

2ª sesión

Ficheiro Rmarkdown: ficheiro de texto que combina código de R (que se pode executar) e texto. Está pensado para facer informes que inclúen cálculos actualizados.

Video sobre RMarkdown: [AQUÍ](#)

Algunhas cousas:

R garda información nun formato propio usando os comandos `save()` e recupéraa usando `load()`

```
# crear un obxecto datos, usando mtcars, colección de datos de coches incluída en R básico
datos=mtcars
nombres=names(datos) # gardar nun vector os nomes das variables de datos
# garda os obxectos nombres e datos nun ficheiro, con formato Rdata, formato propio de R
save(nombres,datos,file="proba.Rdata")
#borro eses obxectos da memoria, para ver despois como se fai a recuperación
rm(nombres, datos)
```

Agora vou recuperar os dous obxectos gardados usando o comando `load()`

```
# carga en memoria os obxectos que se gardaron en "proba.Rdata"
load("proba.Rdata")
```

Limpar todo o que haxa en memoria:

```
# Pódese facer coa escoba do Environment, pero tamén con este código:
rm(list=ls())
```

Atallo cntrl+alt+I

Para inserir un “chunk” onde escribir o código para Rmarkdown

Fin da preliminares: Inicio da sesión

Practicarase a preparación de variables co paquete `dplyr`

Video de introdución a `dplyr`: [AQUÍ](#)

```
# cargar o paquete
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

Crear un data.frame de datos. Usarei de novo os datos de `mtcars`

```
datos=mtcars
```

1º seleccionar variables

Escoller un grupo de variables para traballar con elas separadamente:

```
# coller datos e entón seleccionar as variables mpg,cyl,hp,wt,am, finalmente gardar en datos2
datos %>% select(mpg,cyl,hp,wt,am)->datos2
```

No seguinte explícase como eliminar variables, simplemente poñendo o signo - diante do nome da variable, ou dun vector de nomes de variables:

```
names(datos)[4:7]
```

```
## [1] "hp"    "drat" "wt"    "qsec"
```

```
datos %>% select(-qsec)->borrar # elimina qsec e garda en borrar
```

```
datos %>% select(-c(qsec,gear))-> borrar # elimina qsec e gear e garda en borrar
```

Ata aquí borrar columnas, agora borrar filas, ou sexa, eliminar individuos da base de datos

filter

O comando `filter()` filtra rexistros da base de datos, empregando diferentes criterios:

```
datos %>% select(mpg,cyl,am) %>% filter(cyl==8) # colle soamente os coches con cyl igual a 8
```

```
##           mpg cyl am
## Hornet Sportabout 18.7  8  0
## Duster 360        14.3  8  0
## Merc 450SE        16.4  8  0
## Merc 450SL        17.3  8  0
## Merc 450SLC       15.2  8  0
## Cadillac Fleetwood 10.4  8  0
## Lincoln Continental 10.4  8  0
## Chrysler Imperial 14.7  8  0
## Dodge Challenger  15.5  8  0
## AMC Javelin        15.2  8  0
## Camaro Z28         13.3  8  0
## Pontiac Firebird   19.2  8  0
## Ford Pantera L     15.8  8  1
## Maserati Bora       15.0  8  1
```

```
# o simbolo == indica comparación a==b pregúntase se a e b son iguais, e respondera FALSE
```

```
# filtrar con dúas condicións cyl igual a 8 e am igual a 1
```

```
datos %>% select(mpg,cyl,am) %>% filter(cyl==8,am==1)
```

```
##           mpg cyl am
## Ford Pantera L 15.8  8  1
## Maserati Bora  15.0  8  1
```

```
# filtrar con algunha de dúas condicións ou cyl igual a 8, ou am igual a 1
```

```
datos %>% select(mpg,cyl,am) %>% filter(cyl==8|am==1)
```

```
##           mpg cyl am
## Mazda RX4      21.0  6  1
## Mazda RX4 Wag  21.0  6  1
## Datsun 710      22.8  4  1
## Hornet Sportabout 18.7  8  0
## Duster 360     14.3  8  0
## Merc 450SE     16.4  8  0
## Merc 450SL     17.3  8  0
## Merc 450SLC    15.2  8  0
## Cadillac Fleetwood 10.4  8  0
## Lincoln Continental 10.4  8  0
## Chrysler Imperial 14.7  8  0
```

```
## Fiat 128      32.4  4  1
## Honda Civic  30.4  4  1
## Toyota Corolla 33.9  4  1
## Dodge Challenger 15.5  8  0
## AMC Javelin  15.2  8  0
## Camaro Z28   13.3  8  0
## Pontiac Firebird 19.2  8  0
## Fiat X1-9    27.3  4  1
## Porsche 914-2 26.0  4  1
## Lotus Europa 30.4  4  1
## Ford Pantera L 15.8  8  1
## Ferrari Dino  19.7  6  1
## Maserati Bora 15.0  8  1
## Volvo 142E   21.4  4  1
```

Poñer ! diante dunha expresión lóxica implica negar: filtrar con algunha de dúas condicións ou cyl di.

