

02 Creación de Mapas en R

Dr. Victor Augusto Lizcano S.

2023-11-29

Mapas en R

Para poder visualizar nuestros mapas con un conjunto de datos vectoriales vamos a emplear la librería “ggplot2”. También emplearemos la librería “rnatuarearth” la cual nos proporciona mapas de todo el mundo.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.2
```

```
library(rnatuarearth)
```

```
## Warning: package 'rnatuarearth' was built under R version 4.2.3
```

```
## Support for Spatial objects (`sp`) will be deprecated in {rnatuarearth} and will be removed in a future version
```

```
library(rnatuarearthdata)
```

```
## Warning: package 'rnatuarearthdata' was built under R version 4.2.3
```

```
##
```

```
## Attaching package: 'rnatuarearthdata'
```

```
## The following object is masked from 'package:rnatuarearth':
```

```
##
```

```
##      countries110
```

```
library(sp)
```

```
library(raster)
```

```
theme_set(theme_bw()) ##?theme_bw
```

```
elMundo <- ne_countries(scale = "medium", returnclass = "sf") #Retorna poligonos de cada país del mundo
```

```
str(elMundo)
```

```
## Classes 'sf' and 'data.frame':  241 obs. of  64 variables:
```

```
## $ scalerank : int  3 1 1 1 1 3 3 1 1 1 ...
```

```
## $ featurecla: chr  "Admin-0 country" "Admin-0 country" "Admin-0 country" "Admin-0 country" ...
```

```
## $ labelrank : num  5 3 3 6 6 6 6 4 2 6 ...
```

```
## $ sovereignt: chr  "Netherlands" "Afghanistan" "Angola" "United Kingdom" ...
```

```
## $ sov_a3 : chr  "NL1" "AFG" "AGO" "GB1" ...
```

```
## $ adm0_dif : num  1 0 0 1 0 1 0 0 0 0 ...
```

```
## $ level : num  2 2 2 2 2 2 2 2 2 2 ...
```

```
## $ type : chr  "Country" "Sovereign country" "Sovereign country" "Dependency" ...
```

```
## $ admin : chr  "Aruba" "Afghanistan" "Angola" "Anguilla" ...
```

```
## $ adm0_a3 : chr  "ABW" "AFG" "AGO" "AIA" ...
```

```
## $ geou_dif : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
## $ geounit : chr  "Aruba" "Afghanistan" "Angola" "Anguilla" ...
```

