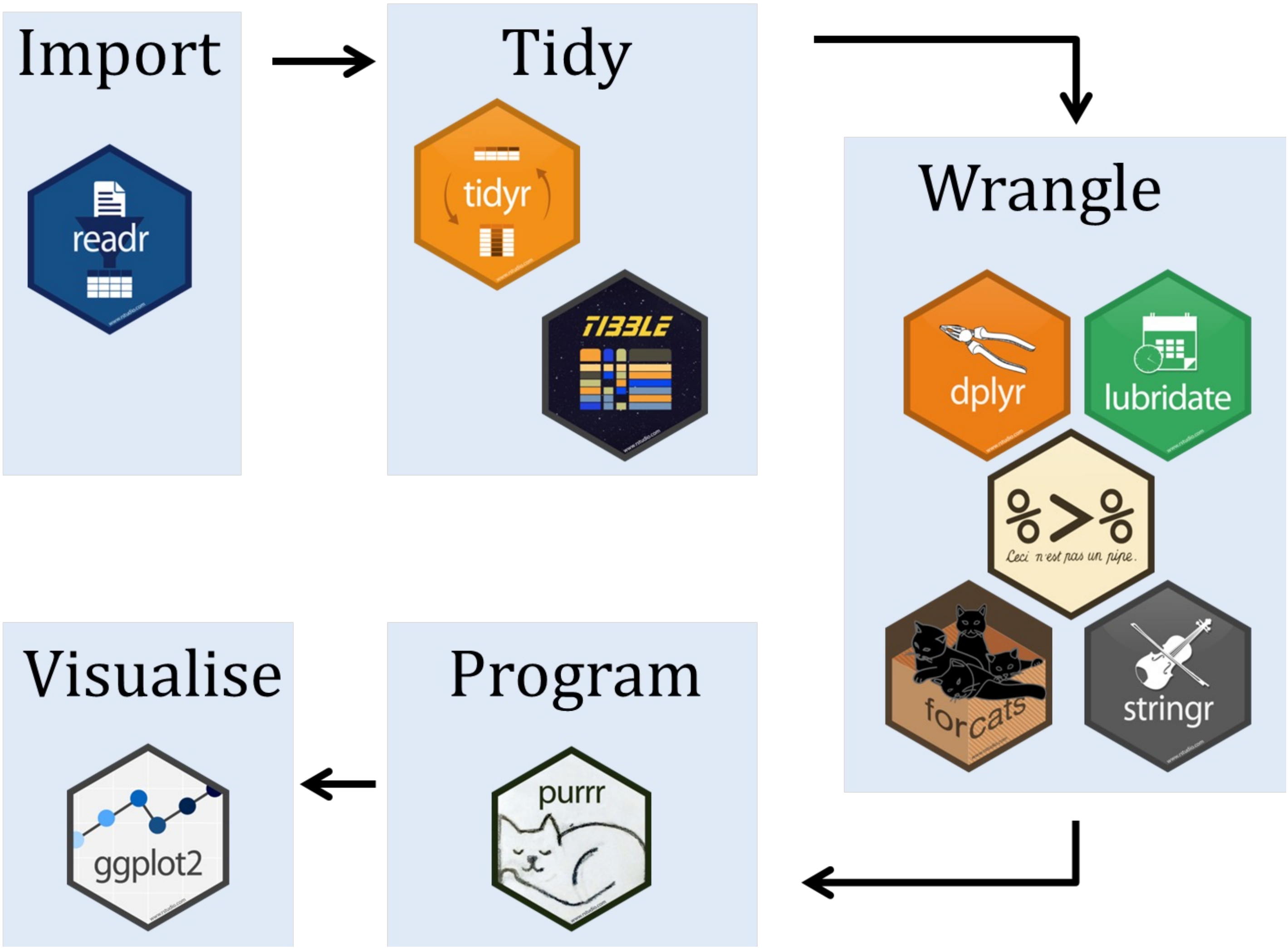
MANIPULATING AND CLEANING DATA: FORCATS

Coral del Val Muñoz

Dpt. Ciencias de la Computación e Inteligencia Artificial,
Universidad de Granada

What is tidyverse?

Colección de paquetes con una gramática, filosofía y estructura similar. Se basan en (Wickham and others 2014).



