

Análise dos dados da horta

Introdução

A horta comunitária é um projeto que envolve o plantio de vegetais para a comunidade. Existem dois tipos de hortas: a tradicional ao ar livre e a horta baseada em aquaponia.

A ideia de coletar dados da horta, como o tamanho das folhas de alface semanalmente, é realizar uma modelagem de crescimento e comparar os dois métodos. O objetivo é diferenciar estatisticamente os tratamentos para identificar qual proporciona o maior e melhor crescimento.

Após a coleta dos dados durante 5 semanas, organizamos em uma planilha do excel para facilitar o trabalho, chamada “horta”.

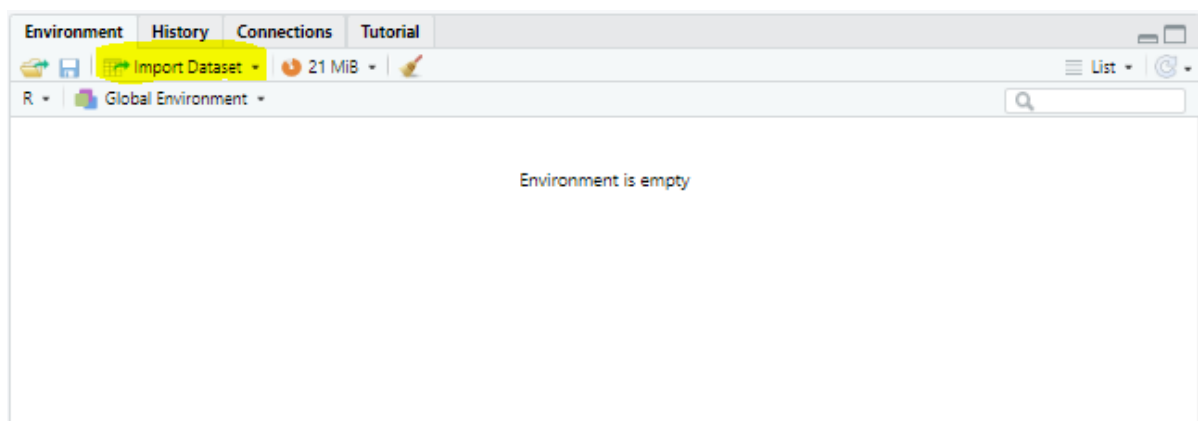
Análise dos dados

Para começar a discutir sobre os dados, é necessário primeiramente fazer uma descritiva geral dos dados, para ter uma ideia de como eles estão se comportando. Isso é feito de forma simples no programa RStudio, para isso vamos importar o nosso banco de dados.

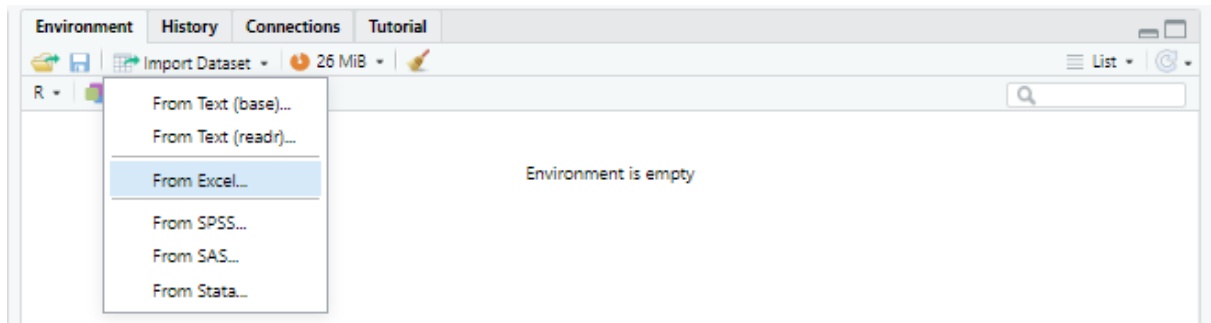
O banco de dados pode ser exportado de forma simples seguindo os seguintes passos:

Importando o banco de dados

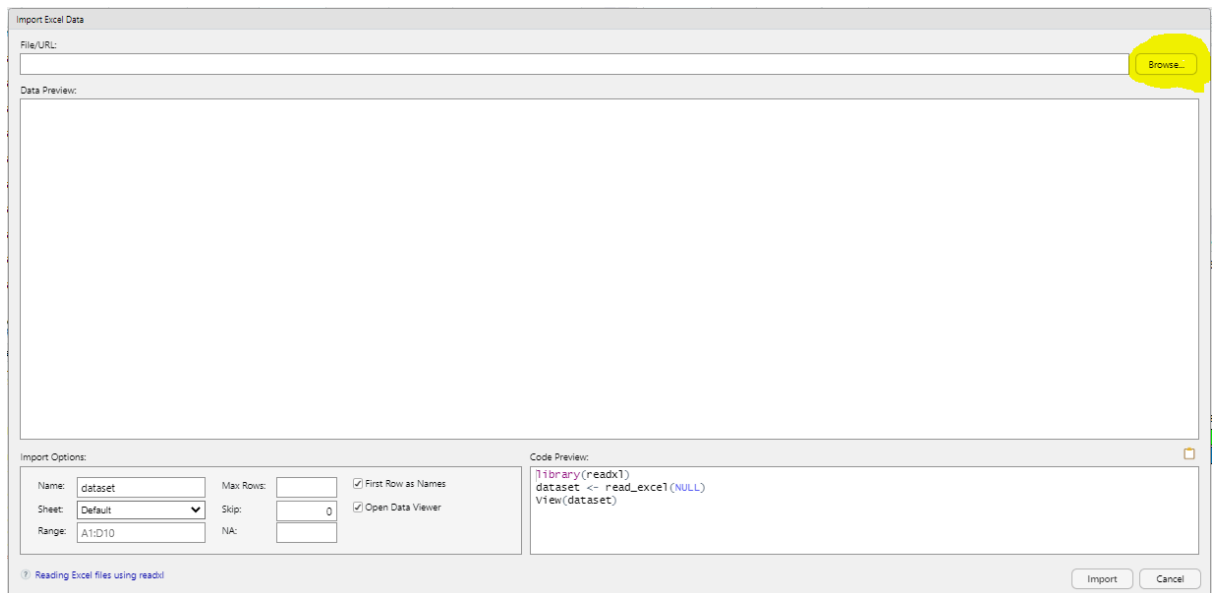
1 - Clique em “Import Dataset” no canto superior direito do RStudio:



2- Clique em “From excel...”:



3 - Na nova janela que aparecer, clique em “Browse” e procure o arquivo excel “horta”



4 - Após escolher o arquivo, escolha o nome do banco de dados em “Name”, nas nossas análises utilizamos “hortas”, se utilizar um nome diferente deste se atente ao digitar os comandos com o nome que utilizou. Digitando o nome, basta clicar em “Import” que o banco de dados estará disponível para análises, você estará vendo ele no canto superior direito do R com seu respectivo nome.