```

## $ gu_a3      : chr "ABW" "AFG" "AGO" "AIA" ...
## $ su_dif     : num 0 0 0 0 0 0 0 0 0 0 ...
## $ subunit    : chr "Aruba" "Afghanistan" "Angola" "Anguilla" ...
## $ su_a3      : chr "ABW" "AFG" "AGO" "AIA" ...
## $ brk_diff   : num 0 0 0 0 0 0 0 0 0 0 ...
## $ name       : chr "Aruba" "Afghanistan" "Angola" "Anguilla" ...
## $ name_long  : chr "Aruba" "Afghanistan" "Angola" "Anguilla" ...
## $ brk_a3     : chr "ABW" "AFG" "AGO" "AIA" ...
## $ brk_name   : chr "Aruba" "Afghanistan" "Angola" "Anguilla" ...
## $ brk_group  : chr NA NA NA NA ...
## $ abbrev     : chr "Aruba" "Afg." "Ang." "Ang." ...
## $ postal     : chr "AW" "AF" "AO" "AI" ...
## $ formal_en  : chr "Aruba" "Islamic State of Afghanistan" "People's Republic of Angola" NA ...
## $ formal_fr  : chr NA NA NA NA ...
## $ note_adm0  : chr "Neth." NA NA "U.K." ...
## $ note_brk   : chr NA NA NA NA ...
## $ name_sort  : chr "Aruba" "Afghanistan" "Angola" "Anguilla" ...
## $ name_alt   : chr NA NA NA NA ...
## $ mapcolor7  : num 4 5 3 6 1 4 1 2 3 3 ...
## $ mapcolor8  : num 2 6 2 6 4 1 4 1 1 1 ...
## $ mapcolor9  : num 2 8 6 6 1 4 1 3 3 2 ...
## $ mapcolor13: num 9 7 1 3 6 6 8 3 13 10 ...
## $ pop_est    : num 103065 28400000 12799293 14436 3639453 ...
## $ gdp_md_est: num 2258 22270 110300 109 21810 ...
## $ pop_year   : num NA NA NA NA NA NA NA NA NA NA ...
## $ lastcensus: num 2010 1979 1970 NA 2001 ...
## $ gdp_year   : num NA NA NA NA NA NA NA NA NA NA ...
## $ economy    : chr "6. Developing region" "7. Least developed region" "7. Least developed region" ...
## $ income_grp: chr "2. High income: nonOECD" "5. Low income" "3. Upper middle income" "3. Upper middle income" ...
## $ wikipedia  : num NA NA NA NA NA NA NA NA NA NA ...
## $ fips_10     : chr NA NA NA NA ...
## $ iso_a2      : chr "AW" "AF" "AO" "AI" ...
## $ iso_a3      : chr "ABW" "AFG" "AGO" "AIA" ...
## $ iso_n3      : chr "533" "004" "024" "660" ...
## $ un_a3       : chr "533" "004" "024" "660" ...
## $ wb_a2       : chr "AW" "AF" "AO" NA ...
## $ wb_a3       : chr "ABW" "AFG" "AGO" NA ...
## $ woe_id      : num NA NA NA NA NA NA NA NA NA NA ...
## $ adm0_a3_is  : chr "ABW" "AFG" "AGO" "AIA" ...
## $ adm0_a3_us  : chr "ABW" "AFG" "AGO" "AIA" ...
## $ adm0_a3_un  : num NA NA NA NA NA NA NA NA NA NA ...
## $ adm0_a3_wb  : num NA NA NA NA NA NA NA NA NA NA ...
## $ continent  : chr "North America" "Asia" "Africa" "North America" ...
## $ region_un   : chr "Americas" "Asia" "Africa" "Americas" ...
## $ subregion   : chr "Caribbean" "Southern Asia" "Middle Africa" "Caribbean" ...
## $ region_wb   : chr "Latin America & Caribbean" "South Asia" "Sub-Saharan Africa" "Latin America & Caribbean" ...
## $ name_len    : num 5 11 6 8 7 5 7 20 9 7 ...
## $ long_len    : num 5 11 6 8 7 13 7 20 9 7 ...
## $ abbrev_len  : num 5 4 4 4 4 5 4 6 4 4 ...
## $ tiny        : num 4 NA NA NA NA 5 5 NA NA NA ...
## $ homepart    : num NA 1 1 NA 1 NA 1 1 1 1 ...
## $ geometry    : sfc_MULTIPOLYGON of length 241; first list element: List of 1
## ..$ :List of 1
## ...$ : num [1:10, 1:2] -69.9 -69.9 -69.9 -70 -70.1 ...

```

```
##   ..- attr(*, "class")= chr [1:3] "XY" "MULTIPOLYGON" "sfg"
##   - attr(*, "sf_column")= chr "geometry"
##   - attr(*, "agr")= Factor w/ 3 levels "constant","aggregate",...: NA NA NA NA NA NA NA NA NA ...
##   ..- attr(*, "names")= chr [1:63] "scalerank" "featurecla" "labelrank" "sovereight" ...
```

#Ahora grafiquemos nuestro mapa

