Universidade
Federal de Juiz de Fora

Geração de números aleatórios pelo Método da Inversa

Jaqueline Lamas da Silva

05/04/2023

Sumário

Geração de 1000 valores

Temos uma variável aleatória X com fdp $f_x(x)$ definida por partes, para gerar números aleatórios desta distribuição aplicamos o Método da Inversa. Ou seja, primeiro geramos numéros de uma uniforme e depois aplicamos ao valores a função quantilica $Q_x(x)$ de X.

$$f_x(x) = \begin{cases} x & \text{se } 0 \leq x < 1 \\ -x+2 & \text{se } 1 \leq x \leq 2 \end{cases}$$

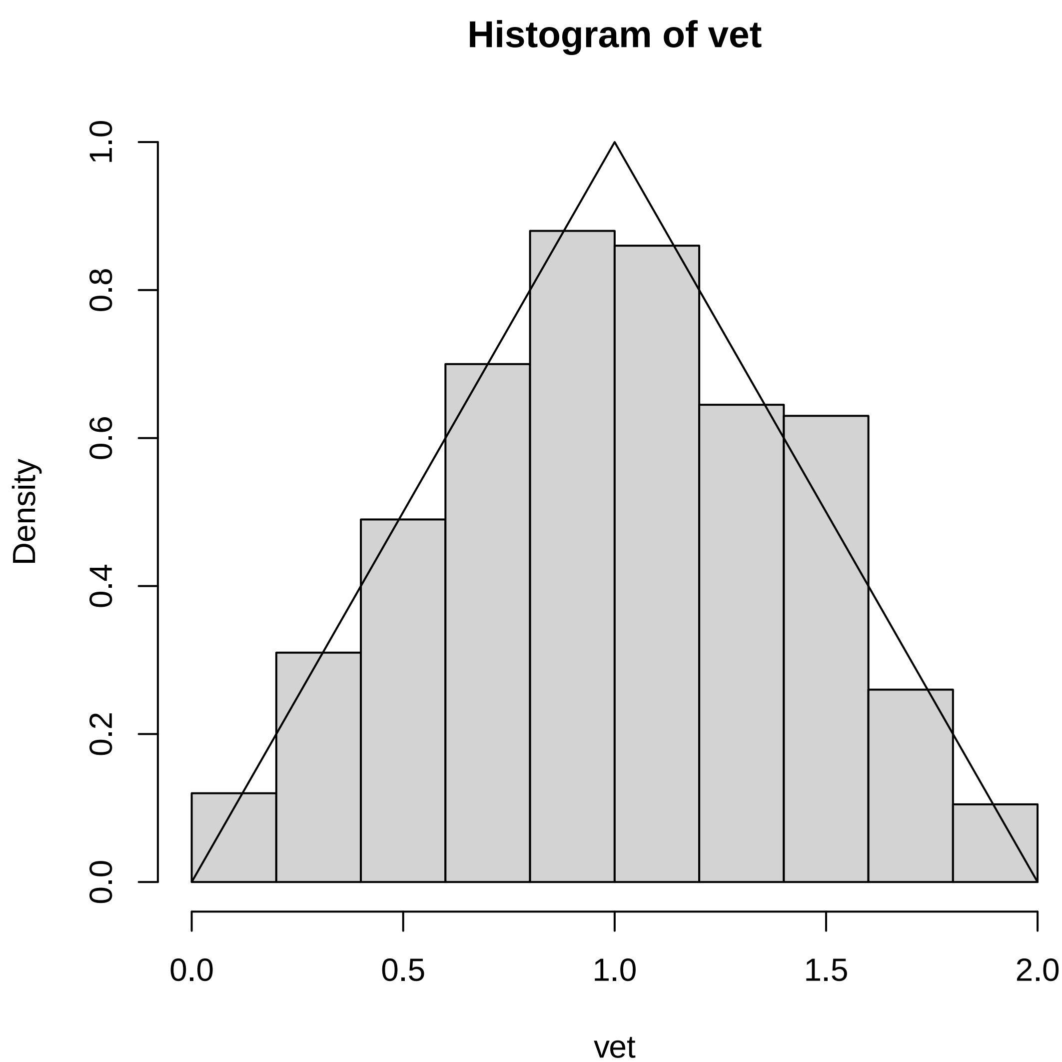
$$Q_x(p) = \begin{cases} \sqrt{2p} & \text{se } 0 \leq p < 1/2 \\ 2-\sqrt{2(1-p)} & \text{se } 1/2 \leq p \leq 1 \end{cases}$$

```
Qx<-function(u)
{
  ifelse(u<0.5, sqrt(2*u), 2-sqrt(2*(1-u)))
}

u<-runif(1000)
vet<-Qx(u)
```

