

# CLASE 04 - MANIPULAR Y TRANSFORMAR DATOS.

Curso Introducción al Análisis de datos con R para la Acuicultura.

Dr. José Gallardo Matus

Pontificia Universidad Católica de Valparaíso

11 July 2022

## 1.- Introducción

- ▶ ¿Para qué manipular datos?
- ▶ Diferencia entre Tidy and messy data.
- ▶ Paquete tidyr.
- ▶ Operador pipe (Tuberías).
- ▶ Paquete dplyr.

## 2). Práctica con R y Rstudio cloud.

- ▶ Realizar manipulación de datos con tidyr y dplyr.
- ▶ Realizar gráficas avanzadas con ggplot2.

# MANIPULACIÓN DE DATOS

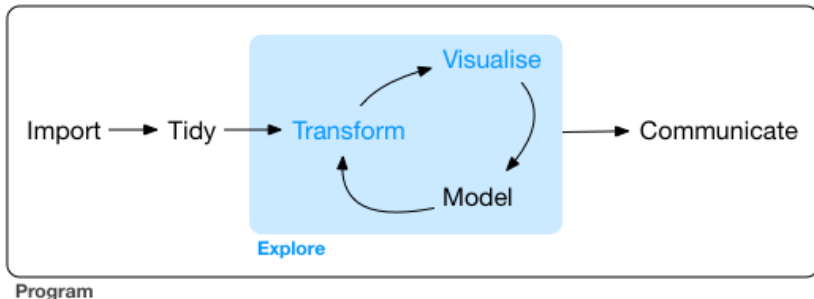
## ¿Para qué manipular datos?

- ▶ Para hacer datos más legibles y organizados.
- ▶ Para dar formato adecuado previo a visualización y análisis estadístico.

## Ejemplos de tareas comunes durante esta etapa:

- ▶ Filtrar datos por categorías.
- ▶ Remover o imputar datos faltantes.
- ▶ Agrupar datos por algún criterio.
- ▶ Seleccionar y calcular estadísticos.
- ▶ Generar variables derivadas a partir de variables existentes.
- ▶ Transformar variables.

# ETAPAS DEL ANÁLISIS DE DATOS



# PAQUETES CLAVE

Importar

transformar

Visualizar



## **Tidy data (datos ordenados)**

- ▶ Cada columna es una variable.
- ▶ Cada fila es una observación.
- ▶ Cada celda es un simple dato o valor.

## **Messy data (desordenados)**

- ▶ Cualquier conjunto de datos que no cumple alguno de estos criterios.

# EJEMPLO DATOS MESSY

**¿Por qué son messy?**

Variable	Replica	Especie A	Especie B	Especie C
peso	1	174	NA	135
peso	2	155	103	138
peso	3	131	138	135
parásitos	1	25	8	5
parásitos	2	12	3	8
parásitos	3	4	11	NA

# EJEMPLO DATOS TIDY

**¿Por qué son tidy?**

Pez	Especie	Sexo	Peso	Parásitos
1	A	Hembra	174	25
2	A	Hembra	155	12
3	A	Hembra	131	4
4	B	Macho	NA	8
5	B	Macho	103	3
6	B	Hembra	138	11
7	C	Hembra	135	5
8	C	Macho	138	8
9	C	Hembra	135	NA



# EL OPERADOR PIPE: %>%.

En programación **pipe** es una técnica que permite pasar información de un proceso o programa a otro por etapas.

Evita pipe cuando: a) Deseas manipular varios objetos a la vez. b) Un paso intermedio genera un objeto que luego deseas analizar separadamente.

datos  funcion(...)

datos  funcion(1)  Funcion(2)

# PAQUETE DPLYR: FUNCIONES BÁSICAS

**select()**: Permite extraer o seleccionar variables/columnas específicas de un data.frame.

**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.  $> 20$ ).

**mutate()**: Permite calcular/generar nuevas variables “derivadas”. Útil para calcular proporciones, tasas.

**arrange()**: Permite ordenar la base de datos según una variable de forma ascendente o descendente.

# PAQUETE DPLYR: SELECT()

Datos %>% select(Año, TSM)

Bahía	Año	TSM
Valparaíso	2019	12
Concepción	2019	10
Valparaíso	2020	13
Concepción	2020	11
Valparaíso	2021	14
Concepción	2021	12



Año	TSM
2019	12
2019	10
2020	13
2020	11
2021	14
2021	12

# PAQUETE DPLYR: FILTER()

Datos %>% filter(Bahía="Valparaíso")

Bahía	Año	TSM
Valparaíso	2019	12
Concepción	2019	10
Valparaíso	2020	13
Concepción	2020	11
Valparaíso	2021	14
Concepción	2021	12



Bahía	Año	TSM
Valparaíso	2019	12
Valparaíso	2020	13
Valparaíso	2021	14

# PAQUETE DPLYR: MUTATE()

Datos %>% mutate(Fahrenheit=(TSM\*1.8)+32)

Bahía	Año	TSM
Valparaíso	2019	12
Concepción	2019	10
Valparaíso	2020	13
Concepción	2020	11
Valparaíso	2021	14
Concepción	2021	12



Bahía	Año	TSM	Fahrenheit
Valparaíso	2019	12	53,6
Concepción	2019	10	50,0
Valparaíso	2020	13	55,4
Concepción	2020	11	51,8
Valparaíso	2021	14	57,2
Concepción	2021	12	53,6

# PAQUETE DPLYR: GROUP\_BY + SUMMARIZE()

```
Datos %>% grup_by(Bahía) %>%  
  summarize(n=n(),  
            promedio=mean(TSM))
```

Bahía	Año	TSM
Valparaíso	2019	12
Concepción	2019	10
Valparaíso	2020	13
Concepción	2020	11
Valparaíso	2021	14
Concepción	2021	12



Bahía	Año	TSM
Valparaíso	2019	12
Valparaíso	2020	13
Valparaíso	2021	14
Concepción	2019	10
Concepción	2020	11
Concepción	2021	12



Bahía	n	Promedio
Valparaíso	3	13
Concepción	3	11

# RESUMEN DE LA CLASE

- ▶ Diferenciamos datos ordenados (Tidy) y desordenados (Messy).
- ▶ Manipulamos datos con tidyr y dplyr.
- ▶ Utilizamos tuberías o pipe `%>%`.
- ▶ Hicimos gráfico ggplot2 usando datos transformados y variables derivadas.