Universidad Carlos III de Madrid

Aprendizaje Automático, Grado en Estadística y Empresa

Práctica 1: : Predicción de la radiación solar en plantas fotovoltaicas

Chiara Magroni, Jorge Salas y Marc Pastor

1 Objetivo

Crear modelos de regresión para predecir la radiación solar, a partir de predicciones de variables meteorológicas.

2 Problema

A finales del 2015 se estimó que el 23,7% de la energía eléctrica mundial se produjo mediante fuentes de energía renovables.

Uno de los mayores problemas que tiene la energía solar es su variabilidad e incertidumbre. Las empresas productoras necesitan una estimación diaria para las siguientes 24 horas. Por ello es importante tener una predicción lo más acertada posible.

Nuestro principal objetivo será predecir la radiación solar diaria en una planta solar de Oklahoma a partir de predicciones de variables meteorológicas del día anterior, usando MLR.

3 Datos

Tenemos el archivo número 21. Vamos a leer los datos `disp21.rds`, que contienen todos los datos de entrenamiento y validación y luego el conjunto `comp21.rds`, que contienen todos los datos para realizar las predicciones de competición.

```
library(tidyverse)
library(mlr3)
library(mlr3verse)
library(mlr3learners)
library(mlr3extralearners)
library(mlr3filters)
library(paradox)
library(mlr3tuning)
library(mlr3viz)
library(mlr3pipelines)
library(NADIA)
datos_disp <- readRDS("./datos/disp_21.rds")
datos_compet <- readRDS("./datos/compet_21.rds")
```

El *dataset* `datos_disp` consta de un conjunto de datos diarios durante un periodo de 12 años (1994-2006), es decir, de un tamaño muestral $n = 12 \cdot 365 = 4380$. Además, consta de 75 variables predictoras, que consisten en 15 variables meteorológicas medidas durante 5 momentos del dia, con lo que el número de predictores es: $p = 15 \cdot 5 = 75$ y una variable respuesta, que es la radiación solar diaria de una cierta planta solar de Oklahoma.

Las variables del conjunto son las siguientes (ver siguiente página):

Variable	**Tipo**	**Descripción**	**Unidades**
apcp_sfc	Numérica continua	Precipitación acumulada en la superficie durante 3 horas.	kg/m^2
dlwrf_sfc	Numérica continua	Promedio de flujo radiativo de onda larga en la superficie.	W/m^2
dswrf_sfc	Numérica continua	Promedio de flujo radiativo de onda corta en la superficie.	W/m^2
pres_msl	Numérica continua	Presión del aire al nivel medio del mar.	Pa
pwat_eatm	Numérica continua	Agua precipitable sobre toda la atmósfera.	kg/m^2
spfh_2m	Numérica continua	Humedad específica a 2 m sobre el suelo.	$kg/kg - l$
tcdc_eatm	Numérica continua	Cobertura total de nubes sobre toda la profundidad de la atmósfera.	%
tcolc_eatm	Numérica continua	Condensado total integrado en la columna sobre toda la atmósfera.	kg/m^2
tmax_2m	Numérica continua	Temperatura máxima en las últimas 3 horas a 2 m sobre el suelo.	K
tmin_2m	Numérica continua	Temperatura mínima en las últimas 3 horas a 2 m sobre el suelo.	K
tmp_2m	Numérica continua	Temperatura actual a 2 m sobre el suelo.	K
tmp_sfc	Numérica continua	Temperatura de la superficie.	K
ulwrf_sfc	Numérica continua	Radiación ascendente de onda larga en la superficie.	W/m^2
ulwrf_tatm	Numérica continua	Radiación ascendente de onda larga en la parte superior de la atmósfera.	W/m^2
uswrf_sfc	Numérica continua	Radiación ascendente de onda corta en la superficie.	W/m^2