Import Excel Data

File/URL:
C:/Users/Jaqueline Macedo/Downloads/horta2.xlsx

Data Preview:

Medidas1 (double)	Medidas2 (double)	Medidas3 (double)	Medidas4 (double)	Medidas5 (double)	Medidas1A (double)	Medidas2A (double)	Medidas3A (double)	Medidas4A (double)	Medidas5A (double)
8.45	7.95	14.45	11.50	14.90	8.30	7.30	9.80	13.75	9.05
9.20	12.25	12.05	13.75	12.20	4.30	8.90	12.25	13.75	11.60
9.05	10.00	9.50	15.25	15.40	13.40	11.00	15.90	17.25	13.65
11.70	11.00	11.75	9.75	13.60	11.20	13.05	14.25	16.75	15.90
8.40	8.75	13.50	13.75	13.75	16.35	15.10	19.40	13.50	13.90
9.30	10.00	11.55	13.70	9.30	12.36	16.85	14.30	17.75	15.65
8.30	9.00	11.75	11.75	13.85	14.25	11.60	20.80	14.25	14.95
8.45	8.75	14.00	16.00	14.75	13.20	16.55	17.80	15.50	14.05
8.30	14.65	13.90	14.25	13.45	14.95	15.15	12.25	17.00	17.05
8.60	9.50	14.20	14.75	13.90	9.80	18.25	12.65	16.00	16.85
8.75	12.00	13.60	13.90	13.80	13.80	14.80	16.35	18.00	14.90
10.80	6.75	10.80	12.75	17.15	7.90	10.90	17.05	18.25	17.85
10.45	7.75	10.55	13.50	13.70	9.75	11.20	17.05	18.50	15.90
10.80	9.75	15.60	11.75	13.25	10.90	14.15	15.05	11.00	12.80
10.80	12.25	13.90	11.65	14.10	14.10	14.10	14.55	14.50	14.95
9.75	14.00	15.20	16.45	15.10	12.40	16.30	18.75	17.75	11.15

Import Options:

Name: hortas Max Rows: First Row as Names
Sheet: Default Skip: 0 Open Data Viewer
Range: A1:D10 NA:

Code Preview:

```
library(readxl)
hortas <- read_excel("horta2.xlsx")
view(hortas)
```

Import Cancel

5 - Clicando no nome “hortas”, será possível visualizar o banco de dados a ser utilizado:

	Medidas1	Medidas2	Medidas3	Medidas4	Medidas5	Medidas1A	Medidas2A	Medidas3A	Medidas4A	Medidas5A
1	8.45	7.95	14.45	11.50	14.90	8.30	7.30	9.80	13.75	9.05
2	9.20	12.25	12.05	13.75	12.20	4.30	8.90	12.25	13.75	11.60
3	9.05	10.00	9.50	15.25	15.40	13.40	11.00	15.90	17.25	13.65
4	11.70	11.00	11.75	9.75	13.60	11.20	13.05	14.25	16.75	15.90
5	8.40	8.75	13.50	13.75	13.75	16.35	15.10	19.40	13.50	13.90
6	9.30	10.00	11.55	13.70	9.30	12.36	16.85	14.30	17.75	15.65
7	8.30	9.00	11.75	11.75	13.85	14.25	11.60	20.80	14.25	14.95
8	8.45	8.75	14.00	16.00	14.75	13.20	16.55	17.80	15.50	14.05
9	8.30	14.65	13.90	14.25	13.45	14.95	15.15	12.25	17.00	17.05
10	8.60	9.50	14.20	14.75	13.90	9.80	18.25	12.65	16.00	16.85
11	8.75	12.00	13.60	13.90	13.80	13.80	14.80	16.35	18.00	14.90
12	10.80	6.75	10.80	12.75	17.15	7.90	10.90	17.05	18.25	17.85
13	10.45	7.75	10.55	13.50	13.70	9.75	11.20	17.05	18.50	15.90
14	10.80	9.75	15.60	11.75	13.25	10.90	14.15	15.05	11.00	12.80
15	10.80	12.25	13.90	11.65	14.10	14.10	14.10	14.55	14.50	14.95
16	9.75	14.00	15.20	16.45	15.10	12.40	16.30	18.75	17.75	11.15
17	9.50	11.75	15.00	15.25	15.15	13.10	10.95	19.50	16.75	14.75
18	9.70	10.75	15.75	13.25	15.55	11.15	11.60	19.00	16.25	16.00
19	8.75	12.00	13.45	13.50	15.30	12.50	11.90	17.60	13.25	17.70
20	8.25	12.50	13.40	15.00	15.10	13.20	13.60	14.15	18.25	18.95
21	7.00	11.00	12.25	11.75	14.45	11.70	12.25	12.60	10.90	16.70
22	8.25	11.25	13.75	11.00	10.55	10.90	15.15	17.90	17.75	9.80
23	9.50	12.75	12.90	13.00	10.25	14.45	15.90	18.25	19.90	17.40

Showing 1 to 15 of 40 entries, 10 total columns

Nesse banco de dados, temos todas as medidas da horta ao ar livre (Medidas1, Medidas2, Medidas3, Medidas4 e Medidas5), sendo que 1, 2, 3, 4 e 5 se referem as semanas, e as medidas da Aquaponia (Medidas1A, Medidas2A, Medidas3A, Medidas 4A e Medidas 5A).

Descritiva geral dos dados

Agora vamos começar a trabalhar com os dados, começando a fazer uma descritiva geral dos dados, isso pode ser realizado de forma simples com o seguinte comando:

Comando : **summary(hortas)**

```
> summary(hortas)
  Medidas1      Medidas2      Medidas3      Medidas4      Medidas5      Medidas1A
Min.   : 5.750   Min.   : 6.750   Min.   : 9.50    Min.   : 9.75    Min.   : 9.30   Min.   : 4.30
1st Qu.: 8.400   1st Qu.: 9.812   1st Qu.:12.25  1st Qu.:12.69   1st Qu.:12.79   1st Qu.: 9.80
Median : 9.100   Median :11.125   Median :13.60   Median :13.75   Median :13.88   Median :11.80
Mean   : 9.111   Mean   :11.082   Mean   :13.34   Mean   :13.65   Mean   :13.75   Mean   :11.56
3rd Qu.: 9.750   3rd Qu.:12.250   3rd Qu.:14.25   3rd Qu.:14.81   3rd Qu.:15.10   3rd Qu.:13.20
Max.   :11.700   Max.   :15.000   Max.   :15.75   Max.   :16.50   Max.   :17.15   Max.   :16.35
NA's   : 2       NA's   : 3       NA's   : 4       NA's   : 4       NA's   : 3

  Medidas2A      Medidas3A      Medidas4A      Medidas5A
Min.   : 7.30    Min.   : 9.80    Min.   :10.90   Min.   : 9.05
1st Qu.:11.90    1st Qu.:14.55   1st Qu.:15.00   1st Qu.:14.22
Median :14.50    Median :17.05   Median :17.00   Median :15.60
Mean   :14.00    Mean   :16.50   Mean   :16.29   Mean   :15.40
3rd Qu.:15.90    3rd Qu.:18.25   3rd Qu.:17.90   3rd Qu.:17.18
Max.   :18.25    Max.   :20.80   Max.   :20.00   Max.   :19.50
NA's   : 3       NA's   : 3       NA's   : 5       NA's   : 5
> |
```

Com esse comando, teremos as informações acima aparecendo no console, como o mínimo e máximo, ou seja, menor e maior tamanho para folha de alface, a média, mediana, etc. Olhando as descritivas gerais dos dados, é possível observar que:

- A média de tamanho de folhas de alface aumenta consideravelmente nas semanas 1-3, estabilizando na semana 4-5 para ambos os tratamentos.
- É possível dizer previamente que a média do tamanho das folhas de alface é maior na horta com aquaponia do que ao ar livre, porém, é necessário realizar testes adequados para confirmar essa informação.

Teste de normalidade, teste de variância e teste-t

Para confirmar que os tratamentos realmente diferem entre si na média de tamanho das folhas de alface, pode ser utilizado um teste-t de amostras independentes, que é utilizado quando os dados de um grupo não tem relação com os dados de outro grupo, ou seja, tratamentos diferentes. Posteriormente, também será realizado um teste-t de amostras dependentes, para verificar se há diferença significativa no tamanho das folhas de alface do mesmo tratamento de uma semana para outra.

O teste-t funciona com as seguintes hipóteses:

- **Hipótese Nula (H_0):** Não há diferença na média dos grupos.
- **Hipóteses Alternativa (H_a):** Há uma diferença na média dos grupos.

Antes de ir de fato para o test-t algumas condições devem ser verificadas, como:

1- Normalidade: As amostras devem ser aproximadamente normalmente distribuídas.

2- Variância homogênea (para teste-t de amostras independentes): As variâncias dos dois grupos devem ser aproximadamente iguais.

