Getting Started

*Utilizarei o símbolo > para indicar o input do console e >>> para o output
-As funções básicas são o R base, o restante são adicionados pela comunidade com CRAN ou github.

DISCRETE PROBABILITY

Pr(A) se referente a notação de chance de ocorrer “A”
event ex. “tirar uma bola do cesto de bolas”

MONTE CARLO SIMULATIONS

Nesse caso, colocamos 2 bolas vermelhas e 3 bolas azuis em um grupo e então, sorteamos uma aleatória e vemos sua cor.
No caso do exemplo de Monte Carlo, fazemos isso um número suficiente de vezes para termos a probabilidade de cada cor ser sorteada do grupo de bolas.

```
> beads <- rep( c(“red”, “blue”), times = c(2,3))        //cria um repositório de urnas
> beads
>>> red red blue blue blue
> sample(beads, 1)                                       //escolhe “1” dos disponíveis em “beads”
>>> blue
```

Agora fazendo o evento 10000 vezes:

```
> B <- 10000
> events <- replicate(B, sample(beads, 1))
> tab <- table(events)                                   //cria uma tabela com os eventos
> tab
>>> events
>>> blue   red
>>>5962   4038
> prop.table(tab)                                        //nos da a proporção
```

*Note que é importante notar que a função sample está como “*without replacement”* mas podemos colocá-la “*with replacement”* , sem necessariamente ter que usar a função *“replicate”*, com:
> events <- sample(beads, B, replace = TRUE)

SETTING THE RANDOM SEED

set.seed()
ex.
> set.seed(1986)

Isso ajuda quando queremos ter um valor exato de seed de números aleatórios, pois setamos a seed em 1986

Uma maneira comum de se escolher uma seed é de “ano-mês-dia”, exemplo:
-01/10/2020
-2020 - 10 - 1 = 2009
Para ver mais:
?set.seed

OBS: o modo de setar a seed muda do R 3.5 para o R 3.6
set.seed(1), muda para:
> set.seed(1, sample.kind="Rounding") # will make R 3.6 generate a seed as in
. R3.5

PROBABILITY DISTRIBUTIONS

-Dois eventos são independentes se um não afeta o outro, como por exemplo, jogar cara ou coroa.
-Eventos não são independentes se um afeta o outro, como por exemplo, tirar uma carta do baralho e não repô-la.

$Pr(\text{Card 2 is a king} \mid \text{Card 1 is a king})=3/51$

*Isto é um exemplo de probabilidade condicional: “qual a chance da segunda carta ser um rei se a primeira carta foi um rei?”

Se o caso for de eventos independentes: $Pr(A \mid B)=Pr(A)$

Equações

Eventos independentes:
$Pr(A \text{ and } B \text{ and } C)=Pr(A)\times Pr(B)\times Pr(C)$
Eventos independentes com probabilidade condicional de ambos os eventos acontecerem:
$Pr(A \text{ and } B)=Pr(A)\times Pr(B \mid A)$
Eventos dependentes para mais de 2 eventos:
$Pr(A \text{ and } B \text{ and } C)=Pr(A)\times Pr(B \mid A)\times Pr(C \mid A \text{ and } B)$

COMBINATIONS AND PERMUTATIONS

paste() //une 2 strings com um espaço no meio
ex.
joining strings with paste
> number <- "Three"
> suit <- "Hearts"
> paste(number, suit)

>>>"Three Hearts"

joining vectors element-wise with paste
> paste(letters[1:5], as.character(1:5))