Comparação Histograma x Densidade teórica

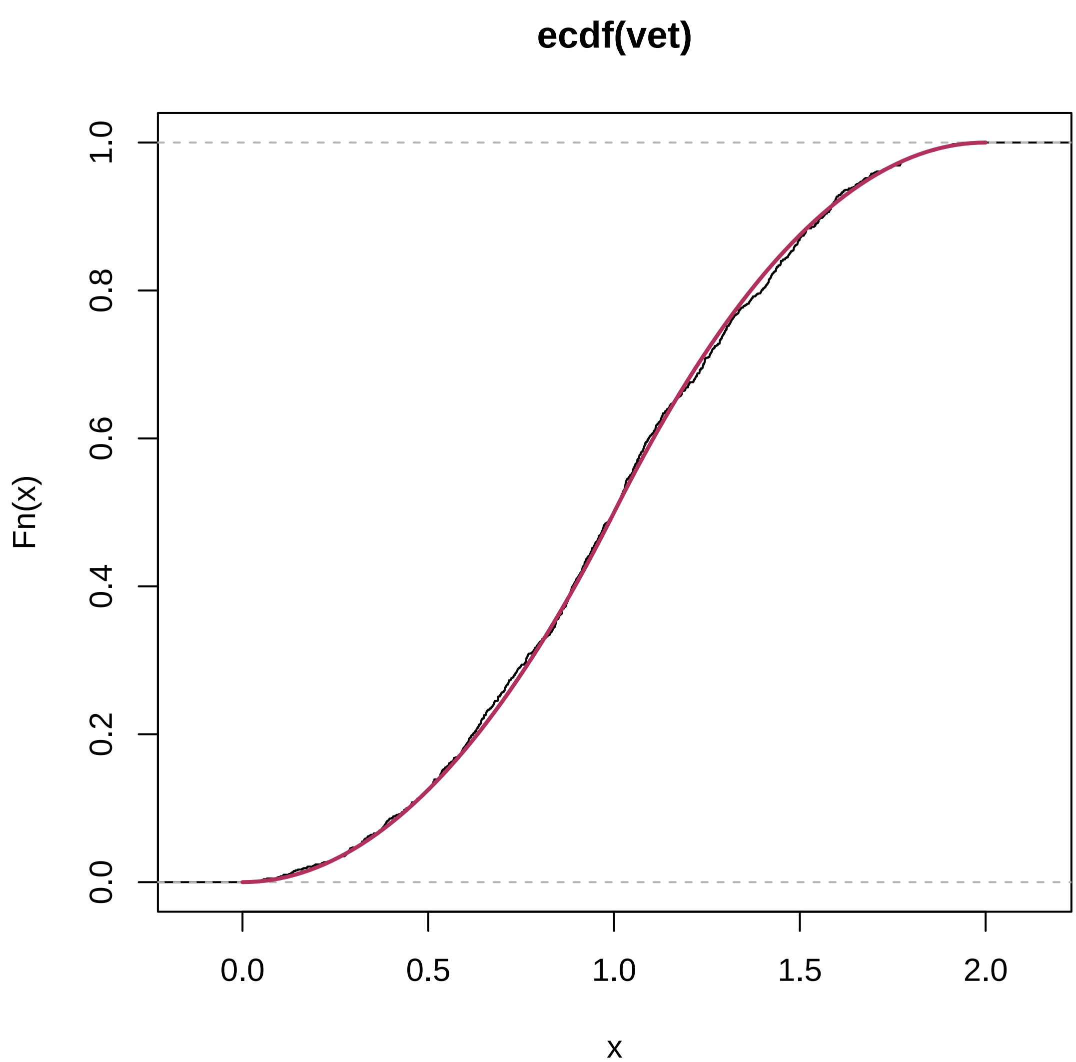
```
fx<- function(x)
{
  ifelse(x<=1, x, 2-x)
}
hist(vet, freq = F, ylim=c(0,1))
curve(fx, from=0, to=2, add = T)
```



