1 - Parte EDA & Limpieza - Sección NVCBP

December 14, 2022

1 Análisis de la Encuesta Multiproposito

1.1 Importación de Paquetes y carga del archivo

[26]:
```
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
from scipy import stats
import numpy as np
```

[27]:
```
data = pd.read_excel('Encuesta_Multiproposito_Suba.xlsx')
```

[28]:
```
data.shape
```

[28]:
```
(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

[29]:
```
data['NVCBP1'].value_counts()
```

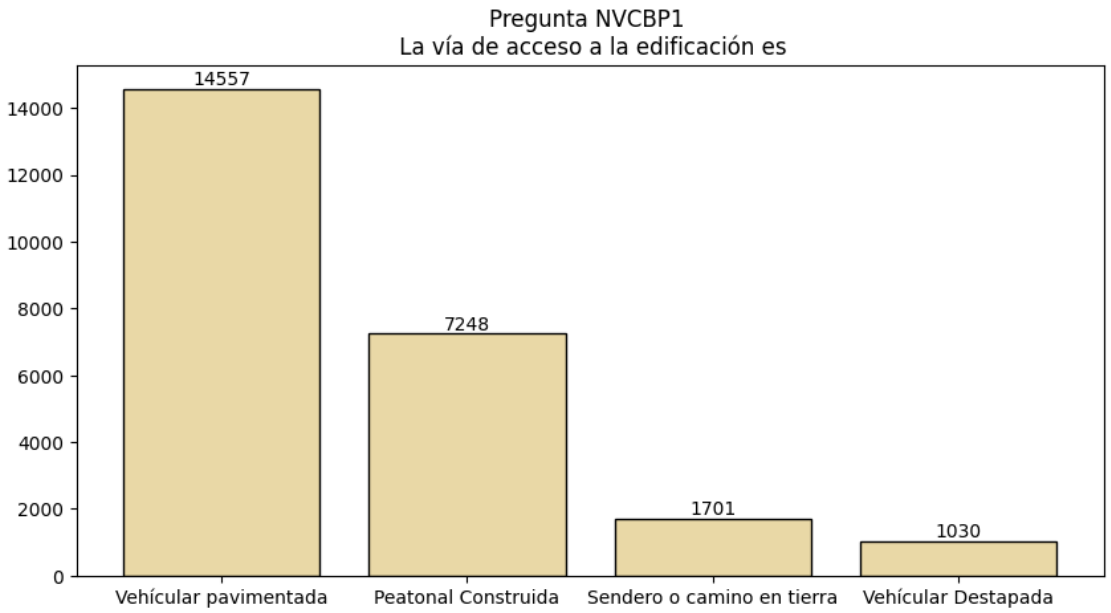
[29]:
```
4    14557
2     7248
1     1701
3     1030
Name: NVCBP1, dtype: int64
```

[30]:
```
data['NVCBP1'] = data['NVCBP1'].replace([1,2,3,4],["Sendero o camino en␣
 ↪tierra", "Peatonal Construida","Vehícular Destapada",
                                                    "Vehícular pavimentada"])
```

[31]:
```
data['NVCBP1'].value_counts()
```

[31]:
```
Vehícular pavimentada         14557
Peatonal Construida            7248
Sendero o camino en tierra     1701
Vehícular Destapada            1030
Name: NVCBP1, dtype: int64
```

[32]:
```
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

[33]:
```
data['NVCBP2'].value_counts()
```

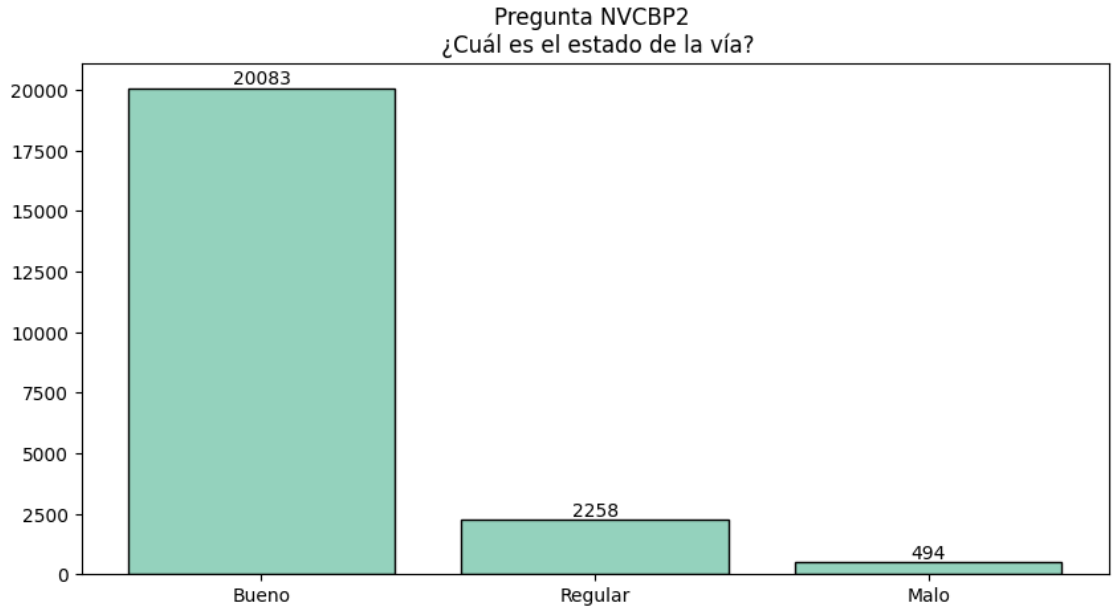
[33]:
```
1.0    20083
2.0     2258
3.0      494
Name: NVCBP2, dtype: int64
```

[34]:
```
data['NVCBP2'] = data['NVCBP2'].replace([1,2,3],["Bueno",
                                                 "Regular",
                                                 "Malo"])
```

[35]:
```
data['NVCBP2'].value_counts()
```

[35]:
```
Bueno      20083
Regular     2258
Malo         494
Name: NVCBP2, dtype: int64
```

