12.- Exploratory Data Analysis_05_servicios_completo_v_01

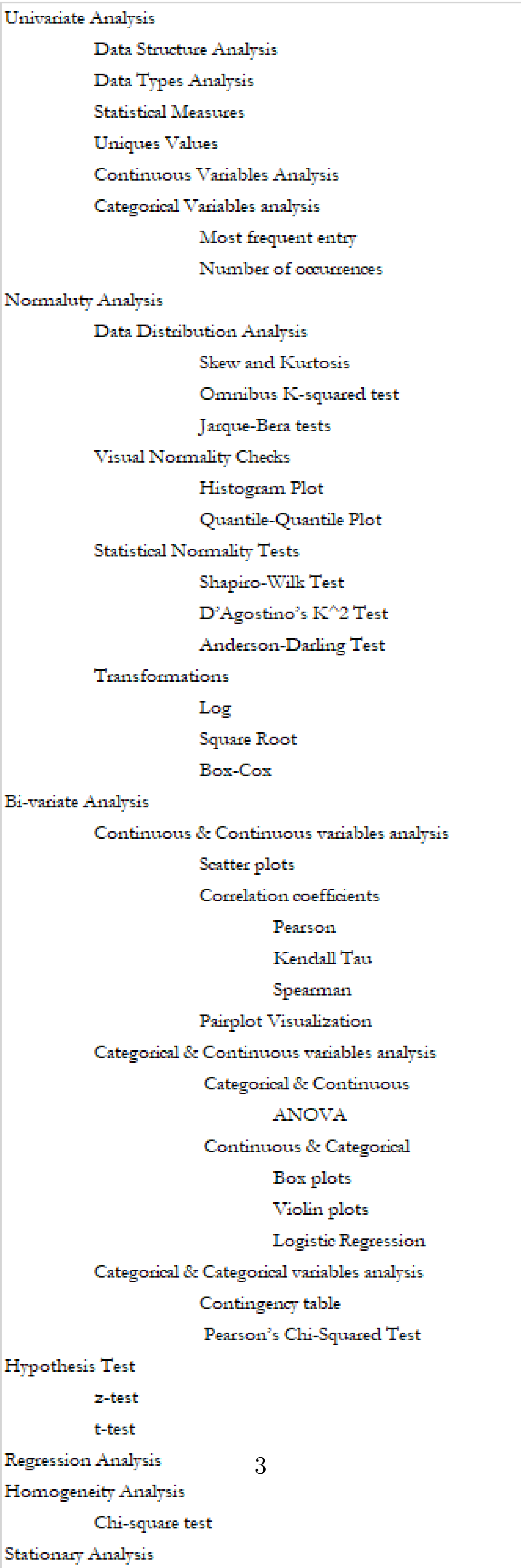
June 16, 2023

#

CUxx_Nombre del caso de uso

Citizenlab Data Science Methodology > II - Data Processing Domain *** > # 12.- EDA - Exploratory Data Analysis Analysis

0.1 Tasks



- Univariate Analysis
  - Data Structure Analysis
  - Data Types Analysis
  - Statistical Measures
  - Uniques Values
  - Continuous Variables Analysis
  - Categorical Variables analysis
    - Most frequent entry
    - Number of occurrences
- Normaluty Analysis
  - Data Distribution Analysis
    - Skew and Kurtosis
    - Omnibus K-squared test
    - Jarque-Bera tests
  - Visual Normality Checks
    - Histogram Plot
    - Quantile-Quantile Plot
  - Statistical Normality Tests
    - Shapiro-Wilk Test
    - D'Agostino's K^2 Test
    - Anderson-Darling Test
  - Transformations
    - Log
    - Square Root
    - Box-Cox
- Bi-variate Analysis
  - Continuous & Continuous variables analysis
    - Scatter plots
    - Correlation coefficients
      - Pearson
      - Kendall Tau
      - Spearman
    - Pairplot Visualization
  - Categorical & Continuous variables analysis
    - Categorical & Continuous
      - ANOVA
    - Continuous & Categorical
      - Box plots
      - Violin plots
      - Logistic Regression
  - Categorical & Categorical variables analysis
    - Contingency table
    - Pearson's Chi-Squared Test
- Hypothesis Test
  - z-test
  - t-test
- Regression Analysis
- Homogeneity Analysis
  - Chi-square test
- Stationary Analysis

0.2 File

- Input File: xxxxxxxxxx
- Output File: No aplica

0.2.1 Encoding

Con la siguiente expresión se evitan problemas con el encoding al ejecutar el notebook. Es posible que deba ser eliminada o adaptada a la máquina en la que se ejecute el código.

[1]:
```
Sys.setlocale(category = "LC_ALL", locale = "es_ES.UTF-8")
```

```
'LC_CTYPE=es_ES.UTF-
8;LC_COLLATE=es_ES.UTF-8;LC_MONETARY=es_ES.UTF-8;LC_MESSAGES=en_US.UTF-
8;LC_PAPER=es_ES.UTF-8;LC_NAME=C;LC_ADDRESS=C;LC_TELEPHONE=C;LC_MEASUREMENT
8;LC_IDENTIFICATION=C'
```

0.3 Settings

0.3.1 Libraries to use

[2]:
```
library(readr)
library(dplyr)
library(sf)
library(tidyr)
library(ggplot2)
library(summarytools)
library(GGally)
library(nortest)
library(lubridate)
```

```
Attaching package: ‘dplyr’


The following objects are masked from ‘package:stats’:

    filter, lag


The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union


Linking to GEOS 3.11.1, GDAL 3.6.2, PROJ 6.2.1; sf_use_s2() is TRUE

WARNING: different compile-time and runtime versions for GEOS found:
```

```
Linked against: 3.11.1-CAPI-1.17.1 compiled against: 3.8.0-CAPI-1.13.1

It is probably a good idea to reinstall sf, and maybe rgeos and rgdal too

Registered S3 method overwritten by 'GGally':
  method from
  +.gg   ggplot2


Attaching package: ‘lubridate’


The following objects are masked from ‘package:base’:

    date, intersect, setdiff, union
```

0.3.2 Paths

[3]:
```
iPath <- "Data/Input/"
oPath <- "Data/Output/"
```

0.4 Data Load

OPCION A: Seleccionar fichero en ventana para mayor comodidad

Data load using the {tcltk} package. Ucomment the line if using this option

[4]:
```
# file_data <- tcltk::tk_choose.files(multi = FALSE)
```

OPCION B: Especificar el nombre de archivo

[5]:
```
iFile <- "CU_34_11_05_servicios_completo.csv"
file_data <- paste0(iPath, iFile)

if(file.exists(file_data)){
    cat("Se leerán datos del archivo: ", file_data)
} else{
    warning("Cuidado: el archivo no existe.")
}
```

```
Se leerán datos del archivo:  Data/Input/CU_34_11_05_servicios_completo.csv
```

Data file to dataframe Usar la función adecuada según el formato de entrada (xlsx, csv, json, ...)

[6]:
```
data <- read_csv(file_data)
```

```
Rows: 272862 Columns: 19
  Column specification
```

```
Delimiter: ","
chr   (5): Servicio, CMUN, CDIS, CSEC, NSEC
dbl  (12): Futbol, nservicios, capacidad, tmed, prec, velmedia,
presMax, t1_…
lgl   (1): is_train
date  (1): Fecha

  Use `spec()` to retrieve the full column specification for this
data.
  Specify the column types or set `show_col_types = FALSE` to quiet
this message.
```

0.5 Data Structure

Estructura de los datos:

[7]:
```
data |> glimpse()
```

```
Rows: 272,862
Columns: 19
$ Fecha             <date> 2022-01-12, 2022-01-31, 2022-01-28,
2022-01-06, 2022…
$ Servicio          <chr> "Delivery", "Taxi", "Taxi",
"Delivery", "Delivery", "…
$ CMUN              <chr> "079", "079", "903", "079", "007",
"022", "079", "079…
$ CDIS              <chr> "14", "01", "01", "04", "04", "01",
"16", "01", "16",…
$ CSEC              <chr> "050", "048", "006", "080", "012",
"004", "041", "033…
$ Futbol            <dbl> 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0,
0, 0, 0, 0, 0, 0,…
$ nservicios        <dbl> 58, 5, 0, 14, 60, 50, 13, 4, 3, 9,
68, 12, 0, 1, 14, …
$ capacidad         <dbl> 80, 69, 56, 80, 70, 56, 80, 69, 69,
69, 80, 80, 69, 6…
$ tmed              <dbl> 7.366319, 8.823406, 7.854915,
4.226603, 4.982656, 7.2…
$ prec              <dbl> -0.009468616, 0.000000000,
0.000000000, 0.010181896, …
$ velmedia          <dbl> 1.5999961, 1.5114967, 2.2536168,
1.0279945, 1.0387037…
```

```
$ presMax          <dbl> 954.7939, 948.9795, 940.2553,
945.1884, 948.9570, 943…
$ t1_1             <dbl> 1094, 1251, 2232, 746, 1080, 2256,
692, 1270, 2229, 8…
$ t3_1             <dbl> 45.4360, 41.6091, 44.2016, 47.1729,
48.5361, 43.2877,…
$ NSEC             <chr> "Madrid - 14.050", "Madrid -
01.048", "Tres Cantos - …
$ area             <dbl> 38753.96, 15289.89, 124539.78,
89206.78, 24473.30, 34…
$ elevation        <dbl> 658, 635, 719, 710, 693, 710, 702,
635, 690, 710, 690…
$ densidad_hab_km2 <dbl> 28229.3737, 81818.7738, 17921.9842,
8362.5928, 44129.…
$ is_train         <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE,
TRUE, TRUE, TRUE,…
```

