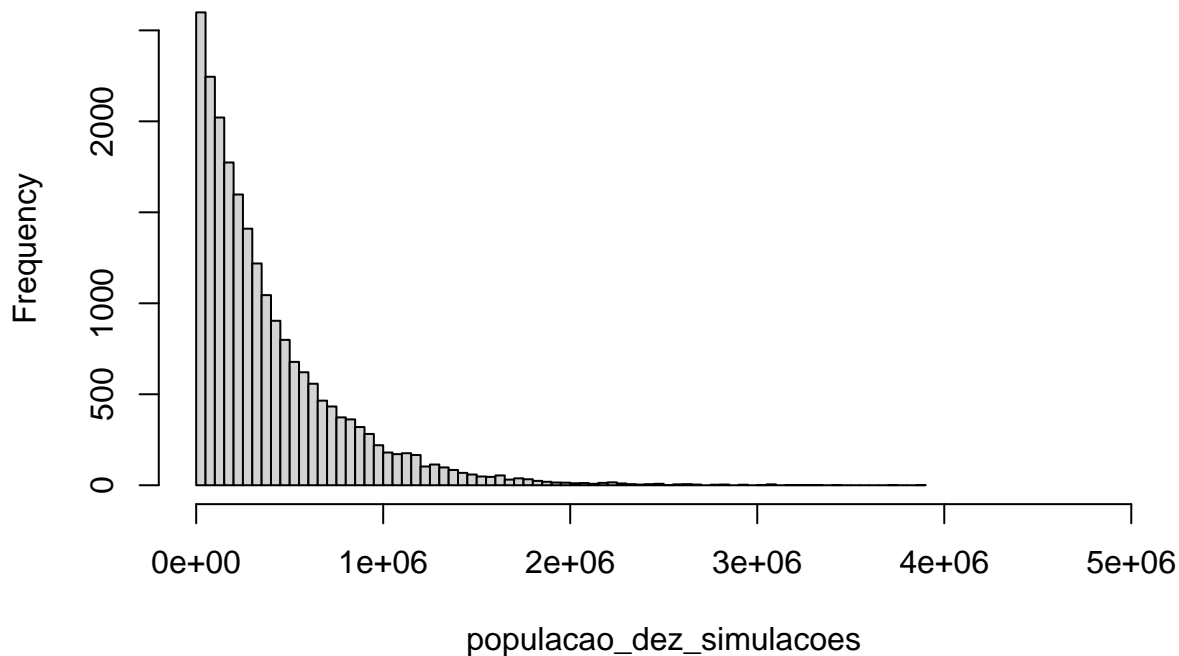
populacao\_dez\_simulacoes

**Pop: % trade: 0.8 revigoratio: 40 max: 4424574 avg: 396528**



populacao\_dez\_simulacoes

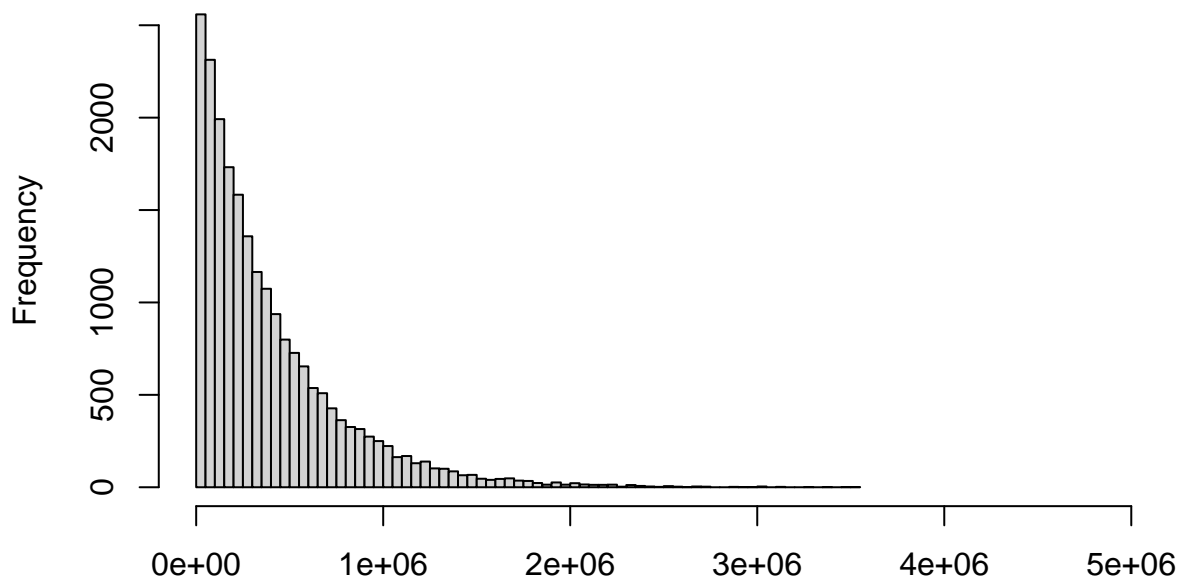
**Pop: % trade: 0.8 revigoratio: 80 max: 3899832 avg: 391115**