[36]:
```
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

[37]:
```
data['NVCBP3'].value_counts()
```

[37]:
```
1    21953
0     2583
Name: NVCBP3, dtype: int64
```

[43]:
```
data['NVCBP3'] = data['NVCBP3'].replace([0,1],["No","Si"])
```

[44]:
```
data['NVCBP3'].value_counts()
```

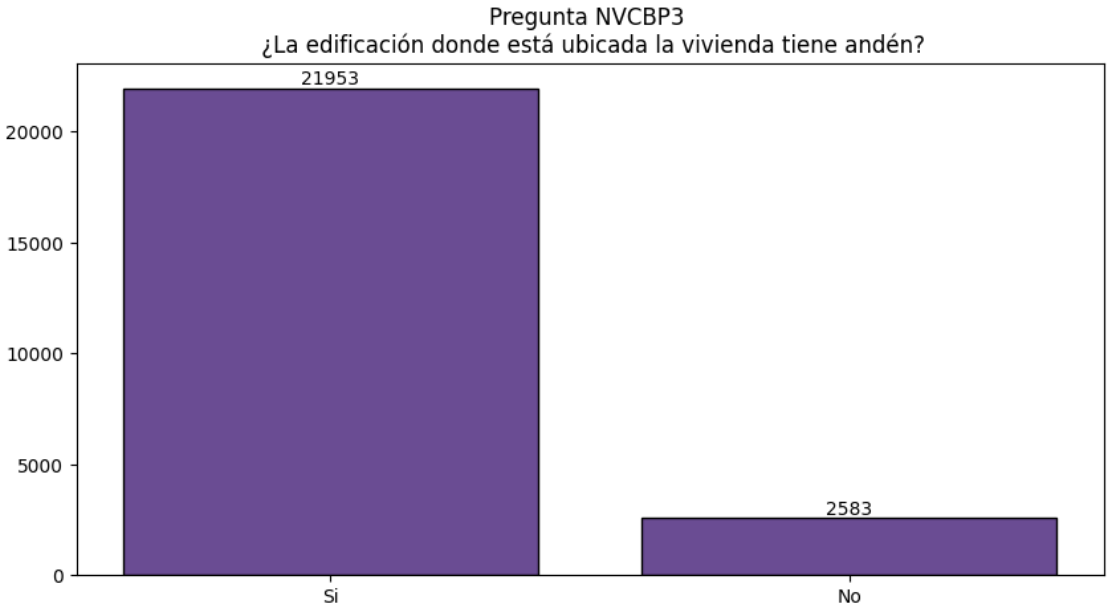
[44]:
```
Si    21953
No     2583
Name: NVCBP3, dtype: int64
```

[45]:
```
data = data.replace({'NVCBP3':2},0)
```

[46]:
```
data['NVCBP3'].value_counts()
```

[46]:
```
Si    21953
No     2583
Name: NVCBP3, dtype: int64
```

[47]:
```
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

[48]:
```
data['NVCBP4'].value_counts()
```

[48]:
```
1    15526
0     9010
Name: NVCBP4, dtype: int64
```

[49]:
```
data['NVCBP4'] = data['NVCBP4'].replace([0,1],["No","Si"])
```

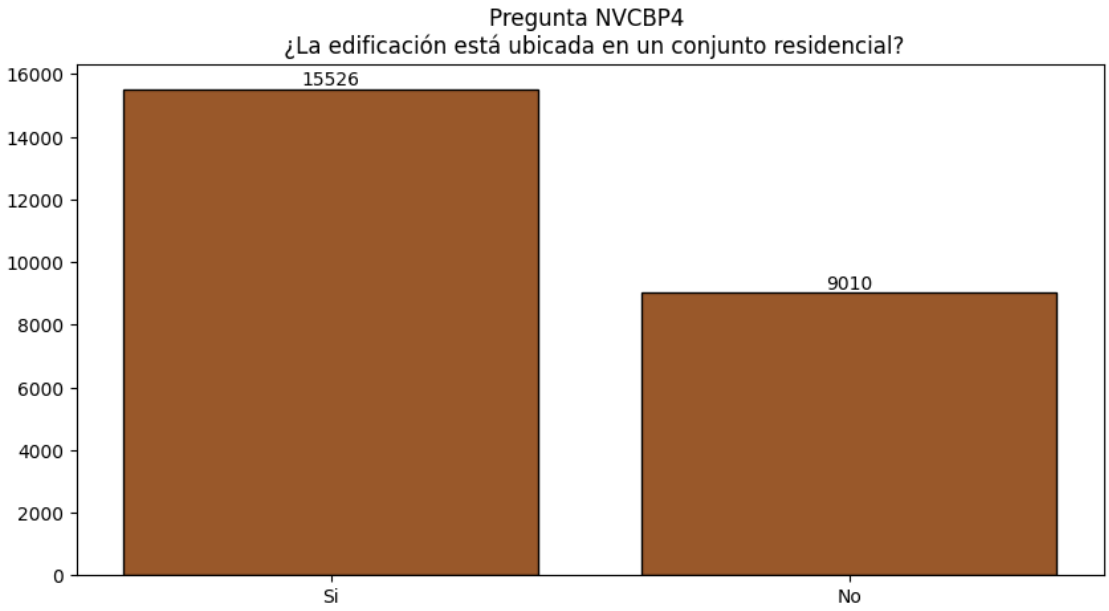
[50]:
```
data = data.replace({'NVCBP4':2},0)
```