```
datos %>% select(mpg,cyl,am) %>% filter(!cyl==8|am==1)
```

```
##          mpg cyl am
## Mazda RX4    21.0   6  1
## Mazda RX4 Wag 21.0   6  1
## Datsun 710    22.8   4  1
## Hornet 4 Drive 21.4   6  0
## Valiant      18.1   6  0
## Merc 240D     24.4   4  0
## Merc 230      22.8   4  0
## Merc 280      19.2   6  0
## Merc 280C     17.8   6  0
## Fiat 128      32.4   4  1
## Honda Civic   30.4   4  1
## Toyota Corolla 33.9   4  1
## Toyota Corona 21.5   4  0
## Fiat X1-9     27.3   4  1
## Porsche 914-2 26.0   4  1
## Lotus Europa  30.4   4  1
## Ford Pantera L 15.8   8  1
## Ferrari Dino   19.7   6  1
## Maserati Bora  15.0   8  1
## Volvo 142E    21.4   4  1
```

O mesmo de antes

```
datos %>% select(mpg,cyl,am) %>% filter(cyl!=8|am==1)
```

```
##          mpg cyl am
## Mazda RX4    21.0   6  1
## Mazda RX4 Wag 21.0   6  1
## Datsun 710    22.8   4  1
## Hornet 4 Drive 21.4   6  0
## Valiant      18.1   6  0
## Merc 240D     24.4   4  0
## Merc 230      22.8   4  0
## Merc 280      19.2   6  0
## Merc 280C     17.8   6  0
## Fiat 128      32.4   4  1
## Honda Civic   30.4   4  1
```

```
## Toyota Corolla 33.9 4 1
## Toyota Corona 21.5 4 0
## Fiat X1-9 27.3 4 1
## Porsche 914-2 26.0 4 1
## Lotus Europa 30.4 4 1
## Ford Pantera L 15.8 8 1
## Ferrari Dino 19.7 6 1
## Maserati Bora 15.0 8 1
## Volvo 142E 21.4 4 1
```

```
# agora a negación afecta a toda a expresión: filtrar o que non cumpre cyl igual a 8, ou am igual a 1
datos %>% select(mpg,cyl,am) %>%
  filter(!(cyl==8|am==1))
```

```
##          mpg cyl am
## Hornet 4 Drive 21.4 6 0
## Valiant 18.1 6 0
## Merc 240D 24.4 4 0
## Merc 230 22.8 4 0
## Merc 280 19.2 6 0
## Merc 280C 17.8 6 0
## Toyota Corona 21.5 4 0
```

Tamén se fixo un exercicio.

Cousas que non se fixeron, pero que estaría ben ter feitas

Crear variables novas con `dplyr`: comando `'mutate()'`

```
# datos%>%mutate(variablenova1=variablevella1+2*variablevella2,variablenova2=log(variablevella1)^2)->da
datos%>%mutate(pesoenkilg=wt*0.453592,kmxlitro=mpg*0,42514)->datos
#creouse novas variables, coas unidades en sistema métrico decimal: pasa librar a kilos e millas por ga
```

Calcular para datos agrupados polos niveis dun atributo: `summarise()`

Por exemplo, calcular o consumo medio dos coches segundo o seu numero de cilindros:

```
#facer varios calculos para unha variable con un comando, separando os resultados polos niveis dun atri
# datos %>% group_by(atributo)%>%
#   summarise(N = n(), media = mean(variable), maximo = max(variable), cuartil1 = quantile(variable,0.2)

#consumo medio dos coches segundo o seu numero de cilindros
datos %>% group_by(cyl)%>%summarise(media = mean(mpg))
```

```
## # A tibble: 3 x 2
##   cyl media
##   <dbl> <dbl>
## 1     4  26.7
## 2     6  19.7
## 3     8  15.1
```

```
# contar nº de coches de cada cantidade de cilindros, o seu consumo medio e a desviación típica do seu
datos %>% group_by(cyl)%>% summarise(N = n(), media = mean(mpg), dtpeso = sd(wt))
```

```
## # A tibble: 3 x 4
##   cyl      N media dtpeso
##   <dbl> <int> <dbl> <dbl>
## 1     4    11  26.7  0.570
```

```
## 2      6      7  19.7  0.356
## 3      8     14  15.1  0.759
```

```
# agora agrupando por cilindros e por automatico ou non (am)
```

```
dados %>% group_by(cyl,am)%>% summarise(N = n(), media = mean(mpg), dtpeso = sd(wt))
```

```
## `summarise()` has grouped output by 'cyl'. You can override using the `groups`
## argument.
```

```
## # A tibble: 6 x 5
```

```
## # Groups:   cyl [3]
```

```
##      cyl      am      N media dtpeso
##   <dbl> <dbl> <int> <dbl> <dbl>
## 1     4      0      3  22.9  0.408
## 2     4      1      8  28.1  0.409
## 3     6      0      4  19.1  0.116
## 4     6      1      3  20.6  0.128
## 5     8      0     12  15.0  0.768
## 6     8      1      2  15.4  0.283
```