

R para contextos humanitarios de emergencia

Manipulación de datos I

Violeta Roizman

Abrir el archivo 03-EJ-manipulacion.Rmd

Para ir haciendo los EJ

Manipulación de datos

El paquete `dplyr` provee una enorme cantidad de funciones útiles para manipular datos



Las funciones más comunes:

- `select()`: selecciona columnas de una tabla
- `filter()` y `slice()`: selecciona (o filtra) filas de una tabla
- `mutate()`: agrega nuevas columnas a una tabla
- `arrange()`: ordena las filas según los valores de una columna
- `group_by()`: agrupa una tabla en base al valor de una o más columnas
- `summarise()`: calcula estadísticas para cada grupo de una tabla.

dplyr y tablas dinámicas

A rasgos generales, las operaciones de dplyr permiten hacer lo que se hace en tablas dinámicas (pivot tables) en Excel.

Funcion de dplyr

- `filter()`
- `group_by()`
- `select()`
- `summarise()`

Sección de tabla dinámica

- "Filtros"
- "Filas"
- "Columnas"
- "Valores"

Tu turno 1

Te dieron una tabla con datos de temperatura mínima y máxima para distintas estaciones meteorológicas de todo el país durante los 365 días de un año. Las columnas son: id_estacion, temperatura_maxima, temperatura_minima y provincia. En base a esos datos, te piden que computes la temperatura media anual de cada estación únicamente de las estaciones de Cusco.

¿En qué orden ejecutarías estos pasos para obtener el resultado deseado?

- usar `summarise()` para calcular la estadística `mean(temperatura_media)` para cada `id_estacion`
- usar `group_by()` para agrupar por la columna `id_estacion`
- usar `mutate()` para agregar una columna llamada `temperatura_media` que sea $(\text{temperatura_minima} + \text{temperatura_maxima})/2$
- usar `filter()` para seleccionar solo las filas donde la columna `provincia` es igual a "Cusco"

Los datos

Retomamos el dataset sobre decisiones de Perú frente a los pedidos de asilo

(es una traducción y simplificación del dataset de UNHCR sobre decisiones de Perú:
<https://data.humdata.org/dataset/unhcr-population-data-for-per>)

```
decisiones <- read_csv("data/decisiones_asilo_peru.csv")  
decisiones
```

```
## # A tibble: 164 x 15  
##   Anio `Codigo Pais Or~ `Codigo Pais As~ `Nombre Pais de~ `Nombre Pais As~  
##   <dbl> <chr>          <chr>          <chr>          <chr>  
## 1 2000 COL            PER            Colombia        Peru  
## 2 2000 CUB            PER            Cuba           Peru  
## 3 2001 RUS            PER            Russian Federat~ Peru  
## 4 2002 COL            PER            Colombia        Peru  
## 5 2002 CUB            PER            Cuba           Peru  
## 6 2003 ARG            PER            Argentina       Peru  
## 7 2003 COL            PER            Colombia       Peru  
## 8 2003 CUB            PER            Cuba           Peru  
## 9 2003 PSE            PER            State of Palest~ Peru  
## 10 2003 IRQ           PER            Iraq           Peru  
## # ... with 154 more rows, and 10 more variables: `Tipo de procedimiento` <chr>,
```

Seleccionando columnas: función select()

Si nos interesan menos columnas podemos usar `select()` así:

```
| select(dataset, c(columna1, columna4))
```

o por número de columnas `select(dataset, c(1, 4))`. También podemos excluir un conjunto de columnas:

```
| select(dataset, -c(columna1, columna2))
```

Supongamos que solo nos interesan: año, país de origen, decisión tomada(4)

Entonces redefinimos el dataset de la siguiente forma

```
decisiones <- select(decisiones,
                      c(Anio, `Codigo Pais Origen`, Reconocidas, Rechazadas,
                        `Proteccion Complementaria`, `Cerradas de otra forma`)
decisiones
```

```
## # A tibble: 164 x 6
##   Anio `Codigo Pais Origen` Reconocidas Rechazadas `Proteccion Complementaria` <dbl>
##   <dbl> <chr>           <dbl>        <dbl>           <dbl>
```

