Clase_4: Group_by y summarize

Joshua Kock

2/14/2019

```
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()
```

```
library(lubridate)
```

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

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

```
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))
```

Analisis de datos con funcion `group_by`.

`group_by()` convierte un objeto de dataframe en grupos. Despues de la agrupacion, las funciones que ejecutas en el data frame se realizan "por grupo"

- Parte del paquete dplyr dentro de tidyverse; no es parte de la Base R
- Funciona mejor con los pipe `%>%` y la funcion de `summarize()`.

Sintaxis basica:

`group_by(objecto, variable por las cuales agruparas separadas por commas)`.

Normalmente, las variables para agrupar son variables de caracteres, factores o integros.

Ejemplo:

```
cirugia_fci_2018 %>%
  count(Cirujano)
```

```
## # A tibble: 8 x 2
##   Cirujano               n
##   <chr>              <int>
## 1 Akram Kadamani       432
## 2 Bayron Guerra        164
## 3 Carlos Roman         467
## 4 Ciro Andres Murcia   121
## 5 Felipe Casas         241
## 6 Manuel Mosquera      364
## 7 Nathaly Ramirez      209
## 8 Paulo Cabrera        444
```

```
cirugia_fci_2018 %>%
  group_by(Cirujano) %>%
  count(Cirujano)
```

```
## # A tibble: 8 x 2
## # Groups:   Cirujano [8]
##   Cirujano               n
##   <chr>              <int>
## 1 Akram Kadamani       432
## 2 Bayron Guerra        164
## 3 Carlos Roman         467
## 4 Ciro Andres Murcia   121
## 5 Felipe Casas         241
## 6 Manuel Mosquera      364
## 7 Nathaly Ramirez      209
## 8 Paulo Cabrera        444
```

Por si mismo `group_by()` no hace mucho solo imprime datos ejemplo:

```
cirugia_fci_2018 %>%
  group_by(Cirujano, Tipo_de_Cirugia, ISO)
```

```
## # A tibble: 2,442 x 38
## # Groups:   Cirujano, Tipo_de_Cirugia, ISO [31]
##    Marca_temporal      Fecha_cirugia Cirujano Residente  Edad Sexo
##    <dttm>              <date>        <chr>    <chr>     <dbl> <chr>
##  1 2018-01-09 09:08:00 2018-01-01    Felipe ~ Manuel A~    76 Feme~
##  2 2018-01-09 09:10:00 2018-01-01    Carlos ~ Laura Ra~    51 Masc~
```

```
##  3 2018-01-09 09:15:00 2018-01-02    Felipe ~ Juan Man~    62 Feme~
##  4 2018-01-09 09:19:00 2018-01-02    Felipe ~ Juan Man~    48 Feme~
##  5 2018-01-09 10:23:00 2018-01-02    Felipe ~ Manuel A~    49 Masc~
##  6 2018-01-09 10:28:00 2018-01-03    Manuel ~ William ~    52 Feme~
##  7 2018-01-09 10:34:00 2018-01-03    Manuel ~ Manuel A~    83 Masc~
##  8 2018-01-09 10:36:00 2018-01-03    Manuel ~ William ~    55 Feme~
##  9 2018-01-09 10:51:00 2018-01-03    Manuel ~ Carlos A~    69 Masc~
## 10 NA                  2018-01-08    Ciro An~ Manuel A~    64 Feme~
## # ... with 2,432 more rows, and 32 more variables: Numero_ingreso <dbl>,
## #   EPS_POS <chr>, Prepagadas <chr>, SISBEN_Regimen_Subsidiado <chr>,
## #   Diagnostico <chr>, Procedimiento_Quirurgico <chr>,
## #   Tiempo_quirurgico <dbl>, Tipo_de_Cirugia <chr>, Reintervencion <chr>,
## #   Clasificacion_Herida_Quirurgica <chr>, ASA <chr>,
## #   Complicacion_Quirurgica <chr>, Tiempo_Hosp <dbl>,
## #   Estancia_urgencias <dbl>, Estancia_piso <dbl>,
## #   Apendice_Complicada <chr>, Egreso_con_Dx_adecuado <chr>,
## #   Ingreso_Ambulatorio <dbl>, Egreso_por_Cirugia_Ambulatoria <chr>,
## #   Acceso_vascular <chr>, Cateter_Peritoneal <dbl>, Examen <lgl>,
## #   Pancreatitis <lgl>, Traqueostomia <chr>, Gastrostomia <chr>,
## #   Lavados <dbl>, Interconsulta <dbl>, Mortalidad <dbl>,
## #   Momento_Reintervencion <dbl>, ISO <dbl>, Reingreso <chr>,
## #   Commentarios <chr>
```