4 Análisis Exploratorio de Datos

Utilizaremos el paquete `mlr3`, por lo que antes de llevar a cabo las particiones de entrenamiento y validación debemos crear una *task*:

```
practica_1_task <- as_task_regr(datos_disp, target = "salida", id =
↪  "radiacion")
practica_1_task$print()
```

```
## <TaskRegr:radiacion> (4380 x 76)
## * Target: salida
## * Properties: -
## * Features (75):
##   - ord (39): apcp_sf1_1, apcp_sf2_1, apcp_sf3_1, apcp_sf5_1,
##     dlwrf_s2_1, dlwrf_s3_1, dlwrf_s4_1, dlwrf_s5_1, dswrf_s4_1,
##     pres_ms1_1, pwat_ea1_1, pwat_ea3_1, spfh_2m2_1, spfh_2m3_1,
##     spfh_2m4_1, tcdc_ea1_1, tcdc_ea2_1, tcdc_ea3_1, tcdc_ea4_1,
##     tcolc_e2_1, tcolc_e3_1, tcolc_e5_1, tmax_2m2_1, tmin_2m1_1,
##     tmin_2m2_1, tmin_2m3_1, tmp_sfc2_1, tmp_sfc4_1, tmp_sfc5_1,
##     ulwrf_s1_1, ulwrf_s2_1, ulwrf_s3_1, ulwrf_s4_1, ulwrf_t4_1,
##     ulwrf_t5_1, uswrf_s1_1, uswrf_s3_1, uswrf_s4_1, uswrf_s5_1
##   - dbl (34): dlwrf_s1_1, dswrf_s1_1, dswrf_s2_1, dswrf_s3_1,
##     dswrf_s5_1, pres_ms2_1, pres_ms3_1, pres_ms4_1, pres_ms5_1,
##     pwat_ea2_1, pwat_ea4_1, pwat_ea5_1, spfh_2m1_1, spfh_2m5_1,
##     tcdc_ea5_1, tcolc_e1_1, tcolc_e4_1, tmax_2m1_1, tmax_2m4_1,
##     tmax_2m5_1, tmin_2m4_1, tmin_2m5_1, tmp_2m_1_1, tmp_2m_2_1,
##     tmp_2m_3_1, tmp_2m_4_1, tmp_2m_5_1, tmp_sfc1_1, tmp_sfc3_1,
##     ulwrf_s5_1, ulwrf_t1_1, ulwrf_t2_1, ulwrf_t3_1, uswrf_s2_1
##   - chr (2): apcp_sf4_1, tmax_2m3_1
```

```
practica_1_task$feature_types
```

```
##             id      type
##  1: apcp_sf1_1   ordered
##  2: apcp_sf2_1   ordered
##  3: apcp_sf3_1   ordered
##  4: apcp_sf4_1 character
##  5: apcp_sf5_1   ordered
##  6: dlwrf_s1_1   numeric
##  7: dlwrf_s2_1   ordered
##  8: dlwrf_s3_1   ordered
##  9: dlwrf_s4_1   ordered
## 10: dlwrf_s5_1   ordered
## 11: dswrf_s1_1   numeric
## 12: dswrf_s2_1   numeric
## 13: dswrf_s3_1   numeric
## 14: dswrf_s4_1   ordered
## 15: dswrf_s5_1   numeric
## 16: pres_ms1_1   ordered
## 17: pres_ms2_1   numeric
```

```
## 18: pres_ms3_1   numeric
## 19: pres_ms4_1   numeric
## 20: pres_ms5_1   numeric
## 21: pwat_ea1_1   ordered
## 22: pwat_ea2_1   numeric
## 23: pwat_ea3_1   ordered
## 24: pwat_ea4_1   numeric
## 25: pwat_ea5_1   numeric
## 26: spfh_2m1_1   numeric
## 27: spfh_2m2_1   ordered
## 28: spfh_2m3_1   ordered
## 29: spfh_2m4_1   ordered
## 30: spfh_2m5_1   numeric
## 31: tcdc_ea1_1   ordered
## 32: tcdc_ea2_1   ordered
## 33: tcdc_ea3_1   ordered
## 34: tcdc_ea4_1   ordered
## 35: tcdc_ea5_1   numeric
## 36: tcolc_e1_1   numeric
## 37: tcolc_e2_1   ordered
## 38: tcolc_e3_1   ordered
## 39: tcolc_e4_1   numeric
## 40: tcolc_e5_1   ordered
## 41: tmax_2m1_1   numeric
## 42: tmax_2m2_1   ordered
## 43: tmax_2m3_1 character
## 44: tmax_2m4_1   numeric
## 45: tmax_2m5_1   numeric
## 46: tmin_2m1_1   ordered
## 47: tmin_2m2_1   ordered
## 48: tmin_2m3_1   ordered
## 49: tmin_2m4_1   numeric
## 50: tmin_2m5_1   numeric
## 51: tmp_2m_1_1   numeric
## 52: tmp_2m_2_1   numeric
## 53: tmp_2m_3_1   numeric
## 54: tmp_2m_4_1   numeric
## 55: tmp_2m_5_1   numeric
## 56: tmp_sfc1_1   numeric
## 57: tmp_sfc2_1   ordered
## 58: tmp_sfc3_1   numeric
## 59: tmp_sfc4_1   ordered
## 60: tmp_sfc5_1   ordered
## 61: ulwrf_s1_1   ordered
## 62: ulwrf_s2_1   ordered
## 63: ulwrf_s3_1   ordered
## 64: ulwrf_s4_1   ordered
## 65: ulwrf_s5_1   numeric
## 66: ulwrf_t1_1   numeric
```

```
## 67: ulwrf_t2_1   numeric
## 68: ulwrf_t3_1   numeric
## 69: ulwrf_t4_1   ordered
## 70: ulwrf_t5_1   ordered
## 71: uswrf_s1_1   ordered
## 72: uswrf_s2_1   numeric
## 73: uswrf_s3_1   ordered
## 74: uswrf_s4_1   ordered
## 75: uswrf_s5_1   ordered
##             id      type
```

```
practica_1_task$data()
```

```
##         salida apcp_sf1_1 apcp_sf2_1 apcp_sf3_1 apcp_sf4_1 apcp_sf5_1
##    1: 11500500        low        low        low        red        low
##    2:  6439800        low        low        low        red        low
##    3:  8325900        low        low        low        red        low
##    4:  6727800        low        low        low        red        low
##    5: 10879500        low        low        low       <NA>        low
##   ---
## 4376:  7253100        low        low        low        red        low
## 4377: 11737500        low        low        low        red        low
## 4378: 11441700        low        low        low        red        low
## 4379: 11361600        low        low        low        red        low
## 4380: 10737300        low        low        low        red        low
##       dlwrf_s1_1 dlwrf_s2_1 dlwrf_s3_1 dlwrf_s4_1 dlwrf_s5_1 dswrf_s1_1
##    1:   259.4927     medium     medium        low        low         NA
##    2:   254.7259     medium     medium     medium     medium         NA
##    3:   215.2800        low        low        low        low         NA
##    4:   240.4085        low        low        low        low         NA
##    5:   233.4355        low        low        low     medium         NA
##   ---
## 4376:   272.1687     medium     medium     medium     medium         NA
## 4377:   248.6420        low        low        low     medium          0
## 4378:   271.8098     medium     medium     medium     medium         NA
## 4379:   266.9530     medium     medium     medium     medium         NA
## 4380:   268.7490     medium     medium     medium     medium         NA
##       dswrf_s2_1 dswrf_s3_1 dswrf_s4_1 dswrf_s5_1 pres_ms1_1 pres_ms2_1
##    1:   30.00000   210.0000     medium   320.0000     medium   102041.3
##    2:   20.00000   117.2727     medium   214.5455     medium   101248.9
##    3:   30.00000   209.0909     medium   312.3636     medium   102311.7
##    4:   29.09091   176.3636     medium         NA     medium   102495.5
##    5:   30.00000   210.0000     medium   317.2727     medium   101168.6
##   ---
## 4376:   22.72727   120.0000        low   118.1818     medium   101143.7
## 4377:   30.00000   210.0000     medium   310.0000     medium   101753.5
## 4378:   30.00000   210.0000     medium   310.0000     medium   101310.9
## 4379:   30.00000   217.2727     medium   314.5455        low   100152.8
## 4380:   30.00000   208.1818     medium   302.7273     medium   101242.0
##       pres_ms3_1 pres_ms4_1 pres_ms5_1 pwat_ea1_1 pwat_ea2_1 pwat_ea3_1
```

```
##    1:   102098.0  101947.34   102032.1        low         NA        low
##    2:   101181.5         NA   101425.3        low  12.705498        low
##    3:   102058.6  101587.11   101479.1        low   5.658528        low
##    4:   102730.9  102601.85   102543.9        low   6.878050        low
##    5:   100744.7  100253.02   100058.9        low         NA        low
##   ---
## 4376:   101187.6         NA   101333.6        low         NA        low
## 4377:   101707.5  101472.64   101494.9        low   9.426621        low
## 4378:   101203.8  100880.65   100893.1        low  11.297846        low
## 4379:   100031.2   99941.84   100200.7        low   5.927980        low
## 4380:   101234.8  101035.92   101103.1        low  13.503863        low
##       pwat_ea4_1 pwat_ea5_1  spfh_2m1_1 spfh_2m2_1 spfh_2m3_1 spfh_2m4_1
##    1:   9.909091  10.191506 0.002708102        low        low        low
##    2:  14.049953  12.424275 0.003655455        low        low        low
##    3:   7.963636  10.400000 0.001867386        low        low        low
##    4:   6.104540   6.566383 0.002950513        low        low        low
##    5:  10.284057  10.302160 0.003106017        low        low        low
##   ---
## 4376:  15.269342  12.589538 0.004837479        low        low        low
## 4377:  10.458240  11.079005 0.004265351        low        low        low
## 4378:   8.119961         NA 0.002376976        low        low        low
## 4379:  11.310755         NA 0.002362970        low        low        low
## 4380:  12.641978  11.951608 0.004738145        low        low        low
##       spfh_2m5_1 tcdc_ea1_1 tcdc_ea2_1 tcdc_ea3_1 tcdc_ea4_1  tcdc_ea5_1
##    1:         NA        low        low        low        low 0.003636364
##    2:         NA        low        low        low        low 0.139090910
##    3: 0.002532765        low        low        low        low 0.092727272
##    4: 0.002662536        low        low        low        low 0.025454545
##    5: 0.003817879        low        low        low        low 0.112727272
##   ---
## 4376: 0.004724741        low        low        low        low 0.020909091
## 4377: 0.004446380        low        low        low        low 0.000000000
## 4378: 0.003679559        low        low        low        low 0.000000000
## 4379: 0.005257790        low        low        low        low 0.029090909
## 4380: 0.004289660        low        low        low        low 0.005454545
##         tcolc_e1_1 tcolc_e2_1 tcolc_e3_1 tcolc_e4_1 tcolc_e5_1 tmax_2m1_1
##    1: 0.0036909091        low        low         NA        low   281.0080
##    2: 0.1630727283        low        low         NA        low   278.1314
##    3: 0.0002454545        low        low 0.02895455        low   272.8655
##    4: 0.0057727273        low        low         NA        low   277.7506
##    5: 0.0007727273        low        low         NA        low   272.9584
##   ---
## 4376: 0.0189090909        low        low         NA        low   283.4169
## 4377: 0.0043363637        low        low         NA        low   278.6881
## 4378: 0.0267454547        low        low         NA        low   281.2797
## 4379: 0.0034545455        low        low         NA        low   285.3169
## 4380: 0.0124363638        low        low         NA        low   283.7364
##       tmax_2m2_1 tmax_2m3_1 tmax_2m4_1 tmax_2m5_1 tmin_2m1_1 tmin_2m2_1
##    1:     medium       blue   284.9180   284.9222     medium     medium
```

```
##    2:     medium       blue         NA   286.1161     medium     medium
##    3:        low        red   278.0504   278.3753     medium     medium
##    4:     medium       blue   277.7049   277.6902     medium     medium
##    5:     medium       blue   287.0893   287.2561     medium     medium
##   ---
## 4376:     medium       blue   284.2229   284.2634     medium     medium
## 4377:     medium       blue   286.1806   286.1856     medium     medium
## 4378:     medium       blue   292.2404   292.2391     medium     medium
## 4379:     medium       blue   293.5498   293.5493     medium       high
## 4380:     medium       blue         NA   289.7636     medium     medium
##       tmin_2m3_1 tmin_2m4_1 tmin_2m5_1 tmp_2m_1_1 tmp_2m_2_1 tmp_2m_3_1
##    1:     medium   284.2279   279.5351   278.7674   278.8486   284.1882
##    2:     medium   283.3834   281.3593         NA   278.2431   283.2228
##    3:     medium   274.7036         NA   268.4732   269.1170   274.5390
##    4:     medium   275.9884   272.0448         NA   272.8366   275.8596
##    5:     medium   283.0985   283.0791   272.9694   274.5721   282.9358
##   ---
## 4376:     medium   283.4188   281.7849   281.4359   282.1464   284.0906
## 4377:     medium   283.7693   278.9074   276.5039   277.5079   283.6476
## 4378:     medium   288.9195         NA   280.2320   281.1184   288.7431
## 4379:       high   292.1280   286.7598         NA   284.4489   292.6151
## 4380:     medium   287.9395         NA   280.6621   281.2869   287.8232
##       tmp_2m_4_1 tmp_2m_5_1 tmp_sfc1_1 tmp_sfc2_1 tmp_sfc3_1 tmp_sfc4_1
##    1:         NA   279.5372          0     medium   287.5540     medium
##    2:         NA   281.3550          0     medium   285.7217     medium
##    3:         NA   275.2895          0        low   281.0873     medium
##    4:         NA   272.0601          0     medium   280.3369     medium
##    5:         NA   284.8116          0     medium   285.2896     medium
##   ---
## 4376:         NA   281.7984          0     medium   285.6688     medium
## 4377:         NA   278.9101          0     medium   288.7767     medium
## 4378:         NA   287.0784         NA     medium   292.0211     medium
## 4379:         NA   286.7714          0     medium   294.8553     medium
## 4380:         NA   284.9988          0     medium   291.5955     medium
##       tmp_sfc5_1 ulwrf_s1_1 ulwrf_s2_1 ulwrf_s3_1 ulwrf_s4_1 ulwrf_s5_1
##    1:     medium     medium     medium     medium     medium   375.4630
##    2:     medium     medium     medium     medium     medium   374.4102
##    3:     medium        low        low        low        low   347.0421
##    4:        low     medium     medium        low        low   343.8910
##    5:     medium     medium     medium     medium     medium   379.4098
##   ---
## 4376:     medium     medium     medium     medium     medium   367.6883
## 4377:     medium     medium     medium     medium     medium   381.9757
## 4378:     medium     medium     medium     medium     medium   407.0642
## 4379:     medium     medium     medium     medium     medium   413.2347
## 4380:     medium     medium     medium     medium     medium   399.8784
##       ulwrf_t1_1 ulwrf_t2_1 ulwrf_t3_1 ulwrf_t4_1 ulwrf_t5_1 uswrf_s1_1
##    1:   230.5168   249.2317   251.0330       high     medium        low
##    2:   224.9276   201.0946   204.2091     medium     medium        low
```

```
##    3:   234.5647   229.2833   230.3898     medium     medium        low
##    4:   241.8758   237.4622   238.5804     medium     medium        low
##    5:   232.5508   231.5248   236.0437     medium     medium        low
##   ---
## 4376:   246.8123   222.2575   213.3253     medium     medium        low
## 4377:   243.2275         NA   245.9714       high       high        low
## 4378:   235.4938         NA   244.7357       high       high        low
## 4379:   269.5006   255.8891   261.7086       high       high        low
## 4380:   247.7985         NA   246.3380     medium     medium        low
##       uswrf_s2_1 uswrf_s3_1 uswrf_s4_1 uswrf_s5_1
##    1:          0        low        low        low
##    2:          0        low        low        low
##    3:          0        low        low        low
##    4:          0        low        low        low
##    5:          0        low        low        low
##   ---
## 4376:          0        low        low        low
## 4377:          0        low        low        low
## 4378:          0        low        low        low
## 4379:          0        low        low        low
## 4380:          0        low        low        low
```

