

Introducción a R

4 de octubre de 2016

Santiago Mota
santiago_mota@yahoo.es



Presentación

- Físico. MBA y Master BI y BigData EOI
- Consultor freelance
- Co-organizador del Meetup de Usuarios de R de Madrid en Medialab
- Director del Master en BI y BigData Online Tenerife EOI
- santiago_mota@yahoo.es
- <http://es.linkedin.com/in/santiagomota>

Índice

1. ¿Por qué aprender R?
2. Casos de uso
3. R como entorno de programación
4. Primeros pasos
5. Donde Continuar

Capítulo 1

¿Por qué aprender R?

Principales razones I

- Es gratuito y multiplataforma
- El número 5 de los 10 mas usados en 2016 ([fuente](#))
- Entorno de programación
- Se usa no sólo para programar
- Se puede ampliar a medida
- Auditable
- Muy superior a excel

Principales razones II

- Reporting semiautomático
- Copiar y pegar código
- Empezó en 1993
- No se puede cerrar
- Apoyo de otras empresas
- Comunidad
- Time to market

Desventajas

- El nombre
- Uso de memoria (relativo)
- Fuentes dispersas
- Mantenimiento de determinados paquetes
- Curva de aprendizaje lenta
- Paquetes especializados (dplyr, data.table)
- Python

Capítulo 2

Casos de uso

Gapminder

- Hans Rosling
- <https://www.gapminder.org/videos/the-joy-of-stats/>
- Minuto 28
- Código de ejemplo:
<https://github.com/mages/googleVis/blob/master/demo/WorldBank.R>
- Reducido: <https://www.youtube.com/watch?v=jbkSRLYSojo>
- Librería googleVis

Markdown

- Genera directamente doc, html y pdf
- Con la salida incluida
- Publicar directamente en rpubs.com
- Incluido en Rstudio
- [Tutorial de R Markdown](#)

Shiny

- Incorporable como html
- Interactivo
- Muy visual
- Shiny Gallery: <http://shiny.rstudio.com/gallery/>

rCharts

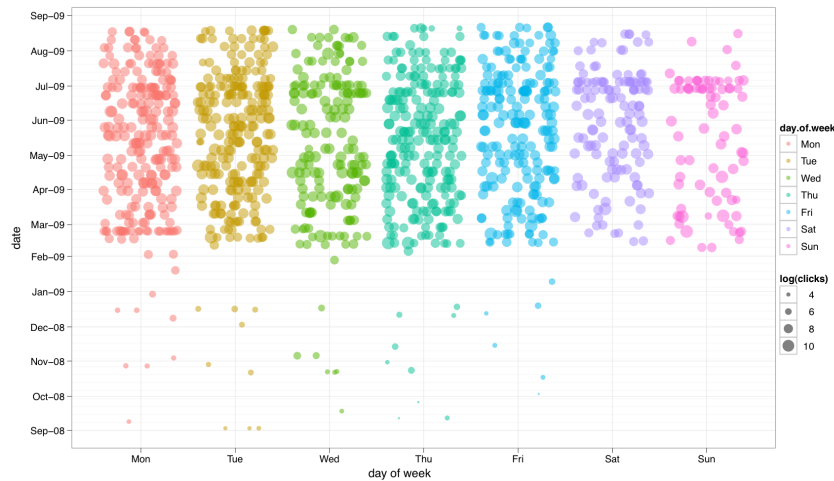
- Gráficos generales
- Interactivos
- Ejemplos: <https://github.com/ramnathv/rCharts>

Leaflet

- Muy fáciles de usar
- Mapas interactivos
- Leaflet y Shiny. Ruta GPS:
<https://rcrastinate.shinyapps.io/GPXshiny/> y código

ggplot2

- Gráficos de alta calidad
- Totalmente personalizables



Webs

- [R bloggers](#)
- [KD nuggets](#)
- [Data Science Central](#)
- [GitHub](#)
- [Kaggle](#)

