

```
In [12]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

In [41]: df = pd.read_excel('inmuebles.xlsx')

In [51]: df.head(5)

Out[51]:
   Referencia  FechaAlta  Tipo Operación  Ciudad  Superficie  Precio Venta  Fecha Venta  Vendedor  Estatus  Días para Vender
0      1      2016-01-18  Estacionamiento  Alquiler  Cancún      215      1154980      2017-01-14  Joaquín  Vendida      362
1      2      2016-01-20      Oficina  Alquiler  Tijuana      287      2851058      2017-01-02  Joaquín  Vendida      348
2      3      2016-01-20      Local  Venta  Monterrey      40      321680      2017-01-11  Jesús  Vendida      387
3      4      2016-01-04  Industrial  Venta  Cancún      178      1142938      2017-01-23  Joaquín  Vendida      366
4      5      2016-02-03  Departamento  Venta  Monterrey      275      2141700      2017-01-08  Jesús  Vendida      340

In [16]: df.shape
Out[16]: (2651, 11)

In [17]: #Información de los datos
df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2651 entries, 0 to 2650
Data columns (total 11 columns):
 #   Column                Non-Null Count  Dtype
---  ---
 0   Referencia            2651 non-null    int64
 1   FechaAlta            2651 non-null    datetime64[ns]
 2   Tipo                 2651 non-null    object
 3   Operación            2651 non-null    object
 4   Ciudad              2651 non-null    object
 5   Superficie           2651 non-null    int64
 6   Precio Venta         2651 non-null    int64
 7   Fecha Venta          2651 non-null    datetime64[ns]
 8   Vendedor             2387 non-null    object
 9   Estatus              2651 non-null    object
10   Días para Vender     2651 non-null    int64
11  dtypes: datetime64[ns] (2), int64 (4), object (5)
memory usage: 227.9+ KB

In [19]: # Campos de la base de datos
for col in df.columns:
    print(col)

Referencia
FechaAlta
Tipo
Operación
Ciudad
Superficie
Precio Venta
Fecha Venta
Vendedor
Estatus
Días para Vender

In [10]: #
Out[10]:
   Superficie  Precio Venta  Días para Vender
count  2651.000000  2.651000e+03  2651.000000
mean      171.201811  1.211656e+06  365.289702
std       74.937985  6.261026e+05  200.002488
min        40.000000  1.638200e+05  90.000000
25%      106.000000  6.986235e+05  190.000000
50%      172.000000  1.140876e+06  323.000000
75%      237.000000  1.642624e+06  542.000000
max      300.000000  2.977318e+06  730.000000

In [11]: #Precio promedio de venta por ciudad: Examina los precios de venta promedio para cada ciudad.
df.groupby('Ciudad')['Precio Venta'].mean().round()

Out[11]:
Ciudad
Cancún      1180772.0
Ciudad de México  1213952.0
Monterrey   1213952.0
Tijuana     1233933.0
Name: Precio Venta, dtype: float64

In [16]: valor_minimo = df['Precio Venta'].min()
valor_maximo = df['Precio Venta'].max()

In [21]: valor_minimo

Out[21]: 163920

In [23]: valor_maximo

Out[23]: 2977318

In [29]: # Promedio
promedio = df['Precio Venta'].mean()
print('Promedio:', promedio)

# Mediana
mediana = df['Precio Venta'].median()
print('Mediana:', mediana)

# Moda
moda = df['Precio Venta'].mode()
print('Moda:', moda)

Promedio: 1211656.4907582044
Mediana: 1140876.0
Moda: 0
338670
1
416430
2
119236
3
446940
4
244896
5
574200
6
792215
7
91348
8
913778
9
1192048
10
1199394
11
1342278
12
2162223
Name: Precio Venta, dtype: int64

In [27]: #Gráfico de líneas para mostrar las ventas en cada mes

