Practico 12

```
## -- Attaching packages --------------------------------------------------------- tidyverse 1.3.0 --
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
## -- Conflicts ------------------------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
##
## Attaching package: 'magrittr'
## The following object is masked from 'package:purrr':
##
##     set_names
## The following object is masked from 'package:tidyr':
##
##     extract
## Parsed with column specification:
## cols(
##   Continent_Name = col_character(),
##   Continent_Code = col_character(),
##   Country_Name = col_character(),
##   Two_Letter_Country_Code = col_character(),
##   Three_Letter_Country_Code = col_character(),
##   Country_Number = col_double()
## )
```

Introducción

En este práctico (. . .)

En particular, utilizaremos datos de la publicación ***Comparative transcriptomics analyses across species, organs, and developmental stages reveal functionally constrained lncRNAs*** (Darbellay & Necsulea, 2019).

A su vez se ejemplificarán las funciones básicas a emplear con un dataset estándar empleado en muchas demostraciones: el dataset 'iris'.

Demostración

Leyendo datos con la librería readr

```
# cargo las librerias que utilizamos en el practico
library(magrittr)
library(tidyverse)

# cargo la tabla CSV
aeropuertos = readr::read_csv('airport-codes.csv')
```

```
## Parsed with column specification:
## cols(
##   ident = col_character(),
##   type = col_character(),
##   name = col_character(),
##   elevation_ft = col_double(),
##   continent = col_character(),
##   iso_country = col_character(),
##   iso_region = col_character(),
##   municipality = col_character(),
##   gps_code = col_character(),
##   iata_code = col_character(),
##   local_code = col_character(),
##   coordinates = col_character()
## )
```

```
# la visualizo
aeropuertos
```

```
## # A tibble: 55,907 x 12
##    ident type  name  elevation_ft continent iso_country iso_region municipality
##    <chr> <chr> <chr>        <dbl> <chr>     <chr>       <chr>      <chr>
##  1 00A   heli~ Tota~           11 <NA>      US          US-PA      Bensalem
##  2 00AA  smal~ Aero~         3435 <NA>      US          US-KS      Leoti
##  3 00AK  smal~ Lowe~          450 <NA>      US          US-AK      Anchor Point
##  4 00AL  smal~ Epps~          820 <NA>      US          US-AL      Harvest
##  5 00AR  clos~ Newp~          237 <NA>      US          US-AR      Newport
##  6 00AS  smal~ Fult~         1100 <NA>      US          US-OK      Alex
##  7 00AZ  smal~ Cord~         3810 <NA>      US          US-AZ      Cordes
##  8 00CA  smal~ Gold~         3038 <NA>      US          US-CA      Barstow
##  9 00CL  smal~ Will~           87 <NA>      US          US-CA      Biggs
## 10 00CN  heli~ Kitc~         3350 <NA>      US          US-CA      Pine Valley
## # ... with 55,897 more rows, and 4 more variables: gps_code <chr>,
## #   iata_code <chr>, local_code <chr>, coordinates <chr>
```

Realizando modificaciones con la librería tidyr

FALTA PONER ALGO ACA

```
# realizo una modificacion con la libreria tidyr
aeropuertos %<>% tidyr::separate(data = ., col = 'coordinates', into = c('lon', 'lat'), sep = ', ')
```

```
# viasualizo el resultado
aeropuertos
```

```
## # A tibble: 55,907 x 13
##    ident type  name  elevation_ft continent iso_country iso_region municipality
##    <chr> <chr> <chr>        <dbl> <chr>     <chr>       <chr>      <chr>
##  1 00A   heli~ Tota~           11 <NA>      US          US-PA      Bensalem
##  2 00AA  smal~ Aero~         3435 <NA>      US          US-KS      Leoti
##  3 00AK  smal~ Lowe~          450 <NA>      US          US-AK      Anchor Point
##  4 00AL  smal~ Epps~          820 <NA>      US          US-AL      Harvest
##  5 00AR  clos~ Newp~          237 <NA>      US          US-AR      Newport
##  6 00AS  smal~ Fult~         1100 <NA>      US          US-OK      Alex
##  7 00AZ  smal~ Cord~         3810 <NA>      US          US-AZ      Cordes
##  8 00CA  smal~ Gold~         3038 <NA>      US          US-CA      Barstow
##  9 00CL  smal~ Will~           87 <NA>      US          US-CA      Biggs
## 10 00CN  heli~ Kitc~         3350 <NA>      US          US-CA      Pine Valley
## # ... with 55,897 more rows, and 5 more variables: gps_code <chr>,
## #   iata_code <chr>, local_code <chr>, lon <chr>, lat <chr>
```

Nuestro *tibble* (clase con la que se designa a las tablas de la libreria tidyverse) no posee datos en la columna *continent*. Nos deshacemos de ella utilizando la función **select()** de la librería **dplyr**.

