semana 4

Luis Ambrocio

6/8/2021

Contents

editando variables de texto

```
cameraData <- read.csv("D:/luism/Descargas/Fixed_Speed_Cameras.csv")
names(cameraData)
```

```
## [1] "ï..X"       "Y"          "fid"        "address"    "direction"
## [6] "street"     "crossstree" "intersecti"
```

Arreglando vectores de caracteres - strsplit ()

- Bueno para dividir automáticamente nombres de variables
- Parámetros importantes: *x*, *split*

```
splitNames = strsplit(names(cameraData),"\\.")
splitNames[[5]]
```

```
## [1] "direction"
```

```
splitNames[[1]]
```

```
## [1] "ï" ""  "X"
```

Aparte rápido - listas

```
mylist <- list(letters = c("A", "b", "c"), numbers = 1:3, matrix(1:25, ncol = 5))
head(mylist)
```

```
## $letters
## [1] "A" "b" "c"
##
## $numbers
```

```
## [1] 1 2 3
##
## [[3]]
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    6   11   16   21
## [2,]    2    7   12   17   22
## [3,]    3    8   13   18   23
## [4,]    4    9   14   19   24
## [5,]    5   10   15   20   25
```

```
mylist[1]
```

```
## $letters
## [1] "A" "b" "c"
```

```
mylist$letters
```

```
## [1] "A" "b" "c"
```

```
mylist[[1]]
```

```
## [1] "A" "b" "c"
```

http://www.biostat.jhsph.edu/~ajaffe/lec_winterR/Lecture%203.pdf

Arreglando vectores de caracteres - sapply ()

- Aplica una función a cada elemento en un vector o lista
- Parámetros importantes: *X*, *FUN*

```
splitNames[[6]][1]
```

```
## [1] "street"
```

```
firstElement <- function(x){x[1]}
sapply(splitNames,firstElement)
```

```
## [1] "ï"          "Y"          "fid"        "address"    "direction"
## [6] "street"     "crossstree" "intersecti"
```

sub()

- Parametros importantes: *pattern*, *replacement*, *x*

```
sub("\\.","",names(cameraData))
```

```
## [1] "ï.X"        "Y"          "fid"        "address"    "direction"
## [6] "street"     "crossstree" "intersecti"
```

gsub()

```
testName <- "this_is_a_test"
sub("_","",testName)
```

```
## [1] "thisis_a_test"
```

```
gsub("_","",testName)
```

```
## [1] "thisisatest"
```

grep(),grepl(), se usan para coincidencias

```
grep("Alameda",cameraData$intersecti)
```

```
## [1]  4  5 36
```

```
table(grepl("Alameda",cameraData$intersecti))
```

```
##
## FALSE  TRUE
##    77     3
```

```
cameraData2 <- cameraData[!grepl("Alameda",cameraData$intersecti),]
head(cameraData2)
```

```
##       ï..X       Y fid                         address direction      street
## 1 -8534734 4760333   1       S CATON AVE & BENSON AVE       N/B   Caton Ave
## 2 -8534751 4760324   2       S CATON AVE & BENSON AVE       S/B   Caton Ave
## 3 -8535640 4760713   3 WILKENS AVE & PINE HEIGHTS AVE       E/B Wilkens Ave
## 6 -8522544 4765716   6        ERDMAN AVE & N MACON ST       E/B      Erdman
## 7 -8522544 4765739   7        ERDMAN AVE & N MACON ST       W/B      Erdman
## 8 -8529948 4774675   8      N CHARLES ST & E LAKE AVE       S/B     Charles
##     crossstree                intersecti
## 1   Benson Ave     Caton Ave & Benson Ave
## 2   Benson Ave     Caton Ave & Benson Ave
## 3 Pine Heights Wilkens Ave & Pine Heights
## 6     Macon St         Erdman  & Macon St
## 7     Macon St         Erdman  & Macon St
## 8     Lake Ave         Charles & Lake Ave
```

```
grep("Alameda",cameraData$intersecti,value=TRUE)
```

```
## [1] "The Alameda  & 33rd St"   "E 33rd  & The Alameda"
## [3] "Harford \n & The Alameda"
```

```
grep("JeffStreet",cameraData$intersecti)
```

```
## integer(0)
```

```
length(grep("JeffStreet",cameraData$intersecti))
```

```
## [1] 0
```

