Tablas para Clase 7

Diplomado en Análisis de datos con R para la acuicultura

true

06/05/2021

Introducción

Manipulación de bases de datos. La manipulación de datos es una de las actividades que más tiempo demanda ya que implica tanto dar el formato adecuado como aplicar transformaciones sobre las variables. Este proceso se lleva a cabo con el fin de generar un conjunto de datos acorde para su *posterior* análisis.

En síntesis la manipulación de datos se resume en cuatro operaciones:

1.- Extraer subconjuntos de datos

Ya sea por filas (por algún criterio) o por columnas (para realizar análisis especificos, donde no es necesario considerar todas las variables presentes en la base de datos original).

2.- Transformarlos

Realizar operaciones aritméticas o lógicas sobre los datos. En general cuando una **variable es continua** realizamos operaciones ***aritméticas***, cuando es **categórica**, ***lógicas***.

3.- Obtener medidas resumen del conjunto de datos (ya sea por variable/s) o por grupos (criterio de agrupamiento por fila)

Se puede resumir la información por variable calculando: valores promedio, desviaciones estándar, valores mínimos o máximos.

4.- Compactar con otros conjuntos de datos

Es posible que en tiempos posteriores surjan nuevas bases de datos que pueden enriquecer nuestros análisis, es muy útil combinar/compactar dicha información de forma homogénea en una sola base de datos.

Visualización de datos. Es importante en la etapa de análisis exploratorio de datos, hacer representaciones gráficas porque permite:

- Detectar posibles valores atípicos "outliers".
- Identificar la posible distribución subyacente de la variable respuesta **(Y)**.
- Analizar posibles relaciones/asociaciones entre la variable respuesta **(Y)** y la variable explicativa **(X)**.
- Sintetizar/mostrar la información proveniente de los datos, en forma amena y resumida.

Nota: La librería **ggplot2** contiene la función ***ggplot()*** con la que se generan gráficos tanto visualmente informativos como de agradable aspecto. Los objectos (dataframes) generados por *tidyverse* se pueden graficar con la función **ggplot()**.

Objetivos de aprendizaje Los objetivos de aprendizaje de esta guía son:

1. Manipular y generar nuevas bases de datos y gráficos a partir de comandos de Tidyverse.

2. Elaborar un reporte dinámico en formato pdf.

Comandos para manipular datos con Tidiverse

Comandos	**Función**
%>%	Este comando se llama ***pipe*** "tubería", es una herramienta para la composición de funciones, lo que facilita la resolución de problemas grandes partiéndolos en pedazos pequeños.
filter()	Filtra los datos por algún criterio establecido por el usuario.
arrange()	Ordena la base de datos según una variable (de forma ascendente).
arrange(desc())	Ordena la base de datos según una variable (de forma descendente).
mutate()	Cambia el contenido de una variable o genera variables derivadas a partir de variables existentes en el conjunto de datos.
summarize()	Resume la información de la/s variable/s en un solo dato. Algunas funciones que se pueden usar para resumir la información son: ***mean*** *para calcular el promedio por variable*, ***sum*** *para sumar las observaciones de una variable*, ***median*** *para calcular la mediana por variable*,***min*** *para encontrar el valor mínimo por variable* y ***max*** *para encontrar el valor máximo por variable.*
group_by	Agrupa las observaciones (filas de la base de datos) por algún criterio establecido por el usuario.

Comandos para realizar gráficos con ggplot2 en Tidiverse

Comandos	**Función**
ggplot():	Permite hacer gráficos en el formato de ggplot2.
geom_point():	Argumento que se adiciona a la función ggplot() para generar diagramas de dispersión.
geom_line():	Argumento que se adiciona a la función ggplot() para generar diagramas de líneas.
geom_col():	Argumento que se adiciona a la función ggplot() para generar diagramas de barras.
geom_histogram():	Argumento que se adiciona a la función ggplot() para generar histogramas.
geom_boxplot():	Argumento que se adiciona a la función ggplot() para generar gráficos de cajas y bigotes.
expand_limits(y=0):	Argumento que se adiciona a la función ggplot() para que el eje de la **Y** empiece en cero.
scale_x_log10():	Argumento que se adiciona a la función ggplot() para que el eje de la **X** este en la escala de logaritmo en base 10.