Cargaremos otra tabla, ** **, en la cual se correlacionan correctamente los aeropuertos y sus continentes. Esto nos servirá más adelante, ya que haremos algunos análisis categorizando en base a continentes.

La misma está en formato alargado, por lo que tendremos que usar la función **pivot_longer()** de la librería **tidyr** para llevar a un formato alargado, tal como posee nuestro tibble *aeropuertos*.

Unimos estas tablas con la función **left_join** de la librería **dplyr**.

```
aeropuertos %>%
  left_join(x = ., y = pais_vs_continente.final, by = c('iso_country' = 'codigo_pais')) %>%
  dplyr::select(-continent) -> aeropuertos

aeropuertos
```

```
## # A tibble: 57,323 x 13
##    ident type  name  elevation_ft iso_country iso_region municipality gps_code
##    <chr> <chr> <chr>        <dbl> <chr>       <chr>      <chr>        <chr>
##  1 00A   heli~ Tota~           11 US          US-PA      Bensalem     00A
##  2 00AA  smal~ Aero~         3435 US          US-KS      Leoti        00AA
##  3 00AK  smal~ Lowe~          450 US          US-AK      Anchor Point 00AK
##  4 00AL  smal~ Epps~          820 US          US-AL      Harvest      00AL
##  5 00AR  clos~ Newp~          237 US          US-AR      Newport      <NA>
##  6 00AS  smal~ Fult~         1100 US          US-OK      Alex         00AS
##  7 00AZ  smal~ Cord~         3810 US          US-AZ      Cordes       00AZ
##  8 00CA  smal~ Gold~         3038 US          US-CA      Barstow      00CA
##  9 00CL  smal~ Will~           87 US          US-CA      Biggs        00CL
## 10 00CN  heli~ Kitc~         3350 US          US-CA      Pine Valley  00CN
## # ... with 57,313 more rows, and 5 more variables: iata_code <chr>,
## #   local_code <chr>, lon <chr>, lat <chr>, continente <chr>
```