Realizamos el análisis exploratorio mediante la librería `skimr`:

```
library(skimr)
skim_exploratorio <- skim(practica_1_task$data())
skim_exploratorio %>% filter(skim_type == "character")
```

Table 2: Data summary

Name	practica_1_task$data()
Number of rows	4380
Number of columns	76
Key	NULL
Column type frequency:	
character	2
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
apcp_sf4_1	526	0.88	3	5	0	3	0
tmax_2m3_1	438	0.90	3	5	0	3	0

```
skim_exploratorio %>% filter(skim_type == "factor")
```

Table 4: Data summary

Name	practica_1_task$data()
Number of rows	4380
Number of columns	76
Key	NULL
Column type frequency:	
factor	39
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
apcp_sf1_1	0	1	TRUE	3	low: 4325, med: 45, hig: 10
apcp_sf2_1	0	1	TRUE	3	low: 4361, med: 15, hig: 4
apcp_sf3_1	0	1	TRUE	3	low: 4360, med: 18, hig: 2
apcp_sf5_1	0	1	TRUE	3	low: 4340, med: 35, hig: 5
dlwrf_s2_1	0	1	TRUE	3	med: 1861, hig: 1858, low: 661
dlwrf_s3_1	0	1	TRUE	3	med: 1879, hig: 1818, low: 683
dlwrf_s4_1	0	1	TRUE	3	med: 1892, hig: 1777, low: 711
dlwrf_s5_1	0	1	TRUE	3	med: 1914, hig: 1812, low: 654
dswrf_s4_1	0	1	TRUE	3	hig: 2404, med: 1553, low: 423
pres_ms1_1	0	1	TRUE	3	med: 3385, low: 627, hig: 368
pwat_ea1_1	0	1	TRUE	3	low: 2420, med: 1650, hig: 310
pwat_ea3_1	0	1	TRUE	3	low: 2351, med: 1676, hig: 353
spfh_2m2_1	0	1	TRUE	3	low: 2064, med: 1332, hig: 984
spfh_2m3_1	0	1	TRUE	3	low: 2008, med: 1383, hig: 989
spfh_2m4_1	0	1	TRUE	3	low: 1978, med: 1473, hig: 929
tcdc_ea1_1	0	1	TRUE	3	low: 4316, med: 54, hig: 10
tcdc_ea2_1	0	1	TRUE	3	low: 4346, med: 32, hig: 2
tcdc_ea3_1	0	1	TRUE	3	low: 4322, med: 51, hig: 7
tcdc_ea4_1	0	1	TRUE	3	low: 4354, med: 25, hig: 1
tcolc_e2_1	0	1	TRUE	3	low: 4346, med: 32, hig: 2
tcolc_e3_1	0	1	TRUE	3	low: 4322, med: 51, hig: 7
tcolc_e5_1	0	1	TRUE	3	low: 4326, med: 47, hig: 7

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
tmax_2m2_1	0	1	TRUE	3	hig: 2325, med: 1838, low: 217
tmin_2m1_1	0	1	TRUE	3	hig: 2280, med: 1944, low: 156
tmin_2m2_1	0	1	TRUE	3	hig: 2455, med: 1816, low: 109
tmin_2m3_1	0	1	TRUE	3	hig: 2452, med: 1820, low: 108
tmp_sfc2_1	0	1	TRUE	3	hig: 2250, med: 1895, low: 235
tmp_sfc4_1	0	1	TRUE	3	hig: 2070, med: 2012, low: 298
tmp_sfc5_1	0	1	TRUE	3	hig: 2260, med: 1897, low: 223
ulwrf_s1_1	0	1	TRUE	3	med: 2215, hig: 1810, low: 355
ulwrf_s2_1	0	1	TRUE	3	hig: 2066, med: 2014, low: 300
ulwrf_s3_1	0	1	TRUE	3	med: 2074, hig: 1843, low: 463
ulwrf_s4_1	0	1	TRUE	3	med: 2186, hig: 1711, low: 483
ulwrf_t4_1	0	1	TRUE	3	hig: 2901, med: 1191, low: 288
ulwrf_t5_1	0	1	TRUE	3	hig: 2755, med: 1340, low: 285
uswrf_s1_1	0	1	TRUE	3	low: 3929, hig: 413, med: 38
uswrf_s3_1	0	1	TRUE	3	low: 2707, med: 1665, hig: 8
uswrf_s4_1	0	1	TRUE	3	low: 4172, med: 196, hig: 12
uswrf_s5_1	0	1	TRUE	3	low: 2337, med: 2030, hig: 13

```
skim_exploratorio %>% filter(skim_type == "numeric") %>%
↪  select(-numeric.hist, -numeric.p0, -numeric.p100)
```

Table 6: Data summary

Name	practica_1_task$data()
Number of rows	4380
Number of columns	76
Key	NULL
Column type frequency:	
numeric	35

