data-quality-analysis-ES

April 28, 2024

1 Análisis de Calidad de Datos

Emma Arenas Villaverde

1.1 Importar Librerías

[ ]:
```r
install.packages("readr") # para leer archivos CSV
install.packages("dplyr") # para manipulación de datos
library(readr)
library(dplyr)
```

1.2 Cargar Dataset

[ ]:
```r
BBDD_Locales <- read_csv2("data/BBDD_Locales.csv") #  la función read_csv2␣
↪viene configurada por defecto para utilizar el punto y coma como delimitador
```

1.3 Información de los Datos

[6]:
```r
dim(BBDD_Locales) # para obtener sus dimensiones
```

1. 766 2. 7

[7]:
```r
str(BBDD_Locales) # para ver su estructura interna
```

```
spc_tbl_ [766 × 7] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ municipio   : chr [1:766] "Municipio29" "Municipio29" "Municipio29"
"Municipio29" …
 $ sector      : chr [1:766] "Menaje" "Otros" "No alimentario" "Otros" …
 $ situacion   : chr [1:766] "calle" "calle" "calle" "calle" …
 $ forma       : chr [1:766] "SL" "SL" "SA" "individual" …
 $ superficie  : num [1:766] 99.3 22.5 NA 24.9 21.2 …
 $ trabajadores: num [1:766] 6 3 3 1 2 5 3 4 1 4 …
 $ antiguedad  : num [1:766] 36.5 14.2 9 11.4 17.1 26.9 13.3 13.5 29.6 6.4 …
 - attr(*, "spec")=
  .. cols(
  ..   municipio = col_character(),
  ..   sector = col_character(),
```

```
..   situacion = col_character(),
..   forma = col_character(),
..   superficie = col_double(),
..   trabajadores = col_double(),
..   antiguedad = col_double()
.. )
- attr(*, "problems")=<externalptr>
```

[8]:
```
head(BBDD_Locales, n = 766) # para ver sus filas
```

municipio <chr>	sector <chr>	situacion <chr>	forma <chr>	superficie <dbl>	trabajado <dbl>
Municipio29	Menaje	calle	SL	99.32	6
Municipio29	Otros	calle	SL	22.51	3
Municipio29	No alimentario	calle	SA	NA	3
Municipio29	Otros	calle	individual	24.85	1
Municipio29	Equipamientos culturales	calle	SA	21.21	2
Municipio29	Alimentario	calle	SA	46.14	5
Municipio29	No alimentario	calle	SA	44.96	3
Municipio29	Ocio y cultura	centro comercial	SL	35.16	4
Municipio29	Alimentario	calle	SA	20.48	1
Municipio29	Menaje	calle	SL	57.44	4
Municipio29	Otros	calle	individual	7.73	1
Municipio29	Reparaciones	calle	SL	44.75	4
Municipio29	Alimentario	calle	SA	24.32	2
Municipio29	Otros	calle	SL	33.23	2
Municipio29	Menaje	calle	individual	65.64	4
Municipio29	Otros	calle	SA	23.01	3
Municipio29	Otros	calle	individual	9.85	2
Municipio29	Alimentario	calle	SL	40.23	5
Municipio29	Menaje	calle	SL	45.82	3
Municipio29	Alimentario	calle	individual	10.30	1
Municipio29	Equipamiento personal	calle	SL	64.58	3
Municipio29	Enseñanza	calle	individual	43.54	3
Municipio29	Menaje	centro comercial	SA	48.11	3
Municipio29	Menaje	calle	individual	61.69	2
Municipio29	Otros	calle	SL	5.26	2
Municipio29	Alimentario	calle	SA	46.44	4
Municipio29	Enseñanza	calle	SL	80.96	3
Municipio29	Restauración	calle	SA	69.24	3
Municipio29	Alimentario	calle	individual	9.98	1
Municipio29	Restauración	calle	SA	76.62	4
Municipio29	Alimentario	calle	individual	12.23	1
Municipio29	Sanidad	calle	SL	22.51	3
Municipio29	Otros	calle	SL	12.55	4
Municipio29	Restauración	calle	SL	59.56	4
Municipio29	Sanidad	calle	individual	11.51	1
Municipio29	Alimentario	calle	SL	34.74	3
Municipio29	Menaje	calle	SL	82.15	4
Municipio29	Menaje	calle	individual	37.58	2
Municipio29	Menaje	calle	SA	27.07	3
Municipio29	Otros	calle	SA	16.45	1
Municipio29	Restauración	calle	SA	51.91	4
Municipio29	Menaje	calle	SA	38.09	4
Municipio29	Alimentario	calle	SL	14.30	1
Municipio29	Menaje	calle	SL	96.53	4
Municipio29	Otros	calle	SA	10.53	2
Municipio29	Enseñanza	calle	SL	71.06	5
Municipio29	Menaje	calle	SL	68.50	4
Municipio29	Alimentario	calle	SL	11.86	2
Municipio29	Restauración	calle	SA	35.91	3
Municipio29	Otros	calle	SL	40.58	2