```
ggplot(data = elMundo) + geom_sf()
```

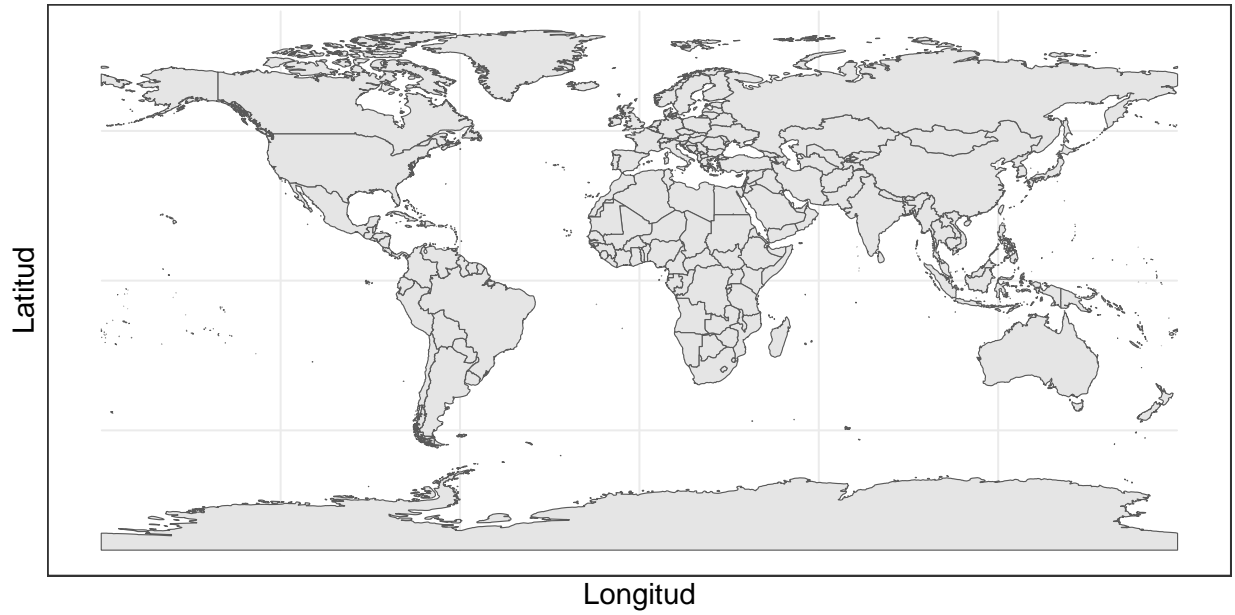


#Decoremos nuestro mapa

```
ggplot(data = elMundo) + geom_sf() + xlab("Longitud") + ylab("Latitud") +
  ggtitle("Países del Mundo", subtitle = paste0("(", length(unique(elMundo$name)), " Países"))
```

Países del Mundo

(241 Países)

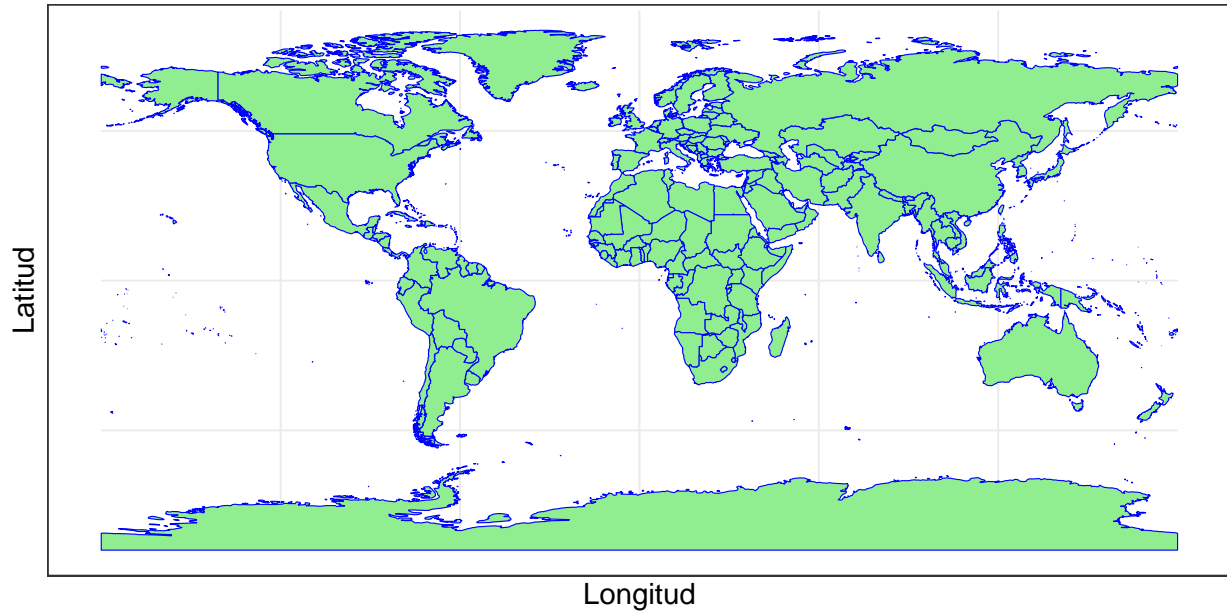


#Asignemosle colores