Muestra de los primeros datos:

```
[8]: data |> slice_head(n = 5)
```

A spec_tbl_df: 5 × 19

Fecha	Servicio	CMUN	CDIS	CSEC	Futbol	nservicios	capacidad	tme
<date>	<chr>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<d
2022-01-12	Delivery	079	14	050	1	58	80	7.30
2022-01-31	Taxi	079	01	048	0	5	69	8.82
2022-01-28	Taxi	903	01	006	0	0	56	7.85
2022-01-06	Delivery	079	04	080	0	14	80	4.22
2022-01-21	Delivery	007	04	012	1	60	70	4.98

Tamaño de Memoria de los datos

```
[9]: object.size(data)
```

```
40731944 bytes
```

Structure of non-numerical features

```
[10]: # Display non-numerical features
data |> select(where(~ !is.numeric(.x))) |> freq()
```

```
Variable(s) ignored: Fecha
```

1. A summarytools: 4 × 5 of type dbl

	Freq	% Valid	% Valid Cum.	% Total	% Total Cum.
Delivery	136431	50	50	50	50
Taxi	136431	50	100	50	100
<NA>	0	NA	NA	0	100
Total	272862	100	100	100	100

2. A summarytools: 173 × 5 of type dbl

	Freq	% Valid	% Valid Cum.	% Total	%
002	124	0.04544422	0.04544422	0.04544422	0.0
003	62	0.02272211	0.06816633	0.02272211	0.0
004	248	0.09088843	0.15905476	0.09088843	0.1
005	7750	2.84026358	2.99931834	2.84026358	2.9
006	4092	1.49965917	4.49897751	1.49965917	4.4
007	7316	2.68120882	7.18018632	2.68120882	7.1
008	62	0.02272211	7.20290843	0.02272211	7.2
009	558	0.20449898	7.40740741	0.20449898	7.4
010	496	0.18177687	7.58918428	0.18177687	7.5
011	62	0.02272211	7.61190638	0.02272211	7.6
012	62	0.02272211	7.63462849	0.02272211	7.6
013	1984	0.72710748	8.36173597	0.72710748	8.3
014	1612	0.59077482	8.95251079	0.59077482	8.9
015	868	0.31810952	9.27062031	0.31810952	9.2
016	62	0.02272211	9.29334242	0.02272211	9.2
017	62	0.02272211	9.31606453	0.02272211	9.3
018	124	0.04544422	9.36150875	0.04544422	9.3
019	62	0.02272211	9.38423086	0.02272211	9.3
020	62	0.02272211	9.40695297	0.02272211	9.4
021	62	0.02272211	9.42967507	0.02272211	9.4
022	1674	0.61349693	10.04317201	0.61349693	10.
023	248	0.09088843	10.13406044	0.09088843	10.
024	62	0.02272211	10.15678255	0.02272211	10.
025	62	0.02272211	10.17950466	0.02272211	10.
026	310	0.11361054	10.29311520	0.11361054	10.
027	62	0.02272211	10.31583731	0.02272211	10.
028	62	0.02272211	10.33855942	0.02272211	10.
029	62	0.02272211	10.36128153	0.02272211	10.
030	62	0.02272211	10.38400364	0.02272211	10.
031	124	0.04544422	10.42944785	0.04544422	10.
159	62	0.02272211	96.75074	0.02272211	96.
160	310	0.11361054	96.86435	0.11361054	96.
161	2480	0.90888434	97.77323	0.90888434	97.
162	124	0.04544422	97.81868	0.04544422	97.
163	62	0.02272211	97.84140	0.02272211	97.
164	124	0.04544422	97.88684	0.04544422	97.
165	62	0.02272211	97.90957	0.02272211	97.
166	62	0.02272211	97.93229	0.02272211	97.
167	372	0.13633265	98.06862	0.13633265	98.
168	62	0.02272211	98.09134	0.02272211	98.
169	62	0.02272211	98.11406	0.02272211	98.
170	124	0.04544422	98.15951	0.04544422	98.
171	248	0.09088843	98.25040	0.09088843	98.
172	372	0.13633265	98.38673	0.13633265	98.
173	62	0.02272211	98.40945	0.02272211	98.
174	62	0.02272211	98.43217	0.02272211	98.
175	62	0.02272211	98.45490	0.02272211	98.
176	496	0.18177687	98.63667	0.18177687	98.
177	496	0.18177687	98.81845	0.18177687	98.
178	62	0.02272211	98.84117	0.02272211	98.
179	62	0.02272211	98.86389	0.02272211	98.

3. A summarytools: 23 × 5 of type dbl

	Freq	% Valid	% Valid Cum.	% Total	% Tot
01	75516	27.6755283	27.67553	27.6755283	27.675
02	26474	9.7023404	37.37787	9.7023404	37.377
03	17050	6.2485799	43.62645	6.2485799	43.626
04	21204	7.7709611	51.39741	7.7709611	51.397
05	9858	3.6128153	55.01022	3.6128153	55.010
06	8990	3.2947057	58.30493	3.2947057	58.304
07	9052	3.3174279	61.62236	3.3174279	61.622
08	12400	4.5444217	66.16678	4.5444217	66.166
09	6014	2.2040445	68.37082	2.2040445	68.370
10	12400	4.5444217	72.91525	4.5444217	72.915
11	11160	4.0899796	77.00523	4.0899796	77.005
12	5704	2.0904340	79.09566	2.0904340	79.095
13	10974	4.0218132	83.11747	4.0218132	83.117
14	5456	1.9995456	85.11702	1.9995456	85.117
15	10540	3.8627585	88.97978	3.8627585	88.979
16	7564	2.7720973	91.75187	2.7720973	91.751
17	6510	2.3858214	94.13770	2.3858214	94.137
18	4154	1.5223813	95.66008	1.5223813	95.660
19	2914	1.0679391	96.72802	1.0679391	96.728
20	7006	2.5675983	99.29561	2.5675983	99.295
21	1922	0.7043854	100.00000	0.7043854	100.00
<NA>	0	NA	NA	0.0000000	100.00
Total	272862	100.0000000	100.00000	100.0000000	100.00

4. A summarytools: 223 × 5 of type dbl

	Freq	% Valid	% Valid Cum.	% Total	%
001	14260	5.2260850	5.226085	5.2260850	5.2
002	9982	3.6582595	8.884344	3.6582595	8.8
003	8618	3.1583731	12.042718	3.1583731	12.
004	7440	2.7266530	14.769371	2.7266530	14.
005	6696	2.4539877	17.223358	2.4539877	17.
006	6324	2.3176551	19.541013	2.3176551	19.
007	5704	2.0904340	21.631447	2.0904340	21.
008	5642	2.0677119	23.699159	2.0677119	23.
009	4960	1.8177687	25.516928	1.8177687	25.
010	4774	1.7496024	27.266530	1.7496024	27.
011	4712	1.7268803	28.993411	1.7268803	28.
012	4650	1.7041581	30.697569	1.7041581	30.
013	4526	1.6587139	32.356283	1.6587139	32.
014	4526	1.6587139	34.014997	1.6587139	34.
015	4340	1.5905476	35.605544	1.5905476	35.
016	4216	1.5451034	37.150648	1.5451034	37.
017	3906	1.4314928	38.582140	1.4314928	38.
018	3844	1.4087707	39.990911	1.4087707	39.
019	3658	1.3406044	41.331516	1.3406044	41.
020	3224	1.1815496	42.513065	1.1815496	42.
021	3472	1.2724381	43.785503	1.2724381	43.
022	3162	1.1588275	44.944331	1.1588275	44.
023	3162	1.1588275	46.103158	1.1588275	46.
024	3162	1.1588275	47.261986	1.1588275	47.
025	2790	1.0224949	48.284481	1.0224949	48.
026	2604	0.9543286	49.238809	0.9543286	49.
027	2232	0.8179959	50.056805	0.8179959	50.
028	2356	0.8634401	50.920245	0.8634401	50.
029	2356	0.8634401	51.783686	0.8634401	51.
030	2108	0.7725517	52.556237	0.7725517	52.
194	186	0.06816633	98.86389	0.06816633	98.
195	124	0.04544422	98.90934	0.04544422	98.
196	124	0.04544422	98.95478	0.04544422	98.
197	186	0.06816633	99.02295	0.06816633	99.
198	186	0.06816633	99.09112	0.06816633	99.
199	124	0.04544422	99.13656	0.04544422	99.
200	62	0.02272211	99.15928	0.02272211	99.
201	124	0.04544422	99.20473	0.04544422	99.
202	124	0.04544422	99.25017	0.04544422	99.
203	124	0.04544422	99.29561	0.04544422	99.
204	124	0.04544422	99.34106	0.04544422	99.
205	124	0.04544422	99.38650	0.04544422	99.
206	124	0.04544422	99.43195	0.04544422	99.
207	124	0.04544422	99.47739	0.04544422	99.
208	62	0.02272211	99.50011	0.02272211	99.
209	124	0.04544422	99.54556	0.04544422	99.
210	124	0.04544422	99.59100	0.04544422	99.
211	124	0.04544422	99.63645	0.04544422	99.
212	124	0.04544422	99.68189	0.04544422	99.
213	124	0.04544422	99.72733	0.04544422	99.
214	124	0.04544422	99.77278	0.04544422	99.