A tibble: 766 × 7

- **Número de registros**: la base de datos contiene un total de **766 registros**, donde cada uno ellos representa una entrada de datos correspondiente a `Municipio29`.

- **Número de columnas**: la estructura de la base de datos cuenta con **7 columnas**, donde cada una de ellas posee atributos específicos relativos al municipio de estudio.

- **Tipología de las variables**: en cuanto a la tipología de las variables, hay almacenados tanto datos de tipo numeric `num` (indicando números decimales), como de tipo character `chr` (reflejando cadenas de texto).

1.4 Detección de Registros Duplicados

En esta parte del estudio, se han identificado los registros duplicados y se han almacenado en `duplicados` para su visualización:

[9]:
```
duplicados<- BBDD_Locales %>%
  filter(duplicated(.)) # para obtener sus duplicados
```

[10]:
```
head(duplicados)
```

A tibble: 2 × 7

municipio <chr>	sector <chr>	situacion <chr>	forma <chr>	superficie <dbl>	trabajadores <dbl>	antiguedad <dbl>
Municipio29	Menaje	calle	SA	16.45	1	16.2
Municipio29	Alimentario	calle	individual	10.30	1	18.1

Tal y como se puede observar, se han detectado **2 registros duplicados**. A partir de aquí, se ha procedido a su eliminación, dejando al actual dataset sin duplicados.

[11]:
```
BBDD_Locales <- BBDD_Locales %>%
    distinct()
```

A continuación, se comprueba si se han eliminado correctamente:

[12]:
```
sum(duplicated(BBDD_Locales))
```

0

1.5 Detección de Valores "NA"

En Variable `superficie` : estimarlo con el estadístico **media**

[13]:
```
na_superficie <- any(is.na(BBDD_Locales$superficie)) # para detectar valores␣
↪"NA"
na_superficie
```

TRUE

Una vez detectados los valores "NA", se realiza la estimación, asegurando excluir los valores faltantes en el cálculo mediante el argumento `na.rm = TRUE` :

[14]:
```
BBDD_Locales <- BBDD_Locales %>%
  mutate(superficie = ifelse(is.na(superficie), mean(superficie, na.rm = TRUE),␣
 ↪superficie)) # para transformar las variables
```

En Variable `trabajadores` : estimarlo con el estadístico **máximo**

[15]:
```
na_trabajadores <- any(is.na(BBDD_Locales$trabajadores)) # para detectar␣
 ↪valores "NA"
na_trabajadores
```

TRUE

En cuanto a la columna `trabajadores` , sus valores "NA" han sido sustituidos por el valor máximo, nuevamente excluyendo los valores faltantes en el cálculo:

[16]:
```
BBDD_Locales <- BBDD_Locales %>%
  mutate(trabajadores = ifelse(is.na(trabajadores), max(trabajadores, na.rm =␣
 ↪TRUE), trabajadores)) # para transformar las variables
```

