Cervantes Rubí Brandon

Objetivo.

Encontrar información de interés para predecir la próxima tendencia inmobiliaria en Melbourne.

Características.

El análisis exploratorio de datos es una etapa fundamental para los proyectos que se desarrollan en la actualidad relacionados con la IA actual y todos los subcampos que esta abarca. Se trata de dar un acercamiento a cómo es que están compuestos estos datos; el tipo, su distribución, las dimensiones de las matrices de valores.

El sector inmobiliario de Melbourne, Australia continúa en auge desde hace algunos años. Es de interés conocer la tendencia inmobiliaria en dicha ciudad debido a que cada vez es más difícil adquirir una unidad de 2 dormitorios a un precio razonable.

Fuente de datos

- Rooms: Número de habitaciones.
- Price: Precio en dolares.
- Method: S - propiedad vendida; SP - propiedad vendida antes; PI - propiedad transferida; PN - vendida antes no revelada; SN - vendida no revelada; NB - sin oferta; VB - oferta del proveedor; W - retirada antes de la subasta; SA - vendida después de subasta; SS - vendida después del precio de subasta no revelado. N/A - precio u oferta más alta no disponible.
- Type: br - dormitorio (s); h - casa, cabaña, villa, semi, terraza; u - unidad, dúplex; t - casa adosada; dev site – en desarrollo; o res - otro residencial.
- SellerG: Agente de bienes raíces.
- Date: Fecha de venta.
- Distance: Distancia del CBD (Centro de negocios).
- Regionname: Región general (oeste, noroeste, norte, noreste ...).
- Propertycount: Número de propiedades que existen en el suburbio.
- Bedroom2: Número de dormitorios (de otra fuente).
- Bathroom: Cantidad de baños.
- Car: Número de estacionamientos.
- Landsize: Tamaño del terreno.
- BuildingArea: Tamaño del edificio.
- CouncilArea: Consejo de gobierno de la zona (Municipio).

Desarrollo.

Durante toda la práctica se utilizaron las siguientes bibliotecas; `pandas` para el manejo de los datos, `numpy` para trabajar con vectores/matrices más cómodamente, `pyplot` de `matplotlib` para graficar los resultados del análisis y por último `seaborn` para manejar aún más estilos y graficas en la visualización de datos. Además de que por buenas prácticas se utilizó un alias para cada una de estas bibliotecas con la palabra reservada `as`.

El análisis exploratorio de datos siempre comienza por la lectura de estos, en este caso se hizo por medio de un `csv` con el método `read_csv` de pandas generando de esta manera un `dataframe` el cual se muestra a continuación.

```
DatosMelbourne = pd.read_csv("Datos/melb_data.csv")
DatosMelbourne
```

	Suburb	Address	Rooms	Type	Price	Method	SellerG	Date	Distance	Postcode	...	Bathroom	Car
0	Abbotsford	85 Turner St	2	h	1480000.0	S	Biggin	3/12/2016	2.5	3067.0	...	1.0	1.0
1	Abbotsford	25 Bloomburg St	2	h	1035000.0	S	Biggin	4/02/2016	2.5	3067.0	...	1.0	0.0
2	Abbotsford	5 Charles St	3	h	1465000.0	SP	Biggin	4/03/2017	2.5	3067.0	...	2.0	0.0
3	Abbotsford	40 Federation La	3	h	850000.0	PI	Biggin	4/03/2017	2.5	3067.0	...	2.0	1.0
4	Abbotsford	55a Park St	4	h	1600000.0	VB	Nelson	4/06/2016	2.5	3067.0	...	1.0	2.0

A partir de este punto se comenzó a trabajar con la caracterización de los datos, como primera función se reviso `head`, la cual nos regresa las n primeras líneas que especifiquemos, su contraparte seria `tail`, la cual nos regresa las n ultimas líneas del `dataframe`.

Se opto por dividir el análisis exploratorio de datos en 4 pasos, el primero de ellos tiene que ver con establecer cual es la estructura de datos inicial que se leyó, para esto nos ayudamos de los atributos que ya tiene el dataframe los cuales se detallaran a continuación.

