Clase_3

Joshua Kock

2/7/2019

Contents


```
library(tidyverse)
```

```
## -- Attaching packages ----------------------------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  2.0.1     v dplyr   0.7.8
## v tidyr   0.8.2     v stringr 1.3.1
## v readr   1.3.1     v forcats 0.3.0
```

```
## Warning: package 'tibble' was built under R version 3.5.2
```

```
## -- Conflicts -------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

http://biostat.mc.vanderbilt.edu/wiki/pub/Main/DataSets/ari.html#levels.clin

```
ari <- read.csv("https://raw.githubusercontent.com/vizual-wanderer/6071402_Electiva_II/master/Base_datos
```

1 Funciones con Operador %>%

Pasa un objeto y luego una funcion

```
#str(ari) #Funcion y luego objeto
#ari %>% str() #objeto luego funcion
```

Ejercicio Seleccionen la talla (lgth) y peso de todos los bebes con un peso a nacer menor a 2500g (biwt), solo los primeros 10 resultados

```
ari %>%
  filter(biwt <= 2500) %>%
  select(lgth) %>%
  head(n = 10)
```

```
##    lgth
## 1    48
## 2    47
## 3    44
## 4    50
```

```
## 5    51
## 6    40
## 7    54
## 8    43
## 9    51
## 10   49
```

Revisen la funcion count de dplyr y haga 2 operaciones con ella y explique en sus propias palabras.

```
ari %>%
  count(weight)
```

```
## # A tibble: 4 x 2
##   weight     n
##   <fct>  <int>
## 1 M         27
## 2 N       2281
## 3 Y        150
## 4 <NA>    2094
```

```
ari %>%
  filter(biwt <= 2500) %>%
  select(weight) %>%
  count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1   775
```

Count muestra por defecto los datos perdidos?

Crea un nuevo objeto ari_2 con las variables de ID, peso, leucocitos, linfocitos, circunferencia craneo, peso en gramos al nacer, talla al nacer, temperatura, oximetria, edad en dias, omfalitis y conjuntivitis.

```
ari2 <- ari %>% select(stno, weight, wbco,
                       lpcc, hcir, biwt, lgth, temp, hrat, age, omph, conj)
```

2 Crear variables con funcion mutate()

Ejercicio: Creen una nueva variable biwt_2 que sea la mitad del peso original Hacerlo de forma base R y con mutate()

Miren como funciona mutate con el buscador de ayuda, miren las funciones de ayuda (Helper functions)

```
ari %>%
  mutate(biwt_2 = biwt/2) %>% #tidyverse
  select(biwt_2) %>%
  head(n = 10)
```

```
##      biwt_2
## 1  1297.942
## 2  1112.776
## 3  1149.186
## 4  1205.976
## 5  1750.000
## 6  1750.000
```

```
## 7  1322.683
## 8  1075.000
## 9  1723.318
## 10  825.000
```

```
ari$biwt_3 <- ari$biwt/2#base R
```

Si no se asgina la nueva variable no se guarda en el dataframe.

```
ari %>%
  mutate(h_lgth = lgth/2) %>%
  head(n = 10) %>%
  select(h_lgth)
```

```
##    h_lgth
## 1   27.50
## 2   24.00
## 3   23.50
## 4   22.00
## 5   26.00
## 6   26.50
## 7   27.00
## 8   25.00
## 9   30.75
## 10  25.50
```

```
ari <-
  ari %>%
  mutate(h_lgth = lgth/2)
```

Se pueden crear mas de una variable a la vez

```
ari %>%
  mutate(tasa_linf_bla = lpcc/wbco,
         tasa_peso_talla = wght/lgth,
         pct_cir = hcir/100) %>%
  select(tasa_linf_bla,tasa_peso_talla, pct_cir) %>%
  head(n = 10)
```

```
##    tasa_linf_bla tasa_peso_talla pct_cir
## 1             NA        95.45455   0.310
## 2             NA        48.95833   0.340
## 3             NA        56.38298   0.340
## 4             NA        54.54545   0.325
## 5             NA        57.69231   0.330
## 6             NA        66.98113   0.360
## 7   0.0001369863        79.62963   0.390
## 8   0.0000000000        46.00000   0.340
## 9             NA        89.43089   0.410
## 10            NA        61.76471   0.360
```

3 Helper functions en mutate

1) ifelse
2) recode
3) case_when

3.1 Ifelse

Ifelse es una funcion que crea una variable indicadora 1/0 basada en una condicion

```
ari %>%
  mutate(bajo_peso = if_else(wght < 2500,1,0)) %>%
  select(bajo_peso) %>%
  head(n = 10)
```