import matplotlib.pyplot as plt

# Crear el gráfico de líneas
ax = df.groupby(df['Fecha Venta'].dt.to_period('M')).size().plot(kind='line')

# Obtener los datos de X y Y del gráfico
x_values = ax.get_lines()[0].get_xdata()
y_values = ax.get_lines()[0].get_ydata()

# Agregar etiquetas de datos
for x, y in zip(x_values, y_values):
    plt.annotate(f'{int(y)}', xy=(x, y), xytext=(0, 5), textcoords='offset points', ha='center')

# Rotar y acercar de la línea
ax = df.groupby(df['Fecha Venta'].dt.to_period('M')).size().plot(kind='line', color='red', linewidth=5)

# Personalizar el gráfico
plt.title('Ventas por Meses')
plt.xlabel('Meses')
plt.ylabel('Número de Ventas')
plt.grid(True)
plt.show()

Ventas por Meses

Número de Ventas
160
140
120
100
80
60
40
20
0
Jan 2017 Apr Jul Oct Jan 2018 Apr Jul Oct Jan 2019 Apr
86 90 97 98 98 90 93 100 97 75 101 131 151 124 184 23

In [86]: #Existe una variabilidad en las ventas mensuales que probablemente puede estar relacionado con aspectos económicos,
#estacionales o del mercado. Se quiere resaltar que hay una tendencia de crecimiento finalizando el año de 2018 y esperando el 2019.
#Después de puede observar una caída fuerte. Lo que implica un análisis mas profundo para conocer posibles causas.

In [131]: # Gráfico de dispersión
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='Superficie', y='Precio Venta')
plt.title('Relación entre Superficie y Precio Venta')
plt.xlabel('Superficie (m²)')
plt.ylabel('Precio Venta ($)')
plt.show()

Relación entre Superficie y Precio Venta

Precio Venta ($)
3.0
2.5
2.0
1.5
1.0
0.5
0
50 100 150 200 250 300
Superficie (m²)

In [134]: #Este es un estilo de consulta con el número que ponga. Lo va a buscar a la base de datos y selecciona toda la fila
df.iloc[174]

Out[14]:
Referencia      875
FechaAlta      2017-03-20 00:00:00
Tipo            Oficina
Operación       Alquiler
Ciudad          Tijuana
Superficie      184
Precio Venta    1152718
Fecha Venta     2017-07-31 00:00:00
Vendedor        Pedro
Estatus         Vendida
Días para Vender 133
Name: 874, dtype: object

In [151]: #Busca desde el 18 hasta el 20
df.iloc[18:20]

Out[151]:
   Referencia  FechaAlta  Tipo Operación  Ciudad  Superficie  Precio Venta  Fecha Venta  Vendedor  Estatus  Días para Vender
18      19  2016-03-02      Oficina  Venta  Cancún      172      1063648      2017-02-13  María  Vendida      348
19      20  2016-03-06  Estacionamiento  Venta  Monterrey      169      1370759      2017-01-14  Luisa  Vendida      314

In [164]: #Busca en la columna Vendedor, que este en la fila 500
df['Vendedor'][500]

Out[164]: 'Luisa'

In [170]: #Resumen estadístico
df.describe()

Out[170]:
   Referencia  FechaAlta  Tipo Operación  Ciudad  Superficie  Precio Venta  Fecha Venta  Vendedor  Estatus  Días para Vender
1597      1598  2016-12-14      Casa  Venta  Ciudad de México      216      2046384      2018-07-03  Pedro  Vendida      566
320      321  2016-08-30      Oficina  Venta  Ciudad de México      274      1577892      2017-05-11  Joaquín  Vendida      254
2282      2283  2017-08-13      Local  Venta  Cancún      98      957284      2016-11-11  NaN  En Proceso      455
1138      1139  2016-08-22      Casa  Venta  Monterrey      274      2584780      2016-02-23  María  Vendida      611
284      285  2016-09-20  Industrial  Venta  Monterrey      119      711382      2017-08-01  Luisa  Vendida      346

