

Clase_5

Joshua Kock

2/22/2019

Cargar paquetes para la sesion

```
library(tidyverse)
```

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

## v ggplot2 3.1.0      v purrr  0.3.0
## v tibble  2.0.1      v dplyr  0.8.0.1
## v tidyr   0.8.2      v stringr 1.4.0
## v readr   1.3.1      v forcats 0.4.0

## Warning: package 'tibble' was built under R version 3.5.2
## Warning: package 'purrr' was built under R version 3.5.2
## Warning: package 'dplyr' was built under R version 3.5.2
## Warning: package 'stringr' was built under R version 3.5.2
## Warning: package 'forcats' was built under R version 3.5.2

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

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date
```

Cargar la base de datos.

```
cirugia_fci_2018 <- read_csv("https://raw.githubusercontent.com/vizual-wanderer/6071402_Electiva_II/mas")
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   Edad = col_double(),
##   Numero_ingreso = col_double(),
##   Tiempo_quirurgico = col_double(),
##   Tiempo_Hosp = col_double(),
##   Estancia_urgencias = col_double(),
##   Estancia_piso = col_double(),
##   Ingreso_Ambulatorio = col_double(),
##   Cateter_Peritoneal = col_double(),
##   Examen = col_logical(),
##   Pancreatitis = col_logical(),
##   Lavados = col_double(),
##   Interconsulta = col_double(),
##   Mortalidad = col_double(),
```

```
## Momento_Reintervencion = col_double(),
## ISO = col_double()
## )

## See spec(...) for full column specifications.
cirugia_fci_2018 <- cirugia_fci_2018 %>%
  mutate(Fecha_cirugia = mdy(Fecha_cirugia),
         Marca_temporal = mdy_hm(Marca_temporal))
```

Ejercicio de factor.

```
cirugia_fci_2018 <- cirugia_fci_2018 %>%
  mutate(Cirujano_fct = factor(Cirujano))

#Caracteres.
typeof(cirugia_fci_2018$Cirujano)

## [1] "character"
class(cirugia_fci_2018$Cirujano)

## [1] "character"
str(cirugia_fci_2018$Cirujano)

## chr [1:2442] "Felipe Casas" "Carlos Roman" "Felipe Casas" ...
attributes(cirugia_fci_2018$Cirujano)

## NULL
#Factores.
typeof(cirugia_fci_2018$Cirujano_fct)

## [1] "integer"
class(cirugia_fci_2018$Cirujano_fct)

## [1] "factor"
str(cirugia_fci_2018$Cirujano_fct)

## Factor w/ 8 levels "Akram Kadamani",...: 5 3 5 5 5 6 6 6 6 4 ...
attributes(cirugia_fci_2018$Cirujano_fct)

## $levels
## [1] "Akram Kadamani"      "Bayron Guerra"      "Carlos Roman"
## [4] "Ciro Andres Murcia"  "Felipe Casas"       "Manuel Mosquera"
## [7] "Nathaly Ramirez"    "Paulo Cabrera"
##
## $class
## [1] "factor"
cirugia_fci_2018 %>%
  filter(Cirujano_fct == 8) %>%
  count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1     0
```

```
cirugia_fci_2018 %>%
  filter(Cirujano_fct == "Paulo Cabrera") %>%
  count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1   444
```

Ejercicio: crear una variable llamada mes con la funcion month de lubridate (variable fecha_cirugia), asigna esta variable a la base y explorar con las funciones previas

```
cirugia_fci_2018 <- cirugia_fci_2018 %>%
  mutate(mes = month(Fecha_cirugia))

typeof(cirugia_fci_2018$mes)
```

```
## [1] "double"
```

```
str(cirugia_fci_2018$mes)
```

```
## num [1:2442] 1 1 1 1 1 1 1 1 1 1 ...
```

```
class(cirugia_fci_2018$mes)
```

```
## [1] "numeric"
```

```
attributes(cirugia_fci_2018$mes)
```

```
## NULL
```

Crear una variable que sea mes_fct y asignar esta a la base de datos. (explorar con las funciones previas)

```
cirugia_fci_2018 <- cirugia_fci_2018 %>%
  mutate(mes_fct = factor(mes))

typeof(cirugia_fci_2018$mes_fct)
```

```
## [1] "integer"
```

```
str(cirugia_fci_2018$mes_fct)
```

```
## Factor w/ 12 levels "1","2","3","4",...: 1 1 1 1 1 1 1 1 1 1 ...
```

```
class(cirugia_fci_2018$mes_fct)
```

```
## [1] "factor"
```

```
attributes(cirugia_fci_2018$mes_fct)
```

```
## $levels
```

```
## [1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10" "11" "12"
```

```
##
```

```
## $class
```

```
## [1] "factor"
```

Analisis de datos exploratorios.

Analisis de una variable (continua)

Tiempo quirurgico