http://www.biostat.jhsph.edu/~ajaffe/lec_winterR/Lecture%203.pdf

Funciones de cadena más útiles

```
library(stringr)
nchar("Jeffrey Leek")
```

```
## [1] 12
```

```
substr("Jeffrey Leek",1,7)
```

```
## [1] "Jeffrey"
```

```
paste("Jeffrey","Leek")
```

```
## [1] "Jeffrey Leek"
```

```
paste0("Jeffrey","Leek")
```

```
## [1] "JeffreyLeek"
```

```
str_trim("Jeff      ")
```

```
## [1] "Jeff"
```

Puntos importantes sobre el texto en conjuntos de datos

- Los nombres de las variables deben ser
  - Todo en minúsculas cuando sea posible
  - Descriptivo (diagnóstico versus Dx)
  - No duplicado
  - No tener guiones bajos ni puntos ni espacios en blanco
- Variables con valores de caracteres
  - Por lo general, debe convertirse en variables de factor (depende de la aplicación)
  - Debe ser descriptivo (use VERDADERO / FALSO en lugar de 0/1 y Hombre / Mujer versus 0/1 o M / F)

regex (expresiones regulares)

Expresiones regulares

- Las expresiones regulares se pueden considerar como una combinación de literales y *metacaracteres*
- Para establecer una analogía con el lenguaje natural, piense en un texto literal que forma las palabras de este idioma y los metacaracteres que definen su gramática.
- Las expresiones regulares tienen un rico conjunto de metacaracteres

Literales

El patrón más simple consta solo de literales. El literal "nuclear" coincidiría con las siguientes líneas:

```
Ooh. I just learned that to keep myself alive after a
nuclear blast! All I have to do is milk some rats
then drink the milk. Aweosme. :}

Laozi says nuclear weapons are mas macho

Chaos in a country that has nuclear weapons -- not good.

my nephew is trying to teach me nuclear physics, or
possibly just trying to show me how smart he is
so I'll be proud of him [which I am].

lol if you ever say "nuclear" people immediately think
DEATH by radiation LOL
```

El literal "Obama" coincidiría con las siguientes líneas

```
Politics r dum. Not 2 long ago Clinton was sayin Obama
was crap n now she sez vote 4 him n unite? WTF?
Screw em both + Mcain. Go Ron Paul!

Clinton conceeds to Obama but will her followers listen??

Are we sure Chelsea didn't vote for Obama?

thinking ... Michelle Obama is terrific!

jetlag..no sleep...early mornig to starbux..Ms. Obama
was moving
```

- El patrón más simple consta solo de literales; se produce una coincidencia si la secuencia de literales se produce en cualquier parte del texto que se está probando
- ¿Y si solo queremos la palabra “Obama”? o frases que terminan en la palabra “Clinton”, o “clinton” o “clinto”?

Necesitamos una forma de expresar - límites de palabras de espacios en blanco - conjuntos de literales - el principio y el final de una línea - alternativas (“guerra” o “paz”) ¡Metacaracteres al rescate!

- ^ representa el comienzo de una línea.

```
^i think
```

marcara las lineas

```
i think we all rule for participating
i think i have been outed
i think this will be quite fun actually
i think i need to go to work
i think i first saw zombo in 1999.
```

- $ represanta el final de una linea

```
morning$
```

marcara las lineas

```
well they had something this morning
then had to catch a tram home in the morning
dog obedience school in the morning
and yes happy birthday i forgot to say it earlier this morning
I walked in the rain this morning
good morning
```

