1 - Parte EDA & Limpieza - Sección NVCBP

October 31, 2022

1 Análisis de la Encuesta Multiproposito

1.1 Importación de Paquetes y carga del archivo

[1]:
```
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
from scipy import stats
import numpy as np
```

[2]:
```
data = pd.read_csv('Encuesta_Suba_Filtrado_Preguntas.csv', sep = ',')
```

[3]:
```
data.shape
```

[3]:
```
(24536, 498)
```

Hay en total 24536 encuestados en la Localidad de Suba

2 Primera Sección (NVCBP)

2.1 NVCBP1

2.1.1 1. La vía de acceso a la edificación es:

- 1 Sendero o camino en tierra
- 2 Peatonal construida
- 3 Vehicular destapada
- 4 Vehicular pavimentada

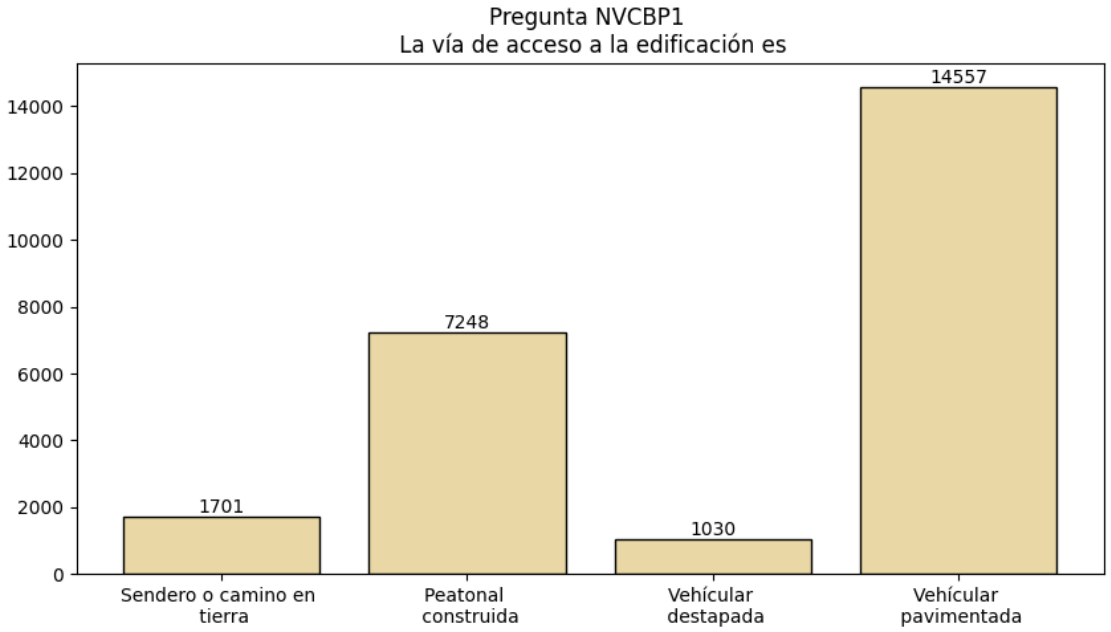
Datos: 24536

[4]:
```
data['NVCBP1'].describe()
```

[4]:
```
count    24536.000000
mean         3.159235
std          1.068032
min          1.000000
25%          2.000000
50%          4.000000
75%          4.000000
```

```
max          4.000000
Name: NVCBP1, dtype: float64
```

[5]:
```
plt.figure(figsize=(10,5))
bars = plt.bar(data['NVCBP1'].value_counts().index.tolist(),data['NVCBP1'].
 ↪value_counts().tolist(), edgecolor = 'black', color = '#e9d8a6')
plt.title('Pregunta NVCBP1 \n La vía de acceso a la edificación es')
plt.xticks([1,2,3,4],['Sendero o camino en \n tierra','Peatonal \n␣
 ↪construida','Vehícular \n destapada','Vehícular \n pavimentada'])
plt.bar_label(bars)
plt.show()
```



2.2 NVCBP2

2.2.1 2. ¿Cuál es el estado de la vía?

1. Bueno
2. Regular
3. Malo

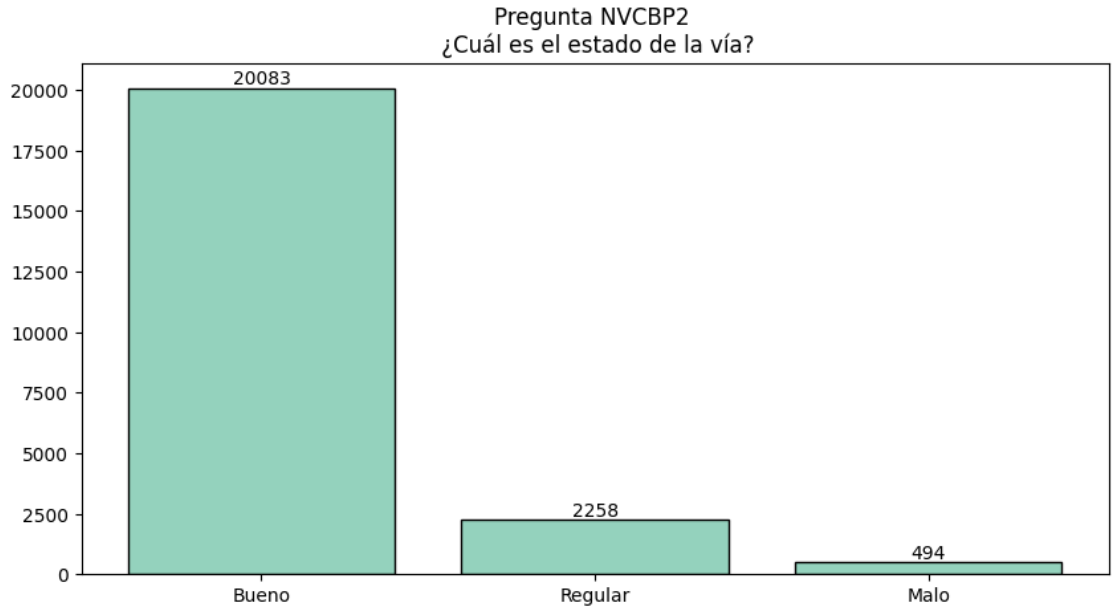
Datos: 22835

[6]:
```
data['NVCBP2'].describe()
```

[6]:
```
count    22835.00000
mean         1.14215
std          0.40647
```

```
min          1.00000
25%          1.00000
50%          1.00000
75%          1.00000
max          3.00000
Name: NVCBP2, dtype: float64
```

[7]:
```
plt.figure(figsize=(10,5))
bars = plt.bar(data['NVCBP2'].value_counts().index.tolist(),data['NVCBP2'].
 ↪value_counts().tolist(), edgecolor = 'black', color = '#94d2bd')
plt.bar_label(bars)
plt.xticks([1,2,3], ['Bueno', 'Regular', 'Malo'])
plt.title('Pregunta NVCBP2 \n ¿Cuál es el estado de la vía?')
plt.show()
```



2.3 NVCBP3

2.3.1 3. ¿La edificación donde está ubicada la vivienda tiene andén?

0. No
1. Si

Datos: 24536

[8]:
```
data['NVCBP3'].value_counts()
```

[8]:
```
1    21953
2     2583
```

```
Name: NVCBP3, dtype: int64
```

[10]:
```
data['NVCBP3'].describe()
```

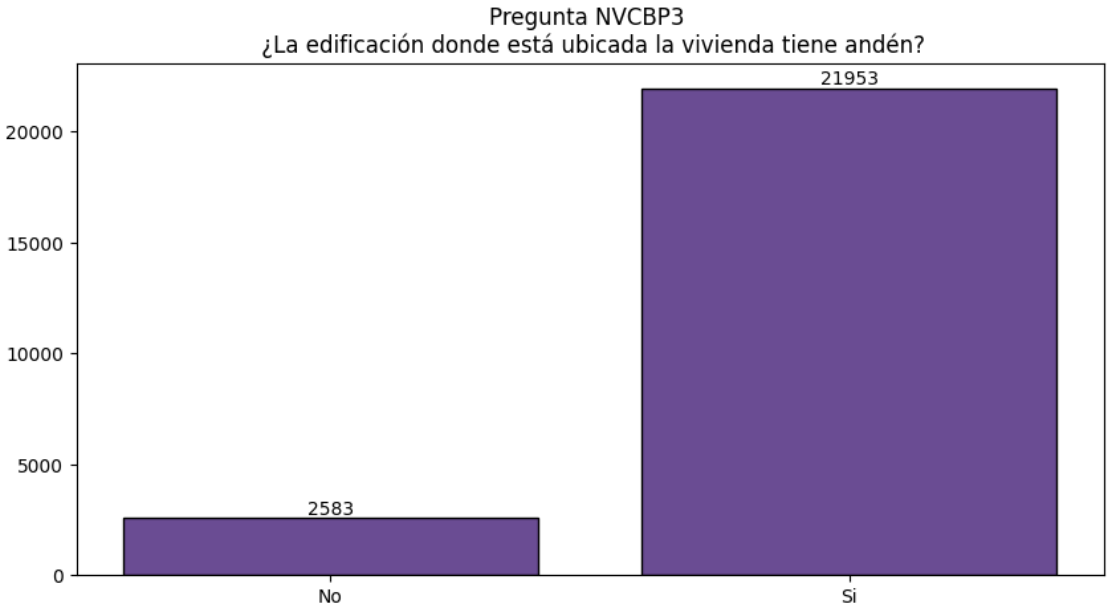
[10]:
```
count    24536.000000
mean         1.105274
std          0.306912
min          1.000000
25%          1.000000
50%          1.000000
75%          1.000000
max          2.000000
Name: NVCBP3, dtype: float64
```

[11]:
```
data = data.replace({'NVCBP3':2},0)
```

[12]:
```
data['NVCBP3'].value_counts()
```

[12]:
```
1    21953
0     2583
Name: NVCBP3, dtype: int64
```

[13]:
```
plt.figure(figsize=(10,5))
bars = plt.bar(data['NVCBP3'].value_counts().index.tolist(),data['NVCBP3'].
 ↪value_counts().tolist(), edgecolor = 'black', color = '#6a4c93')
plt.title('Pregunta NVCBP3 \n ¿La edificación donde está ubicada la vivienda␣
 ↪tiene andén?')
plt.xticks([0,1],['No','Si'])
plt.bar_label(bars)
plt.show()
```



2.4 NVCBP4

2.4.1 4. ¿La edificación está ubicada en un conjunto residencial?