Capítulo 3

R como entorno de programación

Lenguaje R

- Es un lenguaje orientado a objetos. Cada objeto se guarda con un nombre.
- Es un lenguaje interpretado. Las instrucciones se ejecutan en la consola sin necesidad de compilar.
- El proyecto se aloja en <http://www.r-project.org>

Software necesario

- Primero se instala R. Disponible para varias plataformas.
<http://cran.rstudio.com>
- Posteriormente se instala R Studio desde la web:
<http://www.rstudio.com/products/rstudio/download>
- Es necesario instalar en primer lugar R y posteriormente R Studio
- Diferencia entre instalar y cargar

Rstudio

The screenshot displays the RStudio environment with the following components:

- Script Editor:** Contains R code for data cleaning and documentation. The code includes comments about the source of the data (Kaggle Bosch Production Line Performance) and a description of the data (150 observations of 5 variables).
- Environment:** Shows the loaded data object 'iris' with 150 observations and 5 variables.
- Console:** Displays the R version (3.3.1) and the output of the 'str' function applied to the 'iris' object, showing its structure as a data frame with 150 rows and 5 columns.
- Documentation Pane:** Shows the documentation for the 'str' function, titled 'Compactly Display the Structure of an Arbitrary R Object'. It includes a description, usage, and arguments.

```

1 #####
2 ## https://www.kaggle.com/c/bosch-production-line-performance
3 ## Bosch Production Line Performance
4 ## Santiago Mota
5 ## santiago_mota@yahoo.es
6 ## https://es.linkedin.com/in/santiagonota/en
7
8
9 # Reduce manufacturing failures
10
11 # A good chocolate soufflé is decadent, delicious, and delicate. But, it's a
12 # challenge to prepare. When you pull a disappointingly deflated dessert out of
13 # the oven, you instinctively retrace your steps to identify at what point you
14 # went wrong. Bosch, one of the world's leading manufacturing companies, has an
15 # imperative to ensure that the recipes for the production of its advanced
16 # mechanical components are of the highest quality and safety standards. Part of
17 # doing so is closely monitoring its parts as they progress through the
18 # manufacturing processes.
19
20 # Bosch production line
21
22 # Because Bosch records data at every step along its assembly lines, they have
23 # the ability to apply advanced analytics to improve these manufacturing
24 # processes. However, the intricacies of the data and complexities of the
25 # production line pose problems for current methods.
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

```

R version 3.3.1 (2016-06-21) -- "Bug in Your Hair"
Copyright (C) 2016 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R es un software libre y viene sin GARANTIA ALGUNA.
Usted puede redistribuirlo bajo ciertas circunstancias.
Escriba 'license()' o 'licence()' para detalles de distribución.

R es un proyecto colaborativo con muchos contribuyentes.
Escriba 'contributors()' para obtener más información y
'citation()' para saber cómo citar R o paquetes de R en publicaciones.

Escriba 'demo()' para demostraciones, 'help()' para el sistema on-line de ayuda,
o 'help.start()' para abrir el sistema de ayuda HTML con su navegador.
Escriba 'q()' para salir de R.

> ?str
[1] "iris"
> class(iris)
[1] "data.frame"
>

```

Environment

Global Environment

Data

iris 150 obs. of 5 variables

Files **Plots** **Packages** **Help** **Viewer**

R: Compactly Display the Structure of an Arbitrary R Object

Find in Topic

R Documentation

Compactly Display the Structure of an Arbitrary R Object

Description

Compactly display the internal **structure** of an R object, a diagnostic function and an alternative to [summary](#) (and to some extent, [dput](#)). Ideally, only one line for each 'basic' structure is displayed. It is especially well suited to compactly display the (abbreviated) contents of (possibly nested) lists. The idea is to give reasonable output for **any** R object. It calls [args](#) for (non-primitive) function objects.

`strOptions()` is a convenience function for setting [options](#)(`str = .`), see the examples.

Usage

```
str(object, ...)
```

S3 method for class 'data.frame'

