**Estefania Alzate Daza**

**1094928754**

# Evidencia AA3-EV02

En el presente informe realizará el análisis al caso de estudio de venta de inmuebles, inicialmente se cargará la información, posteriormente se preparará el dataset, si evidenciarán las medidas de tendencia central y se visualizarán gráficos:

- Importación de librerías y cargue del dataset

```
[49]: import pandas as pd
      import seaborn as sns
      import matplotlib.pyplot as plt
      from sklearn.preprocessing import StandardScaler

      data=pd.read_csv('Data_Caso_Propuesto.csv')
```

- Visualización de campos y tipos de datos, además observación de la posible existencia de datos nulos

```
data.info()
data.isnull().sum()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 463 entries, 0 to 462
Data columns (total 12 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Codigo                  463 non-null    int64
 1   Ciudad                  463 non-null    object
 2   Departamento            463 non-null    object
 3   Barrio                  40 non-null     object
 4   Direccion               463 non-null    object
 5   Area Terreno            463 non-null    float64
 6   Area Construida         463 non-null    float64
 7   Detalle Disponibilidad  463 non-null    object
 8   Estrato                 463 non-null    object
 9   Precio                  463 non-null    float64
 10  Tipo de Inmueble        463 non-null    object
 11  Datos Adicionales       118 non-null    object
dtypes: float64(3), int64(1), object(8)
memory usage: 43.5+ KB
```

```
Codigo                    0
Ciudad                    0
Departamento              0
Barrio                  423
Direccion                 0
Area Terreno              0
Area Construida           0
Detalle Disponibilidad    0
Estrato                   0
Precio                    0
Tipo de Inmueble          0
Datos Adicionales       345
dtype: int64
```

- Se evidencia que las columnas de barrio y datos adicionales poseen demasiados nulos y la dirección es un texto que prácticamente es único por registro, por lo tanto, eliminamos estas tres columnas antes de proceder con el análisis y también eliminamos los datos duplicados, después de eliminar los duplicados vemos que la cantidad de datos no ha cambiado, eso quiere decir que no existe duplicidad

```
data.drop(['Barrio','Direccion','Datos Adicionales'], axis=1, inplace=True)
data.head(5)
```

| | Codigo | Ciudad | Departamento | Area Terreno | Area Construida | Detalle Disponibilidad | Estrato | Precio | Tipo de Inmueble |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 17180 | BOGOTA | CUNDINAMARCA | 0.00 | 0.0 | COMERCIALIZABLE CON RESTRICCION | TRES | 2.958081e+10 | LOTE COMERCIAL |
| **1** | 19292 | BOGOTA | CUNDINAMARCA | 0.00 | 0.0 | COMERCIALIZABLE | COMERCIAL | 1.646059e+10 | EDIFICIO |
| **2** | 19292 | BOGOTA | CUNDINAMARCA | 0.00 | 0.0 | COMERCIALIZABLE VENTA ANTICIPADA | COMERCIAL | 1.646059e+10 | EDIFICIO |
| **3** | 2575 | SOGAMOSO | BOYACÁ | 1655.08 | 7269.0 | COMERCIALIZABLE CON RESTRICCION | CUATRO | 1.376828e+10 | CLINICA |
| **4** | 11409 | BUGA | VALLE DEL CAUCA | 3217197.00 | 22724.0 | COMERCIALIZABLE FIDUCIA | RURAL | 4.523379e+10 | LOTE MIXTO |

```
data.drop_duplicates(inplace=True)
data.info()
```

