Validación - Obs de superficie

Pao

June 4, 2019

Las observaciones asimiladas están en "obs_YYYYMMDD_HH_asimiladas.dat" y contienen las observaciones de la hora anterior.

Internamente, hay una variable `time.slot` que indica en que subventana se asimiló. Hay 7 ventantas y parecería que el slot 7 de un set de observaciones es igual al slot 1 del set siguiente. *REVISAR*

La variable `obs.id`es númerica y permite clasificar los tipos de observaciones. Los que nos interesan son:

surface observations codes > 9999

- id_ps_obs=14593 -> Presión en superficie
- id_us_obs=82819 -> Viento zonal en 10 metros
- id_vs_obs=82820 -> Viento meridional en 10 metros
- id_ts_obs=83073 -> Temperatura en 2 metros
- id_qs_obs=83330 -> Humedad en 2 metros
- id_rhs_obs=83331 -> Humedad relativa

Pruebas para un tiempo en particular

```
filename <- 'obs_20181120_00_asimiladas.dat'

obs <- read.obs.asim(filename)

knitr::kable(head(obs))
```

obs.id	lon	lat	elev	obs	error	sub.id	ens.obs	ana.obs	time.slot
82819	299.9019	-38.40451	128.0	1.745407	1.4	22	0.0089669	-0.1412853	3
82820	299.9019	-38.40451	128.0	-5.069030	1.4	22	0.0336473	0.6689476	3
83073	299.9019	-38.40451	128.0	293.850006	2.0	22	0.0275071	0.0908800	3
83331	299.9019	-38.40451	128.0	71.000000	10.0	22	0.0434761	0.1745307	3
82819	299.7237	-38.37822	111.9	4.504314	1.4	22	0.0108208	0.1198610	3
82820	299.7237	-38.37822	111.9	-7.208409	1.4	22	0.0260465	-0.6379785	3

La función lee un .dat y filtra solo las observaciones de superfice.

Distribución temporal

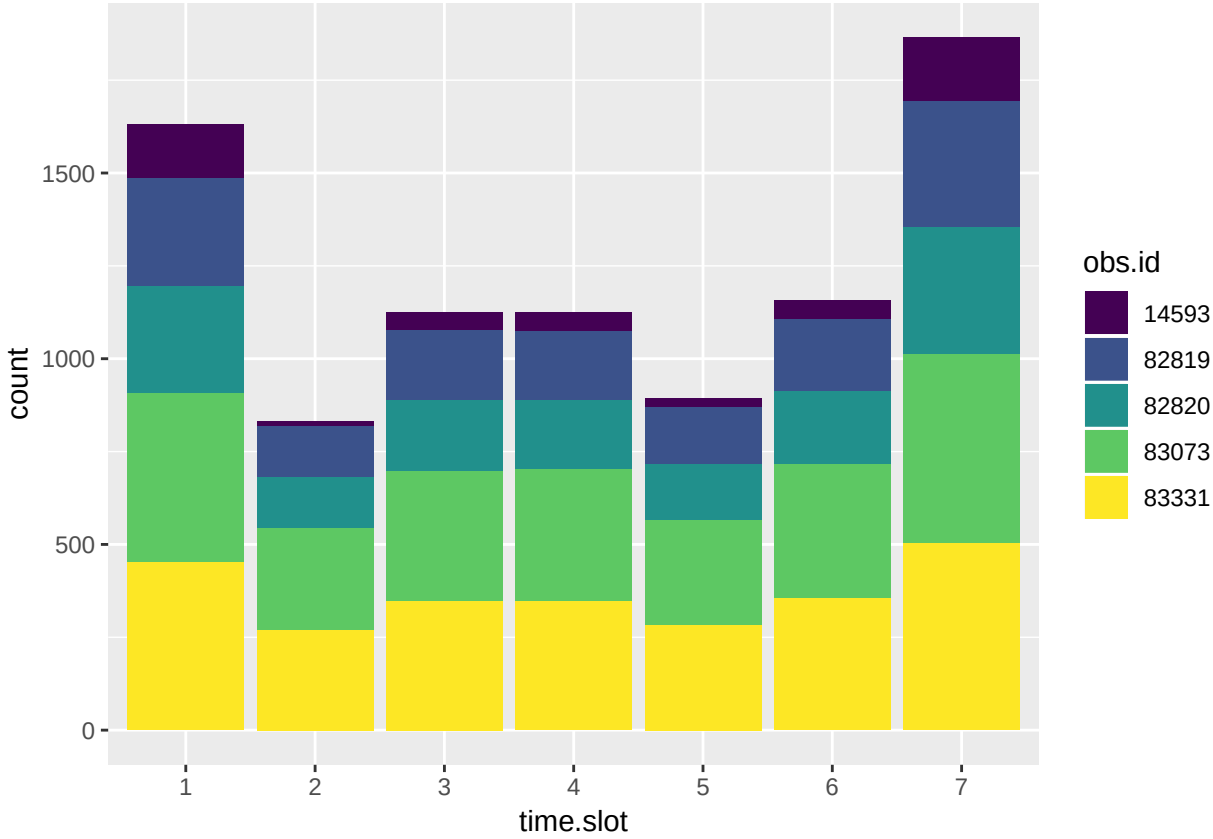
¿Cuántas osbservaciones hay? ¿Cómo se distribuyen en los slots?

```
obs[, .N, keyby = .(time.slot, obs.id)] %>%
  dcast(time.slot ~ obs.id, value.var = 'N') %>%
  knitr::kable()
```