Table 6: Data summary

Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p25	p50	p75
salida	0	1.00	15927415.49	7822850.61	10259475.00	15859950.00	22690500.00
dlwrf_s1_1	438	0.90	313.23	56.36	266.37	316.09	363.39
dswrf_s1_1	3854	0.12	0.08	0.37	0.00	0.00	0.00
dswrf_s2_1	438	0.90	163.87	114.60	50.00	149.09	266.30
dswrf_s3_1	0	1.00	373.82	160.11	230.00	380.86	520.25
dswrf_s5_1	613	0.86	502.23	194.30	336.45	517.64	686.09
pres_ms2_1	0	1.00	101769.68	761.92	101287.86	101708.63	102209.61
pres_ms3_1	526	0.88	101732.80	747.42	101260.58	101683.36	102152.54
pres_ms4_1	438	0.90	101532.16	735.08	101069.34	101482.63	101952.07
pres_ms5_1	0	1.00	101496.19	742.97	101025.59	101432.67	101929.11
pwat_ea2_1	745	0.83	20.94	12.01	10.60	18.52	30.62
pwat_ea4_1	0	1.00	22.01	12.33	11.36	19.61	32.18
pwat_ea5_1	788	0.82	21.89	12.13	11.49	19.53	31.82
spfh_2m1_1	0	1.00	0.01	0.00	0.00	0.01	0.01
spfh_2m5_1	526	0.88	0.01	0.01	0.00	0.01	0.01
tcdc_ea5_1	0	1.00	0.06	0.17	0.00	0.00	0.04
tcolc_e1_1	569	0.87	0.07	0.18	0.00	0.00	0.05
tcolc_e4_1	4051	0.08	0.06	0.14	0.00	0.01	0.05
tmax_2m1_1	0	1.00	286.46	9.24	279.44	287.14	294.33
tmax_2m4_1	832	0.81	293.98	10.09	286.36	294.86	302.50
tmax_2m5_1	657	0.85	294.12	10.04	286.53	295.16	302.62
tmin_2m4_1	0	1.00	292.26	10.25	284.29	293.19	301.10
tmin_2m5_1	876	0.80	290.65	10.42	282.56	291.48	299.79
tmp_2m_1_1	613	0.86	284.39	9.03	277.45	284.98	292.36
tmp_2m_2_1	0	1.00	287.68	10.10	279.65	288.61	296.69
tmp_2m_3_1	0	1.00	292.25	10.25	284.25	293.20	301.10
tmp_2m_4_1	3898	0.11	293.56	10.57	286.06	294.63	302.62
tmp_2m_5_1	657	0.85	290.85	10.35	282.83	291.69	299.93
tmp_sfc1_1	657	0.85	0.00	0.00	0.00	0.00	0.00
tmp_sfc3_1	0	1.00	295.16	9.49	288.06	295.99	303.15
ulwrf_s5_1	0	1.00	429.20	55.91	385.66	431.90	476.72
ulwrf_t1_1	0	1.00	245.65	36.88	228.31	250.51	272.92
ulwrf_t2_1	832	0.81	245.36	37.52	228.07	250.33	274.36
ulwrf_t3_1	0	1.00	249.71	36.88	231.81	255.11	277.65
uswrf_s2_1	0	1.00	0.00	0.00	0.00	0.00	0.00

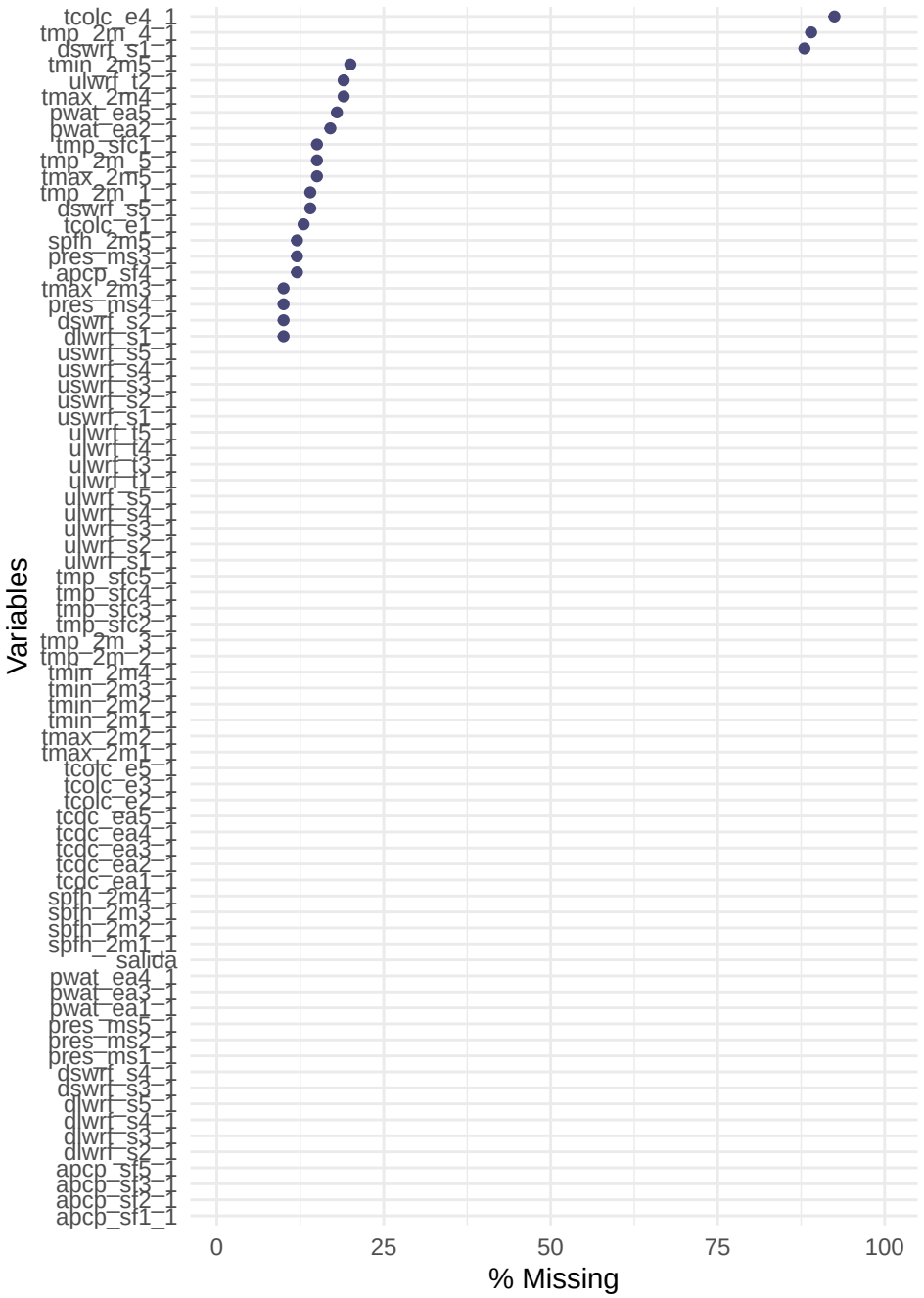
Podemos ver que hay 4380 instancias y 76 atributos, de los cuáles 35 son atributos numéricos, 39 de tipo `factor` y 2 de tipo `character`. Las variables `factor` son todas completas, las `character` tienen alrededor de un 10% de NA's y en algunas numéricas

como `dswrf_s1_1`, `tcolc_e4_1`, `tmp_2m_4_1` existe más de un 85% de porcentaje de NA's (las eliminaremos posteriormente durante el pre-proceso), mientras que las demás numéricas tienen un porcentaje mucho menor de datos faltantes.

```
(variables_numericas_muchos_NA <- skim_exploratorio %>% filter(skim_type
↪  == "numeric" & complete_rate < 0.2) %>% select(skim_variable) %>%
↪  as.data.frame() %>% as.matrix() %>% as.vector())
```

```
## [1] "dswrf_s1_1" "tcolc_e4_1" "tmp_2m_4_1"
```

```
library(naniar)
gg_miss_var(practica_1_task$data()[], show_pct = TRUE) + ylim(1, 100)
```



