

# Capacitación en R y herramientas de productividad - Nivel básico

## Solución Tarea 3

- La **Encuesta Mensual de Alojamiento Turístico (EMAT)** tiene como objetivo general estimar mensualmente la evolución de la actividad económica desarrollada por los establecimientos de alojamiento turístico (EAT); con representación a nivel nacional, regional y por destino turístico.
- Para esta tarea usaremos la base de esta encuesta, cuyas variables son las siguientes:
  - **mes**: mes de funcionamiento del establecimiento.
  - **año**: año de funcionamiento del establecimiento.
  - **fecha**: fecha de funcionamiento del establecimiento.
  - **clase**: establecimiento clasificado como Hotel (hotel y apart-hotel), y Otros (hostales, hosterías, residenciales, cabañas y similares).
  - **region**: nombre de la región.
  - **total\_llegadas\_exp**: número total de pasajeros con residencia en Chile y en el extranjero que llegan a establecimientos de alojamiento turístico (expandido).
  - **total\_pernoctaciones\_exp**: número total de noches que los pasajeros con residencia en Chile y en el extranjero alojan en el establecimiento de alojamiento turístico (expandido).
- El plazo para recibir retroalimentación de esta tarea es de dos semanas.

### Ejercicio 1

Importe el archivo excel “emat.xlsx” de la carpeta data y asígnelo como “emat”.

```
library(readxl)

emat <- read_excel("emat.xlsx")
```

## Ejercicio 2

Agrupe el dataframe “emat” por “anio” y “region”, y calcule la suma del total de pernoctaciones expandidas utilizando el comando `summarise(suma_pernoctaciones = sum(total_pernoctaciones_exp))`. Utilice `pivot_wider()` para transformar el dataframe a formato ancho, usando como nombres de las columnas el “anio” y los valores “suma\_pernoctaciones”. Utilice el argumento `names_prefix = "pernoctaciones_"` para asignarle un prefijo al nombre de las columnas. Almacene este dataframe como “pernoctaciones\_wide”.

```
library(dplyr)
library(tidyr)

pernoctaciones_wide <- emat %>%
  group_by(anio, region) %>%
  summarise(suma_pernoctaciones = sum(total_pernoctaciones_exp)) %>%
  pivot_wider(names_from = anio,
              values_from = suma_pernoctaciones,
              names_prefix = "pernoctaciones_")

pernoctaciones_wide
```

```
## # A tibble: 16 x 3
##       region                pernoctaciones_2017 pernoctaciones_2018
##   <chr>                  <dbl>                <dbl>
## 1 Antofagasta            1732320.            1490444.
## 2 Arica y Parinacota     466405.             351435
## 3 Atacama                593675.            461032.
## 4 Aysén del Gral. Carlos Ibáñez del Ca~ 428741.            335709.
## 5 Biobío                1514068.           1251427.
## 6 Coquimbo              1516372.           1221913.
## 7 La Araucanía          1487893.           1177242.
## 8 Libertador Gral. Bernardo O'Higgins   688956.            643375.
## 9 Los Lagos             2326470.           1845937.
## 10 Los Ríos              749480.            633486.
## 11 Magallanes y de la Antártica Chilena  962289.            740349.
## 12 Maule                 732138.            571198.
## 13 Metropolitana de Santiago 7755130.           6360489.
## 14 Tarapacá              969126.            709404.
## 15 Valparaíso           3211664.           2497288.
## 16 Ñuble                  NA                80577.
```

## Ejercicio 3

Agrupe el dataframe “emat” por “anio” y “region”, y calcule la suma del total de llegadas expandidas utilizando el comando `summarise(suma_llegadas = sum(total_llegadas_exp))`. Utilice `pivot_wider()` para transformar el dataframe a formato ancho, usando como nombres de las columnas el “anio” y los valores “suma\_llegadas”. Utilice el argumento `names_prefix =`