3- Independência: As observações devem ser independentes uma da outras.

Teste de normalidade

Começando com o teste-t de amostras independentes, vamos primeiro verificar se todos os dados seguem aproximadamente uma distribuição normal, e para isso iremos utilizar um teste muito utilizado, o teste **shapiro-will**, que funciona seguindo as seguintes hipóteses:

- **Hipótese Nula (H_0):** Os dados são normalmente distribuídos.
- **Hipótese Alternativa (H_a):** Os dados não são normalmente distribuídos.

No RStudio, para realizar o shapiro teste, iremos utilizar a seguinte função: **shapiro.test(nome_do_bando_de_dados\$coluna_a_ser_testada)**. Como queremos saber se todos os dados seguem uma distribuição normal, vamos ter que testar para as dez colunas existentes.

Comandos: **shapiro.test(hortas\$Medidas1)**

shapiro.test(hortas\$Medidas2)

shapiro.test(hortas\$Medidas1A)

shapiro.test(hortas\$Medidas2A)

shapiro.test(hortas\$Medidas3)

shapiro.test(hortas\$Medidas3A)

shapiro.test(hortas\$Medidas4)

shapiro.test(hortas\$Medidas4A)

shapiro.test(hortas\$Medidas5)

shapiro.test(hortas\$Medidas5A)

Rodando cada linha de comando separadamente, vamos obter a seguinte informação no console do R:

```
> shapiro.test(hortas$Medidas1)

      Shapiro-Wilk normality test

data:  hortas$Medidas1
W = 0.96954, p-value = 0.348
```

O que deve ser observado é o “*p-value*” (p-valor), o padrão do R é realizar testes a 5% de significância, ou seja, a hipótese nula só será rejeitada se o p-valor for menor que 0,05. Como o p-valor do nosso teste para as medidas da semana 1 é maior que 0,05 (0,348), a 5% de significância não rejeita-se H_0 , ou seja, os dados são normalmente distribuídos. Repita esse procedimento para as 5 semanas de cada tratamento, você verá que todas seguem uma distribuição normal a 5% de significância.

Teste de variância

Agora que sabemos que os dados seguem uma distribuição normal, podemos prosseguir com o teste de variância, que deve ser realizado posteriormente ao teste-t de amostras independentes, pois deve ser especificado no teste-t se as variâncias são homogêneas ou não.

Para isso, iremos utilizar o teste F de igualdade de variâncias, um teste de variância utilizado para verificar a homogeneidade das variâncias entre dois grupos ou mais grupos e seus testes de hipóteses são:

Hipótese Nula (H_0): As variâncias dos grupos são iguais ($\sigma^2 = \sigma^2$).

- **Hipóteses Alternativa (H_a):** Pelo menos uma das variâncias dos grupos é diferente ($\sigma^2 \neq \sigma^2$).

A função a ser utilizada segue a seguinte base `var.test(nome_do_banco_de_dados$Coluna1, nome_do_banco_de_dados$Coluna2)`, sendo que coluna1 e coluna 2 se refere às duas colunas a ser comparado as variâncias. Para comparar a variância de cada semana da horta livre com a aquaponia ficará da seguinte forma:

Comandos:

```
teste_de_variância = var.test(hortas$Medidas1, hortas$Medidas1A)
```

```
teste_de_variância2 = var.test(hortas$Medidas2, hortas$Medidas2A)
```

```
teste_de_variância3 = var.test(hortas$Medidas3, hortas$Medidas3A)
```

```
teste_de_variância4 = var.test(hortas$Medidas4, hortas$Medidas4A)
```

```
teste_de_variância5 = var.test(hortas$Medidas5, hortas$Medidas5A)
```

Ao rodar cada linha de comando, será criado um objeto que aparecerá no canto superior direito do RStudio, você irá clicar em cada objeto e observar seu p-valor.

Name	Type	Value
teste_de_variância	list [9] (S3: htest)	List of length 9
statistic	double [1]	0.2407329
parameter	integer [2]	39 36
p.value	double [1]	2.613673e-05
conf.int	double [2]	0.125 0.459
estimate	double [1]	0.2407329
null.value	double [1]	1
alternative	character [1]	'two.sided'
method	character [1]	'F test to compare two variances'
data.name	character [1]	'hortas\$Medidas1 and hortas\$Medidas1A'

Para o primeiro teste de variância, temos que o p-valor $< 0,05$, ou seja, a 5% de significância há evidências para rejeitar H_0 , portanto, as variâncias dos grupos não são iguais (heterogêneas). Verifique todas as outras, você verá que na semana 2 as variâncias são homogêneas (p-valor $> 0,05$), na 3 e na 4 heterogêneas e na semana 5 homogêneas.

Teste t de amostras independentes

Agora que já checamos todas as condições necessárias para se realizar um teste-t, podemos finalmente realizá-lo. Para isso iremos utilizar a função **t.test()**, que segue a seguinte sintaxe: **t.test(banco_de_dados\$Coluna1, banco_de_dados\$Coluna2, var.equal = TRUE/FALSE)**. As colunas 1 e 2 se referem as colunas que as médias serão comparadas, e “var.equal = TRUE/FALSE” diz se as variâncias são homogêneas (TRUE) ou heterogêneas (FALSE). Realizando o teste t para nossas amostras teremos as seguintes linhas de comando:

Comandos:

```
test_t = t.test(hortas$Medidas1, hortas$Medidas1A, var.equal = FALSE)
```

```
test_t2 = t.test(hortas$Medidas2, hortas$Medidas2A, var.equal = TRUE)
```

```
test_t3 = t.test(hortas$Medidas3, hortas$Medidas3A, var.equal = FALSE)
```

```
test_t4 = t.test(hortas$Medidas4, hortas$Medidas4A, var.equal = FALSE)
```

```
test_t5 = t.test(hortas$Medidas5, hortas$Medidas5A, var.equal = TRUE)
```

Rodando cada linha, será criado os 5 objetos com cada teste, abrindo o “test_t3” iremos verificar o p-valor:

Name	Type	Value
test_t3	list [10] (S3: htest)	List of length 10
statistic	double [1]	-6.446032
parameter	double [1]	56.57756
p.value	double [1]	2.725107e-08
conf.int	double [2]	-4.14 -2.18
estimate	double [2]	13.3 16.5
null.value	double [1]	0
stderr	double [1]	0.4899306
alternative	character [1]	'two.sided'
method	character [1]	'Welch Two Sample t-test'
data.name	character [1]	'hortas\$Medidas3 and hortas\$Medidas3A'

Relembrando as hipóteses do teste-t:

- **Hipótese Nula (H_0):** Não há diferença na média dos grupos.
- **Hipóteses Alternativa (H_a):** Há uma diferença na média dos grupos.

Portanto, como o nosso p-valor é menor que 0,05, podemos dizer a 5% de significância que o tamanho das folhas de alface da **semana 3 horta ao ar livre** e **semana 3 aquaponia** diferem entre si, e ao olhar as médias na descritiva dos dados, vemos que a da aquaponia é maior.

Como exercício, verifique se há diferença significativa entre as outras semanas, você verá que em todas as semanas os tratamentos diferem significativamente entre si.

Teste t pareado ar livre

Agora que sabemos que os tratamentos diferem entre si, outra análise que pode ser realizada é o teste-t pareado, onde iremos comparar amostras dependentes, ou seja, iremos observar o comportamento de um mesmo tratamento ao longo do tempo, nesse caso, se há diferença significativa na média de tamanho de folha de alface em todas as semanas.

Para isso iremos utilizar a função **t.test()**, a mesma utilizada anteriormente, porém, agora iremos obrigatoriamente colocar dentro dos parênteses “**paired = TRUE**”, indicando ao programa que será um teste-t pareado. Além disso, nesse caso não irá conter “**var.equal = TRUE**”, pois em um teste-t pareado não é necessário testar se as variâncias são homogêneas.