- `shape`: Este atributo nos regresa cual es la dimensión del dataframe que puede ser visto como una matriz, por ejemplo, en esta ocasión las dimensiones son de 13580 x 21 lo que quiere decir que se tienen 13580 renglones o registros de las casas y por cada uno de estos registros se tienen 21 valores los cuales corresponden a las variables iniciales.

- `dtypes`: Este atributo nos regresa el tipo de dato de cada una de las variables o columnas en el dataframe, en este caso regresa 21 valores los cuales varían entre float y object que corresponden en su mayoría a cadenas.

El segundo paso dentro del análisis de datos corresponde a la identificación de datos faltantes ya que es muy común que al momento de leer con detenimiento los datos nos encontremos con que muchos registros omiten algún valor entre sus variables, es decir, son atributos opcionales o simplemente no se registraron correctamente en su momento. Para ello se hace una sumatoria sobre cada una de las variables para obtener cuantos valores nulos se tienen por cada columna con la siguiente línea.

```
DatosMelbourne.isnull().sum()
```

Lo cual nos regresa la lista de variables con sus valores nulos, con ello podemos saber que el área de construcción (`BuildingArea`) es la columna que cuenta con mas valores nulos con 6450 valores. Recordando que se tienen poco mas de 13400 registros, podemos decir que poco mas de la mitad de los datos se encuentran completos para generar el modelo.

En estos casos es importante conocer si es que la variable es importante dentro del contexto para el cual se está analizando, si NO lo es, entonces seria factible eliminar todos los registros del conjunto de datos que omitan este valor, si lo es, entonces habrá que dejar esos registros y buscar nuevas soluciones como por ejemplo la creación de un modelo secundario que se encargue de interpolar estos valores.

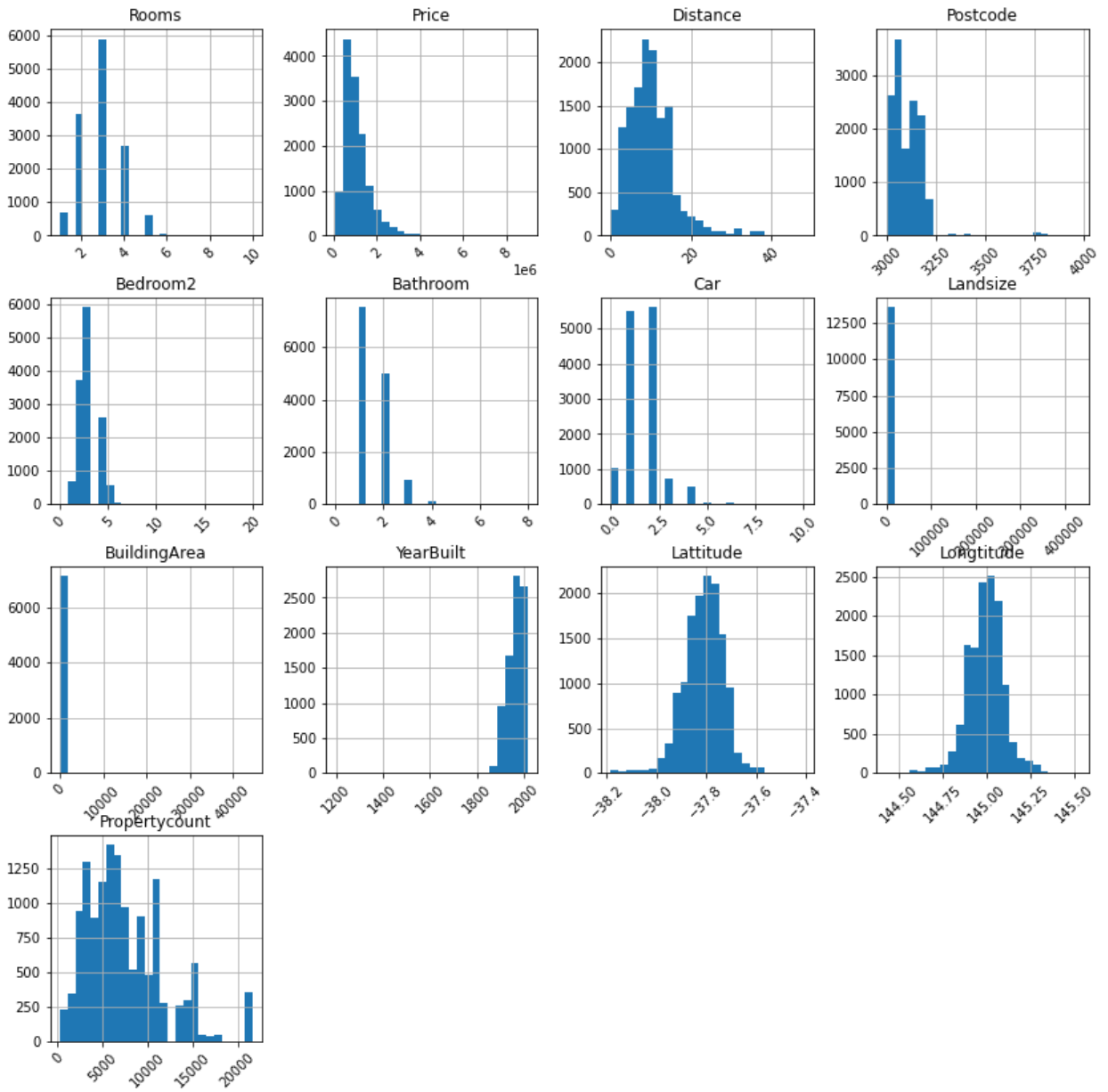
También se revisó el método `info`, el cual es similar a `dtype`, con la diferencia de que además de regresarnos el tipo de dato, nos regresa la cuenta de los valores que no son nulos.

```
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Suburb         13580 non-null  object
 1   Address        13580 non-null  object
 2   Rooms          13580 non-null  int64
 3   Type           13580 non-null  object
```

