



Metodologías activas en Educación

Resultados significativos del análisis de encuestas

Juan Abasolo, Aintzane Etxebarria,
Aitor Iglesias, Asier Romero

EUDIA ikerketa-taldea



Universidad
del País Vasco
Euskal Herriko
Unibertsitatea

BILBOKO
HEZKUNTZA
FAKULTATEA
FACULTAD
DE EDUCACIÓN
DE BILBAO

2019/06/04

1 / 47

Aprendizaje basado en retos

3 fases

Presentación del reto +
propuestas de solución

Investigación, evaluación y
revisión

Publicación en plataforma
educativa



Ventajas

**Apple (2011)
Observatorio de Tecnología Educativa
del TM (2016)**

Aprovechar interes de estudiantes

Comprensión más profunda

Acercamiento a la realidad de la comunidad

Tendencia a desarrollo de comunicación a alto nivel

Piñeiro Otero

Enriquecimiento de la experiencia

Potenciación de la atención y reflexión

Incremento de la satisfacción





Marco metodológico del Aprendizaje Basado en Retos de Apple (2011)

]

Videocast

Pasos del proyecto

1^a fase

Presentación y comienzo del trabajo

2^a fase

Alumnos de disciplinas y grados diferentes. Investigación acerca de los productos a generar

3^a fase

Publicación del material en la plataforma educativa de Internet



Hezkuntza Berrikuntzako Proiekta

Hasiera Unitate Didaktikoak Ikerketari buruz Argitalpenak Kontaktua Eman zure iritzia!

Log In

Proposamen didaktikoak

Hezkuntzan, Unitate Didaktikoa ikasgai edo eduki jakin eta zehatz baten inguruan hezkuntza-ikasketa prozesua antolatzeko modu bat da, edukia ikaslearen mailara, eskura dauden baliabideetara eta lortu beharreko helburuetara egokitzen duena. Curriculumak barne hartzen dituen ezagutzak eta gaitasunak ikasgai ezberdinetan bereizten dituzte unitate didaktikoek eta horrela, hezkuntzaren antolaketan denbora-planifikazioa eta beharrezko ebaluazioa bideratu. (Wikipedia)

OHAR METODOLOGIKOA

Komeni da honako pausu hauek jarraitzea:

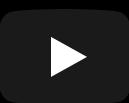
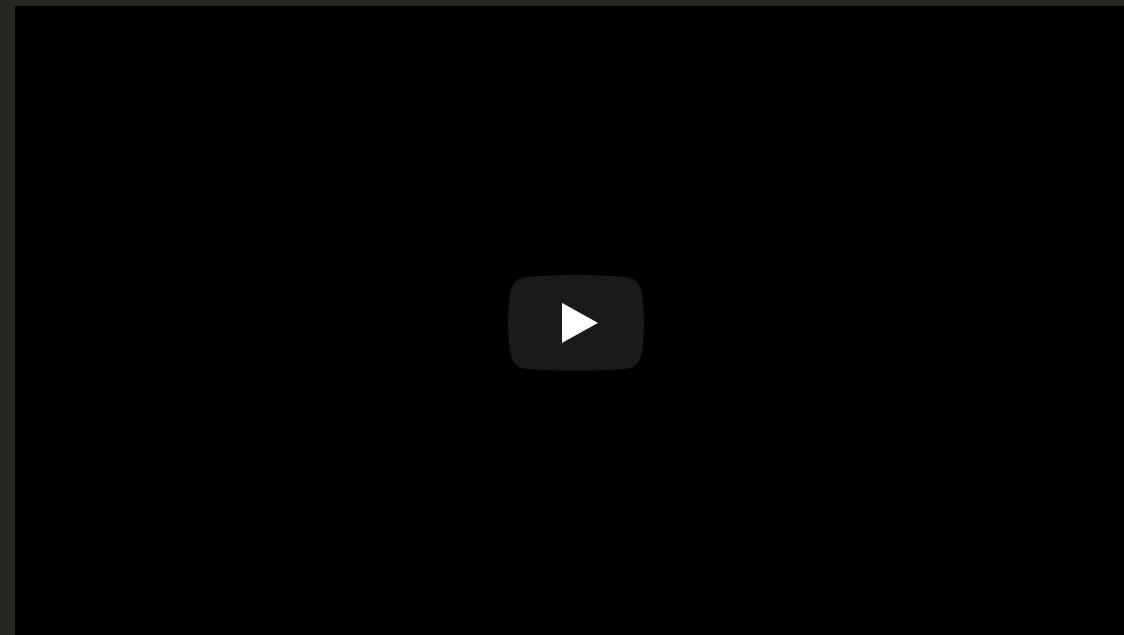
- 1- Bideoa ikusi
- 2- Burutu sekuentzia didaktikoa
- 3- Eman zure iritzia

Informazio gehiago

1. Proposamen didaktikoa: Zatikiak

Proposamen honen bitartez, Lehen Hezkuntzako 5. mailako ikasleek egunerokotasuneko egoerez baliatuz, oinarrizko zatiki, zenbaki hamartar eta portzentaien arteko baliokidetasunak egiten eta arazoak ebazten ikasiko dute.





• • •

esate baterako.