Comandos:

```
test_tp = t.test(hortas$Medidas1, hortas$Medidas2, paired = TRUE)
test_t2p= t.test(hortas$Medidas2, hortas$Medidas3, paired = TRUE)
test_t3p= t.test(hortas$Medidas3, hortas$Medidas4, paired = TRUE)
test_t4p= t.test(hortas$Medidas4, hortas$Medidas5, paired = TRUE)
```


Ao rodar cada linha o objeto com seu respectivo nome será criado e aparecerá no canto superior direito, abrindo o “test_tp”, você irá observar o seu p-valor:

Name	Type	Value
test_tp	list [10] (S3: htest)	List of length 10
statistic	double [1]	-4.691878
parameter	double [1]	37
p.value	double [1]	3.639392e-05
conf.int	double [2]	-2.69 -1.07
estimate	double [1]	-1.880263
null.value	double [1]	0
stderr	double [1]	0.4007485
alternative	character [1]	'two.sided'
method	character [1]	'Paired t-test'
data.name	character [1]	'hortas\$Medidas1 and hortas\$Medidas2'

Como podemos observar o p-valor é $< 0,05$, ou seja, há diferença significativa no tamanho da folha de alface entre a semana 1 e 2 (o alface cresceu). Agora observe o “test_t3p”, que compara a semana 3 com a 4, a semana onde seu tamanho começa a estabilizar.

Name	Type	Value
test_t3p	list [10] (S3: htest)	List of length 10
statistic	double [1]	-0.8367852
parameter	double [1]	35
p.value	double [1]	0.4083906
conf.int	double [2]	-1.01 0.42
estimate	double [1]	-0.2944444
null.value	double [1]	0
stderr	double [1]	0.3518758
alternative	character [1]	'two.sided'
method	character [1]	'Paired t-test'
data.name	character [1]	'hortas\$Medidas3 and hortas\$Medidas4'

Nessas duas semanas (3-4), não há evidências a 5% de significância para se rejeitar H_0 , portanto, não há diferença significativa entre as médias de tamanho das duas semanas.

Verificando as outras, será possível observar que nas semanas 1-2 e 2-3 tem diferença significativa, enquanto que nas semanas 3-4 e 4-5, não há diferença significativa, o que se era esperado, pois chega um momento que o alface para de crescer.

Teste t pareado aquaponia

Como já fizemos para a horta ao ar livre, fica fácil de fazer para aquaponia, basta copiar e colar os comandos utilizados para testar ao ar livre e mudar as colunas analisadas para aquaponia, sem esquecer de mudar o nome do objeto a ser criado.

Comandos:

```
test_tpA = t.test(hortas$Medidas1A, hortas$Medidas2A, paired = TRUE)
test_t2pA = t.test(hortas$Medidas2A, hortas$Medidas3A, paired = TRUE)
test_t3pA = t.test(hortas$Medidas3A, hortas$Medidas4A, paired = TRUE)
test_t4pA = t.test(hortas$Medidas4A, hortas$Medidas5A, paired = TRUE)
```

Rodando os comandos, vamos então observar o p-valor nos objetos criados, abra o objeto que compara a semana 2 com a semana 3 (test_t2pA):

Name	Type	Value
test_t2pA	list [10] (S3: htest)	List of length 10
statistic	double [1]	-4.618266
parameter	double [1]	36
p.value	double [1]	4.79379e-05
conf.int	double [2]	-3.6 -1.4
estimate	double [1]	-2.504054
null.value	double [1]	0
stderr	double [1]	0.5422065
alternative	character [1]	'two.sided'
method	character [1]	'Paired t-test'
data.name	character [1]	'hortas\$Medidas2A and hortas\$Medidas3A'

Como o p-valor é $< 0,05$, há diferença significativa entre o tamanho das folhas de alface a 5% de significância entre a semana 2 e 3, ou seja, o alface cresceu (já que a média da semana 3 é maior que a média da semana 2). Agora vamos testar para a semana 3 e 4 (test_t3pA):

Name	Type	Value
test_t3pA	list [10] (S3: htest)	List of length 10
statistic	double [1]	0.02945218
parameter	double [1]	34
p.value	double [1]	0.9766762
conf.int	double [2]	-0.971 1.000

Agora o p-valor é $> 0,05$, portanto, não há diferença significativa a 5% de significância entre o tamanho das folhas de alface nas semanas 3 e 4, então é possível concluir que nos dois tratamentos os

alfaces estabilizaram os seus tamanhos nas mesmas semanas. Quando for interpretar as outras semanas (1-2 e 4-5), verá que a 1-2 terá diferença significativa, enquanto a 4-5 não.

Elaboração gráfica

Com todas as análises feitas, podemos agora visualizar essas diferenças graficamente utilizando o *dplyr* e o *ggplot2*. Vamos então começar elaborando um gráfico somente da primeira semana, comparando os dois tratamentos.

Para isso, iremos precisar utilizar primeiramente o *dplyr*, que pode ser instalado e requerido com os seguintes comandos:

```
install.packages("dplyr")  
library(dplyr)
```

Ao rodar o primeiro comando, aguarde o programa instalar o pacote, assim que instalado, pode rodar a linha que carrega o pacote **"library(dplyr)"**.

Vamos começar organizando os nossos dados de forma que crie uma tabela somente com a semana 1 horta livre e aquaponia:

Organizando os dados em uma nova tabela

Comandos:

```
Semana1 <- data.frame( # Cria um objeto chamado "semana1" e cria um data frame.  
Grupo = rep(c("Semana1", "Semana1A"), each = nrow(hortas)), # Cria uma coluna  
chamada "Grupo", contendo a quantidade de Semana1 e Semana1A do banco de dados "hortas"  
Valor = c(hortas$Medidas1, hortas$Medidas1A) # Cria uma coluna chamada "Valor", com  
o valor de cada medida de cada semana, do bando de dados "hortas"  
)
```

Rodando esses comandos, teremos o objeto *Semana1* contendo a seguinte tabela:

	Grupo	Valor
1	Semana1	8.45
2	Semana1	9.20
3	Semana1	9.05
4	Semana1	11.70
5	Semana1	8.40
6	Semana1	9.30
7	Semana1	8.30
8	Semana1	8.45
9	Semana1	8.30
10	Semana1	8.60
11	Semana1	8.75
12	Semana1	10.80
13	Semana1	10.45
14	Semana1	10.80
15	Semana1	10.80
16	Semana1	9.75
17	Semana1	9.50
18	Semana1	9.70

Gráfico de barras

Com a tabela criada, agora podemos ir para o gráfico. Para isso iremos instalar e carregar o pacote **ggplot2**:

```
install.packages("ggplot2")
library(ggplot2)
```

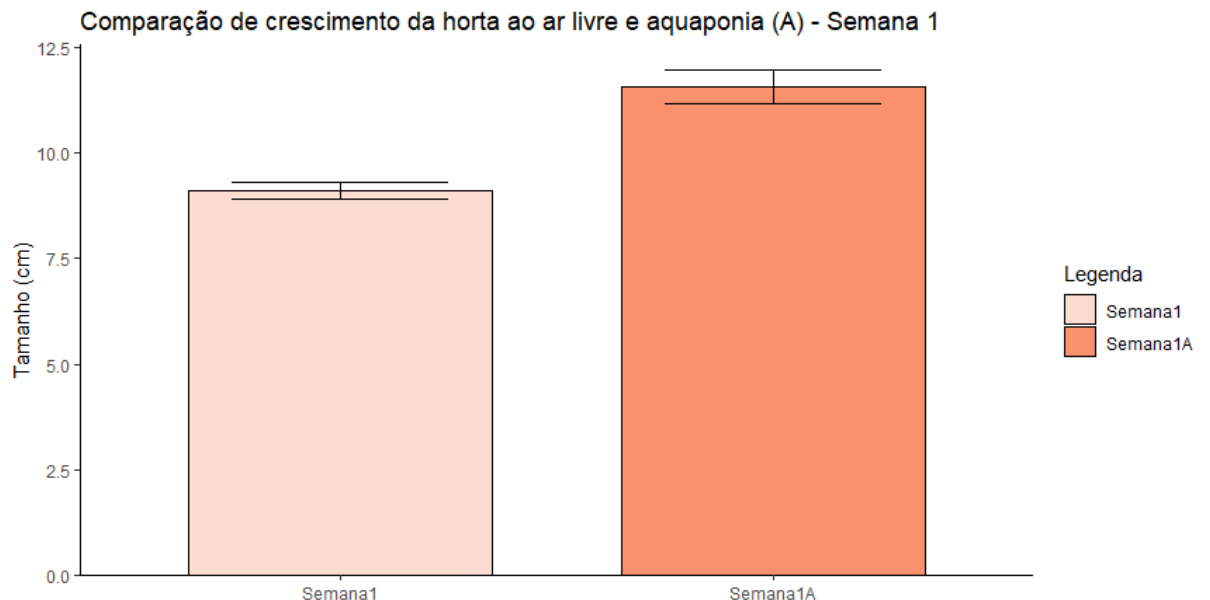
Com o programa instalado e carregado podemos começar a elaborar o gráfico:

Comandos:

```
ggplot(Semana1, aes(x = Grupo, y = Valor, fill = Grupo)) + # Chama a tabela "Semana1", definindo
  o eixo "x" sendo a coluna "Grupo" e o eixo "y" a coluna Valor, além de dizer para organizar pelo
  "Grupo"
geom_bar(stat = "summary", position = "dodge", fun = "mean", width = 0.7, color = "black")
+ # Comando que faz um gráfico de barras, definindo que será pela média, definindo a cor da borda da
  barra e seu tamanho
labs(title = "Comparação de crescimento da horta ao ar livre e aquaponia (A) - Semana 1",
  y = "Tamanho (cm)", fill = "Legenda", x = "") + # Define título, nome dos eixos e da legenda.
  scale_fill_brewer(palette = "Reds") + # Define a cor em paletas.
  scale_y_continuous(expand = expansion(mult = c(0, 0.05))) + # Ajusta a barra no eixo x
geom_errorbar(stat = "summary", fun.data = "mean_se", width = 0.5) + # Coloca a barra de
  erro no gráfico
theme_classic() # Define o plano de fundo do gráfico
```

Note que o “+” é sempre utilizado para interligar funções diferentes, facilitando a organização e programação.

Rodando esses comandos teremos o seguinte gráfico de barras:



Há diversos temas e paletas de cores para escolher, **alguns temas** são: `theme_gray()`, `theme_bw()`, `theme_linedraw()`, `theme_light()`, `theme_dark()`, `theme_minimal()`, `theme_classic()`, `theme_void()`.

Algumas **paletas de cores**: Blues, BuGn, BuPu, GnBu, Greens, Greys, Oranges, OrRd, PuBu, PuBuGn, PuRd, Purples, RdPu, Reds, YlGn, YlGnBu, YlOrBr, YlOrRd, BrBG, PiYG, PRGn, PuOr, RdBu, RdGy, RdYlBu, RdYlGn, Spectral, Accent, Dark2, Paired, Pastel1, Pastel2, Set1, Set2, Set3.

Vá testando e veja qual mais te atrai!

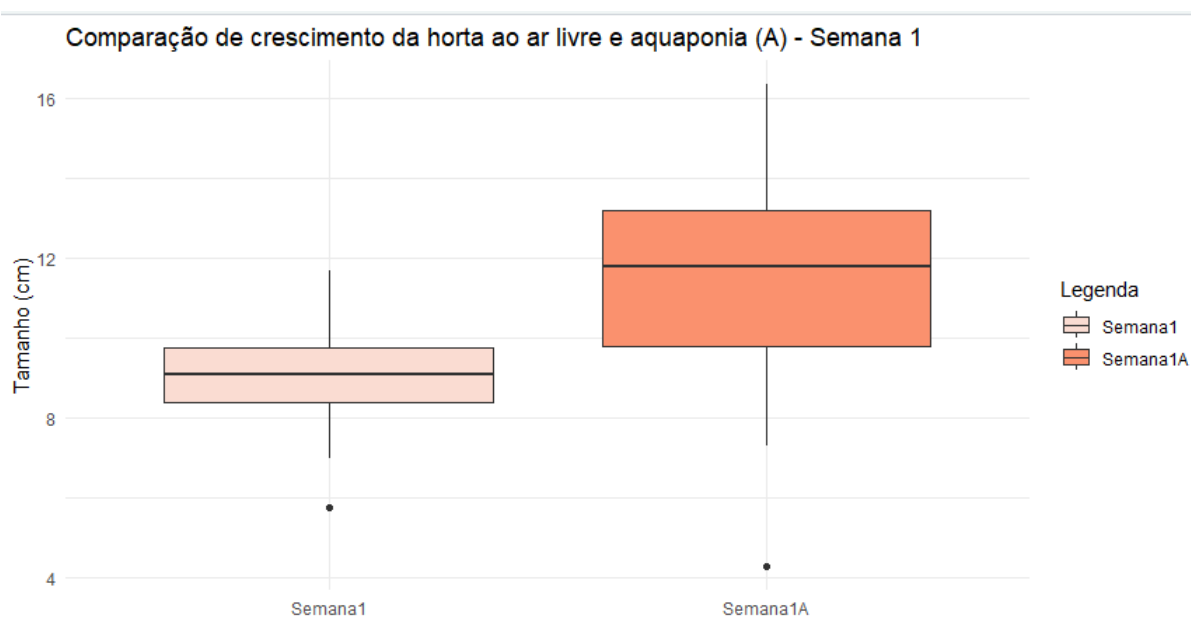
Box-plot

Também é interessante fazer um box-plot por grupos, possibilitando a comparação dos dois tratamentos:

Comandos:

```
ggplot(Semana1, aes(x = Grupo, y = Valor, fill = Grupo)) + # Definindo o eixo "x" sendo a
coluna "Grupo" e o eixo "y" a coluna Valor, além de dizer para organizar pelo "Grupo"
  geom_boxplot() + # Comando que faz o boxplot
labs(title = "Comparação de crescimento da horta ao ar livre e aquaponia (A) - Semana 1 ",
  x = "", y = "Tamanho (cm)", fill = "Legenda") + # Definindo nome dos eixos etc
  scale_fill_brewer(palette = "Reds") + # Define as cores por paleta
  theme_minimal() # Define o plano de fundo
```

Rodando os comandos, temos o seguinte boxplot:



Agora que fizemos para a primeira semana, fica simples modificar os comandos e fazer para as próximas semanas, basta modificar primeiro o comando da tabela:

Comandos:

```
Semana2 <- data.frame(
  Grupo = rep(c("Semana2", "Semana2A"), each = nrow(hortas)),
  Valor = c(hortas$Medidas2, hortas$Medidas2A)
)
```

*** O que está em negrito é o que foi modificado**

```
E para o gráfico é mais simples ainda, basta modificar qual tabela ele está puxando:
ggplot(Semana2, aes(x = Grupo, y = Valor, fill = Grupo)) +
  geom_bar(stat = "summary", position = "dodge", fun = "mean", width = 0.7, color = "black") +
  labs(title = "Comparação de crescimento da horta ao ar livre e aquaponia (A) - Semana 2",
        y = "Tamanho (cm)", fill = "Legenda", x = "") +
  scale_fill_brewer(palette = "Blues") +
  scale_y_continuous(expand = expansion(mult = c(0, 0.05))) +
  geom_errorbar(stat = "summary", fun.data = "mean_se", width = 0.5) +
  theme_classic()
```

Gráfico de barras com todas as semanas

Para uma visualização geral dos dados podemos também fazer um gráfico de barras com todas as semanas, mas para isso vamos precisar organizar os dados de uma forma totalmente diferente, renomeando também as colunas:

Comandos:

```
dados <- hortas %>% # Começa a transformação dos dados com o pipe (%>%) do pacote
dplyr
pivot_longer(cols = everything(), # Converte os dados de formato largo para longo
names_to = "Semana", # Nomeia a nova coluna de nomes de variáveis como
"Semana"
values_to = "Tamanho") %>% # Nomeia a nova coluna de valores como
"Tamanho"
mutate(
Tratamento = ifelse(str_detect(Semana, "AS"), "Aquaponia", "Horta ao Ar Livre"),
# Cria a coluna "Tratamento" com base no sufixo "A" nas colunas originais
Semana = str_remove(Semana, "A"), # Remove o sufixo "A" dos nomes das colunas
para obter apenas "MedidasX"
Semana = case_when( # Reclassifica os nomes das semanas para valores mais descritivos
Semana == "Medidas1" ~ "Semana 1", # Substitui "Medidas1" por "Semana 1"
Semana == "Medidas2" ~ "Semana 2", # Substitui "Medidas2" por "Semana 2"
Semana == "Medidas3" ~ "Semana 3", # Substitui "Medidas3" por "Semana 3"
Semana == "Medidas4" ~ "Semana 4", # Substitui "Medidas4" por "Semana 4"
Semana == "Medidas5" ~ "Semana 5" # Substitui "Medidas5" por "Semana 5"
)
)
```

Rodando esses comandos teremos uma nova tabela:

	Semana	Tamanho	Tratamento
1	Semana 1	8.45	Horta ao Ar Livre
2	Semana 2	7.95	Horta ao Ar Livre
3	Semana 3	14.45	Horta ao Ar Livre
4	Semana 4	11.50	Horta ao Ar Livre
5	Semana 5	14.90	Horta ao Ar Livre
6	Semana 1	8.30	Aquaponia
7	Semana 2	7.30	Aquaponia
8	Semana 3	9.80	Aquaponia
9	Semana 4	13.75	Aquaponia
10	Semana 5	9.05	Aquaponia
11	Semana 1	9.20	Horta ao Ar Livre
12	Semana 2	12.25	Horta ao Ar Livre
13	Semana 3	12.05	Horta ao Ar Livre
14	Semana 4	13.75	Horta ao Ar Livre
15	Semana 5	12.20	Horta ao Ar Livre

Antes de começarmos a fazer o gráfico vamos organizar em níveis os dados, fazendo com que na hora de fazer o gráfico de barras o tratamento “horta ao ar livre” apareça primeiro.

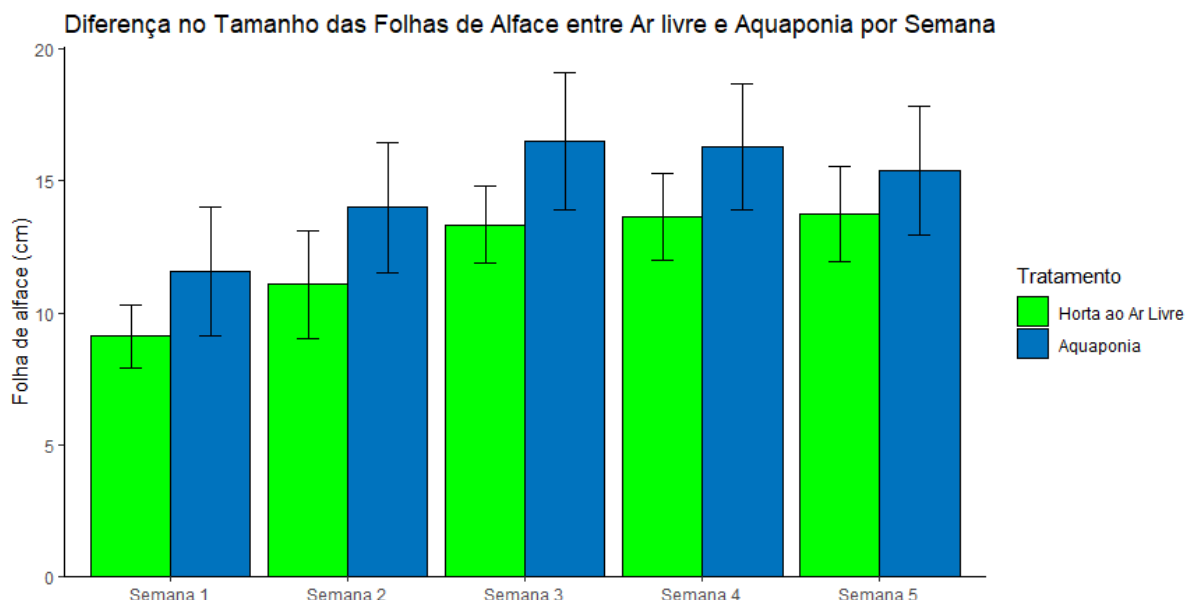
Comando:

```
dados$Tratamento <- factor(dados$Tratamento, levels = c("Horta ao Ar Livre", "Aquaponia"))
```

Pronto!! Agora vamos começar a elaborar o gráfico:

```
ggplot(dados, aes(x = Semana, y = Tamanho, fill = Tratamento)) +  
  geom_bar(stat = "summary", position = "dodge", fun = "mean", color = "black") +  
  labs(title = "Diferença no Tamanho das Folhas de Alface entre Ar livre e Aquaponia por  
Semana",  
    x = "",  
    y = "Folha de alface (cm)",  
    fill = "Tratamento") +  
  theme_classic() +  
  geom_errorbar(stat = "summary", position = position_dodge(0.9), fun.data = mean_sdl,  
    fun.args = list(mult = 1), width = 0.25) +  
  scale_fill_manual(values = c("green", "#0077BE")) + # Nesse estamos escolhendo as cores  
manualmente  
  scale_y_continuous(expand = expansion(mult = c(0, 0.05)))
```

Rodando temos:



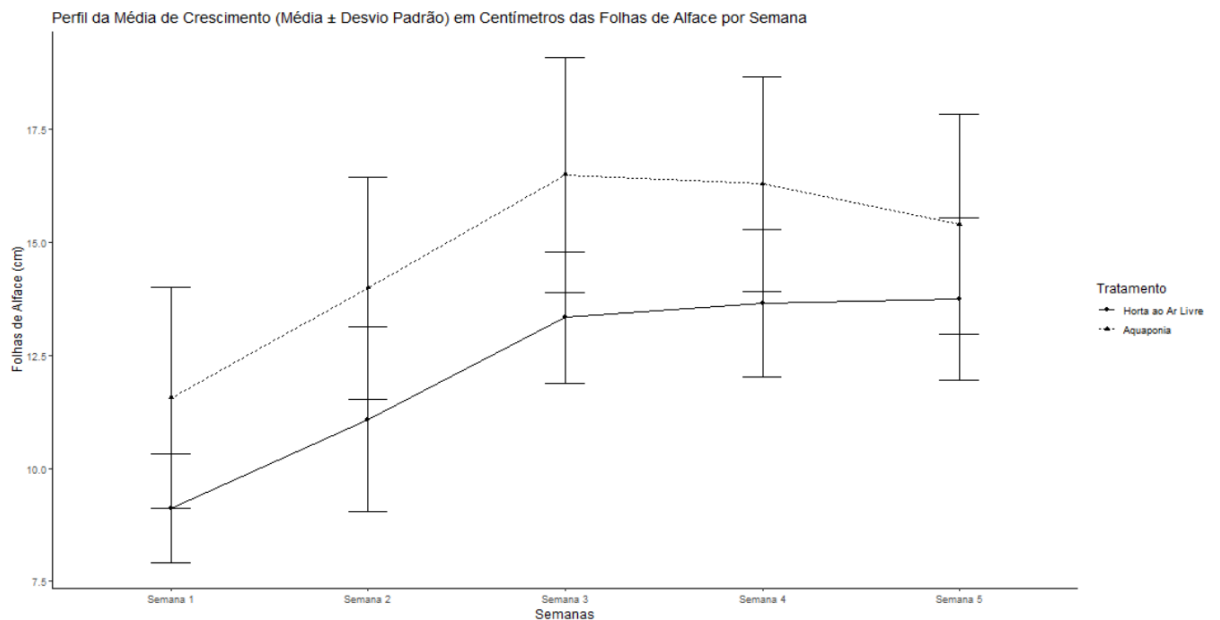
Agora temos uma visualização geral dos dados mostrando tudo o que vimos anteriormente, temos um crescimento nas 3 primeiras semanas e uma estabilização na terceira pra frente.