0. No
1. Si

Datos: 24536

[14]:
```
data['NVCBP4'].value_counts()
```

[14]:
```
1    15526
2     9010
Name: NVCBP4, dtype: int64
```

[15]:
```
data['NVCBP4'].describe()
```

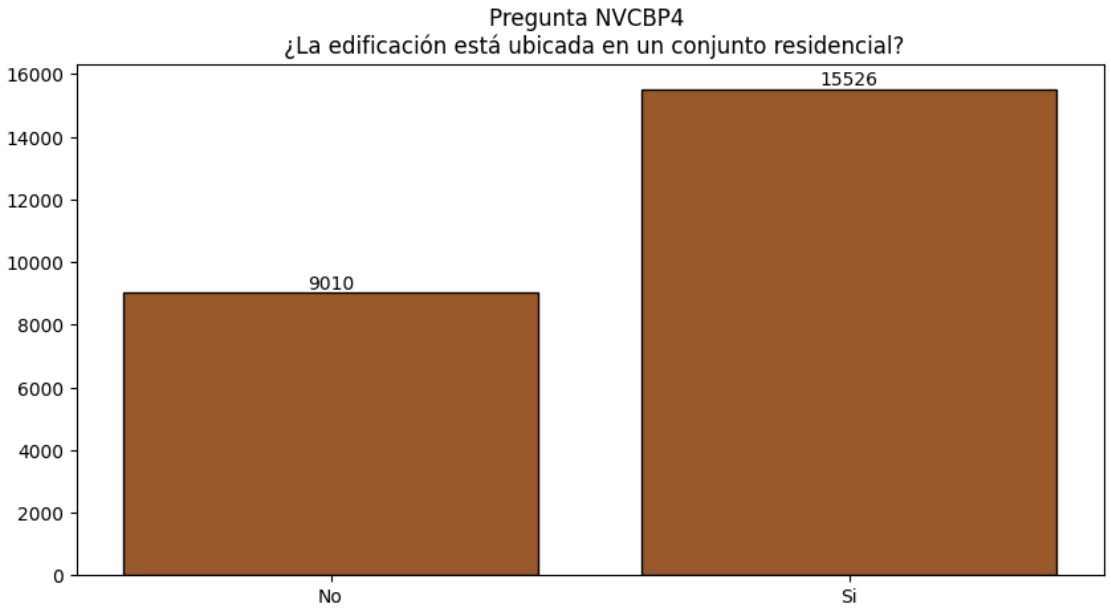
[15]:
```
count    24536.000000
mean         1.367216
std          0.482056
min          1.000000
25%          1.000000
50%          1.000000
75%          2.000000
max          2.000000
Name: NVCBP4, dtype: float64
```

[16]:
```
data = data.replace({'NVCBP4':2},0)
```

[17]:
```
data['NVCBP4'].value_counts()
```

[17]:
```
1    15526
0     9010
Name: NVCBP4, dtype: int64
```

[18]:
```
plt.figure(figsize=(10,5))
bars = plt.bar(data['NVCBP4'].value_counts().index.tolist(),data['NVCBP4'].
 ↪value_counts().tolist(), edgecolor = 'black', color = '#99582a')
plt.title('Pregunta NVCBP4 \n ¿La edificación está ubicada en un conjunto␣
 ↪residencial?')
plt.xticks([0,1],['No','Si'])
plt.bar_label(bars)
plt.show()
```



2.5 NVCBP5

2.5.1 5. La iluminación de la vía de acceso a la edificación en las noches es:

1. Suciente
2. Insuficiente
3. No tiene

Datos: 24536

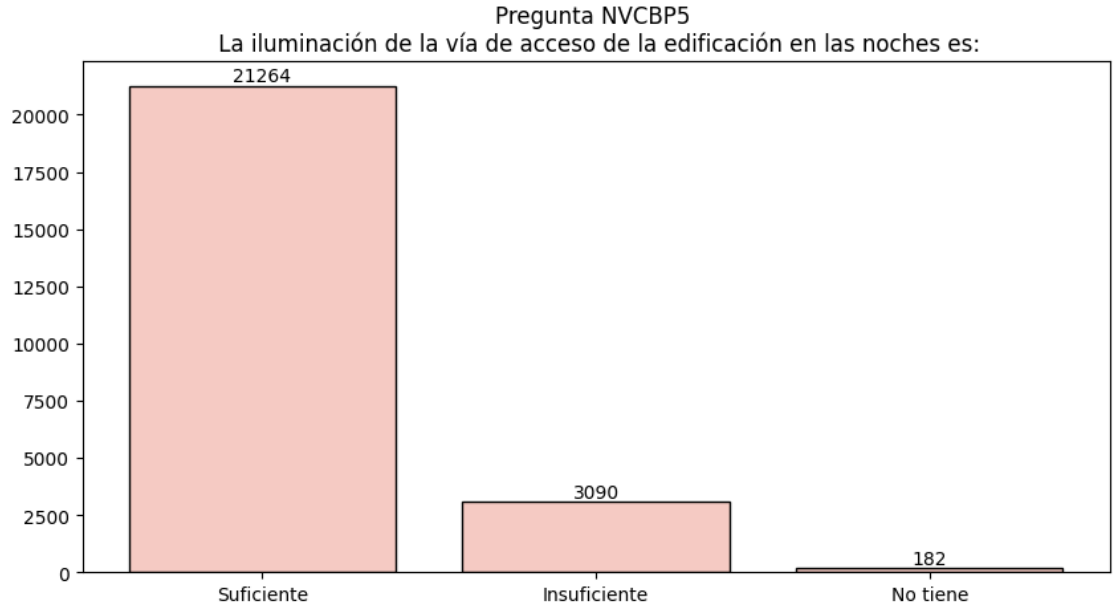
[19]:
```
data['NVCBP5'].value_counts()
```

[19]:
```
1    21264
2     3090
3      182
Name: NVCBP5, dtype: int64
```

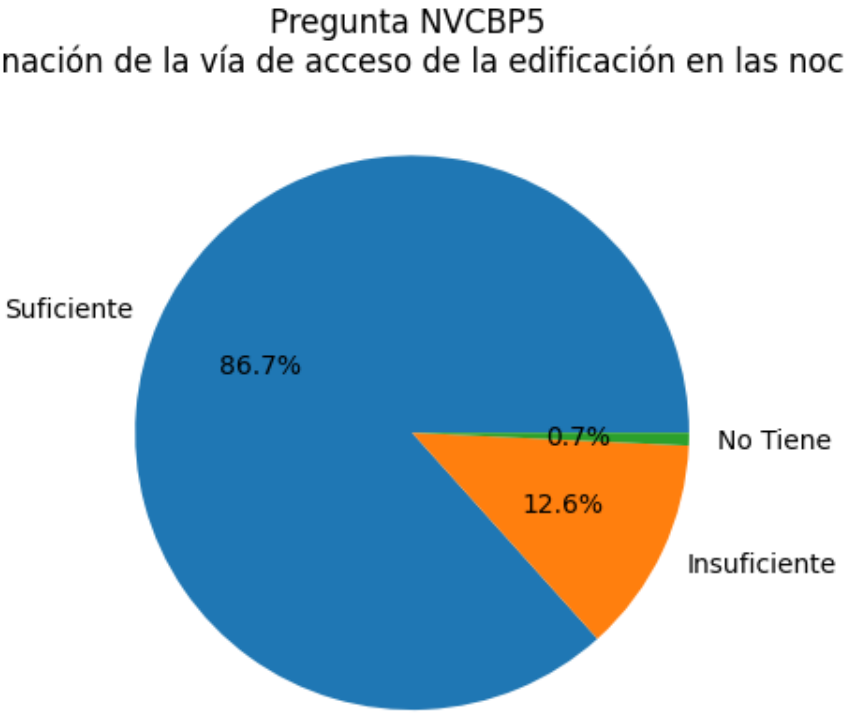
[20]:
```
data['NVCBP5'].describe()
```

[20]:
```
count    24536.000000
mean         1.140773
std          0.368506
min          1.000000
25%          1.000000
50%          1.000000
75%          1.000000
max          3.000000
Name: NVCBP5, dtype: float64
```

[21]:
```
plt.figure(figsize=(10,5))
bars = plt.bar(data['NVCBP5'].value_counts().index.tolist(),data['NVCBP5'].
 ↪value_counts().tolist(), edgecolor = 'black', color = '#f5cac3')
plt.xticks([1,2,3], ['Suficiente', 'Insuficiente', 'No tiene'])
plt.title('Pregunta NVCBP5 \n La iluminación de la vía de acceso de la␣
 ↪edificación en las noches es:')
plt.bar_label(bars)
plt.show()
```



[22]:
```
plt.pie(data['NVCBP5'].value_counts().tolist(), labels =␣
↪['Suficiente','Insuficiente','No Tiene'],autopct='%1.1f%%')
plt.title('Pregunta NVCBP5 \n La iluminación de la vía de acceso de la␣
↪edificación en las noches es:')
plt.show()
```



2.6 NVCBP6

2.6.1 6. ¿Cuántos pisos tiene la edificación donde está ubicada la vivienda?