```
ggplot(data = elMundo) + geom_sf(color="blue", fill = "lightgreen") + xlab("Longitud") + ylab("Latitud") +  
  ggtitle("Países del Mundo", subtitle = paste0("(", length(unique(elMundo$name)), " Países"))
```

Países del Mundo

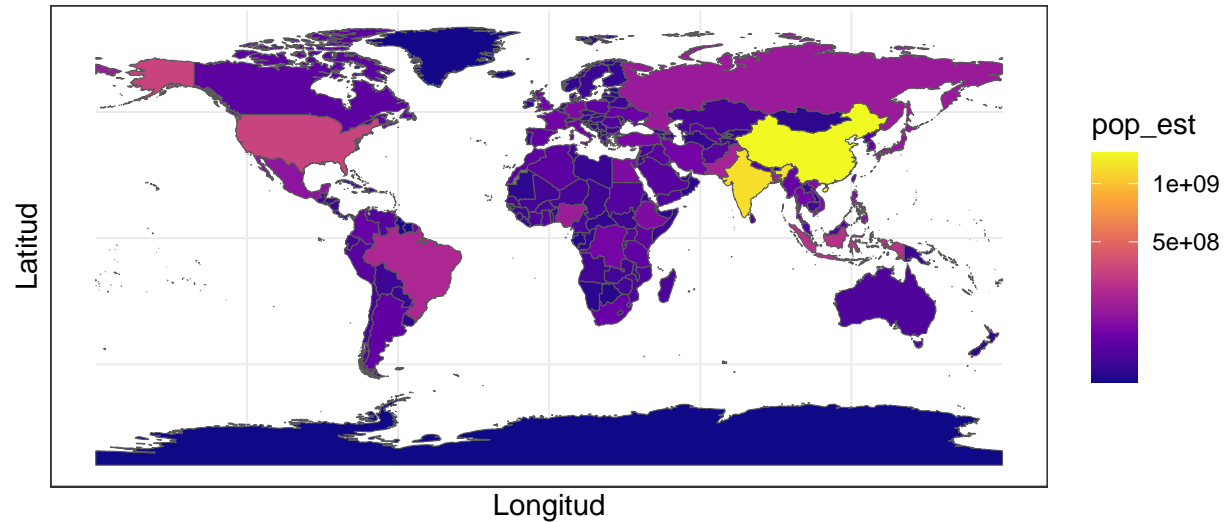
(241 Países)