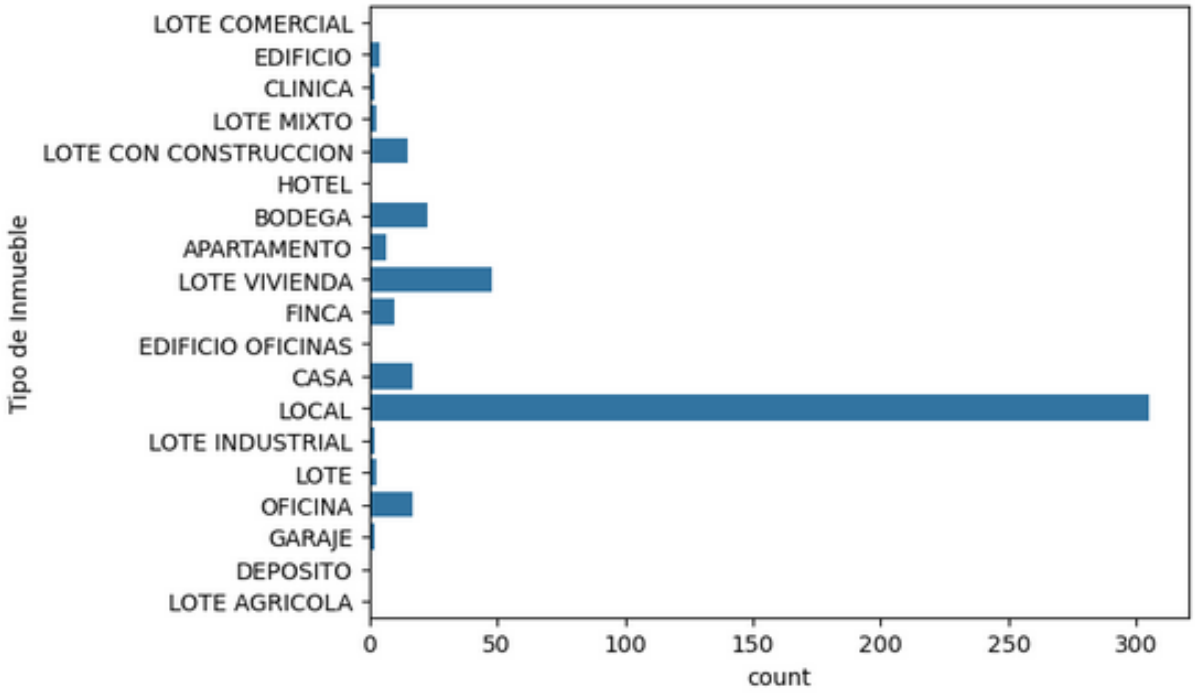
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 463 entries, 0 to 462
Data columns (total 9 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Codigo                 463 non-null    int64
 1   Ciudad                 463 non-null    object
 2   Departamento           463 non-null    object
 3   Area Terreno           463 non-null    float64
 4   Area Construida        463 non-null    float64
 5   Detalle Disponibilidad 463 non-null    object
 6   Estrato                463 non-null    object
 7   Precio                 463 non-null    float64
 8   Tipo de Inmueble       463 non-null    object
dtypes: float64(3), int64(1), object(5)
memory usage: 32.7+ KB
```

- Imprimimos las medidas de tendencia central, es esta tabla podemos ver cantidad de datos, la media, desviación estándar, cuartiles, dato máximo y mínimo; posteriormente imprimimos una grafica de barras donde evidenciamos que el tipo de inmueble con mayor cantidad de datos son los locales

```
data.describe()
```

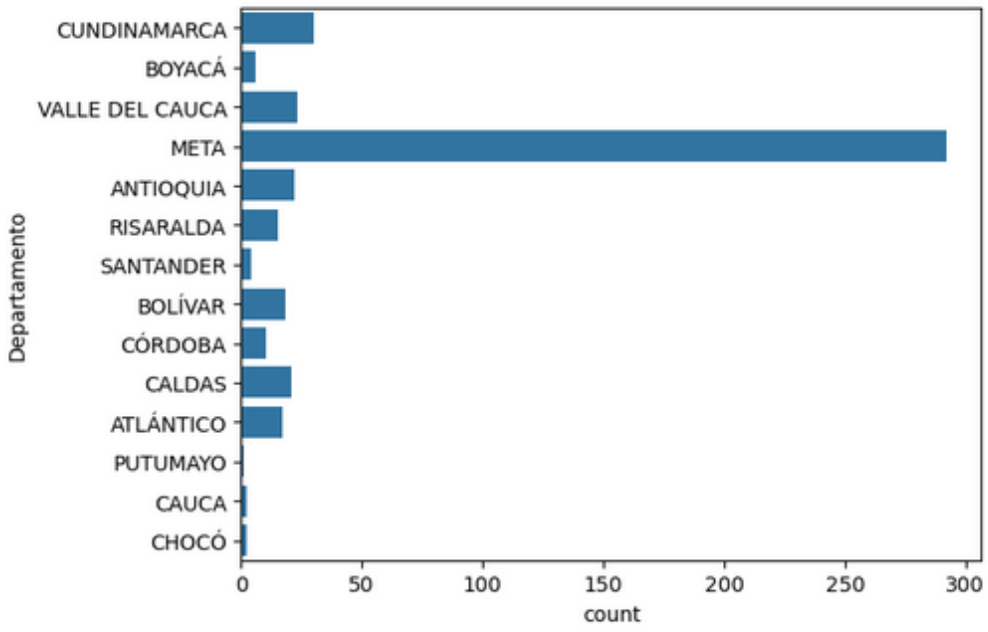
| | Codigo | Area Terreno | Area Construida | Precio |
|---|---|---|---|---|
| **count** | 463.000000 | 4.630000e+02 | 463.000000 | 4.630000e+02 |
| **mean** | 18003.151188 | 1.515204e+04 | 87.517279 | 6.672032e+08 |
| **std** | 1992.191499 | 1.827101e+05 | 1137.469077 | 3.272992e+09 |
| **min** | 2575.000000 | 0.000000e+00 | 0.000000 | 4.650000e+06 |
| **25%** | 18184.500000 | 0.000000e+00 | 0.000000 | 1.230500e+07 |
| **50%** | 18332.000000 | 0.000000e+00 | 0.000000 | 1.587000e+07 |
| **75%** | 18539.500000 | 0.000000e+00 | 0.000000 | 1.379955e+08 |
| **max** | 19344.000000 | 3.217197e+06 | 22724.000000 | 4.523379e+10 |