Comandos	Función
facet_wrap(~ variable):	Argumento que se adiciona a la función ggplot() para que realice tantos gráficos según los niveles que tenga la variable.

Ejercicios

Ejercicio 1. Elaborar archivo Rmarkdown

Elabore un archivo o file con extensión **.Rmd** y configurelo para exportar el resultado como un documento dinámico **pdf**. Utilice el siguiente ejemplo para completar la información de **metadatos**: Título: Reporte Manipulación de base de datos, nombre del autor: Su nombre.

Luego, guarde inmediatamente su script como **script_7_nombre_apellido.Rmd**. Al finalizar la actividad deberá exportar y almacenar este ***script*** en su carpeta drive de tareas.

Ejercicio 2. Configuración del reporte

En el primer bloque de códigos o **chunk** configure los comandos de la siguiente manera ***knitr::opts_chunk$set(echo = TRUE)*** y cargue las librerías **readxl**, **stats**,**dplyr**, **tidyr** y **ggplot2** usando la función ***library()***.

```
knitr::opts_chunk$set(echo = TRUE)
library(readxl)
library(stats)
library(dplyr)
library(tidyr)
library(ggplot2)
```

Ejercicio 3. Importar la base de Datos1

Cree un objeto llamado **datos1** e importe el set de datos **Datos1** usando la función ***read_excel()*** de la librería **readxl**. Explore el set de datos usando las funciones **head()** y **str()**.

```
datos1 <- read_excel("Datos1.xlsx")
head(datos1)
```

```
## # A tibble: 6 x 5
##   Especies              Year ProdNac DispOvas Importacion
##   <chr>                <dbl>   <dbl>    <dbl>       <dbl>
## 1 Salmón del Atlántico  2016    361.     478.        3.32
## 2 Salmón Plateado       2016    125.     187.        0
## 3 Trucha Arcoiris       2016    101.     187.        0
## 4 Salmón del Atlántico  2017    427.     587.        7.54
## 5 Salmón Plateado       2017    109.     207.        0
## 6 Trucha Arcoiris       2017    106.     164.        0
```

```
str(datos1)
```

```
## tibble[,5] [6 x 5] (S3: tbl_df/tbl/data.frame)
##  $ Especies   : chr [1:6] "Salmón del Atlántico" "Salmón Plateado" "Trucha Arcoiris" "Salmón del Atl
##  $ Year       : num [1:6] 2016 2016 2016 2017 2017 ...
##  $ ProdNac    : num [1:6] 361 125 101 427 109 ...
##  $ DispOvas   : num [1:6] 478 187 187 587 207 ...
##  $ Importacion: num [1:6] 3.32 0 0 7.54 0 ...
```

Ejercicio 4. Filtrar información de la base de datos por algún criterio

Del conjunto de datos **datos1** seleccione solo la información del **Year 2017** usando la función ***filter()***, recuerde usar el simbolo %>% para que pueda realizar el filtrado.

```
datos1 %>%
  filter(Year == 2017)
```

```
## # A tibble: 3 x 5
##   Especies              Year ProdNac DispOvas Importacion
##   <chr>                <dbl>   <dbl>    <dbl>       <dbl>
## 1 Salmón del Atlántico  2017    427.     587.        7.54
## 2 Salmón Plateado       2017    109.     207.        0
## 3 Trucha Arcoiris       2017    106.     164.        0
```

Ejercicio 5. Filtrar información de la base de datos por año y especie

Del set de datos **datos1** seleccione solo la información del **Year 2016** y **Especie Salmón del Atlántico** usando la función ***filter()***.

```
datos1 %>%
  filter(Year == 2016, Especies == "Salmón del Atlántico")
```

```
## # A tibble: 1 x 5
##   Especies              Year ProdNac DispOvas Importacion
##   <chr>                <dbl>   <dbl>    <dbl>       <dbl>
## 1 Salmón del Atlántico  2016    361.     478.        3.32
```

Ejercicio 6. Calcular medidas resumen

Del conjunto de datos **datos1** seleccione solo la información de la **Trucha Arcoiris** para los dos años evaluados, usando la función ***filter()***. Halle la media y el valor mínimo de la variable **ProdNac**, genere los objetos ***Media_ProdNac*** y ***Min_ProdNac***, respectivamente. Use la función ***summarize()***.