Finalmente, se comprueba que ya no existen valores vacíos:

[17]:
```
sum(is.na(BBDD_Locales$superficie)) # para contar la cantidad de NA
sum(is.na(BBDD_Locales$trabajadores)) # para contar la cantidad de NA
```

0

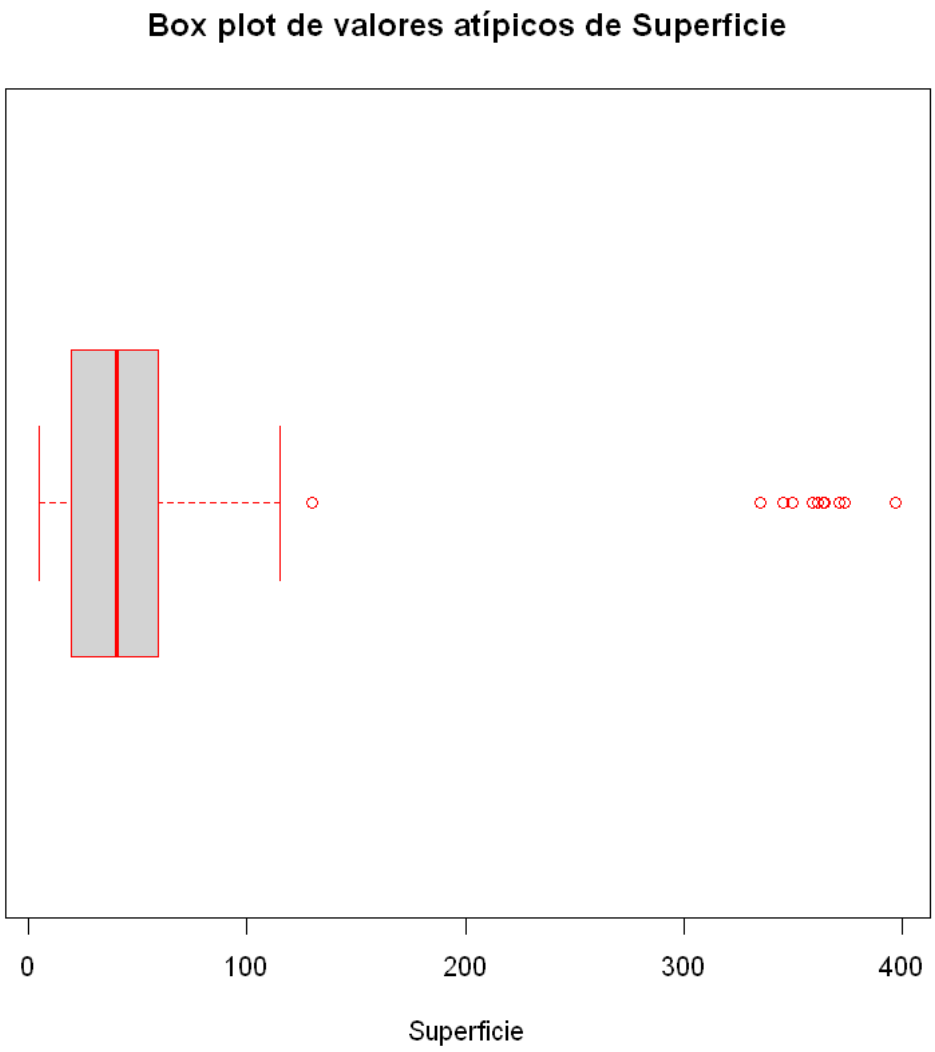
0

1.6 Detección de Valores Atípicos

Box-plot

Se crea un gráfico de los valores atípicos de la columna `superficie`:

[18]:
```
boxplot_superficie <- boxplot(BBDD_Locales$superficie,
                              horizontal = TRUE,
                              border = "red",
                              main = "Box plot de valores atípicos de␣
 ↪Superficie",
                              xlab = "Superficie") # para crear un gráfico␣
 ↪box-plot
```



Se detectan **12 desviaciones**:

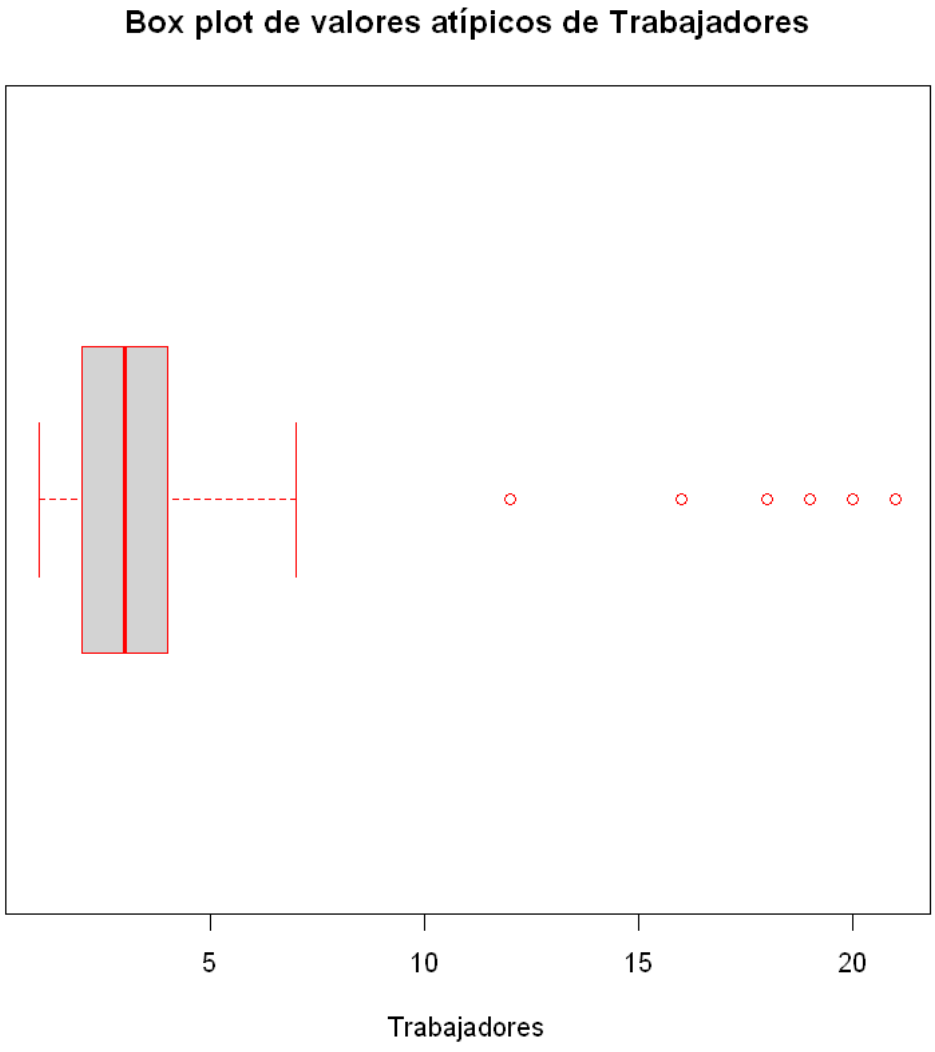
[19]:
```
atipicos_superficie <- boxplot_superficie$out # para extraer los atípicos del␣
↪box-plot
atipicos_superficie
numero_atipicos_superficie <- length(atipicos_superficie) # para conocer el␣
↪número de átipicos
numero_atipicos_superficie
```

1. 358.65 2. 345.27 3. 364.19 4. 363.84 5. 370.98 6. 334.64 7. 349.5 8. 397.07 9. 358.93 10. 361.02 11. 129.65 12. 373.53

12

A continuación, se realiza exactamente el mismo proceso para la columna `trabajadores` .