```
##    bajo_peso
## 1          0
## 2          1
## 3          0
## 4          1
## 5          0
## 6          0
## 7          0
## 8          1
## 9          0
## 10         0
```

Ejericio con ifelse genere una variable indicadora para datos perdidos de recuento globulos blancos wbco y contar estos datos (usar funcion is.na)

```
ari %>%
  mutate(perdido = if_else(!is.na(wbco), 1, 0)) %>%
  select(perdido) %>%
  head(n = 10)
```

```
##    perdido
## 1        0
## 2        0
## 3        0
## 4        1
## 5        0
## 6        1
## 7        1
## 8        1
## 9        1
## 10       0
```

genere una variable indicadora para la variable clin identificado los pacientes que se les tomo muestra (sample o yes) que sean de etiopia. Cuantos son?

```
ari %>%
  filter(country == "Ethiopia") %>%
  mutate(clin_pos = if_else(str_detect(clin, "Yes"),
                                       1,0)) %>%
  count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1   816
```

3.2 recode

Recode como el nombre lo indica recodifica una variable que sea un factor en el caso de R. sintaxis: recode(.x, ... default = NULL, missing = NULL) .x: un vector (variable) a recodificar ...: especificacion de recodificar (valor actual = nuevo valor) .default: si lo especificas los valores que no cumplan con la regla anterior adquieren este valor. missing: si especificado los valores perdidos se les asignara este valor.

Estamos recodificando Y como 1 y N como 0 en la variable de enfermedad

```
ari %>%
  mutate(enfermedad = recode(sickc, "Y" = 1, "N" = 0)) %>%
  select(sickc, enfermedad) %>%
  head(n = 10)
```

```
##    sickc enfermedad
## 1      N          0
## 2      N          0
## 3      N          0
## 4      Y          1
## 5      N          0
## 6      Y          1
## 7      Y          1
## 8      Y          1
## 9      Y          1
## 10     N          0
```

Ejercicio: recodigica la variable impcl en 4 grupos P, S, M y otros y nombrar la variable patron:

No P,S or M No P,S,M-oth P only S only M only P and S P and M M and S P,M and S

```
ari %>%
  mutate(patron = recode(impcl,
                 "P only" = "P",
                 "S only" = "S",
                 "M only" = "M",
                 .default = "otro")
         ) %>%
  count(patron)
```

```
## # A tibble: 5 x 2
##   patron     n
##   <fct>  <int>
## 1 otro    3026
## 2 M         54
## 3 P       1030
## 4 S        434
## 5 <NA>       8
```

3.3 Case when

case when es una funcion para variables mas complejas que ifelse y recode, es util cuando la nueva variable es la funcion de multiples variables.

Sintaxis: case_when(...) a la izquierda van las variables y reglas que tiene que cumplir a la derecha la variable en la que se tiene que crear esto se separa con ~

Quiero saber que bebes de etiopia tienen peso normal al nacer (2500 a 3800g) en etiopia

```r
ari %>%
  mutate(
    etiopia_peso = case_when(
    country == "Ethiopia" & wght >= 2500 & wght <= 3800 ~ "et_peso_norm"
  )
) %>%
  select(etiopia_peso) %>%
  head(n = 10)
```

```
##    etiopia_peso
## 1          <NA>
## 2          <NA>
## 3  et_peso_norm
## 4          <NA>
## 5  et_peso_norm
## 6  et_peso_norm
## 7          <NA>
## 8          <NA>
## 9          <NA>
## 10 et_peso_norm
```

ejercicio case when determinar los bebes de papua nueva guinea que tengan un peso mayor a 3000 y no se hicieron la prueba clinica o la negaron (clin).

```r
ari %>%
  mutate(variable = case_when(
    country == "Papua New Guinea" & wght >3000 ~ "pap_peso")
  ) %>%
  count(variable)
```

```
## # A tibble: 2 x 2
##   variable     n
##   <chr>    <int>
## 1 pap_peso  2090
## 2 <NA>      2462
```

determinar los pacientes con conteo de blancos mayor a 10mil con una frecuencia respiratoria rr mayor a 60 con puncion lumbar positiva (lp.pos)

```r
ari %>%
  mutate(sepsis = case_when(
    wbco > 10000 & rr >60 & lp.pos == 1 ~ "sepsis_snc"
  )) %>%
  count(sepsis)
```

```
## # A tibble: 2 x 2
##   sepsis         n
##   <chr>      <int>
## 1 sepsis_snc     8
## 2 <NA>        4544
```