

Laboratorio de procesamiento de información meteorológica

TP final: Corriente en chorro en capas bajas
(LLJ)

29/11/2022

Objetivos

- a) Leer los datos del viento zonal y meridional para los niveles entre 1000 y 500 hPa.
- b) Calcular el criterio de Bonner en el nivel de 850 hPa para todos los tiempos (nivel superior 600 hPa).
- c) Definir una variable donde su valor es 1 si se cumple el criterio de bonner.
- d) Graficar en 4 paneles la evolución del criterio de Bonner junto con el viento en vectores.
- e) Guardar la variable del criterio de Bonner en un archivo binario de doble precisión y Little Endian. (Para los puntos donde no se cumple el criterio definir un valor de undef). Crear un archivo de control (header).

Consideraciones

- Para estimar la posición del LLJ en el norte argentino se puede utilizar el criterio de Bonner, donde la condición para que se cumpla el mismo es que el viento en el nivel elegido sea superior a 12 m/s y la cortante vertical del viento entre ese nivel y un nivel superior supere los 6 m/s. Se tienen los datos de análisis del GDAS, de un evento de LLJ, en formato binario
- La zona de estudio será el Norte Argentino
- El periodo a analizar será 26/03/2007 – 28/03/2007

Vamos al código en R

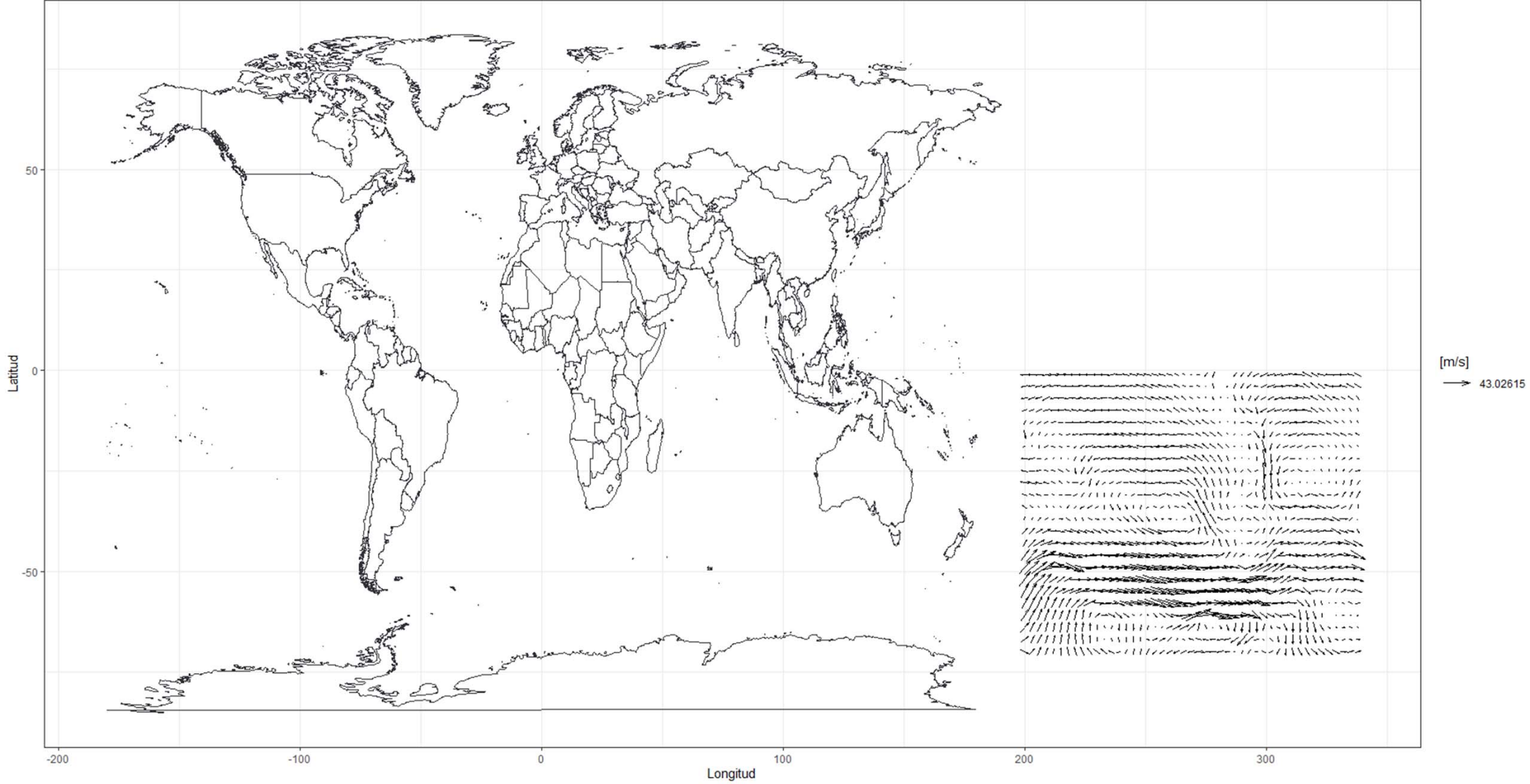
ERRORES

Omitir el uso de la función :

#Usamos la función de metR para convertir longitudes:

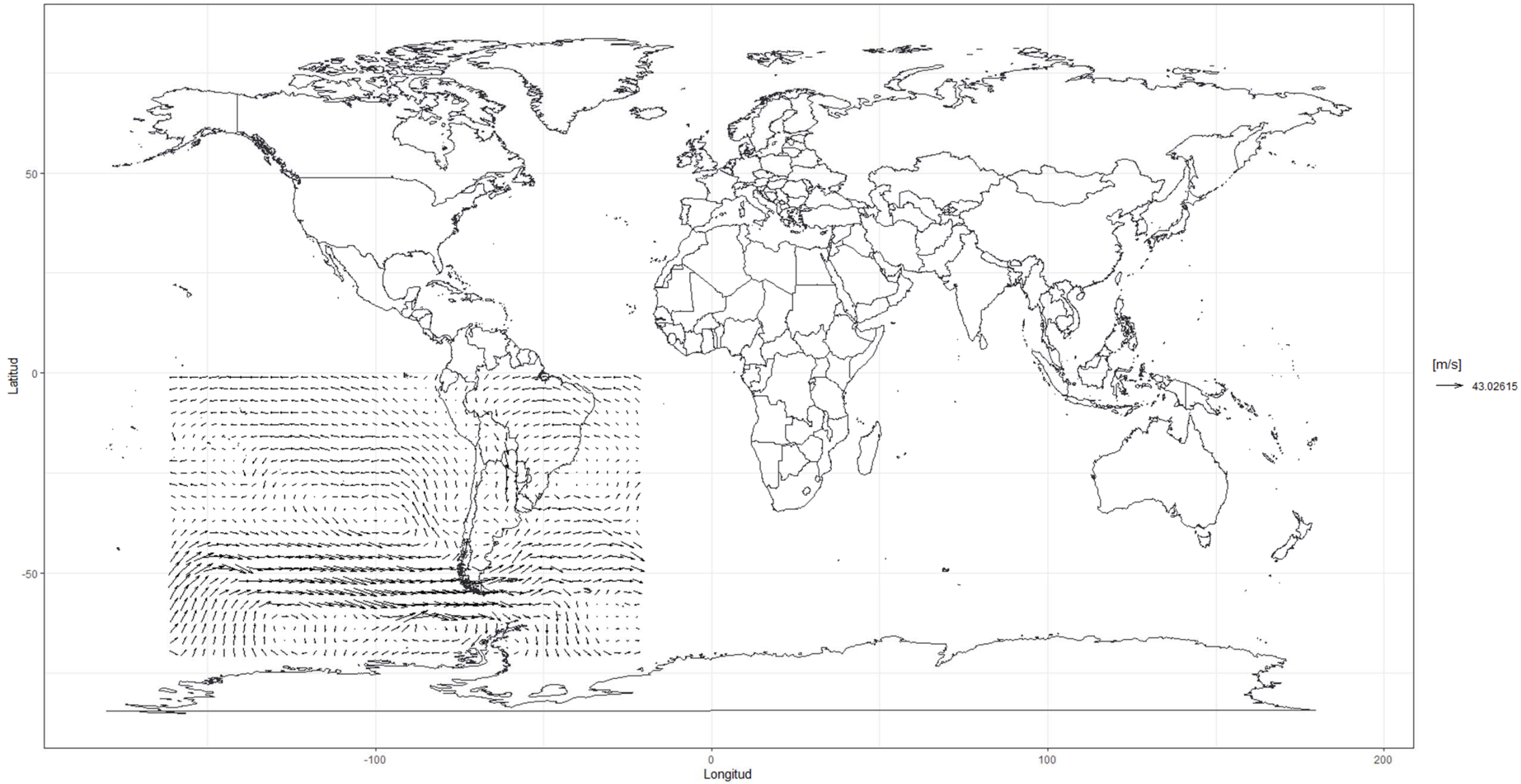
```
df$lon <- ConvertLongitude(df$lon)
```

