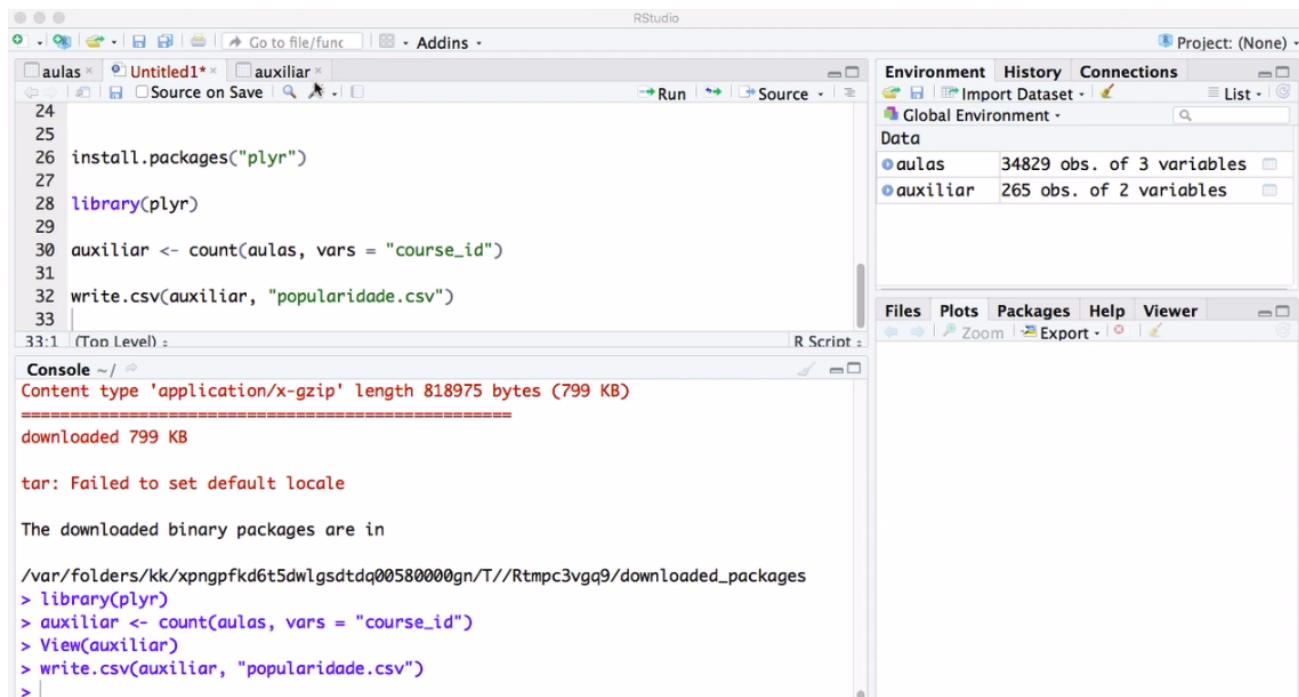


01

Criando gráficos

Transcrição

Trabalharemos com os **cursos** para descobrir a duração deles até a conclusão. Para isso, utilizaremos um novo banco de dados, mas antes limparemos a área de trabalho, que está com todas as informações da sessão anterior, para evitar confusão:



```

24
25
26 install.packages("plyr")
27
28 library(plyr)
29
30 auxiliar <- count(aulas, vars = "course_id")
31
32 write.csv(auxiliar, "popularidade.csv")
33
33:1 (Top Level) :