5. A summarytools: 4403 × 5 of type dbl

	Freq	% Valid	% Valid C
Ajalvir - 01.001	62	0.02272211	0.0227221
Ajalvir - 01.002	62	0.02272211	0.0454442
Alameda del Valle - 01.001	62	0.02272211	0.0681663
Álamo, El - 01.001	62	0.02272211	0.0908884
Álamo, El - 01.002	62	0.02272211	0.1136105
Álamo, El - 01.003	62	0.02272211	0.1363326
Álamo, El - 01.004	62	0.02272211	0.1590547
Alcalá de Henares - 01.001	62	0.02272211	0.1817768
Alcalá de Henares - 01.002	62	0.02272211	0.2044989
Alcalá de Henares - 01.003	62	0.02272211	0.2272210
Alcalá de Henares - 01.004	62	0.02272211	0.2499431
Alcalá de Henares - 01.005	62	0.02272211	0.2726653
Alcalá de Henares - 01.006	62	0.02272211	0.2953874
Alcalá de Henares - 01.007	62	0.02272211	0.3181095
Alcalá de Henares - 01.008	62	0.02272211	0.3408316
Alcalá de Henares - 01.009	62	0.02272211	0.3635537
Alcalá de Henares - 01.010	62	0.02272211	0.3862758
Alcalá de Henares - 01.011	62	0.02272211	0.4089979
Alcalá de Henares - 01.012	62	0.02272211	0.4317200
Alcalá de Henares - 01.013	62	0.02272211	0.4544421
Alcalá de Henares - 01.014	62	0.02272211	0.4771642
Alcalá de Henares - 01.015	62	0.02272211	0.4998863
Alcalá de Henares - 01.016	62	0.02272211	0.5226085
Alcalá de Henares - 01.017	62	0.02272211	0.5453306
Alcalá de Henares - 01.018	62	0.02272211	0.5680527
Alcalá de Henares - 01.020	62	0.02272211	0.5907748
Alcalá de Henares - 01.021	62	0.02272211	0.6134969
Alcalá de Henares - 01.022	62	0.02272211	0.6362190
Alcalá de Henares - 01.023	62	0.02272211	0.6589411
Alcalá de Henares - 01.024	62	0.02272211	0.6816632
Villanueva del Pardillo - 01.005	62	0.02272211	99.38650
Villanueva del Pardillo - 01.006	62	0.02272211	99.40923
Villanueva del Pardillo - 01.007	62	0.02272211	99.43195
Villanueva del Pardillo - 01.008	62	0.02272211	99.45467
Villar del Olmo - 01.001	62	0.02272211	99.47739
Villarejo de Salvanés - 01.001	62	0.02272211	99.50011
Villarejo de Salvanés - 01.002	62	0.02272211	99.52284
Villarejo de Salvanés - 01.003	62	0.02272211	99.54556
Villarejo de Salvanés - 01.004	62	0.02272211	99.56828
Villarejo de Salvanés - 01.005	62	0.02272211	99.59100
Villaviciosa de Odón - 01.001	62	0.02272211	99.61372
Villaviciosa de Odón - 01.002	62	0.02272211	99.63645
Villaviciosa de Odón - 01.003	62	0.02272211	99.65917
Villaviciosa de Odón - 01.004	62	0.02272211	99.68189
Villaviciosa de Odón - 01.005	62	0.02272211	99.70461
Villaviciosa de Odón - 01.006	62	0.02272211	99.72733
Villaviciosa de Odón - 01.007	62	0.02272211	99.75006
Villaviciosa de Odón - 01.008	62	0.02272211	99.77278
Villaviciosa de Odón - 01.009	62	0.02272211	99.79550
Villaviciosa de Odón - 01.010	62	0.02272211	99.81822
Villaviciosa de Odón - 01.011	62	0.02272211	99.84095

6. A summarytools: 4 × 5 of type dbl

	Freq	% Valid	% Valid Cum.	% Total	% Total Cu
FALSE	54571	19.99949	19.99949	19.99949	19.99949
TRUE	218291	80.00051	100.00000	80.00051	100.00000
<NA>	0	NA	NA	0.00000	100.00000
Total	272862	100.00000	100.00000	100.00000	100.00000

Structure of numerical features

[11]:

```
data |> select(where(is.numeric)) |> descr()
```

A summarytools: 15 × 12 of type dbl

	area	capacidad	densidad_hab_km2	eleva
Mean	1.773638e+06	6.934424e+01	2.694664e+04	6.710
Std.Dev	8.135112e+06	8.082869e+00	2.003986e+04	8.656
Min	7.404143e+03	5.000000e+01	3.840141e+00	4.590
Q1	2.963574e+04	6.400000e+01	9.545095e+03	6.300
Median	5.509161e+04	6.900000e+01	2.524896e+04	6.630
Q3	1.742421e+05	8.000000e+01	4.149213e+04	6.890
Max	1.808169e+08	8.000000e+01	1.165063e+05	1.507
MAD	4.874931e+04	7.413000e+00	2.365104e+04	4.299
IQR	1.446064e+05	1.600000e+01	3.194704e+04	5.900
CV	4.586682e+00	1.165615e-01	7.436870e-01	1.290
Skewness	9.931306e+00	-2.413341e-01	5.352551e-01	3.999
SE.Skewness	4.689232e-03	4.689232e-03	4.689232e-03	4.689
Kurtosis	1.501482e+02	-4.501317e-01	-2.545992e-01	2.476
N.Valid	2.728620e+05	2.728620e+05	2.728620e+05	2.728
Pct.Valid	1.000000e+02	1.000000e+02	1.000000e+02	1.000

0.6 Data Types

Tipo de datos

[12]:

```
sapply(data, class)
glimpse(data)
```

Fecha 'Date' **Servicio** 'character' **CMUN** 'character' **CDIS** 'character' **CSEC** 'character' **Futbol** 'numeric' **nservicios** 'numeric' **capacidad** 'numeric' **tmed** 'numeric' **prec** 'numeric' **velmedia** 'numeric' **presMax** 'numeric' **t1__1** 'numeric' **t3__1** 'numeric' **NSEC** 'character' **area** 'numeric' **elevation** 'numeric' **densidad__hab__km2** 'numeric' **is__train** 'logical'

```
Rows: 272,862
Columns: 19
$ Fecha             <date> 2022-01-12, 2022-01-31, 2022-01-28,
2022-01-06, 2022…
$ Servicio          <chr> "Delivery", "Taxi", "Taxi",
"Delivery", "Delivery", "…
$ CMUN              <chr> "079", "079", "903", "079", "007",
"022", "079", "079…
$ CDIS              <chr> "14", "01", "01", "04", "04", "01",
"16", "01", "16",…
$ CSEC              <chr> "050", "048", "006", "080", "012",
"004", "041", "033…
$ Futbol           <dbl> 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0,
0, 0, 0, 0, 0, 0,…
$ nservicios       <dbl> 58, 5, 0, 14, 60, 50, 13, 4, 3, 9,
68, 12, 0, 1, 14, …
$ capacidad        <dbl> 80, 69, 56, 80, 70, 56, 80, 69, 69,
69, 80, 80, 69, 6…
$ tmed             <dbl> 7.366319, 8.823406, 7.854915,
4.226603, 4.982656, 7.2…
$ prec             <dbl> -0.009468616, 0.000000000,
0.000000000, 0.010181896, …
$ velmedia         <dbl> 1.5999961, 1.5114967, 2.2536168,
1.0279945, 1.0387037…
$ presMax          <dbl> 954.7939, 948.9795, 940.2553,
945.1884, 948.9570, 943…
$ t1_1             <dbl> 1094, 1251, 2232, 746, 1080, 2256,
692, 1270, 2229, 8…
$ t3_1             <dbl> 45.4360, 41.6091, 44.2016, 47.1729,
48.5361, 43.2877,…
$ NSEC             <chr> "Madrid - 14.050", "Madrid -
01.048", "Tres Cantos - …
$ area             <dbl> 38753.96, 15289.89, 124539.78,
89206.78, 24473.30, 34…
$ elevation        <dbl> 658, 635, 719, 710, 693, 710, 702,
635, 690, 710, 690…
$ densidad_hab_km2 <dbl> 28229.3737, 81818.7738, 17921.9842,
8362.5928, 44129.…
$ is_train         <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE,
TRUE, TRUE, TRUE,…
```