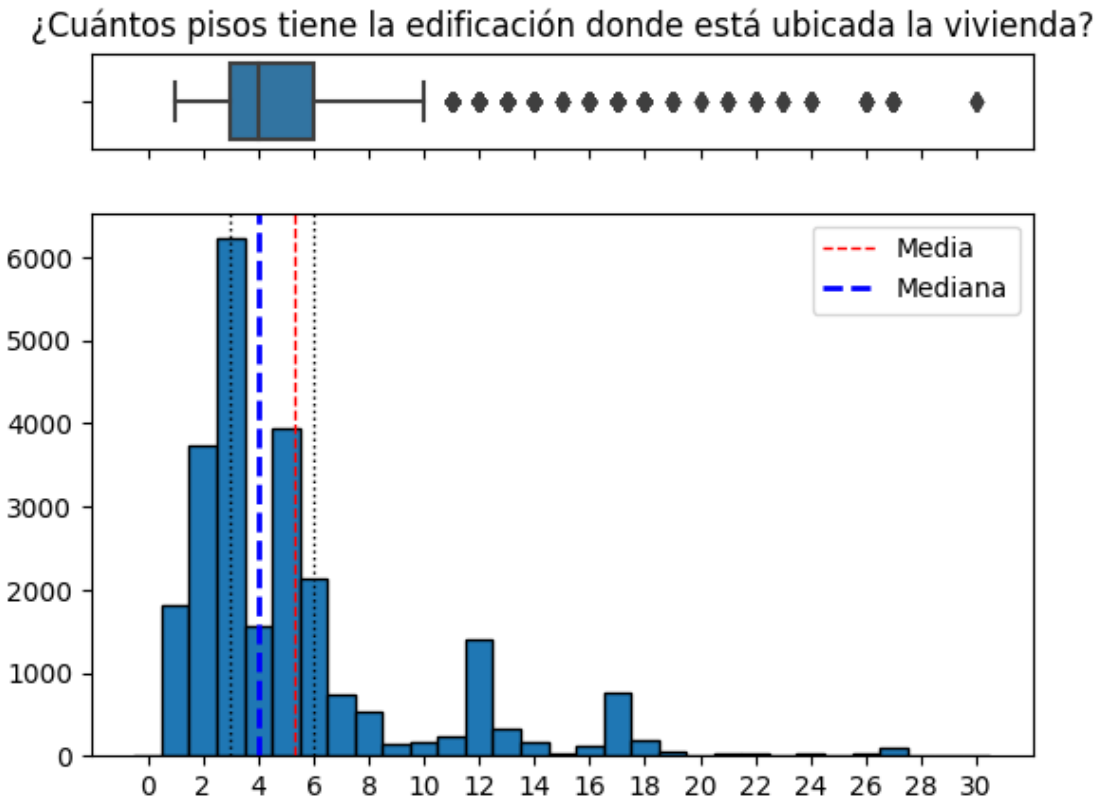
Datos: 24536

[23]:
```
data['NVCBP6'].describe()
```

[23]:
```
count    24536.000000
mean         5.335915
std          4.409705
min          1.000000
25%          3.000000
50%          4.000000
75%          6.000000
max         30.000000
```

```
Name: NVCBP6, dtype: float64
```

[24]:
```
# creating a figure composed of two matplotlib.Axes objects (ax_box and ax_hist)
f, (ax_box, ax_hist) = plt.subplots(2, sharex=True,␣
 ↪gridspec_kw={"height_ratios": (.15, .85)})

# assigning a graph to each ax
sns.boxplot(x = data['NVCBP6'], ax=ax_box)
#ax_box.boxplot(data['NVCBP6'], vert = False)
#sns.histplot(data=data, x='NVCBP6', ax=ax_hist)
ax_hist.hist(data['NVCBP6'], bins = np.arange(32) - 0.5, edgecolor = 'k')
ax_hist.axvline(data['NVCBP6'].mean(), color='r', linestyle='dashed',␣
 ↪linewidth=1, label = 'Media')
ax_hist.axvline(data['NVCBP6'].median(), color='b', linestyle='dashed',␣
 ↪linewidth=2, label = 'Mediana')
ax_hist.axvline(data['NVCBP6'].quantile(0.25), color='k', linestyle=':',␣
 ↪linewidth=1)
ax_hist.axvline(data['NVCBP6'].quantile(0.75), color='k', linestyle=':',␣
 ↪linewidth=1)
ax_hist.legend()
# Remove x axis name for the boxplot
ax_box.set_title('¿Cuántos pisos tiene la edificación donde está ubicada la␣
 ↪vivienda?')
ax_hist.set_xticks(range(0,32,2), align = 'center')
ax_box.set(xlabel='')
plt.show()
```



2.7 NVCBP7

2.7.1 7. ¿La edificación donde está ubicada la vivienda tiene ascensor?

0. No
1. Si

Datos: 12772

[27]:
```
data['NVCBP7'].value_counts()
```

[27]:
```
1.0    7242
2.0    5530
Name: NVCBP7, dtype: int64
```

[28]:
```
data['NVCBP7'].describe()
```

[28]:
```
count    12772.000000
mean         1.432978
std          0.495507
min          1.000000
25%          1.000000
```

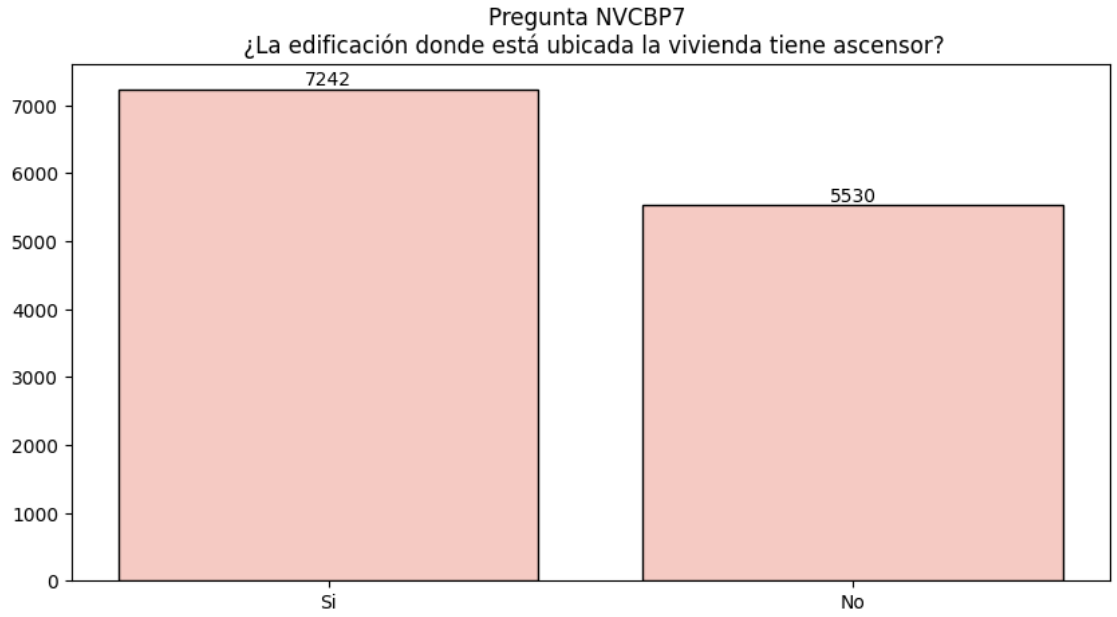
```
50%          1.000000
75%          2.000000
max          2.000000
Name: NVCBP7, dtype: float64
```

[29]:
```
data = data.replace({'NVCBP7':2},0)
```

[30]:
```
data['NVCBP7'].value_counts()
```

[30]:
```
1.0    7242
0.0    5530
Name: NVCBP7, dtype: int64
```

[31]:
```
plt.figure(figsize=(10,5))
bars = plt.bar(['Si','No'],data['NVCBP7'].value_counts().tolist(), edgecolor =␣
↪'black', color = '#f5cac3')
plt.title('Pregunta NVCBP7 \n ¿La edificación donde está ubicada la vivienda␣
↪tiene ascensor?')
plt.bar_label(bars)
plt.show()
```



2.7.2 NVCBP9

2.7.3 9. ¿Algún espacio de la vivienda está dedicado a negocios de industria, comercio o servicios?