[51]:
```
data['NVCBP4'].value_counts()
```

[51]:
```
Si    15526
No     9010
Name: NVCBP4, dtype: int64
```

[52]:
```
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

[53]:
```
data['NVCBP5'].value_counts()
```

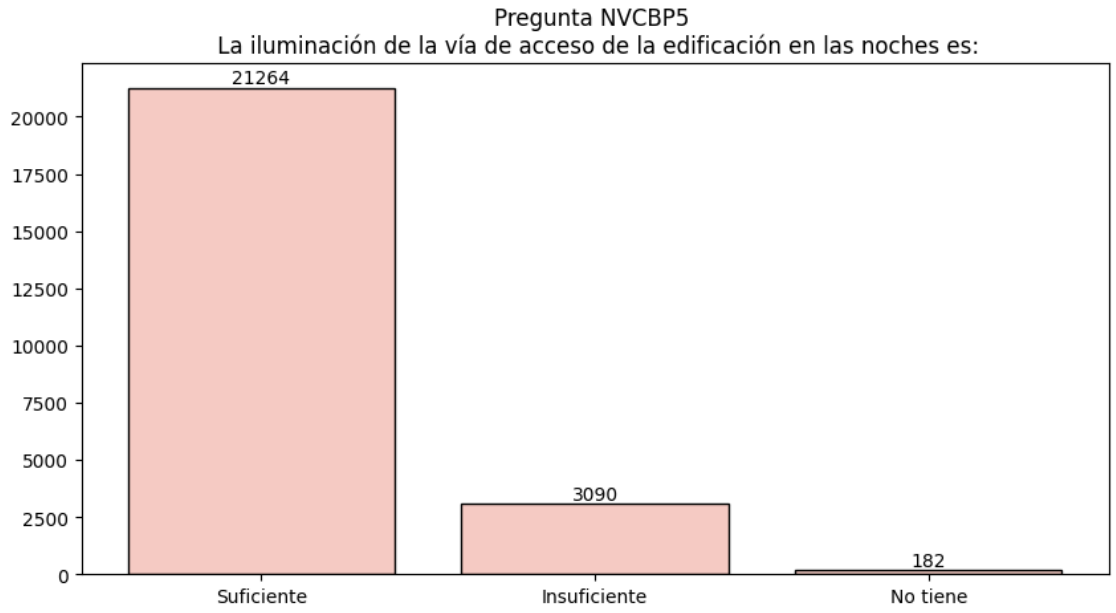
[53]:
```
1    21264
2     3090
3      182
Name: NVCBP5, dtype: int64
```

[54]:
```
data['NVCBP5'] = data['NVCBP5'].
 ↪replace([1,2,3],["Suficiente","Insuficiente","No tiene"])
```

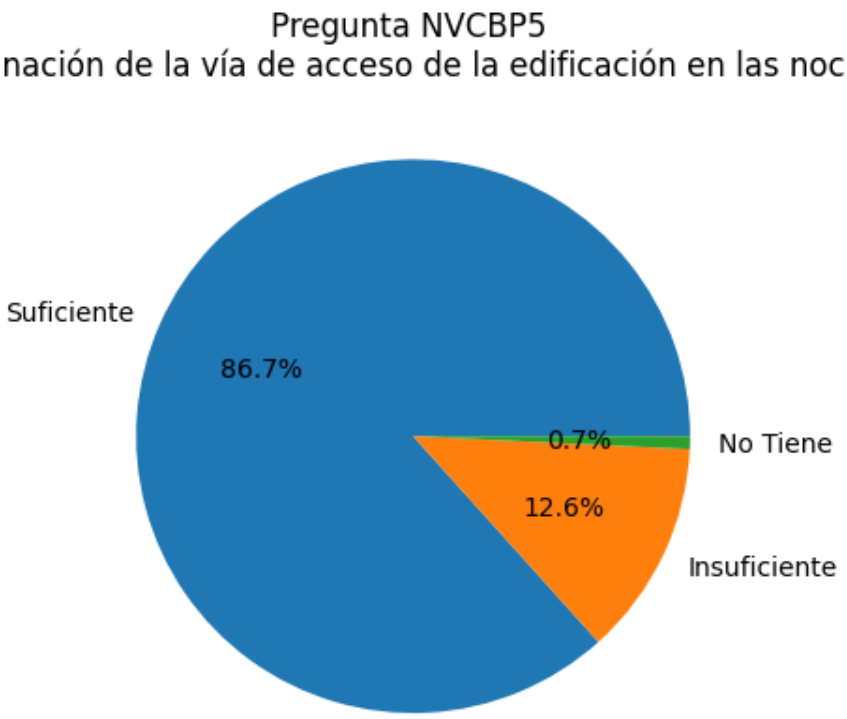
[55]:
```
data['NVCBP5'].value_counts()
```

[55]:
```
Suficiente      21264
Insuficiente     3090
No tiene          182
Name: NVCBP5, dtype: int64
```

[56]:
```
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()
```



[59]:
```
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?

