# Revolutionize the way you teach and blog tutorial package

Filip Schouwenaars
DataCamp

github.com/filipsch/erum-presentation

github.com/datacamp/tutorial

# R Markdown

````
---
title: "My example"
output: html_document
---

When you start R, `mtcars` is available.

Let's explore with `head()` and `tail()`.

```{r}
head(mtcars)
tail(mtcars)
```

That's enough for now!
````

# My example

When you start R, `mtcars` is available.

Let's explore with `head()` and `tail()`.

```
head(mtcars)
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```
tail(mtcars)
```

```
##                mpg cyl  disp  hp drat    wt qsec vs am gear carb
## Porsche 914-2  26.0   4 120.3  91 4.43 2.140 16.7  0  1    5    2
## Lotus Europa   30.4   4  95.1 113 3.77 1.513 16.9  1  1    5    2
## Ford Pantera L 15.8   8 351.0 264 4.22 3.170 14.5  0  1    5    4
## Ferrari Dino   19.7   6 145.0 175 3.62 2.770 15.5  0  1    5    6
## Maserati Bora  15.0   8 301.0 335 3.54 3.570 14.6  0  1    5    8
## Volvo 142E     21.4   4 121.0 109 4.11 2.780 18.6  1  1    4    2
```

That's enough for now!

# R Markdown + tutorial

````
---
title: "My example"
output: html_document
---

```{r include=FALSE}
tutorial::go_interactive()
```

When you start R, `mtcars` is available.

Let's explore with `head()` and `tail()`.

```{r}
head(mtcars)
tail(mtcars)
```

That's enough for now!
````

## My example

When you start R, `mtcars` is available.

Let's explore with `head()` and `tail()`.

script.R | R Console

```
1  head(mtcars)
2  tail(mtcars)
```

Run

Powered by DataCamp

That's enough for now!

# tutorial

- Install from CRAN

```
> install.packages("tutorial")
```

- Add chunk to top of R Markdown file

````
```{r include=FALSE}
tutorial::go_interactive()
```
````

github.com/datacamp/tutorial

# How it works

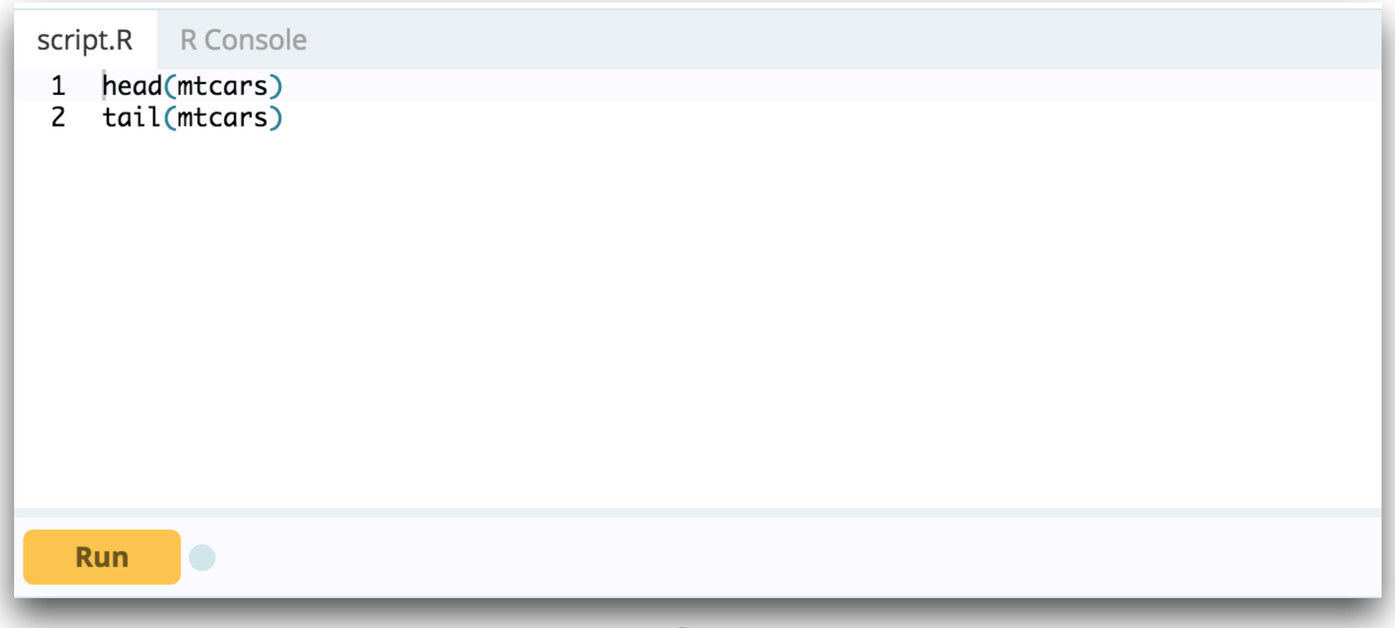
```
```{r}
head(mtcars)
tails(mtcars)
```
```