Datos: 24536

- 0. No
- 1. Si

NVCBP9A1,NVCBP9A2,NVCBP9A3,NVCBP9A4:

¿A qué negocio se dedica este espacio?

- 0. No
- 1. Si

Datos: 609

Hay preguntas que no se contestan para las preguntas anexas que no deben responder porque no tienen un negocio

[32]:
```
comercio = ['NVCBP9','NVCBP9A1','NVCBP9A2','NVCBP9A3','NVCBP9A4']
```

[33]:
```
data[comercio[0]].describe()
```

[33]:
```
count    24536.000000
mean         1.975179
std          0.155581
min          1.000000
25%          2.000000
50%          2.000000
75%          2.000000
max          2.000000
Name: NVCBP9, dtype: float64
```

[34]:
```
for i in comercio:
    data = data.replace({i:2},0)
```

[35]:
```
for i in comercio:
    print(data[i].count())
```

```
24536
609
609
609
609
```

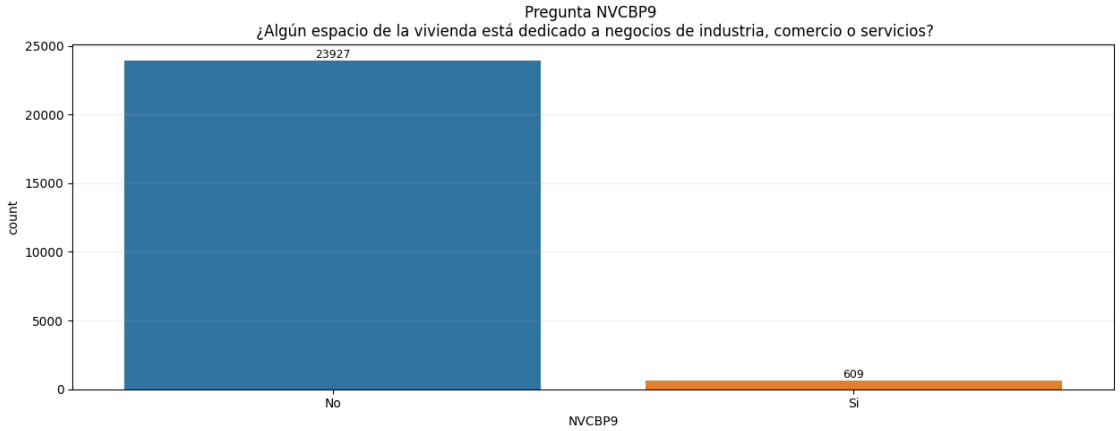
[36]:
```
for i in comercio:
    print(data[i].value_counts())
```

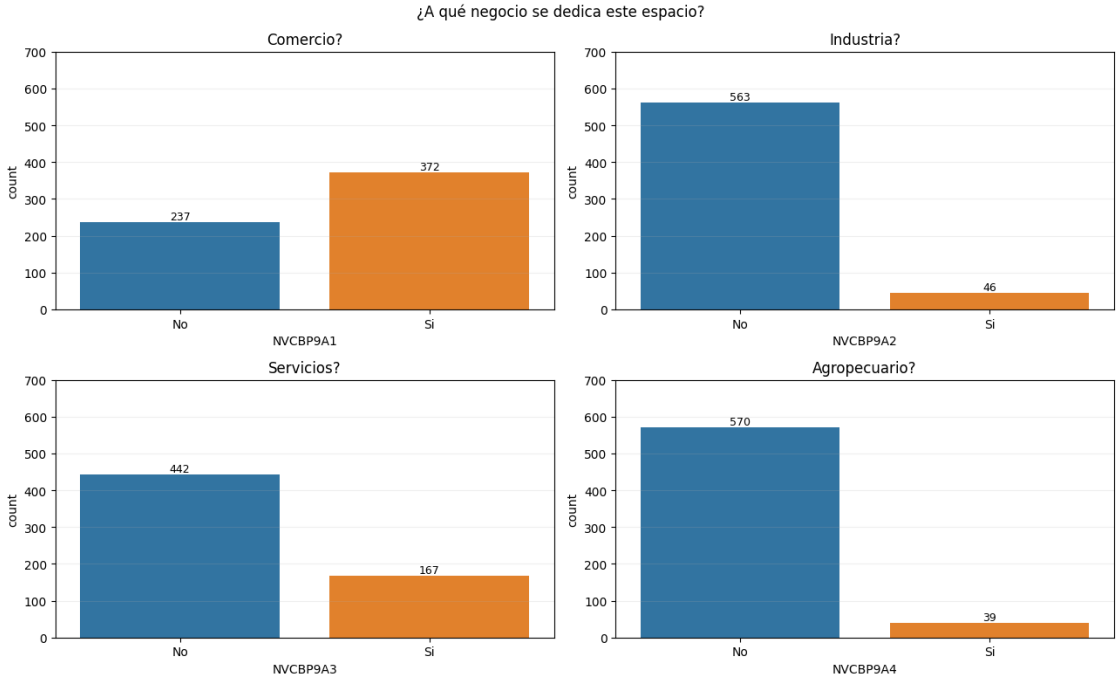
```
0    23927
1      609
Name: NVCBP9, dtype: int64
1.0    372
0.0    237
```

```
Name: NVCBP9A1, dtype: int64
0.0    563
1.0     46
Name: NVCBP9A2, dtype: int64
0.0    442
1.0    167
Name: NVCBP9A3, dtype: int64
0.0    570
1.0     39
Name: NVCBP9A4, dtype: int64
```

[54]:
```
comercio_2 =['NVCBP9A1','NVCBP9A2','NVCBP9A3','NVCBP9A4']
l_com = ['Comercio?','Industria?','Servicios?','Agropecuario?']
```

[59]:
```
fig, ax = plt.subplots(figsize=(15, 5))
g = sns.countplot(ax=ax, data = data, x ='NVCBP9')

for bars in ax.containers:
    ax.bar_label(bars, fmt='%.0f', fontsize=9)

ax.set_title('Pregunta NVCBP9 \n ¿Algún espacio de la vivienda está dedicado a␣
 ↪negocios de industria, comercio o servicios?')
ax.set_xticklabels(['No','Si'])

plt.grid(alpha = 0.2, axis = 'y')
plt.show()



fig, axes = plt.subplots(2,2, figsize = (13,8), squeeze=False)
fig.subplots_adjust(top=0.9)
axli = axes.flatten()
fig.suptitle('¿A qué negocio se dedica este espacio?')
for ax,cols,names in zip(axli,comercio_2,l_com):
    sns.countplot(x = cols, data = data, ax = ax)
    ax.grid(alpha = 0.2, axis = 'y')
    ax.set_title(f'{names}')
    ax.set_xticks([0,1],['No','Si'])
    ax.set_ylim(0,700)
    ax.margins(y=0.1) # make room for the labels
    for bars in ax.containers:
        ax.bar_label(bars, fmt='%.0f', fontsize=9)
plt.tight_layout()

#plt.show()
```





2.8 NVCBP10

2.8.1 10 .Tipo de vivienda:

1. Casa
2. Apartamento
3. Cuarto(s)
4. Otro

Datos: 24536

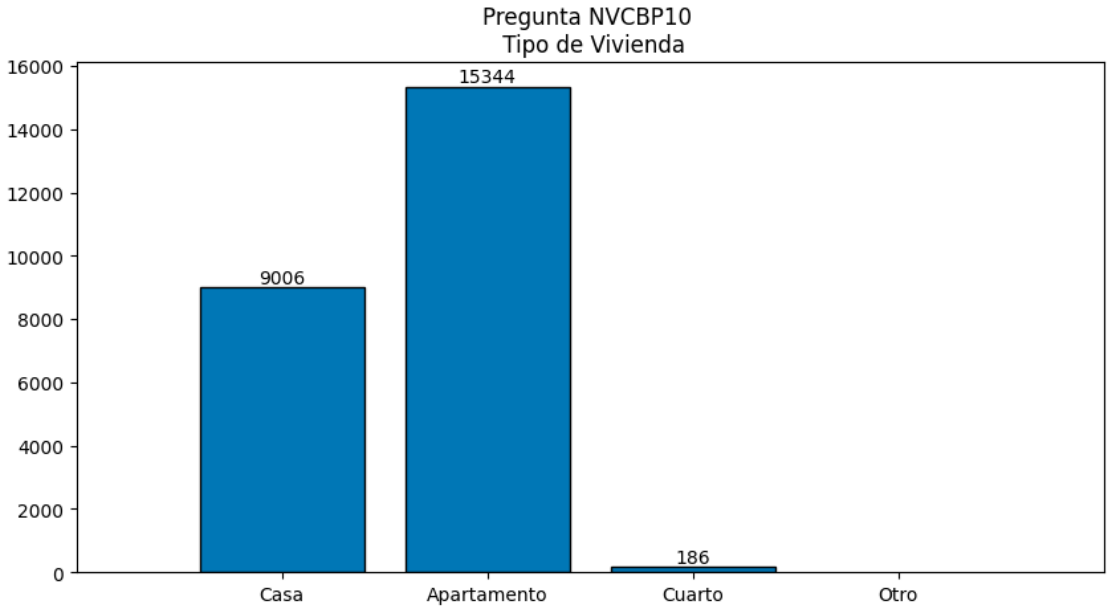
[60]:
```
data['NVCBP10'].describe()
```