El tercer paso se centra en la detección de valores atípicos, es decir, valores que no hacen mucho sentido cuando se comparan con la mayoría de los valores que corresponden al conjunto de datos, valores que se alejan mucho de la media. Para identificarlos en primera instancia se utilizan graficas como histogramas y diagramas de cajas, sin embargo, para confirmarlos se utilizan estadísticas como valores mínimos y máximos en comparación con la media.

```
DatosMelbourne.hist(figsize=(14,14), xrot=45, bins=25)
plt.show()
```

Se hicieron histogramas para cada una de las variables numéricas con el módulo `pyplot` de `matplotlib`, con `figsize` y `xrot` se modifico el tamaño de los histogramas y con `bins` podemos hacer aun mas preciso el histograma que estemos creando `bins = 25` subdivide el histograma en 25 barras.



En primera instancia los valores atípicos también se pueden tratar de errores al momento de tomar la medición, sin embargo, hay que identificar cuales son para corroborar si en efecto se trata de un error o de un valor que nos pueda indicar que los datos están corruptos. En la mayoría de los casos simplemente se elimina para que no afecte mucho al modelo que se desarrolle, o en el mejor de los casos se puede corregir por el valor correcto.

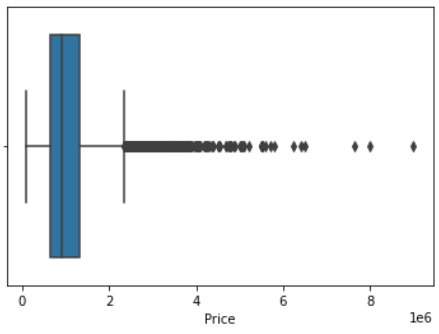
En estos datos, los histogramas de `BuildingArea`, Price y `LandSize` tienen valores sesgados a la izquierda, por otro lado, `YearBuild` se encuentra sesgado hacia la derecha. Estos sesgos indican valores atípicos.

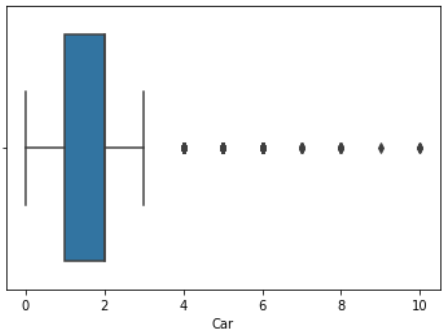
El método `describe` de pandas nos muestra un resumen de las estadísticas de cada una de las columnas indicando la cuenta de los registros, la media, la desviación estándar, el valor mínimo, los cuartiles que también indican la mediana al 50%, y el valor máximo.