Pero una vez que se agrupa un objeto, todas las funciones subsiguientes se ejecutan por separado "por grupo"

```
cirugia_fci_2018 %>%
  group_by(Cirujano, Tipo_de_Cirugia, ISO) %>%
  count()
```

```
## # A tibble: 31 x 4
## # Groups:   Cirujano, Tipo_de_Cirugia, ISO [31]
##    Cirujano       Tipo_de_Cirugia   ISO     n
##    <chr>          <chr>           <dbl> <int>
##  1 Akram Kadamani Programada          3     1
##  2 Akram Kadamani Programada         NA   264
##  3 Akram Kadamani Urgencias           3     1
##  4 Akram Kadamani Urgencias          NA   166
##  5 Bayron Guerra  Programada         NA    13
##  6 Bayron Guerra  Urgencias          NA   151
##  7 Carlos Roman   Programada         NA    96
##  8 Carlos Roman   Urgencias           1     3
##  9 Carlos Roman   Urgencias           3     1
## 10 Carlos Roman   Urgencias          NA   367
## # ... with 21 more rows
```

A continuacion, usaremos la funcion `class()` para mostrar si el dataframe esta agrupado

- Por ahora, solo piensa en `class()` como una funcion que proporciona informacion sobre un objeto.
- Similar a `typeof()`, pero `class()` proporciona informacion diferente sobre el objeto

```
class(cirugia_fci_2018)
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

```
df_temp <- cirugia_fci_2018 %>%
  group_by(Cirujano, Residente)
```

```r
class(df_temp)
```

```
## [1] "grouped_df" "tbl_df"     "tbl"        "data.frame"
```

Para desagrupar un objeto se usa la funcion `ungroup()`.

```r
df_temp <- ungroup(df_temp)
class(df_temp)
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

Ejericio: Usando group by - Determine los grupos de residentes que hay.

```r
cirugia_fci_2018 %>%
  group_by(Residente) %>%
  count()
```

```
## # A tibble: 24 x 2
## # Groups:   Residente [24]
##    Residente            n
##    <chr>            <int>
##  1 Alirio Zarata       72
##  2 Andrea Suarez       17
##  3 Camila Devia       155
##  4 Camila Rodriguez   102
##  5 Carlos Aguana       21
##  6 Daniel Morales     104
##  7 Diego Delgado       76
##  8 Interno             48
##  9 Jose Sanchez        95
## 10 Juan Lotero        139
## # ... with 14 more rows
```

- Determine los grupo de cirujano y residente que operan.

```r
cirugia_fci_2018 %>%
  group_by(Cirujano, Residente) %>%
  count()
```

```
## # A tibble: 165 x 3
## # Groups:   Cirujano, Residente [165]
##    Cirujano       Residente            n
##    <chr>          <chr>            <int>
##  1 Akram Kadamani Alirio Zarata        8
##  2 Akram Kadamani Andrea Suarez        1
##  3 Akram Kadamani Camila Devia        20
##  4 Akram Kadamani Camila Rodriguez    15
##  5 Akram Kadamani Carlos Aguana        5
##  6 Akram Kadamani Daniel Morales       9
##  7 Akram Kadamani Diego Delgado       19
##  8 Akram Kadamani Interno              4
##  9 Akram Kadamani Jose Sanchez        11
## 10 Akram Kadamani Juan Lotero         30
## # ... with 155 more rows
```

- Determine los grupos de cirujano y residentes con un tiempo quirurgico mayor a 200 min (usar filter).

```r
cirugia_fci_2018 %>%
  group_by(Cirujano, Residente) %>%
  filter(Tiempo_quirurgico > 200) %>%
  count()
```

```
## # A tibble: 25 x 3
## # Groups:   Cirujano, Residente [25]
##    Cirujano        Residente           n
##    <chr>           <chr>           <int>
##  1 Akram Kadamani  Daniel Morales      1
##  2 Akram Kadamani  Juan Lotero         1
##  3 Akram Kadamani  Laura Ramirez       1
##  4 Akram Kadamani  Roman Guerrero      1
##  5 Manuel Mosquera Alirio Zarata       2
##  6 Manuel Mosquera Camila Devia        1
##  7 Manuel Mosquera Daniel Morales      1
##  8 Manuel Mosquera Juan Lotero         2
##  9 Manuel Mosquera Julian Senosain     1
## 10 Manuel Mosquera Laura Ramirez       2
## # ... with 15 more rows
```

Analisis de datos con funcion `summarize()`.