```

Console ~/

```

Content type 'application/x-gzip' length 818975 bytes (799 KB)
=====
downloaded 799 KB

tar: Failed to set default locale

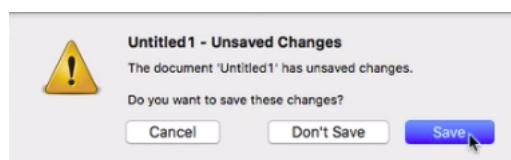
The downloaded binary packages are in

/var/folders/kk/xpnpfkd6t5dwlgstdq00580000gn/T//Rtmpc3vgq9 downloaded_packages
> library(plyr)
> auxiliar <- count(aulas, vars = "course_id")
> View(auxiliar)
> write.csv(auxiliar, "popularidade.csv")
> |

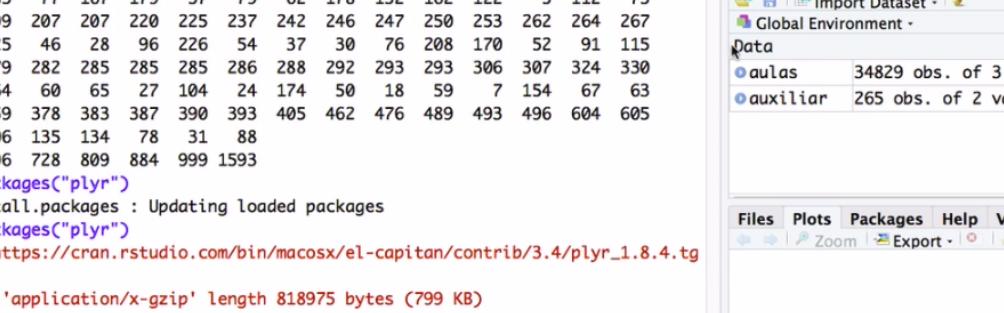
```

Para isso, fecharemos as abas de visualização:

- do objeto `auxiliar`;
- de `aulas`;
- o arquivo de R Script, que ao fecharmos, abrirá uma mensagem questionando se desejamos salvar.



Salvaremos como `script1` com a extensão `.r`, que poderá ser aberta tanto no R quanto no RStudio, ou até mesmo em alguns editores de texto. Após fecharmos esses arquivos, a janela de Script ficará oculta.



The screenshot shows the RStudio interface with the following details:

- Console:** Displays a table of 20 numerical values and a series of R command outputs. The table values are: 118, 137, 85, 77, 167, 179, 57, 79, 62, 178, 152, 162, 122, 5, 112, 75, 196, 197, 199, 207, 207, 220, 225, 237, 242, 246, 247, 250, 253, 262, 264, 267, 72, 116, 125, 46, 28, 96, 226, 54, 37, 30, 76, 208, 170, 52, 91, 115, 268, 270, 279, 282, 285, 285, 285, 286, 288, 292, 293, 293, 306, 307, 324, 330, 47, 6, 64, 60, 65, 27, 104, 24, 174, 50, 18, 59, 7, 154, 67, 63, 336, 338, 359, 378, 383, 387, 390, 393, 405, 462, 476, 489, 493, 496, 604, 605, 34, 58, 106, 135, 134, 78, 31, 88, 609, 653, 696, 728, 809, 884, 999, 1593.
- Environment:** Shows the global environment with two data frames: `aulas` (34829 obs. of 3 variables) and `auxiliar` (265 obs. of 2 variables).
- Output:** Shows the results of R commands, including package installation, URL download, and tar errors.

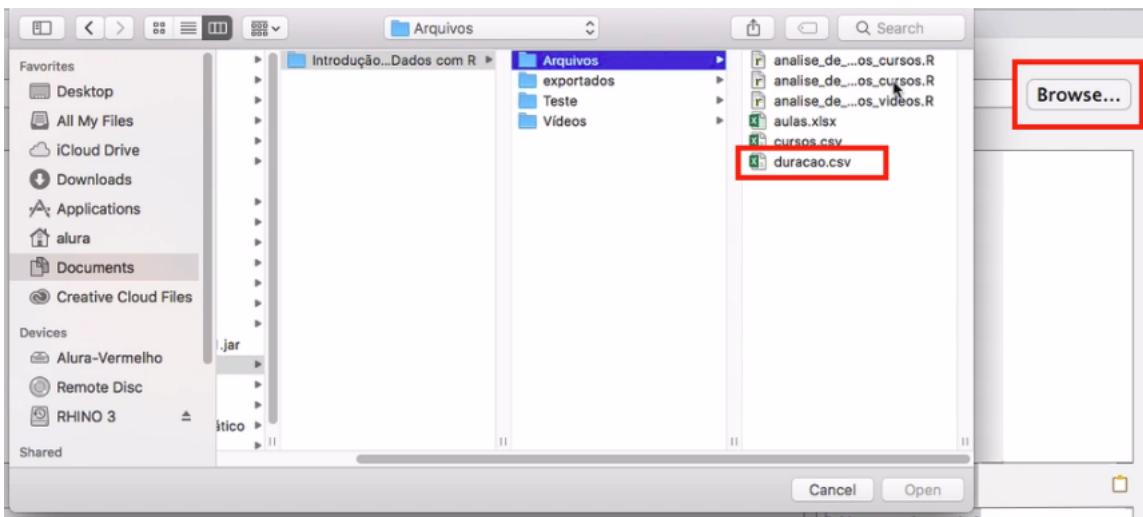
Ainda poderemos limpar o Console, por meio do comando `rm`, abreviação de *remove*. Entre parênteses, especificaremos `list=ls()`:

```
rm(list=ls())
```

