

Geração de Números Aleatórios

Jaqueline Lamas da Silva

10/05/2023

Sumário

Questão 1:	1
Questão 2:	4
Questão 3:	7
Questão 4:	9
Letra a:	9
Letra b:	9
Letra c:	9
Letra d:	9
Letra e:	11
Letra f:	12
Letra g:	13

Questão 1:

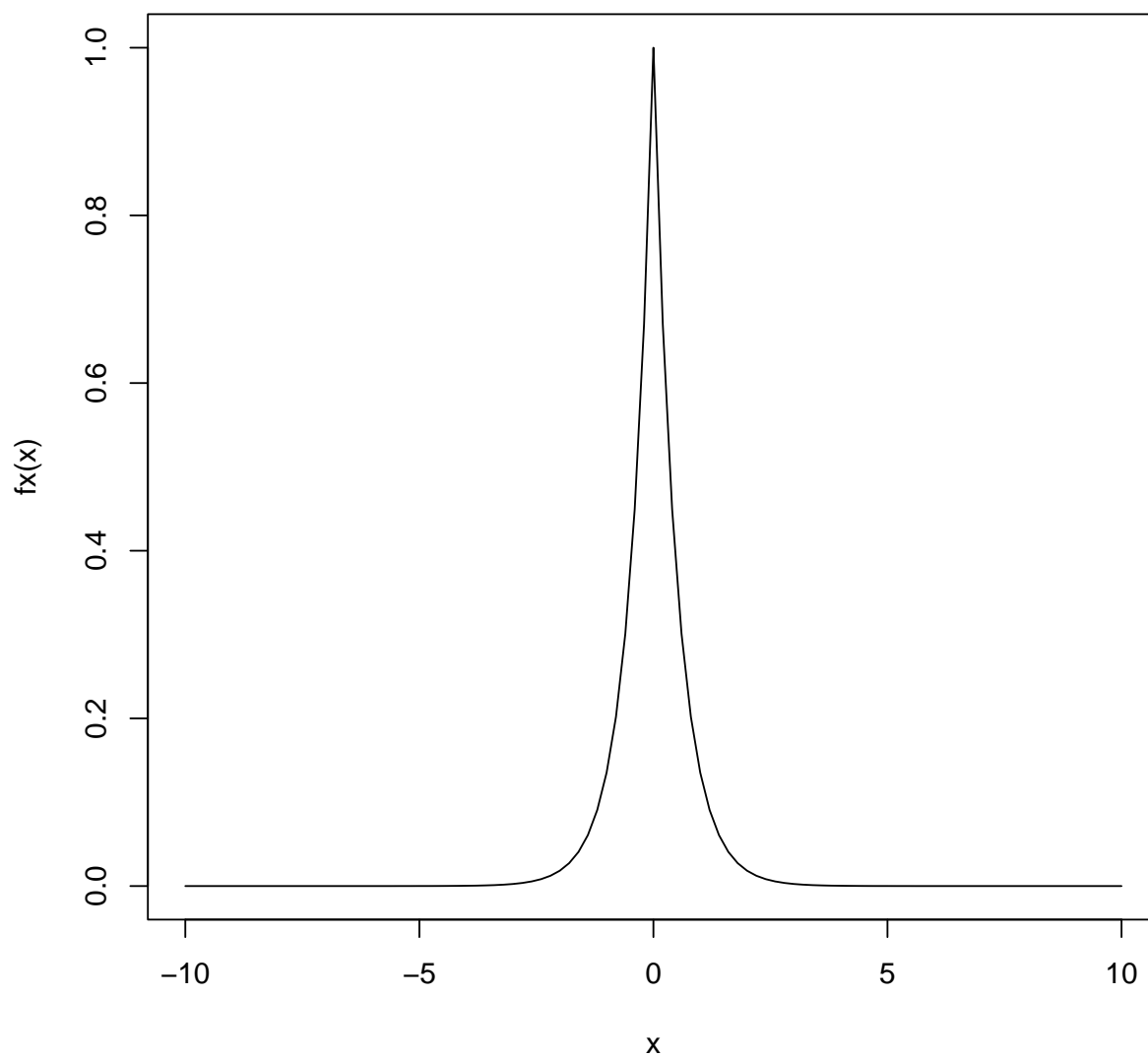
Apresente o código de um método para gerar uma variável aleatória com a seguinte função de densidade de probabilidade:

$$f_X(x) = e^{-2|x|}, x \in \mathbb{R}$$

```
# Densidade  
fx<-function(x) exp(-2*abs(x))  
fx(-10)
```

```
## [1] 2.061154e-09
```

```
curve(fx(x), from=-10, to=10)
```



```
Fx<-function(x)
{
  integrate(fx, lower = -10, upper = x)$value
}
```

```
Fx<-Vectorize(Fx)
Fx(1.891281e-06)
```

```
## [1] 0.5000019
```

```
Fx(1)
```

```
## [1] 0.932332
```

```
Fx(-7)
```

```
## [1] 4.147338e-07
```