`summarize()` realiza los calculos por filas; luego se colapsa en una sola fila. uso y sintaxis: 'summarize(.data,....)

argumentos: - `.data`: un dataframe; se omitir si se coloca despues de pipe `%>%`. - `...`: Par(es) nombre(s)-valor(es) y su respectiva funcion(es) de resumen. El nombre sera el nombre de la variable en el resultado. El valor debe ser una expresion que devuelve un solo valor como `min()`, `n()` etc.

Valor: (lo que `summarize` devuelve / crea) - Objeto de la misma clase que `.data`. ; el objeto tendra una observacion "por grupo"

funciones utiles en summarize: ?dplyr::summarize

ejemplo:

```r
#Edad promedio de la cohorte 2018
cirugia_fci_2018 %>%
  summarize(prom_edad = mean(Edad, na.rm = TRUE))
```

```
## # A tibble: 1 x 1
##   prom_edad
##       <dbl>
## 1      51.3
```

```r
#Mediana de tiempo quirurgico para el 2018.
cirugia_fci_2018 %>%
  summarize(med_temp_qx = median(Tiempo_quirurgico, na.rm = TRUE))
```

```
## # A tibble: 1 x 1
##   med_temp_qx
##         <dbl>
## 1          60
```

se pueden hacer multiples operaciones

```r
cirugia_fci_2018 %>%
  summarize(prom_edad = mean(Edad, na.rm = TRUE),
            med_temp_qx = median(Tiempo_quirurgico, na.rm = TRUE),
            total = n())
```

```
## # A tibble: 1 x 3
##   prom_edad med_temp_qx total
##       <dbl>       <dbl> <int>
## 1      51.3          60  2442
```

Ejericio: Determine los tiempos promedios, min y max de estadia hospitalaria. La mediana de estancia en piso.

```r
cirugia_fci_2018 %>%
  summarize(
    t_prom_hosp = mean(Estancia_piso, na.rm = TRUE),
    t_median_hosp = median(Estancia_piso, na.rm = TRUE),
    min_est_hosp = min(Estancia_piso, na.rm = TRUE),
    max_est_hosp = max(Estancia_piso, na.rm = TRUE),
    total = n()
  )
```

```
## # A tibble: 1 x 5
##   t_prom_hosp t_median_hosp min_est_hosp max_est_hosp total
##         <dbl>         <dbl>        <dbl>        <dbl> <int>
## 1        2.81             0            0           75  2442
```

Combinando `group_by()` con `summarize()`.

`summarize()` realiza calculos en todas las filas del dataframe y luego colapsa el dataframe en una sola fila Cuando se agrupa el dataframe, `summarize()` realiza calculos en filas dentro de un grupo y luego se colapsa en una sola fila para cada grupo.

ejemplo:

```r
cirugia_fci_2018 %>%
  group_by(Residente) %>%
  summarize(edad_median_pac_resi = mean(Edad, na.rm = TRUE)) %>%
  arrange(desc(edad_median_pac_resi))
```

```
## # A tibble: 24 x 2
##    Residente       edad_median_pac_resi
##    <chr>                          <dbl>
##  1 Carlos Aguana                   55.5
##  2 Juan Lotero                     55.2
##  3 Jose Sanchez                    54.5
##  4 Julian Senosain                 53.7
##  5 Laura Ramirez                   53.3
##  6 Paula Meneses                   52.6
##  7 Paula Florez                    52.4
##  8 Roman Guerrero                  52.3
##  9 Sin Ayudante                    51.9
## 10 Andrea Suarez                   51.5
## # ... with 14 more rows
```