- Podemos enumerar un conjunto de caracteres que aceptaremos en un momento dado al marcar.

```
[Bb][Uu][Ss][Hh]
```

marcara las lineas

```
The democrats are playing, "Name the worst thing about Bush!"
I smelled the desert creosote bush, brownies, BBQ chicken
BBQ and bushwalking at Molonglo Gorge
Bush TOLD you that North Korea is part of the Axis of Evil
I’m listening to Bush - Hurricane (Album Version)
```

```
^[Ii] am
```

marcará

```
i am so angry at my boyfriend i can’t even bear to
look at him

i am boycotting the apple store

I am twittering from iPhone

I am a very vengeful person when you ruin my sweetheart.

I am so over this. I need food. Mmmm bacon...
```

- Del mismo modo, puede especificar un rango de letras [a-z] o [a-zA-Z]; observe que el orden no importa

```
^[0-9][a-zA-Z]
```

marcará

```
7th inning stretch
2nd half soon to begin. OSU did just win something
3am - cant sleep - too hot still.. :(
5ft 7 sent from heaven
1st sign of starvagtion
```

- Cuando se usa al comienzo de una clase de carácter([]), el "^" también es un metacarácter e indica caracteres coincidentes que NO están en la clase indicada.

```
[^?.]$
```

marcará

```
i like basketballs
6 and 9
dont worry... we all die anyway!
Not in Baghdad
helicopter under water? hmmm
```

- "." es usado para referirse a cualquier caracter

```
9.11
```

marcará

```
its stupid the post 9-11 rules
if any 1 of us did 9/11 we would have been caught in days.
NetBios: scanning ip 203.169.114.66
Front Door 9:11:46 AM
Sings: 0118999881999119725...3 !
```

- Esto no significa "pipe" en el contexto de las expresiones regulares; en cambio, se traduce como "or"; podemos usarlo para combinar dos expresiones, las subexpresiones se llaman alternativas

```
flood|fire
```

marcará

```
is firewire like usb on none macs?
the global flood makes sense within the context of the bible
yeah ive had the fire on tonight
... and the floods, hurricanes, killer heatwaves, rednecks, gun nuts, etc.
```

```
flood|earthquake|hurricane|coldfire
```

marcará

```
Not a whole lot of hurricanes in the Arctic.
We do have earthquakes nearly every day somewhere in our State
hurricanes swirl in the other direction
coldfire is STRAIGHT!
'cause we keep getting earthquakes
```

- Las alternativas pueden ser expresiones reales y no solo literales.

```
^[Gg]ood|[Bb]ad
```

marcará

```
good to hear some good knews from someone here
Good afternoon fellow american infidels!
good on you-what do you drive?
Katie... guess they had bad experiences...
my middle name is trouble, Miss Bad News
```

Las subexpresiones a menudo se incluyen entre paréntesis para restringir las alternativas.

```
^([Gg]ood|[Bb]ad)
```

marcará

```
bad habbit
bad coordination today
good, becuase there is nothing worse than a man in kinky underwear
Badcop, its because people want to use drugs
Good Monday Holiday
Good riddance to Limey
```

- El signo de interrogación indica que la expresión indicada es opcional

```
[Gg]eorge( [Ww]\.)? [Bb]ush
```

queríamos hacer coincidir un "." como un período literal; para hacer eso, tuvimos que "escapar" del metacarácter, precediéndolo con una barra invertida. En general, tenemos que hacer esto para cualquier metacarácter que queramos incluir en nuestra coincidencia

marcará

```
i bet i can spell better than you and george bush combined
BBC reported that President George W. Bush claimed God told him to invade I
a bird in the hand is worth two george bushes
```

- Los signos * y + son metacaracteres que se utilizan para indicar repetición; * significa "cualquier número, incluido ninguno, del artículo" y + significa "al menos uno del artículo"

```
(.*)
```

marcará

```
anyone wanna chat? (24, m, germany)
hello, 20.m here... ( east area + drives + webcam )
(he means older men)
()
```

```
[0-9]+ (.*)[0-9]+
```

marcará

```
working as MP here 720 MP battallion, 42nd birgade
so say 2 or 3 years at colleage and 4 at uni makes us 23 when and if we fin
it went down on several occasions for like, 3 or 4 *days*
Mmmm its time 4 me 2 go 2 bed
```