Habitos de
estudio

CHAEA

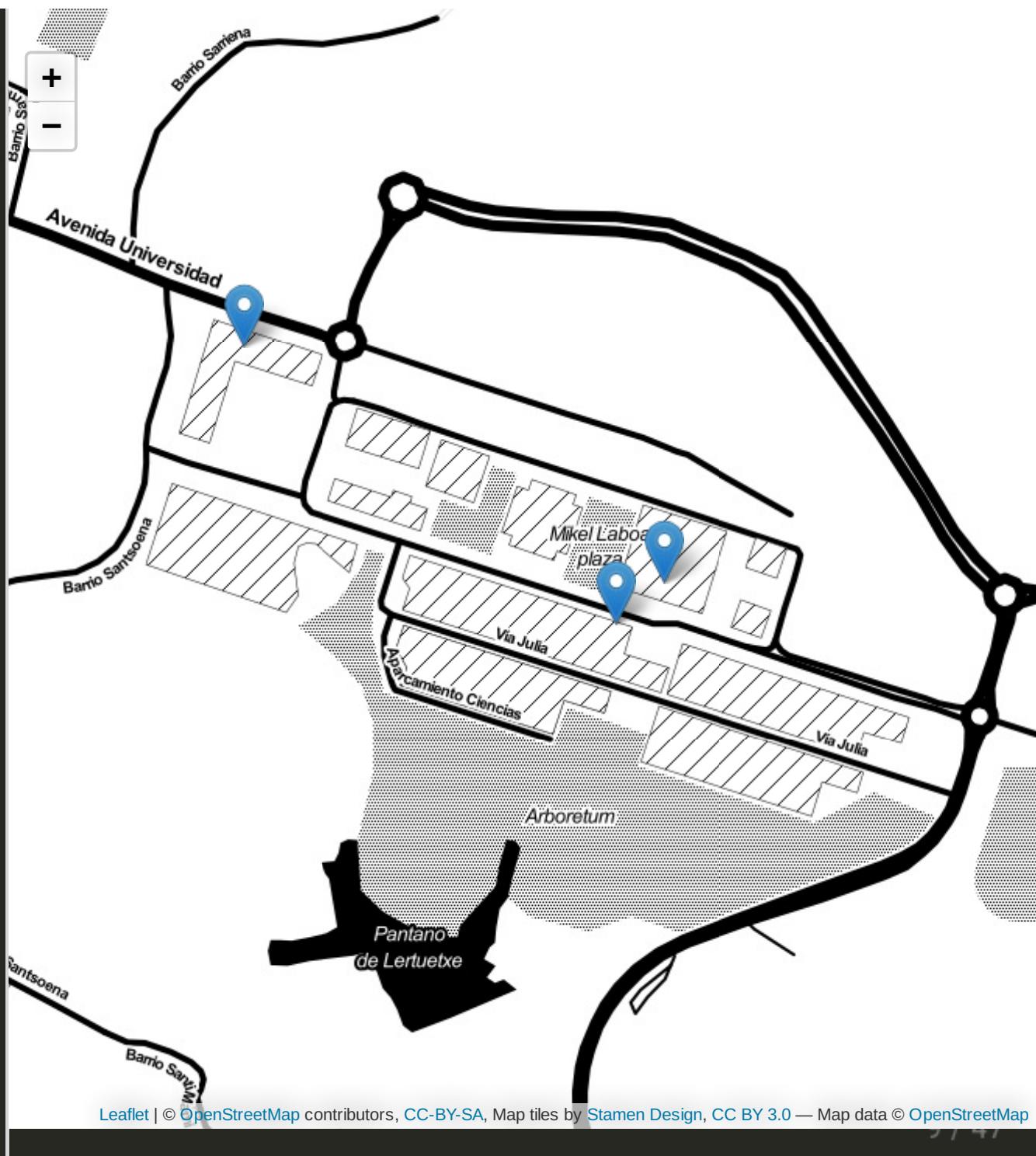


Motivación
EMSI

Trabajo
cooperativo

stand by

Recogida de datos



Muestra

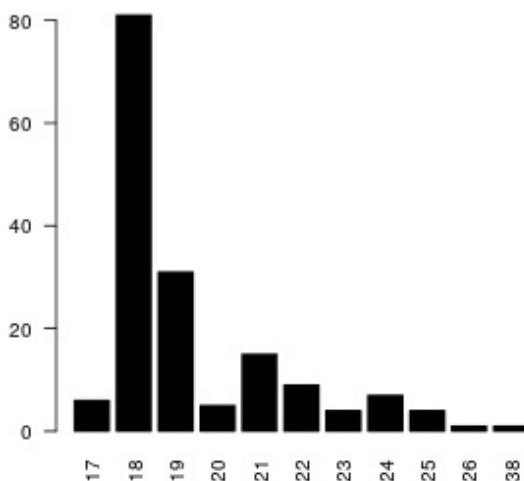
n = 164

3 facultades

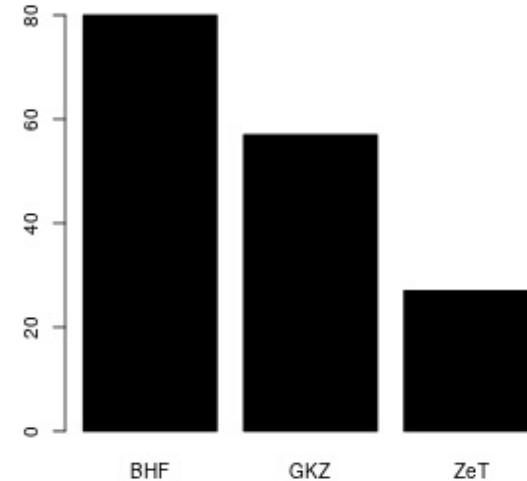
8 carreras



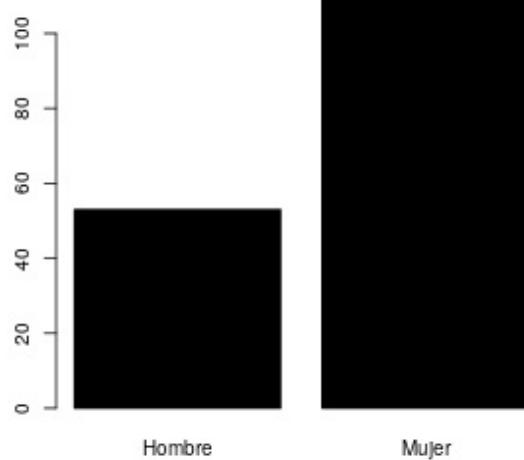
Banaketa adinaren arabera



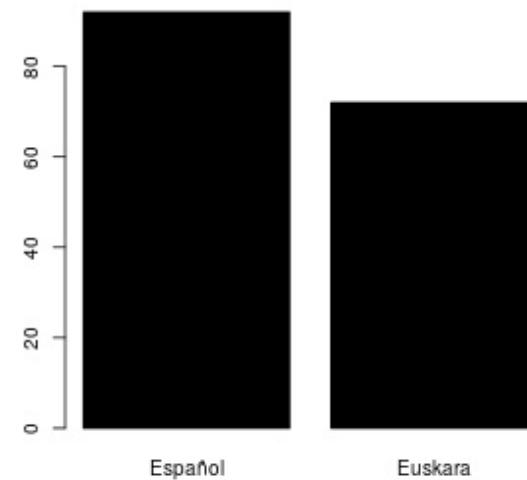
Banaketa fakultatearen arabera



Banaketa generoaren arabera



Banaketa ama hizkuntzaren arabera



CHAEA

Cuestionario de Estilos de Aprendizaje

Alonso, Gallego & Honey (1996)



Kolb (1984)

Construyó la base teórica.