Viento en 850 hPa



Usando la función

Viento en 850 hPa



Cargar mal los datos

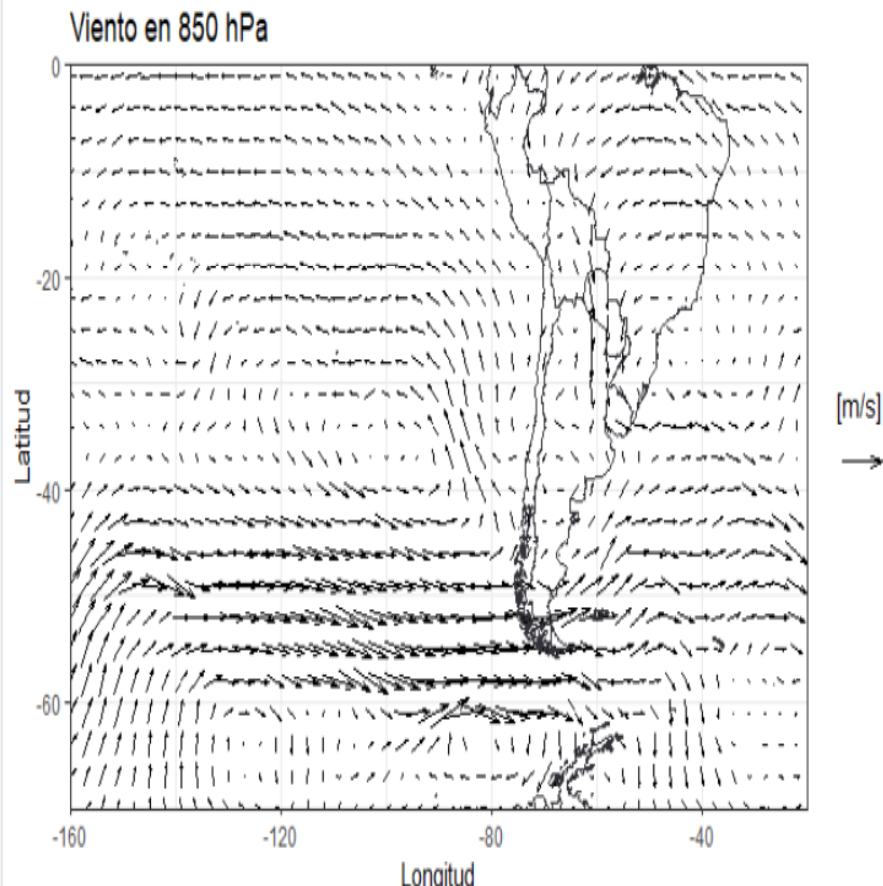
```
#Con ciclo for abrimos todos los archivos
for (i in 1:length(archivos)) {
  datos_array<-array(
    readBin(paste0(datos,archivos[i]),"numeric", size=4, n=nrecords, endian="little"),
    dim=c(nlons,nlats,nlevs,nvars,ntime)
  )
}
```



```

202
203 # 5) Graficado de viento -----
204
205 #Grafiquemos el viento: para eso tenemos que agregar a u y a v a nuestro df:
206
207 Viento_zonal_850_12<-c(Viento_zonal_850[,2])
208 Viento_meridional_850_12<-c(Viento_meridional_850[,2])
209
210 df$Viento_zonal_850_12 = as.vector(Viento_zonal_850_12) #Lo guardamos en el data frame como vector
211 df$Viento_meridional_850_12 = as.vector(Viento_meridional_850_12) #Lo guardamos en el data frame en la columna V
212
213
214 # 5) Graficado de viento -----
215
216
217 #Vamos a usar la capa geometrica geom_arrow() de metR pensada para graficar
218
219 # Ajustemos el tamaño con scale_mag() de metR y usemos el parametro skip
220 # para que no grafique todas las flechas.
221
222 ggplot(df,aes(x=lon,y=lats))+
223   geom_arrow(aes(dx=Viento_zonal_850_12, dy=Viento_meridional_850_12), |
224             skip = 2,
225             size = 0.3,
226             arrow.type = "open",
227             color = "black")+
228   coord_quickmap(xlim = range(df$lon), ylim = range(df$lats), expand = FALSE)+
229   scale_mag() +
230   mi_mapa +
231   labs(title="Viento en 850 hPa",