```
plt.Figure(figsize=(10,7))
sns.countplot(data['Tipo de Inmueble'])
plt.show()
```

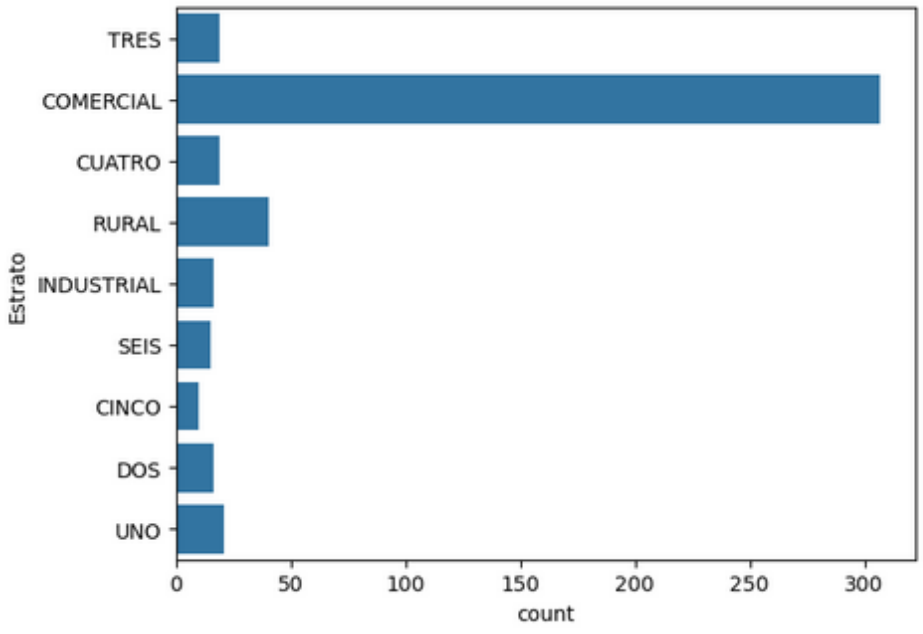


- Realizamos grafica de barras con el departamento donde evidenciamos que el de mayor influencia es el Meta y con respecto al estado donde vemos que la categorización de uso comercial es la mayor

```
plt.Figure(figsize=(10,7))
sns.countplot(data['Departamento'])
plt.show()
```



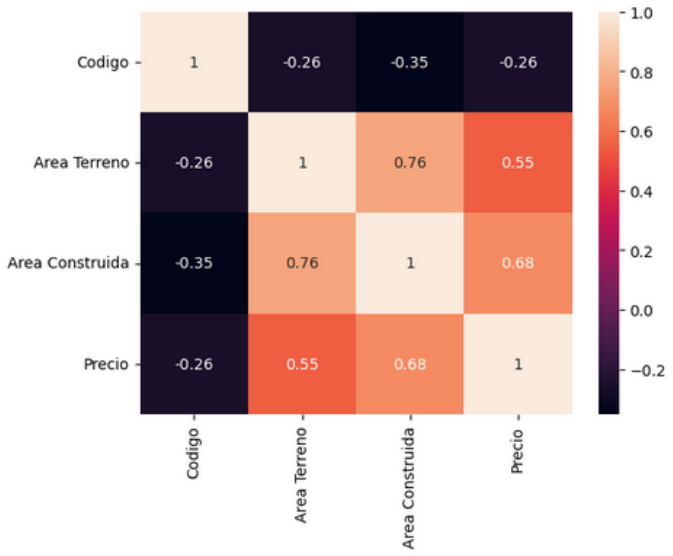