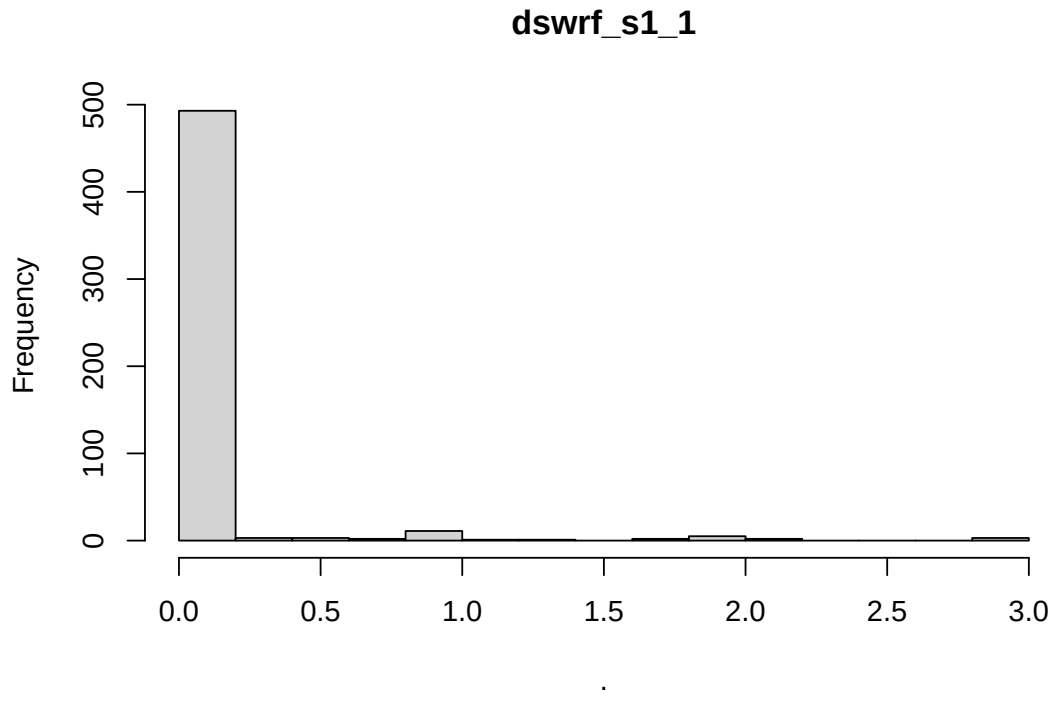
Hay que diferenciar variables que toman valores constantes de aquellas que toman valores muy pequeños, ya que en ambos casos la desviación típica es igual o muy próxima a 0, estas son: `dswrf_s1_1`, `dswrf_s2_1`, `spfh_2m1_1`, `spfh_2m5_1`, `tcdc_ea5_1`, `tcolc_e1_1`, `tcolc_e4_1`, `tmp_sfc1_1` y `uswrf_s2_1`.

```
(variables_numericas_poca_variabilidad <- skim_exploratorio %>%
↪  filter(skim_type == "numeric" & numeric.sd < 1) %>%
↪  select(skim_variable) %>% as.data.frame() %>% as.matrix() %>%
↪  as.vector())
```

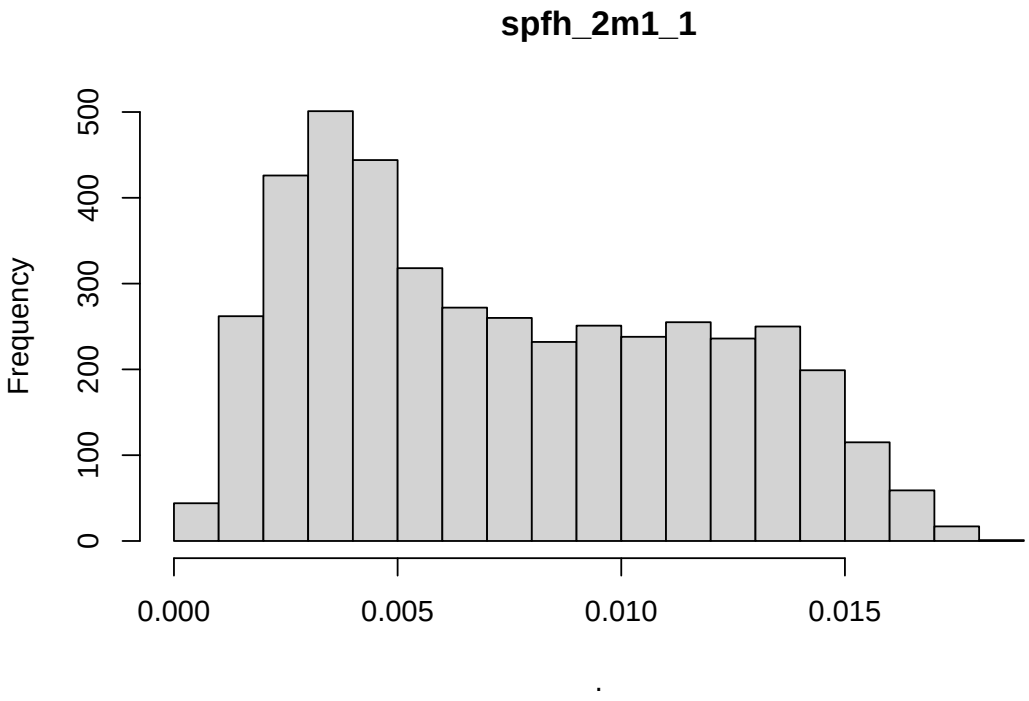
```
## [1] "dswrf_s1_1" "spfh_2m1_1" "spfh_2m5_1" "tcdc_ea5_1" "tcolc_e1_1"
## [6] "tcolc_e4_1" "tmp_sfc1_1" "uswrf_s2_1"
```

Vamos a ver si realmente son constantes o solamente es que toman valores muy pequeños y por ello la desviación típica tiende a cero. Para ello nos basaremos en los histogramas:

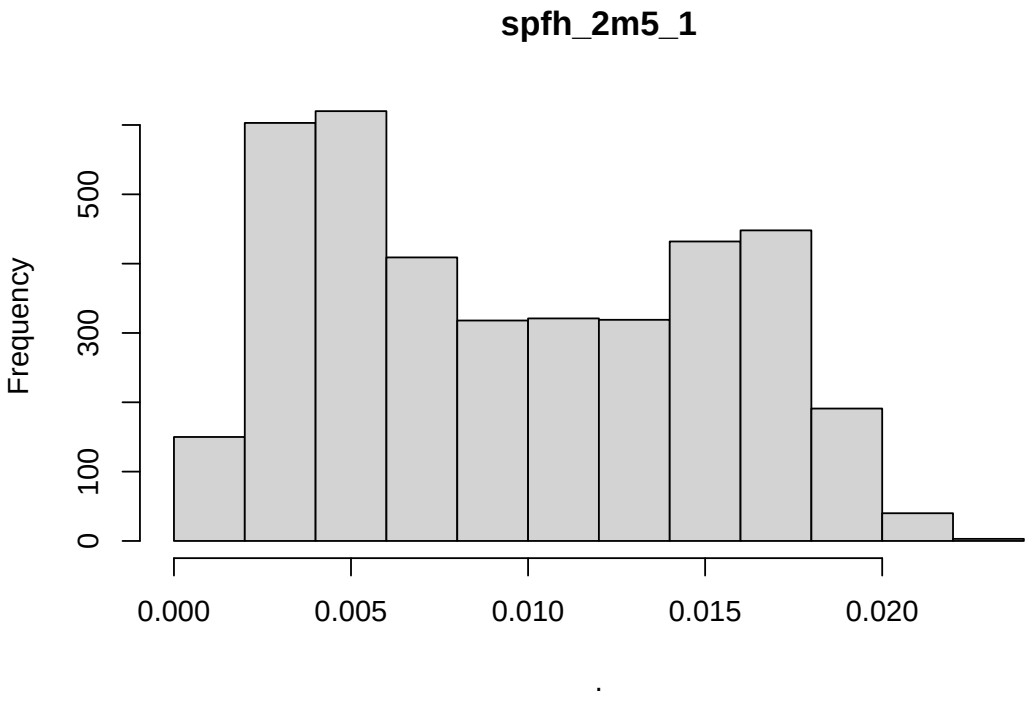
```
for(i in variables_numericas_poca_variabilidad){
        datos_disp %>% select(i) %>% as.matrix() %>% as.vector() %>%
↪  na.omit() %>% hist(main = i)
}
```



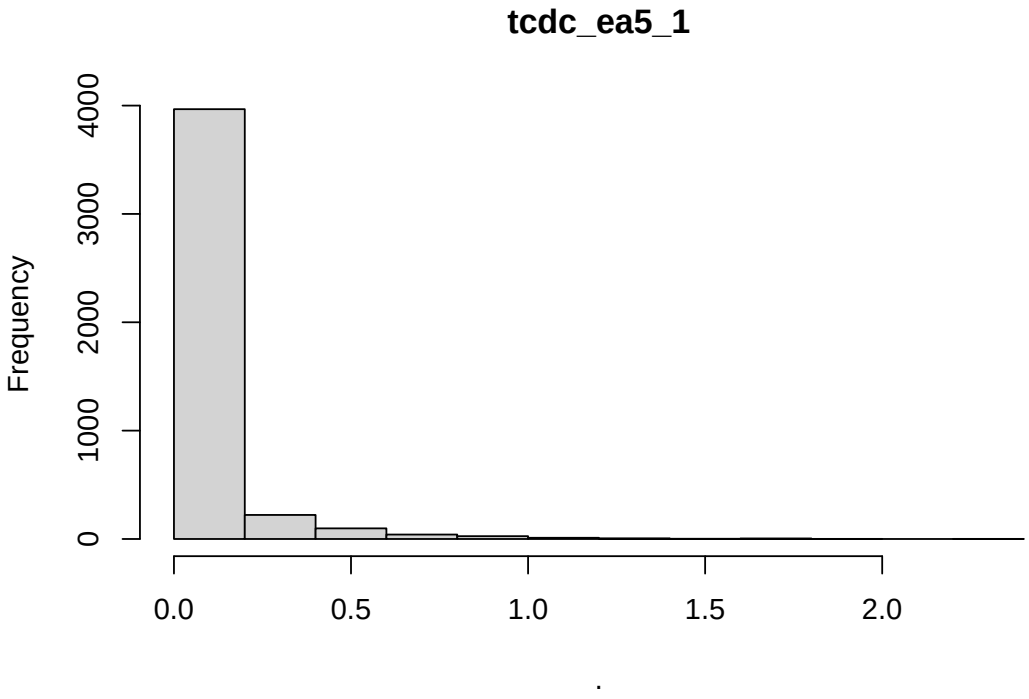
.