```
str(object, ...)
```

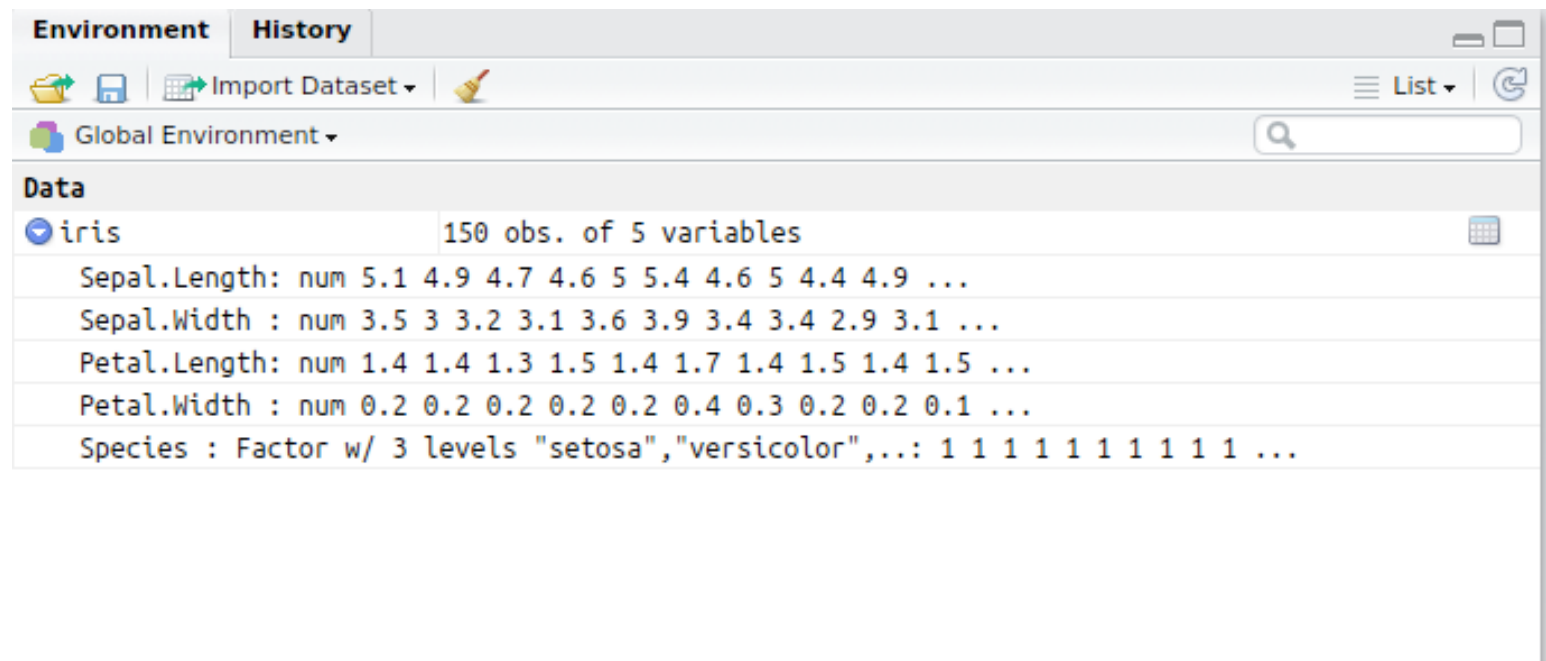
Default S3 method:

```
str(object, max.level = NA,
     vec.len = str0$vec.len, digits.d = str0$digits.d,
     nchar.max = 128, give.attr = TRUE,
     give.head = TRUE, give.length = give.head,
     width = getOption("width"), nest.lev = 0,
     indent.str = paste(rep.int(" ", max(0, nest.lev + 1)),
                        collapse = "..."),
     comp.str = "$ ", no.list = FALSE, envir = baseenv(),
     strict.width = str0$strict.width,
     formatNum = str0$formatNum, list.len = 99, ...)
```

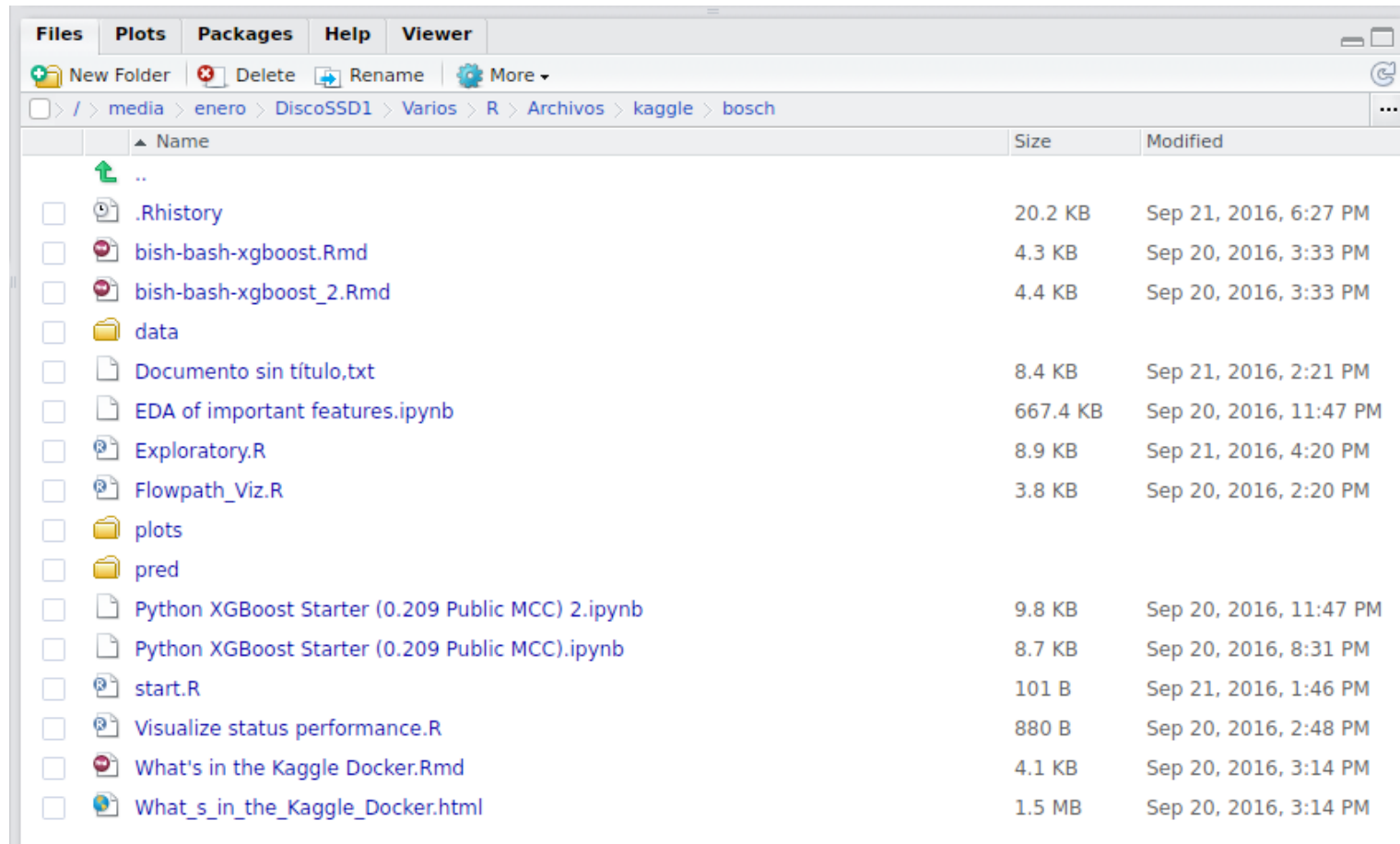
`strOptions(strict.width = "no", digits.d = 3, vec.len = 4,
 formatNum = function(x, ...)
 format(x, trim = TRUE, drop0trailing = TRUE, ...))`

Arguments

Ventana entorno



Ventana inferior derecha



	Name	Size	Modified
	..		
<input type="checkbox"/>	.Rhistory	20.2 KB	Sep 21, 2016, 6:27 PM
<input type="checkbox"/>	bish-bash-xgboost.Rmd	4.3 KB	Sep 20, 2016, 3:33 PM
<input type="checkbox"/>	bish-bash-xgboost_2.Rmd	4.4 KB	Sep 20, 2016, 3:33 PM
<input type="checkbox"/>	data		
<input type="checkbox"/>	Documento sin título.txt	8.4 KB	Sep 21, 2016, 2:21 PM
<input type="checkbox"/>	EDA of important features.ipynb	667.4 KB	Sep 20, 2016, 11:47 PM
<input type="checkbox"/>	Exploratory.R	8.9 KB	Sep 21, 2016, 4:20 PM
<input type="checkbox"/>	Flowpath_Viz.R	3.8 KB	Sep 20, 2016, 2:20 PM
<input type="checkbox"/>	plots		
<input type="checkbox"/>	pred		
<input type="checkbox"/>	Python XGBoost Starter (0.209 Public MCC) 2.ipynb	9.8 KB	Sep 20, 2016, 11:47 PM
<input type="checkbox"/>	Python XGBoost Starter (0.209 Public MCC).ipynb	8.7 KB	Sep 20, 2016, 8:31 PM
<input type="checkbox"/>	start.R	101 B	Sep 21, 2016, 1:46 PM
<input type="checkbox"/>	Visualize status performance.R	880 B	Sep 20, 2016, 2:48 PM
<input type="checkbox"/>	What's in the Kaggle Docker.Rmd	4.1 KB	Sep 20, 2016, 3:14 PM
<input type="checkbox"/>	What_s_in_the_Kaggle_Docker.html	1.5 MB	Sep 20, 2016, 3:14 PM

Comandos de consola

- Elegir directorio de trabajo: Ctrl+Shift+H
- Ejecutar la linea seleccionada: Ctrl+Enter
- Zoom in: Ctrl++
- Zoom out: Ctrl+-
- Ultimo comando: Flecha arriba
- Ver objeto: Pinchar en ese objeto dentro de la ventana Enviromment (o View(objeto))
- Borrar un objeto: rm(objeto)
- ?

Objetos en memoria

- Asignación: `datos <- 10`
- Datasets incorporados: `data("iris")`
- `head(iris)` / `tail(iris)`
- Clase del objeto: `class(iris)`
- Información del objeto: `str(iris)` y `summary(iris)`
- Gráfico del objeto: `plot(iris)`
- Editar estilo excel: `fix(iris)`
- Operaciones directas en la consola: `10 * 12`

Análisis básico