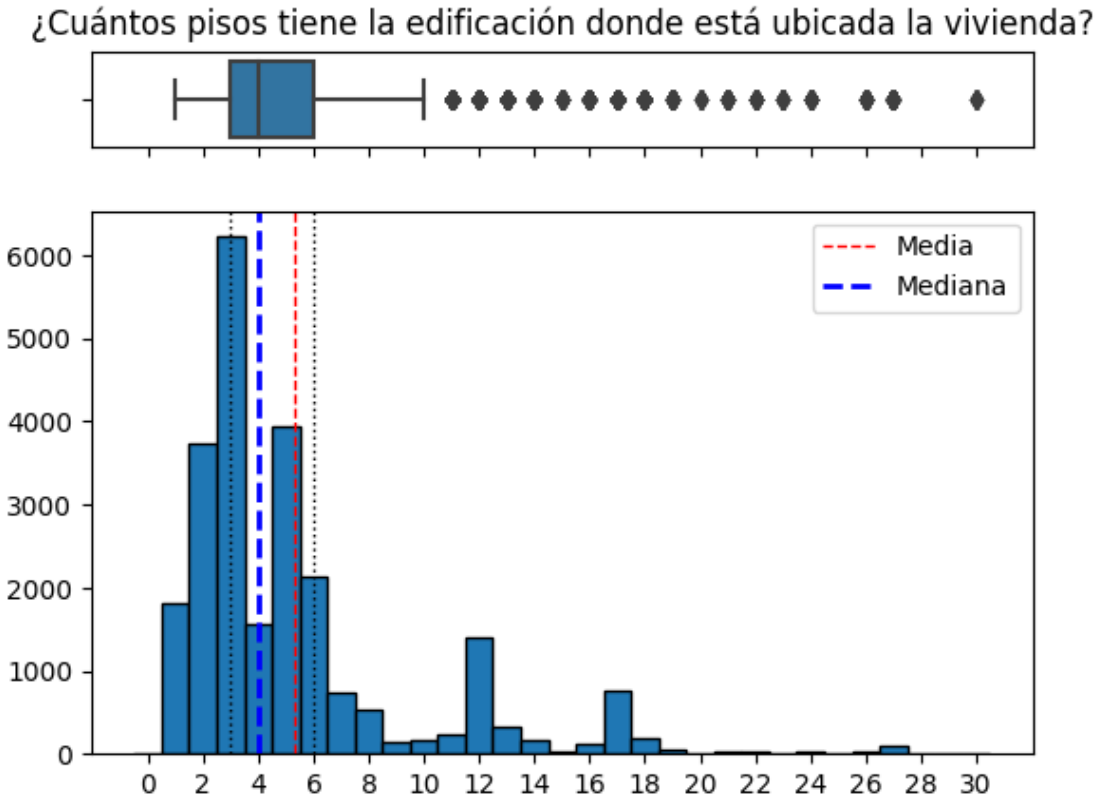
Datos: 24536

[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
```

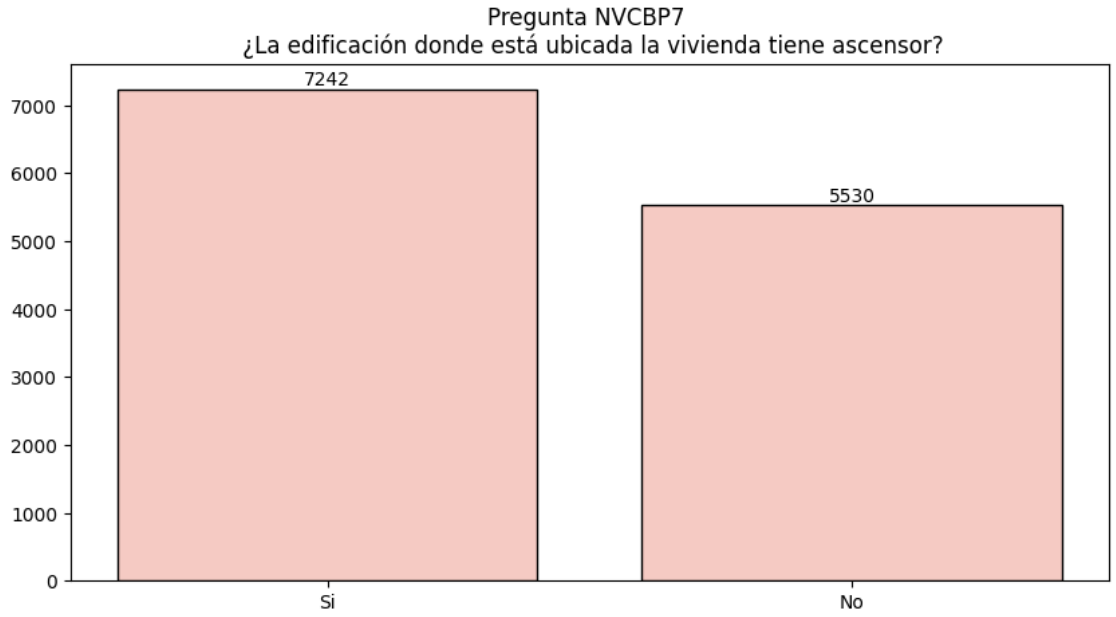
[60]:
```
data['NVCBP7'] = data['NVCBP7'].replace([0,1],["No","Si"])
```

[61]:
```
data = data.replace({'NVCBP7':2},0)
```

[62]:
```
data['NVCBP7'].value_counts()
```

[62]:
```
Si    7242
No    5530
Name: NVCBP7, dtype: int64
```

[63]:
```
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

[64]:
```
comercio = ['NVCBP9','NVCBP9A1','NVCBP9A2','NVCBP9A3','NVCBP9A4']
```

[65]:
```
data[comercio[0]].describe()
```

[65]:
```
count    24536.000000
mean         0.024821
```

```
std          0.155581
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
max          1.000000
Name: NVCBP9, dtype: float64
```

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

[66]:
```
for i in comercio:
    data[i] = data[i].replace([0,1],["No","Si"])
```

[67]:
```
for i in comercio:
    print(data[i].count())
```

```
24536
609
609
609
609
```

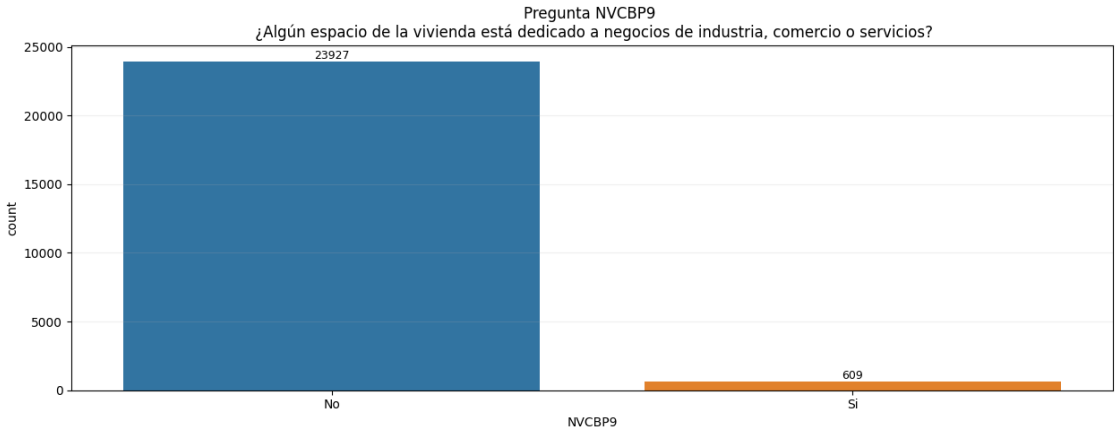
[68]:
```
for i in comercio:
    print(data[i].value_counts())
```