Ejercicio: Determine el tiempo promedio de cirugia de cada cirujano

```
cirugia_fci_2018 %>%
  group_by(Cirujano) %>%
  summarize(
    t_med_quir = mean(Tiempo_quirurgico, na.rm = TRUE),
    t_min_cx = min(Tiempo_quirurgico, na.rm = TRUE),
    t_max_cx = max(Tiempo_quirurgico, na.rm = TRUE)
  )
```

```
## # A tibble: 8 x 4
##   Cirujano           t_med_quir t_min_cx t_max_cx
##   <chr>                   <dbl>    <dbl>    <dbl>
## 1 Akram Kadamani           47.6        1      300
## 2 Bayron Guerra            69.7       30      180
## 3 Carlos Roman             61.4       15      180
## 4 Ciro Andres Murcia       46.3       10      180
## 5 Felipe Casas             52.5       10      180
## 6 Manuel Mosquera          72.3       10      320
## 7 Nathaly Ramirez          54.9        1      180
## 8 Paulo Cabrera            81.6       30      700
```

Determine el tiempo promedio de una cirugia de cada cirujano con residente.

```
cirugia_fci_2018 %>%
  group_by(Cirujano, Residente) %>%
  summarize(
    t_prom_cir_resi = mean(Tiempo_quirurgico, na.rm = TRUE)
  )
```

```
## # A tibble: 165 x 3
## # Groups:   Cirujano [?]
##    Cirujano       Residente        t_prom_cir_resi
##    <chr>          <chr>                      <dbl>
##  1 Akram Kadamani Alirio Zarata               26.2
##  2 Akram Kadamani Andrea Suarez               30
##  3 Akram Kadamani Camila Devia                43
##  4 Akram Kadamani Camila Rodriguez            37.3
##  5 Akram Kadamani Carlos Aguana               50
##  6 Akram Kadamani Daniel Morales              64.4
##  7 Akram Kadamani Diego Delgado               41.1
##  8 Akram Kadamani Interno                     40
##  9 Akram Kadamani Jose Sanchez                50
## 10 Akram Kadamani Juan Lotero                 71.3
## # ... with 155 more rows
```

Combinando summarize() y count ().

La funcion para contar `n()` no toma argumentos y devuelve el tamaño del grupo actual.

numero de complicaciones quirurgicas por cirujano.

```
cirugia_fci_2018 %>%
  group_by(Cirujano, Complicacion_Quirurgica) %>%
  summarize(comp_qx = n())
```

```
## # A tibble: 38 x 3
## # Groups:   Cirujano [?]
##    Cirujano       Complicacion_Quirurgica                      comp_qx
##    <chr>          <chr>                                          <int>
##  1 Akram Kadamani No hubo                                        431
##  2 Akram Kadamani <NA>                                             1
##  3 Bayron Guerra  Falsa ruta en paso de sonda vesical              1
##  4 Bayron Guerra  Fistula biliar por muñon abierto                 1
##  5 Bayron Guerra  Lesion serosa del ciego, requirio cecorrafia     1
##  6 Bayron Guerra  Lesiones intestinales                           1
##  7 Bayron Guerra  Llamado intraoperatorio a urologia              1
##  8 Bayron Guerra  Muerte                                          1
##  9 Bayron Guerra  No hubo                                       158
## 10 Carlos Roman   Lesion via biliar corregida por CPRE            1
## # ... with 28 more rows
```

```
cirugia_fci_2018 %>%
  group_by(Cirujano) %>%
  count(Complicacion_Quirurgica)
```

```
## # A tibble: 38 x 3
## # Groups:   Cirujano [8]
##    Cirujano       Complicacion_Quirurgica                        n
##    <chr>          <chr>                                      <int>
##  1 Akram Kadamani No hubo                                      431
##  2 Akram Kadamani <NA>                                           1
##  3 Bayron Guerra  Falsa ruta en paso de sonda vesical            1
##  4 Bayron Guerra  Fistula biliar por muñon abierto               1
##  5 Bayron Guerra  Lesion serosa del ciego, requirio cecorrafia   1
##  6 Bayron Guerra  Lesiones intestinales                         1
##  7 Bayron Guerra  Llamado intraoperatorio a urologia            1
##  8 Bayron Guerra  Muerte                                        1
##  9 Bayron Guerra  No hubo                                     158
## 10 Carlos Roman   Lesion via biliar corregida por CPRE          1
## # ... with 28 more rows
```