**Honey & Mumford
(1986)**

Construyeron *Learning Styles Questionnaire (LSQ)*

**Gallego & Honey
(1996)**

Portaron al castellano

Perspectiva cuatridimensional

Activo

Pragmático

Reflexivo

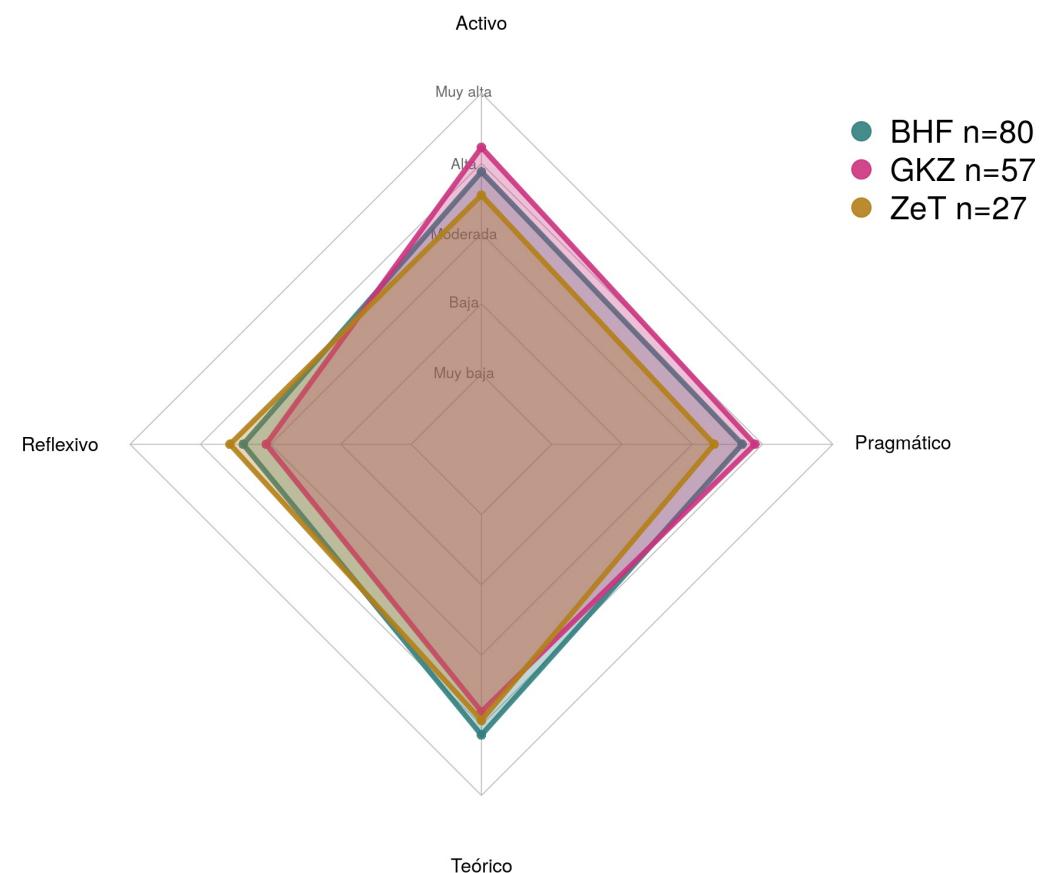
Teórico

CHAEA

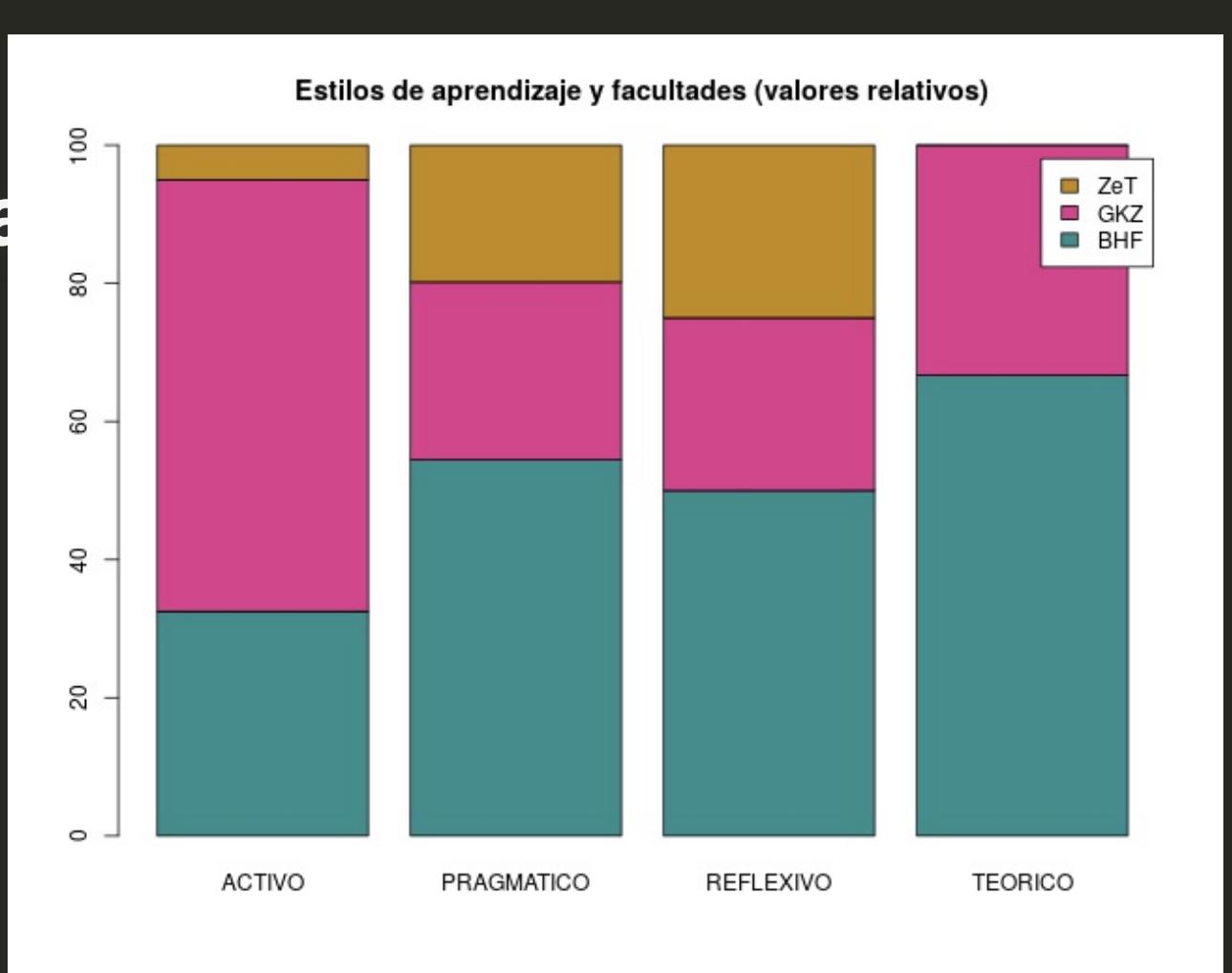
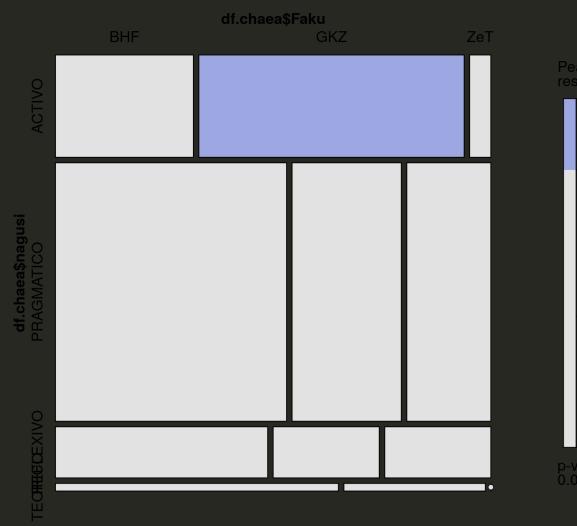
Perspectiva cuatridimensional



Estilos de aprendizaje y facultades



Perspectiva monodimensional



```
##  
## Pearson's Chi-squared test  
##  
## data: table(df.chaea$Faku, df.chaea$nagusi)  
## X-squared = 20.068, df = 6, p-value = 0.002693
```

EMSI

XXXXX

XXXX



XXX

XXXX.

XXX

XXXX

]]

XXX

XXX

Perspectiva xxxx

XXX

EMSI

Hello Ninja

As a presentation ninja, you certainly should not be satisfied by the "Hello World" example. You need to understand more about two things:

1. The **remark.js** library;
2. The **xaringan** package;

Basically **xaringan** injected the chakra of R Markdown (minus Pandoc) into **remark.js**. The slides are rendered by remark.js in the web browser, and the Markdown source needed by remark.js is generated from R Markdown (**knitr**).

remark.js

You can see an introduction of remark.js from [its homepage](#). You should read the [remark.js Wiki](#) at least once to know how to

- create a new slide (Markdown syntax^{*} and slide properties);
- format a slide (e.g. text alignment);
- configure the slideshow;
- and use the presentation (keyboard shortcuts).

It is important to be familiar with remark.js before you can understand the options in [xaringan](#).

[*] It is different with Pandoc's Markdown! It is limited but should be enough for presentation purposes. Come on... You do not need a slide for the Table of Contents! Well, the Markdown support in remark.js [may be improved](#) in the future.



I was so happy to have
discovered remark.js!