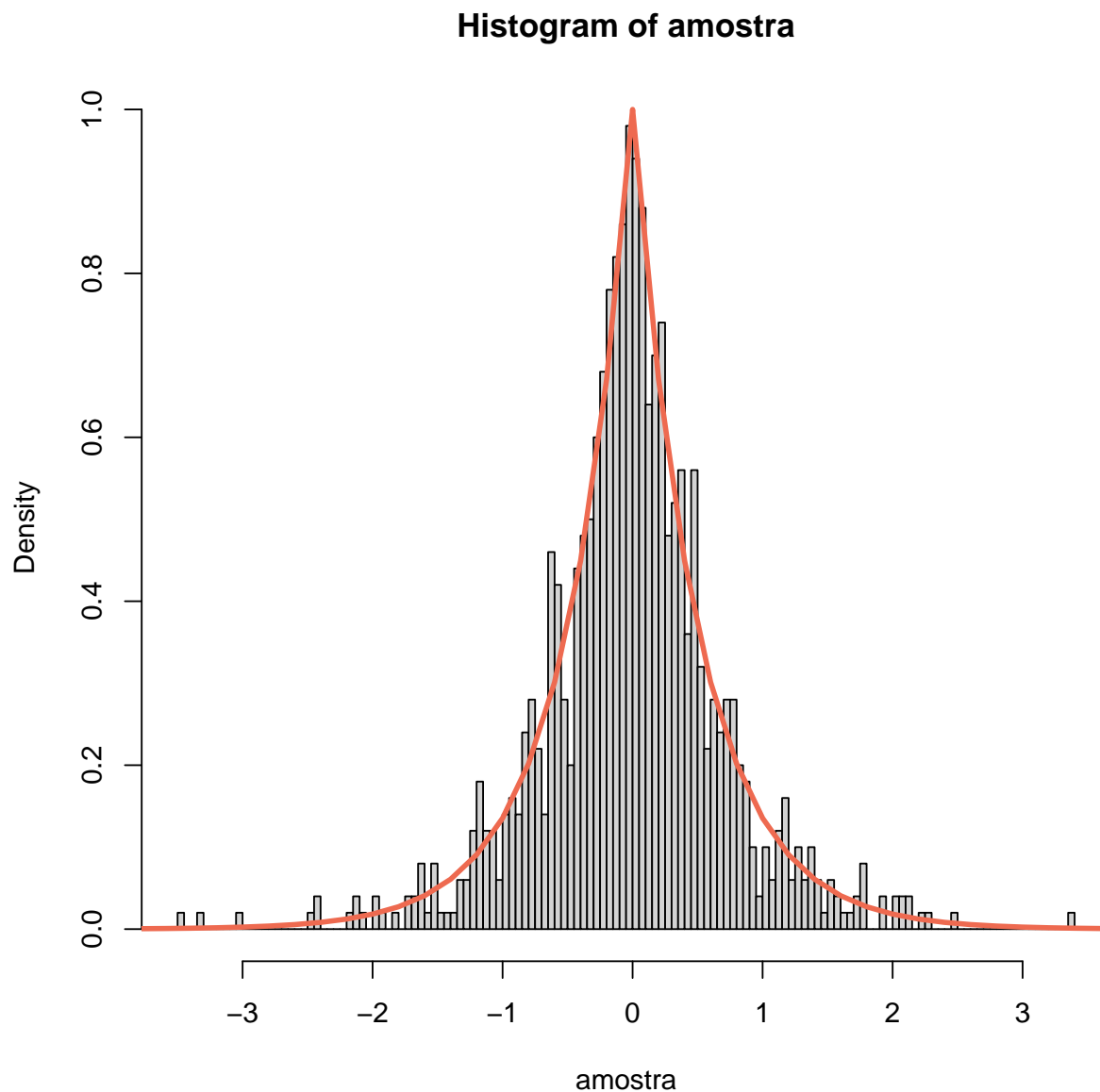
```
Qxi<-function(x,prob)
{
  (Fx(x)-prob)^2
}

Qx<-function(p)
{
  optimize(Qxi,prob=p, maximum = F, lower = -10, upper = 10)$minimum
}
Qx<-Vectorize(Qx)

amostra<-Qx(runif(1000))
ks.test(amostra,Fx)
```

```
##
## Asymptotic one-sample Kolmogorov-Smirnov test
##
## data: amostra
## D = 0.014058, p-value = 0.989
## alternative hypothesis: two-sided
```

```
hist(amostra, breaks=100, freq = F)
curve(fx(x), from=-10, to=10, add=T, col="coral2", lwd=3)
```