```
ggplot(data = elMundo) + geom_sf(aes(fill = pop_est)) + # pop_est es una de las variables contenidas en elMundo
  scale_fill_viridis_c(option = "plasma", trans = "sqrt") + # plasma es la variante de colores y sqrt la transformación
  xlab("Longitud") + ylab("Latitud") +
  ggtitle("Países del Mundo", subtitle = paste0("(", length(unique(elMundo$name)), " Países"))
```

Países del Mundo

(241 Países)



Proyección y extensión

Los sistemas de coordenadas geográficas se basan en una superficie esferoidal (ya sea verdaderamente esférica o elipsoidal) que se aproxima a la superficie de la Tierra. Un **datum** suele definir la superficie (radio ex para una esfera, eje mayor y eje menor o aplanamiento inverso para un elipsoide) y la posición de la superficie en relación con el centro de la Tierra. Por ejemplo:

Esferoide

Datum

Semieje mayor

Semieje menor

Achatamiento

Clarke 1866

NAD 1927

6378206.4 m

6356583.8 m

294.978698214

GRS80 1980

NAD 1983

6378137.0 m

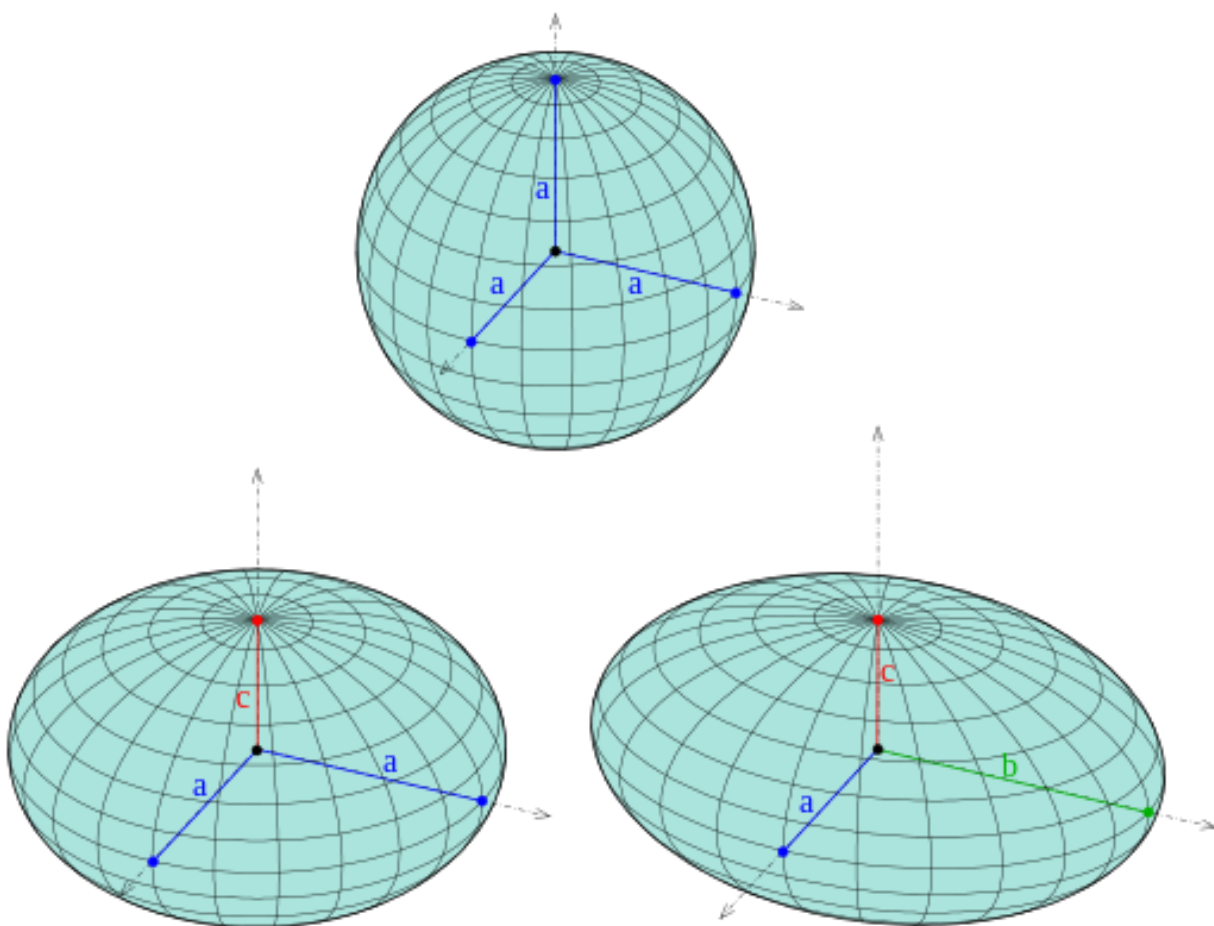


Figure 1: Elipsoides donde a , b y c son las longitudes de los semiejes del elipsoide respecto de los ejes x , y , z

6356752.3 m
294.978698214
WGS84 1984