"llegadas\_" para asignarle un prefijo al nombre de las columnas. Almacene este dataframe como "llegadas\_wide".

```
llegadas_wide <- emat %>%
  group_by(anio, region) %>%
  summarise(suma_llegadas = sum(total_llegadas_exp, na.rm = TRUE)) %>%
  pivot_wider(names_from = anio,
              values_from = suma_llegadas,
              names_prefix = "llegadas_")

llegadas_wide
```

```
## # A tibble: 16 x 3
##   region                llegadas_2017 llegadas_2018
##   <chr>                  <dbl>         <dbl>
## 1 Antofagasta           830601.        672546.
## 2 Arica y Parinacota    215125.        170620.
## 3 Atacama               312959.        244462.
## 4 Aysén del Gral. Carlos Ibáñez del Campo 223437.        174001.
## 5 Biobío                850786.        705229.
## 6 Coquimbo              689974.        552887.
## 7 La Araucanía          688321.        532178.
## 8 Libertador Gral. Bernardo O'Higgins    324763.        281674.
## 9 Los Lagos             1261881.        968149.
## 10 Los Ríos              387748.        325137.
## 11 Magallanes y de la Antártica Chilena    549918.        419988.
## 12 Maule                 405381.        319520.
## 13 Metropolitana de Santiago    3486712.       2890448.
## 14 Tarapacá              452883.        315212.
## 15 Valparaíso           1648604.       1297277.
## 16 Ñuble                  NA             41848.
```

## Ejercicio 4

Haga un `left_join()` usando "pernoctaciones\_wide" y "llegadas\_wide" usando como variable llave "region".

```
pernoctaciones_wide %>%
  left_join(llegadas_wide, by = "region")
```

```
## # A tibble: 16 x 5
##   region    pernoctaciones_2017 pernoctaciones_2018 llegadas_2017 llegadas_2018
##   <chr>          <dbl>          <dbl>          <dbl>          <dbl>
## 1 Antofaga~    1732320.        1490444.        830601.        672546.
## 2 Arica y ~    466405.         351435         215125.        170620.
## 3 Atacama      593675.         461032.        312959.        244462.
```

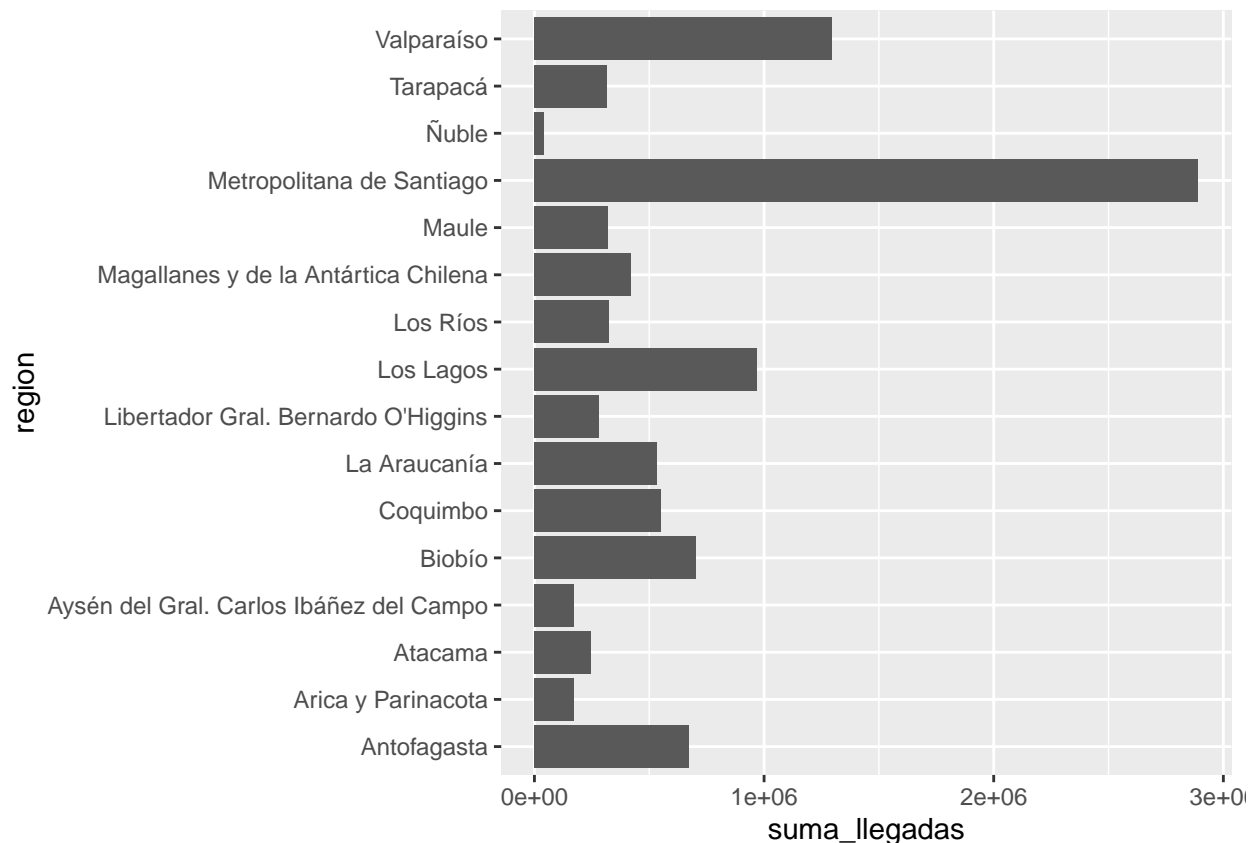
## 4 Aysén de~	428741.	335709.	223437.	174001.
## 5 Biobío	1514068.	1251427.	850786.	705229.
## 6 Coquimbo	1516372.	1221913.	689974.	552887.
## 7 La Arauc~	1487893.	1177242.	688321.	532178.
## 8 Libertad~	688956.	643375.	324763.	281674.
## 9 Los Lagos	2326470.	1845937.	1261881.	968149.
## 10 Los Ríos	749480.	633486.	387748.	325137.
## 11 Magallan~	962289.	740349.	549918.	419988.
## 12 Maule	732138.	571198.	405381.	319520.
## 13 Metropol~	7755130.	6360489.	3486712.	2890448.
## 14 Tarapacá	969126.	709404.	452883.	315212.
## 15 Valparaí~	3211664.	2497288.	1648604.	1297277.
## 16 Ñuble	NA	80577.	NA	41848.