Using xaringan

xaringan

Provides an R Markdown output format `xaringan::moon_reader` as a wrapper for `remark.js`, and you can use it in the YAML metadata, e.g.

```
---
```

```
title: "A Cool Presentation"
output:
  xaringan::moon_reader:
    yolo: true
    nature:
      autoplay: 30000
---
```

See the help page `?xaringan::moon_reader` for all possible options that you can use.

remark.js vs xaringan

Some differences between using remark.js (left) and using **xaringan** (right):

- | | |
|---|---|
| <ol style="list-style-type: none">1. Start with a boilerplate HTML file;2. Plain Markdown;3. Write JavaScript to autoplay slides;4. Manually configure MathJax;5. Highlight code with *;6. Edit Markdown source and refresh browser to see updated slides; | <ol style="list-style-type: none">1. Start with an R Markdown document;2. R Markdown (can embed R/other code chunks);3. Provide an option autoplay;4. MathJax just works;*5. Highlight code with {{}};6. The RStudio addin "Infinite Moon Reader" automatically refreshes slides on changes; |
|---|---|

[*] Not really. See next page.

Math Expressions

You can write LaTeX math expressions inside a pair of dollar signs, e.g. `\alpha+\beta` renders $\alpha + \beta$. You can use the display style with double dollar signs:

```
$$\bar{X}=\frac{1}{n}\sum_{i=1}^n X_i$$
```

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$$

Limitations:

1. The source code of a LaTeX math expression must be in one line, unless it is inside a pair of double dollar signs, in which case the starting `$$` must appear in the very beginning of a line, followed immediately by a non-space character, and the ending `$$` must be at the end of a line, led by a non-space character;
2. There should not be spaces after the opening `$` or before the closing `$`.
3. Math does not work on the title slide (see [#61](#) for a workaround).

R Code

```
# a boring regression
fit = lm(dist ~ 1 + speed, data = cars)
coef(summary(fit))

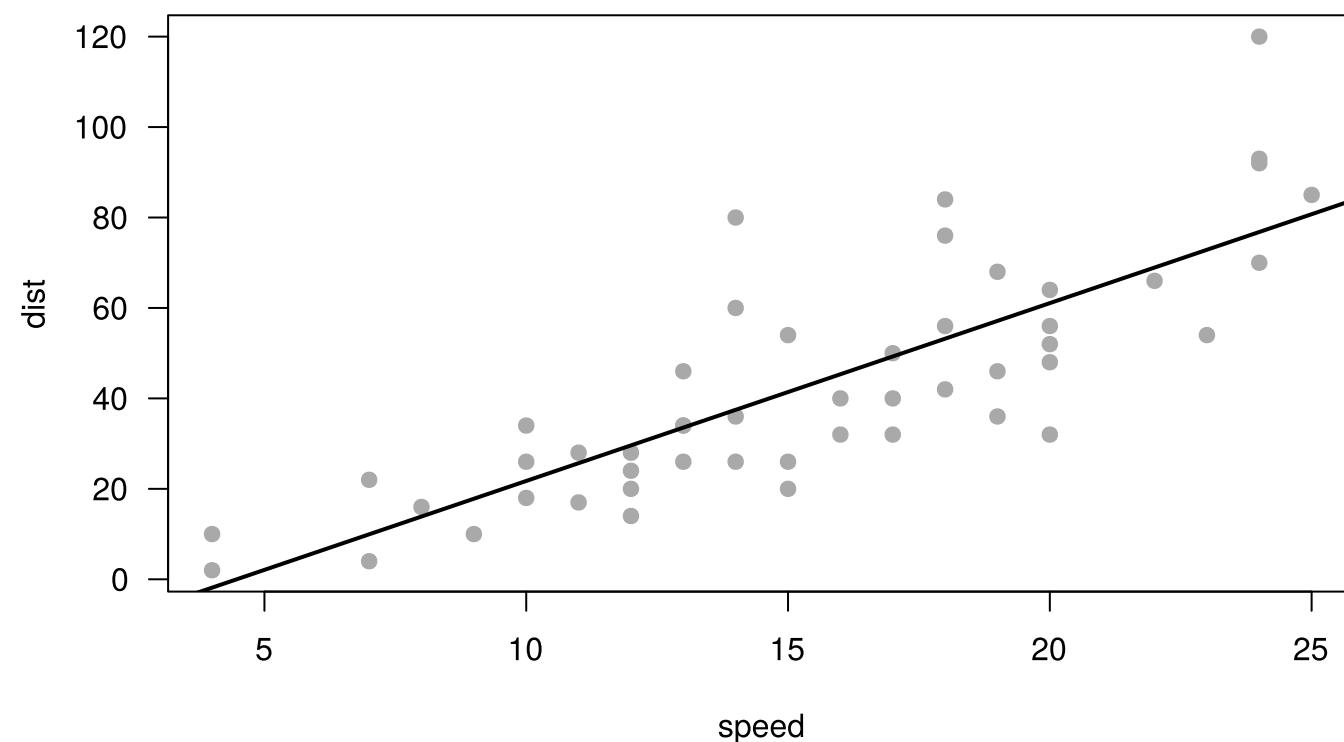
#           Estimate Std. Error   t value   Pr(>|t|)
# (Intercept) -17.579095  6.7584402 -2.601058 1.231882e-02
# speed        3.932409  0.4155128  9.463990 1.489836e-12

dojutsu = c('地爆天星', '天照', '加具土命', '神威', '須佐能乎', '無限月読')
grep('天', dojutsu, value = TRUE)

# [1] "地爆天星" "天照"
```

R Plots

```
par(mar = c(4, 4, 1, .1))
plot(cars, pch = 19, col = 'darkgray', las = 1)
abline(fit, lwd = 2)
```