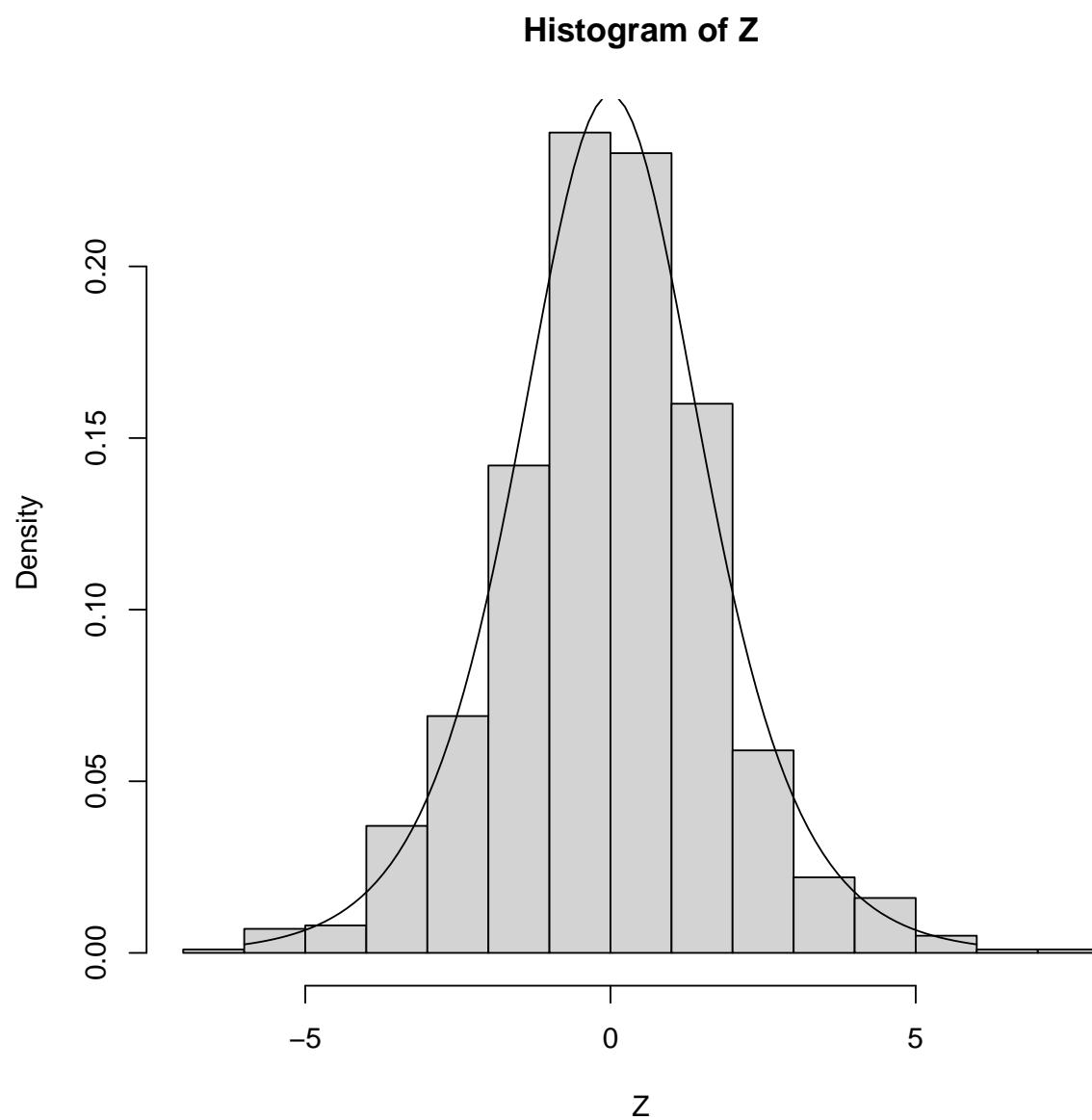
Questão 2:

Logistic(μ, β)

pdf $f(x|\mu, \beta) = \frac{1}{\beta} \frac{e^{-(x-\mu)/\beta}}{[1+e^{-(x-\mu)/\beta}]^2}, \quad -\infty < x < \infty, \quad -\infty < \mu < \infty, \quad \beta > 0$

mean and
variance $EX = \mu, \quad \text{Var } X = \frac{\pi^2 \beta^2}{3}$

```
u<-runif(1000)
Z<-log(u/(1-u))
hist(Z, freq = F)
curve(dlogis(x), from=-6, to=6, add = T)
```