0.7 Statistical Measures

[13]:

```
data  |> descr()
```

A summarytools: 15 × 12 of type dbl

	area	capacidad	densidad_hab_km2	eleva
Mean	1.773638e+06	6.934424e+01	2.694664e+04	6.710
Std.Dev	8.135112e+06	8.082869e+00	2.003986e+04	8.656
Min	7.404143e+03	5.000000e+01	3.840141e+00	4.590
Q1	2.963574e+04	6.400000e+01	9.545095e+03	6.300
Median	5.509161e+04	6.900000e+01	2.524896e+04	6.630
Q3	1.742421e+05	8.000000e+01	4.149213e+04	6.890
Max	1.808169e+08	8.000000e+01	1.165063e+05	1.507
MAD	4.874931e+04	7.413000e+00	2.365104e+04	4.299
IQR	1.446064e+05	1.600000e+01	3.194704e+04	5.900
CV	4.586682e+00	1.165615e-01	7.436870e-01	1.290
Skewness	9.931306e+00	-2.413341e-01	5.352551e-01	3.999
SE.Skewness	4.689232e-03	4.689232e-03	4.689232e-03	4.689
Kurtosis	1.501482e+02	-4.501317e-01	-2.545992e-01	2.476
N.Valid	2.728620e+05	2.728620e+05	2.728620e+05	2.728
Pct.Valid	1.000000e+02	1.000000e+02	1.000000e+02	1.000

0.8 Uniques values

[14]:
```
# Rthe number of unique values in each column.
data |> summarise(across(everything(), n_distinct))
```

A tibble: 1 × 19

Fecha	Servicio	CMUN	CDIS	CSEC	Futbol	nservicios	capacidad	tmed	prec
<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>
31	2	171	21	221	2	62	24	136406	39601

0.9 CrossTab

Select columns

Hacer los cruces que tengan sentido

[15]:
```
data |> select(where(~ !is.numeric(.x))) |> colnames()
Column1 <- "Fecha"
Column2 <- "Servicio"
```

1. 'Fecha' 2. 'Servicio' 3. 'CMUN' 4. 'CDIS' 5. 'CSEC' 6. 'NSEC' 7. 'is_train'

Operation

[16]:
```
# Referencia cruzada de variables
ctable(data[[Column1]], data[[Column2]])
```

$cross_table A table: 32 × 3 of type dbl

	Delivery	Taxi	Total
2022-01-01	4401	4401	8802
2022-01-02	4401	4401	8802
2022-01-03	4401	4401	8802
2022-01-04	4401	4401	8802
2022-01-05	4401	4401	8802
2022-01-06	4401	4401	8802
2022-01-07	4401	4401	8802
2022-01-08	4401	4401	8802
2022-01-09	4401	4401	8802
2022-01-10	4401	4401	8802
2022-01-11	4401	4401	8802
2022-01-12	4401	4401	8802
2022-01-13	4401	4401	8802
2022-01-14	4401	4401	8802
2022-01-15	4401	4401	8802
2022-01-16	4401	4401	8802
2022-01-17	4401	4401	8802
2022-01-18	4401	4401	8802
2022-01-19	4401	4401	8802
2022-01-20	4401	4401	8802
2022-01-21	4401	4401	8802
2022-01-22	4401	4401	8802
2022-01-23	4401	4401	8802
2022-01-24	4401	4401	8802
2022-01-25	4401	4401	8802
2022-01-26	4401	4401	8802
2022-01-27	4401	4401	8802
2022-01-28	4401	4401	8802
2022-01-29	4401	4401	8802
2022-01-30	4401	4401	8802
2022-01-31	4401	4401	8802
Total	136431	136431	272862

$proportions A matrix: 32 × 3 of type dbl

	Delivery	Taxi	Total
2022-01-01	0.5	0.5	1
2022-01-02	0.5	0.5	1
2022-01-03	0.5	0.5	1
2022-01-04	0.5	0.5	1
2022-01-05	0.5	0.5	1
2022-01-06	0.5	0.5	1
2022-01-07	0.5	0.5	1
2022-01-08	0.5	0.5	1
2022-01-09	0.5	0.5	1
2022-01-10	0.5	0.5	1
2022-01-11	0.5	0.5	1
2022-01-12	0.5	0.5	1
2022-01-13	0.5	0.5	1
2022-01-14	0.5	0.5	1
2022-01-15	0.5	0.5	1
2022-01-16	0.5	0.5	1
2022-01-17	0.5	0.5	1
2022-01-18	0.5	0.5	1
2022-01-19	0.5	0.5	1
2022-01-20	0.5	0.5	1
2022-01-21	0.5	0.5	1
2022-01-22	0.5	0.5	1
2022-01-23	0.5	0.5	1
2022-01-24	0.5	0.5	1
2022-01-25	0.5	0.5	1
2022-01-26	0.5	0.5	1
2022-01-27	0.5	0.5	1
2022-01-28	0.5	0.5	1
2022-01-29	0.5	0.5	1
2022-01-30	0.5	0.5	1
2022-01-31	0.5	0.5	1
Total	0.5	0.5	1

0.10 Analyzing Numerical Variables

0.10.1 Selecting continuous variables

[17]:

```
# Numeric colums
cdata <- data |> select(where(is.numeric))
```

0.10.2 Global view of the numerical variables

Global view on the dataset to identify some very unusual patterns.

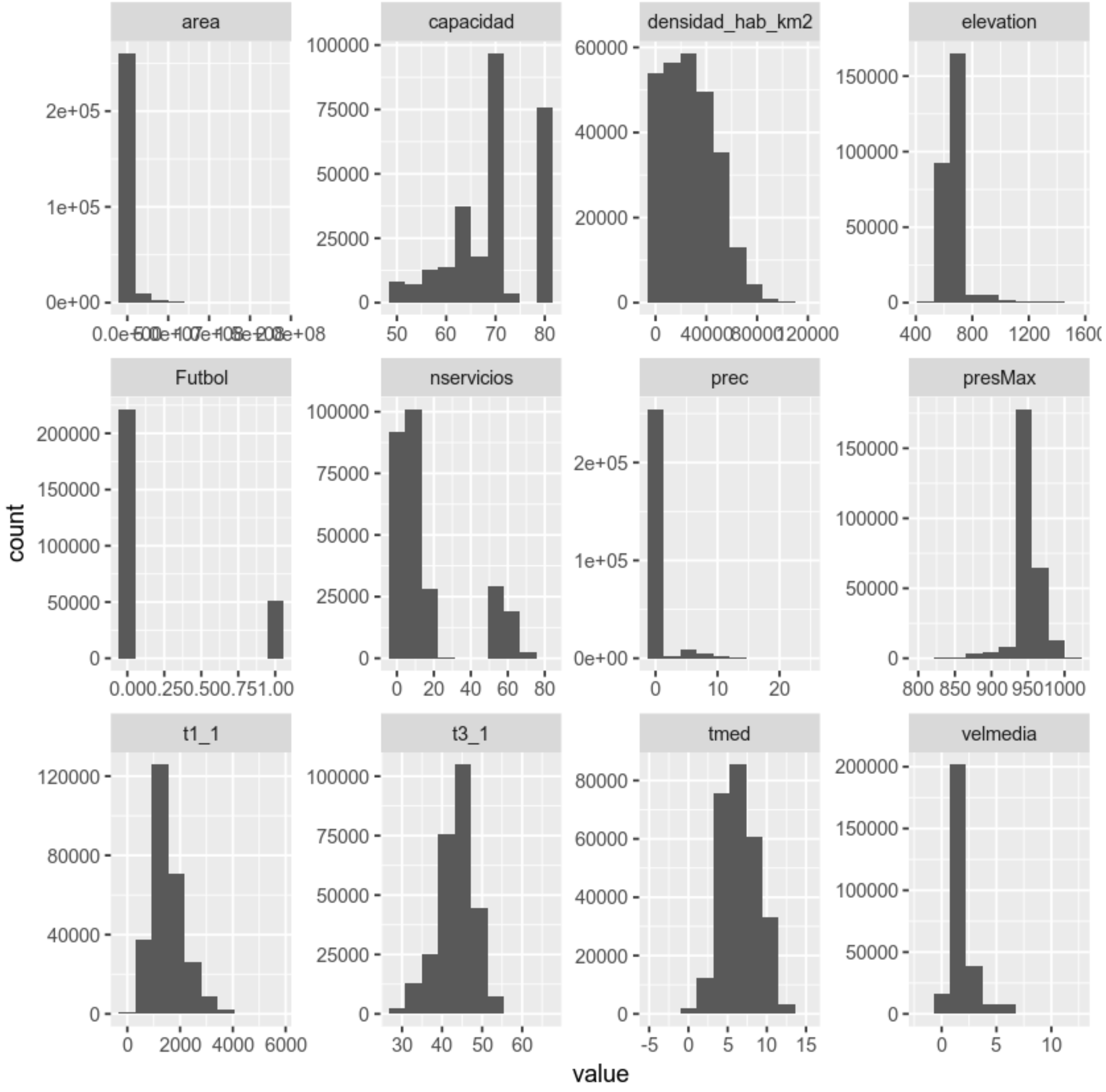
NOTA: Esto puede tardar si hay muchas variables