232        x = "Longitud",
233        y = "Latitud",
234        mag = "[m/s]") +
235   theme_bw()
236
237 # 6) Graficado Criterio de Bonner -----
238

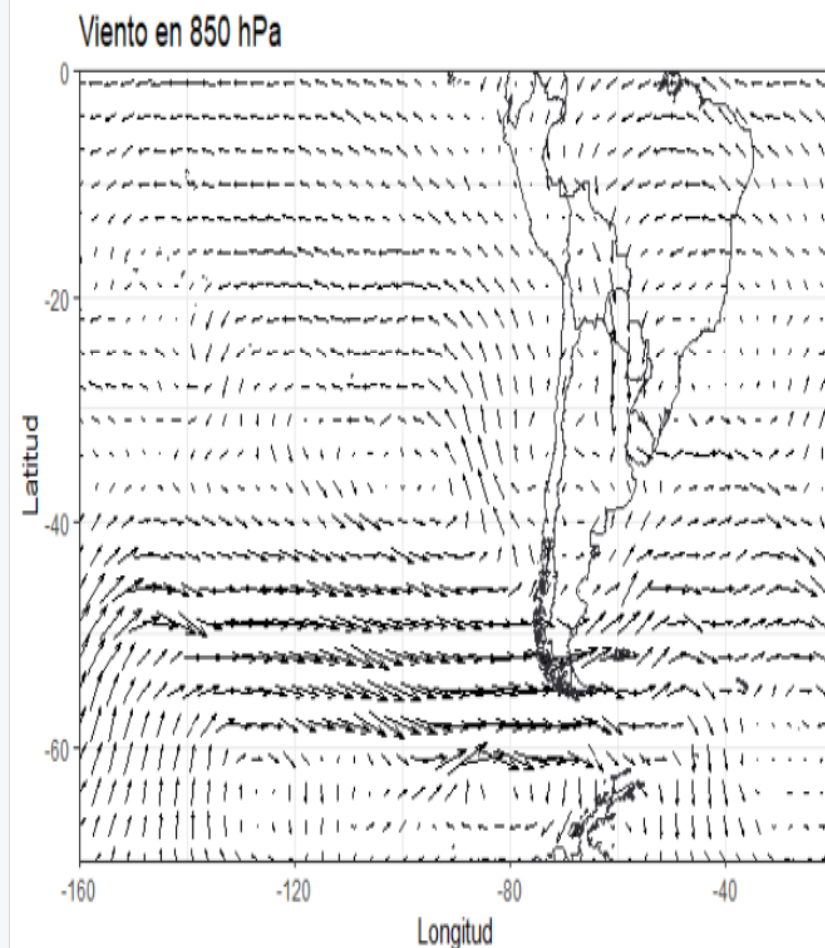
```



```

203 # 5) Graficado de viento -----
204
205 #Grafiquemos el viento: para eso tenemos que agregar a u y a v a nuestro df:
206
207 Viento_zonal_850_12<-c(Viento_zonal_850[,4])
208 Viento_meridional_850_12<-c(Viento_meridional_850[,4])
209
210 df$Viento_zonal_850_12 = as.vector(Viento_zonal_850_12) #Lo guardamos en el data frame como vector
211 df$Viento_meridional_850_12 = as.vector(Viento_meridional_850_12) #Lo guardamos en el data frame en la columna V
212
213
214 # 5) Graficado de viento -----
215
216
217 #Vamos a usar la capa geometrica geom_arrow() de metR pensada para graficar
218
219 # Ajustemos el tamaño con scale_mag() de metR y usemos el parametro skip
220 # para que no grafique todas las flechas.
221
222 ggplot(df,aes(x=lon,y=lats))+
223   geom_arrow(aes(dx=Viento_zonal_850_12, dy=Viento_meridional_850_12),
224             skip = 2,
225             size = 0.3,
226             arrow.type = "open",
227             color = "black")+
228   coord_quickmap(xlim = range(df$lon), ylim = range(df$lats), expand = FALSE)+
229   scale_mag() +
230   mi_mapa +
231   labs(title="Viento en 850 hPa",
232        x = "Longitud",
233        y = "Latitud",
234        mag = "[m/s]") +
235   theme_bw()
236
237 # 6) Graficado Criterio de Bonner -----