Import

Tidy

Wrangle

Program

Visualise

http://www.seec.uct.ac.za/r-tidyverse

forcats

- Función: da una serie de herramentas para resolver problemas típicos con factores, incluyendo el cambio de niveles y valores

```
# The easiest way to get readr is to install the whole tidyverse
install.packages("tidyverse")

# Alternatively, install just readr
install.packages(”forcats")

#Usage
library(tidyverse)
```

What is forcats()?

`fct_reorder()`: Reordering a factor by another variable.

`fct_infreq()`: Reordering a factor by the frequency of values.

`fct_relevel()`: Changing the order of a factor by hand.

`fct_lump()`: Collapsing the least/most frequent values of a factor into “other”.

count()

- **Select rows in a** dataframe (df).
- Dataset `starwars.`
- Column `species`

```
starwars %>%
filter(!is.na(species)) %>%
count(species, sort = TRUE)
#> # A tibble: 37 x 2
#> species n
#> <chr> <int>
 #> 1 Human 35
 #> 2 Droid 6
 #> 3 Gungan….
```

fct_lump(): Combining levels

We can use fct_lump() to “lump” (collapse) all the infrequent values of variable into one factor, “other.”

```
starwars %>%
mutate(skin_color = fct_lump(skin_color, n = 5)) %>%
count(skin_color, sort = TRUE)


#> # A tibble: 6 x 2
#> skin_color n #> <fct> <int>
#> 1 Other 41
#> 2 fair 17
#> 3 light 11
#> 4 dark 6
#> 5 green 6
#> 6 grey 6
```

fct_lump(): Combining levels that have at least a certain proportion

```
starwars %>%
mutate(skin_color = fct_lump(skin_color, prop = .1, other_level
= "extra")) %>% count(skin_color, sort = TRUE)


#> # A tibble: 3 x 2
#> skin_color n
#>   <fct> <int>
#> 1 extra 59
#> 2 fair 17
#> 3 light 11
```

Chance “other” for something else

ejercicio

- Instala y carga la librería tidyverse
- Usa el dataset starwars
- Calcula:
- Intenta averiguar si la media del peso (average_mass) difiere según el color de ojos. Nos interesan los datos solo para los 6 colores de ojos mayoritarias. Elimina los NA.

Ejercicio: pistas

- Crea una variable en el dataset starwars eye_color que resuma los 6 colores mas importantes
- Usa esa variable para agrupar los datos
- Calcula la media de peso de esos grupos

ejercicio

```
avg_mass_eye_color <- starwars %>%
filter(!is.na(mass)) %>%
  mutate(eye_color = fct_lump(eye_color, n =
6)) %>%
  group_by(eye_color) %>%
  summarise(mean_mass = mean(mass, na.rm =
TRUE))

avg_mass_eye_color
```

fct_reorder(): Reordering factors

We can use fct_reorder() if we want to order a variable according to a factor, for example according to the avg_mass_eye_color

```
avg_mass_eye_color %>%

mutate(eye_color = fct_reorder(eye_color, mean_mass))
```

```
A tibble: 7 x 2
  eye_color mean_mass
  <fct>         <dbl>
1 black          76.3
2 blue           86.5
3 brown          66.1
4 orange        282. 
5 red            81.4
6 yellow         81.1
7 Other          68.4
```

fct_infreq(): Reordering a factor by the frequency of values

```
starwars %>%

mutate(eye_color = fct_infreq(eye_color))
```

fct_collapse(): messy vectors handling

Messy factors are problem, here a way to solve it.

```
gender <- c("f", "m ", "male ","male", "female", "FEMALE",
"Male", "f", "m")
gender <- as_factor(gender)
gender <- fct_collapse(
           gender,
           Female = c("f", "female", "FEMALE"),
           Male   = c("m ", "m", "male ", "male", "Male")
           )
fct_count(gender)
```

fct_anon(): anonymization of categories in vectors

For example in some cases when information is sensiblenwe want to anonimize the categories

```
gender <- c("f", "m ", "male ","male", "female", "FEMALE",
"Male", "f", "m")
gender <- as_factor(gender)
gender <- fct_anon(gender)
fct_count(gender)
## # A tibble: 2 x 2
## f n
## <fct> <int>
## 1 1 5
## 2 2 4
```