Estatística Computacional

Patrícia de Siqueira Ramos

PPGEAB
UNIFAL-MG

4 de Abril de 2018

Distribuições de frequências no R

Usando um conjunto de dados pequeno

- Funções para trabalhar com vetores

```
notas = read.table("~/Downloads/notas.txt", h=T)
notas
names(notas)
plot(notas$mat, notas$est)
hist(notas$mat)
cor(notas)

# alguns comandos vistos
max(notas$mat)
min(notas$mat)
range(notas$mat)
range(notas)
```

Histogramas

```
# histogramas
hist(notas$mat, breaks=10)
hist(notas$mat, breaks=20)

# definindo os intervalos exatos
hist(notas$mat, breaks=c(40, 60, 80, 100))
# melhor
classes = seq(45, 100, 5)
hist(notas$mat, breaks=classes)
# mais opções
hist(notas$mat, breaks=classes,main=" ", col="red")
```

Frequências relativas

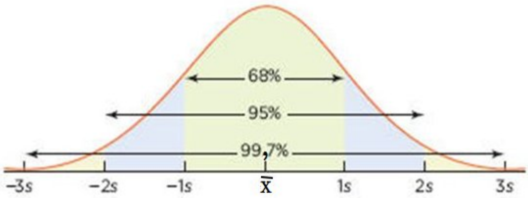
- Frequência relativa: obter a fração de observações se encontra em cada intervalo de valores de uma amostra
- Isso permite aproximar a probabilidade que determinada variável aleatória assuma valores em um intervalo

```
d = c(2,2,3,3,1)
hist(d)
# Feio! Vamos alterar os intervalos:
hist(d, breaks=seq(0.5, 3.5, 1))
# Melhor. Agora vamos obter as frequências relativas:
hist(d, breaks=seq(0.5, 3.5, 1), probability=T)
# Frequência relativa do 1 nesses dados é 1/5 = 0,2,
# por exemplo
```

Regra empírica: 68% - 95% - 99,7%

- Essa regra afirma que se uma distribuição de frequências de um conjunto de dados é normalmente distribuída, então, aproximadamente:

  - 68% dos dados estarão dentro de um desvio padrão da média: $\bar{X} \pm 1S$
  - 95% dos dados estarão dentro de dois desvios padrão da média: $\bar{X} \pm 2S$
  - $99,7\%$ dos dados estarão dentro de três desvios padrão da média: $\bar{X} \pm 3S$



Exemplo da regra empírica

- Usando um conjunto de dados disponível no R, vamos verificar a regra empírica:

```
data(faithful)
help(faithful)
names(faithful)

# variável escolhida
x = faithful$eruptions

n = length(x)
k = sqrt(n) # n. classes é a raiz de n
hist(x, k)
```

Exemplo da regra empírica

- Queremos ver a porcentagem dos dados que está dentro de cada intervalo da regra empírica
- Primeiro vamos calcular a média e o desvio padrão amostrais

```
Xb = mean(x)
S = sd(x)
```

- Para ver o número de observações dentro de $1S$:

Exemplo da regra empírica

- Queremos ver a porcentagem dos dados que está dentro de cada intervalo da regra empírica
- Primeiro vamos calcular a média e o desvio padrão amostrais

```
Xb = mean(x)
S = sd(x)
```

- Para ver o número de observações dentro de $1S$:

```
sum(x > Xb - S & x < Xb + S)
```

- Para ver a proporção das observações dentro de $1S$:

Exemplo da regra empírica

- Queremos ver a porcentagem dos dados que está dentro de cada intervalo da regra empírica
- Primeiro vamos calcular a média e o desvio padrão amostrais

```
Xb = mean(x)
S = sd(x)
```

- Para ver o número de observações dentro de $1S$:

```
sum(x > Xb - S & x < Xb + S)
```

- Para ver a proporção das observações dentro de $1S$:

```
sum(x > Xb - S & x < Xb + S) / n
```

- Fazer o mesmo para $2S$ e $3S$. Qual a conclusão?

Exemplo da regra empírica

- Considerando esses dados como uma distribuição, de que forma poderíamos obter os quantis dela, por exemplo, $q_{0,95}$, ou seja, o ponto que deixa 95% dos dados abaixo dele?