[20]:
```
boxplot_trabajadores <- boxplot(BBDD_Locales$trabajadores,
                                horizontal = TRUE,
                                border = "red",
                                main = "Box plot de valores atípicos de␣
 ↪Trabajadores",
                                xlab = "Trabajadores") # para crear un gráfico␣
 ↪box-plot
```



Se detectan **14 desviaciones**:

[21]:
```
atipicos_trabajadores <- boxplot_trabajadores$out # para extraer los atípicos␣
 ↪del box-plot
atipicos_trabajadores
```

```
numero_atipicos_trabajadores <- length(atipicos_trabajadores) # para conocer el␣
 ↪número de átipicos
numero_atipicos_trabajadores
```

1. 21 2. 18 3. 21 4. 16 5. 20 6. 18 7. 19 8. 21 9. 18 10. 18 11. 21 12. 18 13. 12 14. 18

14

Z-score

Respecto al método z-score, primeramente se usa la función `scale()` a fin de calcular las desviaciones de `superficie`. Luego, se establece un **umbral de 2** para identificar valores atípicos.

[22]:
```
z_scores_superficie <- scale(BBDD_Locales$superficie) # para calcular las␣
 ↪desviaciones
atipicos_z_score_superficie <- which(abs(z_scores_superficie) > 2)
atipicos_z_score_superficie
numero_atipicos_superficie2 <- length(atipicos_z_score_superficie) # para␣
 ↪conocer el número de átipicos
numero_atipicos_superficie2
```

1. 90 2. 111 3. 186 4. 260 5. 270 6. 306 7. 334 8. 339 9. 347 10. 451 11. 696

11

Ante esto, se observa que la diferencia de desviación entre métodos es de **1**. Esto se debe a que box-plot se basa en cuartiles y rangos intercuartiles (IQR) para identificar valores atípicos, lo que puede llevar a que los valores extremos se consideren atípicos solo si están muy lejos de la mayoría de los datos. El método z-score, por su parte, identifica un número diferente de valores atípicos; ya que se centra en la distancia entre cada punto de datos y la media.

[23]:
```
plot(z_scores_superficie, main = "Z-Scores de Superficie", xlab = "Índice",␣
 ↪ylab = "Z-Score",
     pch = 19, col = ifelse(abs(z_scores_superficie) > 2, "red", "black"))
```

Con la columna `trabajadores` se hace exactamente lo mismo, y esta vez la diferencia ha sido nula:

[24]:
```
z_scores_trabajadores <- scale(BBDD_Locales$trabajadores) # para calcular las␣
 ↪desviaciones
atipicos_z_score_trabajadores <- which(abs(z_scores_trabajadores) > 2)
atipicos_z_score_trabajadores
numeros_atipicos_trabajadores2 <- length(atipicos_z_score_trabajadores) # para␣
 ↪conocer el número de átipicos
numeros_atipicos_trabajadores2
```