	Rooms	Price	Distance	Postcode	Bedroom2	Bathroom	Car	Landsize	BuildingArea
count	13580.000000	1.358000e+04	13580.000000	13580.000000	13580.000000	13580.000000	13518.000000	13580.000000	7130.000000
mean	2.937997	1.075684e+06	10.137776	3105.301915	2.914728	1.534242	1.610075	558.416127	151.967650
std	0.955748	6.393107e+05	5.868725	90.676964	0.965921	0.691712	0.962634	3990.669241	541.014538
min	1.000000	8.500000e+04	0.000000	3000.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	2.000000	6.500000e+05	6.100000	3044.000000	2.000000	1.000000	1.000000	177.000000	93.000000
50%	3.000000	9.030000e+05	9.200000	3084.000000	3.000000	1.000000	2.000000	440.000000	126.000000
75%	3.000000	1.330000e+06	13.000000	3148.000000	3.000000	2.000000	2.000000	651.000000	174.000000
max	10.000000	9.000000e+06	48.100000	3977.000000	20.000000	8.000000	10.000000	433014.000000	44515.000000

Otra forma muy común de detectar los valores atípicos son los gráficos de caja o boxplots, este tipo de grafica se basa en los cuartiles pues la caja abarca los cuartiles 1, 2 y 3 (25%, 50% y 75%). El siguiente código muestra este tipo de grafica para cada una de las variables sobre las cuales se sospecha la existencia de valores atípicos.

```
VariablesValoresAtipicos = ['Price', 'Car', 'Landsize', 'BuildingArea', 'YearBuilt']
for col in VariablesValoresAtipicos:
  sns.boxplot(col, data=DatosMelbourne)
  plt.show()
```





A modo de ejemplo se muestran los gráficos de caja de las variables precio y carro (Estacionamiento) y es aquí donde tendremos que hacer un análisis ya que en efecto pueden existir casas mas caras que otras y la media puede estar muy alejada de la casa mas cara, por el lado de los estacionamientos pueden existir casas que cuenten con 10 espacios de estacionamientos, sin embargo, son muy pocas casas las que presentan esta condición.

Hasta este punto todas las estadísticas se han centrado en las variables numéricas, el ultimo paso tiene que ver con las variables categóricas, para ello la frecuencia de aparición, un conteo de registros y el número de categorías por variable son métricas que nos pudieran ayudar a describir de mejor manera los registros.

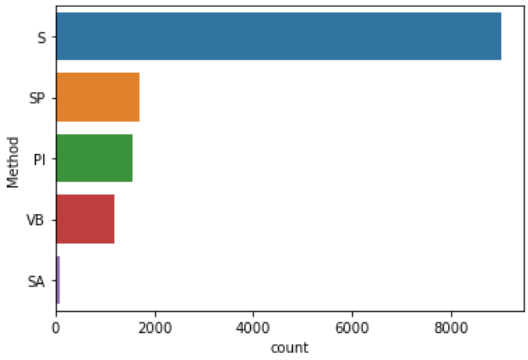
Para ello, se hace de manera muy similar a los valores numéricos con `describe`, pero en la sección de parámetros se especifica que queremos trabajar con los de tipo de dato objetos. Y las estadísticas que se nos mostrarán serán las que se mencionaron anteriormente.

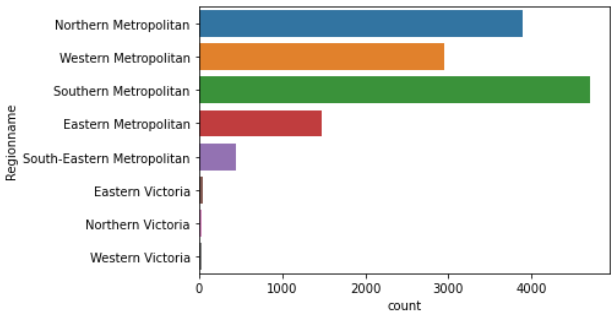
```
DatosMelbourne.describe(include='object')
```