Seleccionando filas: función filter()

Supongamos que solo nos interesan los pedidos provenientes de Venezuela

Podemos usar la función filter mas una condición lógica

```
| filter(dataset, condicion)
```

En este caso seria:

```
decisiones_venezuela <- filter(decisiones, `Codigo Pais Origen`=="VEN")  
decisiones_venezuela
```

```
## # A tibble: 11 x 6  
##   Anio `Codigo Pais Origen` Reconocidas Rechazadas `Proteccion Comun`  
##   <dbl> <chr>           <dbl>      <dbl>           <dbl>  
## 1 2007 VEN               5          0            0  
## 2 2008 VEN               5          0            0  
## 3 2009 VEN               5          0            0  
## 4 2011 VEN               5          0            0  
## 5 2012 VEN               5          5            0  
## 6 2014 VEN               5         14            0  
## 7 2015 VEN              71         51            0  
## 8 2016 VEN              64         69            0
```

funcion filter(): condiciones

Las condiciones para `filter()` pueden expresarse en función de una columna del dataset o de un vector de longitud igual a la cantidad de filas del dataset.

Si quiero las decisiones tomadas después de 2010:

```
decisiones_despues_2010 <- filter(decisiones, Anio > 2010)
```

Si quiero cualquier decisión menos las de los años 2010 y 2015:

```
decisiones_2010_2015 <- filter(decisiones, !(Anio %in% c(2010, 2015)))
```

Si quiero solo las entradas de donde se rechazaron más que las que se aceptaron:

```
decisiones_mas_rechazos <- filter(decisiones, Rechazadas > Reconocidas)
```

Tu turno 2: verbos select y filter

1. Seleccionar solo las variables Anio, Código País Origen
2. Filtrar las entradas de los países Turquía y Argentina

Creando nuevas variables: función mutate()



Creando nuevas variables: función mutate()

- Queremos agregar a decisiones una columna con el total de pedidos de asilo correspondientes a cada fila (mas allá de la decisión tomada).
- Podemos definirla como la suma de las posibles decisiones

```
decisiones <- mutate(decisiones,  
                      total = Reconocidas +  
                            `Proteccion Complementaria` +  
                            `Cerradas de otra forma` +  
                            Rechazadas)  
#View(decisiones)
```

Tu turno 3: verbo mutate

1. Generar una nueva columna que indique para cada fila si fueron rechazados mas pedidos que los aceptados
2. Generar una nueva columna que indique para cada fila si el tipo de procedimiento es UNHCR
3. Generar una nueva columna con la cantidad de pedidos rechazados o con proteccion complementaria

Ordenando los datos según valores: función arrange()

Si quiero ordenar una tabla según los valores de una o más variables puedo

```
arrange(dataset, columna)
```

Por ejemplo, si quiero saber cuales fueron los años y países con mas pedidos de asilo

```
arrange(decisiones, desc(Reconocidas))
```

```
## # A tibble: 164 x 7
##       Anio `Codigo Pais Or~ Reconocidas Rechazadas `Proteccion Com~
##       <dbl> <chr>          <dbl>        <dbl>           <dbl>
## 1    2019 VEN            578         171             0
## 2    2018 VEN            390         191             0
## 3    2012 COL            120          0             0
## 4    2017 VEN            102         412             0
## 5    2019 CUB            79          59             0
## 6    2014 COL            71          104            0
## 7    2015 VEN            71          51             0
## 8    2018 CUB            70          96             0
## 9    2016 VEN            64          69             0
```

Combinando funciones

Queremos:

- Una tabla de las decisiones tomadas por Perú frente a pedidos de asilo, ordenada por la cantidad de total de pedidos por país cada año.