```
plt.Figure(figsize=(10,7))
sns.countplot(data['Estrato'])
plt.show()
```



- Efectuamos una matriz de correlación donde podemos que las variables que más tienen relación el área del terreno y área construida, además de los mismos campos del área con el precio de venta

```
correlacion = data.corr(numeric_only=True)
correlacion
sns.heatmap(correlacion,xticklabels=correlacion.columns,yticklabels=correlacion.columns,annot=True)
```

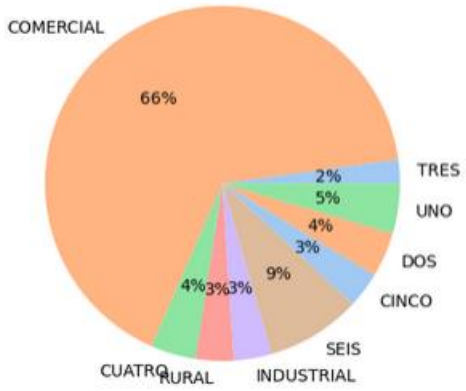
```
<Axes: >
```




- Hacemos un diagrama de pastel con los estratos donde podemos verde manera porcentual que la categorización comercial es mayor y corresponde al 66% del dataset

```
total_estrato = data['Estrato'].groupby(data['Estrato']).count()
etiquetas = data['Estrato'].unique()
print(etiquetas)
#etiquetas = etiquetas['Estrato']
colors = sns.color_palette('pastel')[0:6]
plt.pie(total_estrato, labels = etiquetas, colors = colors, autopct='%.0f%%')
plt.show()
```

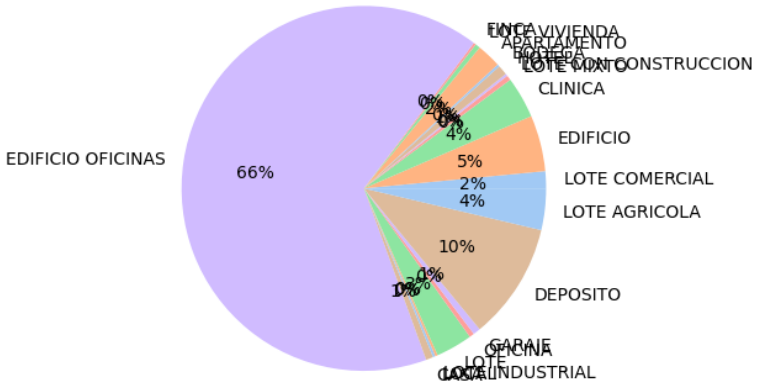
```
['TRES' 'COMERCIAL' 'CUATRO' 'RURAL' 'INDUSTRIAL' 'SEIS' 'CINCO' 'DOS'
 'UNO']
```



- Por ultimo tenemos el diagrama de pastel por tipo de inmueble donde vemos que edificios y oficinas son el mayor porcentaje, posterior a ello tenemos un diagrama de bigotes del tipo de inmueble oficina por los departamentos

```
total_estrato = data['Tipo de Inmueble'].groupby(data['Tipo de Inmueble']).count()
etiquetas = data['Tipo de Inmueble'].unique()
print(etiquetas)
#etiquetas = etiquetas['Estrato']
colors = sns.color_palette('pastel')[0:6]
plt.pie(total_estrato, labels = etiquetas, colors = colors, autopct='%.0f%%')
plt.show()
```

```
['LOTE COMERCIAL' 'EDIFICIO' 'CLINICA' 'LOTE MIXTO'
 'LOTE CON CONSTRUCCION' 'HOTEL' 'BODEGA' 'APARTAMENTO' 'LOTE VIVIENDA'
 'FINCA' 'EDIFICIO OFICINAS' 'CASA' 'LOCAL' 'LOTE INDUSTRIAL' 'LOTE'
 'OFICINA' 'GARAJE' 'DEPOSITO' 'LOTE AGRICOLA']
```




```
filtro = data[data['Tipo de Inmueble'] == 'OFICINA']
fumador_valor = sns.boxplot(x=filtro["Departamento"], y=filtro["Precio"])
```