WGS 1984
6378137.0 m
6356752.3142 m
298.2572235634

El geoide, el elipsoide, el esferoide y el datum, y cómo se relacionan

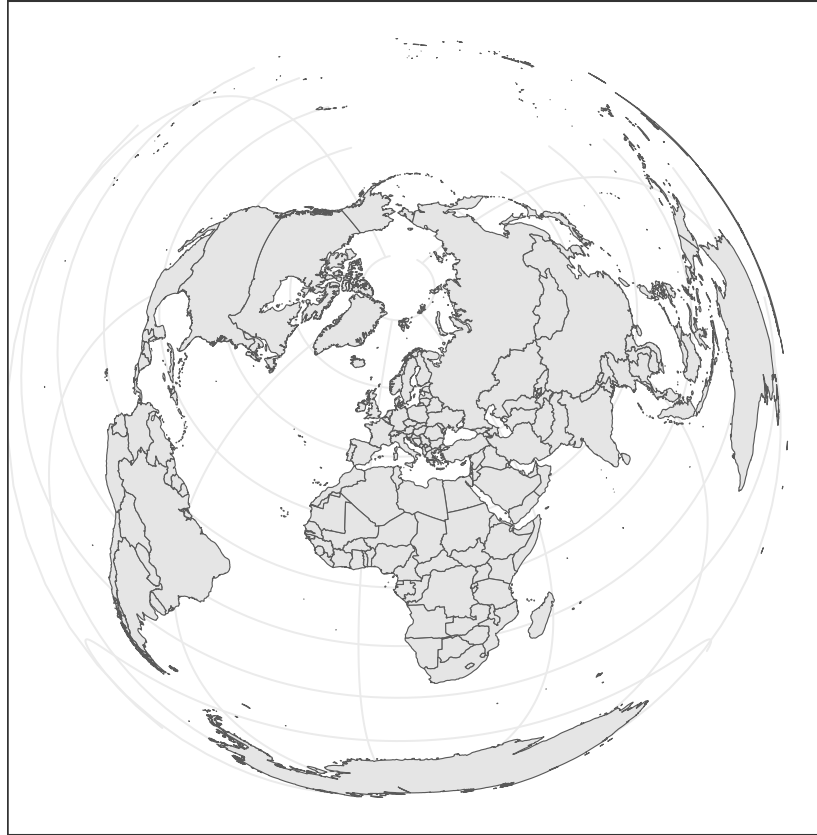
Todo sistema de referencia de coordenadas geográficas hace referencia (valga la redundancia) a un **datum**. Por otra parte tenga en cuenta que este src no se proyecta y se representa en latitud y longitud.

Por otra parte, los sistemas de referencia de coordenadas proyectados, son transformaciones que convierten un conjunto de puntos de una superficie curva (es decir está en coordenadas geográficas) en una superficie plana (generalmente medida en metros).

El datum es una parte integral de la proyección, ya que los sistemas coordinados proyectados se basan en coordenadas geográficas, que a su vez están referenciadas a un datum. Es posible, e incluso común, que los datasets estén en la misma proyección, pero que se haga referencia a diferentes datums y, por lo tanto, tengan diferentes valores de coordenadas. Por ejemplo, se puede hacer referencia a los sistemas de coordenadas del plano de estado a los datums NAD83 y NAD27. Las transformaciones de coordenadas geográficas a coordenadas proyectadas son las mismas, pero como las coordenadas geográficas son diferentes en función del datum, las coordenadas proyectadas resultantes también serán diferentes.