Podemos obtenerla así:

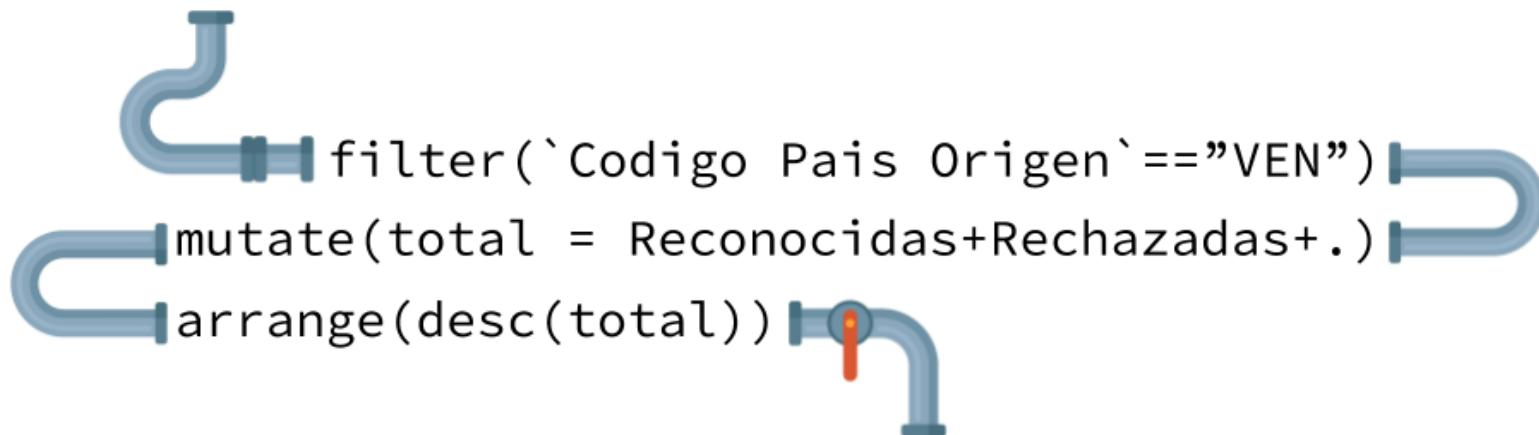
```
decisiones_venezuela <- filter(decisiones, `Codigo Pais Origen` == "VEN")
decisiones_venezuela <- mutate(decisiones_venezuela,
                                total = Reconocidas +
                                         `Proteccion Complementaria` +
                                         `Cerradas de otra forma` + Rechazadas)
decisiones_venezuela <- arrange(decisiones_venezuela, desc(total))
```

Pero estamos repitiendo demasiadas veces la variable venezuela, existe una herramienta que nos ayuda a combinar funciones encadenadas:

pipe (o tubo)

Combinando funciones

decisiones



Una tabla de las decisiones tomadas por Perú frente a pedidos de asilo de venezolan@s, ordenada por la cantidad de total de pedidos cada año.

Combinando funciones con pipe

%>%

Esto es un pipe

Si reemplazamos los tubos por esto %>% obtenemos el resultado deseado

```
decisiones %>%
  filter(`Codigo Pais Origen` == "VEN") %>%
  mutate(total = Reconocidas + `Proteccion Complementaria` + `Cerradas de o·
  arrange(desc(total))
```

```
## # A tibble: 11 x 7
##       Anio `Codigo Pais Origen` Reconocidas Rechazadas `Proteccion Complementaria` Cerradas de Oficina
##     <dbl> <chr>           <dbl>      <dbl>           <dbl>
## 1    2019 VEN            578        171             0
## 2    2018 VEN            390        191             0
## 3    2017 VEN            102        412             0
## 4    2016 VEN             64         69             0
## 5    2015 VEN             71         51             0
```

Encuesta:

https://PollEv.com/multiple_choice_polls/pb0m4Fh9kGkLRYKSmIsCq/respond

Cómo modificarías decisiones para que contenga las decisiones correspondientes a pedidos de colombian@s ordenadas por cantidad de rechazos?

a)

```
decisiones <- decisiones %>%
  filter(decisiones, `Codigo Pais Origen` == "COL") %>%
  arrange(decisiones, desc(Rechazadas))
```

b)

```
decisiones <- decisiones %>%
  filter(`Codigo Pais Origen` == "COL") %>%
  arrange(desc(Rechazadas))
```

c)

```
decisiones %>%
  filter(`Codigo Pais Origen` == "COL") %>%
  arrange(desc(Rechazadas))
```