```
cirugia_fci_2018 %>%  
  summarize_at(  
    .vars = vars(Tiempo_quirurgico),  
    .funs = funs(mean, sd, min, max, .args = list(na.rm = TRUE))  
  )
```

```
## Warning: funs() is soft deprecated as of dplyr 0.8.0  
## please use list() instead  
##  
## # Before:  
## funs(name = f(.))  
##  
## # After:  
## list(name = ~f(.))  
## This warning is displayed once per session.  
  
## # A tibble: 1 x 4  
##   mean    sd   min   max  
##   <dbl> <dbl> <dbl> <dbl>  
## 1  62.6  40.6     1   700
```

Que pasa si no se incluye: .args=list(na.rm=TRUE)

```
cirugia_fci_2018 %>%  
  summarize_at(  
    .vars = vars(Tiempo_quirurgico),  
    .funs = funs(mean, sd, min, max)  
  )
```

```
## # A tibble: 1 x 4  
##   mean    sd   min   max  
##   <dbl> <dbl> <dbl> <dbl>  
## 1    NA    NA    NA    NA
```

Variable categoricas o discretas.

Tabla de frecuencias bidireccionales “tabulación cruzada”, son importante para la calidad de los datos Cuando crea un analisis categórico var a partir de una unica “entrada” categorica.

Las tablas bidireccionales nos muestran si lo hicimos correctamente Tablas bidireccionales utiles para comprender patrones de omision en encuestas

group_by(var1) %>% count(var2)

```
cirugia_fci_2018 %>%  
  group_by(Sexo) %>%  
  count(Tipo_de_Cirugia)
```

```
## # A tibble: 4 x 3  
## # Groups:   Sexo [2]  
##   Sexo      Tipo_de_Cirugia     n
```

```
##   <chr>      <chr>      <int>
## 1 Femenino  Programada    390
## 2 Femenino  Urgencias     805
## 3 Masculino Programada    435
## 4 Masculino Urgencias     812
```

No se ve muy bien podemos arreglarlo con `spread`

```
cirugia_fci_2018 %>%
  group_by(Sexo) %>%
  count(Tipo_de_Cirugia) %>%
  as_factor() %>%
  spread(Sexo,n)
```

```
## # A tibble: 2 x 3
##   Tipo_de_Cirugia Femenino Masculino
##   <chr>          <int>      <int>
## 1 Programada      390        435
## 2 Urgencias       805        812
```

Categorica con categorica

```
cirugia_fci_2018 %>%
  group_by(Cirujano) %>%
  count(Clasificacion_Herida_Quirurgica) %>%
  spread(Cirujano, n) %>%
  filter(!is.na(Clasificacion_Herida_Quirurgica))
```

```
## # A tibble: 4 x 9
##   Clasificacion_H~ `Akram Kadamani` `Bayron Guerra` `Carlos Roman`
##   <chr>            <int>          <int>          <int>
## 1 Contaminada      66            36            93
## 2 Limpia           165            49            87
## 3 Limpia Contamin~ 193            79           286
## 4 Sucia            8             NA             1
## # ... with 5 more variables: `Ciro Andres Murcia` <int>, `Felipe
## #   Casas` <int>, `Manuel Mosquera` <int>, `Nathaly Ramirez` <int>, `Paulo
## #   Cabrera` <int>
```

categorica y continua.

```
group_by(var_categorica) %>% summarize_at(.vars = vars(var_continua))
```

```
cirugia_fci_2018 %>%
  group_by(Cirujano) %>%
  summarize_at(
    .vars = vars(Tiempo_quirurgico),
    .funs = funs(mean, sd, min, max, .args=list(na.rm=TRUE))
  )
```

```
## # A tibble: 8 x 5
##   Cirujano          mean    sd  min  max
##   <chr>          <dbl> <dbl> <dbl> <dbl>
## 1 Akram Kadamani    47.6  33.2    1   300
```

## 2	Bayron Guerra	69.7	22.5	30	180
## 3	Carlos Roman	61.4	30.5	15	180
## 4	Ciro Andres Murcia	46.3	24.1	10	180
## 5	Felipe Casas	52.5	26.4	10	180
## 6	Manuel Mosquera	72.3	55.9	10	320
## 7	Nathaly Ramirez	54.9	26.9	1	180
## 8	Paulo Cabrera	81.6	50.6	30	700