[60]:
```
count    24536.000000
mean         1.640528
std          0.495402
min          1.000000
25%          1.000000
50%          2.000000
75%          2.000000
max          3.000000
Name: NVCBP10, dtype: float64
```

[61]:
```
data['NVCBP10'].value_counts()
```

[61]:
```
2    15344
1     9006
3      186
Name: NVCBP10, dtype: int64
```

[65]:
```
plt.figure(figsize=(10,5))
bars = plt.bar(data['NVCBP10'].value_counts().index.tolist(),data['NVCBP10'].
 ↪value_counts().tolist(), edgecolor = 'black', color = '#0077b6')
plt.xticks([1,2,3,4], ['Casa', 'Apartamento', 'Cuarto', 'Otro'])
plt.bar_label(bars)
plt.title('Pregunta NVCBP10 \n Tipo de Vivienda')
plt.xlim([0,5])
plt.show()
```



2.9 NCVP11AA

2.9.1 11AA .Estrato para la tarifa del servicio (Energia Eléctrica)

Datos: 24518

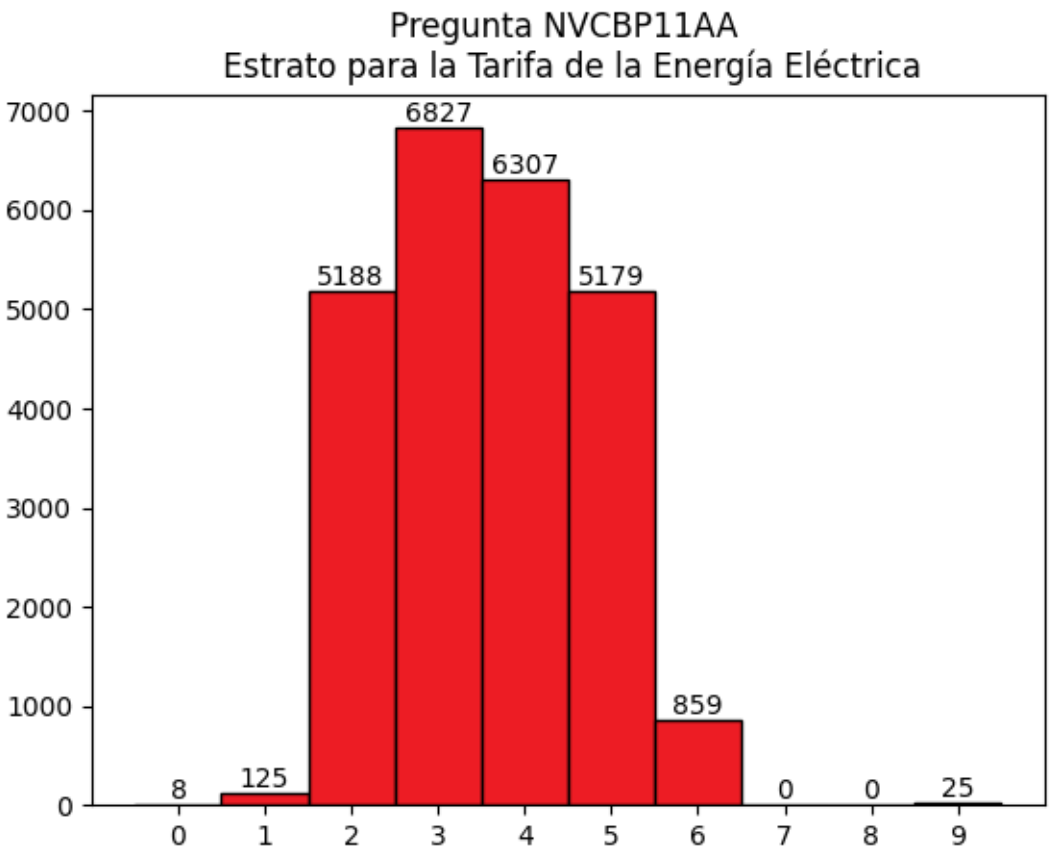
[67]:
```
data['NVCBP11AA'].describe()
```

[67]:
```
count    24518.000000
mean         3.568154
std          1.168924
min          0.000000
25%          3.000000
50%          4.000000
75%          4.000000
max          9.000000
Name: NVCBP11AA, dtype: float64
```

[68]:
```
data['NVCBP11AA'].value_counts()
```

[68]:
```
3.0    6827
4.0    6307
2.0    5188
5.0    5179
6.0     859
1.0     125
9.0      25
0.0       8
Name: NVCBP11AA, dtype: int64
```

[69]:
```
counts, edges, bars = plt.hist(data['NVCBP11AA'], bins = np.arange(11) - 0.5,␣
 ↪edgecolor = 'black', color = '#ED1C24')
#ticklabels = [i for i in range(5)]
#plt.xticks(range(5), ticklabels)
plt.xticks(range(10))
plt.bar_label(bars)
plt.title('Pregunta NVCBP11AA \n Estrato para la Tarifa de la Energía␣
 ↪Eléctrica')
plt.xlim([-1,10])
plt.show()
```



2.10 NVCBP11A

2.10.1 11A. ¿Cuenta con el servicio público de Energía Electrica?

Datos: 24536

[77]:
```
data['NVCBP11A'].count()
```

[77]:
```
24536
```

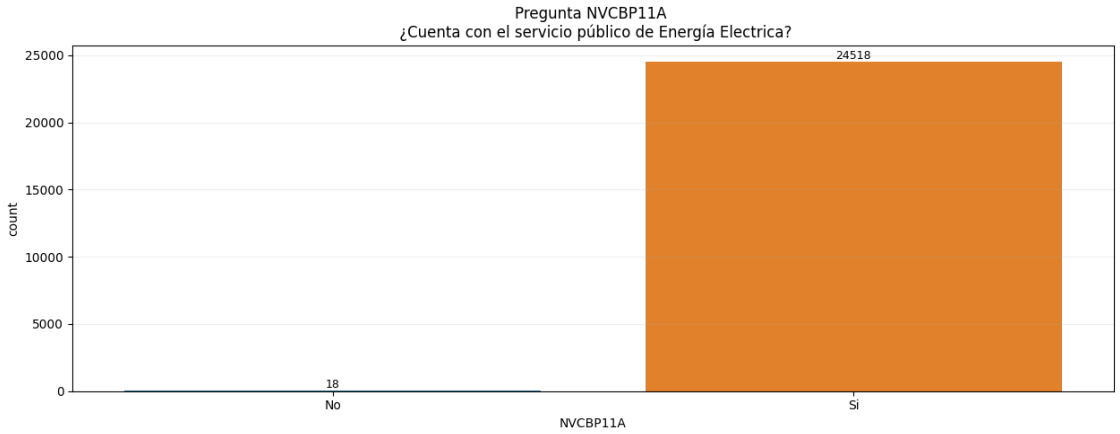
[78]:
```
data = data.replace({'NVCBP11A':2},0)
```

[79]:
```
data['NVCBP11A'].value_counts()
```

[79]:
```
1    24518
0       18
Name: NVCBP11A, dtype: int64
```

[83]:
```
fig, ax = plt.subplots(figsize=(15, 5))
g = sns.countplot(ax=ax, data = data, x ='NVCBP11A')
```

```
for bars in ax.containers:
    ax.bar_label(bars, fmt='%.0f', fontsize=9)

ax.set_title('Pregunta NVCBP11A \n ¿Cuenta con el servicio público de Energía␣
 ↪Electrica?')
ax.set_xticklabels(['No','Si'])

plt.grid(alpha = 0.2, axis = 'y')
plt.show()
```



2.11 NVCBP11B

2.11.1 11B. ¿Cuenta con el servicio público de Acueducto?