## Ejercicio 5

- Filtre el dataframe “emat” donde el año sea igual al 2018.
- Luego, agrupe el dataframe por la variable “region” y calcule la suma de llegadas expandidas usando el comando `summarise(suma_llegadas = sum(total_llegadas_exp))`.
- Finalmente, genere un gráfico de barras mapeando en el eje x la suma de llegadas y en el eje y las regiones.

```
library(ggplot2)

emat %>%
  filter(anio == 2018) %>%
  group_by(region) %>%
  summarise(suma_llegadas = sum(total_llegadas_exp)) %>%
  ggplot() +
  aes(x = suma_llegadas, y = region) +
  geom_bar(stat = "identity")
```



## Ejercicio 6

- Utilizando el dataframe “emat” y la función `mutate()` convierta la variable “anio” a tipo factor, usando el comando `mutate(anio = as.factor(anio))`.
- Luego filtre por las filas donde la clase sea distinto a NA, usando el comando `filter(!is.na(clase))`.
- Agrupe por “clase” y “anio”, y usando `summarise()` calcule la suma del total de pernотaciones expandido. Llame a esta variable “suma\_pernотaciones”.
- Genere un gráfico de barras mapeando en el eje x “clase” y en el eje y “suma\_pernотaciones”.
- El color de cada barra debe corresponder al año y las barras deben quedar una al lado de la otra.
- Agregue títulos a los ejes y al gráfico.