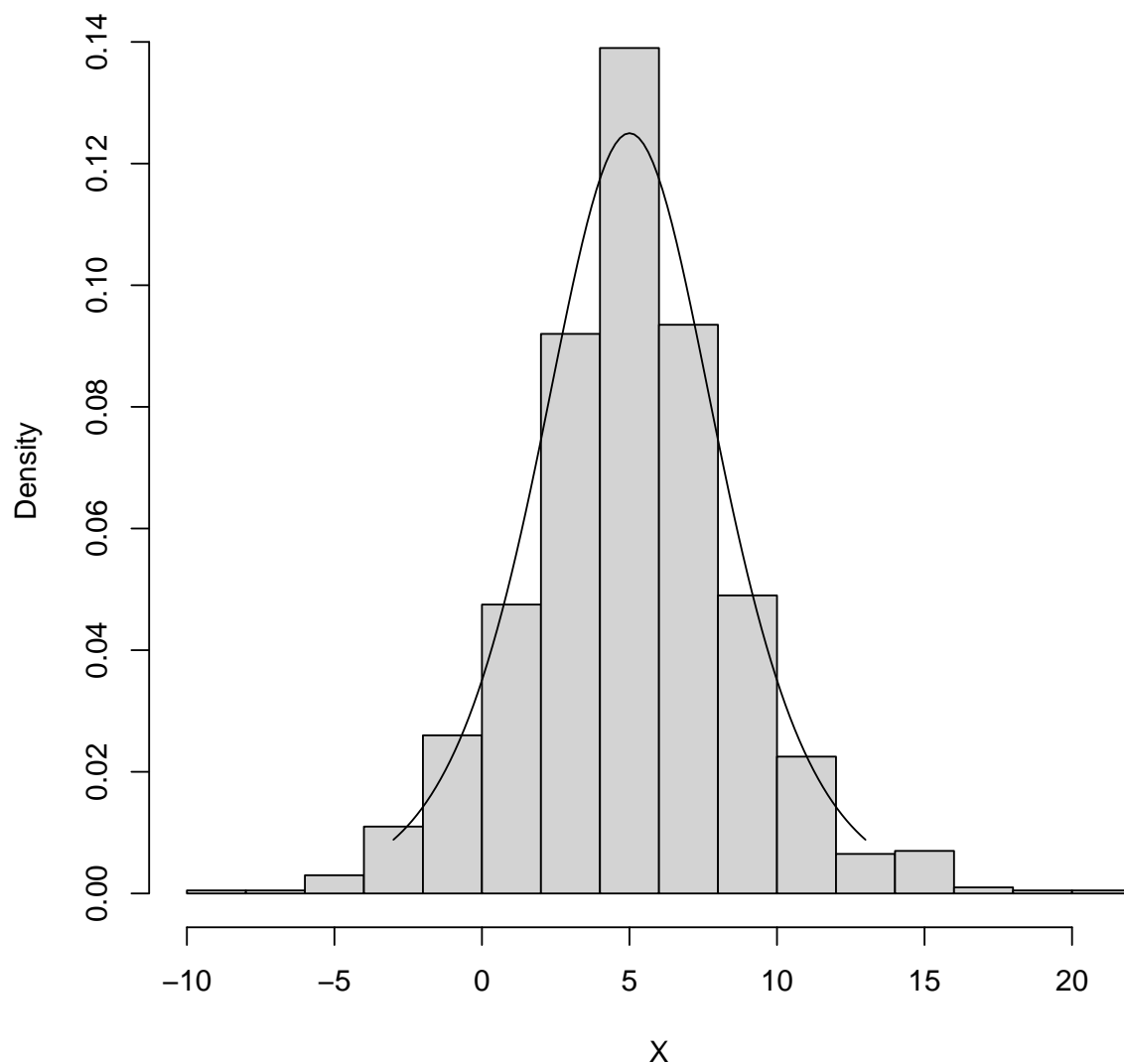
```
ks.test(Z, "plogis")
```

```
##
## Asymptotic one-sample Kolmogorov-Smirnov test
##
```

```
## data: Z  
## D = 0.021367, p-value = 0.7512  
## alternative hypothesis: two-sided
```

```
# A distribuição logística pertence a família locação-escala  
mu<-5  
sigma<-2  
X<-sigma*Z+5  
hist(X, freq=F)  
curve(dlogis(x, location = mu, scale = sigma), from=mu-4*sigma, to=mu+4*sigma, add = T)
```

Histogram of X



```
ks.test(X, "plogis") # Rejeitou, a X não tem locacao 0 e scala 1
```

```
##  
## Asymptotic one-sample Kolmogorov-Smirnov test  
##  
## data: X  
## D = 0.70905, p-value < 2.2e-16  
## alternative hypothesis: two-sided
```

```
ks.test(X, "plogis", location=mu, scale=sigma)
```

```
##  
## Asymptotic one-sample Kolmogorov-Smirnov test  
##  
## data: X  
## D = 0.021367, p-value = 0.7512  
## alternative hypothesis: two-sided
```

Questão 3:

```
alfa<-c(0.8, 0.87, 0.92, 0.95)  
  
GerarU.alfa<-function(alfa)  
{  
  u<-runif(1)  
  while(u>=alfa)  
  {  
    u<-runif(1)  
  }  
  return(u)  
}  
  
u<-runif(1000)  
par(mfrow=c(4,2))  
for(i in seq_along(alfa))  
{  
  amostra<-replicate(1000, GerarU.alfa(alfa[i]))  
  hist(amostra, freq=F, breaks = 20, xlim=c(0,1),  
       main=paste0("Histograma da amostra (gerador 1), alfa = ", alfa[i]))
```

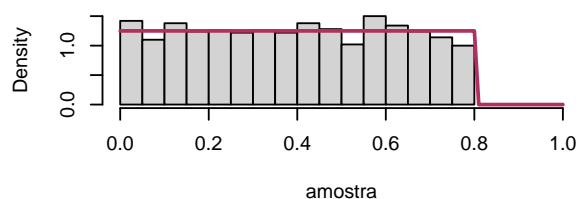
```

curve(dunif(x,max=alfa[i]),from = 0, to=1, add=T, col="maroon", lwd=2)

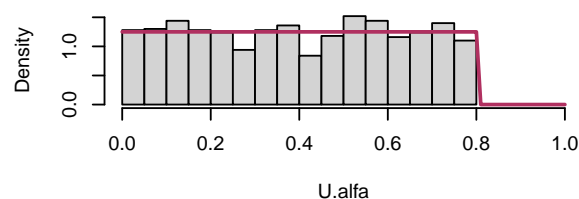
U.alfa<-alfa[i]*u
hist(U.alfa, freq=F, breaks = 20 ,xlim=c(0,1),
     main=paste0("Histograma da amostra (gerador 2), alfa = ", alfa[i]))
curve(dunif(x,max=alfa[i]),from = 0, to=1, add=T, col="maroon", lwd=2)
}