time.slot	14593	82819	82820	83073	83331
1	145	290	290	455	452
2	13	137	137	272	272

time.slot	14593	82819	82820	83073	83331
3	47	190	190	351	348
4	51	186	186	353	350
5	22	152	152	283	283
6	49	196	196	360	357
7	170	341	341	507	506

```
ggplot(obs, aes(factor(time.slot))) +
  geom_bar(aes(fill = factor(obs.id))) +
  scale_fill_viridis_d(name = 'obs.id') +
  scale_x_discrete(name = 'time.slot')
```



En el primer y último slot hay un poco más de observaciones porque son las horas en punto *(queda ver que slot1(20181120_00) = 20181119_2300 = slot7(20181119_23))*. En el medio más o menos se mantiene.

Analicemos la diferencia entre cada observación y el análisis *(¿Cómo hace la comparación? Con el análisis de las 00?)*.

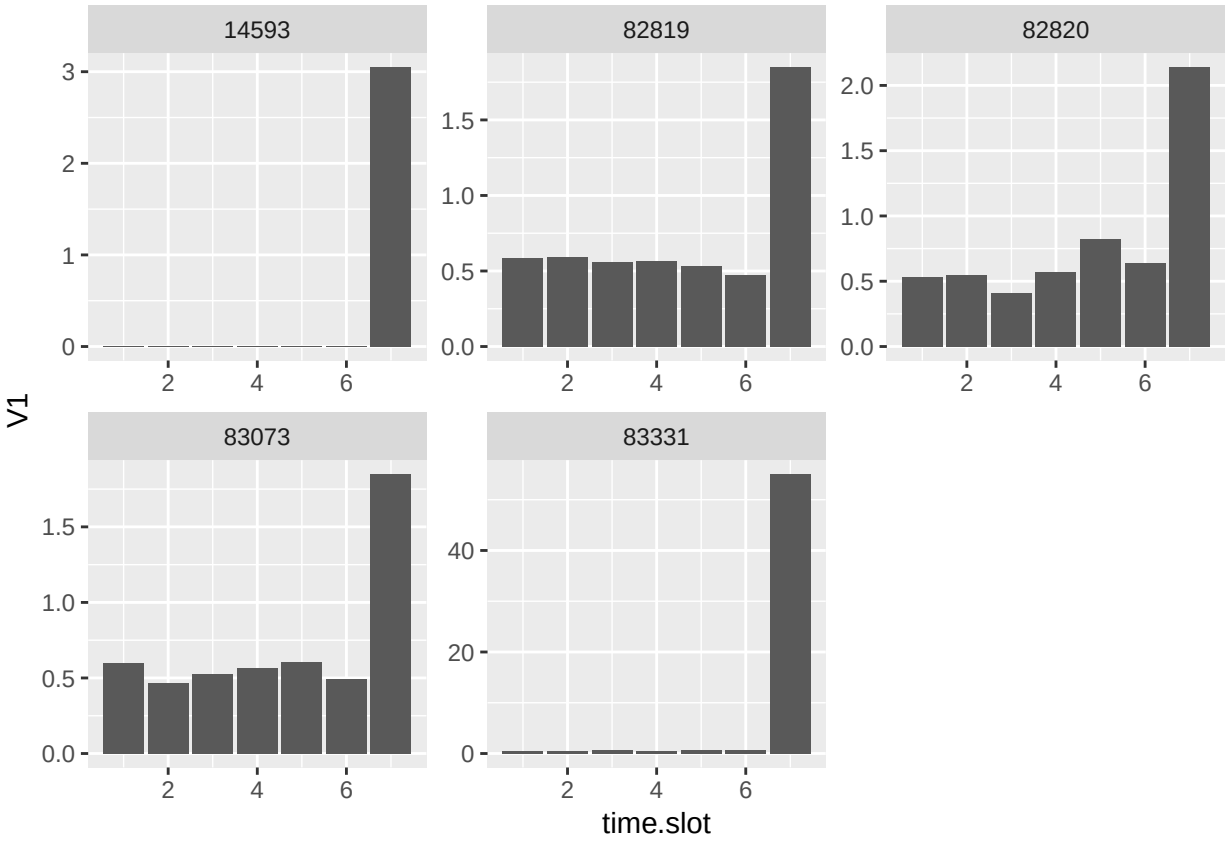
Uso un RMSE para que las diferencias positivas y negativas no se cancelen entre si. Es decir:

$$\sqrt{\sum (ana.obs)^2/N}$$

Para cada tipo de observación y slot.

```
obs[, sqrt(sum(ana.obs^2)/.N), by = .(obs.id, time.slot)] %>%
  ggplot(aes(time.slot, V1)) +
```

```
  geom_col() +
  facet_wrap(~ obs.id, scales = 'free')
```

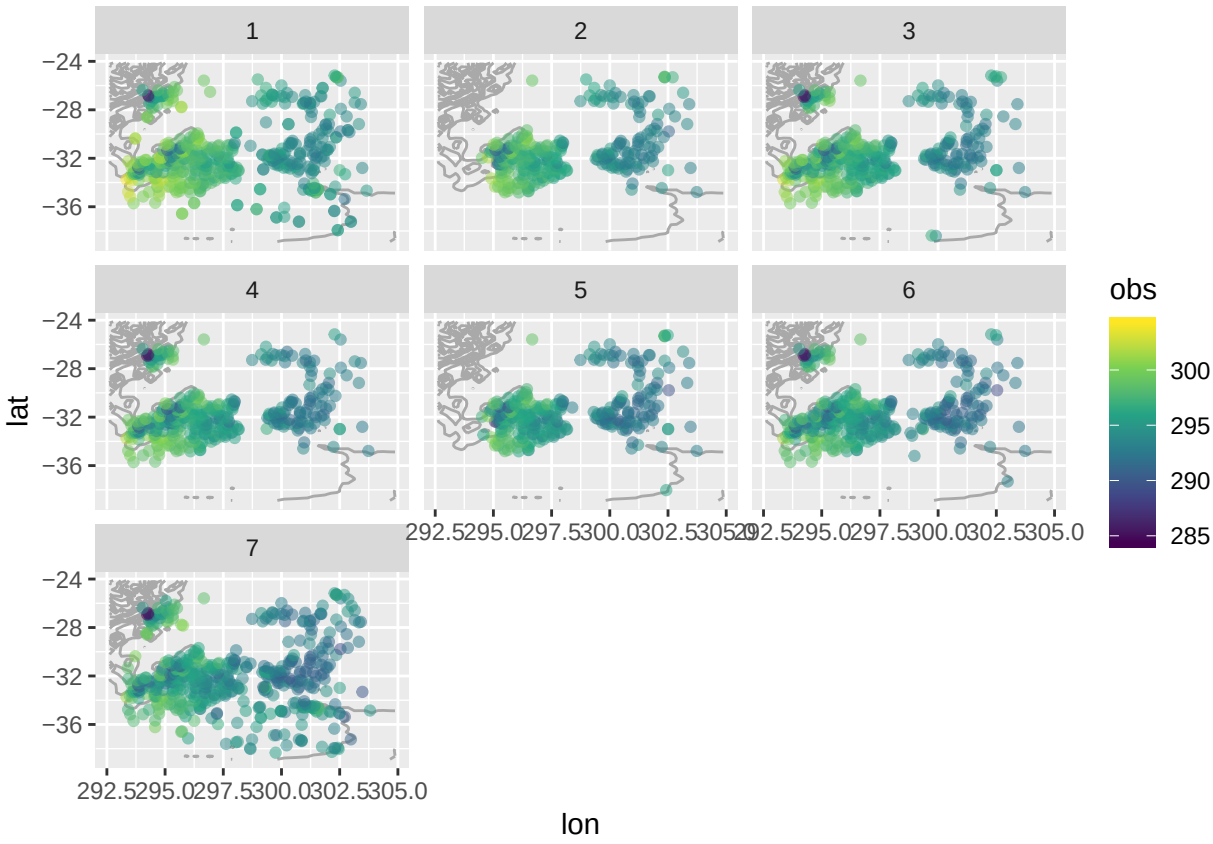


En el caso de la presión, el RMSE es tan chico en los primeros slots que no se ve (del orden de 10-3). Lo extraño es que para todas las observaciones, el RMSE se dispara en el último slot. *¿Tendrá que ver con como se genera la diferencia entre la observación y el análisis? Es decir, estas son las observaciones entre las 23:00 y las 00:00 (con nombre 00), se comparan con el analisis de las 23 o el de las 00?*

Distribución espacial

Veamos como se distribuyen especialmente las observaciones. Me quedo solo con temperatura a 2 metros.

```
topo <- GetTopography(360-67.5, 360-55, -24, -39, 0.25)
# topo[, lon := ConvertLongitude(lon)]

obs[obs.id == 83073] %>%
  ggplot(aes(lon, lat)) +
  geom_contour(data = topo, aes(z = h), color = "darkgray") +
  geom_point(aes(color = obs), alpha = 0.5) +
  scale_color_viridis_c() +
  facet_wrap(~time.slot)
```




```
obs[obs.id == 83073] %>%
  ggplot(aes(lon, lat)) +
  geom_contour(data = topo, aes(z = h), color = "darkgray") +
  geom_point(aes(color = ana.obs), alpha = 0.5) +
  scale_color_divergent() +