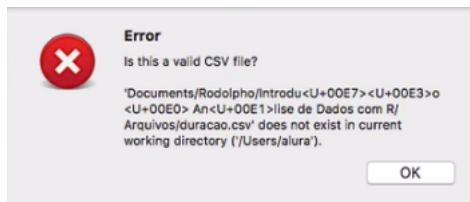
Assim, limparemos a **lista** do Console.

The screenshot shows the RStudio interface. The top bar includes tabs for 'Console', 'Environment', 'History', 'Connections', and 'Project: (None)'. The 'Console' tab is active, showing the command `rm(list=ls())` which has been run. The 'Environment' tab shows a message 'Environment is empty'. The 'Plots' tab is also visible.

A área de trabalho está pronta para uma nova análise! Importaremos um novo banco de dados `duracao.csv`, com informações sobre o tempo que os alunos demoram para concluir os cursos. Clicaremos em "*Import Dataset > From Text (readr)... > Browse*" e selecionaremos "*From Text (readr)...*", pois agora trabalharemos o arquivo `duracao.csv`, considerado um arquivo de texto, e não `.xlsx`, do Excel.



Após a seleção, abre-se uma mensagem de erro para confirmar se o arquivo é .csv :



Podemos clicar em "OK" e, em seguida, em "Import", para importar o banco de dados. O banco de dados será exibido na parte superior à esquerda do programa. Nele, veremos as variáveis `user_id` com códigos dos alunos, `course_id` com códigos de cursos e `timeToFinish`, com que ainda não trabalhamos.

```

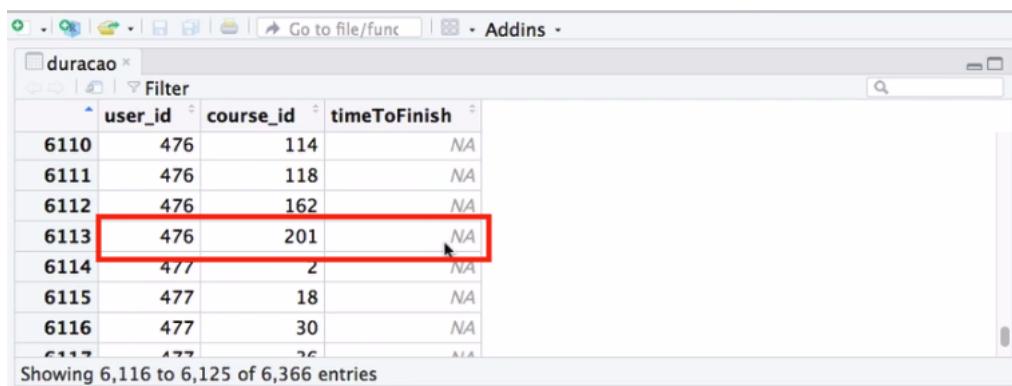
duracao <- read_csv("Documents/Rodolpho/Introdução...Dados com R/Arquivos/duracao.csv")
Parsed with column specification:
cols(
  user_id = col_integer(),
  course_id = col_integer(),
  timeToFinish = col_integer()
)
View(duracao)

```

	user_id	course_id	timeToFinish
1	477	19	0
2	110	251	0
3	474	149	0
4	474	140	0
5	474	167	0
6	30	213	0
7	292	120	0

Essa nova variável, nomeada pelo desenvolvedor do banco de dados, informa quantos dias o aluno demorou para concluir um curso. Se leremos a primeira linha da tabela, por exemplo, obteremos a informação de que o aluno 477 levou 0 dias para concluir o curso 19. Ou seja, ele começou e terminou o curso no mesmo dia, algo que acontece em muitos cursos.