Exemplo da regra empírica

- Considerando esses dados como uma distribuição, de que forma poderíamos obter os quantis dela, por exemplo, $q_{0,95}$, ou seja, o ponto que deixa 95% dos dados abaixo dele?

```
x_ord = sort(x)
# posição do ponto que deixa 95% à esquerda
pos = round(95 / 100 * n)
x_ord[pos]
```

Mais funções no R

Relembrando: como escrever funções próprias no R

Sintaxe:

```
minha_funcao = function(arg1, arg2, ...){
  # algo interessante
}
```

Exemplo da regra empírica

- Elaborar uma função chamada `regra_emp` que recebe um vetor com observações de uma variável e retorna as proporções da regra empírica e o histograma dos dados

Exemplo da regra empírica

```
regra_emp = function(var){
   n = length(var)
   k = sqrt(n)
   hist(var, breaks=k)
   Xb = mean(var)
   S = sd(var)
   d1 = sum(var > Xb - S & var < Xb + S) / n
   d2 = sum(var > Xb - 2 * S & var < Xb + 2 * S) / n
   d3 = sum(var > Xb - 3 * S & var < Xb + 3 * S) / n
   return(list(um = d1, dois = d2, tres = d3))
}

# uso
data("faithful")
x = faithful$eruptions
regra_emp(x)
```

Preencher matriz

- Função `matrizA` que crie uma matriz quadrada $n \times n$ segundo as regras:

$$\begin{array}{ll} i * j, & \text{se } i = j \\ i + j, & \text{se } i \neq j \end{array}$$

Preencher matriz

- Função `matrizA` que crie uma matriz quadrada $n \times n$ segundo as regras:

$$\begin{array}{ll} i * j, & \text{se } i = j \\ i + j, & \text{se } i \neq j \end{array}$$

```
matrizA = function(n){
   A = matrix(0,n,n)
   for(i in 1:n){
      for(j in 1:n){
         if(i == j){
            A[i,j] = i * j
         }else{
            A[i,j] = i + j
         }
      }
   }
   return(A)
}
```

- Teste o uso da função para matrizes 3×3, 5×5, 10×10

Impor uma condição

- Retirar amostras de tamanho n de uma normal padrão até obter uma amostra cuja média seja menor do que um valor ϵ:

Impor uma condição

- Retirar amostras de tamanho *n* de uma normal padrão até obter uma amostra cuja média seja menor do que um valor ϵ:

```
amostra = function(epsilon, n){
   i = 0
   repeat{
      i = i + 1
      media = abs(mean(rnorm(n)))
      print(media)
      if(media < epsilon) break
   }
   list(media=media, n.iteracoes=i)
}
```

- Teste o uso da função para valores de $\epsilon = 0,01$ e para outros valores
- Modificar a função para que ela também vá mostrando os valores das médias obtidas a cada passo

Criar f.d.p. - exponencial

- Seja $X \sim exp(2)$. Sua função densidade de probabilidade é $f(x) = 2e^{-2x}, x \geq 0$. Criar uma função chamada `fexp2` que retorne o valor da função em um ponto específico.

Criar f.d.p. - exponencial

- Seja $X \sim exp(2)$. Sua função densidade de probabilidade é $f(x) = 2e^{-2x}, x \geq 0$. Criar uma função chamada `fexp2` que retorne o valor da função em um ponto específico.

```
fexp2 = function(x) {
   if(x < 0){
      fx =  0
   }else{
      fx = 2 * exp(-2 * x)
   }
return(fx)
}

# uso
fexp2(x = 3)
```

Criar f.d.p. - exponencial

- Forma equivalente usando `ifelse`:

```
fexp2 = function(x) {
   fx = ifelse(x < 0, 0, 2 * exp(-2 * x))
   return(fx)
}

# uso
fexp2(x = 3)

# como é o formato da f.d.p.
curve(fexp2(x), from = 0, to = 4, ylab = "y")
```

Criar f.d.p. - exponencial geral

- Seja $X \sim exp(\alpha)$. Sua função densidade de probabilidade é

  $$f(x) = \alpha e^{-\alpha x}, x \geq 0.$$

  Criar a função `fexp` que recebe o valor de um ponto x a ser avaliado e do parâmetro a da distribuição.

Criar f.d.p. - exponencial geral

- Seja $X \sim exp(\alpha)$. Sua função densidade de probabilidade é

$$f(x) = \alpha e^{-\alpha x}, x \geq 0.$$

Criar a função `fexp` que recebe o valor de um ponto x a ser avaliado e do parâmetro a da distribuição.

```
fexp = function(x, a){
   fdp = ifelse(x < 0, 0, a * exp(-a * x))
   return(fdp)
}
# função mais geral com a=2
fexp(x = 3, a = 2)
# função do R para x=3 com alpha=2
dexp(x = 3, rate = 2)
# integral de -Inf a +Inf da fexp com a=2
integrate(fexp, -Inf, Inf, a = 2)
```

Criar f.d.p. - exponencial geral

- Verificar que `fexp2` também é f.d.p.
- Verificar que `dexp` com `a = 2` do R também é f.d.p.
- Obter a integral de 0 a 2 da `fexp2`.
- Obter a integral de 2 a Inf da `fexp2`.