Perfil de média

Também é possível fazer um gráfico de perfil de média, possibilitando uma visualização melhor ainda do comportamento de crescimento dos alfaces:

Comandos:

```
ggplot(dados, aes(x = Semana, y = Tamanho, group = Tratamento, fill = Tratamento)) +  
  stat_summary(fun = mean, geom = "line", aes(linetype = Tratamento)) + # Adiciona  
linhas representando a média do tamanho para cada tratamento, diferenciadas por tipo de linha  
  stat_summary(fun = mean, geom = "point", aes(shape = Tratamento)) + # Adiciona  
pontos representando a média do tamanho para cada tratamento, diferenciados por forma  
  theme_classic() +  
  stat_summary(fun.data = mean_sdl, fun.args = list(mult = 1),  
    geom = "errorbar", width = 0.2) + # barra de erro  
  labs(title = "Perfil da Média de Crescimento (Média ± Desvio Padrão) em Centímetros  
das Folhas de Alface por Semana",  
    x = "Semanas",  
    y = "Folhas de Alface (cm)",  
    color = "Tratamento",  
    linetype = "Tratamento",  
    shape = "Tratamento")
```



Modelo de Gompertz

O ideal é aplicar um modelo de curva nos dados, pois os modelos de curva permitem fazer previsões para valores não observados, pode ajudar a identificar tendências e padrões nos dados que não são visíveis apenas pela comparação de médias. Isso pode incluir a detecção de pontos de inflexão, sazonalidades e efeitos de longo prazo. Além de testar se há uma diferença significativa entre grupos, o modelo de curva proporciona uma forma mais intuitiva e visual de comunicar os resultados, pois uma curva bem ajustada pode ilustrar de forma clara e impactante as conclusões da análise.

O modelo de Gompertz é um modelo simples que descreve o crescimento de uma variável no tempo. Esse modelo é utilizado em várias áreas, especialmente quando o crescimento apresenta uma fase inicial rápida que diminui ao longo do tempo, aproximando-se de um valor limite. Podem ser expressos pela equação:

$$y(t) = A \cdot \exp(-b \cdot \exp(-c \cdot t))$$

onde:

$y(t)$ é a variável de interesse (crescimento/tamanho) no tempo t

A = valor máximo de tamanho da folha de alface.

b = afeta a posição do ponto de inflexão.

c = taxa de crescimento inicial.

Como não possuímos os valores dos parâmetros b e c , vamos assumir valores padrão razoáveis, pois seus valores reais serão ajustados no resultado final do modelo. Para isso consideramos b inicial igual a 1 e c inicial igual a 0,1. O parâmetro A será o tamanho máximo observado na horta ao ar livre e na aquaponia.

Para aplicar a modelagem, primeiro importarmos o banco de dados, agora o outro excel chamado “horta1”, alterando as colunas “Tempo” e “Tamanho” para numérico.

Data Preview:

Tratamento (character)	Tempo (numeric)	ID (double)	Tamanho (numeric)
Ar livre	1	1	8.45
Ar livre	1	2	9.20
Ar livre	1	3	9.05
Ar livre	1	4	11.70
Ar livre	1	5	8.40
Ar livre	1	6	9.30
Ar livre	1	7	8.30
Ar livre	1	8	8.45

Previewing first 50 entries.

Import Options:

Name: dados

Max Rows:

☒ First Row as Names

Sheet: Default

Skip: 0

☒ Open Data Viewer

Range: A1:D10

NA:

Code Preview:

```
library(readxl)
dados <- read_excel("horta1.xlsx", col_types = c("text",
"numeric", "numeric", "numeric"))
view(dados)
```

? Reading Excel files using readxl

Import

Cancel

Em seguida, vamos separar os dados em dois grupos: ar livre e aquaponia. Para isso, criamos um novo subconjunto de dados chamado “dados_horta” que contém apenas as observações onde o tratamento é “Ar livre”, e subconjunto chamado “dados_aquaponia” que contém apenas as observações onde o tratamento é “Aquaponia”

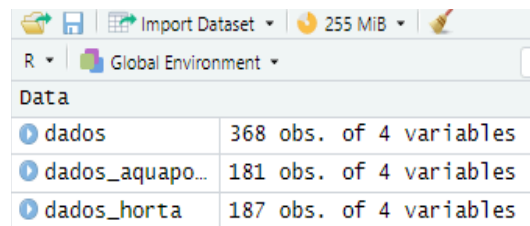
Assim, chamamos o dataframe original e aplicamos um filtro (função do pacote dplyr) para incluir apenas as linhas onde a coluna Tratamento tem o valor "Ar livre" ou "Aquaponia", como é observado no comando abaixo:

Separar os dados por grupo

dados_horta <- dados %>% filter(Tratamento == "Ar livre")

dados_aquaponia <- dados %>% filter(Tratamento == "Aquaponia")

```
# Separar os dados por grupo
dados_horta <- dados %>% filter(Tratamento == "Ar livre")
dados_aquaponia <- dados %>% filter(Tratamento == "Aquaponia")
```



Data	
dados	368 obs. of 4 variables
dados_aquapo...	181 obs. of 4 variables
dados_horta	187 obs. of 4 variables

Após isso, vamos definir os valores iniciais dos parâmetros. Os parâmetros b e c serão iguais para os dois tratamentos, já o parâmetro A será o valor máximo do tamanho da folha de alface de cada tratamento.

Para isso utilizou-se o comando:

A_inicial_horta: Este é o nome da variável que vai armazenar o valor inicial para o parâmetro A para o modelo ajustado aos dados da horta.

max(dados_horta\$Tamanho): Esta parte do código calcula o valor máximo observado na coluna Tamanho do dataframe “dados_horta”. O mesmo processo deve ser feito para a aquaponia comando:

Escolher valores iniciais para os parâmetros

A_inicial_horta <- max(dados_horta\$Tamanho) # Máximo observado para horta

A_inicial_aquaponia <- max(dados_aquaponia\$Tamanho) # Máximo observado para aquaponia

b_inicial <- 1

c_inicial <- 0.1

```
# Escolher valores iniciais para os parâmetros
A_inicial_horta <- max(dados_horta$Tamanho) # Máximo observado para horta
A_inicial_aquaponia <- max(dados_aquaponia$Tamanho) # Máximo observado para aquaponia
b_inicial <- 1
c_inicial <- 0.1
```

values	
A_inicial_aq...	20.8
A_inicial_ho...	17.15
b_inicial	1
c_inicial	0.1

Agora sim podemos aplicar a modelagem, para isso basta utilizar o comando abaixo. Para observar os resultados usamos os summary.

```
# Ajuste do modelo de Gompertz para horta
modelo_horta <- nls(Tamanho ~ A * exp(-b * exp(-c * Tempo)),
  data = dados_horta,
  start = list(A = A_inicial_horta, b = b_inicial, c = c_inicial))

# Ajuste do modelo de Gompertz para aquaponia
modelo_aquaponia <- nls(Tamanho ~ A * exp(-b * exp(-c * Tempo)),
  data = dados_aquaponia,
  start = list(A = A_inicial_aquaponia, b = b_inicial, c = c_inicial))

# Resumo dos modelos ajustados
summary(modelo_horta)
summary(modelo_aquaponia)
```

```
Formula: Tamanho ~ A * exp(-b * exp(-c * Tempo))

Parameters:
  Estimate Std. Error t value Pr(>|t|)
A  14.2745    0.4073  35.050 < 2e-16 ***
b   0.9779    0.1417   6.901 8.08e-11 ***
c   0.7527    0.1524   4.940 1.75e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.666 on 184 degrees of freedom

Number of iterations to convergence: 14
Achieved convergence tolerance: 9.592e-06

> summary(modelo_aquaponia)

Formula: Tamanho ~ A * exp(-b * exp(-c * Tempo))

Parameters:
  Estimate Std. Error t value Pr(>|t|)
A  16.1569    0.3643  44.347 < 2e-16 ***
b   1.1313    0.3992   2.834 0.005128 **
c   1.1904    0.3488   3.413 0.000796 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.519 on 178 degrees of freedom

Number of iterations to convergence: 28
Achieved convergence tolerance: 6.218e-06
```

Em ambos, a alta significância dos parâmetros e o baixo erro padrão residual sugerem que o modelo de Gompertz é um bom ajuste para descrever o crescimento dos alfaces na horta na aquaponia.