Nas linhas abaixo, veremos que os dias aumentam; o aluno 45 fez o curso 181 em 5 dias. Seguindo para as próximas linhas da tabela, encontraremos que o aluno 476, que fez vários cursos na empresa, não concluiu (NA) o curso 201.



user_id	course_id	timeToFinish
6110	476	114
6111	476	118
6112	476	162
6113	476	201
6114	477	2
6115	477	18
6116	477	30
6117	477	26

Showing 6,116 to 6,125 of 6,366 entries

NA é sigla de "Not Available", "Não Disponível" em inglês. Isso indica que no banco de dados, não há número de dias naquela variável, e que o aluno começou e não terminou, ou ainda que não tinha terminado o curso quando a empresa nos passou o banco de dados.

De qualquer forma, somente a empresa tem acesso a essa informação. Discutiremos isso futuramente. Agora, abriremos uma nova janela R Script para colocar os novos comandos, clicando no ícone verde com sinal de mais (+) branco, localizado no canto superior esquerdo do programa.

Digitar `timeToFinish` toda vez que quisermos trabalhar com a variável poderia dificultar a análise. O nome dela possui uma escrita que a linguagem entende, mas é difícil para digitar, pois não há espaçamento e requer a utilização da tecla "Shift" a todo momento, começando com letra minúscula e no meio passa a ter letras maiúsculas no começo de cada palavra - isso é um padrão no desenvolvimento de banco de dados, sobretudo em SQL, por conta da legibilidade.

Usaremos um nome mais fácil para trabalharmos no RStudio; no novo documento de R Script que abrirmos, digitaremos `duracao` e atribuiremos (`<-`) a ela a função `rename()`, de renomear, que está no pacote `plyr`. Assim, estaremos solicitando que a variável seja renomeada. Especificaremos o banco de dados em que a variável se localiza, no caso `duracao`, e após a vírgula (`,`) digitaremos `replace` – comando de substituição, sinal de igual (`=`) e, em seguida, passaremos as variáveis que queremos renomear.

Ao passarmos mais de uma variável, utilizamos um comando chamado `c`, ou "Criação de um Vetor", em que especificaremos, entre parênteses, as variáveis que desejamos renomear, com seus respectivos novos nomes. Reparem que colocaremos os nomes antigos e novos **entre aspas** (`"`) e entre elas inserimos sinal de igual (`=`).

Atenção: dependendo da máquina, a navegação com aspas muda, e poderão ser confundidas com o acento trema (`"`). Por isso, devemos ter cuidado na hora de digitar os nomes e sempre fechar as aspas após abri-las.

Para renomearmos outras variáveis, inseriremos vírgula e abrimos aspas. Alteraremos `"course_id"` para `curso` e a variável que mais nos interessava, `timeToFinish`, para `dias`, que será uma informação interessante para nós, com nome curto e fácil para trabalharmos. O comando ficará da seguinte forma:

```
duracao <- rename(duracao, replace = c("user_id"="aluno", "course_id"="curso", "timeToFinish"="di
```

Executaremos para certificar que não cometemos erros de sintaxe.

```
> duracao <- rename(duracao, replace = c("user_id"="aluno", "course_id"="curso", "timeToFinish"="di
```