```
> data("iris")
> summary(iris)
  Sepal.Length   Sepal.Width   Petal.Length   Petal.Width   Species
Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50
1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50
Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50
Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500

> str(iris)
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species      : Factor w/ 3 levels "setosa","versicolor",...: 1 1 1 1 1 1 1 1 1 1 ...

> 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

> View(iris)
```


Mas operaciones básicas

- Editar estilo excel: `fix(iris)`
- Instalar un paquete: `install.packages(lattice)`
- Cargar paquete: `library(lattice)`
- Demo del paquete: `demo(lattice)`
- Numero de filas/columnas: `nrow(lattice)/ncol(lattice)`

Capítulo 4

Primeros pasos

Tipos de datos

- Character
- Numeric (numeros reales)
- Integer
- Complex
- Logical(True/False)

Números

- Cuando no se definen, los números en R son números reales de doble precisión)
- Si queremos que el número sea entero, hay que definirlo con el sufijo L (1L)

Valores especiales

- Inf representa el infinito ($1/0$ sería Inf).
- Existe Inf y -Inf.
- NA representa que el valor no existe.
- NaN que representa un valor indefinido o desconocido (Not a Number).

Atributos

- Todos los objetos tienen atributos y se listan con `attribute()`. Los mas importantes son:
- Nombre: `names(iris)`
- Clase: `class(iris)`
- Longitud: `length(iris)`

Vectores

- Un vector agrupa elementos del mismo tipo
- Se pueden crear de diferentes formas: `x <- c(1,2)` `x <- 1:2`
`assign("x", c(1, 2))`