- {} se denominan cuantificadores de intervalo; el permítanos especificar el número mínimo y máximo de coincidencias de una expresión

```
[Bb]ush( +[^ ]+ +){1,5} debate
```

marcará

```
Bush has historically won all major debates he's done.
in my view, Bush doesn't need these debates..
bush doesn't need the debates? maybe you are right
That's what Bush supporters are doing about the debate.
Felix, I don't disagree that Bush was poorly prepared for the debate.
indeed, but still, Bush should have taken the debate more seriously.
Keep repeating that Bush smirked and scowled during the debate
```

- m, n significa al menos m pero no más de n coincidencias
- m significa exactamente m coincidencias
- m, significa al menos m coincidencias
- En la mayoría de las implementaciones de expresiones regulares, los paréntesis no solo limitan el alcance de las alternativas dividido por un "|", sino que también se pueden usar para "recordar" el texto que coincide con la subexpresión incluida.
- Nos referimos al texto emparejado con \1, \2, etc.

Entonces la expresion

```
+([a-zA-Z]+) +\1 +
```

marcará

```
time for bed, night night twitter!
blah blah blah blah
my tattoo is so so itchy today
i was standing all all alone against the world outside...
hi anybody anybody at home
estudiando css css css css.... que desastritooooo
```

- El * es "codicioso", por lo que siempre coincide con la cadena _ más larga_ posible que satisfaga la expresión regular. Entonces

```
^s(.*)s
```

matches

```
sitting at starbucks
setting up mysql and rails
studying stuff for the exams
spaghetti with marshmallows
stop fighting with crackers
sore shoulders, stupid ergonomics
```

La codicia de * se puede apagar con?, Como en

```
^s(.*)s
```

marca

```
sitting at starbucks
setting up mysql and rails
studying stuff for the exams
spaghetti with marshmallows
stop fighting with crackers
sore shoulders, stupid ergonomics
```

Resumen

- Las expresiones regulares se utilizan en muchos idiomas diferentes; no es exclusivo de R.
- Las expresiones regulares se componen de literales y metacaracteres que representan conjuntos o clases de caracteres / palabras
- El procesamiento de texto a través de expresiones regulares es una forma muy poderosa de extraer datos de fuentes "hostiles" (no todos los datos vienen como un archivo CSV)
- Usado con las funciones `grep`,`grepl`, `sub`,`gsub` y otras que involucran la búsqueda de cadenas de texto (Gracias a Mark Hansen por algo de material en esta conferencia).

fechas

fecha actual

```
d1 = date()
d1
```

```
## [1] "Sat Aug 07 19:26:38 2021"
```

```
class(d1)
```

```
## [1] "character"
```

en clase date

```
d2 = Sys.Date()
d2
```

```
## [1] "2021-08-07"
```

```
class(d2)
```

```
## [1] "Date"
```

Dar formato a las fechas

`%d` = dia como numero (0-31), `%a` = día de la semana abreviado,`%A` = día de la semana no abreviado, `%m` = mes (00-12),`%b` = mes abreviado, `%B` = mes sin abreviar,`%y` = año de 2 dígitos, `%Y` = año de cuatro dígitos

```
format(d2,"%a %b %d")
```

```
## [1] "sáb. ago. 07"
```

creando fechas

```
x = c("1jun.1960", "2jun.1960", "31mar.1960", "30jul.1960"); z = as.Date(x, "%d%b%Y")
z
```

```
## [1] "1960-06-01" "1960-06-02" "1960-03-31" "1960-07-30"
```

```
z[1] - z[2]
```

```
## Time difference of -1 days
```

```
as.numeric(z[1]-z[2])
```

```
## [1] -1
```

convertiendo a juliano

```
weekdays(d2)
```

```
## [1] "sábado"
```

```
months(d2)
```

```
## [1] "agosto"
```

```
julian(d2)
```

```
## [1] 18846
## attr(,"origin")
## [1] "1970-01-01"
```