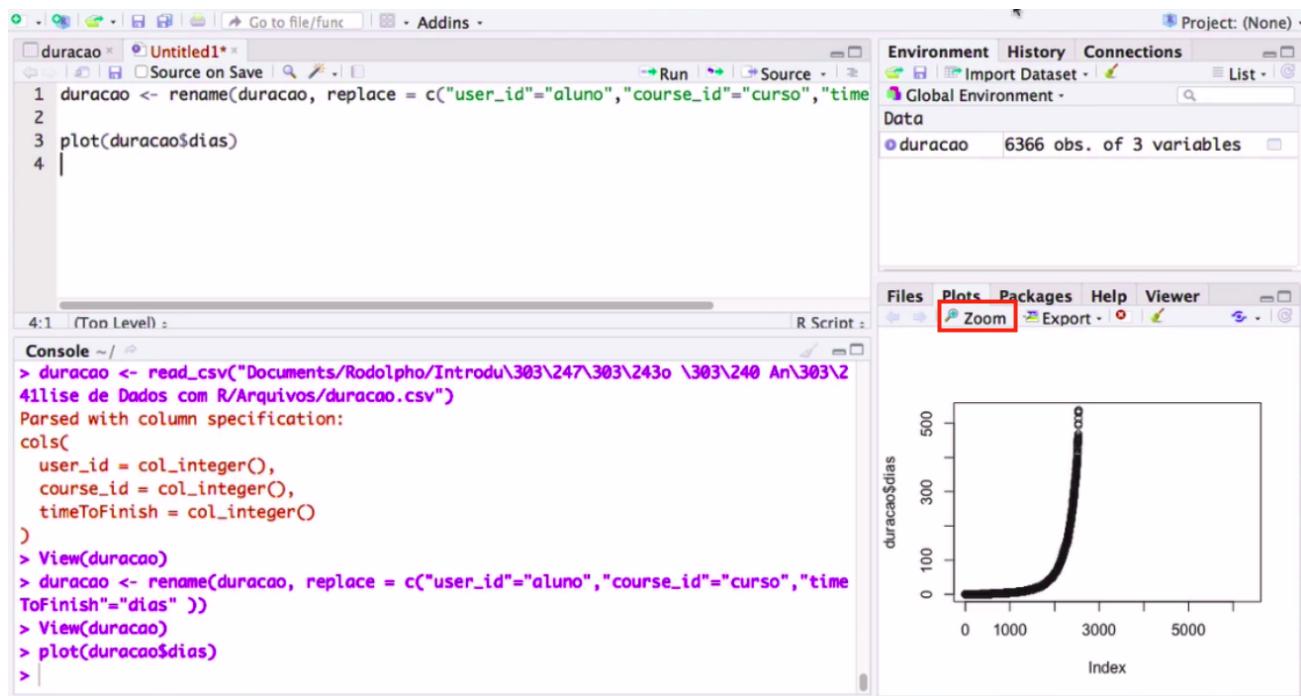
Funcionou! No Console, tivemos o retorno correto. Caso apareça alguma mensagem de erro, basta voltar e corrigir. Agora, o banco de dados está com as variáveis renomeadas. Se clicarmos em "duracao" na janela superior direita, na janela ao lado esquerdo, o banco de dados será exibido com os novos nomes, mais intuitivos, das colunas:

- user_id aparece como aluno ;
- course_id aparece como curso ;
- timeToFinish aparece como dias .

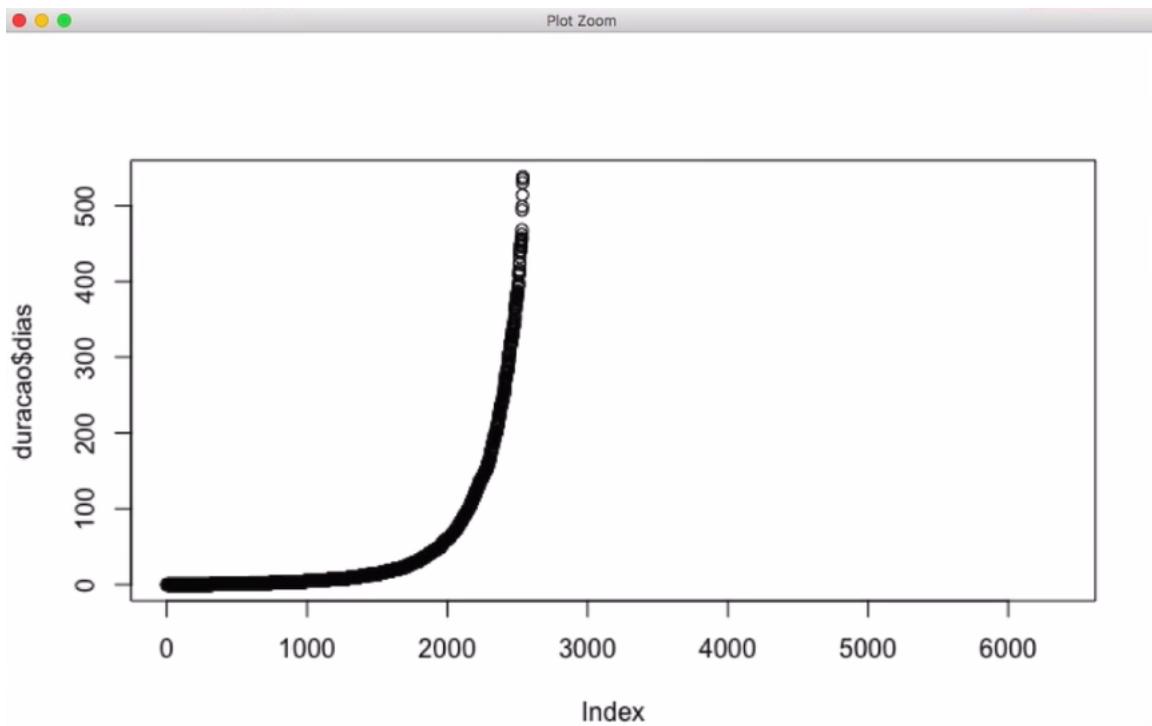
Analisaremos os cursos e quantos dias os alunos demoram para concluir-los. Vamos tentar criar um gráfico com esses dados, pois anteriormente vimos que trabalhar com eles de forma bruta é complicado. No RStudio, a função `plot()` é utilizada para criar gráficos. Digitaremos no novo arquivo de R Script, com o banco de dados `duracao`, seguido de cifrão (`$`) e `dias`, entre parênteses.

```
plot(duracao$dias)
```