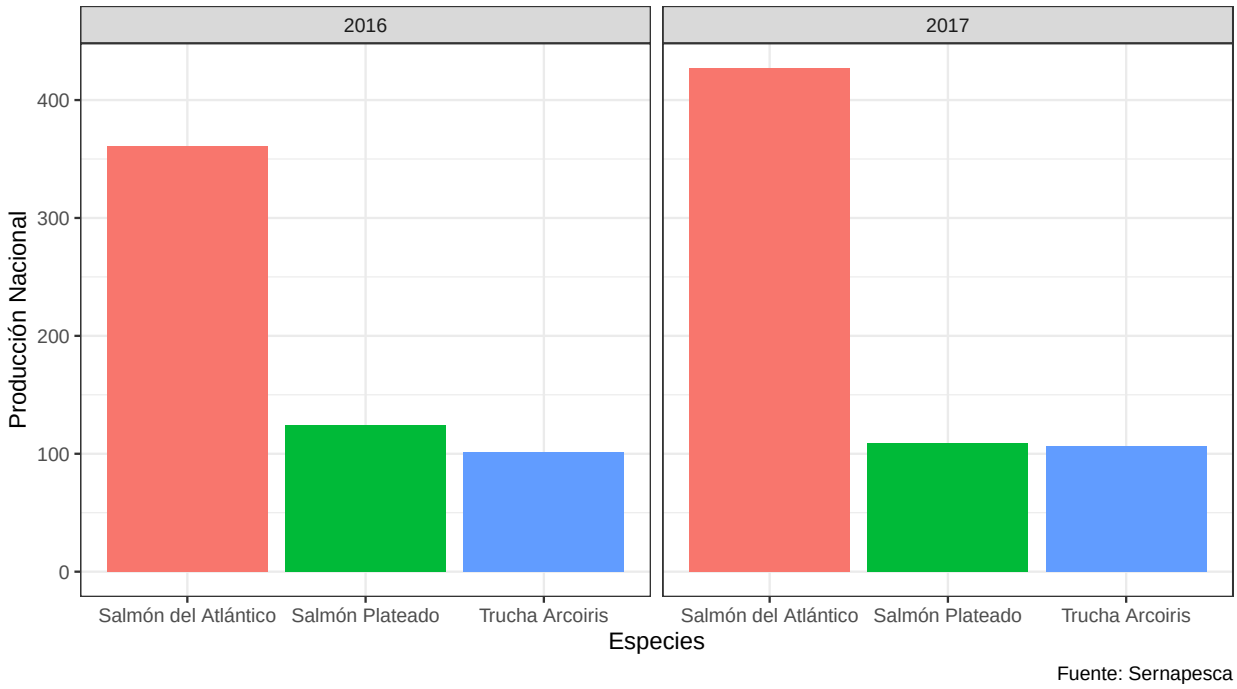
```
datos1 %>%
  filter(Especies == "Trucha Arcoiris")%>%
  summarize(Media_ProdNac= mean(ProdNac),Min_ProdNac = min(ProdNac))
```

```
## # A tibble: 1 x 2
##   Media_ProdNac Min_ProdNac
##           <dbl>       <dbl>
## 1          104.        101.
```

Ejercicio 7. Graficar diagrama de barras con ggplot()

Con conjunto de datos **datos1** realice un gráfico de barras donde la variable X=**Especies** y en el eje Y =**ProdNac**, colore por Especies.Divida los diagramas de barras por Year, use el comando facet_wrap(~ Year).Además, use el comando expand_limits(y=0) para que el eje Y comience en cero.