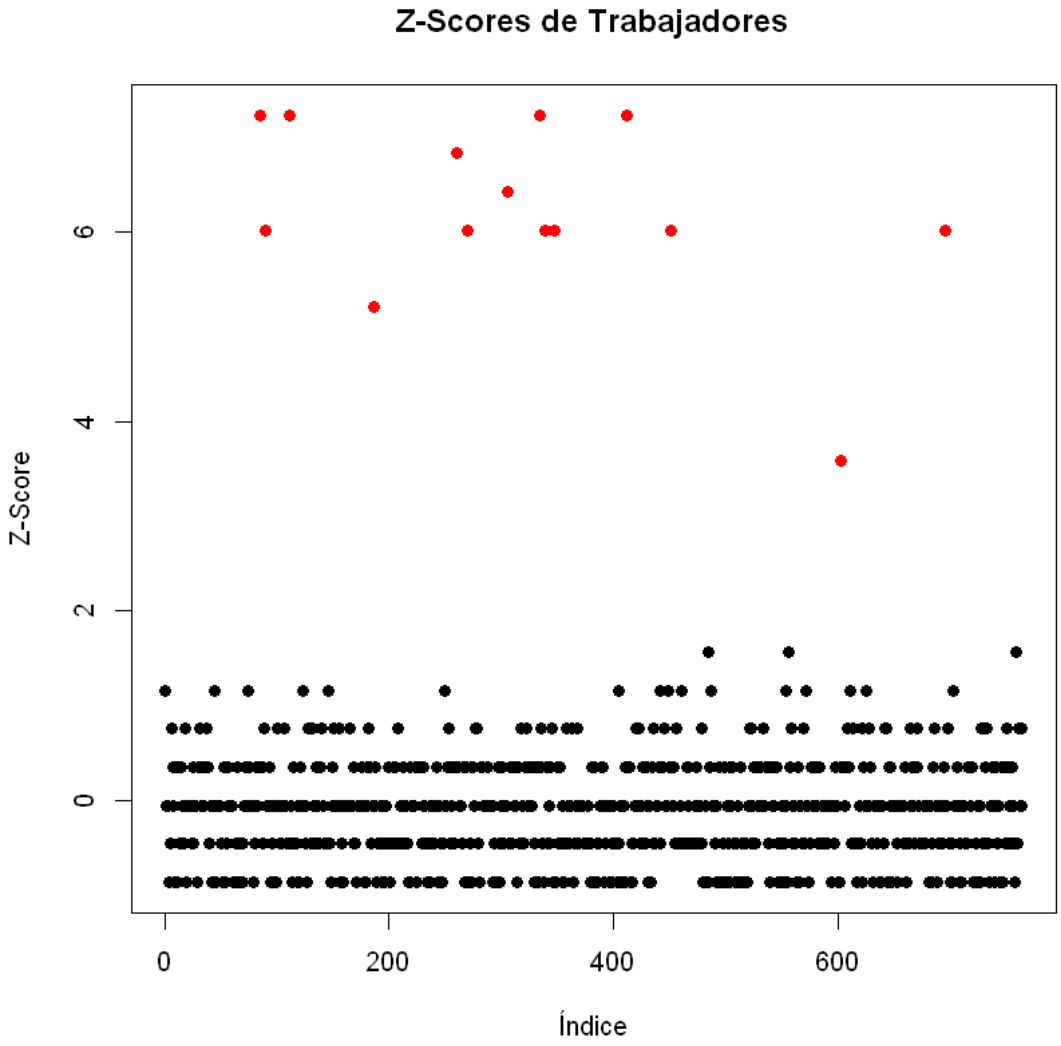
1. 85 2. 90 3. 111 4. 186 5. 260 6. 270 7. 306 8. 334 9. 339 10. 347 11. 412 12. 451 13. 602 14. 696

14

En este caso, puede ser que los datos sigan una distribución relativamente simétrica y los valores

atípicos se encuentren lejos de la media.

[25]:
```
plot(z_scores_trabajadores, main = "Z-Scores de Trabajadores", xlab = "Índice",␣
↪ylab = "Z-Score",
    pch = 19, col = ifelse(abs(z_scores_trabajadores) > 2, "red", "black"))
```



1.7 Algunos Cálculos Estadísticos...

Superficie media por forma mercantil

[26]:
```
resultados_superficie <- BBDD_Locales %>%
  group_by(forma) %>%
  summarize(superficie_media = mean(superficie, na.rm = TRUE))
```

```
resultados_superficie
```

A tibble: 4 × 2

forma	superficie_media
<chr>	<dbl>
SA	59.61605
SL	46.13969
cooperativa	361.38500
individual	30.49525

Antigüedad mínima y máxima por situación del local

[27]:
```
resultados_antiguedad <- BBDD_Locales %>%
  group_by(situacion) %>%
  summarize(
    antiguedad_minima = min(antiguedad, na.rm = TRUE),
    antiguedad_maxima = max(antiguedad, na.rm = TRUE)
  )
resultados_antiguedad
```

A tibble: 2 × 3

situacion	antiguedad_minima	antiguedad_maxima
<chr>	<dbl>	<dbl>
calle	0.5	79.8
centro comercial	0.7	36.9