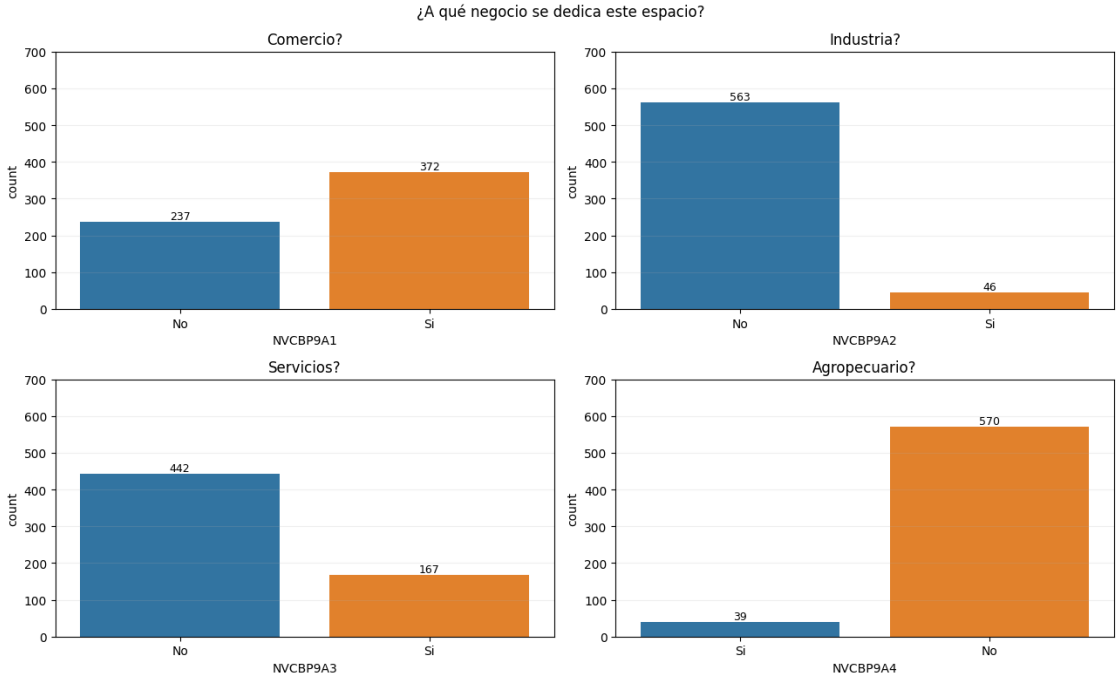
```
No    23927
Si      609
Name: NVCBP9, dtype: int64
Si    372
No    237
Name: NVCBP9A1, dtype: int64
No    563
Si     46
Name: NVCBP9A2, dtype: int64
No    442
Si    167
Name: NVCBP9A3, dtype: int64
No    570
Si     39
Name: NVCBP9A4, dtype: int64
```

[69]:
```
comercio_2 =['NVCBP9A1','NVCBP9A2','NVCBP9A3','NVCBP9A4']
l_com = ['Comercio?','Industria?','Servicios?','Agropecuario?']
```

[70]:
```
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

[71]:
```
data['NVCBP10'].value_counts()
```

[71]:
```
2    15344
1     9006
3      186
Name: NVCBP10, dtype: int64
```

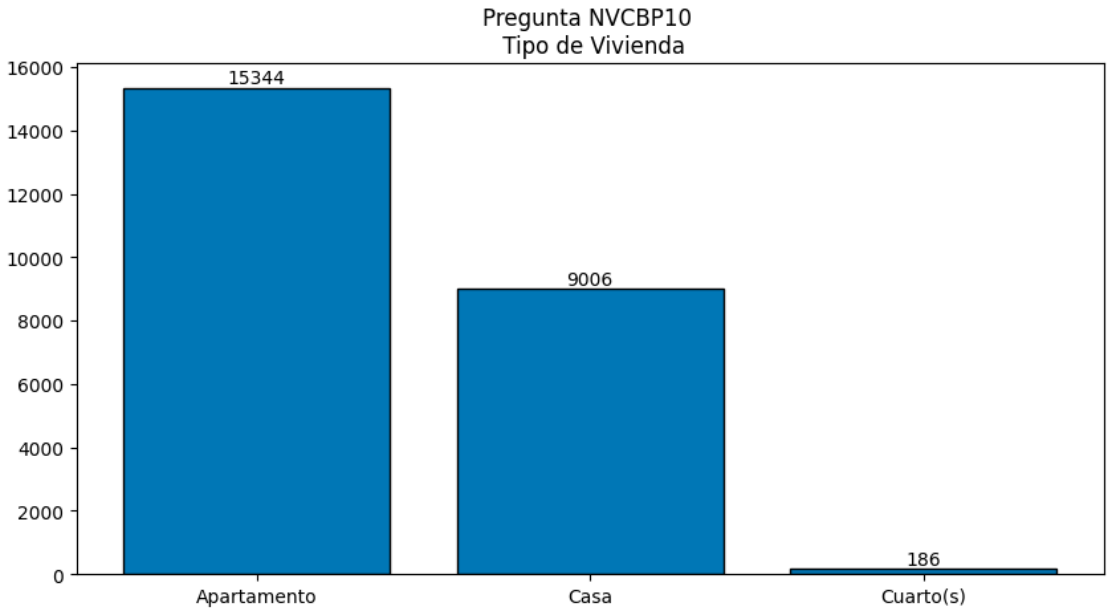
[72]:
```
data['NVCBP10'] = data['NVCBP10'].
  ↪replace([1,2,3,4],["Casa","Apartamento","Cuarto(s)","Otro"])
```

[73]:
```
data['NVCBP10'].value_counts()
```

[73]:
```
Apartamento    15344
Casa            9006
Cuarto(s)        186
```