In [40]: df['Estatus'].value_counts().plot(kind='pie',
                                             title='Estatus',
                                             autopct='%1.1f%%',
                                             startangle=90, # Muestra el porcentaje con 1 decimal
                                             figsize=(5, 5), # Tamaño del gráfico
                                             wedgeprops={'edgecolor': 'black'}) # Bordes de los sectores

Out[40]: <Axes: title='center': 'Estatus', ylabel='count'>

Estatus

En Proceso      10.0%
Vendida         90.0%
Count
10
9
8
7
6
5
4
3
2
1
0
Estatus

In [176]: import matplotlib.pyplot as plt
import pandas as pd

# Contar la frecuencia de cada tipo de inmueble
conteo_tipos = df['Tipo'].value_counts()

# Crear el gráfico de barras
conteo_tipos.plot(kind='bar', title='Tipos de Inmuebles')

# Agregar etiquetas a las barras
for i, n in enumerate(conteo_tipos):
    plt.text(i, conteo_tipos.iloc[i], str(conteo_tipos.iloc[i]), ha='center', va='bottom')

# Rotar de los ejes
plt.xlabel('Inmuebles')
plt.ylabel('Cantidad de Inmuebles')
plt.xticks(rotation=45)
# Mostrar el gráfico
plt.show()

Tipos de Inmuebles

Cantidad de inmuebles
400
350
300
250
200
150
100
50
0
0
Departamento Industrial Estacionamiento Oficina Casa Local Terreno
408 391 381 379 374 362 356

In [45]: #Alquiler vs Ventas
df['Operación'].value_counts().plot(kind='pie',
                                     title='Distribuciones de operaciones',
                                     autopct='%1.1f%%', # Muestra el porcentaje con 1 decimal
                                     figsize=(6, 6),
                                     wedgeprops={'edgecolor': 'black'})

Out[45]: <Axes: title='center': 'Distribuciones de operaciones', ylabel='count'>

Distribuciones de operaciones

Alquiler      1308
En Proceso    1343
Vendida       1398
Count
13
12
11
10
9
8
7
6
5
4
3
2
1
0
Operación

In [181]: #Este gráfico de pie nos muestra un breve resumen de las ventas y los alquileres. Donde las ventas superan ligeramente
#las que son de alquiler.

In [1]: #Gráfico de Barras horizontales de vendedores

import matplotlib.pyplot as plt

# Contar la frecuencia de cada tipo de inmueble
conteo_tipos = df['Vendedor'].value_counts()

# Crear el gráfico de barras horizontales
conteo_tipos.plot(kind='barh', title='Vendedores')

# Agregar etiquetas a las barras (ahora en el eje horizontal)
for i, v in enumerate(conteo_tipos):
    plt.text(i, str(v), str(v), va='center')

# Rotar de los ejes
plt.xlabel('Cantidad de Ventas')
plt.ylabel('Nombre del Vendedor')

# Mostrar el gráfico
plt.show()

Vendedores

Todos los vendedores tienen un desempeño bastante cercano, donde el primero hasta el último solo hay una
diferencia de 45 unidades, esta brecha puede ser por diferentes factores como lo son la experiencia, la red de contactos,
la atención al cliente o las capacidades de negociación. Para mejorar estos aspectos se puede hacer:
- Brindar capacitaciones en técnicas de ventas, manejo de objeciones y cierre de ventas.
- Juntar las mejores ventas con aquellos que están por debajo, soluciones personalizadas e incentivos.
Podemos destacar a Luisa y a Carmen como vendedoras estrella, ya que sus ventas superan al promedio en comparación con sus compañeros.