Datos: 24536

[80]:
```
data['NVCBP11B'].count()
```

[80]:
```
24536
```

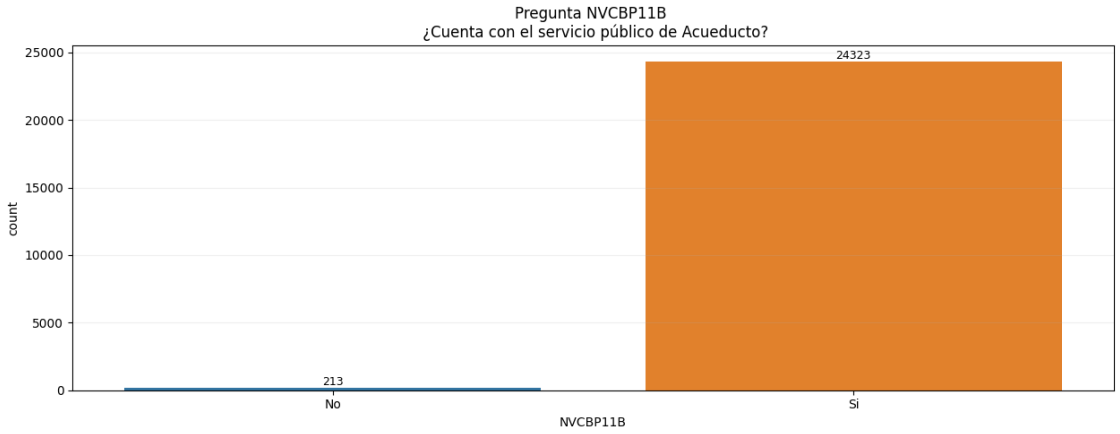
[81]:
```
data = data.replace({'NVCBP11B':2},0)
```

[82]:
```
data['NVCBP11B'].value_counts()
```

[82]:
```
1    24323
0      213
Name: NVCBP11B, dtype: int64
```

[92]:
```
fig, ax = plt.subplots(figsize=(15, 5))
g = sns.countplot(ax=ax, data = data, x ='NVCBP11B')

for bars in ax.containers:
    ax.bar_label(bars, fmt='%.0f', fontsize=9)
```

```
ax.set_title('Pregunta NVCBP11B \n ¿Cuenta con el servicio público de Acueducto?
 ↪')
ax.set_xticklabels(['No','Si'])

plt.grid(alpha = 0.2, axis = 'y')
plt.show()
```



2.12 NVCBP11C

2.12.1 11C. ¿Cuenta con el servicio público de Alcantarillado?

Datos: 24536

[89]:
```
data['NVCBP11C'].count()
```

[89]:
```
24536
```

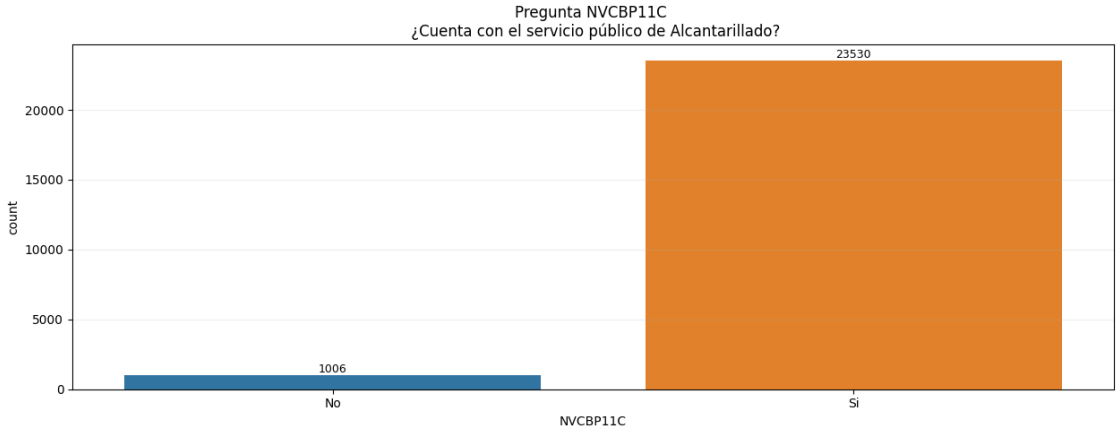
[90]:
```
data = data.replace({'NVCBP11C':2},0)
```

[91]:
```
data['NVCBP11C'].value_counts()
```

[91]:
```
1    23530
0     1006
Name: NVCBP11C, dtype: int64
```

[93]:
```
fig, ax = plt.subplots(figsize=(15, 5))
g = sns.countplot(ax=ax, data = data, x ='NVCBP11C')

for bars in ax.containers:
    ax.bar_label(bars, fmt='%.0f', fontsize=9)
```

```
ax.set_title('Pregunta NVCBP11C \n ¿Cuenta con el servicio público de␣
 ↪Alcantarillado?')
ax.set_xticklabels(['No','Si'])

plt.grid(alpha = 0.2, axis = 'y')
plt.show()
```



2.13 NVCBP11D

2.13.1 11D. ¿Cuenta con el servicio público de Recolección de Basuras?

Datos: 24536

[94]:
```
data['NVCBP11D'].count()
```

[94]:
```
24536
```

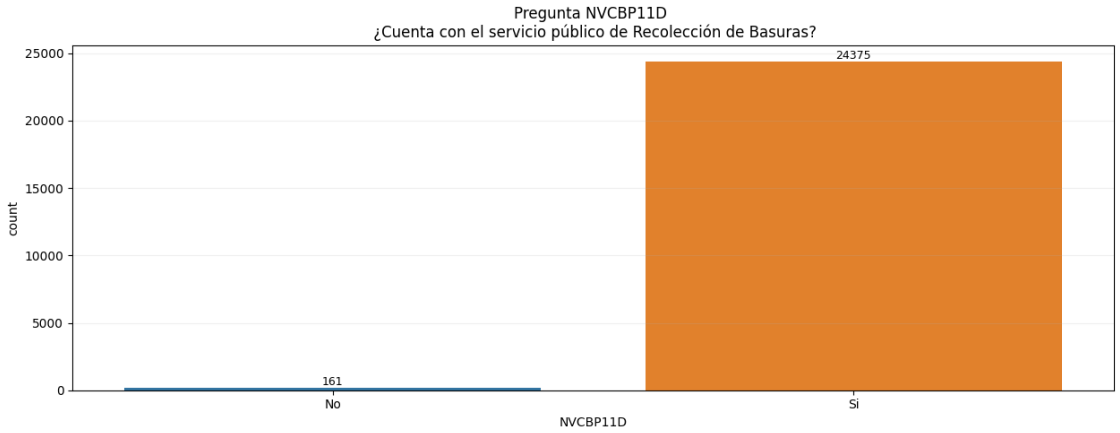
[95]:
```
data = data.replace({'NVCBP11D':2},0)
```

[96]:
```
data['NVCBP11D'].value_counts()
```

[96]:
```
1    24375
0      161
Name: NVCBP11D, dtype: int64
```

[98]:
```
fig, ax = plt.subplots(figsize=(15, 5))
g = sns.countplot(ax=ax, data = data, x ='NVCBP11D')

for bars in ax.containers:
    ax.bar_label(bars, fmt='%.0f', fontsize=9)

ax.set_title('Pregunta NVCBP11D \n ¿Cuenta con el servicio público de␣
 ↪Recolección de Basuras?')
```

```
ax.set_xticklabels(['No','Si'])

plt.grid(alpha = 0.2, axis = 'y')
plt.show()
```



2.14 NVCBP14

2.14.1 14. La vivienda está cerca de:

1. (NVCBP14A) Fábricas o Industrias
2. (NVCBP14B) Basureros o botaderos de basuras
3. (NVCBP14C) Plazas de Mercado o Mataderos
4. (NVCBP14D) Terminales de Buses
5. (NVCBP14E) Bares o discotecas
6. (NVCBP14L) Prostíbulos
7. (NVCBP14F) Expendios de droga (ollas)
8. (NVCBP14G) Lotes baldíos o sitios oscuros y peligrosos
9. (NVCBP14H) Lineas de alta tensión
10. (NVCBP14I) Caños de aguas resiguales
11. (NVCBP14J) Zona de riesgo de incendio forestal
12. (NVCBP14K) Talleres de mecánica, servitecas o estaciones de gasolina