Os históricos das simulações acima todos apresentaram boa uniformidade na distribuição das populações, e nota-se que os valores de trading revigoration de 40 pontos produzem menores populações finais que os valores 20 e 80. Assim sendo, esses será usado como base para o aprofundamento da simulação. Colocando-se agora os valores de revigoration no laço interno, pode-se fazer uma análise da variabilidade gerada pelo percentual inicial de territórios que estão comercializando.

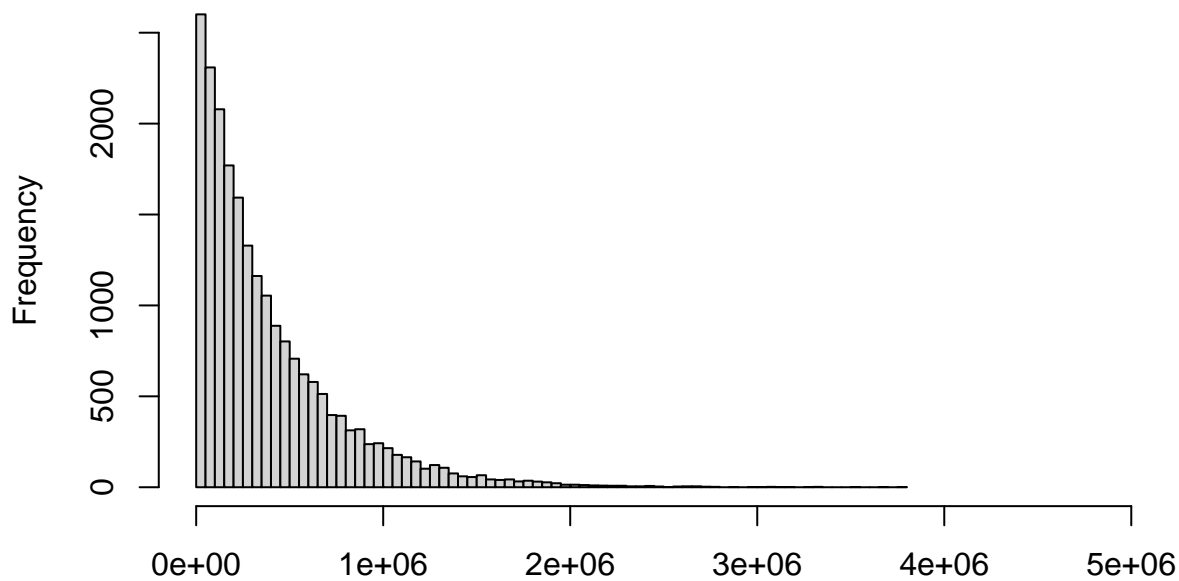
```
for (rev_ in c(40)) {  
  for (per_ in initial_trading_perc) {  
    populacao_dez_simulacoes <- agent_data_clean_2[  
      agent_data_clean_2$initial_trading_perc==per_ &  
        agent_data_clean_2$trading_revigoration==rev_,]$Population  
    hist(populacao_dez_simulacoes,breaks=100,xlim=range(0,5000000),  
        main = paste("Pop: % trade:",per_," revigoratio:",rev_," max:",max(as.integer(populacao_dez_
```