In [80]: import pandas as pd
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt

# Dividir en las características relevantes
X = df[['Superficie', 'Precio Venta']]

# Escalar los datos
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Número de clusters
num_clusters = 3

# Crear y ajustar el modelo K-means
kmeans = KMeans(n_clusters=num_clusters, random_state=42)
kmeans.fit(X_scaled)

# Agregar los labels de los clusters al DataFrame original
df['Cluster'] = kmeans.labels_

# Visualizar los clusters
plt.scatter(df['Superficie'], df['Precio Venta'], c=df['Cluster'], cmap='rainbow')
plt.xlabel('Superficie')
plt.ylabel('Precio Venta')
plt.title('Clustering de Inmuebles con K-means')
plt.show()

Clustering de Inmuebles con K-means

Precio Venta
3.0
2.5
2.0
1.5
1.0
0.5
0
50 100 150 200 250 300
Superficie

In [1]: #El gráfico de K-means podemos resaltar tres grupos, con características distintas
en cuanto tamaño y precio. Podemos concluir que podemos tener tres nichos como por ejemplo viviendas
de lujo, viviendas con tamaño medio y por último casas pequeñas.
Tenemos una relación positiva debido a que ambas variables 'Superficie' y 'Precio Venta' aumentan, no es un
patrón exacto pero hay una tendencia.

In [1]: df.head(1)

In [1]: import matplotlib.pyplot as plt
import seaborn as sns

# Descripción de los datos
print(df['Precio Venta'].describe())

# Configuración de la figura
plt.figure(figsize=(6, 5))

# Trazar el histograma y la curva de densidad
sns.kdeplot(df['Precio Venta'], color='blue', bins=100, kde=True, stat='density', alpha=0.4)

# Mostrar el gráfico
plt.title('Histograma de Precio Venta')
plt.xlabel('Precio Venta')
plt.ylabel('Densidad')
plt.show()

Histograma de Precio Venta

Densidad
0.0005
0.0004
0.0003
0.0002
0.0001
0
0 1000000 2000000 3000000 4000000 5000000 6000000 7000000 8000000 9000000 10000000
Precio Venta

In [54]: #Propiedades que mas se demoraron en vender
propiedades_ventas = df.sort_values(by='Días para Vender', ascending=False)

# Ordenar propiedades_ventas head(15) to_string()

Referencia  FechaAlta  Tipo Operación  Ciudad  Superficie  Precio Venta  Fecha Venta  Vendedor  Estatus  Días para Vender
1427      1428  2016-10-06      Terreno  Alquiler  Ciudad de México      132      882420      2018-10-06  Carmen  Vendida      730
1484      1485  2016-10-28      Terreno  Venta  Tijuana      249      1878954      2018-10-28  Luisa  Vendida      730
1482      1483  2016-10-28  Departamento  Alquiler  Monterrey      164      1421560      2018-10-28  Luisa  Vendida      730
898      899  2016-03-30  Industrial  Alquiler  Monterrey      167      1191044      2018-03-30  Joaquín  Vendida      730
928      929  2016-08-09      Casa  Venta  Monterrey      280      1702590      2018-04-05  Pedro  Vendida      730
1046      1047  2016-03-21      Casa  Alquiler  Tijuana      97      834394      2018-03-20  Jesús  Vendida      729
892      893  2016-07-27      Casa  Venta  Cancún      184      849464      2018-03-24  Jesús  Vendida      729
721      722  2016-03-28      Oficina  Alquiler  Cancún      110      849300      2018-01-29  Jesús  Vendida      729
1610      1611  2016-12-20      Local  Venta  Ciudad de México      281      2671748      2018-12-18  Carmen  Vendida      728
1084      1085  2016-11-25  Estacionamiento  Alquiler  Cancún      295      2613405      2018-11-23  Carmen  Vendida      728
1395      1396  2016-09-29      Industrial  Alquiler  Tijuana      121      490897      2018-09-27  María  Vendida      728