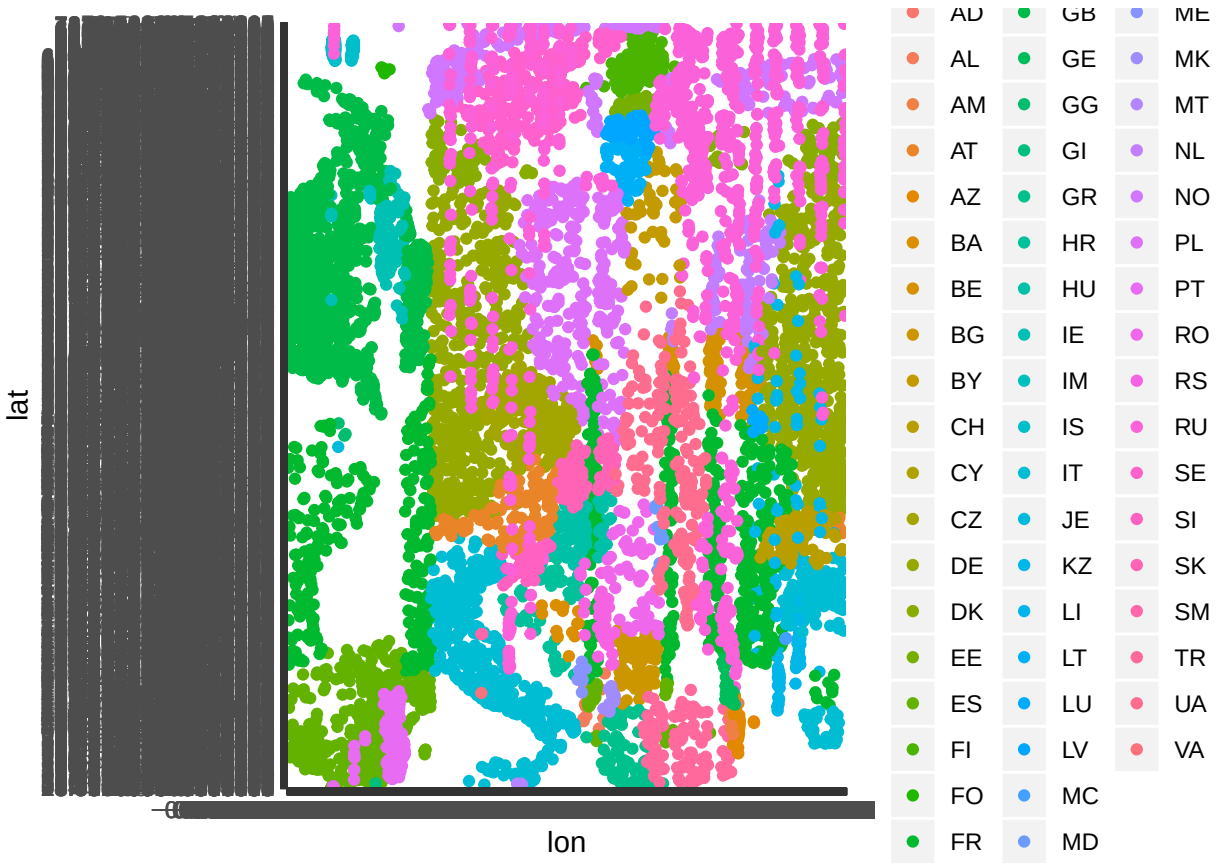
Visualizando con la librería ggplot2

PONER ACA COSAS DEL GGPLOT2

Visualizaremos ahora nuestro set de datos. Empezaremos graficando, tomando la variable **lon** como coordenada x de nuestro gráfico y la variable **lat** como coordenada y, especificando estos argumentos con la función **aes()** de la librería.

A su vez, debemos elegir una geometría que represente a nuestros puntos. En este caso, llamaremos a la función **geom_point()** para especificar que la misma será la geometría de puntos.

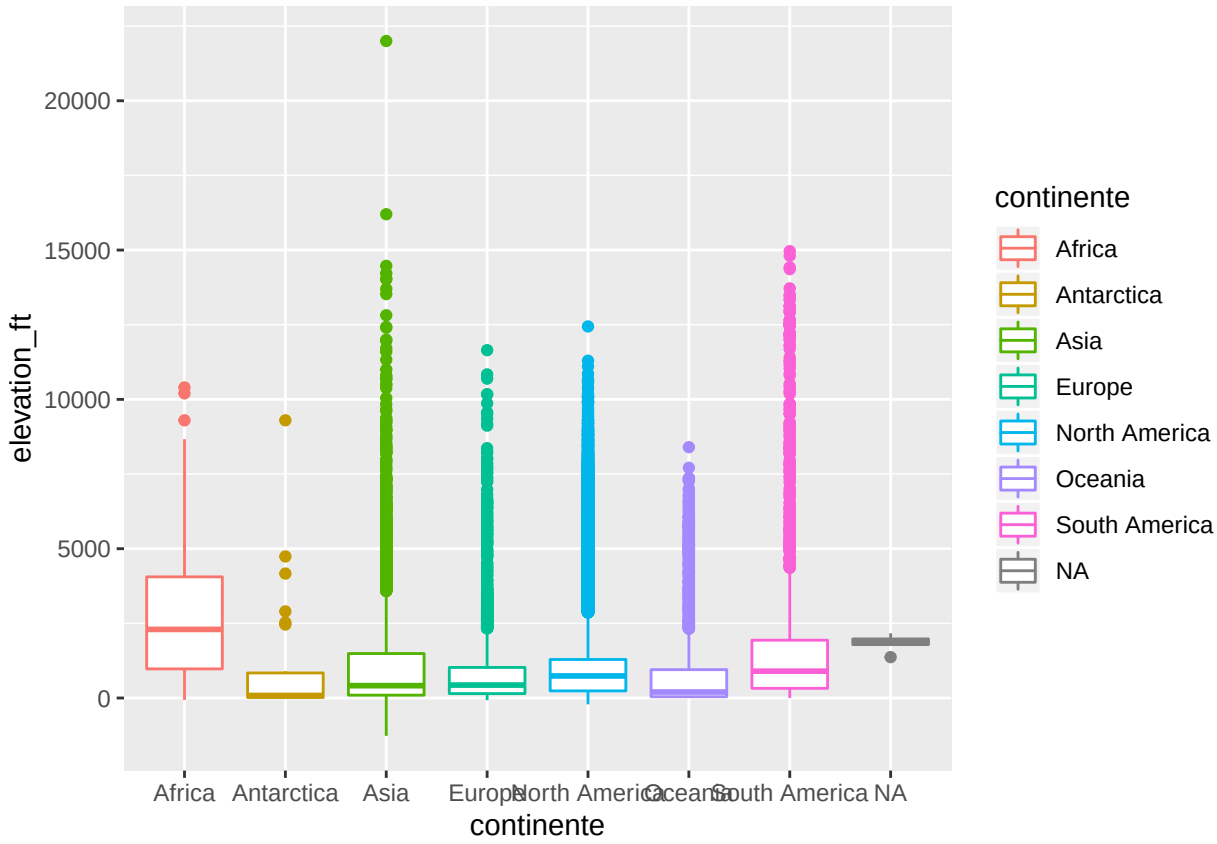
```
aeropuertos %>%
  dplyr::filter(., continente == 'Europe') %>%
  ggplot(data = .,
         mapping = aes(x = lon, y = lat, color = iso_country)) +
  geom_point()
```