```
> x <- c(1,2)
> x
[1] 1 2
> x <- 1:2
> x
[1] 1 2
> ?vector
> assign("x", c(1, 2))
> x
[1] 1 2
```

Selección de elementos

- Por nombre: `sample_df[c('a', 'c')]`
- Por columna: `sample_df[, 1]`
- Por fila: `sample_df[1,]`
- Una fila: `sample_df$a`

Factores

- Los factores es un tipo de datos que permite representar datos categóricos
- Pueden ser ordenados o no
- Se pueden modificar los niveles
- `x <- factor(1:3, labels=c("A", "B", "C"))`

Listas

- Las listas es un tipo de vector que contiene elementos de diferentes tipos
- `X <- list('a', TRUE, 1)`

Data frame

- Puede entenderse como una matrix en la que se permiten distintos tipos de columnas
- Es el tipo mas utilizado
- Se pueden leer objetos con: `read.table()` o `read.csv()`

Funciones

- Las funciones se definen con `x ← funtion()`
- Se pueden dar valores por defecto
- Pueden agruparse en paquetes

Gráficos

- Están los del paquete básico y los de lattice y ggplot2
- `plot(objeto)`
- `?plot`
- `?par`

Capítulo 5

Donde continuar

Librerías importantes

- `ggplot`
- `data.table`
- `dplyr`
- `xgboost`
- `caret`

Bibliografía

- [Introducción a R](#) (castellano)
- [R para principiantes](#) (castellano)
- [An introduction to R](#) (inglés)
- [R programming for data science](#) (inglés)
- [R for data science](#) (inglés / pago)
- [R For Dummies](#) (inglés / pago)

Webs

- [R bloggers](#)
- [KD nuggets](#)
- [Data Science Central](#)
- [GitHub](#)
- [Kaggle](#)

Chuletas

- Sobre R en general: [Una](#), [dos](#), [tres](#) y [cuatro](#)
- [Varias de Rstudio](#). Entre ellas Markdown, Rstudio o Shiny
- [ggplot2](#)
- [Expresiones regulares](#)
- [dplyr](#)
- [data.table](#)
- En Rstudio (\Help\Cheatsheets)

MOOC

- Introducción a R ([Datacamp](#))
- Introduction to R ([Microsoft](#))
- [R programming](#) Johns Hopkins. Peng y Leek ([data science](#))
- [Statistical Learning](#). Stanford.

Otros

- Asociación de usuarios de R de España
- Meetup Grupo de Usuarios de R de Madrid
- Jornadas nacionales de usuarios de R
- [Stackoverflow](#) (poner [r] en la búsqueda)
- Lista de correo: r-help-es@r-project.org
- Pautas de Google en el estilo de programación

GRACIAS

Datos de contacto:

Santiago Mota Herce

Twitter: @mota_santiago

E-mail: santiago_mota@yahoo.es

LinkedIn: <https://es.linkedin.com/in/santiagomota>

Capítulo 6

Extras

Kaggle

- [Kaggle](#)
- Marchamo “de facto” para data science (primeros = TRABAJO)
- Mas de 50.000 usuarios en todo el mundo (creciendo)
- Zona de test para los algoritmos mas avazandos (xgboost)
- [What has Kaggle learned from 2 million machine learning models?](#)
- [Lessons Learned from Running Hundreds of Kaggle Competitions](#)

Kaggle

- Kaggle datasets
- Leaderboard
- Kernels
- Forum
- Trabajos
- Coste: \$100.000 aprox