[18]:

```
#pairs(cdata)
# cdata |> ggpairs()
```

0.10.3 Histograms

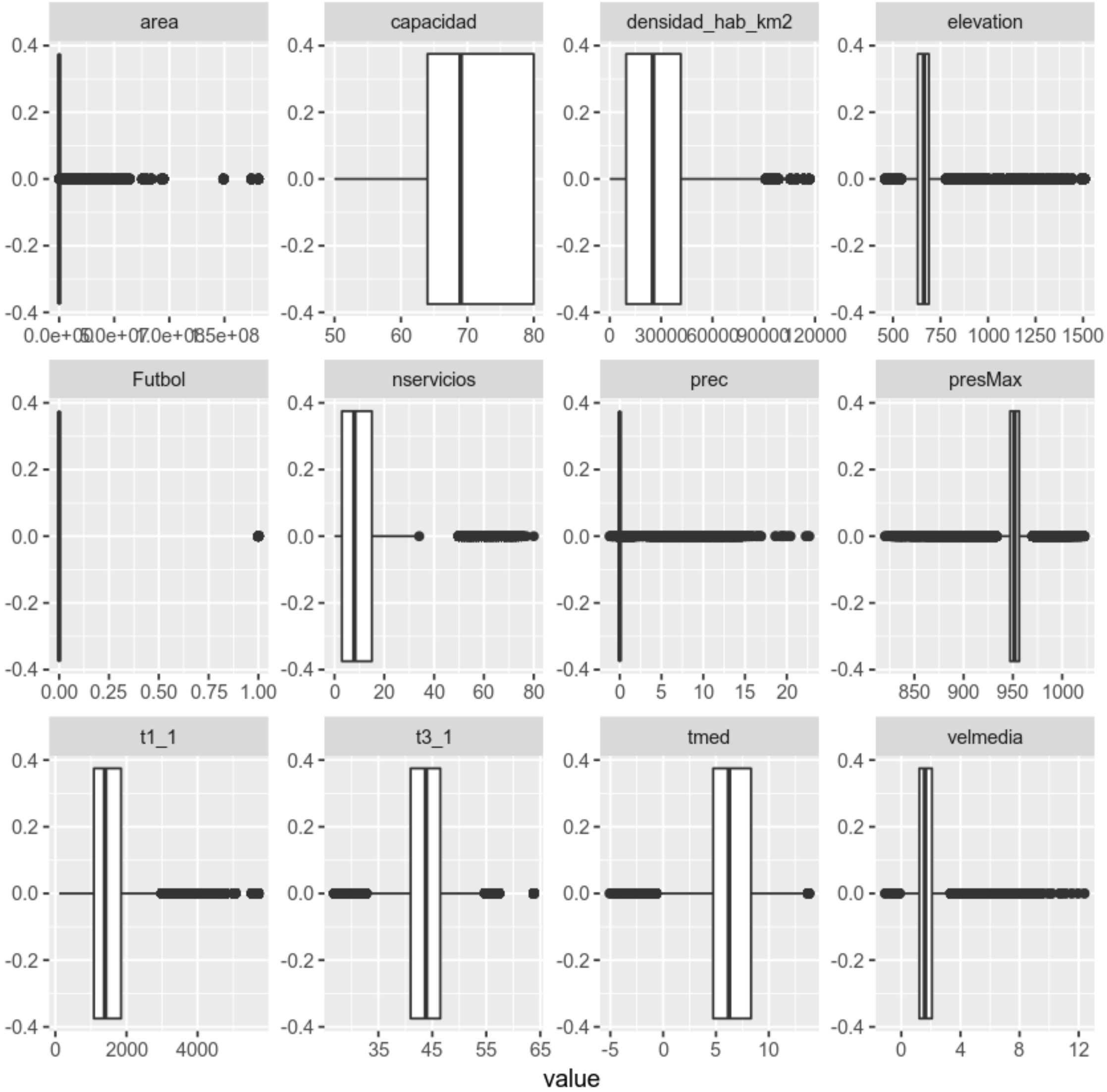
```
[19]: cdata |>
        pivot_longer(cols = everything()) |>
        ggplot(aes(x = value)) +
        geom_histogram(bins = 10) +
        facet_wrap(~name, scales = "free")
```



0.10.4 Box plot

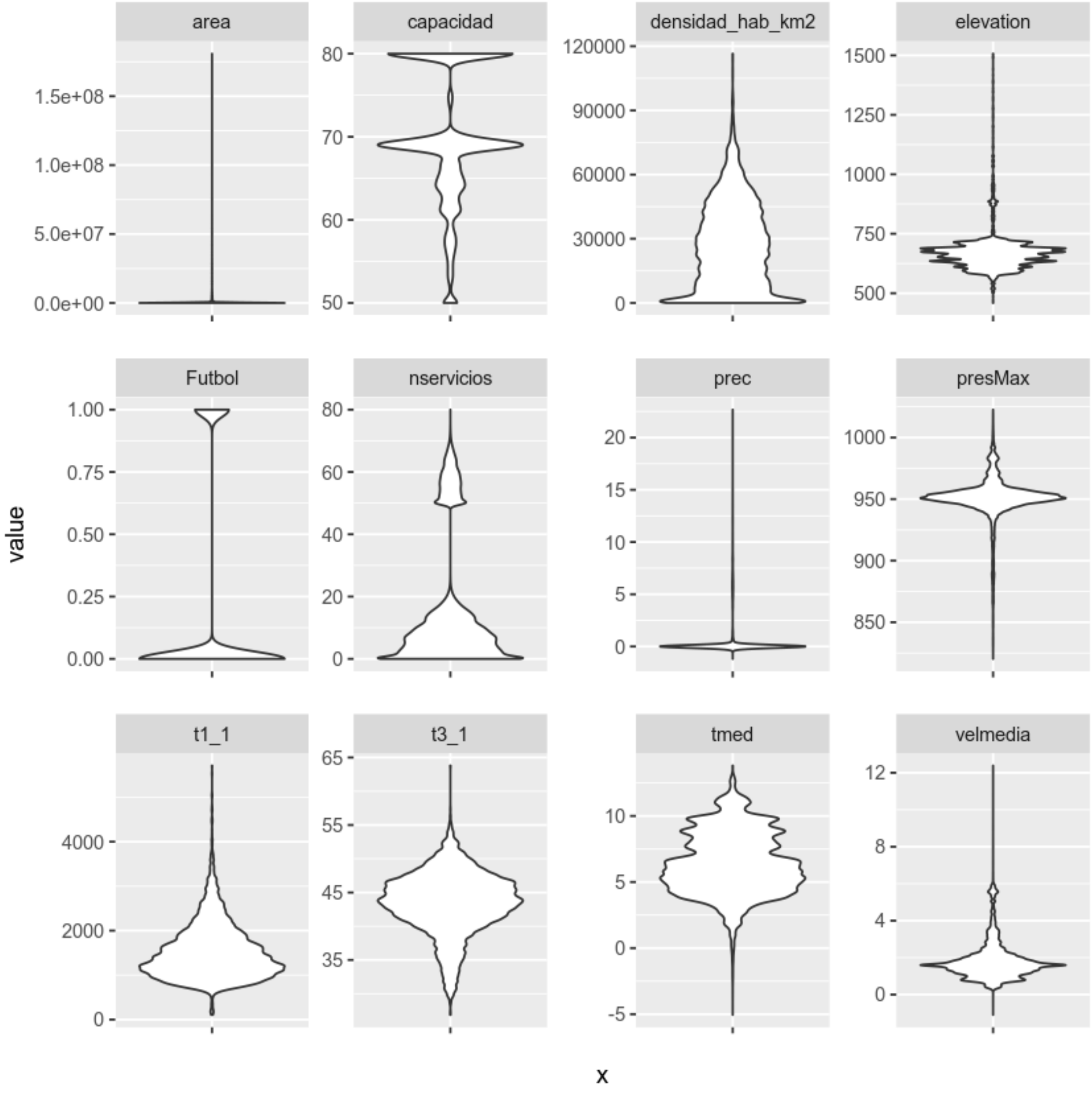
[20]:

```
cdata |>
  pivot_longer(cols = everything()) |>
  ggplot(aes(x = value)) +
  geom_boxplot() +
  facet_wrap(~name, scales = "free")
```