Desta forma, pedimos para que o programa pinte quantos dias os cursos levam para ser concluídos. A função `plot()` escolhe um gráfico – a princípio, não sabemos qual – a partir das informações que fornecemos, e do tipo de dados. Ao clicarmos em "Run" na janela inferior direita, será aberta uma janela de gráfico, com o padrão de exibição do RStudio para gráficos.



Podemos aplicar Zoom e expandir a janela para analisarmos melhor o gráfico:

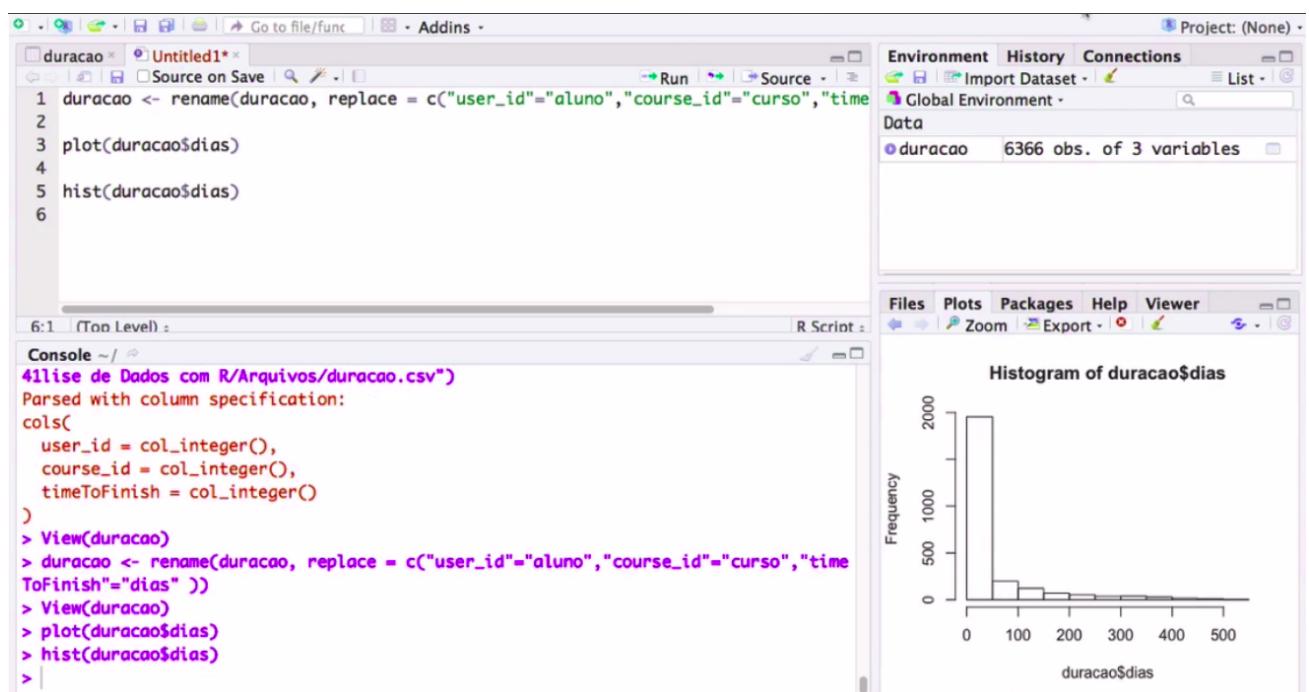


Nele, temos um índice ("Index", na horizontal), um posicionamento das observações, e na vertical, a duração em dias ("duracao\$dias"). Não conseguimos extrair nenhuma informação a partir desse gráfico, e constatamos que não foi uma boa ideia deixar para o programa escolher o gráfico para a análise.