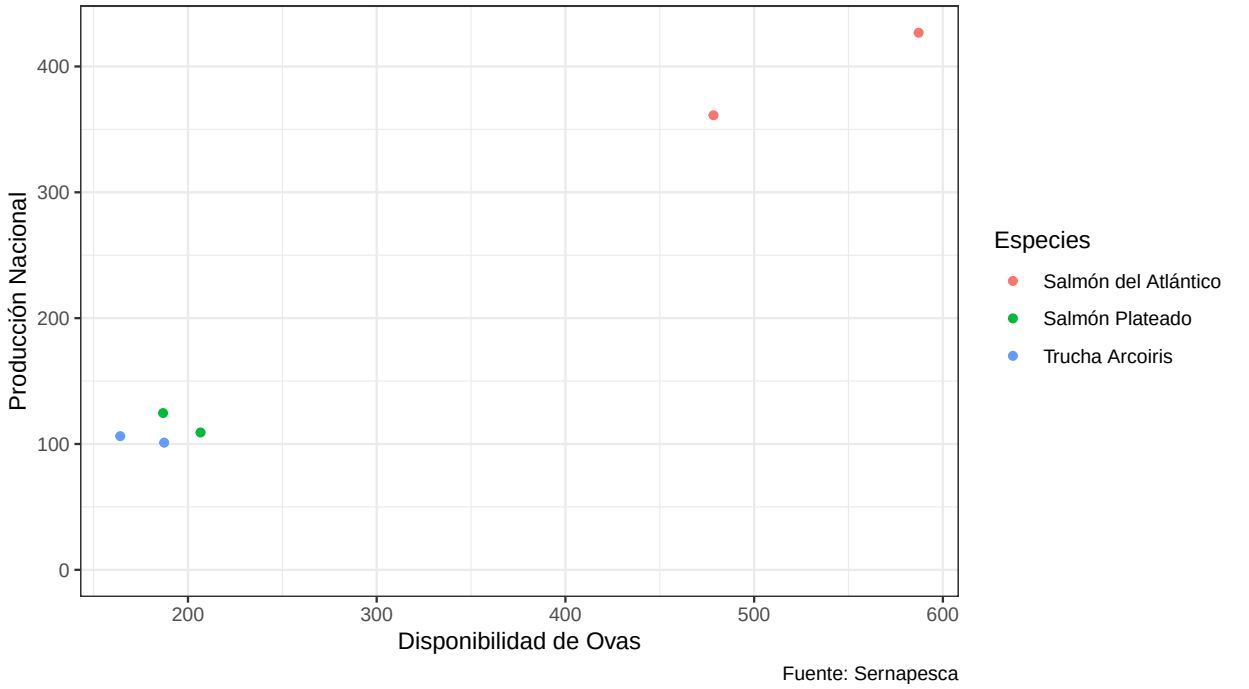
```
ggplot(datos1, aes(x= Especies, y= ProdNac, fill=Especies))+
  geom_col()+
 labs(y= "Producción Nacional", x= "Especies",
      caption = "Fuente: Sernapesca")+
  expand_limits(y=0)+
  facet_wrap(~ Year)+
  guides(fill=FALSE, color=FALSE)+
  theme_bw()
```



Ejercicio 8. Graficar scatter plot con ggplot ()

Con el conjunto de datos **datos1** realice un gráfico de dispersión donde la variable X=**DispOvas** y la variable Y =**ProdNac**. Colore por especies.Además, use el comando expand_limits(y=0) para que el eje Y comience en cero.

```
ggplot(datos1, aes(x= DispOvas, y= ProdNac, color=Especies))+
  geom_point()+
 labs(y= "Producción Nacional", x= "Disponibilidad de Ovas",
      caption = "Fuente: Sernapesca")+
  expand_limits(y=0)+
  theme_bw()
```