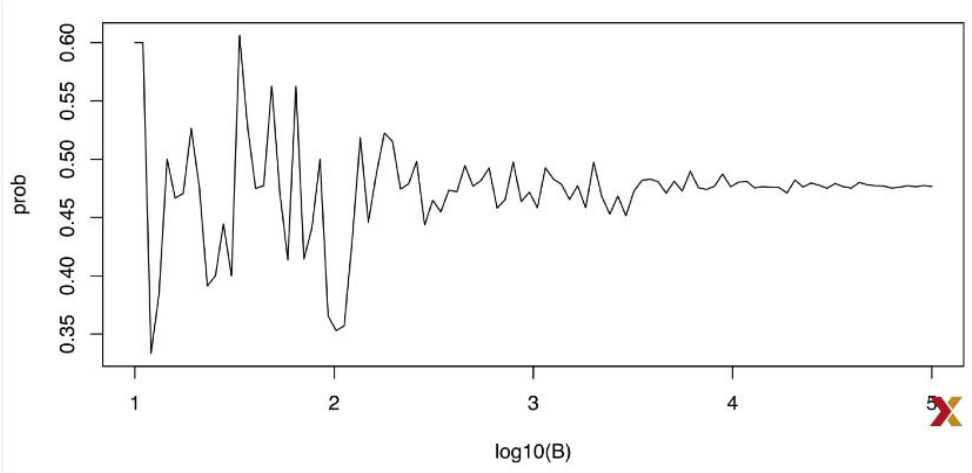
expand.grid() //nos da a combinação de 2 vetores ou listas
ex.
generating combinations of 2 vectors with expand.grid
> expand.grid(pants = c("blue", "black"), shirt = c("white", "grey", "plaid"))
ex2.
> suits <- c("Diamonds", "Clubs", "Hearts", "Spades")
> numbers <- c("Ace", "Deuce", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine", "Ten", "Jack", "Queen", "King")
> deck <- expand.grid(number = numbers, suit = suits)
> deck <- paste(deck$number, deck$suit)

permutations(n,r) // (do pacote "gtools"), nos da os diferentes modos de que "r" itens sejam selecionados de um *set "*n" com a ordem importando.
ex.
> library(gtools)
> permutations(5,2) # ways to choose 2 numbers in order from 1:5
> all_phone_numbers <- permutations(10, 7, v = 0:9)

combinations(n,r) // (do pacote "gtools"), nos da os diferentes modos de que "r" itens sejam selecionados de um *set "*n" com a ordem **não** importando.
ex.
> combinations(3,2)

QUAL VALOR PARA SIMULAR MONTE CARLO É O IDEAL?

-Rode até que o valor escolhido para o número de monte carlos esteja estável:



THE ADDITION RULE

Pr(A or B)=Pr(A)+Pr(B)−Pr(A and B)

A probabilidade de um evento A ou evento B ocorrer é a probabilidade de A + probabilidade de B menos a probabilidade de ambos acontecerem juntos.

Ex. tirar uma carta A e CARTA-ROSTO ou CARTA-ROSTO e um A:

$$\Pr(\text{ace then facecard}) = \frac{4}{52} \times \frac{16}{51}$$

$$\Pr(\text{facecard then ace}) = \frac{16}{52} \times \frac{4}{51}$$

$$\Pr(\text{ace then facecard} \mid \text{facecard then ace}) = \frac{4}{52} \times \frac{16}{51} + \frac{16}{52} \times \frac{4}{51} = 0.0483$$

CONTINUOUS PROBABILITY

-Ao trabalhar com medidas muito precisas e diferentes entre si, é interessante se trabalhar com intervalos.

-Caso queira-se saber qual a probabilidade de se escolher uma pessoa com altura acima de 1,70, usa-se a função **CDF:**

ex.
> 1 - F(1.70)
>>> 0.3768473 //37% das pessoas estão acima dessa altura
ex2. para saber a faixa que está entre 1.50 e 1.70 (F(b) - F(a))
> F(1.70) - F(1.50)

-É possível se obter F(a) = pnorm(a, avg, s)

ex.
> 1 - pnorm(1.70, mean(x), sd(x))

PROBABILITY DENSITY

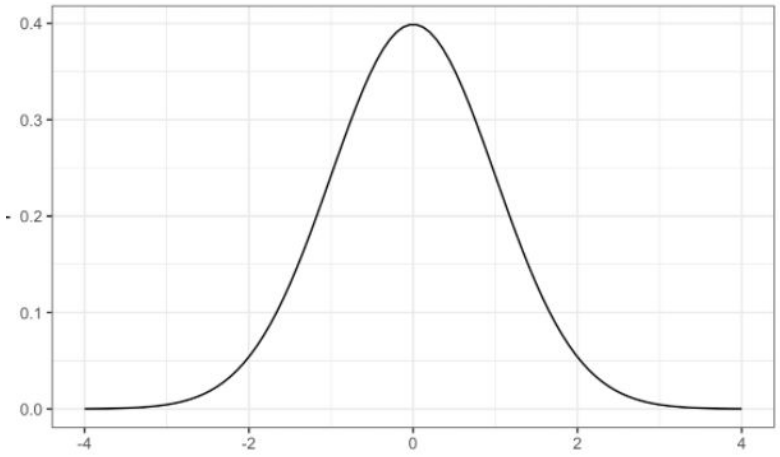
$$F(a) = \Pr(X \leq a) = \int_{-\infty}^{a} f(x)\,dx$$