```



Preguntas

Una cuestión en donde hice foco fue en la carga de archivos. Por que es donde el código demora mas tiempo de ejecución.

Por eso mostré distintos perfiles con librería profvits comparando y viendo mas en detalle la carga de datos y sus tiempos de ejecución.

Haciendo memoria de las clases teóricas tenemos una librería en R para paralelizar(`library(parallel)`) y usar la función `mclapply`.

Una de las maneras más simples de paralelizar es usar la función **`mclapply`** esta función se comporta de manera similar a la función **`lapply`**, pero distribuye las diferentes llamadas a la función en diferentes sub-instancias o procesos cada una de las cuales puede correr en un procesador diferente.

Repasamos lo que hace la función **`lapply`**:

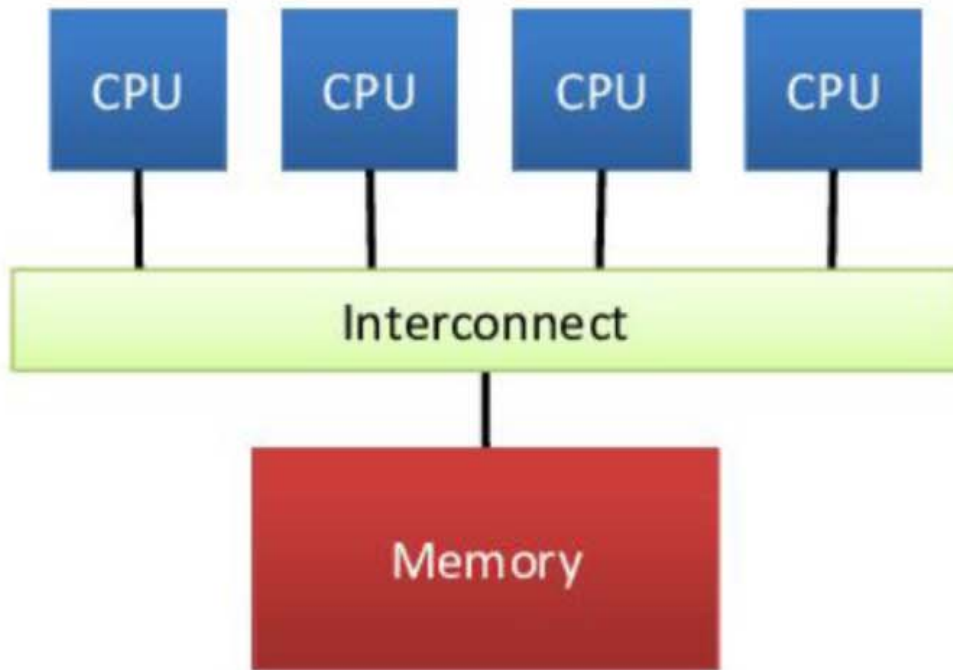
```
lapply( lista , funcion , parametros_adicionales )
```