1888      1889  2016-10-30  Estacionamiento  Alquiler  Ciudad de México      103      528733      2018-10-28  María  Vendida      728
1206      1207  2016-08-31  Industrial  Venta  Tijuana      158      1473582      2018-08-29  Carmen  Vendida      728
1406      1407  2016-12-17      Local  Venta  Tijuana      165      1393095      2018-12-15  Carmen  Vendida      728
1337      1338  2016-09-07  Departamento  Alquiler  Cancún      99      938916      2018-09-14  Pedro  Vendida      727

In [49]: df['Fecha Venta'] = pd.to_datetime(df['Fecha Venta'])

In [70]: df['Mes'] = df['Fecha Venta'].dt.to_period('M') # Agrupar por mes
precio_promedio_mensuales = df.groupby('Mes')['Precio Venta'].mean().reset_index()

In [76]: plt.figure(figsize=(12, 6))
sns.lineplot(df['Precio Venta'].dt.to_period('M'), x=precio_promedio_mensuales['Mes'], y=precio_promedio_mensuales['Precio Venta'], marker='o')
plt.title('Evolución de los Precios de Venta de Inmuebles por Mes')
plt.xlabel('Mes')
plt.ylabel('Precio Promedio de Venta')
plt.grid(rotation=45) # Rotar las etiquetas del eje x para mejor legibilidad
plt.gcf()
plt.tight_layout() # Ajustar el diseño para que no se superpongan elementos
plt.show()

Evolución de los Precios de Venta de Inmuebles por Mes

Precio Promedio de Venta
1.26
1.24
1.22
1.2
1.18
1.16
1.14
1.12
2016-01 2016-02 2016-03 2016-04 2016-05 2016-06 2016-07 2016-08 2016-09 2016-10 2016-11 2016-12 2017-01 2017-02 2017-03 2017-04 2017-05 2017-06 2017-07 2017-08 2017-09 2017-10 2017-11 2017-12 2018-01 2018-02 2018-03 2018-04 2018-05 2018-06 2018-07 2018-08 2018-09 2018-10 2018-11 2018-12 2019-01 2019-02 2019-03 2019-04 2019-05 2019-06

In [1]: #¿Cómo han evolucionado los precios de venta de los inmuebles en los últimos años?
# Para analizar la evolución de los precios de las ventas se agrupa los promedios por meses,
# esto permite visualizar la evolución de los precios a lo largo del tiempo.

In [98]: df_ventas = df[df['Operación'] == 'Venta']
precio_promedio = df_ventas.groupby(['Ciudad', 'Tipo'])['Precio Venta'].mean().reset_index()
precio_promedio.rename(columns={'Precio Venta': 'Precio Promedio'}, inplace=True)

# Formatear la columna 'PrecioPromedio'
precio_promedio['PrecioPromedio'] = precio_promedio['PrecioPromedio'].apply(lambda x: f'{x:.2f}')

# Imprimir el DataFrame
print(precio_promedio)

Ciudad  Tipo PrecioPromedio
0      Cancún  Casa      1,106,847.25
1      Cancún  Departamento  1,032,479.60
2      Cancún  Estacionamiento  1,229,744.48
3      Cancún  Industrial  1,307,819.64
4      Cancún  Local  1,278,478.16
5      Cancún  Oficina  1,122,743.72
6      Cancún  Terreno  1,026,216.72
7  Ciudad de México  Casa  1,228,839.05
8  Ciudad de México  Departamento  1,169,701.49
9  Ciudad de México  Estacionamiento  1,159,254.59
10  Ciudad de México  Industrial  1,283,821.59
11  Ciudad de México  Local  1,300,703.68
12  Ciudad de México  Oficina  1,207,923.62
13  Ciudad de México  Terreno  1,309,398.07
14  Monterrey  Casa  1,282,846.78
15  Monterrey  Departamento  1,365,772.85
16  Monterrey  Estacionamiento  1,180,566.28
17  Monterrey  Industrial  1,006,846.96
18  Monterrey  Local  1,249,315.49
19  Monterrey  Oficina  1,285,124.52
20  Monterrey  Terreno  1,350,894.64
21  Tijuana  Casa  1,260,817.66
22  Tijuana  Departamento  1,281,255.63
23  Tijuana  Estacionamiento  1,207,611.12
24  Tijuana  Industrial  1,177,282.88
25  Tijuana  Local  1,211,232.09
```



26	Tijuana	Oficina	1,265,099.91
27	Tijuana	Terreno	1,200,037.53