Ejercicio 9. Importar la base de Datos2

Cree un objeto llamado **datos2** e importe el set de datos **Datos2** usando la función ***read_excel()*** de la librería **readxl**. Explore el set de datos usando las funciones **head()** y **str()**.

```
datos2 <- read_excel("Datos2.xlsx")
head(datos2)
```

```
## # A tibble: 6 x 4
##   Mes     Region_Lagos Region_Aysen Region_Magallanes
##   <chr>          <dbl>        <dbl>              <dbl>
## 1 Enero             NA            1                 NA
## 2 Febrero           NA            1                  1
## 3 Marzo              2            2                  1
## 4 Abril             NA            2                  4
## 5 Mayo               1            3                 NA
## 6 Junio             NA            6                  2
```

```
str(datos2)
```

```
## tibble[,4] [12 x 4] (S3: tbl_df/tbl/data.frame)
##  $ Mes              : chr [1:12] "Enero" "Febrero" "Marzo" "Abril" ...
##  $ Region_Lagos     : num [1:12] NA NA 2 NA 1 NA 4 NA NA 1 ...
##  $ Region_Aysen     : num [1:12] 1 1 2 2 3 6 NA NA NA 1 ...
##  $ Region_Magallanes: num [1:12] NA 1 1 4 NA 2 NA 1 NA 1 ...
```

Ejercicio 10. Reemplazar valores faltantes

Reemplace los valores faltantes **NA** por **0** de las variables Region_Lagos, Region_Aysen y Region_Magallanes de la base de datos **datos2**. Use la función ***replace_na()*** y como argumento de

esta función haga una **lista** con estas variables e igualelas a cero.

```
datos2 %>%
replace_na(list(Region_Lagos = 0, Region_Aysen = 0, Region_Magallanes = 0)) %>%
  mutate(Region_Lagos =Region_Lagos, Region_Aysen = Region_Aysen,
         Region_Magallanes = Region_Magallanes)
```

```
## # A tibble: 12 x 4
##    Mes        Region_Lagos Region_Aysen Region_Magallanes
##    <chr>             <dbl>        <dbl>             <dbl>
##  1 Enero                 0            1                 0
##  2 Febrero               0            1                 1
##  3 Marzo                 2            2                 1
##  4 Abril                 0            2                 4
##  5 Mayo                  1            3                 0
##  6 Junio                 0            6                 2
##  7 Julio                 4            0                 0
##  8 Agosto                0            0                 1
##  9 Septiembre            0            0                 0
## 10 Octubre               1            1                 1
## 11 Noviembre             1            0                 1
## 12 Diciembre             0            4                 0
```

Ejercicio 11. Generar variables derivadas

Genere una nueva variable llamada **NTotal** a partir de la suma de los datos de las variables Region_Lagos, Region_Aysen y Region_Magallanes de la base de datos **datos2** generada en el **ejercicio 10**. También, genere una variable llamada **Per_NTotal** que es el porcentaje mensual de casos de ISA confirmados. Use la función ***mutate***.

```
datos2 %>%
replace_na(list(Region_Lagos = 0, Region_Aysen = 0, Region_Magallanes = 0)) %>%
  mutate(Region_Lagos =Region_Lagos, Region_Aysen = Region_Aysen,
         Region_Magallanes = Region_Magallanes,
         NTotal =    Region_Lagos+Region_Aysen+Region_Magallanes,
         Per_NTotal = (NTotal/sum(NTotal))*100)
```

```
## # A tibble: 12 x 6
##    Mes        Region_Lagos Region_Aysen Region_Magallanes NTotal Per_NTotal
##    <chr>             <dbl>        <dbl>             <dbl>  <dbl>      <dbl>
##  1 Enero                 0            1                 0      1        2.5
##  2 Febrero               0            1                 1      2        5
##  3 Marzo                 2            2                 1      5       12.5
##  4 Abril                 0            2                 4      6       15
##  5 Mayo                  1            3                 0      4       10
##  6 Junio                 0            6                 2      8       20
##  7 Julio                 4            0                 0      4       10
##  8 Agosto                0            0                 1      1        2.5
##  9 Septiembre            0            0                 0      0        0
## 10 Octubre               1            1                 1      3        7.5
## 11 Noviembre             1            0                 1      2        5
## 12 Diciembre             0            4                 0      4       10
```