A su vez podriamos realizar un boxplot para visualizar la dispersión que posee cada continente en torno a esta variable. Para ello podemos utilizar la función **geom_boxplot()** para visualizar nuestras variables con esa geometría.

```
aeropuertos %>%
  ggplot(data = .,
         mapping = aes(x = continente, y = elevation_ft, color = continente)) +
  geom_boxplot()
```

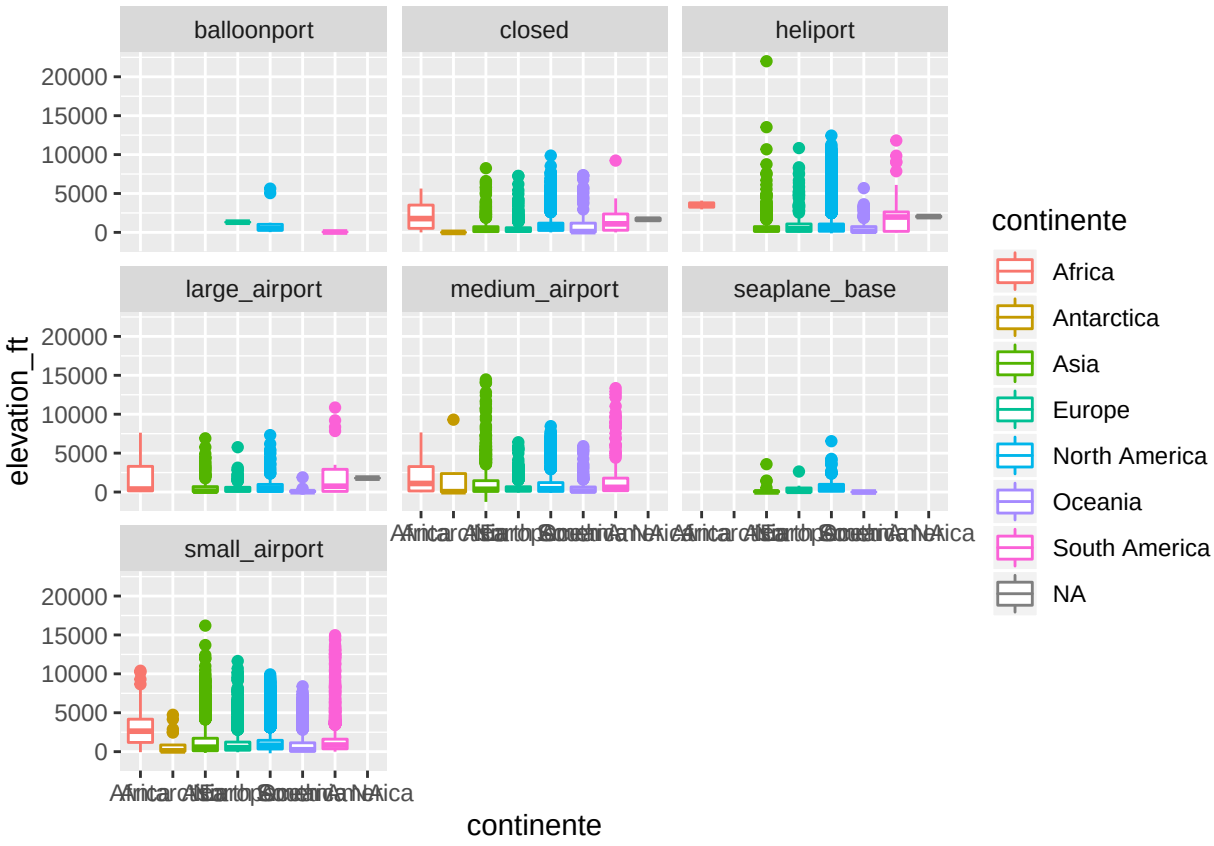
```
## Warning: Removed 7648 rows containing non-finite values (stat_boxplot).
```



Podemos a su vez subdividir nuestro analisis en base a continentes (o cualquier factor). Lo haremos con la función **facet_wrap()**.

```
aeropuertos %>%
  ggplot(data = .,
         mapping = aes(x = continente, y = elevation_ft, color = continente)) +
  geom_boxplot() +
  facet_wrap(~type)
```

```
## Warning: Removed 7648 rows containing non-finite values (stat_boxplot).
```


balloonport
closed
heliport
large_airport
medium_airport
seaplane_base
small_airport
elevation_ft
20000
15000
10000
5000
0
continente
Africa
Antarctica
Asia
Europe
North America
Oceania
South America
NA

Ejercicio