Datos: 24536

[100]:
```
problemas = [i for i in data.columns if ('NVCBP14') in i]
```

[101]:
```
for i in problemas:
    print(i, data[i].count())
```

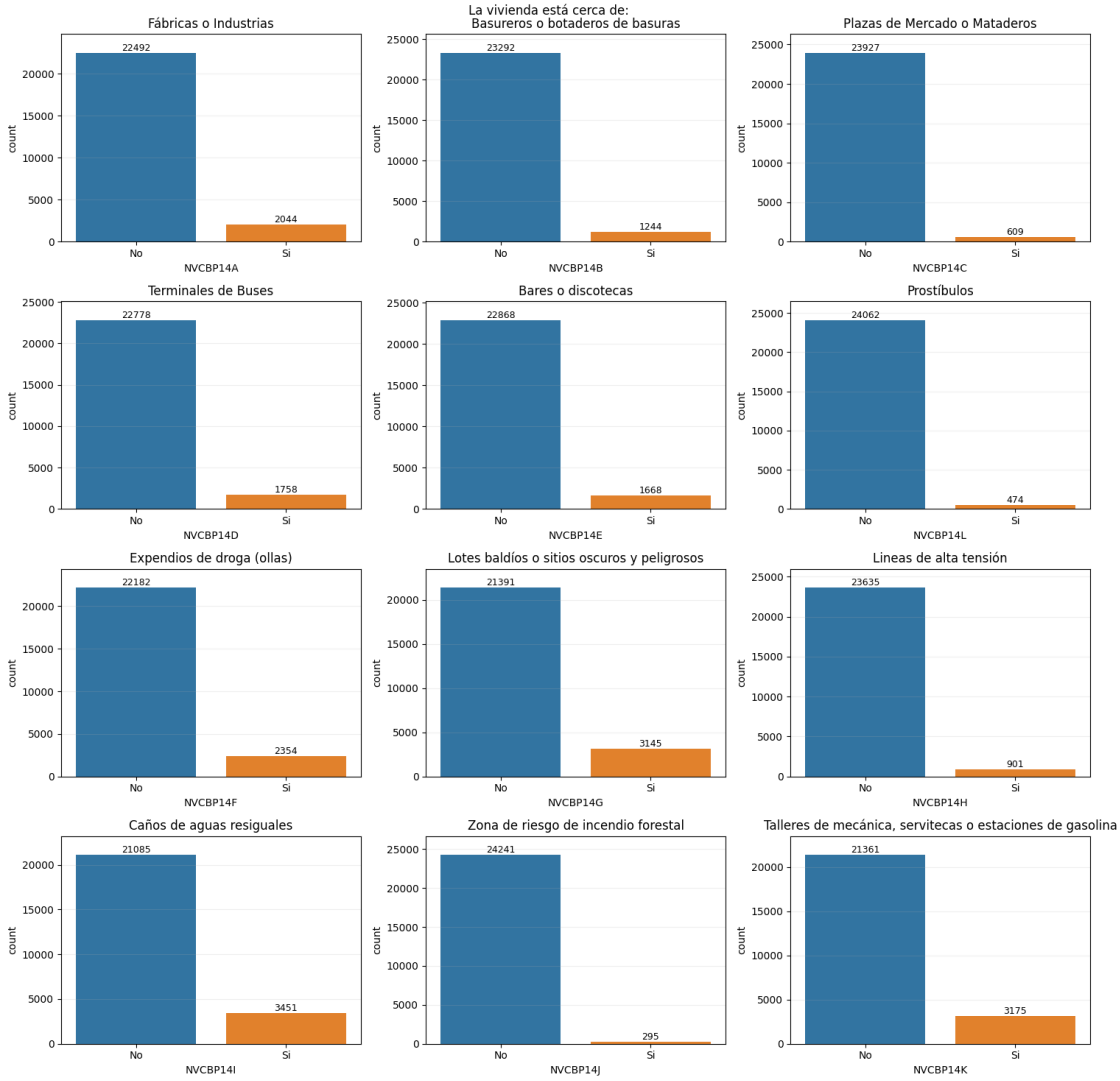
```
NVCBP14A 24536
NVCBP14B 24536
NVCBP14C 24536
NVCBP14D 24536
NVCBP14E 24536
```

```
NVCBP14L 24536
NVCBP14F 24536
NVCBP14G 24536
NVCBP14H 24536
NVCBP14I 24536
NVCBP14J 24536
NVCBP14K 24536
```

[102]:
```
for i in problemas:
    data = data.replace({i:2},0)
```

[103]:
```
list_pro = ['Fábricas o Industrias',
            'Basureros o botaderos de basuras',
            'Plazas de Mercado o Mataderos',
            'Terminales de Buses',
            'Bares o discotecas',
            'Prostíbulos',
            'Expendios de droga (ollas)',
            'Lotes baldíos o sitios oscuros y peligrosos',
            'Lineas de alta tensión',
            'Caños de aguas resiguales',
            'Zona de riesgo de incendio forestal',
            'Talleres de mecánica, servitecas o estaciones de gasolina']
```

[107]:
```
fig, axes = plt.subplots(4,3, figsize = (15,15), squeeze=False)
fig.subplots_adjust(top=0.9)
axli = axes.flatten()
fig.suptitle('La vivienda está cerca de:')
for ax,cols,names in zip(axli,problemas,list_pro):
    sns.countplot(x = cols, data = data, ax = ax)
    ax.grid(alpha = 0.2, axis = 'y')
    ax.set_title(f'{names}')
    ax.set_xticks([0,1],['No','Si'])
    ax.margins(y=0.1) # make room for the labels
    for bars in ax.containers:
        ax.bar_label(bars, fmt='%.0f', fontsize=9)
plt.tight_layout()

plt.show()
```



2.15 NVCBP15

2.15.1 15. ¿Cuales de los siguientes problemas presenta el entorno cercano a su vivienda?

1. (NVCBP15A) Ruido
2. (NVCBP15B) Exceso de anuncios publicitarios
3. (NVCBP15C) Inseguridad
4. (NVCBP15D) Contaminación del aire
5. (NVCBP15E) Malos Olores
6. (NVCBP15F) Disposición inadecuada de las basuras.
7. (NVCBP15G) Invasión del espacio público
8. (NVCBP15H) Presencia de insectos, roedores o animales que causen molestia
9. (NVCBP15I) Contaminación de Cuerpos de Agua
10. (NVCBP15J) Abandono de escombros

11. (NVCBP15K) Disposición inadecuada de residuos hospitalarios
12. (NVCBP15L) Arboles que ponen en riesgo las viviendas o sus habitantes
13. (NVCBP15M) Lugares con presencia de orina o excremento humano

Datos: 24536

[108]:
```
problemas15 = [i for i in data.columns if ('NVCBP15') in i]
print(problemas15)
```

```
['NVCBP15A', 'NVCBP15B', 'NVCBP15C', 'NVCBP15D', 'NVCBP15E', 'NVCBP15F',
'NVCBP15G', 'NVCBP15H', 'NVCBP15I', 'NVCBP15J', 'NVCBP15K', 'NVCBP15L',
'NVCBP15M']
```

[109]:
```
for i in problemas15:
    print(i,data[i].count())
```

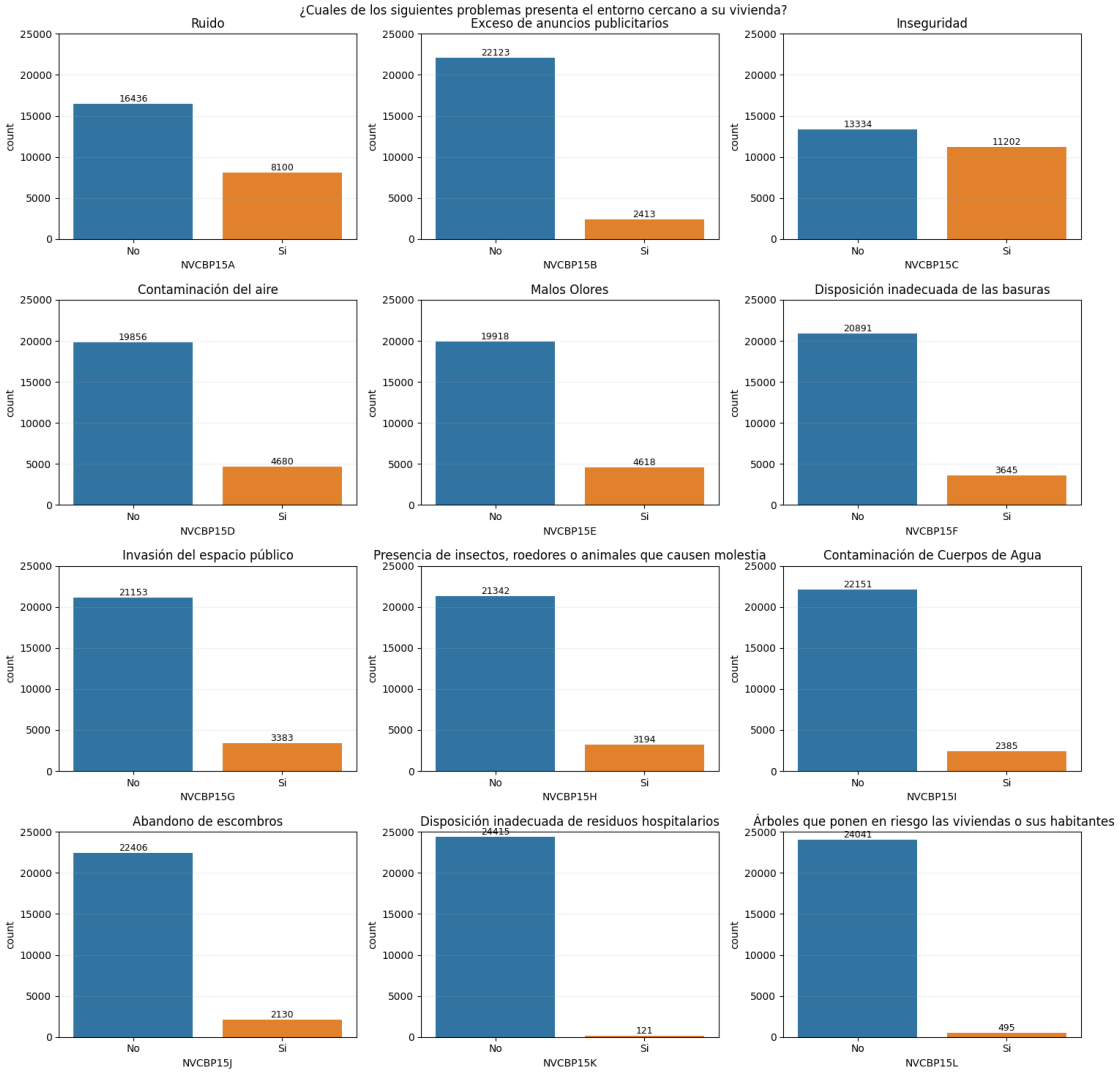
```
NVCBP15A 24536
NVCBP15B 24536
NVCBP15C 24536
NVCBP15D 24536
NVCBP15E 24536
NVCBP15F 24536
NVCBP15G 24536
NVCBP15H 24536
NVCBP15I 24536
NVCBP15J 24536
NVCBP15K 24536
NVCBP15L 24536
NVCBP15M 24536
```