  facet_wrap(~time.slot)
```

Acá se ve también que en el slot 7 la diferecia analisis - observación es mucho más grande que para el resto. El analisis sobreestima la observación es la mayoría de los casos.

```
obs2 <- read.obs.asim('obs_20181120_01_asimiladas.dat')

obs2[, .N, keyby = .(time.slot, obs.id)] %>%
  dcast(time.slot ~ obs.id, value.var = 'N') %>%
  knitr::kable()
```

time.slot	14593	82819	82820	83073	83331
1	115	256	256	427	424
2	14	134	134	266	266
3	44	188	188	355	352
4	50	191	191	356	353
5	21	152	152	279	279
6	57	207	207	375	371
7	138	306	305	476	475

Leo las observaciones del tiempo siguiente (20181120_01) para ver si las del primer slot coinciden con las del último slot del tiempo anterior (20181120_00) y al menos en cantidad no se parecen. Sigo sin entender como se dividen los slots en cada ventana temporal.

Eso también me genera problemas a la hora de comparar las observaciones con el pronóstico. Si comparo el pronóstico que verifica a las 00 del 20181120, algunas observaciones van a ser casi una hora viejas. No parece una buena comparación. Por eso estoy pensando que capaz es mejor organizar las observaciones en ventanas centradas en la hora de verificación de cada pronóstico.

Revisando este nuevo tiempo veo que ana.obs es cero en muchísisimas observaciones, básicamente todas las que no están en el slot 7.