```
Name: NVCBP10, dtype: int64
```

[75]:
```
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()
```

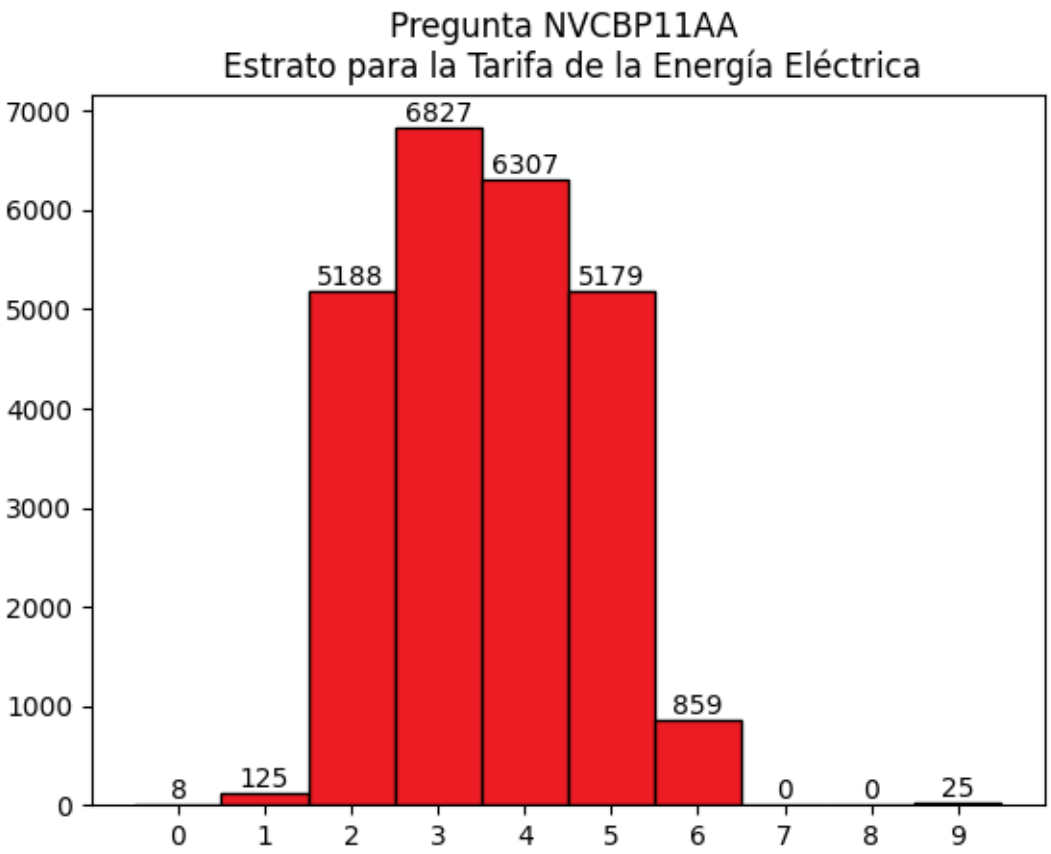
[67]:
```
count    24518.000000
mean         3.568154
std          1.168924
min          0.000000
25%          3.000000
50%          4.000000
75%          4.000000
max          9.000000
```