spfh_2m1_1
Frequency
0
100
200
300
400
500
0.000
0.005
0.010
0.015

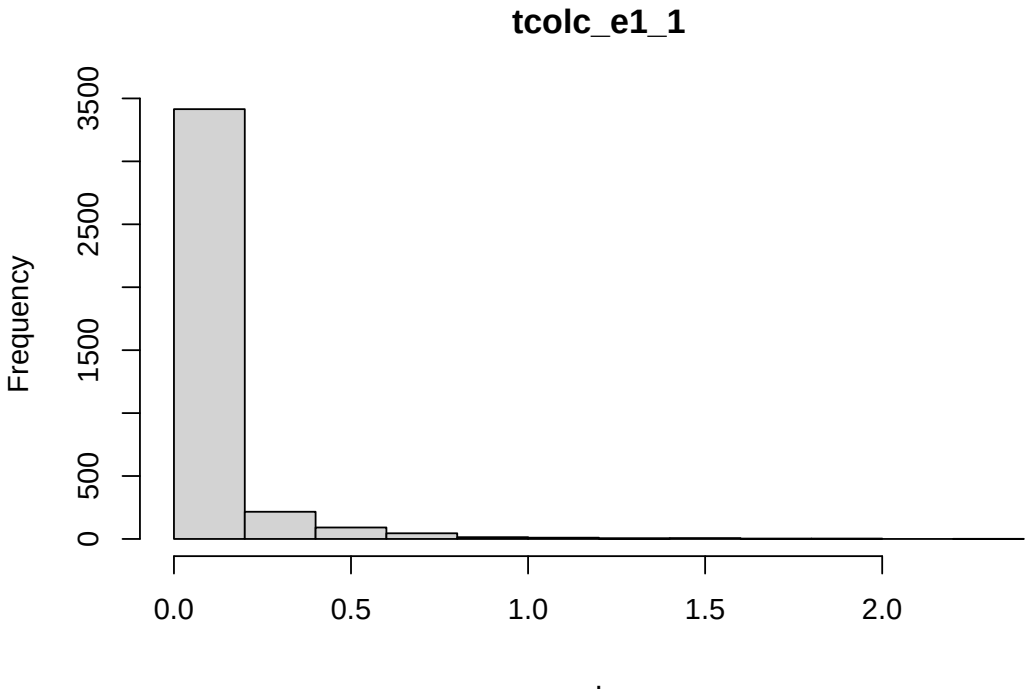
.


spfh_2m5_1
Frequency
0
100
300
500
0.000
0.005
0.010
0.015
0.020

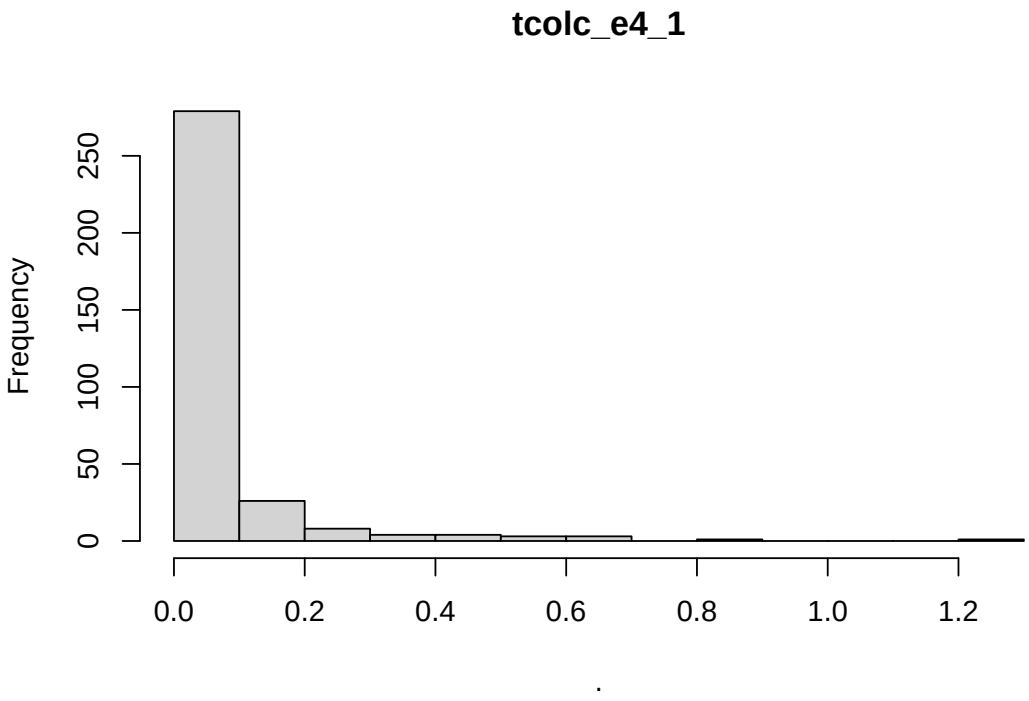
.


tcdc_ea5_1
Frequency
0
1000
2000
3000
4000
0.0
0.5
1.0
1.5
2.0

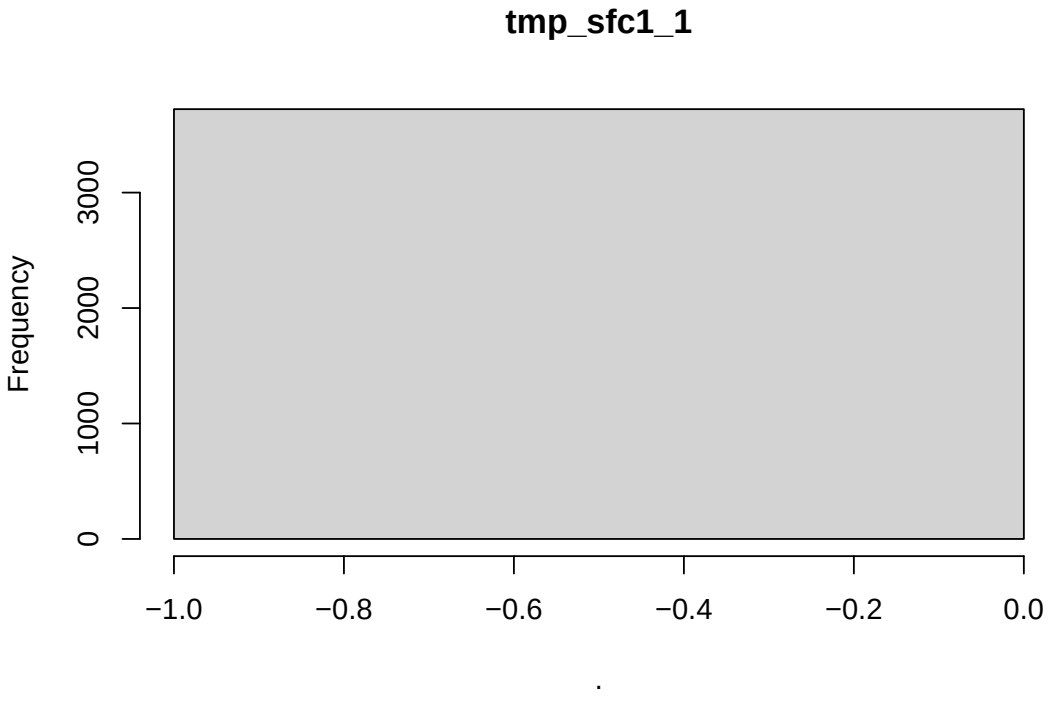
.