Ejercicio 12. Realizar diagrama de barras

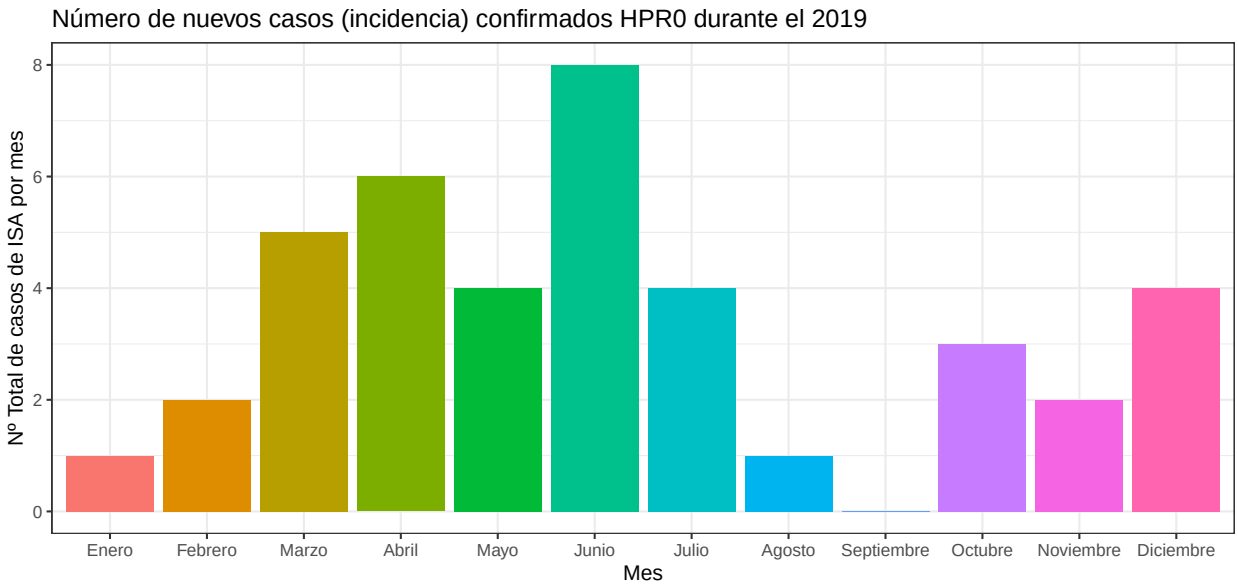
Cargue la base de datos **datos2_new.txt**, estos datos son los que se generaron con la información del ejercicio anterior. Use la función ***ggplot()*** para realizar un diagrama de barras con la variables **X** = Mes, e **Y** = NTotal. Para hacer el gráfico de barras primero, debes volver la variable meses **factor** y sus levels deben ser cada uno de los meses en orden, esto es para que en el eje de la **X** queden organizadas las etiquetas según los meses del año. Colore las barras por mes.Además, use el comando expand_limits(y=0) para que el eje Y comience en cero.

```
datos2_new<-read.table("datos2_new.txt", header = TRUE, sep = " ")

datos2_new[,1] <- factor(datos2_new$Mes,
                         levels= c("Enero","Febrero","Marzo","Abril","Mayo",
                                   "Junio","Julio","Agosto","Septiembre",
datos2_new
```

```
##           Mes Region_Lagos Region_Aysen Region_Magallanes NTotal Per_NTotal
## 1       Enero            0            1                  0      1        2.5
## 2     Febrero            0            1                  1      2        5.0
## 3       Marzo            2            2                  1      5       12.5
## 4       Abril            0            2                  4      6       15.0
## 5        Mayo            1            3                  0      4       10.0
## 6       Junio            0            6                  2      8       20.0
## 7       Julio            4            0                  0      4       10.0
## 8      Agosto            0            0                  1      1        2.5
## 9  Septiembre            0            0                  0      0        0.0
## 10    Octubre            1            1                  1      3        7.5
## 11  Noviembre            1            0                  1      2        5.0
## 12  Diciembre            0            4                  0      4       10.0
```

```
ggplot(datos2_new, aes(x=Mes, y= NTotal, fill= Mes))+
  geom_col() +
  expand_limits(y=0)+
  labs(title= "Número de nuevos casos (incidencia) confirmados HPR0 durante el 2019",
       y= "Nº Total de casos de ISA por mes", x= "Mes",
       caption = "Fuente: Sernapesca")+
  guides(fill=FALSE, color=FALSE)+
  theme_bw()
```



Fuente: Sernapesca