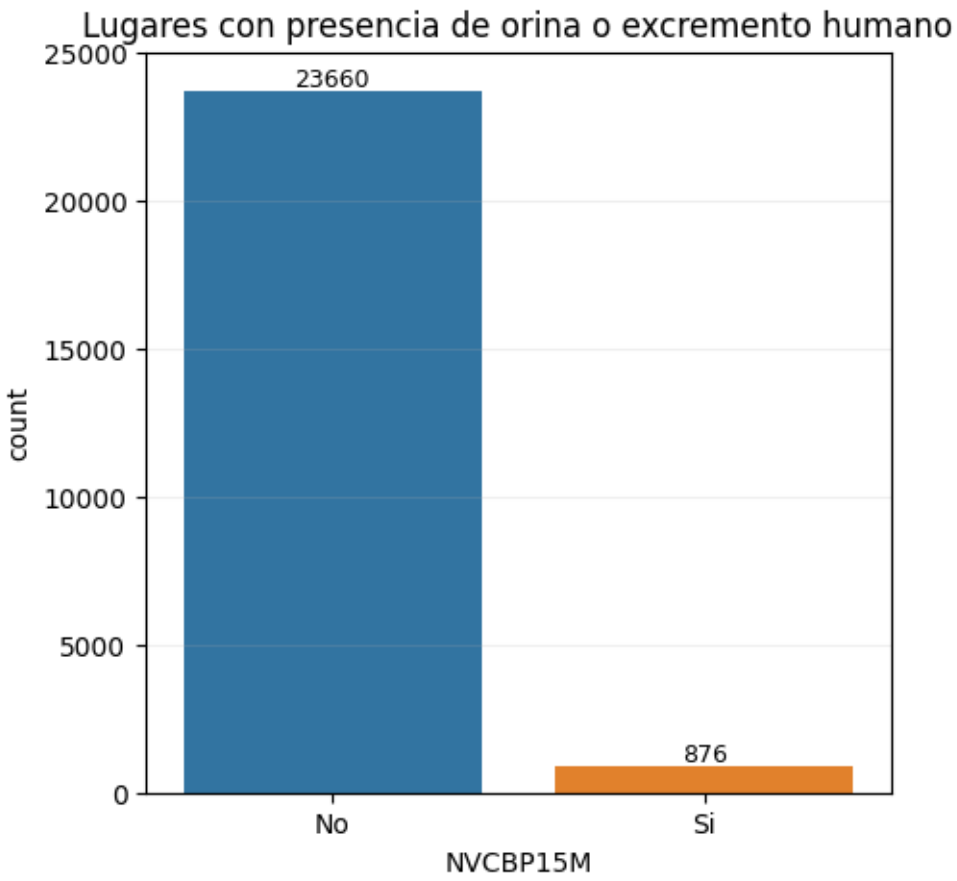
[110]:
```
for i in problemas15:
    data = data.replace({i:2},0)
```

[111]:
```
list_problemas = ['Ruido','Exceso de anuncios publicitarios',
'Inseguridad','Contaminación del aire',
'Malos Olores',
'Disposición inadecuada de las basuras',
'Invasión del espacio público',
'Presencia de insectos, roedores o animales que causen molestia',
'Contaminación de Cuerpos de Agua',
'Abandono de escombros',
'Disposición inadecuada de residuos hospitalarios',' Árboles que ponen en␣
 ↪riesgo las viviendas o sus habitantes',
'Lugares con presencia de orina o excremento humano']
```

[114]:
```
fig, axes = plt.subplots(4,3, figsize = (15,15), squeeze=False)
fig.subplots_adjust(top=0.9)
axli = axes.flatten()
```

```
fig.suptitle('¿Cuales de los siguientes problemas presenta el entorno cercano a␣
 ↪su vivienda?')
for ax,cols,names in zip(axli,problemas15,list_problemas):
    sns.countplot(x = cols, data = data, ax = ax)
    ax.grid(alpha = 0.2, axis = 'y')
    ax.set_title(f'{names}')
    ax.set_xticks([0,1],['No','Si'])
    ax.set_ylim(0,25000)
    ax.margins(y=0.1) # make room for the labels
    for bars in ax.containers:
        ax.bar_label(bars, fmt='%.0f', fontsize=9)
plt.tight_layout()

plt.show()

fig, ax = plt.subplots(figsize=(5, 5))
g = sns.countplot(ax=ax, data = data, x ='NVCBP15M')

for bars in ax.containers:
    ax.bar_label(bars, fmt='%.0f', fontsize=9)

ax.set_title('Lugares con presencia de orina o excremento humano')
ax.set_xticklabels(['No','Si'])
ax.set_ylim(0,25000)

plt.grid(alpha = 0.2, axis = 'y')
plt.show()
```


¿Cuales de los siguientes problemas presenta el entorno cercano a su vivienda?
Ruido
16436
8100
NVCBP15A
Exceso de anuncios publicitarios
22123
2413
NVCBP15B
Inseguridad
13334
11202
NVCBP15C
Contaminación del aire
19856
4680
NVCBP15D
Malos Olores
19918
4618
NVCBP15E
Disposición inadecuada de las basuras
20891
3645
NVCBP15F
Invasión del espacio público
21153
3383
NVCBP15G
Presencia de insectos, roedores o animales que causen molestia
21342
3194
NVCBP15H
Contaminación de Cuerpos de Agua
22151
2385
NVCBP15I
Abandono de escombros
22406
2130
NVCBP15J
Disposición inadecuada de residuos hospitalarios
24415
121
NVCBP15K
Árboles que ponen en riesgo las viviendas o sus habitantes
24041
495
NVCBP15L
count
No
Si



2.16 NVCBP16

2.16.1 16. En total ¿cuántos grupos de personas (hogares) preparan los alimentos por separado en esta vivienda y atienden necesidades básicas con cargo a un presupuesto común?

Datos: 24536

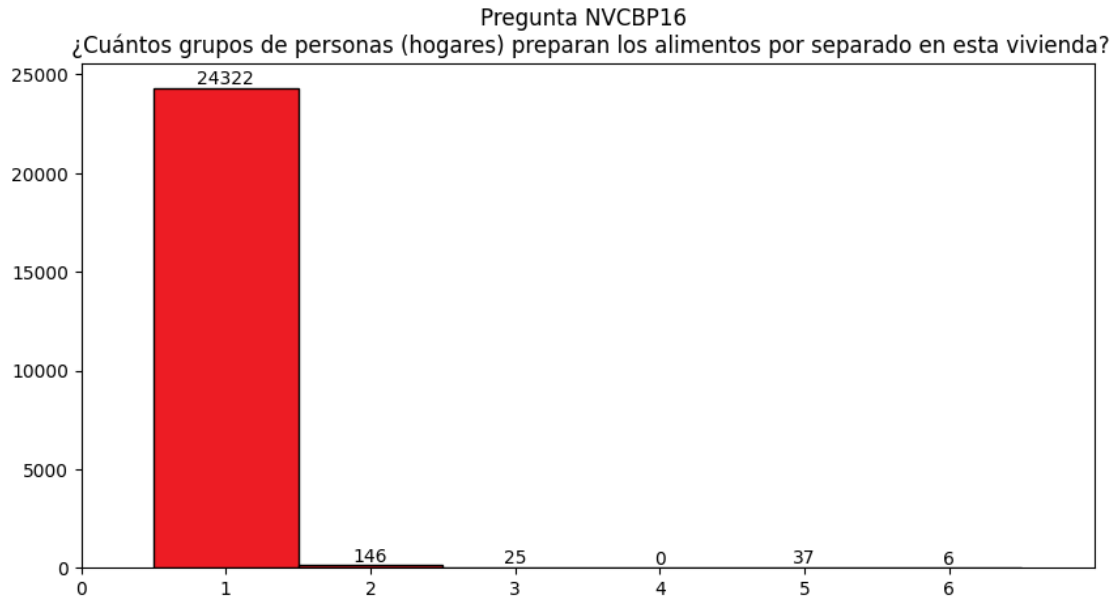
[116]:
```
data['NVCBP16'].count()
```

[116]:
```
24536
```

[117]:
```
data['NVCBP16'].value_counts()
```

[117]:
```
1    24322
2      146
5       37
3       25
6        6
Name: NVCBP16, dtype: int64
```

[121]:
```
plt.figure(figsize=(10,5))
counts, edges, bars = plt.hist(data['NVCBP16'], bins = np.arange(1,8) - 0.5,␣
 ↪edgecolor = 'black', color = '#ED1C24')
#ticklabels = [i for i in range(5)]
#plt.xticks(range(5), ticklabels)
plt.xticks(range(7))
plt.bar_label(bars)
plt.title('Pregunta NVCBP16 \n ¿Cuántos grupos de personas (hogares) preparan␣
 ↪los alimentos por separado en esta vivienda?')
plt.xlim([0,7])
plt.show()
```



[122]:
```
data.to_excel('Encuesta_Multiproposito_Suba.xlsx', index = False)
```

[ ]:
```
```