tcolc_e1_1
Frequency
0
500
1500
2500
3500
0.0
0.5
1.0
1.5
2.0

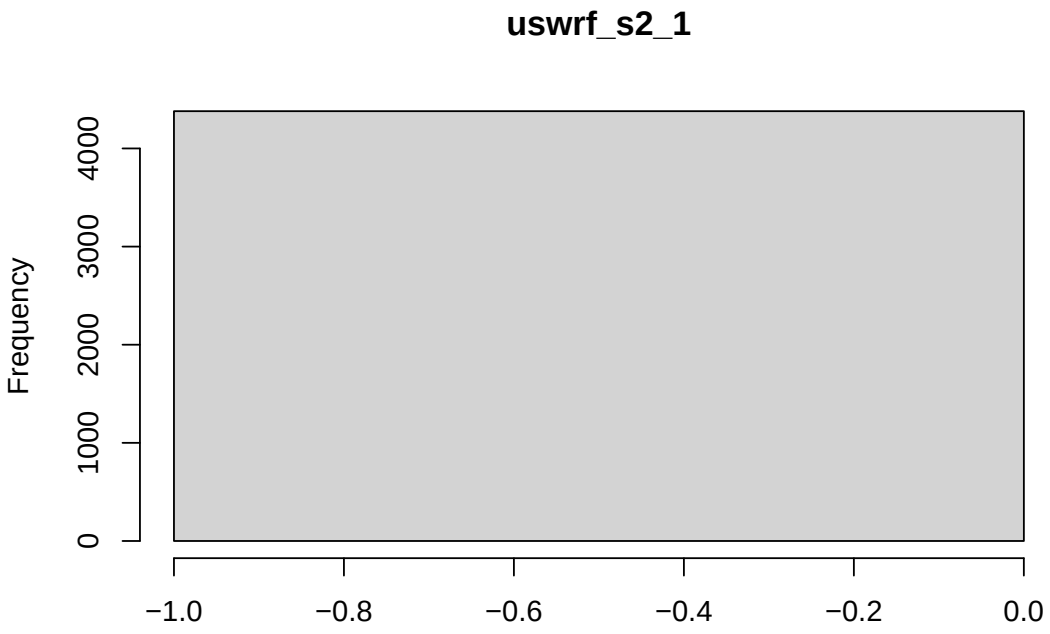
.


tcolc_e4_1
Frequency
0
50
100
150
200
250
0.0
0.2
0.4
0.6
0.8
1.0
1.2

.


tmp_sfc1_1
Frequency
0
1000
2000
3000
−1.0
−0.8
−0.6
−0.4
−0.2
0.0

.



.

Podemos ver que las variables `dswrf_s1_1`, `tcdc_ea5_1`, `tcolc_e1_1`, `tcolc_e4_1`, `tmp_sfc1_1` y `uswrf_s2_1` son efectivamente constantes, mientras que `spfh_2m1_1` y `spfh_2m5_1` tienen poca variabilidad por tener valores muy pequeños, pero sí tienen variabilidad. Con lo que guardamos dichas variables para posteriores acciones:

```
variables_numericas_constantes <- c("dswrf_s1_1", "tcdc_ea5_1",
↪  "tcolc_e1_1", "tcolc_e4_1", "tmp_sfc1_1", "uswrf_s2_1")
```

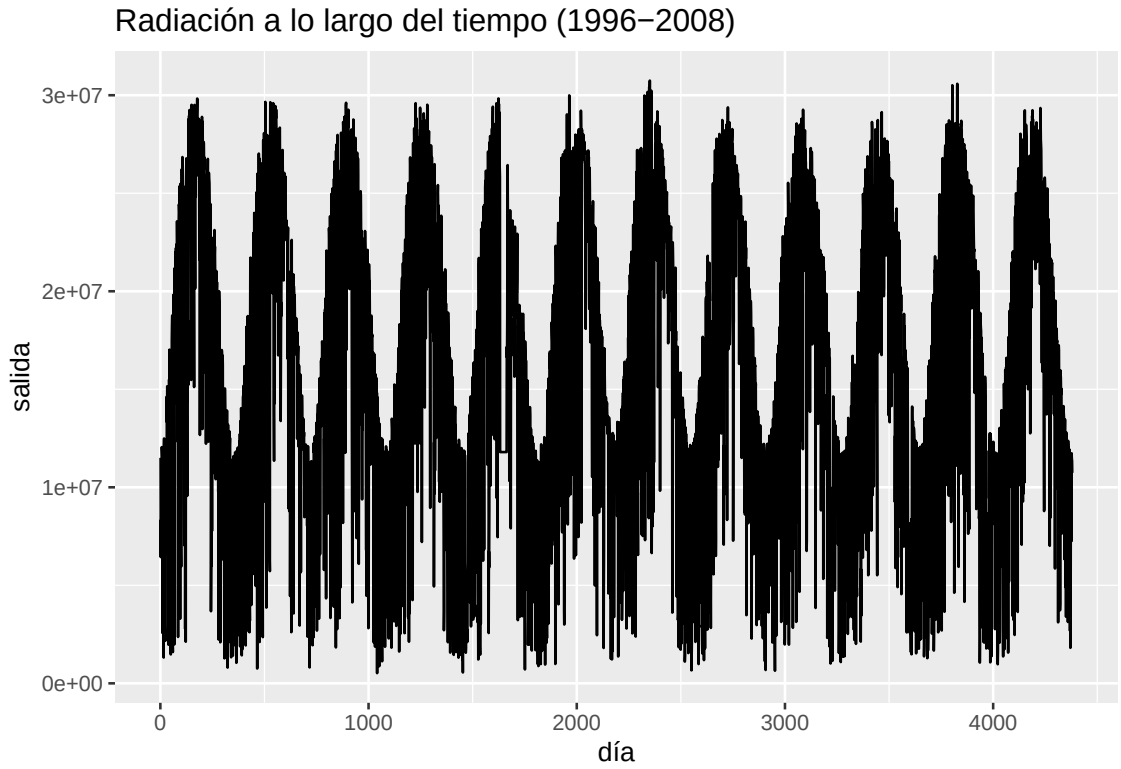
Convertiremos todos los datos que no sean numéricos en `factor`, especialmente los `character`, que pueden dar problemas en futuras aplicaciones:

```
variables_character <- skim_exploratorio %>% filter(skim_type ==
↪  "character") %>% select(skim_variable) %>% as.matrix() %>% as.vector()
datos_disp[, variables_character[1]] <- as.factor(datos_disp[,
↪  variables_character[1]])
datos_disp[, variables_character[2]] <- as.factor(datos_disp[,
↪  variables_character[2]])
practica_1_task <- as_task_regr(datos_disp, target = "salida", id =
↪  "radiacion")
```

4.1 Variable respuesta

A continuación graficaremos la variable respuesta en el tiempo:

```
vector_indicador <- 1:nrow(datos_disp)
datos_grafico_respuesta <- data.frame(indice = vector_indicador, salida =
↪  datos_disp$salida)
ggplot(data = datos_grafico_respuesta, aes(x = vector_indicador, y =
↪  salida)) + geom_line() + xlab("día") + ylab("salida") +
↪  ggtitle("Radiación a lo largo del tiempo (1996-2008)")
```



Podemos ver que existe una clara estacionalidad en la radiación captada por las placas solares, aunque el nivel de la serie parece constante en el tiempo, es decir, no se observan tendencias crecientes ni decrecientes en los datos, esto es, los niveles de radiación son parecidos año tras año.

5 Métrica: Relative Absolute Error

El error absoluto relativo (RAE en inglés) para un modelo de regresión con variable respuesta y_i puede ser definido como:

$$RAE(\hat{y}_i) = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{\sum_{i=1}^{n} |y_i - \overline{y}|}$$

dónde $\hat{y}_i$ es la predicción de la variable respuesta que hace el modelo de regresión e $\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$. Se puede interpretar como un ratio entre el error absoluto de predicción con el modelo escogido y el error absoluto para una predicción *naive* basada en la media de la respuesta $\overline{y}$. Esta medida no está definida para el caso $y_i = y \quad \forall i = 1, \ldots, n$.

Este ratio puede ser usado mediante la librería `mlr3` mediante instanciado a través del diccionario `mlr_measures` o mediante la función asociada `msr()`:

```
mlr_measures$get("regr.rae")
msr("regr.rae")
```

6 Mejor método de imputación y de escalado

En este apartado vamos a comparar distintos métodos de imputación y escalado en base a su RAE en el conjunto de testeo para un modelo de vecino más cercano con hiper-parámetros por defecto.