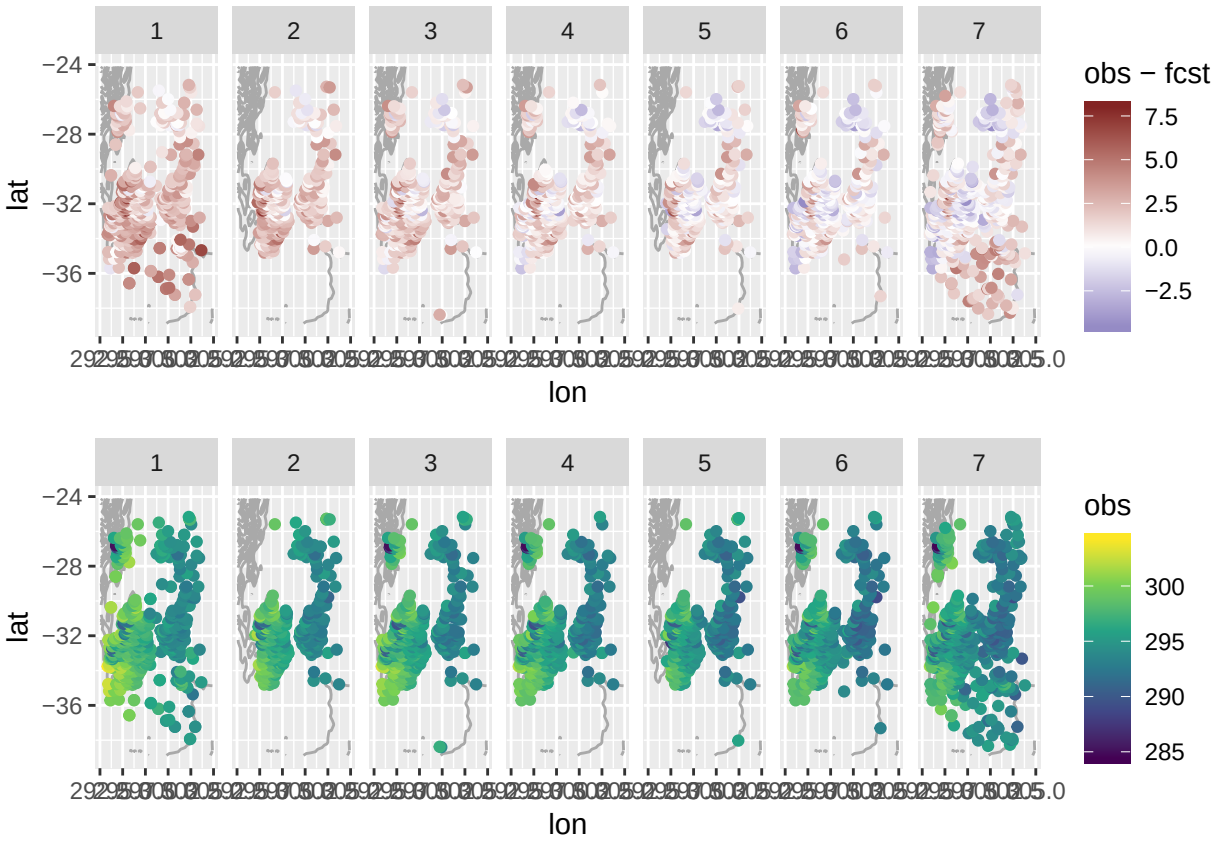
Forecast: lectura, interpolación, etc

Para interpolar el pronóstico a cada observación uso el paqute `interp` que parece funcionar de maravillas.

Mientras resuelvo todo lo anterior, voy a comparar derecho con lo que hay. Por ahora leo solo temperatura a 2 metros de 1 miembro del ensable. Luego veremos como nos las arreglamos para leer todo junto.

```
file.nc <- 'NPP_2018-11-20_00_FC00.nc'

# Leo
t2 <- ReadNetCDF(file.nc, vars = c("XLONG", "XLAT", "T2"), subset = list(ens = 1))
t2[, XLONG := ConvertLongitude(XLONG)]

# Solo t2m
t2.asim <- obs[obs.id == 83073]
t2.asim$fcst <- with(t2, interp(XLONG, XLAT, T2, output = 'points',
                                xo = t2.asim$lon, yo = t2.asim$lat))[['z']]
```

```
p1 <- ggplot(t2.asim, aes(lon, lat)) +
  geom_contour(data = topo, aes(z = h), color = "darkgray") +
  geom_point(aes(color = obs-fcst)) +
  scale_color_divergent() +
  facet_wrap(~time.slot, ncol = 7)

p2 <- ggplot(t2.asim, aes(lon, lat)) +
  geom_contour(data = topo, aes(z = h), color = "darkgray") +
  geom_point(aes(color = obs)) +
  scale_color_viridis_c() +
  facet_wrap(~time.slot, ncol = 7)

p1/p2
```


1
2
3
4
5
6
7
−24
−28
−32
−36
lat
lon
obs − fcst
7.5
5.0
2.5
0.0
−2.5
obs
300
295
290
285