**Apêndice Aula 5**: [Inferência no Modelo Linear – 19/09 (n) / 21/09 (v)](https://edisciplinas.usp.br/course/view.php?id=103128#section-6)

**Códigos Úteis no R**

--------------------------------------------------------------------------------------------------------------------

**Referências**

[**https://jonnyphillips.github.io/Ciencia\_de\_Dados/Estatisticas\_Resumidas.html**](https://jonnyphillips.github.io/Ciencia_de_Dados/Estatisticas_Resumidas.html)

**Importando arquivos Excel**

install.packages(“readxl”)

library(readxl)

dados <- read\_excel(‘C:/<caminho do arquivo>/<arquivo>.xlsx’

**Gráfico boxplot**

library(tidyverse)

ggplot(<tabela>, aes(<variável>) +

 geom\_boxplot()

**Regressão linear**

dados %>%

 lm(y ~ x + controle1 + controle2, data = .) %>%

 summary()

**Estatísticas Resumidas (summarize)**

| **Estatística** | **Função em R** |
| --- | --- |
| Média | mean(variável) |
| Mediana | median(variável) |
| Desvio padrão | sd(variável) |
| Quantil (10%) | quantile(variável, probs=0.1) |
| Máximo | max(variável) |
| Mínimo | min(variável) |

A nossa nova tabela agregada pode conter mais de uma estatística resumida, cada uma em uma coluna nova:

Base\_de\_dados %>% [summarize](https://rdrr.io/pkg/tidylog/man/summarize.html)(media\_var1=[mean](https://rdrr.io/r/base/mean.html)(var1),

 mediana\_ var1=[median](https://rdrr.io/r/stats/median.html)(var1),

 sd\_ var1=[sd](https://rdrr.io/r/stats/sd.html)(var1))

**Grupos (group\_by)**

O poder de summarize() é ampliado exponencialmente quando os resumos / agregações são feitos no nível de grupos e não para o banco de dados inteiro. Para definir os grupos relevantes, podemos usar a função group\_by():

objeto <- banco\_de\_dados %>% [group\_by](https://rdrr.io/pkg/tidylog/man/group_by.html)(var\_x)

Na prática, group\_by() sozinho não é útil para nada. Temos que combinar com mais uma função subsequente para gerar resultados interessantes. Por exemplo, vamos calcular a média da distância por aeroporto:

banco\_de\_dados %>% [group\_by](https://rdrr.io/pkg/tidylog/man/group_by.html)(origin) %>% [summarize](https://rdrr.io/pkg/tidylog/man/summarize.html)(mean\_distance=[mean](https://rdrr.io/r/base/mean.html)(distance))

**Referências**

[**http://rstudio-pubs-static.s3.amazonaws.com/408638\_6679293d4c7a415eaebe00faa3aea0cb.html**](http://rstudio-pubs-static.s3.amazonaws.com/408638_6679293d4c7a415eaebe00faa3aea0cb.html)

**Pra que serve o teste t de Student?**

Compara duas médias e mostra se as diferenças entre as médias são significativas. Em outras palavras, permite que você avalie se essas diferenças ocorreram por um mero por acaso ou não. Por exemplo, se um grupo experimental difere de um grupo controle, se uma amostra difere da população, se um grupo difere antes de depois de um procedimento. O test t serve justamente para isso, para estimar se a diferença das médias de dois conjuntos de dados é ou não estatisticamente significativa.

**Como interpretar o resultado do teste t**

Como todo teste estatístico, a teste t também tem como produto a medida do valor de p. Ou seja, no final das contas, teremos calculado a probabilidade da diferença encontrada (entre as médias) terem sido por acaso. Se esse valor for menor que 5% ( p < 0.05), a tradição científica é de rejeitarmos a hipótese de que as diferenças sejam por acaso (rejeitamos a hipótese nula) e alegamos termos encontrado uma diferença estatísticamente significativa.

**Os vários tipos de teste t**

Existem 3 tipos comuns de teste t:

1. teste t para duas amostras independentes (ou não pareadas):
2. teste t para duas amostras dependentes (ou pareadas)
3. teste t para uma amostra (para comparar com a média da população)

**Realizando o test t padrão do R: teste t para amostras independentes**

O R tem uma função muito simples de usar para realizar o teste t: t.test().

Para usar essa função basta incluir como argumentos os valores obtidos de cada grupo da pesquisa e a função já calcula a média de cada grupo e faz a comparação estatística. Veja o exemplo a seguir, no qual existem dois grupos diferentes de pacientes (controle e experimental), com os valores de uma medida fictícia:

controle <- c(21, 28, 24, 23, 23, 19, 28, 20, 22, 20, 26, 26)

experimental <- c(26, 27, 23, 25, 25, 29, 30, 31, 36, 23, 32, 22)

Um gráfico de boxplot pode nos mostrar que o grupo experimental tem uma média maior.

Para comparar a média de cada um desses dois grupos, usamos a função t.test().

t.test(controle, experimental)

O teste t no R tem como padrão admitir que os grupos não sejam pareados, isto é, que sejam independentes. Além disso, o teste t assume que o nível de significância seja de 0.05 (αα= 5%) e que teste seja realizado de forma bicaudal. Todos esses parâmetros podem ser modificados quando necessário.

**Argumentos do test t no R**

O teste-t usa valores default em vários de seus argumentos que são importantes serem conhecidos:

paired = FALSE -> por default o teste admite que os grupos não sejam pareados

conf.level = 0.95 -> por default o teste admite um nível de confiança de 95% (ou αα= 5%)

alternative = two.sided - > por default o teste é bicaudal, podemos escolher também “greater” ou “less”

Quando desejarmos realizar um teste t pareado, basta alterar o argumento paired para TRUE: paired = TRUE.