Lançar moeda

- Simular o lançamento de uma moeda (dez vezes, por exemplo):

```
sample(0:1, 10, rep=T)
```

Lançar moeda

- Simular o lançamento de uma moeda (dez vezes, por exemplo):

```
sample(0:1, 10, rep=T)
```

- Por meio de uma função que simule *n* lançamentos:

```
moeda = function(n){
   sample(0:1, n, rep=T)
}

# uso
e1 = moeda(30)
e1
```

Lançar moeda

- Usar comando `sum` para comparar os resultados desse experimento com as probabilidades teóricas.
- Por exemplo, no experimento da moeda, ao lançá-la 30 vezes, podemos contar caras e coroas:

```
# soma de caras
sum(e1==0)
# %caras

# soma de coroas

# %coroas

# e obter um histograma
hist(e1,breaks=c(-0.5,0.5,1.5), prob=T)
# e um gráfico de barras?
```

Questões de V ou F

- Numa prova com 10 questões de V ou F, qual a probabilidade de acertar 9 chutando?

```
sample(c("C","E"), 10, rep=T)
```

Questões de V ou F

- Numa prova com 10 questões de V ou F, qual a probabilidade de acertar 9 chutando?

```
sample(c("C","E"), 10, rep=T)
```

- Por meio de uma função para *n* questões:

```
CE = function(n){
  sample(c("C","E"), n, rep=T)
}
# uso
n = 10
(p1 = CE(n))
sum(p1 == "C")
sum(p1 == "C")/n
sum(p1 == "E")
sum(p1 == "E")/n
# fazer um gráfico de barras - por que não um histograma?
barplot(table(p1))
```

Questões de V ou F

- E para m provas? Como os resultados poderiam ser mostrados?

$$\begin{array}{cccc} & n & C & E \\ 1 & 10 & 4 & 6 \\ 2 & 10 & 3 & 7 \\ \vdots & \vdots & \vdots & \vdots \\ m & 10 & 5 & 5 \end{array}$$

Questões de V ou F - *m* provas

```
m = 10
res = matrix(0, m, 3)
colnames(res) = c("n", "C", "E")
res[,1] = n
for(i in 1:m){
   prova = CE(n)
   res[i,2] = sum(prova=="C") # certas
   res[i,3] = sum(prova=="E") # erradas
}
```

Questões de V ou F

```
# acertou e errou quantas das 10 questões em cada prova?
res
# qual a frequência de cada número de acertos?
table(res[,2])  # C
sum(table(res[,2]))  # C
table(res[,3])  # E
res
# em porcentagem
table(res[,2]) / m * 100
barplot(table(res[,2])/ m * 100)
```

Questões de V ou F

```
# acertou e errou quantas das 10 questões em cada prova?
res
# qual a frequência de cada número de acertos?
table(res[,2])  # C
sum(table(res[,2]))  # C
table(res[,3])  # E
res
# em porcentagem
table(res[,2]) / m * 100
barplot(table(res[,2])/ m * 100)
```

- Como poderíamos comparar esses resultados com as probabilidades teóricas?

```
# comparar com a binomial
dbinom(0:10, 10, 0.5)
```

Questões de V ou F

- E para $m = 1000$ provas? Como ficaria?
- E para $m = 10000$ provas?

Lista de exercícios 3

1. Simule n dados de uma distribuição normal padrão, ou seja, $N(\mu = 0, \sigma = 1)$ e:

a) utilize a função `regra_emp` para verificar se os dados simulados atendem à regra empírica. Teste com $n = 100$, 1000 e 10000.
b) obtenha o `QQplot` dos dados simulados.

Lista de exercícios 3

2. Modificar a função `moeda` com n lançamentos para que as ocorrências sejam "C" (cara) e "K" (coroa) e que:
a) além dos resultados "C" e "K", a função também retorne as frequências dos resultados (absolutas e relativas - podendo-se usar o comando `table` para isso).
b) depois de usar a função, faça um gráfico de barras usando o resultado dos lançamentos da moeda com $n = 100$ e $n = 1000$.

3. Criar a função `dado` que permita lançar um dado n vezes e:
a) tenha o mesmo retorno da função anterior.
b) compare com as probabilidades teóricas.