Comparação função de distribuição empírica + teste

```
Fn<-ecdf(vet)
plot(Fn)

Fx<-function(x)
{
  ifelse(x<=1,(x^2)/2, (((-x^2)+4*x-2)/2))
}
curve(Fx, from=0, to=2, add = T, col="maroon", lwd=2)
```



```
ks.test(vet, y=Fx)
```

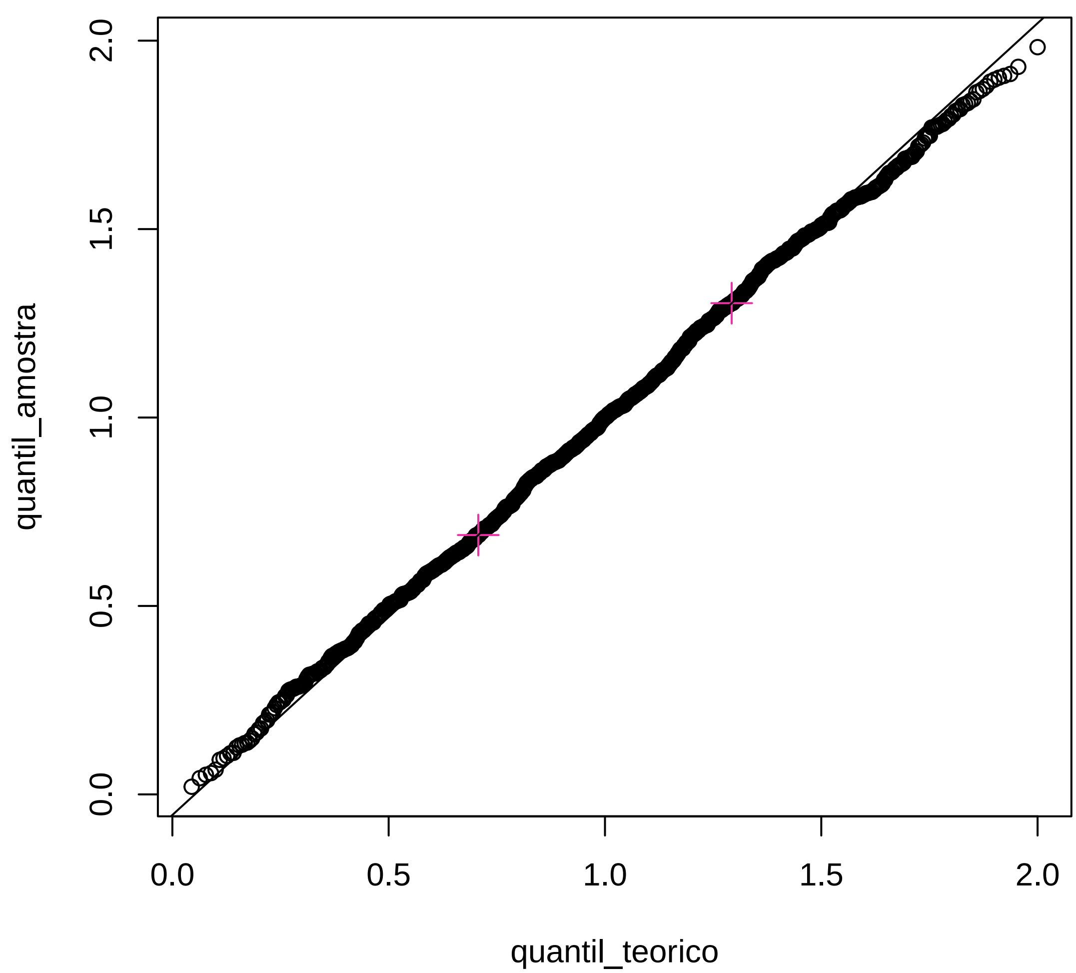
```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  vet
## D = 0.020172, p-value = 0.8105
## alternative hypothesis: two-sided
```