```
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
```

[76]:
```
data['NVCBP11A'].value_counts()
```

[76]:
```
1    24518
0       18
Name: NVCBP11A, dtype: int64
```

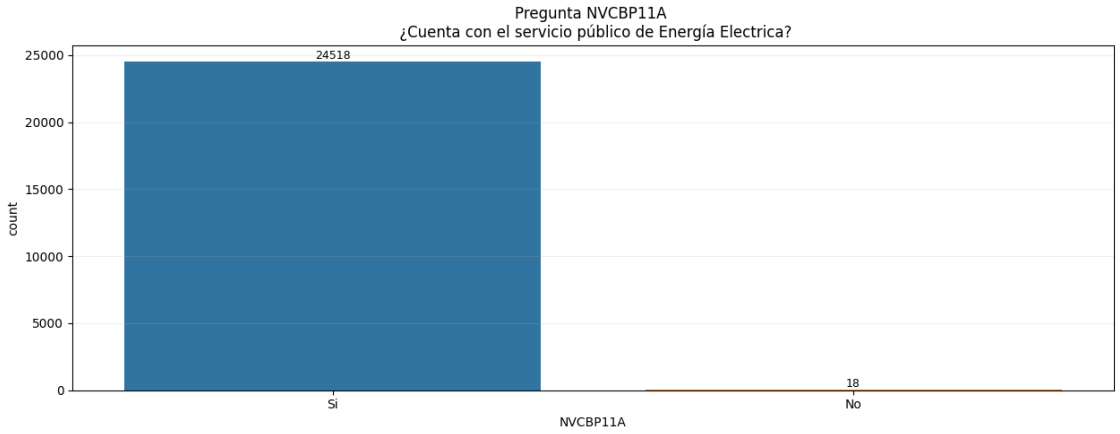
[78]:
```
data = data.replace({'NVCBP11A':2},0)
```

[77]:
```
data['NVCBP11A'] = data['NVCBP11A'].replace([0,1],["No","Si"])
```

[78]:
```
data['NVCBP11A'].value_counts()
```

[78]:
```
Si    24518
No       18
Name: NVCBP11A, dtype: int64
```

[79]:
```
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
```

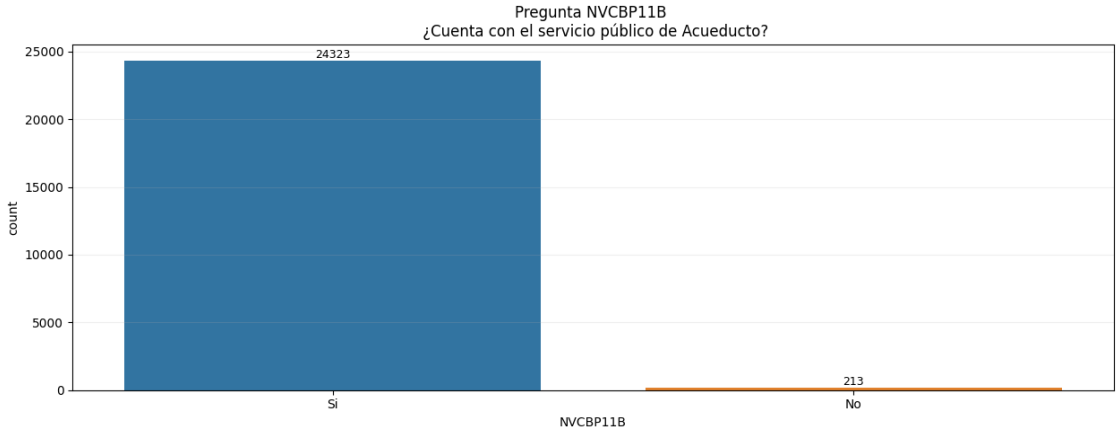
```
Name: NVCBP11B, dtype: int64
```

[80]:
```
data['NVCBP11B'] = data['NVCBP11B'].replace([0,1],["No","Si"])
```

[81]:
```
data['NVCBP11B'].value_counts()
```

[81]:
```
Si    24323
No      213
Name: NVCBP11B, dtype: int64
```

[83]:
```
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()
```

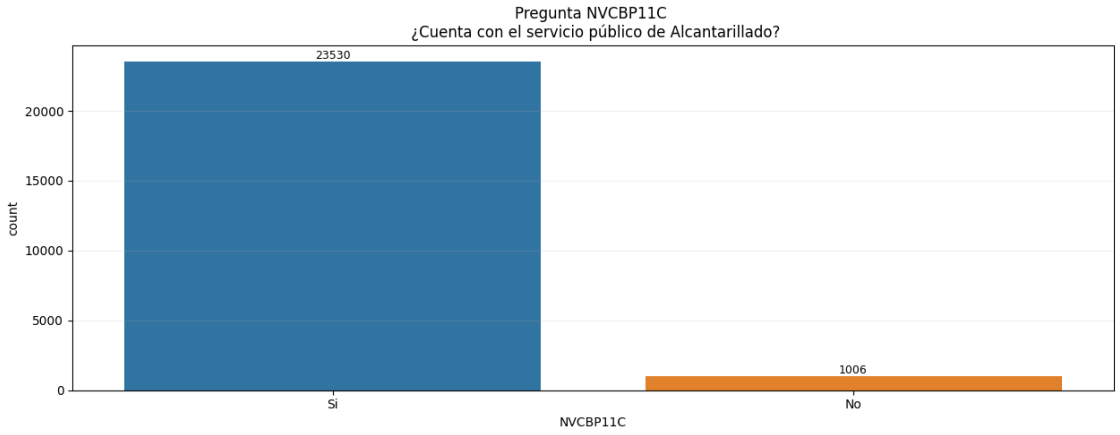
[91]:
```
1    23530
0     1006
Name: NVCBP11C, dtype: int64
```

[84]:
```
data['NVCBP11C'] = data['NVCBP11C'].replace([0,1],["No","Si"])
```

[85]:
```
data['NVCBP11C'].value_counts()
```

[85]:
```
Si    23530
No     1006
Name: NVCBP11C, dtype: int64
```

[86]:
```
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()
```

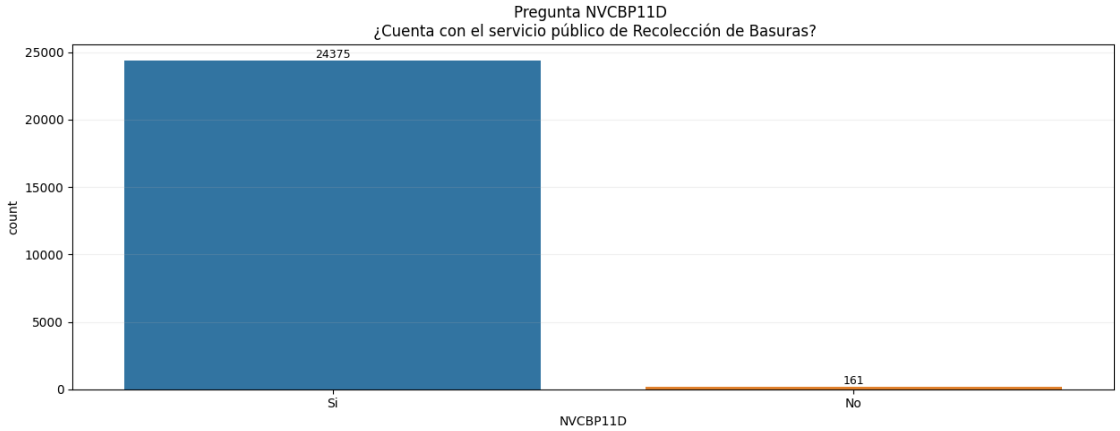
[96]:
```
1    24375
0      161
Name: NVCBP11D, dtype: int64
```

[87]:
```
data['NVCBP11D'] = data['NVCBP11D'].replace([0,1],["No","Si"])
```

[88]:
```
data['NVCBP11D'].value_counts()
```

[88]:
```
Si    24375
No      161
Name: NVCBP11D, dtype: int64
```

[89]:
```
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

[91]:
```
problemas = [i for i in data.columns if ('NVCBP14') in i]
```

[98]:
```
for i in problemas:
    print(i, data[i].value_counts())
```

```
NVCBP14A No    22492
Si      2044
Name: NVCBP14A, dtype: int64
NVCBP14B No    23292
Si      1244
Name: NVCBP14B, dtype: int64
NVCBP14C No    23927
Si       609
Name: NVCBP14C, dtype: int64
NVCBP14D No    22778
```