0.10.5 Violin plot

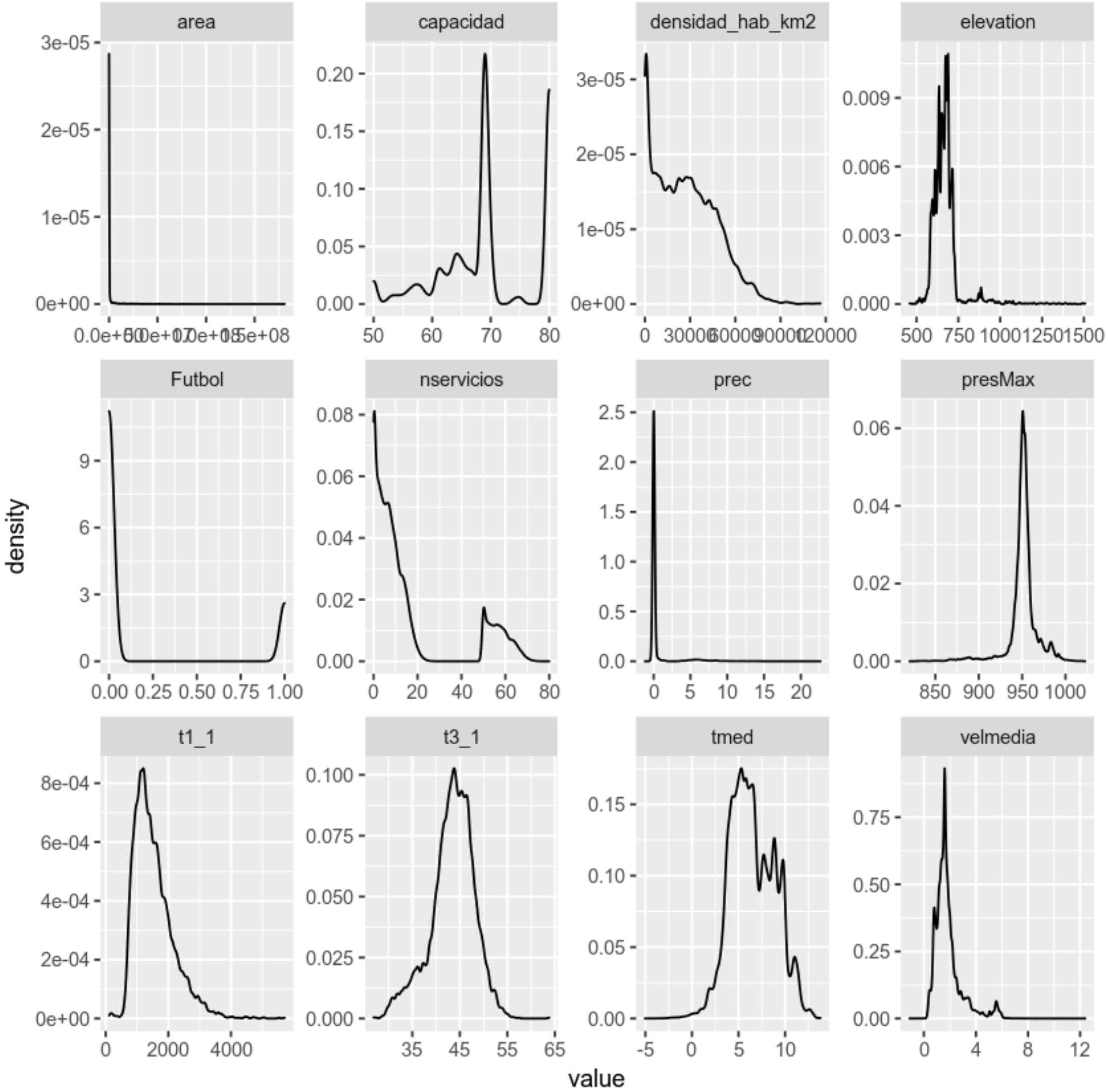
[21]:
```
cdata |>
  pivot_longer(cols = everything()) |>
  ggplot(aes(x = "", y = value)) +
  geom_violin() +
  facet_wrap(~name, scales = "free")
```



0.10.6 Distribution plot

[22]:

```
cdata |>
  pivot_longer(cols = everything()) |>
  ggplot(aes(x = value)) +
  geom_density() +
  facet_wrap(~name, scales = "free")
```



0.11 Analyzing Categorical Variables

0.11.1 Selecting categorical variables

[23]:

```
# Category colums
char_cols <- data |> select(where(~ !is.numeric(.x))) |> colnames()
char_cols
```

1. ’Fecha’ 2. ’Servicio’ 3. ’CMUN’ 4. ’CDIS’ 5. ’CSEC’ 6. ’NSEC’ 7. ’is_train’

[24]:

```
# Category colums
char_data <- data |> select(where(~ !is.numeric(.x)))
char_data <- char_data[,!names(char_data) %in% c("Fecha", "is_train")]
char_data
```

A tibble: 272862 × 5

Servicio	CMUN	CDIS	CSEC	NSEC
<chr>	<chr>	<chr>	<chr>	<chr>
Delivery	079	14	050	Madrid - 14.050
Taxi	079	01	048	Madrid - 01.048
Taxi	903	01	006	Tres Cantos - 01.006
Delivery	079	04	080	Madrid - 04.080
Delivery	007	04	012	Alcorcón - 04.012
Taxi	022	01	004	Boadilla del Monte - 01.004
Delivery	079	16	041	Madrid - 16.041
Taxi	079	01	033	Madrid - 01.033
Taxi	079	16	108	Madrid - 16.108
Taxi	079	05	024	Madrid - 05.024
Delivery	079	04	129	Madrid - 04.129
Delivery	079	20	090	Madrid - 20.090
Taxi	079	01	018	Madrid - 01.018
Taxi	123	01	032	Rivas-Vaciamadrid - 01.032
Delivery	127	01	031	Rozas de Madrid, Las - 01.031
Delivery	079	14	036	Madrid - 14.036
Taxi	079	04	072	Madrid - 04.072
Delivery	079	11	149	Madrid - 11.149
Delivery	079	08	019	Madrid - 08.019
Taxi	005	02	028	Alcalá de Henares - 02.028
Delivery	013	02	007	Aranjuez - 02.007
Delivery	079	08	035	Madrid - 08.035
Delivery	074	05	004	Leganés - 05.004
Taxi	079	17	022	Madrid - 17.022
Delivery	054	01	002	Escorial, El - 01.002
Delivery	079	11	060	Madrid - 11.060
Taxi	134	01	028	San Sebastián de los Reyes - 01.028
Taxi	004	01	003	Álamo, El - 01.003
Delivery	079	19	016	Madrid - 19.016
Delivery	079	06	073	Madrid - 06.073
Delivery	161	02	003	Valdemoro - 02.003
Delivery	161	02	004	Valdemoro - 02.004
Delivery	161	02	012	Valdemoro - 02.012
Delivery	161	02	015	Valdemoro - 02.015
Delivery	164	01	001	Valdetorres de Jarama - 01.001
Delivery	164	01	002	Valdetorres de Jarama - 01.002
Delivery	171	01	003	Villa del Prado - 01.003
Delivery	171	01	004	Villa del Prado - 01.004
Delivery	172	01	003	Villalbilla - 01.003
Delivery	177	01	003	Villanueva del Pardillo - 01.003
Delivery	177	01	005	Villanueva del Pardillo - 01.005
Delivery	180	01	005	Villarejo de Salvanés - 01.005
Delivery	181	01	001	Villaviciosa de Odón - 01.001
Delivery	181	01	003	Villaviciosa de Odón - 01.003
Delivery	181	01	005	Villaviciosa de Odón - 01.005
Delivery	902	01	001	Puentes Viejas - 01.001
Delivery	903	01	006	Tres Cantos - 01.006
Delivery	903	01	008	Tres Cantos - 01.008
Delivery	903	01	010	Tres Cantos - 01.010
Delivery	903	01	012	Tres Cantos - 01.012

0.11.2 Most frequent entry

- Ver salida de `summarytools::freq()` arriba

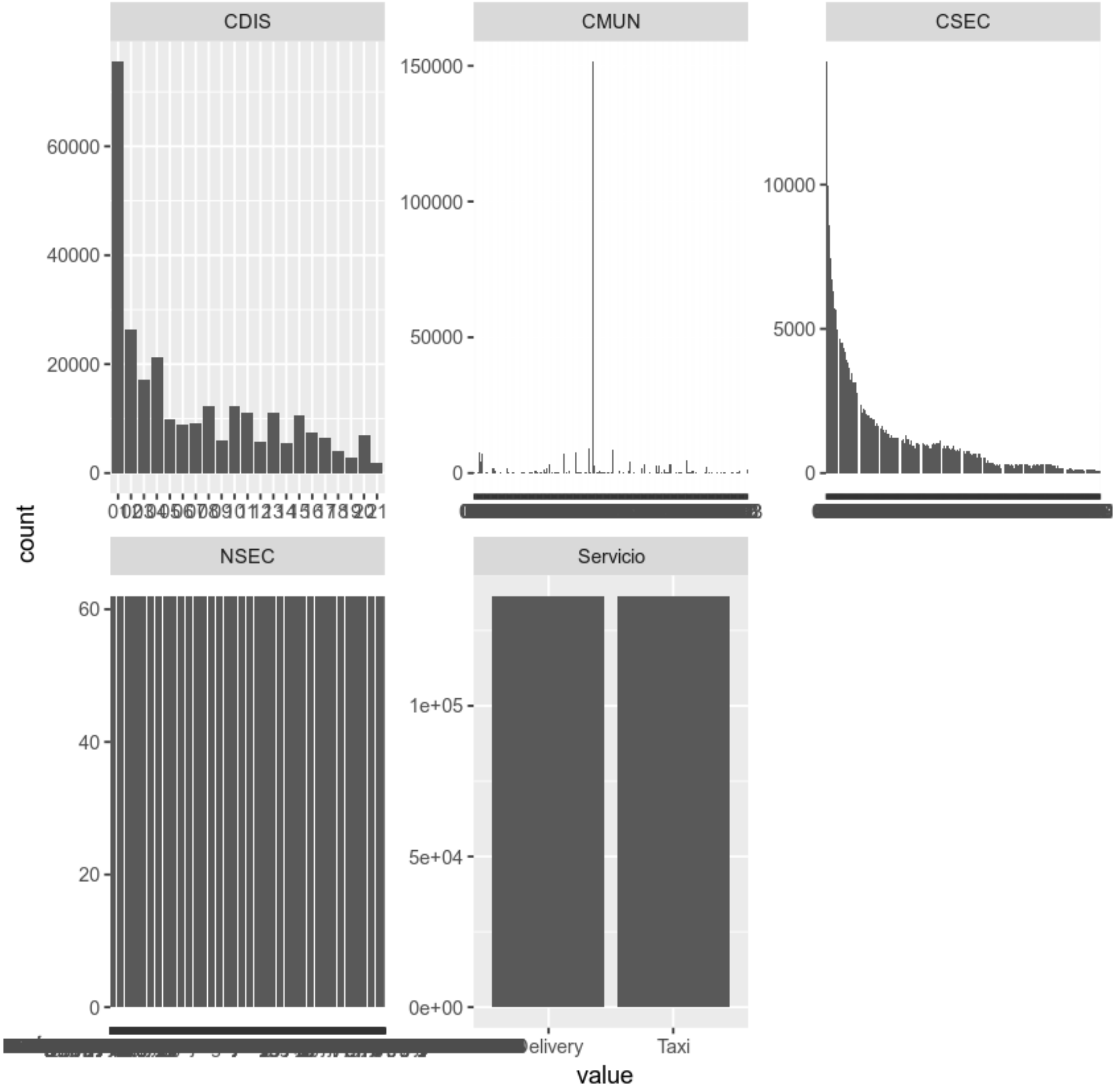
[25]:

```
# Calculate and visualizate the ratio of the most frequent entry for each feature
```