```

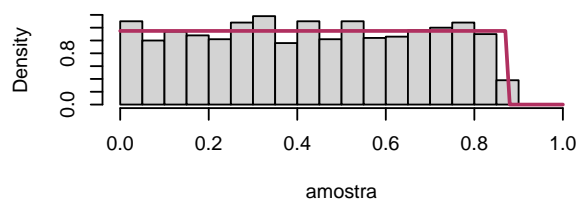
Histograma da amostra (gerador 1), alfa = 0.8



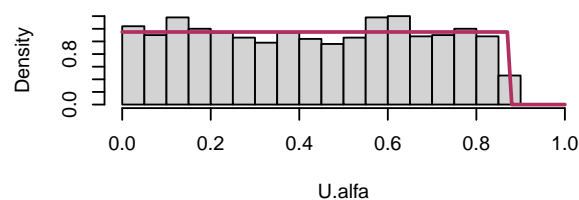
Histograma da amostra (gerador 2), alfa = 0.8



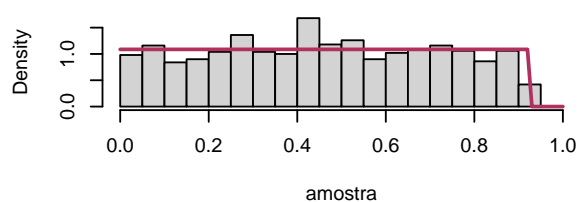
Histograma da amostra (gerador 1), alfa = 0.87



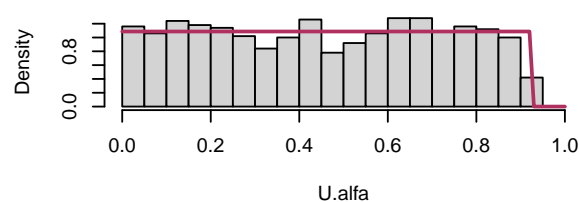
Histograma da amostra (gerador 2), alfa = 0.87



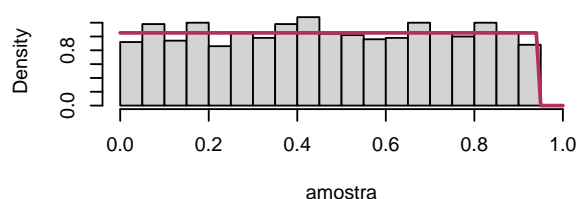
Histograma da amostra (gerador 1), alfa = 0.92



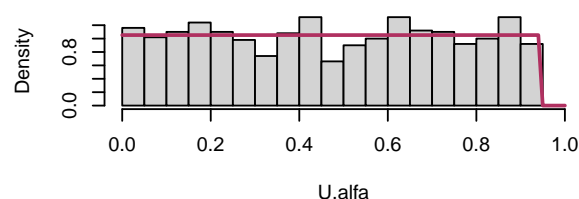
Histograma da amostra (gerador 2), alfa = 0.92



Histograma da amostra (gerador 1), alfa = 0.95



Histograma da amostra (gerador 2), alfa = 0.95



Comparando pelos gráficos as duas formas de gerar a distribuição $U(0, \alpha)$ parecem equivalentes. Não observei nenhuma melhoria de um método para o outro.

Questão 4:

Letra a:

```
geradorCM<-function(m=((2^13)-1), A=17, seed=666,n) # congruencial multiplicativo
{
  u<-numeric(n)
  anterior<-seed
  for(i in 1:n)
  {
    x<-(anterior*A)%m
    u[i]<-x
    anterior<-x
  }
  return(u)
}
```

Letra b:

```
X<-geradorCM(n=500)
```

Letra c:

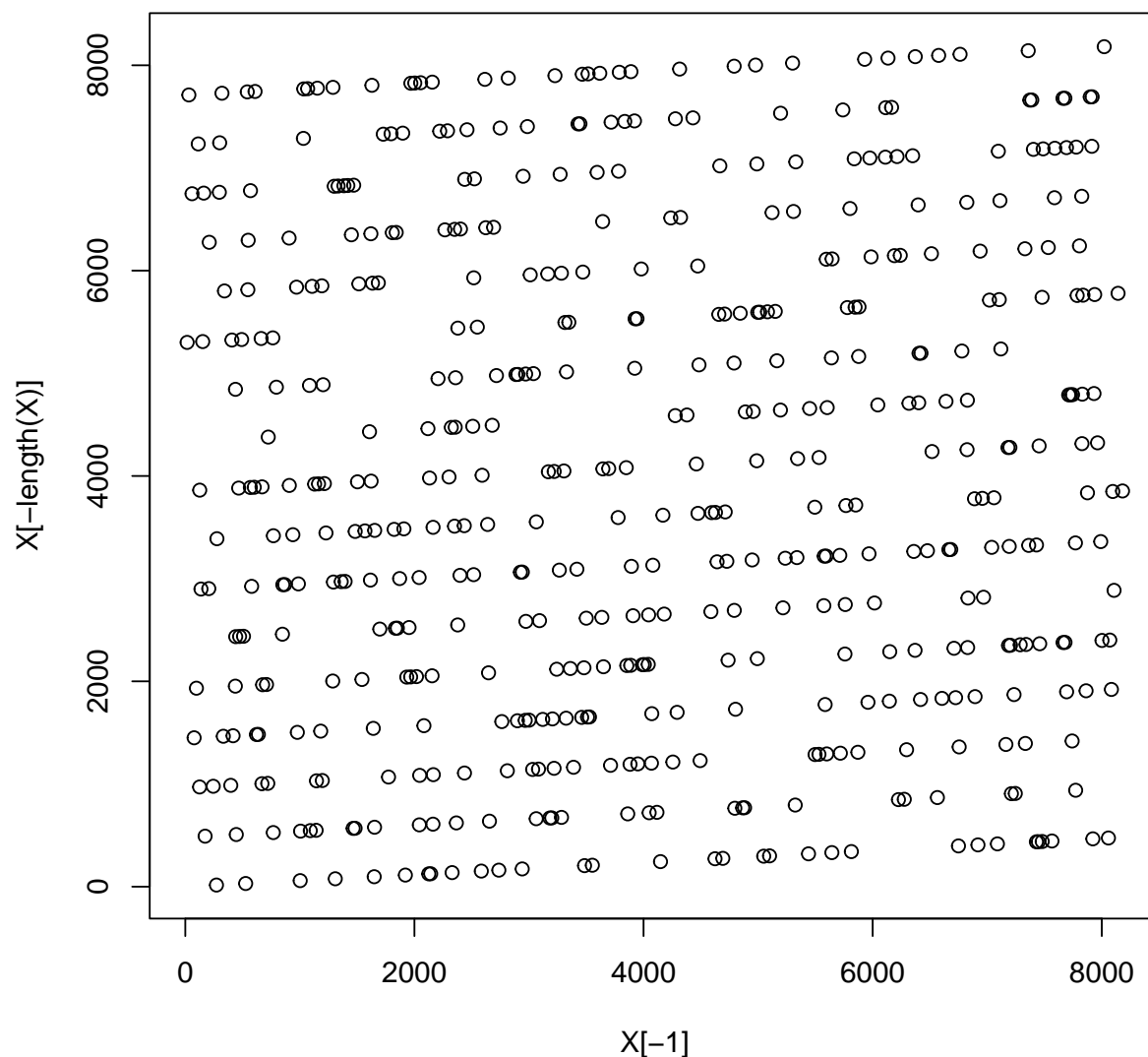
```
cor(X[-1],X[-length(X)])
```

```
## [1] 0.05589453
```

Os pares x_i e x_{i+1} possuem uma correlação baixa.

Letra d:

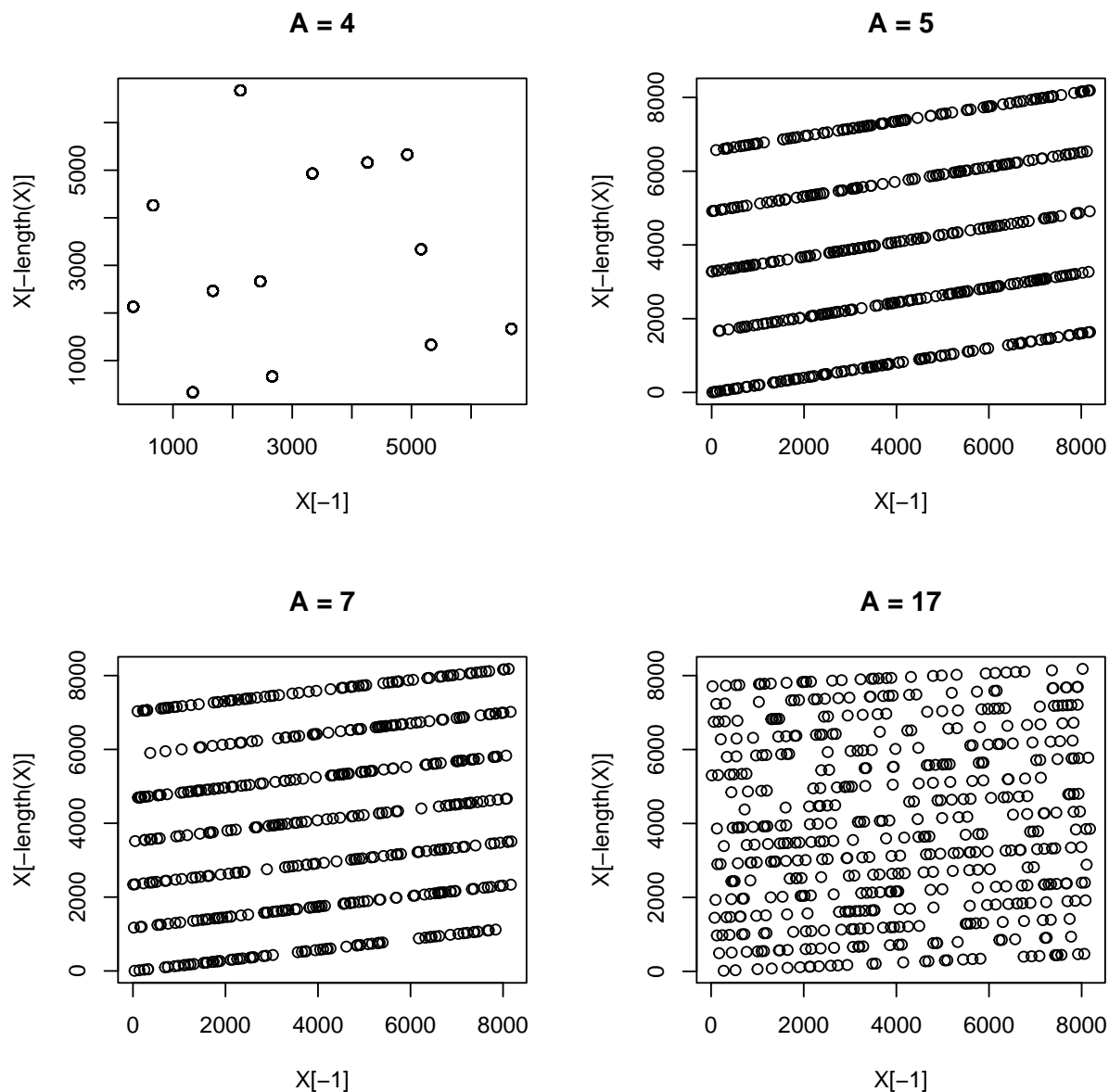
```
plot(X[-1],X[-length(X)])
```



Os pontos estão situados em diversas linhas.

Variando o valor de A:

```
a<-c(4,5,7,17)
par(mfrow=c(2,2))
for(i in seq_along(a))
{
  X<-geradorCM(n=500, A=a[i])
  plot(X[-1],X[-length(X)], main=paste0("A = ", a[i]))
}
```



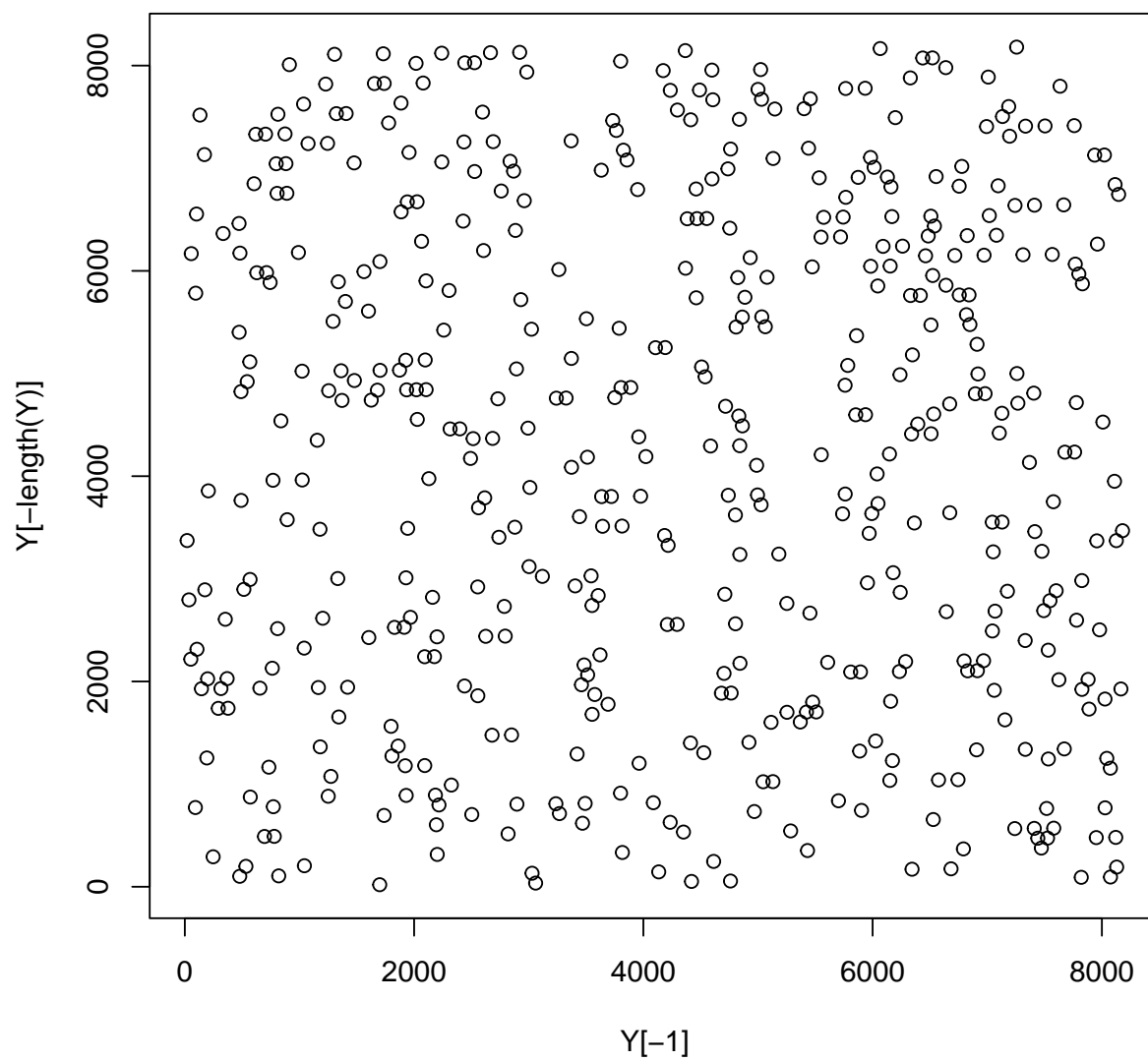
Para valores pares de “a” o período é curto e os números aleatórios gerados começam a repetir. Para $a = 3$ foi observado 3 retas, para $a = 5$, 5 retas e assim por diante. Para valores pequenos de a os números possuem uma correlação mais forte.