Resumir datos: group_by y summarise

Para agrupar los datos usamos `group_by()`

Supongamos que queremos saber cuantos asilos aceptados en total corresponden a cada país de origen

decisiones

```
# A tibble: 164 x 7
  Anio `Codigo Pais Ori~ Reconocidas Rechazadas
    <dbl> <chr>          <dbl>      <dbl>
1 2000 COL             0          0
2 2000 CUB             0          10
3 2001 RUS             5          0
4 2002 COL             5          5
5 2002 CUB             5          0
6 2003 ARG             0          0
7 2003 COL            19          0
8 2003 CUB             0          0
9 2003 PSE             5          0
10 2003 IRQ            5          0
# ... with 154 more rows
```

decisiones %>%
`group_by(`Codigo Pais Origen`)`

```
# A tibble: 164 x 7
# Groups:   Codigo Pais Origen [36]
  Anio `Codigo Pais Ori~ Reconocidas Rechazadas
    <dbl> <chr>          <dbl>      <dbl>
1 2000 COL             0          0
2 2000 CUB             0          10
3 2001 RUS             5          0
4 2002 COL             5          5
5 2002 CUB             5          0
6 2003 ARG             0          0
7 2003 COL            19          0
8 2003 CUB             0          0
9 2003 PSE             5          0
10 2003 IRQ            5          0
# ... with 154 more rows
```

Los datos no se modifican, solo se agrega un indicador de que esta agrupado

Resumir datos: group_by y summarise

Una vez que están agrupados queremos elegir como resumirlo

```
decisiones %>%
  group_by(`Codigo Pais Origen`) %>%
  summarise(total_aceptado = sum(Reconocidas))

## `summarise()` ungrouping output (override with `.`groups` argument)

## # A tibble: 36 x 2
##   `Codigo Pais Origen` total_aceptado
##   <chr>                  <dbl>
## 1 ARG                      0
## 2 BGD                     36
## 3 BOL                     25
## 4 BRA                     5
## 5 CMR                     20
## 6 COD                     5
## 7 COL                   684
## 8 CUB                   434
## 9 DOM                     30
## 10 ECU                    0
## # ... with 26 more rows
```

Resumir datos: group_by y summarise

Cual es el TOP 3 de país de origen aceptado?

```
decisiones %>%
  group_by(`Codigo Pais Origen`) %>%
  summarise(total_aceptado = sum(Reconocidas)) %>%
  arrange(desc(total_aceptado))
```

```
## `summarise()` ungrouping output (override with ` `.groups` argument)
```

```
## # A tibble: 36 x 2
##   `Codigo Pais Origen` total_aceptado
##   <chr>                  <dbl>
## 1 VEN                   1235
## 2 COL                   684
## 3 CUB                   434
## 4 HTI                   81
## 5 SYR                   47
## 6 TUR                   44
## 7 BGD                   36
## 8 PSE                   35
## 9 DOM                   30
## 10 IRQ                  30
```

summarise por separado

También summarise solo

```
decisiones %>%  
  summarise(total_aceptados = sum(Reconocidas))
```

```
## # A tibble: 1 x 1  
##   total_aceptados  
##       <dbl>  
## 1      2782
```

Tu turno 4: verbos dplyr

1. Obtener una tabla con la cantidad de pedidos de asilo rechazados por Perú durante cada año
2. Ordena esta tabla de mayor a menor según la cantidad de pedidos aceptados
3. Que cantidad de pedidos fueron rechazados en total durante el período estudiado?
4. Crea una nueva variable

Resumen de esta sección:

- Seleccionamos variables con `select()`
- Filtramos filas con `filter()`
- Ordenamos tablas con `arrange()`
- Hacemos tablas resumen con `summarise()`
- Creamos nuevas variables con `mutate()`
- Hacemos operaciones por grupos con `group_by()`

Licencia y material usado

Licencia: [Creative Commons Attribution-ShareAlike 4.0 International License](#).

Este material está inspirado y utiliza explicaciones de:

- [R para Clima](#) de Paola Corrales y Elio Campitelli
- [Master the Tidyverse](#) de Garrett Grolemund