Clase 04 - Manipulación de datos con dplyr

Curso Análisis de datos con R para Biociencias

Dra. María Angélica Rueda | maria.rueda.c@pucv.cl | Pontificia Universidad Católica de Valparaíso

20 January 2022

PLAN DE LA CLASE

1.- Introducción

- ¿Para qué manipular datos?
- Librería dplyr: Tuberías.
- Librería dplyr: Comandos clave.

2). Práctica con R y Rstudio cloud.

- Realizar manipulación de datos con dplyr.
- Realizar gráficas avanzadas con ggplot2.

MANIPULACIÓN DE DATOS

¿Para qué manipular datos?

- Para dar el formato adecuado a nuestro set de datos previo al análisis estadístico.
- Para hacerlos más legibles y organizados.
- Etapa clave para una correcta visualización de datos.

Ejemplos de tareas comunes durante esta etapa:

- Filtrar datos por categorías.
- Remover datos o imputar datos faltantes.
- Agrupar datos por algún criterio.
- Seleccionar y resumir variables.
- Generar variables derivadas a partir de variables existentes.
- Calcular medidas resumen de las variables en estudio.

LIBRERÍA DPLYR: FUNCIONES CLAVE

La librería **dplyr** posee varias funciones que permiten manipular data.frames de forma ágil e intuitiva.

Funciones claves:

arrange: Permite ordenar la base de datos según una variable de forma ascendente o descendente.
select(): Permite extraer o seleccionar variables/columnas específicas de un data.frame.
mutate(): Permite calcular/generar nuevas variables "derivadas". Útil para calcular proporciones, tasas.
filter(): Para filtrar desde una tabla de datos un subconjunto de filas. Ej. solo un nivel de un factor, observaciones que cumplen algún criterio (ej. > 20).
group_by(): Permite agrupar filas con base a los niveles de alguna variable o factor.
replace_na: Permite reemplazar datos faltante (NA) por algún valor.
summarize: Permite obtener medidas resumen de las variables.

LIBRERÍA DPLYR: EL OPERADOR PIPE (TUBERÍA).

dplyr usa el operador pipe **%>%** como una tubería para enlazar un data.frame con una o más funciones.

```
x <- rnorm(5)
y <- rnorm(5)
dat <- data.frame(x,y)
dat %>% max
```

```
## [1] 1.222585
```

```
dat %>% arrange(y)
```

```
##              x           y
## 1  0.09107899 -0.12112223
## 2  0.61628125 -0.01474268
## 3  1.22258472  0.05582328
## 4  0.08305760  0.19677217
## 5 -2.13655889  1.03248295
```

ESTUDIO DE CASO: MUESTREO DE PECES

Objeto: peces

Pez	Especie	Sexo	Peso	Parásitos
1	A	Hembra	174	0
2	A	Hembra	155	2
3	A	Hembra	131	25
4	B	Macho		8
5	B	Macho	103	33
6	B	Hembra	138	15
7	C	Hembra	135	5
8	C	Macho	138	20
9	C	Hembra	135	

FUNCIÓN REPLACE_NA() CON LIBRERÍA TIDYR

```
peces <- peces %>%
  replace_na(list(Peso= 120, Parasitos= 25))
```

FUNCIÓN SELECT()

```
select(peces, Especie, Sexo)
```

```
## # A tibble: 9 x 2
##   Especie Sexo
##   <chr>   <chr>
## 1 A       Hembra
## 2 A       Hembra
## 3 A       Hembra
## 4 B       Macho
## 5 B       Macho
## 6 B       Hembra
## 7 C       Hembra
## 8 C       Macho
## 9 C       Hembra
```

FUNCIÓN SELECT() CON PIPE

```
peces %>% select(Especie, Sexo)
```

```
## # A tibble: 9 x 2
##   Especie Sexo
##   <chr>   <chr>
## 1 A       Hembra
## 2 A       Hembra
## 3 A       Hembra
## 4 B       Macho
## 5 B       Macho
## 6 B       Hembra
## 7 C       Hembra
## 8 C       Macho
## 9 C       Hembra
```

FUNCIÓN FILTER() CON PIPE

```
peces %>% filter(Sexo == "Macho")
```

```
## # A tibble: 3 x 5
##     Pez Especie Sexo   Peso Parasitos
##   <dbl> <chr>   <chr> <dbl>     <dbl>
## 1     4 B       Macho   120         8
## 2     5 B       Macho   103        33
## 3     8 C       Macho   138        20
```

MÚLTIPLES FUNCIONES Y TUBERÍAS

```
peces %>% select(Especie, Sexo, Peso) %>%
  filter(Sexo == "Macho")
```

```
## # A tibble: 3 x 3
##   Especie Sexo   Peso
##   <chr>   <chr> <dbl>
## 1 B       Macho   120
## 2 B       Macho   103
## 3 C       Macho   138
```

FUNCIÓN SUMMARIZE()

```
peces %>% select(Especie, Sexo, Peso, Parasitos) %>%
          summarize(n = n(),
          Promedio_Peso = mean(Peso),
          Maximo_Parasitos = max(Parasitos))
```

```
## # A tibble: 1 x 3
##       n Promedio_Peso Maximo_Parasitos
##   <int>          <dbl>             <dbl>
## 1     9           137.                33
```