knitr::knit_hooks

tutorial package

```
<script src="https://cdn.datacamp.com/
  datacamp-light-latest.min.js"></script>
<div data-datacamp-exercise data-lang="r">
    <code data-type="sample-code">
        head(mtcars)
        tail(mtcars)
    </code>
</div>
```

DataCamp Light




github.com/datacamp/datacamp-light

# Examples

- Book: advanced R
- Vignette: jsonlite

DataCamp

Course Outline

# Create a Histogram

100xp

Another function that Filip introduced in the video, is `hist()`. Basically, the `hist()` function visualizes the distribution of your data by placing all values in bins and displaying the bin frequencies of each bin. Have a look at the usage of the `hist()` function:

```
hist(x, breaks = "Sturges")
```

`x` is the vector of values for which you want to create a histogram. The `breaks` argument specifies how many bins, or histogram cells, your plot should contain. By default, the "Sturges" algorithm is used to automatically determine the number of bins.

This slightly positive correlation has got you thinking about the `rating` variable of the `movies` data frame. Maybe a quick peek at its distribution might help your understanding.

## Instructions

- Create a histogram of the `rating` variable of `movies`.
- Do the same thing, but this time set the number of bins to 20 with the `breaks` argument.

Take Hint (-30xp)

**Incorrect submission**

Did you correctly specify the argument `breaks` in your call of `hist()`?

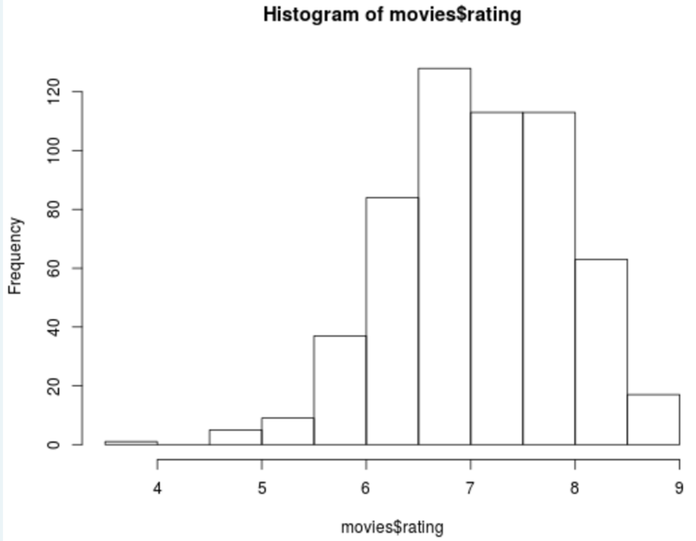
*How helpful is this feedback?*

script.R

```
1  # movies is already pre-loaded
2
3  # Create a histogram for rating
4  hist(movies$rating)
5
6  # Create a histogram for rating, with 20 bins
7  hist(movies$rating, breaks = 10)
```

Submit Answer

Plots




Previous Plot

2/2

R Console

```
> # movies is already pre-loaded

>

> # Create a histogram for rating

> hist(movies$rating)

>

> # Create a histogram for rating, with 20 bins

> hist(movies$rating, breaks = 10)

>
```

my_exercise.Rmd

````
```{r include=FALSE}
tutorial::go_interactive()
```


You can use `<-` for variable assignment.
Try it yourself: create a variable `a`, that
is equal to 5.

```{r ex="create_a", type="sample-code"}
# Create a

```


```{r ex="create_a", type="solution"}
# Create a
a <- 5
```


```{r ex="create_a", type="sct"}
test_object("a")
```
````

github.com/datacamp/testwhat

# Conclusion

```
```{r include=FALSE}
tutorial::go_interactive()
```
```

- Make static code chunks interactive
- Every HTML file generated with R Markdown
- Try it yourself!
- Future
    - Support multiple package versions
    - More features from DataCamp

# Questions?

github.com/filipsch/erum-presentation

github.com/datacamp/tutorial

github.com/datacamp/testwhat

github.com/datacamp/datacamp-light