0.11.3 Visualization of categorical variables

[26]:

```
# returns a visualization of the number and frequency of categorical features
char_data |>
  pivot_longer(cols = everything()) |>
  ggplot(aes(x = value)) +
  geom_bar() +
  facet_wrap(~name, scales = "free")
```



0.12 Statistical Normality Tests

[27]:
```
cdata_long <- cdata |>
  pivot_longer(cols = everything())
```

0.12.1 Test de Shapiro-Wilk

Si hay muchos datos este no se puede hacer

[28]:
```
#tapply(cdata_long$value, cdata_long$name, shapiro.test)
```

0.12.2 Test de Anderson-Darling

[29]:
```
tapply(cdata_long$value, cdata_long$name, ad.test)
```

```
$area

	Anderson-Darling normality test

data:  X[[i]]
A = 80340, p-value < 2.2e-16


$capacidad

	Anderson-Darling normality test

data:  X[[i]]
A = 11050, p-value < 2.2e-16


$densidad_hab_km2

	Anderson-Darling normality test

data:  X[[i]]
A = 2833.2, p-value < 2.2e-16


$elevation

	Anderson-Darling normality test

data:  X[[i]]
```

```
A = 19128, p-value < 2.2e-16


$Futbol

        Anderson-Darling normality test

data:  X[[i]]
A = 74480, p-value < 2.2e-16


$nservicios

        Anderson-Darling normality test

data:  X[[i]]
A = 35080, p-value < 2.2e-16


$prec

        Anderson-Darling normality test

data:  X[[i]]
A = 86614, p-value < 2.2e-16


$presMax

        Anderson-Darling normality test

data:  X[[i]]
A = 17182, p-value < 2.2e-16


$t1_1

        Anderson-Darling normality test

data:  X[[i]]
A = 5041.6, p-value < 2.2e-16


$t3_1

        Anderson-Darling normality test

data:  X[[i]]
```

```
A = 1680.9, p-value < 2.2e-16


$tmed

        Anderson-Darling normality test

data:  X[[i]]
A = 1191, p-value < 2.2e-16


$velmedia

        Anderson-Darling normality test

data:  X[[i]]
A = 13336, p-value < 2.2e-16
```

0.12.3 Test de Lilliefors

[30]:

```
tapply(cdata_long$value, cdata_long$name, lillie.test)
```

```
$area

        Lilliefors (Kolmogorov-Smirnov) normality test

data:  X[[i]]
D = 0.42034, p-value < 2.2e-16


$capacidad

        Lilliefors (Kolmogorov-Smirnov) normality test

data:  X[[i]]
D = 0.18547, p-value < 2.2e-16


$densidad_hab_km2

        Lilliefors (Kolmogorov-Smirnov) normality test

data:  X[[i]]
D = 0.0894, p-value < 2.2e-16
```

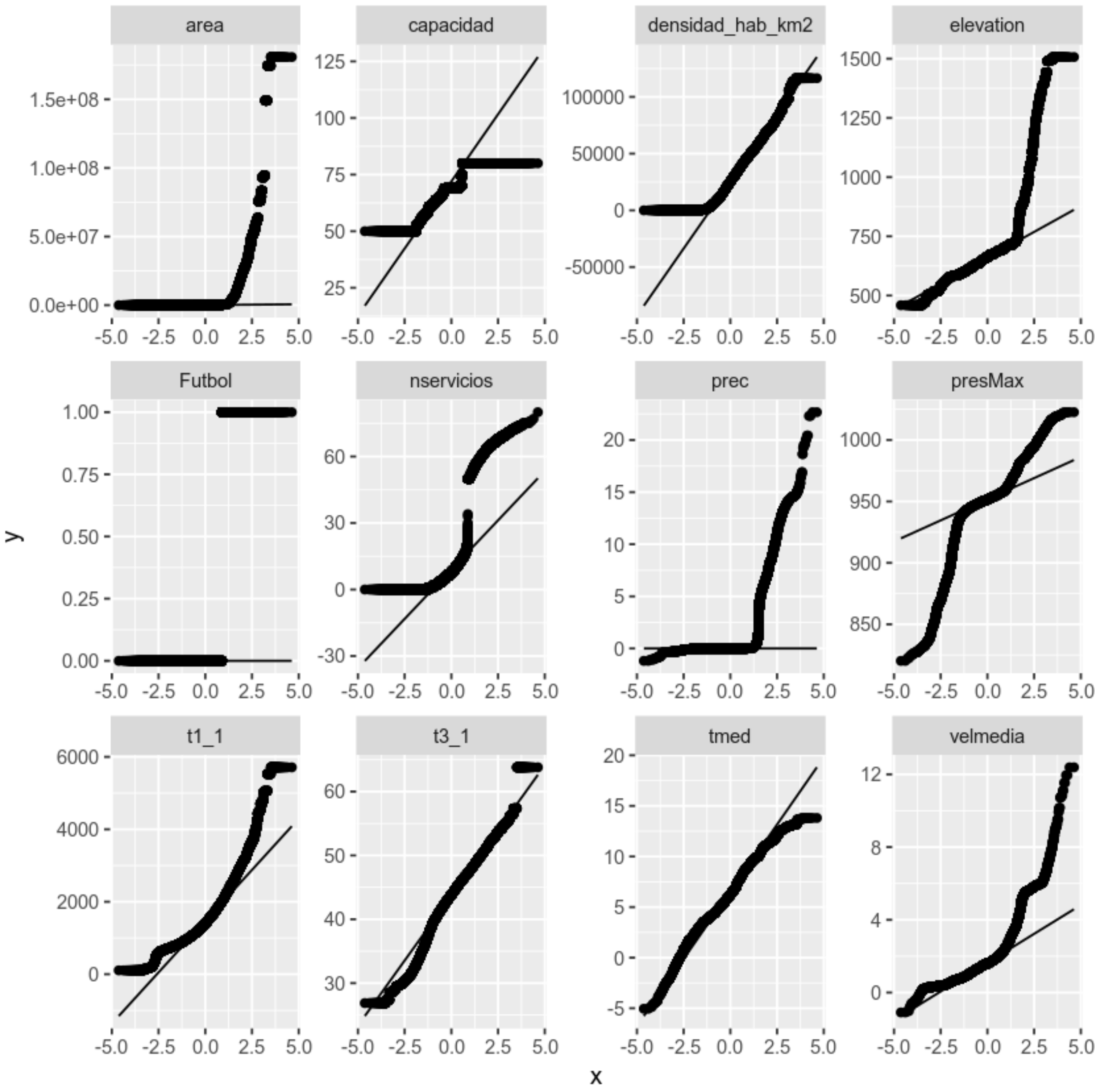
```
$elevation

	Lilliefors (Kolmogorov-Smirnov) normality test

data:  X[[i]]
D = 0.21372, p-value < 2.2e-16


$Futbol

	Lilliefors (Kolmogorov-Smirnov) normality test

data:  X[[i]]
D = 0.49651, p-value < 2.2e-16


$nservicios

	Lilliefors (Kolmogorov-Smirnov) normality test

data:  X[[i]]
D = 0.27719, p-value < 2.2e-16


$prec

	Lilliefors (Kolmogorov-Smirnov) normality test

data:  X[[i]]
D = 0.46853, p-value < 2.2e-16


$presMax

	Lilliefors (Kolmogorov-Smirnov) normality test

data:  X[[i]]
D = 0.18147, p-value < 2.2e-16


$t1_1

	Lilliefors (Kolmogorov-Smirnov) normality test

data:  X[[i]]
D = 0.088341, p-value < 2.2e-16
```

```
$t3_1

        Lilliefors (Kolmogorov-Smirnov) normality test

data:  X[[i]]
D = 0.052758, p-value < 2.2e-16


$tmed

        Lilliefors (Kolmogorov-Smirnov) normality test

data:  X[[i]]
D = 0.04842, p-value < 2.2e-16


$velmedia

        Lilliefors (Kolmogorov-Smirnov) normality test

data:  X[[i]]
D = 0.16245, p-value < 2.2e-16
```