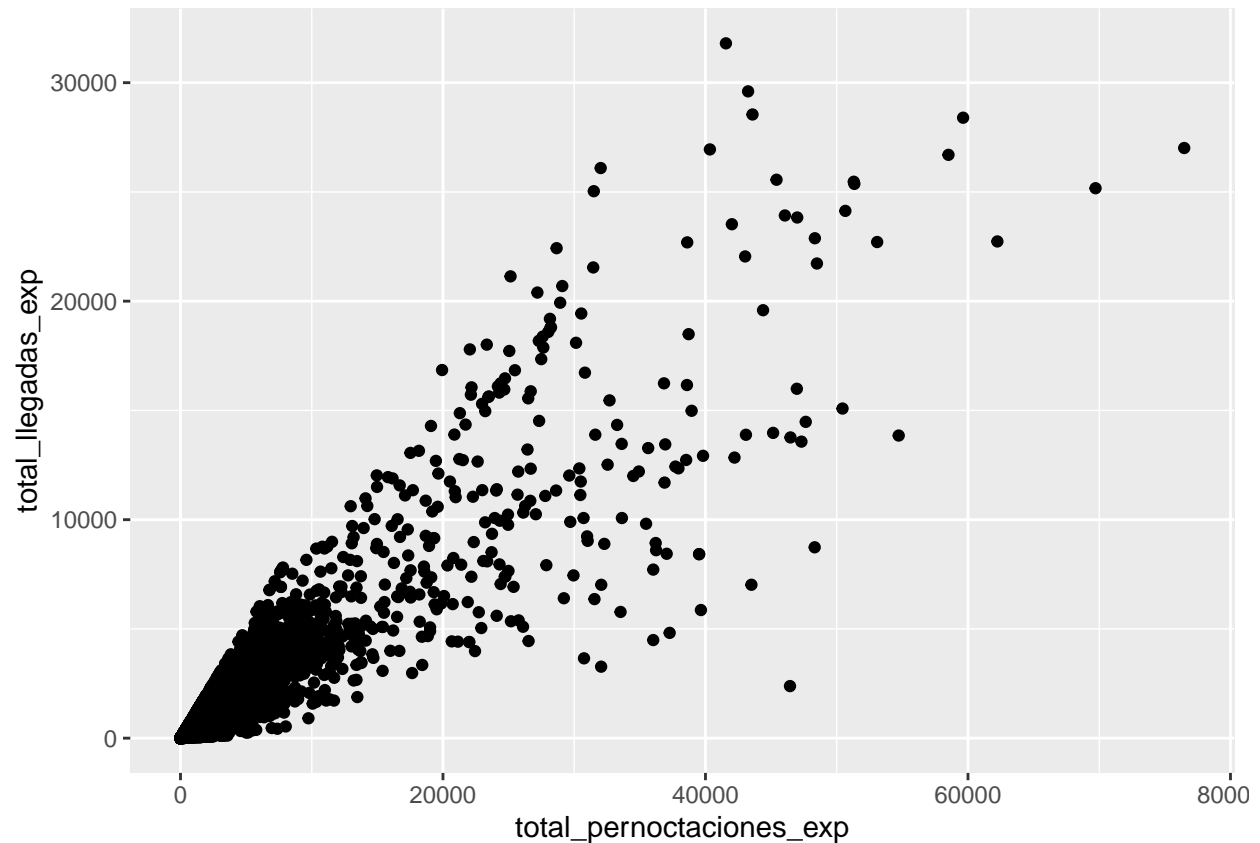
```
emat %>%
  mutate(anio = as.factor(anio)) %>%
  filter(!is.na(clase)) %>%
  group_by(clase, anio) %>%
  summarise(suma_pernотaciones = sum(total_pernотaciones_exp)) %>%
  ggplot() +
  aes(x = clase, y = suma_pernотaciones, fill = anio) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(x = "Clase", y = "Total pernотaciones", title = "Total pernотaciones por clase y año")
```



## Ejercicio 7

- Filtre el dataframe “emat” donde el año sea igual al 2017.
- Genere un gráfico de puntos o scatterplot mapeando en el eje x “total\_pernoctaciones\_exp” y en el eje y “total\_llegadas\_exp”.

```
emat %>%  
  filter(año == 2017) %>%  
  ggplot() +  
  aes(x = total_pernoctaciones_exp, y = total_llegadas_exp) +  
  geom_point()
```



## Ejercicio 8

- Filtre el dataframe “emat” por las filas donde la clase sea distinto a NA, usando el comando `filter(!is.na(clase))`.
- Agrupe por “fecha” y “clase”, y calcule la suma de “total\_llegadas\_exp” usando `summarise()`. Llame a esta variable “suma\_llegadas”.
- Genere un gráfico de líneas mapeando en el eje x “fecha” y en el eje y “suma\_llegadas”. Adicionalmente, agregue el parámetro color para que cada línea tenga un color distinto según la clase.

```
emat %>%
  filter(!is.na(clase)) %>%
  group_by(fecha, clase) %>%
  summarise(suma_llegadas = sum(total_llegadas_exp)) %>%
  ggplot() +
  aes(x = fecha, y = suma_llegadas, color = clase) +
  geom_line(size = 1)
```