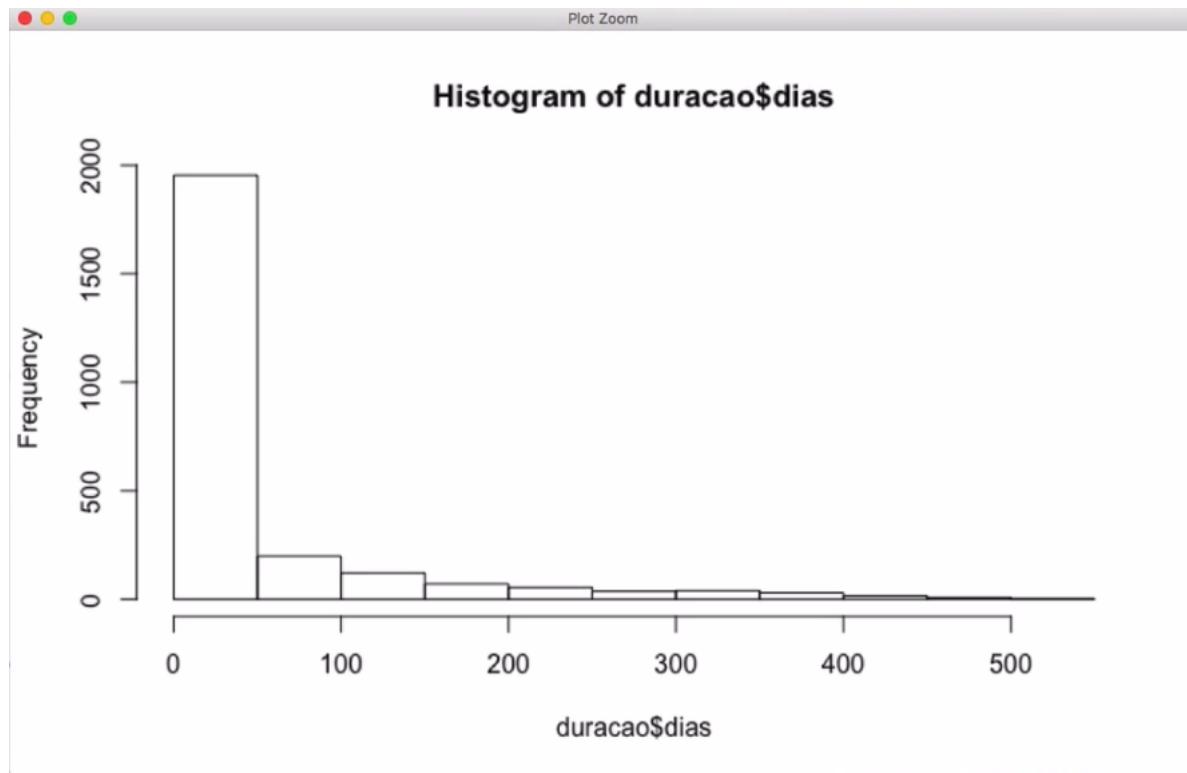
Existe um gráfico melhor, do qual poderemos extrair mais informação sobre a distribuição dos dias, que é o nosso objetivo. Poderemos quantificar os cursos e os dias necessários para conclusão. O gráfico adequado para essa análise é o histograma, cujo comando no RStudio é `hist` :

```
hist(duracao$dias)
```

Ao ser executado, teremos um gráfico informativo:



E com o zoom, teremos uma melhor visualização:



Conseguimos visualizar a distribuição da duração dos cursos. No eixo horizontal observamos a variável `dias`, que vai de 0 a mais de 500. No eixo vertical, temos a `frequência`, e por meio dela podemos enumerar quantas vezes os cursos que duraram 0 dias apareceram na amostra, por exemplo. São quase 2000 deles realizados em um único dia. Depois, há aproximadamente 250 cursos que duraram entre 50 e 100 dias. E assim o número vai diminuindo. No entanto, ainda no final do eixo horizontal encontraremos alguns cursos que levaram mais de 500 dias para a conclusão.