	Suburb	Address	Type	Method	SellerG	Date	CouncilArea	Regionname
count	13580	13580	13580	13580	13580	13580	12211	13580
unique	314	13378	3	5	268	58	33	8
top	Reservoir	36 Aberfeldie St	h	S	Nelson	27/05/2017	Moreland	Southern Metropolitan
freq	359	3	9449	9022	1565	473	1163	4695

Por ejemplo, el nombre de la región (`Regionname`) cuenta con 13580 registros, 8 categorías donde la que mas se repite es Southern Metropolitan y lo hace 4695 veces.

Podemos además graficar histogramas por cada una de estas variables para visualizar esta frecuencia, a continuación, se muestran los histogramas de Method y Regionname.





Con estas graficas además podemos ver cual es la frecuencia de las demás categorías, no solamente el valor de la que más se repita. Además, es posible agrupar por categoría y obtener la media de aparición.

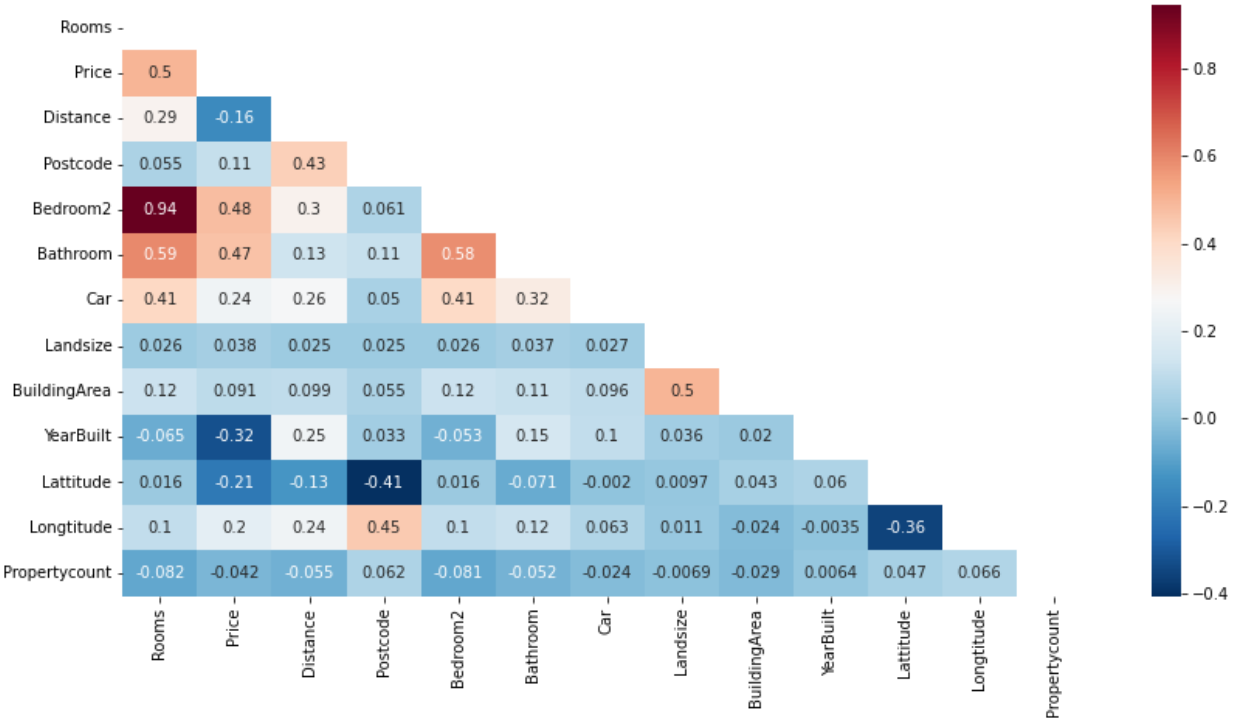
```
for col in DatosMelbourne.select_dtypes(include='object'):
  if DatosMelbourne[col].nunique() < 10:
    display(DatosMelbourne.groupby(col).agg(['mean']))
```

El último paso dentro del análisis de correlación de variables tiene que ver con la identificación de correlaciones para medir el grado de similitud entre pares de variables, es decir que tan parecido se comportan. Estas medidas se pueden obtener en una matriz de correlaciones la cual tiene índices que van de -1 a 1, donde los mas cercanos a 1 indican un valor de correlación fuerte y los mas cercanos a 0 indican un nivel de correlación débil.