O tamanho máximo esperado dos alfaces na aquaponia é maior do que na horta (16,16cm na aquaponia e 14,27m no ar livre). A taxa inicial de crescimento (parâmetro c) também é mais rápida na aquaponia, o qual marca 1.19 enquanto a taxa de crescimento inicial no ar livre é de 0.75.

Em seguida iremos fazer as predições do modelo ajustado, isso permite ver como os modelos de Gompertz ajustados se comportam em relação aos dados observados. Isso não só ajuda a validar a adequação do modelo, mas também pode ser usado para fazer previsões futuras de crescimento com base nos parâmetros estimados. Somente após a predição é possível realizar a visualização gráfica. Observe o comando abaixo:

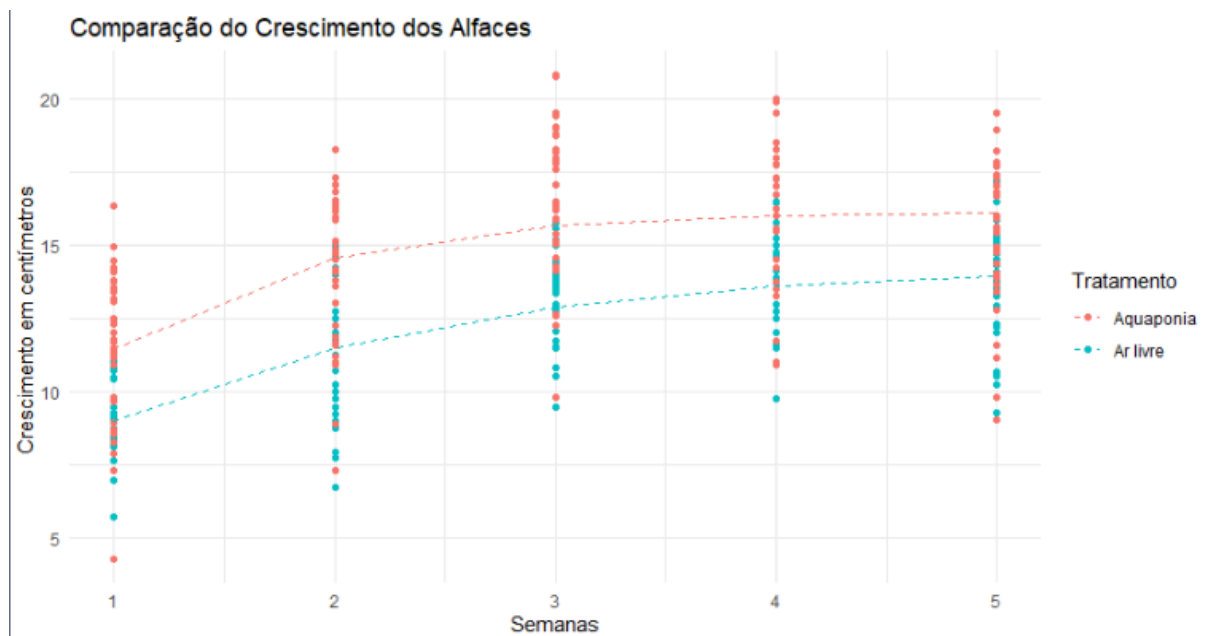
```

# Predições dos modelos ajustados
dados_horta$pred <- predict(modelo_horta, newdata = dados_horta)
dados_aquaponia$pred <- predict(modelo_aquaponia, newdata = dados_aquaponia)

# Combinar os dados preditos em um único dataframe
dados_pred <- rbind(dados_horta, dados_aquaponia)

# Plot dos dados e dos modelos ajustados
ggplot(dados_pred, aes(x = Tempo, y = Tamanho, color = Tratamento)) +
  geom_point() + # Dados observados
  geom_line(aes(y = pred), linetype = "dashed") + # Modelos ajustados
  labs(title = "Comparação do Crescimento dos Alfaces", x = "Semanas", y =
"Crescimento em centímetros ") +
  theme_minimal()

```

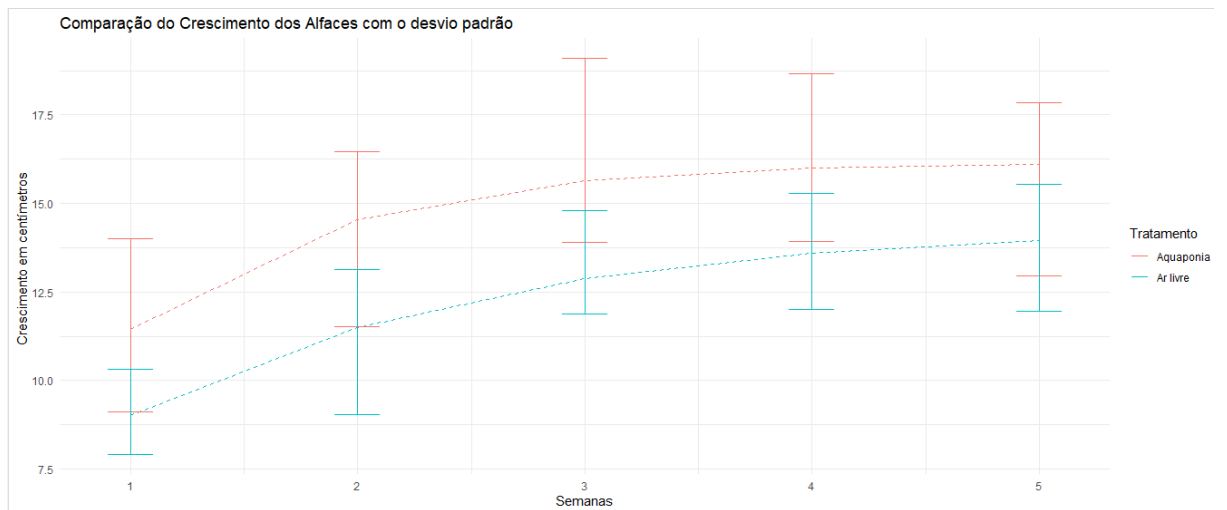


Caso queira fazer o gráfico sem pontos, e adicionar o desvio padrão, basta utilizar o comando abaixo:

```

# Plot dos dados e dos modelos ajustados
ggplot(dados_pred, aes(x = Tempo, y = Tamanho, color = Tratamento)) +
  geom_line(aes(y = pred), linetype = "dashed") + # Modelos ajustados
  stat_summary(fun.data = mean_sdl, fun.args = list(mult = 1),
    geom = "errorbar", width = 0.2) + # barra de erro
  labs(title = "Comparação do Crescimento dos Alfaces com o desvio padrão", x =
"Semanas", y = "Crescimento em centímetros ") +
  theme_minimal()

```



No tratamento de Aquaponia (linha vermelha), a barra de desvio padrão é mais longa, indicando uma maior variabilidade no crescimento das alfaces, já no tratamento de Ar livre (linha azul), a barra de desvio padrão também é longa, indicando uma variabilidade considerável, embora ligeiramente menor que no tratamento de Aquaponia.