FUNCIÓN SUMMARIZE() + GROUP_BY()

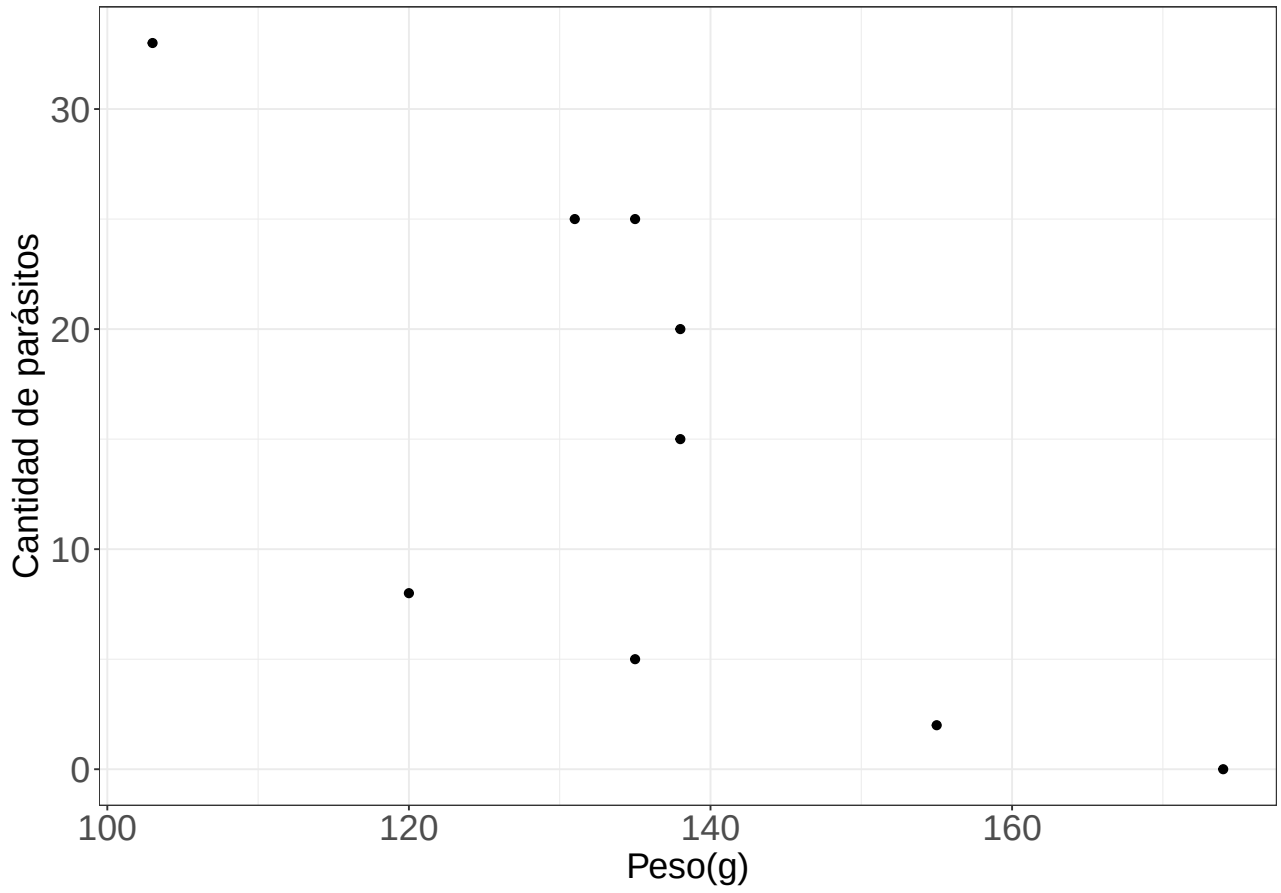
```
peces %>% group_by(Especie) %>%
          summarize(n = n(),
          Promedio_Peso = mean(Peso),
          Maximo_Parasitos = max(Parasitos))
```

```
## # A tibble: 3 x 4
##   Especie     n Promedio_Peso Maximo_Parasitos
##   <chr>   <int>          <dbl>             <dbl>
## 1 A           3           153.                25
## 2 B           3           120.                33
## 3 C           3           136                 25
```

GENERAR NUEVA BASE DE DATOS + FUNCIÓN MUTATE()

```
datos<- peces %>% select(Especie, Sexo, Peso, Parasitos)%>%
  mutate(Densidad_parasitos = Parasitos/Peso)
```

HACER PLOT CON GGPLOT2



PRÁCTICA ANÁLISIS DE DATOS

1.- Guía de trabajo Rmarkdown disponible en drive.
Clase 04

2.- La tarea se realiza en Rstudio.cloud. **Clase 04 - Manipular datos con dplyr**

RESUMEN DE LA CLASE

- Manipulamos datos con dplyr.
- Aplicamos tuberías con pipe **%>%**.
- Hicimos gráfico con ggplot2 usando la base generada con dplyr.
- Comunicamos un análisis exploratorio de datos de forma efectiva.