6.1 Eliminación de las variables que toman valores constantes y/o tienen muchos NA

Eliminaremos del modelo aquellos predictores que tienen valores constante o varianzas muy próximas a cero y también aquellos que tienen un porcentaje de NA muy elevado (véase Análisis Exploratorio de Datos):

```
datos_disp <- datos_disp %>%
↪  select(-(all_of(c(variables_numericas_constantes,
↪  variables_numericas_muchos_NA))))
datos_compet <- datos_compet %>%
↪  select(-(all_of(c(variables_numericas_constantes,
↪  variables_numericas_muchos_NA))))
practica_1_task <- as_task_regr(datos_disp, target = "salida", id =
↪  "radiacion")
```

6.2 Particiones de entrenamiento y test

A continuación dividiremos el conjunto de datos `datos_disp` en particiones de entrenamiento y testeo, correspondiendo los datos de los primeros 9 años a datos de entrenamiento, y los 3 últimos años a validación:

```
set.seed(100430509) # NIA de Marc Pastor
#source("./info/Ajuste Hiper-parámetros/ResamplingHoldoutOrder.R")
#desc_inner <- rsmp("holdoutorder", ratio = 6/9)
desc_inner <- rsmp("custom")
desc_inner$instantiate(practica_1_task,
                       train = list(1:(9*365)),
                       test = list((9*365+1):(12*365)))
id_train <- desc_inner$train_set(i = 1)
id_test <- desc_inner$test_set(i = 1)

# Se crean dos nuevas task, una con los datos de train y otra con los de
↪  test.
# Dado que se va a aplicar un filtrado, y para no alterar task_datos, se
↪  emplea
# antes del filtro el método $clone() para hacer una copia.
task_train <- practica_1_task$clone()$filter(id_train)
task_test  <- practica_1_task$clone()$filter(id_test)
```

6.3 Métodos de escalado

6.4 Métodos de escalado

6.4.1 Normalización de los datos

Eliminamos las constantes, normalizamos los datos y hacemos una codificación *one-hot* de las variables cualitativas:

```
preproc_inicial <- po("removeconstants") %>>% po("encode")
practica_1_task <- preproc_inicial$train(practica_1_task)[[1]]
```

6.5 Métodos de imputación multivariante

En esta sección usaremos distintos métodos de imputación multivariante ya que disponemos de datos multivariantes, tanto variables continuas, como categóricas, etc.

6.5.1 Imputación mediante AMELIA (*Multiple Imputation of Incomplete Multivariate Data*)

AMELIA es un procedimiento para imputar datos multivariantes. Entre sus supuestos el principal es asumir que los datos (tanto observados como no) siguen una distribución normal multivariante. Si denotamos el *dataset* de tamaño $(n \times k)$ como D, enconces esta asunción es:

$$D \sim \mathcal{N}_k(\mu, \Sigma).$$

En nuestro caso los datos no son solamente continuos, sino que hay variables categóricas y discretas, por lo que esta asunción no se va a dar. Por ello, descartaremos este procedimiento.

6.5.2 MICE: *Multiple Imputation by Chained Equations*

MICE es un método de imputación múltiple que se basa en el supuesto de que dadas las variables usadas en el proceso de imputación, los datos faltantes son MAR (*Missing At Random*), lo cuál significa que la probabilidad de que un valor sea faltante depende solo de los valores observados y no de los valores que no han sido observados. En otras palabras, después de controlar todos los datos disponibles (es decir, las variables incluidas en el modelo de imputación), cualquier dato faltante es completamente aleatorio. Implementar MICE cuando los datos no son MAR podría dar lugar a estimaciones sesgadas. De aquí en adelante, supondremos que nuestros datos son MAR.

Muchos de los modelos de imputación múltiples inicialmente desarrollados, asumen una distribución conjunta de todas las variables, por ejemplo la distribución normal, lo cuál no suele ocurrir en conjuntos de datos grandes, con decenas de variables de distintos tipos. MICE ofrece una alternativa flexible basada en modelos de regresión donde los datos faltantes se modelan en función de las variables disponibles en los datos. Esto implica que cada variable puede ser modelada en base a su distribución, por ejemplo las variables binarias con regresión logística y las continuas con regresión lineal, etc.

Procedimiento MICE

El algoritmo MICE puede ser dividido en 4 grandes pasos:

1. Se ejecuta una imputación simple, por ejemplo imputación mediante la media, para cada valor faltante en el *dataset*. Estas imputaciones sencillas pueden ser pensadas como imputaciones base.
2. Las imputaciones base para cada variable X_i vuelven a ser asignadas el valor de faltante/missing.
3. Los valores observados de la variable X_i en el paso 2 se modelan como un modelo de regresión en función del resto de variables en el *dataset*. Estos modelos de regresión operan bajo los supuestos que uno haria cuando realiza regresión logística, lineal o Poisson fuera del contexto de datos faltantes.
4. Los valores faltantes de la variable X_i son reemplazados por las predicciones (imputaciones) del modelo de regresión. Cuando X_i posteriormente se utilice en los modelos de regresión para otras variables, se utilizarán tanto los valores observados como los imputados.
5. Se repiten los pasos 2-4 para cada variable que tiene datos faltantes. El proceso para cada una de las variables constituye una iteración o ciclo. Al final de cada ciclo, todos los valores faltantes han sido sustituidos por predicciones de regresiones que representan las relaciones entre los datos observados.
6. Los pasos 2-4 se repiten para un número de ciclos, con las imputaciones siendo actualizadas en cada ciclo.

El número de ciclos a realizarse puede ser escogido por el investigador, aunque generalmente se llevan a cabo 10. La idea es que al final de los ciclos, la distribución de los parámetros que gobiernan las imputaciones (por ejemplo los coeficientes de los modelos de regresión) deben haber convergido, en el sentido de volverse estables.

```
imp <- PipeOpMice$new()
# learner
learner <- lrn('regr.kknn')

graph <- imp %>>% po(learner)

graph_learner <- GraphLearner$new(graph, id = 'mice.learner')
graph_learner$id <-  'mice.learner'
# resampling
set.seed(100430509)
knn_resample <- resample(practica_1_task, graph_learner, desc_inner)
```

```
## INFO  [13:36:20.475] [mlr3] Applying learner 'mice.learner' on task 'radiacion' (iter
```

```
## Error in value[[3L]](cond): Error in solve.default(xtx + diag(pen)): system is computa
##
## This happened PipeOp impute_mice_B's $train()
```

```
knn_rae <- knn_resample$aggregate(msr("regr.rae"))
```

```
## Error in eval(expr, envir, enclos): object 'knn_resample' not found
```

```
print(knn_rae)
```

```
## Error in print(knn_rae): object 'knn_rae' not found
```

Parece que uno de los sistemas de ecuaciones es singular, por lo que el programa no encuentra solución, y por ello el algoritmo no nos es útil.

6.5.3 missForest

Se trata de un procedimiento que *random forest* para predecir el valor de los datos faltantes:

```
imp <- PipeOpmissForest$new()
# learner
learner <- lrn('regr.kknn')

graph <- imp %>>%  learner

graph_learner <- GraphLearner$new(graph, id = 'missForest.learner')
graph_learner$id <-  'missForest.learner'
# resampling
set.seed(100430509)
knn_resample <- resample(practica_1_task, graph_learner, desc_inner)
```

```
## INFO  [13:36:39.283] [mlr3] Applying learner 'missForest.learner' on task 'radiacion'
```

```
knn_rae <- knn_resample$aggregate(msr("regr.rae"))
print(knn_rae)
```

```
##  regr.rae
## 0.4343481
```

6.5.4 Miss Ranger

Utilizaremos el algoritmo `MissRanger`, que es una versión mejorada de `MissForest` en la que se añade el emparejamiento predictivo de medias entre iteraciones de los random forest. Esto evita en primer lugar imputación con valores que no estén presentes en los datos y en segundo lugar, el emparejamiento predictivo de medias intenta incrementar la varianza de las distribuciones condicionales para alcanzar un nivel realista.

```
imp <- PipeOpmissRanger$new()
# learner
learner <- lrn('regr.kknn')

graph <- imp %>>% learner

graph_learner <- GraphLearner$new(graph, id = 'missRanger.learner')
graph_learner$id <-  'missRanger.learner'
# resampling
set.seed(100430509)
knn_resample <- resample(practica_1_task, graph_learner, desc_inner)
```

```
## INFO  [13:47:34.371] [mlr3] Applying learner 'missRanger.learner' on task 'radiacion'
```

```
knn_rae <- knn_resample$aggregate(msr("regr.rae"))
print(knn_rae)
```

```
##  regr.rae
## 0.4399763
```

7 Bibliografía

https://mlr3.mlr-org.com/reference/mlr_measures_regr.rae.html \ https://gking.harvard.edu/files/gking/files/amelia_jss.pdf \ ncbi.nlm.nih.gov/pmc/articles/PMC3074241/