**Pop: % trade: 0.1 revigoratio: 40 max: 3517968 avg: 393469**



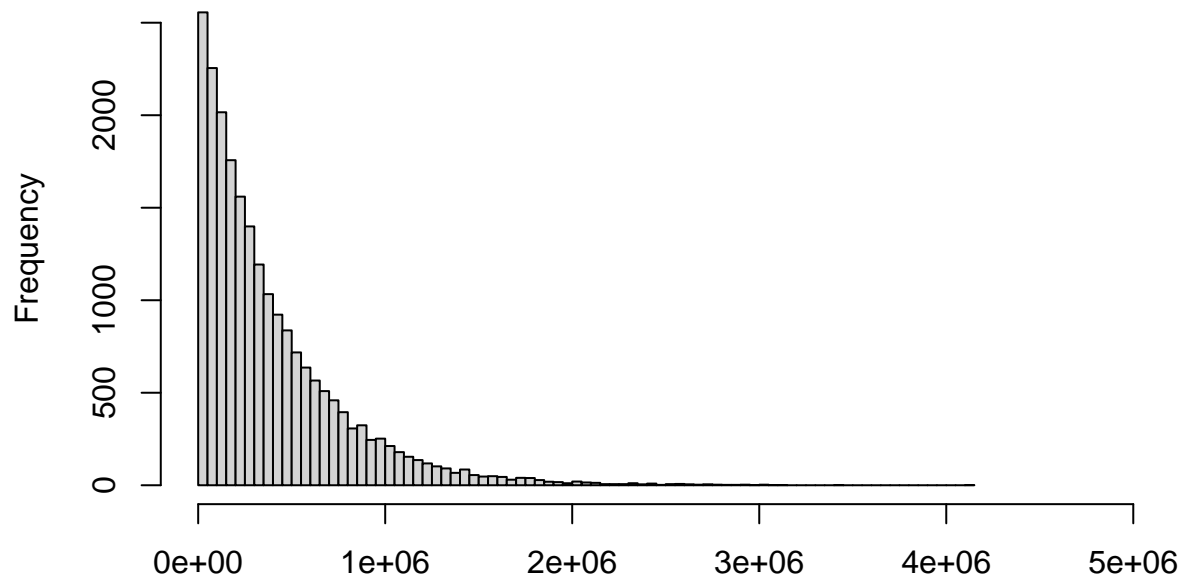
populacao\_dez\_simulacoes

**Pop: % trade: 0.2 revigoratio: 40 max: 3795664 avg: 389407**



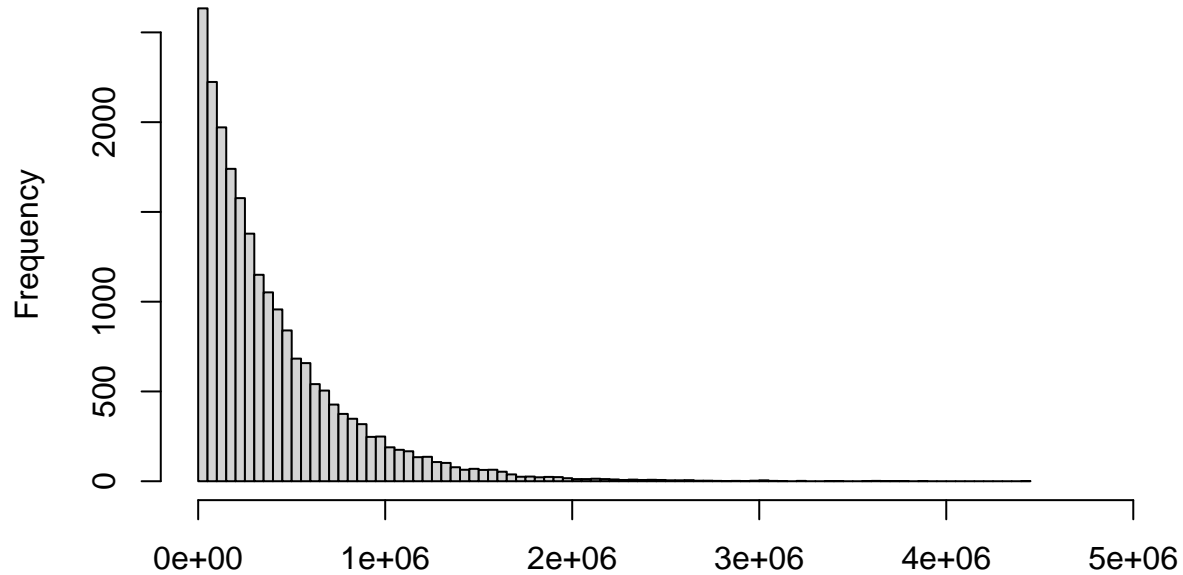
populacao\_dez\_simulacoes

**Pop: % trade: 0.4 revigatorio: 40 max: 4141333 avg: 392601**



populacao\_dez\_simulacoes

**Pop: % trade: 0.8 revigatorio: 40 max: 4424574 avg: 396528**



populacao\_dez\_simulacoes

O maior resultado alcançado foi para o percentual inicial de 80% dos territórios comercializando.

Volta-se agora a uma nova simulação, dessa vez com cem anos completos, dada a escolha dos seguintes valores fixados:

num\_nodes = 80 alpha,beta,gamma = 0.41,0.61,0.02 initial\_trading\_perc = 0.8 trading\_revigoration = 40

O processo de investigação continua.