0.12.4 QQ-plots

[31]:

```
cdata |>
  pivot_longer(cols = everything()) |>
  ggplot(aes(sample = value)) +
  geom_qq() +
  geom_qq_line() +
  facet_wrap(~name, scales = "free")
```



0.13 Bivariate analysis

- Ver gráficos de dispersión y ggpairs arriba
- Completar si es necesario con alguna comparación específica (gráfico de dispersión o boxplot por grupos)

Correlaciones

```
[32]: cor(cdata, use = "pairwise.complete.obs")
```

A matrix: 12 × 12 of type dbl

	Futbol	nservicios	capacidad	tmed
Futbol	1.0000000000	0.965537900	0.01695169	0.0120379536
nservicios	0.9655379005	1.000000000	0.18569366	0.0299971395
capacidad	0.0169516892	0.185693656	1.00000000	0.1898186150
tmed	0.0120379536	0.029997139	0.18981861	1.0000000000
prec	0.0100939261	0.010773351	-0.01242114	0.0426040672
velmedia	0.0027871981	-0.005490080	-0.08321610	-0.0021432471
presMax	-0.0020373676	-0.015623395	0.03982993	-0.0007697094
t1_1	0.0006808222	-0.026383611	-0.29650904	-0.1180439751
t3_1	-0.0021894861	0.023409098	0.24662839	0.0947745176
area	-0.0002523514	-0.033626213	-0.32552945	-0.1024089989
elevation	-0.0001096484	-0.005618554	-0.23682267	-0.1210573259
densidad_hab_km2	0.0019383112	0.084518064	0.36601631	0.1372330715

0.14 Regression analysis

0.14.1 Modelo completo regresión lineal simple

[33]:

```
# modelo <- lm(xxxx ~ ., data = cdata)
# summary(modelo)
```

[34]:

```
#plot(modelo)
```

0.14.2 Selección de variables

Puede que dé error por la estructura de los datos, en ese caso dejarlo indicado

[35]:

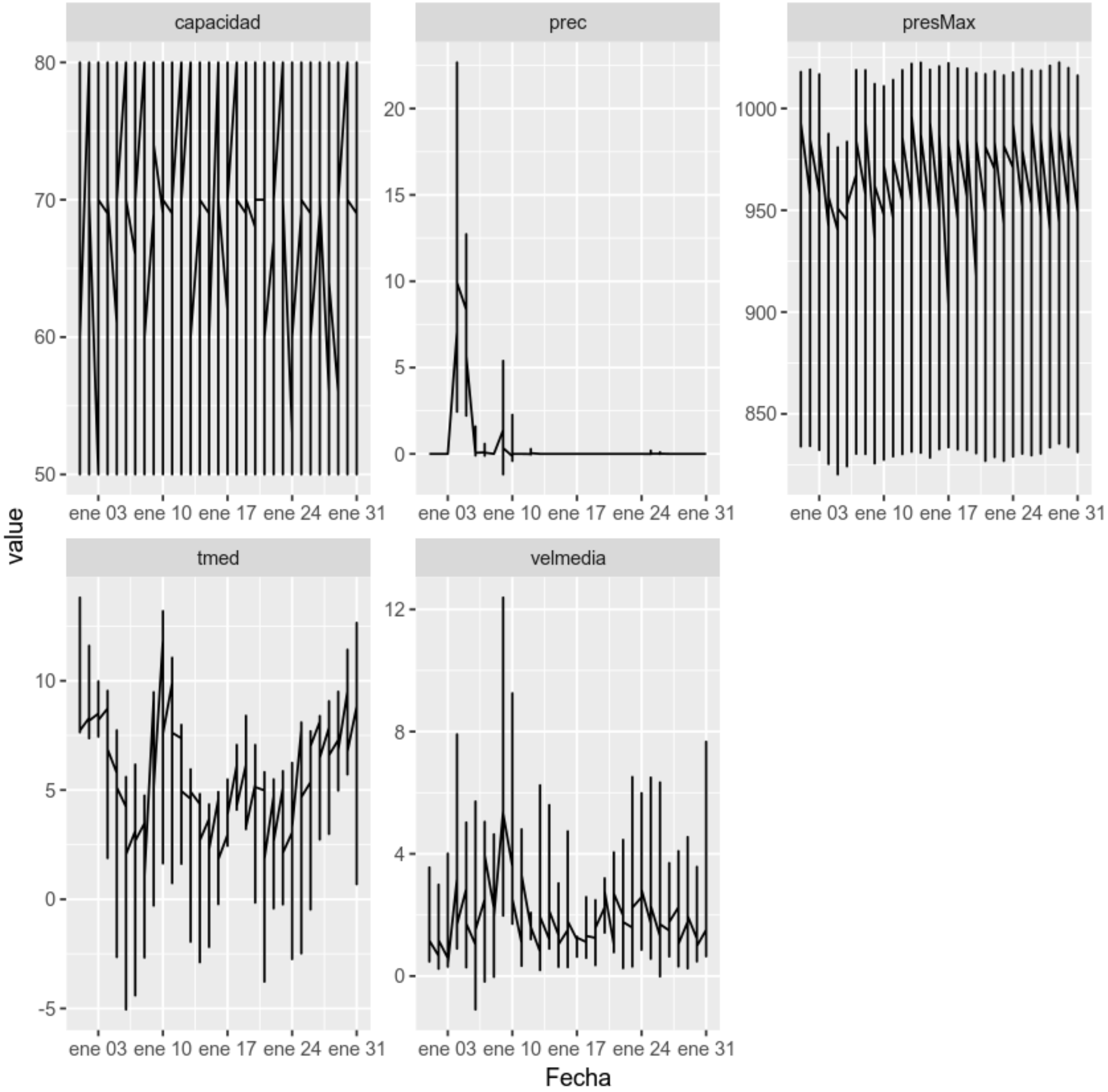
```
# modelo2 <- step(modelo, trace = FALSE)
# summary(modelo2)
```

0.15 Stationary analysis

- Si hay una variable fecha, usarla
- Si hay mes, o semana, convertir a fecha

[36]:

```
data |>
  pivot_longer(cols = capacidad:presMax) |>
  ggplot(aes(x = Fecha, y = value)) +
  geom_line() +
  facet_wrap(~name, scales = "free")
```



Todas las series, probablemente habría que filtrar por geografía

0.16 Data Save

- Solo si se han hecho cambios
- No aplica

Identificamos los datos a guardar

[37]:
```
data_to_save <- data
```

Estructura de nombre de archivos:

- Código del caso de uso, por ejemplo “CU_04”

- Número del proceso que lo genera, por ejemplo "_06".
- Resto del nombre del archivo de entrada
- Extensión del archivo

Ejemplo: "CU_04_06_01_01_zonasgeo.json, primer fichero que se genera en la tarea 01 del proceso 05 (Data Collection) para el caso de uso 04 (vacunas) y que se ha transformado en el proceso 06

Importante mantener los guiones bajos antes de proceso, tarea, archivo y nombre

0.16.1 Proceso 12

[38]:
```r
caso <- "CU_34"
proceso <- '_12'
tarea <- "_05"
archivo <- ""
proper <- "_servicios_completo"
extension <- ".csv"
```

OPCION A: Uso del paquete "tcltk" para mayor comodidad

- Buscar carpeta, escribir nombre de archivo SIN extensión (se especifica en el código)
- Especificar sufijo2 si es necesario
- Cambiar datos por datos_xx si es necesario

[39]:
```r
# file_save <- paste0(caso, proceso, tarea, tcltk::tkgetSaveFile(), proper,␣
 ↪extension)
# path_out <- paste0(oPath, file_save)
# write_csv(data_to_save_xxxxx, path_out)

# cat('File saved as: ')
# path_out
```

OPCION B: Especificar el nombre de archivo

- Los ficheros de salida del proceso van siempre a Data/Output/.

[40]:
```r
file_save <- paste0(caso, proceso, tarea, archivo, proper, extension)
path_out <- paste0(oPath, file_save)
write_csv(data_to_save, path_out)

cat('File saved as: ')
path_out
```

```
File saved as:
```

'Data/Output/CU_34_12_05_servicios_completo.csv'

Copia del fichero a Input Si el archivo se va a usar en otros notebooks, copiar a la carpeta Input

```
[41]: path_in <- paste0(iPath, file_save)
      file.copy(path_out, path_in, overwrite = TRUE)
```

```
TRUE
```

0.17 REPORT

A continuación se realizará un informe de las acciones realizadas

0.18 Main Actions Carried Out

- Se ha realizado exploratorio de los datos del caso de uso

0.19 Main Conclusions

- Los datos son adecuados para el caso de uso

0.20 CODE TO DEPLOY (PILOT)

A continuación se incluirá el código que deba ser llevado a despliegue para producción, dado que se entiende efectúa operaciones necesarias sobre los datos en la ejecución del prototipo

Description

- No hay nada que desplegar en el piloto, ya que estos datos son estáticos o en todo caso cambian con muy poca frecuencia, altamente improbable durante el proyecto.

CODE