**Utilizando o R para fazer Simulação – Informações Básicas**

**Adriana Backx Noronha Viana**

**César Alexandre de Souza**

Desenvolvendo Simulação no R

*1) Utilizar loop*

for (i in 1:XXXX) { #abri loop

} #fecha loop

Exemplo:

x <- 0

for (i in 1:1000) {

x <- x + 1

}

> x

[1] 1000

*2) Definir variável utilizando amostragem*

Função sample()

portas <- c(“A”, “B”, “C”)

amostra <- sample(portas)

amostra #retorna vetor

amostra[1] #retorna o primeiro valor do vetor

premio <- sample(portas) [1]

escolha <- sample(portas) [1]

3) Comparação e escolhas

premio <- sample(portas) [1]

escolha <- sample(portas) [1]

portas != premio #retorna TRUE ou FALSE (função diferente)

portas != escolha

*Função which: retorna o índice no vetor do que se está procurando*

LETTERS

which(LETTERS == "R")

Para o problema em si:

which(portas != escolha)

which(portas != premio)

Para que sejam consideradas ambas as condições ao mesmo tempo, utilizamos o símbolo &

which(portas != escolha & portas != premio)

Um pouco mais sobre a função which

m <- matrix(1:12, 3, 4)

div.3 <- m %% 3 == 0

which(div.3)

which(div.3, arr.ind = TRUE)

O apresentador do show vai abrir a porta que **não têm o prêmio** e **não foi escolhida**

abrir <- portas[which(portas != escolha & portas != premio)][1]

Observar que quando as portas forem iguais, sempre vai escolher o primeiro pela ordem. Para evitar isso, pode-se utilizar sample nesse caso também.

abrir <- sample(portas[which(portas != escolha & portas != premio)])[1]

*Função if: condicional*

m <- matrix(1:12, 3, 4)

if(m[3,1] %%3==0) {m[3,1]=0}

Para o exemplo:

xdata = c()

if(premio==escolha){xdata=c(xdata,"nãotrocouvence")}

**Problema de Monty Hall Completo**

portas <- c("A","B","C")

xdata = c()

for (i in 1:1000)

{

premio <- sample(portas) [1]

escolha <- sample(portas) [1]

abrir <- sample(portas[which(portas != escolha & portas != premio)])[1]

troca <- portas[which(portas != escolha & portas != abrir)]

if(escolha==premio){xdata=c(xdata,"naotrocouvence")}

if(troca==premio){xdata=c(xdata,"trocouvence")}

}

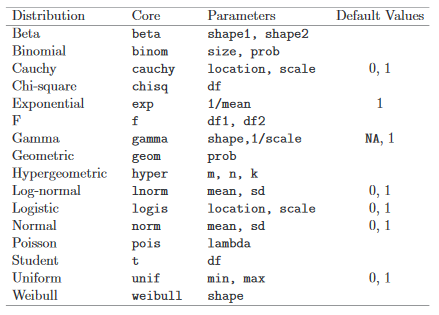
length(which(xdata=="trocouvence"))

length(which(xdata=="naotrocouvence"))

Referência:

<http://bodowinter.com/tutorial/bw_doodling_monty_hall.pdf>

**Distribuições de Probabilidade no R**



Fonte: <http://www.stat.ufl.edu/archived/casella/ShortCourse/MCMC-UseR.pdf>

**Exemplo – Utilização R para Simulação**

Uma empresa que vende produto X sob um mercado puro/perfeitamente competitivo quer saber a distribuição de probabilidade dos lucros deste produto e a probabilidade que a empresa perca dinheiro quando comercializá-lo.

A equação para o lucro é: Lucro = Receita - Custo = (Q\*P) - (Q\*CV+CF)

Hipóteses:

A Quantidade Demandada (Q) flutua entre 8.000 e 12.000 unidades e está distribuída uniformemente.

O Custo Variável (CV) está normalmente distribuído (com média = 7, desvio-padrão = 2); (o custo deverá ser no mínimo 1).

O Preço de Mercado (P) está normalmente distribuído (com média = 10, desvio-padrão = 3); (o preço de mercado deverá ser no mínimo 1).

O Custo Fixo (CF) é R$5.000.

Apresentar:

1) a distribuição de probabilidade dos lucros deste produto e

2) a probabilidade que a empresa perca dinheiro quando comercializá-lo.

Considere uma simulação com 500 iterações.

***Entrada no R***

set.seed(1) #definição de semente

Q = runif(500,8000,12000) #distribuição uniforme: tamanho amostra, mínimo, máximo

hist(Q,nclass=10) #histograma: variável, número de classes~

CV = rnorm(500,7,2)

CV = ifelse(CV<=1,1,CV)

P = rnorm(500,10,2)

P = ifelse(P<=1,1,P)

CF = 5000

L = Q\*P - (Q\*CV+CF)

hist(L,nclass=10)

contagem = ifelse(L<0,1,0)

sum(contagem)

prob\_negativo = sum(contagem)/length(L)

*Modificação no Exemplo:*

Parte 1: Fazer o mesmo exemplo, mas considere uma simulação com no mínimo 1000 iterações. Comparar resultados

**R:** Com uma amostra de 500 observações a probabilidade é igual a 0.214. Com uma amostra de 1000 observações a probabilidade é de 0.198.

Parte 2: Modificar valores do CV e Preço:

O Custo Variável (CV) está normalmente distribuído (com média = 7, desvio-padrão = 2); (o custo deverá ser no mínimo 1 e no máximo 10).

O Preço de Mercado (P) está normalmente distribuído (com média = 10, desvio-padrão = 3); (o preço de mercado não pode ser menor que 3,50).

Considere 1000 iterações.

**R:** A probabilidade é igual a 0.187.

**Script**

set.seed(1) #definição de semente

Q = runif(1000,8000,12000) #distribuição uniforme: tamanho amostra, mínimo, máximo

hist(Q,nclass=10) #histograma: variável, número de classes~

CV = rnorm(1000,7,2)

CV = ifelse(CV<=1,1,CV)

CV = ifelse(CV>=10,10,CV)

hist(CV,nclass=10)

P = rnorm(1000,10,2)

P = ifelse(P<=3.5,3.5,P)

hist(P,nclass=10)

CF = 5000

L = Q\*P - (Q\*CV+CF)

hist(L,nclass=10)

contagem = ifelse(L<0,1,0)

sum(contagem)

prob\_negativo = sum(contagem)/length(L)

prob\_negativo