Letra e:

```
Y<-geradorCM(n=500,A=85)
cor(Y[-1],Y[-length(Y)])
```

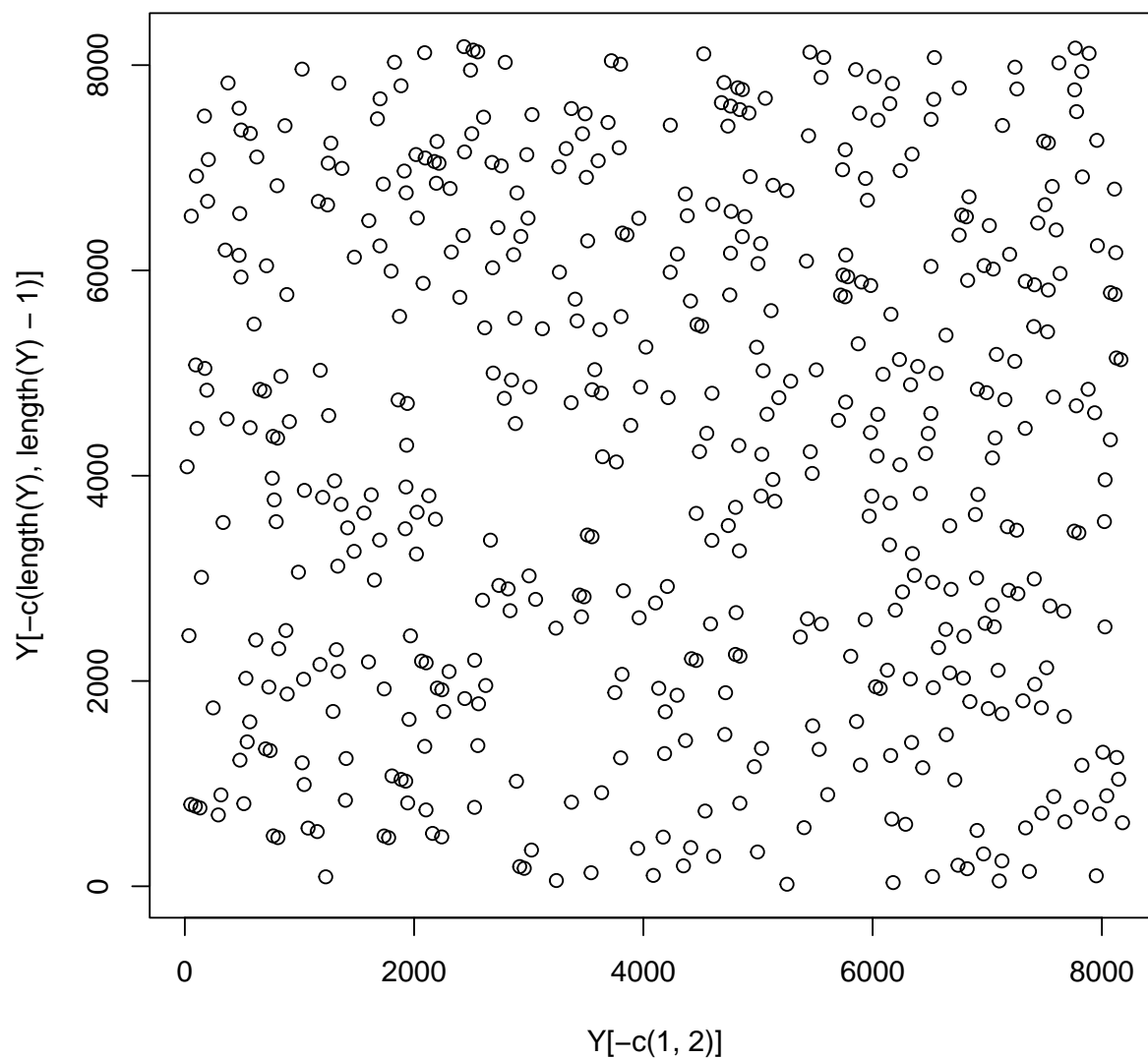
```
## [1] -0.009556836
```

```
plot(Y[-1],Y[-length(Y)])
```



Letra f:

```
plot(Y[-c(1,2)],Y[-c(length(Y),length(Y)-1)])
```



```
cor(Y[-c(1,2)],Y[-c(length(Y),length(Y)-1)])
```

```
## [1] 0.0009671911
```

Para $a = 85$ já apresenta uma correlação muito fraca, apesar dele não ser primo.

Letra g:

Recomenda-se utilizar números primos altos para melhorar a qualidade do gerador.