**Teste t pareado**

O teste t para amostras independentes não exige que os grupos tenham o mesmo tamanho, mas observe que o teste t pareado exige que existam exatamente o mesmo número de medidas antes e depois.

antes <- c(21, 28, 24, 23, 23, 19, 28, 20, 22, 20, 26, 26)

depois <- c(26, 27, 23, 25, 25, 29, 30, 31, 36, 23, 32, 22)

Para usar o teste t pareado, iremos simplesmente alterar o argumento paired:

t.test(antes, depois, paired = TRUE)

**O teste-t de Student para uma amostra**

O teste-t de Student para uma amostra compara a medida média de um grupo com a média da população.

1. Primeira etapa é inserir os dados no R.

# medidas da amostra de 25 mulheres de minas gerais (dados fictícios)

altura <- c(152.0, 153.1, 154.6, 157.8,

 158.8, 159.6, 161.1, 161.6,

 162.7, 163.7, 164.1, 165.5,

 165.8, 168.4, 168.4, 169.1,

 169.1, 170.2, 172.4, 172.9,

 173.1, 173.3, 175.6, 176.9,

 179.0)

1. Segunda Etapa: o teste propriamente dito

Para fazer o teste usamos a função t.test() e incluímos como argumentos a variável com os dados altura e a média da população mu=160, com a qual desejamos fazer a comparação:

t.test(altura, mu= 160)

**A análise das notas de estatística entre homens e mulheres.**

homens <- c(15,9,7,13,10,11,14,8,12,5,10,10,6,6,5,13,12,5,12,6)

mulheres <- c(13,11,11,10,11,11,15,14,10,9,13,11,9,9,12,9,12,9,15,13,9,15,11,13,11,11,12,11,11,11)

E as médias de cada grupo:

mean(homens)

mean(mulheres)

A média das notas das mulheres foi maior. A questão estatística é: foi um mero acaso? Veja que como será usado um teste t independente, pois são **dois grupos** diferentes, não há problema no fato dos grupos terem tamanhos diferentes.

t.test(homens, mulheres)

# Testes de Médias

Uma outra família de testes estatísticos é testes de médias, amplamante conhecidos como ‘t-tests’. Este teste exige uma comparação entre os nossos dados e o valor ‘X’, então vamos encaminhar o vetor de atrasos para a função t.test(), especificando o argumento mu para o valor de comparação (X). E não esquecemos de usar tidy() no final para simplificar o resultado:

teste\_media <- banco\_de\_dados %>% [filter](https://rdrr.io/r/stats/filter.html)(var\_x!="nome\_da\_variavel") %>% pull(var\_y) %>%

 [t.test](https://rdrr.io/r/stats/t.test.html)(mu=13.4) %>%

 [tidy](https://generics.r-lib.org/reference/tidy.html)()

teste\_media

Para comunicar os resultados do t-test, é frequentemente útil gerar gráficos para mostrar a média e o intervalo de confiança. Como podemos construir um gráfico deste tipo? Colocamos a média dos nossos dados (um ponto com geom\_point) e uma linha horizontal para indicar o ponto de comparação (com geom\_hline que gera uma linha horizontal).

Agora, vamos adicionar uma linha que mostra o intervalo de confiança (as variáveis ‘conf.low’ e ‘conf.high’), com a geometria geom\_errorbar(), que exige três variáveis: x, ymin e ymax para definir os limites da linha baseado nos valores em nosso tibble. Observe se o intervalo de confiança de 95% não sobreposiciona o valor de X.

teste\_media %>% mutate(Variavel="Atraso na partida") %>%

 ggplot() +

 geom\_point(aes(x=Variavel, y=estimate)) +

 geom\_hline(yintercept=13.4, lty=2, color="blue")

# Comparando Médias

Que tal se não temos um ponto de comparação fixa, mas queremos comparar se a média em um grupo é diferente da média num outro grupo? Isso significa que dividimos os nossos dados, baseado numa variável categórica em nosso tibble, e comparamos as duas distribuições pela variável contínua.

Podemos continuar usando a função t.test, mas agora que temos que definir uma variável discreta e uma variável contínua, precisamos encaminhar o nosso tibble completo e não um vetor (uma variável única), então não precisamos do pull(). Em vez disso, e dado que estamos usando funções fora do tidyverse, temos que indicar em t.test que ela deve trabalhar com os dados que estamos encaminhando pelo pipe com o argumento data=.. (‘.’ significa usar dados produzidos no fluxo do pipe anterior).

Há mais um ajuste necessário: Vários testes, incluindo t.test, exigem que definimos esta análise usando uma fórmula, com a variável contínua na esquerda e a variável discreta (os grupos que queremos comparar) na direita: var\_Y ~ var\_x. A variável dependente sempre fica na esquerda, e está separada das variáveis independentes com o símbolo ~. Por exemplo:

dependente ~ independente

Podemos adicionar mais vaiáveis independentes com o símbolo ‘+’:

dependente ~ independente1 + independente2

Fórmulas são usadas em vários contextos, incluindo testes estatísticos e regressões.

Vamos realizar o teste para avaliar se a média de atrasos é ígual entre JFK e EWR:

Base\_de\_dados %>% [filter](https://rdrr.io/r/stats/filter.html)(var\_ind!="var\_x") %>%

 [t.test](https://rdrr.io/r/stats/t.test.html)(var\_y ~ var\_x, data=.) %>% [tidy](https://generics.r-lib.org/reference/tidy.html)()