A probablidade de observar um valor mais extremo que o observado no caso da hipótese nula ser verdadeira é grande, maior que 0.05. Ou seja, não temos evidências para rejeitar a hipótese nula, dessa forma os dados parecem vir da distribuição $F_x(x)$. Isso é um indicativo que nosso gerador é bom.

QQ-plot + qqline

```
probs_acumul<-seq_along(vet)/length(vet)
quantil_amostra<-sort(vet)
quantil_teorico<-Qx(probs_acumul)
plot(x=quantil_teorico, y=quantil_amostra)

pt_amostra<-quantile(vet, c(0.25,0.75))
pt_teorico<-Qx(c(0.25, 0.75))
points(x=pt_teorico, y=pt_amostra, pch=3, col="maroon2", cex=2)
b1<-diff(pt_amostra)/diff(pt_teorico)
b0<-pt_amostra[1]-(b1*pt_teorico[1])
abline(a=b0, b=b1)
```



Divisão da amostra em 10 intervalos

```
q.teoricos<-Qx(0:10/10)
categorias<-cut(vet, breaks=q.teoricos,labels = paste0("Q",1:10))
tabs<-table(categorias)
e<-length(vet)/10
Xq<-sum(((tabs-e)^2)/e)
pchisq(Xq,9, lower.tail=F)
```

```
## [1] 0.7238042
```

Não há evidências para rejeitar H0 ao nível de significância de 5%, a amostra gerada parace seguir a distribuição teórica.

Divisão da amostra em 100 intervalos

```
q.teoricos<-Qx(0:100/100)
categorias<-cut(vet, breaks=q.teoricos,labels = paste0("Q",1:100))
tabs<-table(categorias)
e<-length(vet)/100
Xq<-sum(((tabs-e)^2)/e)
pchisq(Xq,99, lower.tail=F)
```

```
## [1] 0.9285985
```

Novamente não há evidências para rejeitar H0 ao nível de significância de 5%, e podemos concluir que a amostra gerada parace seguir a distribuição teórica.

Comparação de medias e variâncias

Gere amostras de tamanhos n = 10, 20, 30, 50 e 100 e compare a média e variância dessas amostras com a média e variância verdadeiras (calcule os valores exatos da distribuição triangular).

```
sizes<-c(10,20,30,50,100)
Qx<-function(u)
{
  ifelse(u<0.5, sqrt(2*u), 2-sqrt(2*(1-u)))
}

u<-lapply(sizes,runif)
amostras<-lapply(u, Qx)

mediaT<-1
varT<-1/6

medias<-sapply(amostras, mean)
variancias<-sapply(amostras, var)
tabela<-rbind(medias, variancias)
tabela<-cbind(c(mediaT, varT),tabela)
colnames(tabela)<-c("Teorico",paste0("n = ",sizes))
knitr::kable(tabela, caption="Tabela com os valores das médias
             e variâncias para os diferentes tamanhos de amostra")
```