Miremos algunos ejemplos en R:

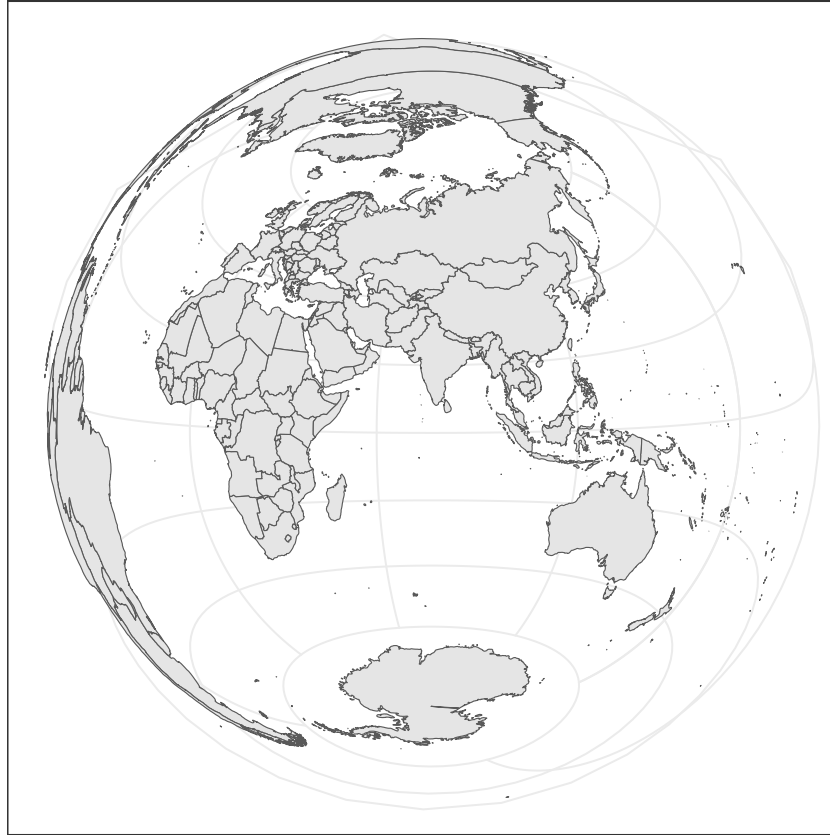
```
ggplot(data = elMundo) +  
  geom_sf() +  
  coord_sf(crs = "+proj=laea +lat_0=52 +lon_0=10 +x_0=4321000 +y_0=3210000 +ellps=GRS80 +units=m +no_
```

#+proj=laea: es el tipo de proyección que se está utilizando, en este caso, Lambert Azimuthal Equal Area.
#+lat_0=52: es el paralelo de origen en grados, que es la latitud en la que el plano del mapa toca la esfera.
#+lon_0=10: es el meridiano central en grados, que es la longitud en la que el plano del mapa toca la esfera.
#+x_0=4321000 y +y_0=3210000: son los falsos este y norte respectivamente. Son las coordenadas en el sistema de proyección.
#+ellps=GRS80: es el elipsoide de referencia que se está utilizando.
#+units=m: es la unidad de medida utilizada en el sistema de proyección.
#+no_defs: este argumento se utiliza para suprimir la carga de las definiciones de elipsoide y datum por defecto.

Cambiemos la longitud y latitud en la que el plano del mapa toca la esfera en 76,3