Tables

If you want to generate a table, make sure it is in the HTML format (instead of Markdown or other formats), e.g.,

```
knitr::kable(head(iris), format = 'html')
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa

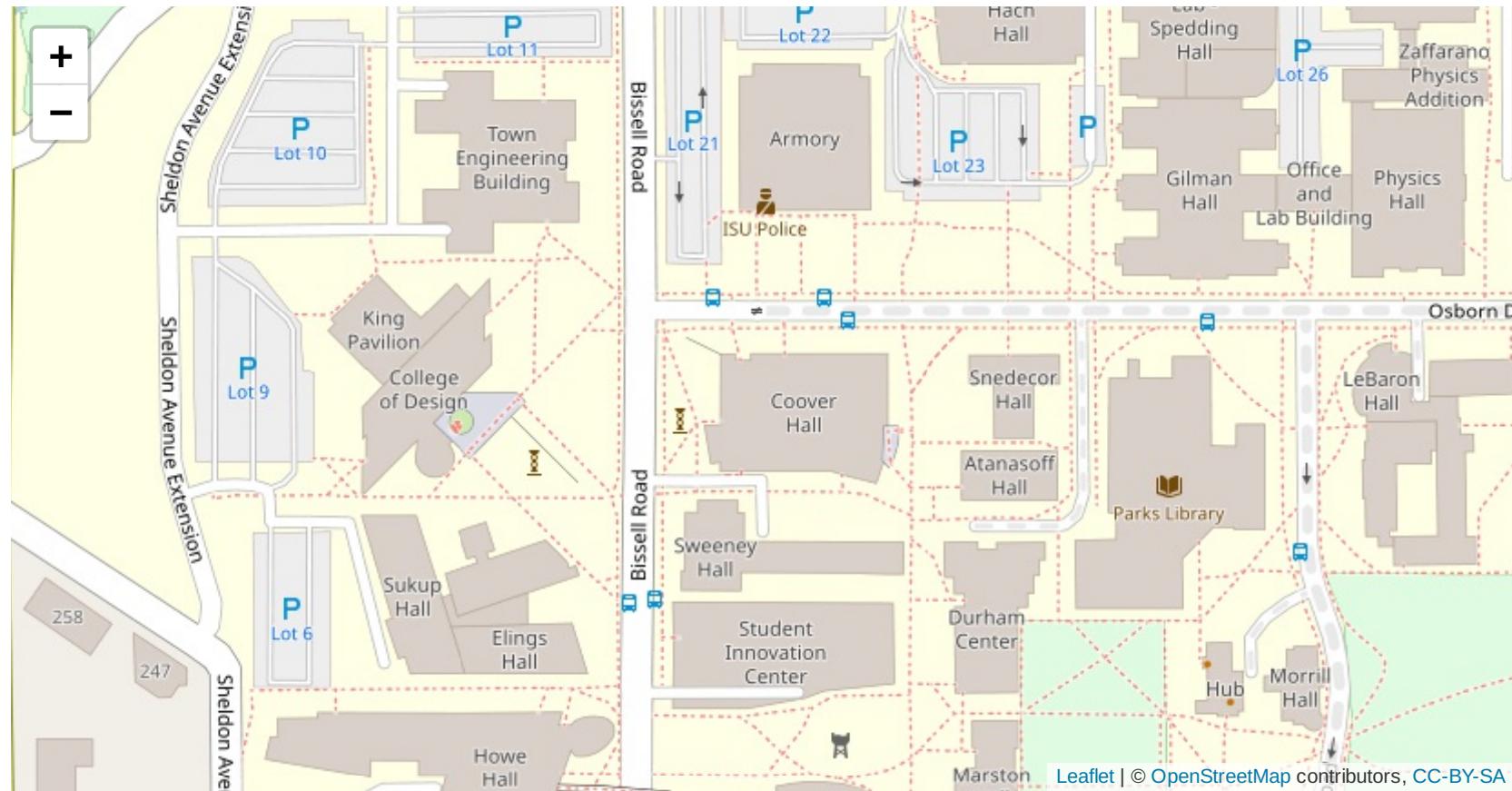
HTML Widgets

I have not thoroughly tested HTML widgets against **xaringan**. Some may work well, and some may not. It is a little tricky.

Similarly, the Shiny mode (`runtime: shiny`) does not work. I might get these issues fixed in the future, but these are not of high priority to me. I never turn my presentation into a Shiny app. When I need to demonstrate more complicated examples, I just launch them separately. It is convenient to share slides with other people when they are plain HTML/JS applications.

See the next page for two HTML widgets.

```
library(leaflet)
leaflet() %>% addTiles() %>% setView(-93.65, 42.0285, zoom = 17)
```



```
DT::datatable(  
  head(iris, 10),  
  fillContainer = FALSE, options = list(pageLength = 8)  
)
```

Show 8 entries

Search:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5	3.4	1.5	0.2	setosa

Showing 1 to 8 of 10 entries

Previous

1

2

Next

Some Tips

- When you use the "Infinite Moon Reader" addin in RStudio, your R session will be blocked by default. You can click the red button on the right of the console to stop serving the slides, or use the *daemonized* mode so that it does not block your R session. To do the latter, you can set the option

```
options(servr.daemon = TRUE)
```

in your current R session, or in `~/.Rprofile` so that it is applied to all future R sessions. I do the latter by myself.

To know more about the web server, see the **servr** package.

- Do not forget to try the `yolo` option of `xaringan::moon_reader`.