libreria Lubridate

```
library(lubridate); ymd("20140108")
```

```
## [1] "2014-01-08"
```

```
mdy("08/04/2013")
```

```
## [1] "2013-08-04"
```

```
dmy("03-04-2013")
```

```
## [1] "2013-04-03"
```

lidiar con los tiempos

```
ymd_hms("2011-08-03 10:15:03")
```

```
## [1] "2011-08-03 10:15:03 UTC"
```

```
ymd_hms("2011-08-03 10:15:03",tz="Pacific/Auckland")
```

```
## [1] "2011-08-03 10:15:03 NZST"
```

http://www.r-statistics.com/2012/03/do-more-with-dates-and-times-in-r-with-lubridate-1-1-0/

Algunas funciones tienen una sintaxis ligeramente diferente

```
x = dmy(c("1jan2013", "2jan2013", "31mar2013", "30jul2013"))
wday(x[1])
```

```
## [1] 3
```

```
wday(x[1],label=TRUE)
```

```
## [1] mar\\.
## Levels: dom\\. < lun\\. < mar\\. < mié\\. < jue\\. < vie\\. < sáb\\.
```

Notas y otros recursos

- More information in this nice lubridate tutorial http://www.r-statistics.com/2012/03/do-more-with-dates-and-times-in-r-with-lubridate-1-1-0/
- The lubridate vignette is the same content http://cran.r-project.org/web/packages/lubridate/vignettes/lubridate.html
- En última instancia, desea sus fechas y horas como clase “Fecha” o las clases “POSIXct”, “POSIXlt”. Para obtener más información, escriba `? POSIXlt`

Recursos de datos

Sitios de gobierno abierto

- Naciones Unidas http://data.un.org/
- EE. UU. Http://www.data.gov/
  - [Lista de ciudades / estados con datos abiertos] (http://simplystatistics.org/2012/01/02/list-of-cities-states-with-open-data-help-me-find/)
- Reino Unido http://data.gov.uk/
- Francia http://www.data.gouv.fr/
- Ghana http://data.gov.gh/
- Australia http://data.gov.au/
- Alemania https://www.govdata.de/
- Hong Kong http://www.gov.hk/en/theme/psi/datasets/
- Japón http://www.data.go.jp/
- Muchos más http://www.data.gov/opendatasites

Gapminder

Gapminder Foundation es una empresa sin fines de lucro registrada en Estocolmo, Suecia , que promueve el desarrollo global sostenible y el logro de los Objetivos de Desarrollo del Milenio de las Naciones Unidas mediante un mayor uso y comprensión de las estadísticas y otra información sobre el desarrollo social, económico y ambiental a nivel local, nacional y niveles globales

http://www.gapminder.org/

Datos de encuestas de Estados Unidos

http://www.asdfree.com/

Mercado de infomchips

http://www.infochimps.com/marketplace

Colecciones de científicas de datos

Hilary Mason http://bitly.com/bundles/hmason/1 * Peter Skomoroch https://delicious.com/pskomoroch/dataset * Jeff Hammerbacher http://www.quora.com/Jeff-Hammerbacher/Introduction-to-Data-Science-Data-Sets * Gregory Piatetsky-Shapiro http://www.kdnuggets.com/gps.html * http://blog.mortardata.com/post/67652898761/6-dataset-lists-curated-by-data-scientists

Colecciones más especializadas

- Stanford Large Network Data
- UCI Machine Learning
- Conjuntos de datos de KDD Nugets
- CMU Statlib
- [Ómnibus de expresión genética] (http://www.ncbi.nlm.nih.gov/geo/)
- Datos de ArXiv
- Conjuntos de datos públicos en Amazon Web Services

Algunas APIs con referencias R

- twitter and twitteR package
- figshare and rfigshare
- PLoS and rplos
- rOpenSci
- Facebook and RFacebook
- Google maps and RGoogleMaps