```
ggplot(data = elMundo) +  
  geom_sf() +  
  coord_sf(crs = "+proj=laea +lat_0=3 +lon_0=76 +ellps=GRS80 +units=m +no_defs")
```



Recordemos que la **proyección** es el método empleado para traducir una posición en el globo o geoide en una posición en un mapa bidimensional (es decir con x,y coordenadas). Las proyecciones pueden preservar ciertas propiedades como el área, la dirección (ángulos) o las longitudes. En el caso del **elipsoide**, este es un modelo matemático o de referencia empleado para aproximar la forma de la superficie terrestre.

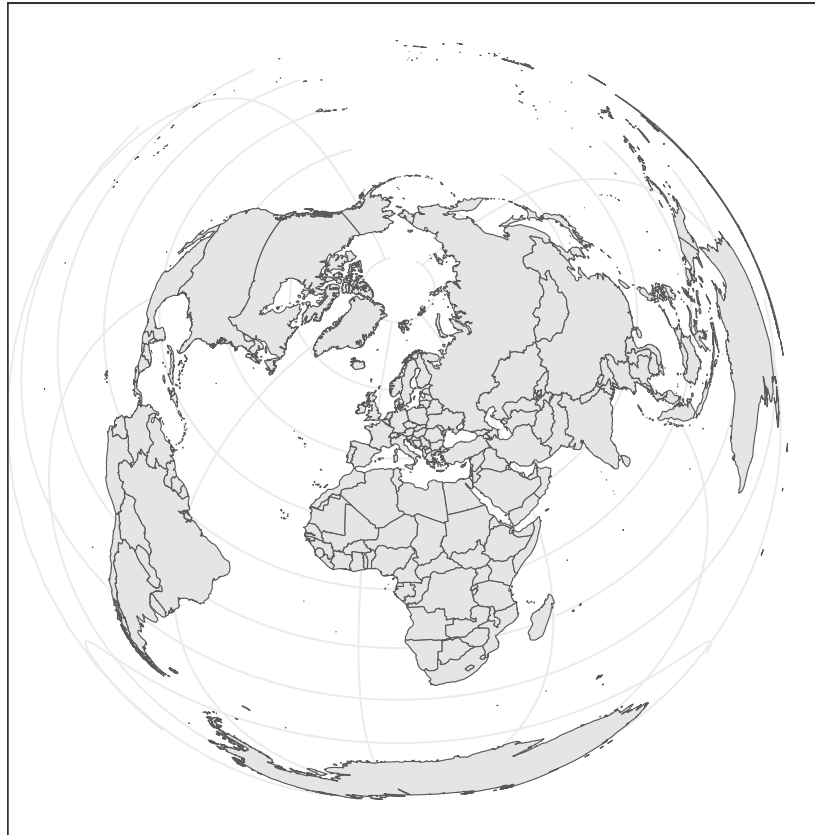
Nota: el falso este y el falso norte son valores lineales que se aplican al origen de las coordenadas x e y, respectivamente. Se utilizan en los sistemas de coordenadas proyectadas para asegurar que todas las coordenadas x e y sean positivas (Ver).

A continuación vamos a ver visualizaciones de nuestro mapa mundi empleando identificadores EPSG para algunos SRC (Ver más).

```
#epsg 3035
#+proj=laea +lat_0=52 +lon_0=10 +x_0=4321000 +y_0=3210000 +ellps=GRS80 +units=m +no_defs
#https://spatialreference.org/ref/epsg/3035/
```

```
ggplot(data = elMundo) +
  geom_sf() +
  coord_sf(crs = "+init=epsg:3035")
```

```
## Warning in CPL_crs_from_input(x): GDAL Message 1: +init=epsg:XXXX syntax is
## deprecated. It might return a CRS with a non-EPSG compliant axis order.
```



```
#epsg 4326
#+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs
#https://spatialreference.org/ref/epsg/4326/
ggplot(data = elMundo) +
  geom_sf() +
  coord_sf(crs = "+init=epsg:4326")
```