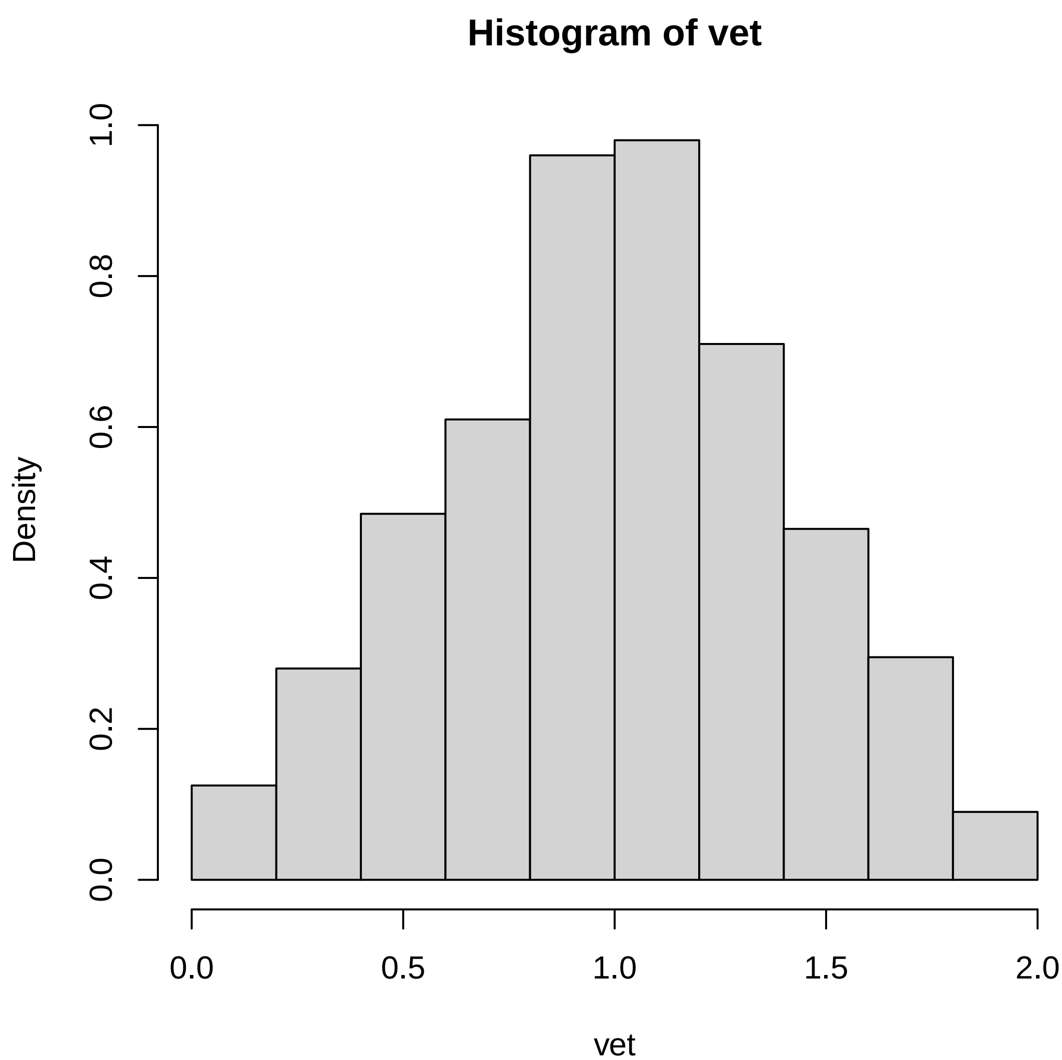
Table 1: Tabela com os valores das médias e variâncias para os diferentes tamanhos de amostra

	Teorico	n = 10	n = 20	n = 30	n = 50	n = 100
medias	1.0000000	0.9828073	1.0056050	1.2073182	0.9996788	0.9429543
variancias	0.1666667	0.1543518	0.1978959	0.1518107	0.1514512	0.1857778

Após comparar as médias e as variâncias também podemos dizer que nossa amostra parace seguir a distribuição triângular proposta, os valores se aproximam do valor teórico a medida que aumentamos o tamanho da amostra.

Gerando os números sem usar "ifelse"

```
Qx<-function(u)
{
  x<-as.numeric(u<0.5)
  qx<-(sqrt(2*u)^x)*(2-sqrt(2*(1-u))^(1-x))
  return(qx)
}
u<-runif(1000)
vet<-Qx(u)
hist(vet, freq = F)
```



Outra abordagem para gerar uma triangular:

Outra abordagem para gerar uma triangular é pela soma de uniformes independentes. $U1$: Uniforme (0,1) $U2$: Uniforme (0,1)

Δ=$U1 + U2$

```
u1<-runif(1000)
u2<-runif(1000)
vet<-u1+u2
hist(vet, freq = F)
curve(fx, from=0, to=2, add = T)
```