-A probabilida de um único valor não é definida para uma distribuição contínua, porém, a quantia com o mair número de valores similares de um único valor é a função densa f(x)
> dnorm() //nos da a densidade de uma função

- dnorm(z, mu, sigma)

ex.

```
> library(tidyverse)
> x <- seq(-4, 4, length = 100)
> data.frame(x, f = dnorm(x)) %>%
>   ggplot(aes(x, f)) +
>   geom_line()
```



Code: Monte Carlo simulation of tallest person over 7 feet

```
> B <- 10000
> tallest <- replicate(B, {
 >   simulated_data <- rnorm(800, avg, s)    # generate 800 normally distributed random heights
>   max(simulated_data)    # determine the tallest height
> })
```

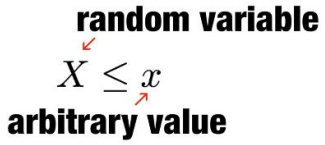
> mean(tallest >= 7*12)   # proportion of times that tallest person exceeded 7 feet (84 inches)

OTHER CONTINUOUS DISTRIBUTIONS

- student-t
- chi-squared
- exponential
- gamma
- beta

RANDOM VARIABLES, SAMPLING MODELS, AND THE CENTRAL LIMIT THEOREM

-Letras minúsculas são usadas para valores observados e letras maiúsculas são usadas para eventos aleatórios:



*Em Pr(X=x), pergunta-se quão frequentemente o valor X é igual ao valor x.

CENTRAL LIMIT THEOREM [CLT]

-Quando o número de sorteios independentes (*sample size)* é grande, a propabilidade de distribuição desses sorteios é aproximadamente *normal.*

-Se uma variável random tem uma *probability distributtion* que é aproximada com a *normal distribution,* então, tudo que precisamos descrever é que a probabilidade de distribuição é a média(*average*)[*expected value*] e desvio padrão*(standard deviation*)[*standard error*].

-Se houver muitos sorteios (não funcionando para números pequenos de sorteios) , a média de sorteios será aproximada do valor esperado pela probabilidade. Por exemplo, se houver uma urna com 20 valores R$-1.00 e 18 valores R$1.00, a média do sorteio será a média da urna, no caso 0.05 (ou seja,o cassino ganharia em média 0.05 centavos por jogo da urna).

-A urna só tem 2 possibilidades, A(-1) ou B(1) que se encontram numa proporção P, ou seja, a média é:

A*P + B*(1-P)

E como a média é a soma, temos N vezes de sorteios:

(N*A*P + N*B*(1-P)) / N, que é a mesma da fórmula acima.

-Mas qual a faixa de possibilidades que o cassino pode esperar nesse jogo da urna?

Nesse caso, procuramos pelo *standard error*(SE).

Se os sorteios forem independentes, o *standard error* é dado pela raíz quadrada dos números do sorteio vezes o *standard error* dos números da urna.

$$\sqrt{\text{number of draws}} \times \text{standard deviation of the numbers in the urn}$$

Com auxílio da matemática, conseguimos notar que se uma urna contém 2 valores com proporções P e 1-P, o desvio padrão é de:

$$|b-a|\sqrt{p(1-p)}$$

, que no caso da urna citada, seria de 0.9986.

Usando a fórmula, também nota-se que para 1.000 jogos, o desvio é de R$32.00.

-Se a probabilidade de ganhar é muito baixa, como na loteria, então o teorema do limite central não se aplica nem mesmo em grandes quantidades de sorteios. Nesse caso, a aproximação normal não é uma boa, podendo-se utilizar a *“Poisson distribution”*. *https://en.wikipedia.org/wiki/Poisson_distribution*