A continuación, se muestra la matriz de correlaciones.

	Rooms	Price	Distance	Postcode	Bedroom2	Bathroom	Car	Landsize	BuildingArea	YearBuilt	Lattitude
Rooms	1.000000	0.496634	0.294203	0.055303	0.944190	0.592934	0.408483	0.025678	0.124127	-0.065413	0.015948
Price	0.496634	1.000000	-0.162522	0.107867	0.475951	0.467038	0.238979	0.037507	0.090981	-0.323617	-0.212934
Distance	0.294203	-0.162522	1.000000	0.431514	0.295927	0.127155	0.262994	0.025004	0.099481	0.246379	-0.130723
Postcode	0.055303	0.107867	0.431514	1.000000	0.060584	0.113664	0.050289	0.024558	0.055475	0.032863	-0.406104
Bedroom2	0.944190	0.475951	0.295927	0.060584	1.000000	0.584685	0.405325	0.025646	0.122319	-0.053319	0.015925
Bathroom	0.592934	0.467038	0.127155	0.113664	0.584685	1.000000	0.322246	0.037130	0.111933	0.152702	-0.070594
Car	0.408483	0.238979	0.262994	0.050289	0.405325	0.322246	1.000000	0.026770	0.096101	0.104515	-0.001963
Landsize	0.025678	0.037507	0.025004	0.024558	0.025646	0.037130	0.026770	1.000000	0.500485	0.036451	0.009695
BuildingArea	0.124127	0.090981	0.099481	0.055475	0.122319	0.111933	0.096101	0.500485	1.000000	0.019665	0.043420
YearBuilt	-0.065413	-0.323617	0.246379	0.032863	-0.053319	0.152702	0.104515	0.036451	0.019665	1.000000	0.060445
Lattitude	0.015948	-0.212934	-0.130723	-0.406104	0.015925	-0.070594	-0.001963	0.009695	0.043420	0.060445	1.000000
Longtitude	0.100771	0.203656	0.239425	0.445357	0.102238	0.118971	0.063395	0.010833	-0.023810	-0.003470	-0.357634
Propertycount	-0.081530	-0.042153	-0.054910	0.062304	-0.081350	-0.052201	-0.024295	-0.006854	-0.028840	0.006361	0.047086

Como se ve, la diagonal principal se compone de índices 1, el cual indica un nivel de correlación alto ya que se compara la variable con si misma. Al ser una matriz simétrica podemos tomar solamente la parte superior o inferior y nos describirá la misma información. Para hacerlo un poco mas sencillo de leer podemos pintar la matriz de correlaciones con un mapa de calor, esta muestra colores más cálidos (naranjas, rojos) cuando el índice se acerca a 1 o al valor máximo y más fríos (azules) cuando los valores se acercan a -1 o al valor mínimo. El blanco indica una nula correlación, es decir, no hay similitud entre las variables con un índice de 0.



Conclusiones

El análisis exploratorio de datos es el primer paso que se tiene que hacer una vez se cuente con los datos para poder tener un primer acercamiento a como es que se comporta el conjunto de datos. En primera instancia cual es su dimensionalidad y sus tipos de datos para después obtener estadísticas tanto para las variables numéricas (Medias, Desviaciones, Cuartiles, etc.) como para las variables categóricas (Frecuencias de aparición, numero de categorías, etc.).

Para visualizar todas estas estadísticas de una mejor manera es factible utilizar graficas como histogramas, gráficos de caja o mapas de calor. En el caso de las variables numéricas es importante detectar valores atípicos ya que pueden indicar anomalías dentro del conjunto de datos y de ser posible corregirlos para prevenir eliminarlos. Además, tenemos que identificar la cantidad de valores nulos y en que variables se encuentran para poder determinar si es factible eliminarlos u optar por otras soluciones para generar nuevos datos.