```
output:  
  xaringan::moon_reader:  
    yolo: true
```

Some Tips

- Slides can be automatically played if you set the autoplay option under nature, e.g. go to the next slide every 30 seconds in a lightning talk:

```
output:  
  xaringan::moon_reader:  
    nature:  
      autoplay: 30000
```

- A countdown timer can be added to every page of the slides using the countdown option under nature, e.g. if you want to spend one minute on every page when you give the talk, you can set:

```
output:  
  xaringan::moon_reader:  
    nature:  
      countdown: 60000
```

Then you will see a timer counting down from 01 : 00, to 00 : 59, 00 : 58, ... When the time is out, the timer will continue but the time turns red.

Some Tips

- The title slide is created automatically by **xaringan**, but it is just another remark.js slide added before your other slides.

The title slide is set to class: center, middle, inverse, title-slide by default. You can change the classes applied to the title slide with the titleSlideClass option of nature (title-slide is always applied).

```
output:  
  xaringan::moon_reader:  
    nature:  
      titleSlideClass: [top, left, inverse]
```

- If you'd like to create your own title slide, disable **xaringan**'s title slide with the seal = FALSE option of moon_reader.

```
output:  
  xaringan::moon_reader:  
    seal: false
```

Some Tips

- There are several ways to build incremental slides. See [this presentation](#) for examples.
- The option `highlightLines: true` of `nature` will highlight code lines that start with `*`, or are wrapped in `{{ }}`, or have trailing comments `#<<`;

```
output:  
  xaringan::moon_reader:  
    nature:  
      highlightLines: true
```

See examples on the next page.

Some Tips

An example using a leading *:

```
```r
if (TRUE) {
* message("Very important!")
}
```

```

Output:

```
if (TRUE) {
  message("Very important!")
}
```

This is invalid R code, so it is a plain fenced code block that is not executed.

An example using {{}}:

```
```{r tidy=FALSE}
if (TRUE) {
{{ message("Very important!") }}
}
```

```

Output:

```
if (TRUE) {
  message("Very important!")
}
```

```
## Very important!
```

It is valid R code so you can run it. Note that {{}} can wrap an R expression of multiple lines.

Some Tips

An example of using the trailing comment #<< to highlight lines:

```
```{r tidy=FALSE}
library(ggplot2)
ggplot(mtcars) +
 aes(mpg, disp) +
 geom_point() + #<<
 geom_smooth() #<<
```
```

Output:

```
library(ggplot2)
ggplot(mtcars) +
  aes(mpg, disp) +
  geom_point() +    #<<
  geom_smooth()
```

Some Tips

When you enable line-highlighting, you can also use the chunk option `highlight.output` to highlight specific lines of the text output from a code chunk. For example, `highlight.output = TRUE` means highlighting all lines, and `highlight.output = c(1, 3)` means highlighting the first and third line.

```
```{r, highlight.output=c(1, 3)}
head(iris)
```
```

| | ## Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|------|-----------------|-------------|--------------|-------------|---------|
| ## 1 | 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| ## 2 | 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| ## 3 | 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| ## 4 | 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| ## 5 | 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| ## 6 | 5.4 | 3.9 | 1.7 | 0.4 | setosa |

Question: what does `highlight.output = c(TRUE, FALSE)` mean? (Hint: think about R's recycling of vectors)

Some Tips

- To make slides work offline, you need to download a copy of remark.js in advance, because **xaringan** uses the online version by default (see the help page ? `xaringan::moon_reader`).
- You can use `xaringan::summon_remark()` to download the latest or a specified version of remark.js. By default, it is downloaded to `libs/remark-latest.min.js`.
- Then change the chakra option in YAML to point to this file, e.g.

```
output:  
  xaringan::moon_reader:  
    chakra: libs/remark-latest.min.js
```

- If you used Google fonts in slides (the default theme uses *Yanone Kaffeesatz*, *Droid Serif*, and *Source Code Pro*), they won't work offline unless you download or install them locally. The Heroku app [google-webfonts-helper](#) can help you download fonts and generate the necessary CSS.

Macros

- remark.js allows users to define custom macros (JS functions) that can be applied to Markdown text using the syntax ![:macroName arg1, arg2, ...] or ![:macroName arg1, arg2, ...](this). For example, before remark.js initializes the slides, you can define a macro named scale:

```
remark.macros.scale = function (percentage) {  
  var url = this;  
  return '';  
};
```

Then the Markdown text

```
![scale 50%](image.jpg)
```

will be translated to

```

```

Macros (continued)

- To insert macros in **xaringan** slides, you can use the option `beforeInit` under the option `nature`, e.g.,

```
output:  
  xaringan::moon_reader:  
    nature:  
      beforeInit: "macros.js"
```

You save your `remark.js` macros in the file `macros.js`.

- The `beforeInit` option can be used to insert arbitrary JS code before `remark.create()`. Inserting macros is just one of its possible applications.

CSS

Among all options in `xaringan::moon_reader`, the most challenging but perhaps also the most rewarding one is `css`, because it allows you to customize the appearance of your slides using any CSS rules or hacks you know.

You can see the default CSS file [here](#). You can completely replace it with your own CSS files, or define new rules to override the default. See the help page `?xaringan::moon_reader` for more information.

CSS

For example, suppose you want to change the font for code from the default "Source Code Pro" to "Ubuntu Mono". You can create a CSS file named, say, `ubuntu-mono.css`:

```
@import url(https://fonts.googleapis.com/css?family=Ubuntu+Mono:400,700,400italic);  
.remark-code, .remark-inline-code { font-family: 'Ubuntu Mono'; }
```

Then set the `css` option in the YAML metadata:

```
output:  
  xaringan::moon_reader:  
    css: ["default", "ubuntu-mono.css"]
```

Here I assume `ubuntu-mono.css` is under the same directory as your Rmd.

See [yihui/xaringan#83](#) for an example of using the `Fira Code` font, which supports ligatures in program code.

Themes

Don't want to learn CSS? Okay, you can use some user-contributed themes. A theme typically consists of two CSS files `foo.css` and `foo-ffonts.css`, where `foo` is the theme name. Below are some existing themes:

```
names(xaringan:::list_css())  
  
## [1] "chocolate-fonts"    "chocolate"          "default-fonts"  
## [4] "default"              "duke-blue"          "fc-fonts"  
## [7] "fc"                   "hygge-duke"        "hygge"  
## [10] "kunoichi"             "lucy-fonts"        "lucy"  
## [13] "metropolis-fonts"    "metropolis"       "middlebury-fonts"  
## [16] "middlebury"           "ninjutsu"          "rladies-fonts"  
## [19] "rladies"              "robot-fonts"       "robot"  
## [22] "rutgers-fonts"       "rutgers"           "shinobi"  
## [25] "tamu-fonts"           "tamu"              "uo-fonts"  
## [28] "uo"                  "uol-fonts"         "uol"
```

Themes

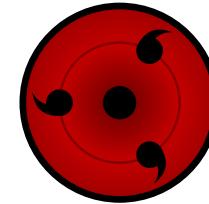
To use a theme, you can specify the `css` option as an array of CSS filenames (without the `.css` extensions), e.g.,

```
output:  
  xaringan::moon_reader:  
    css: [default, metropolis, metropolis-fonts]
```

If you want to contribute a theme to **xaringan**, please read [this blog post](#).



Sharingan



The R package name **xaringan** was derived¹ from **Sharingan**, a dōjutsu in the Japanese anime *Naruto* with two abilities:

- the "Eye of Insight"
- the "Eye of Hypnotism"

I think a presentation is basically a way to communicate insights to the audience, and a great presentation may even "hypnotize" the audience.^{2,3}

[1] In Chinese, the pronunciation of X is *Sh* /ʃ/ (as in *shrimp*). Now you should have a better idea of how to pronounce my last name *Xie*.

[2] By comparison, bad presentations only put the audience to sleep.

[3] Personally I find that setting background images for slides is a killer feature of remark.js. It is an effective way to bring visual impact into your presentations.

Naruto terminology

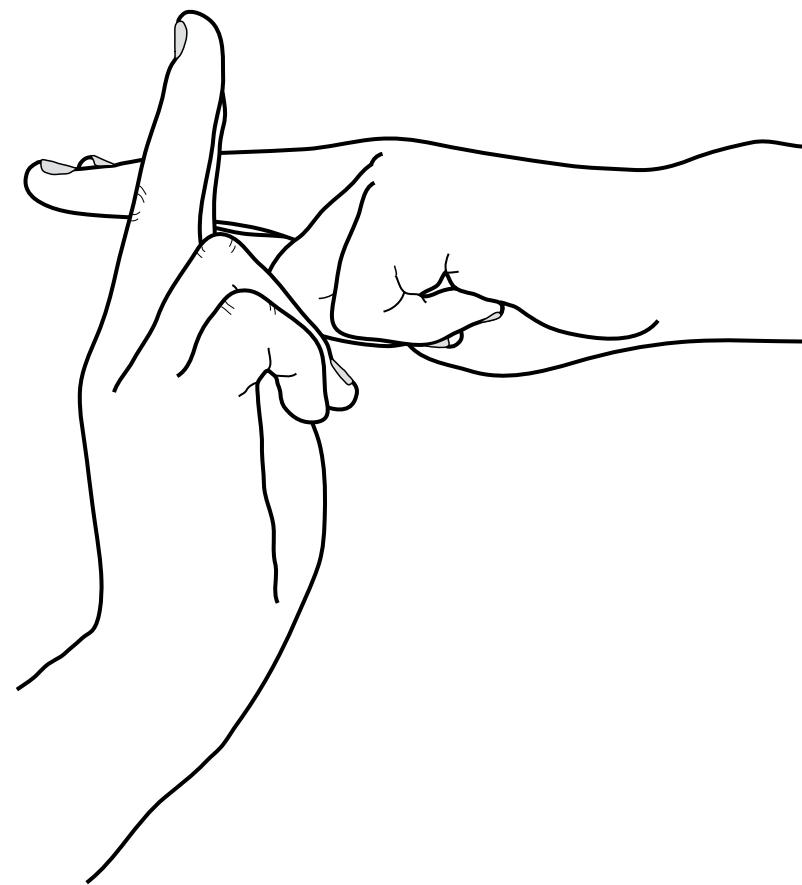
The **xaringan** package borrowed a few terms from Naruto, such as

- **Sharingan** (写輪眼; the package name)
- The **moon reader** (月読; an attractive R Markdown output format)
- **Chakra** (查克拉; the path to the remark.js library, which is the power to drive the presentation)
- **Nature transformation** (性質变化; transform the chakra by setting different options)
- The **infinite moon reader** (無限月読; start a local web server to continuously serve your slides)
- The **summoning technique** (download remark.js from the web)

You can click the links to know more about them if you want. The jutsu "Moon Reader" may seem a little evil, but that does not mean your slides are evil.

Hand seals (印)

Press h or ? to see the possible ninjutsu you can use in remark.js.



Thanks!

Slides created via the R package **xaringan**.

The chakra comes from **remark.js**, **knitr**, and **R Markdown**.



Metodologías activas en Educación

Resultados significativos del análisis de encuestas

Juan Abasolo, Aintzane Etxebarria,
Aitor Iglesias, Asier Romero

EUDIA ikerketa-taldea



Universidad
del País Vasco
Euskal Herriko
Unibertsitatea

BILBOKO
HEZKUNTZA
FAKULTATEA
FACULTAD
DE EDUCACIÓN
DE BILBAO

2019/06/04