Entonces... porque no usamos esta función `mcapply` para que cada core del CPU cargue los archivos.

Suena bien, tenemos 8 archivos, supongo 8 cores, 1 core por archivo.

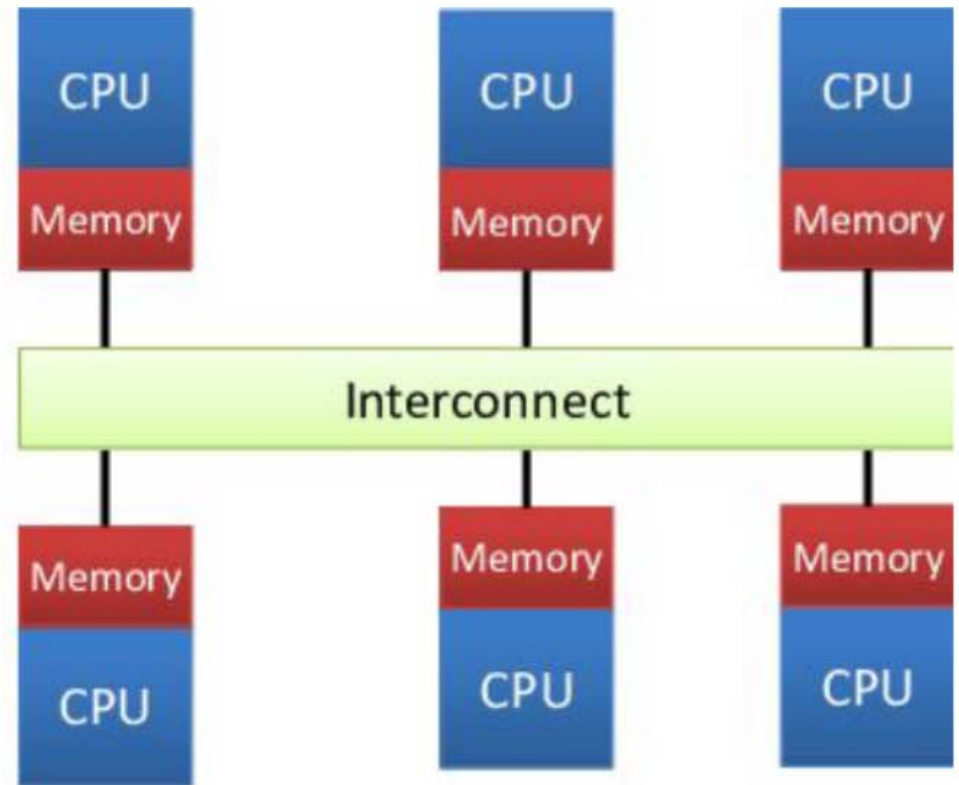
¿Es valido hacer esto?

Tipos de máquinas paralelas



(a)

Memoria compartida



(b)

Memoria distribuida

FIN