Determine el tipo de cirugia(urgencia, programada) por cirujano.

```
cirugia_fci_2018 %>%
  group_by(Cirujano) %>%
  count(Tipo_de_Cirugia) %>%
  arrange(desc(Tipo_de_Cirugia))
```

```
## # A tibble: 16 x 3
## # Groups:   Cirujano [8]
##    Cirujano           Tipo_de_Cirugia     n
##    <chr>              <chr>           <int>
##  1 Akram Kadamani     Urgencias         167
##  2 Bayron Guerra      Urgencias         151
##  3 Carlos Roman       Urgencias         371
##  4 Ciro Andres Murcia Urgencias         106
##  5 Felipe Casas       Urgencias         222
##  6 Manuel Mosquera    Urgencias         152
##  7 Nathaly Ramirez    Urgencias         199
##  8 Paulo Cabrera      Urgencias         249
##  9 Akram Kadamani     Programada        265
```

```
## 10 Bayron Guerra      Programada         13
## 11 Carlos Roman       Programada         96
## 12 Ciro Andres Murcia Programada         15
## 13 Felipe Casas       Programada         19
## 14 Manuel Mosquera    Programada        212
## 15 Nathaly Ramirez    Programada         10
## 16 Paulo Cabrera      Programada        195
```

```
cirugia_fci_2018 %>%
  group_by(Cirujano, Tipo_de_Cirugia) %>%
  summarize(total = n())
```

```
## # A tibble: 16 x 3
## # Groups:   Cirujano [?]
##    Cirujano           Tipo_de_Cirugia total
##    <chr>              <chr>           <int>
##  1 Akram Kadamani     Programada        265
##  2 Akram Kadamani     Urgencias         167
##  3 Bayron Guerra      Programada         13
##  4 Bayron Guerra      Urgencias         151
##  5 Carlos Roman       Programada         96
##  6 Carlos Roman       Urgencias         371
##  7 Ciro Andres Murcia Programada         15
##  8 Ciro Andres Murcia Urgencias         106
##  9 Felipe Casas       Programada         19
## 10 Felipe Casas       Urgencias         222
## 11 Manuel Mosquera    Programada        212
## 12 Manuel Mosquera    Urgencias         152
## 13 Nathaly Ramirez    Programada         10
## 14 Nathaly Ramirez    Urgencias         199
## 15 Paulo Cabrera      Programada        195
## 16 Paulo Cabrera      Urgencias         249
```

Determine el numero de interconsulta por cada grupo de cirujano.

```
cirugia_fci_2018 %>%
  group_by(Cirujano) %>%
  filter(!is.na(Interconsulta)) %>%
  count(Interconsulta, sort = TRUE)
```

```
## # A tibble: 8 x 3
## # Groups:   Cirujano [8]
##   Cirujano           Interconsulta     n
##   <chr>                      <dbl> <int>
## 1 Carlos Roman                   1    52
## 2 Paulo Cabrera                  1    47
## 3 Manuel Mosquera                1    33
## 4 Felipe Casas                   1    24
## 5 Akram Kadamani                 1    20
## 6 Bayron Guerra                  1    20
## 7 Nathaly Ramirez                1    20
## 8 Ciro Andres Murcia             1    14
```

Determine re-intervencion por cada cirujano.

```
cirugia_fci_2018 %>%
  group_by(Cirujano) %>%
```

```
  filter(Reintervencion != "NO REINTERVENCION") %>%
  count(Reintervencion, sort = TRUE)
```

```
## # A tibble: 59 x 3
## # Groups:   Cirujano [8]
##    Cirujano           Reintervencion          n
##    <chr>              <chr>               <int>
##  1 Paulo Cabrera      Lavado                 42
##  2 Manuel Mosquera    Lavado                 31
##  3 Akram Kadamani     Lavado                 22
##  4 Carlos Roman       Lavado                 19
##  5 Felipe Casas       Lavado                 19
##  6 Nathaly Ramirez    Lavado                 16
##  7 Manuel Mosquera    Infeccion/Coleccion    10
##  8 Akram Kadamani     Infeccion/Coleccion     7
##  9 Ciro Andres Murcia Lavado                  7
## 10 Paulo Cabrera      Infeccion/Coleccion     7
## # ... with 49 more rows
```

Determine el numero de cirugias al mes de cada cirujano, y el tiempo promedio en la cirugia. (Sin residente y despues hacer ejericio con residente) Ordenar por mes.

```
cirugia_fci_2018 %>%
  group_by(Cirujano, mes = month(Fecha_cirugia)) %>%
  summarize(
    t_med_cx = mean(Tiempo_quirurgico, na.rm = TRUE),
    total_cx = n()
  ) %>%
  view()
```