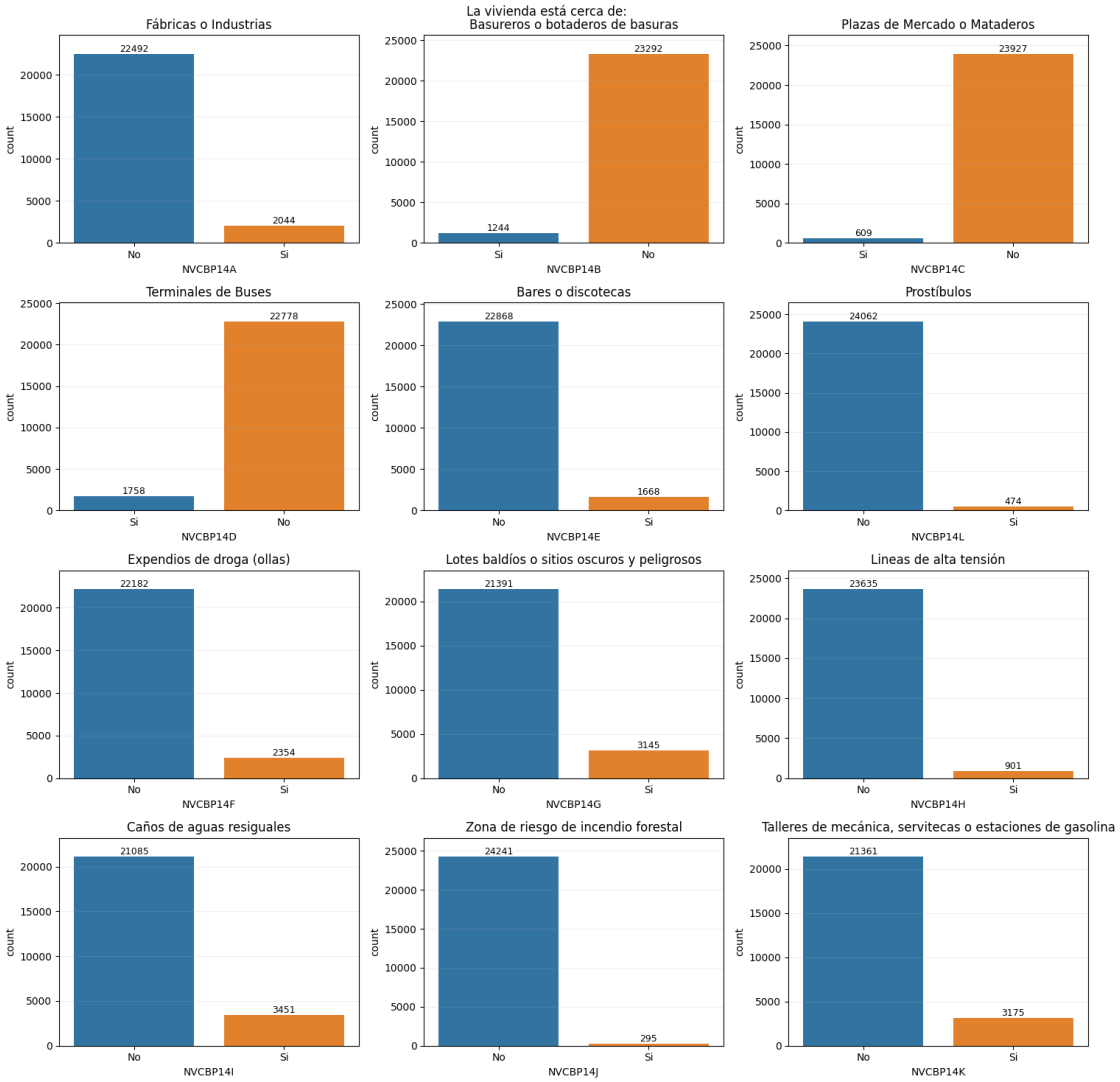
```
Si     1758
Name: NVCBP14D, dtype: int64
NVCBP14E No    22868
Si     1668
Name: NVCBP14E, dtype: int64
NVCBP14L No    24062
Si      474
Name: NVCBP14L, dtype: int64
NVCBP14F No    22182
Si     2354
Name: NVCBP14F, dtype: int64
NVCBP14G No    21391
Si     3145
Name: NVCBP14G, dtype: int64
NVCBP14H No    23635
Si      901
Name: NVCBP14H, dtype: int64
NVCBP14I No    21085
Si     3451
Name: NVCBP14I, dtype: int64
NVCBP14J No    24241
Si      295
Name: NVCBP14J, dtype: int64
NVCBP14K No    21361
Si     3175
Name: NVCBP14K, dtype: int64
```

[99]:
```
for i in problemas:
    data = data.replace({i:2},0)
```

[100]:
```
for i in problemas:
    data[i] = data[i].replace([0,1],["No","Si"])
```

[101]:
```
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']
```

[102]:
```
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

[103]:
```
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']
```

[104]:
```
for i in problemas15:
    print(i,data[i].value_counts())
```

```
NVCBP15A 0    16436
1     8100
Name: NVCBP15A, dtype: int64
NVCBP15B 0    22123
1     2413
Name: NVCBP15B, dtype: int64
NVCBP15C 0    13334
1    11202
Name: NVCBP15C, dtype: int64
NVCBP15D 0    19856
1     4680
Name: NVCBP15D, dtype: int64
NVCBP15E 0    19918
1     4618
Name: NVCBP15E, dtype: int64
NVCBP15F 0    20891
1     3645
```

```
Name: NVCBP15F, dtype: int64
NVCBP15G 0    21153
1     3383
Name: NVCBP15G, dtype: int64
NVCBP15H 0    21342
1     3194
Name: NVCBP15H, dtype: int64
NVCBP15I 0    22151
1     2385
Name: NVCBP15I, dtype: int64
NVCBP15J 0    22406
1     2130
Name: NVCBP15J, dtype: int64
NVCBP15K 0    24415
1      121
Name: NVCBP15K, dtype: int64
NVCBP15L 0    24041
1      495
Name: NVCBP15L, dtype: int64
NVCBP15M 0    23660
1      876
Name: NVCBP15M, dtype: int64
```

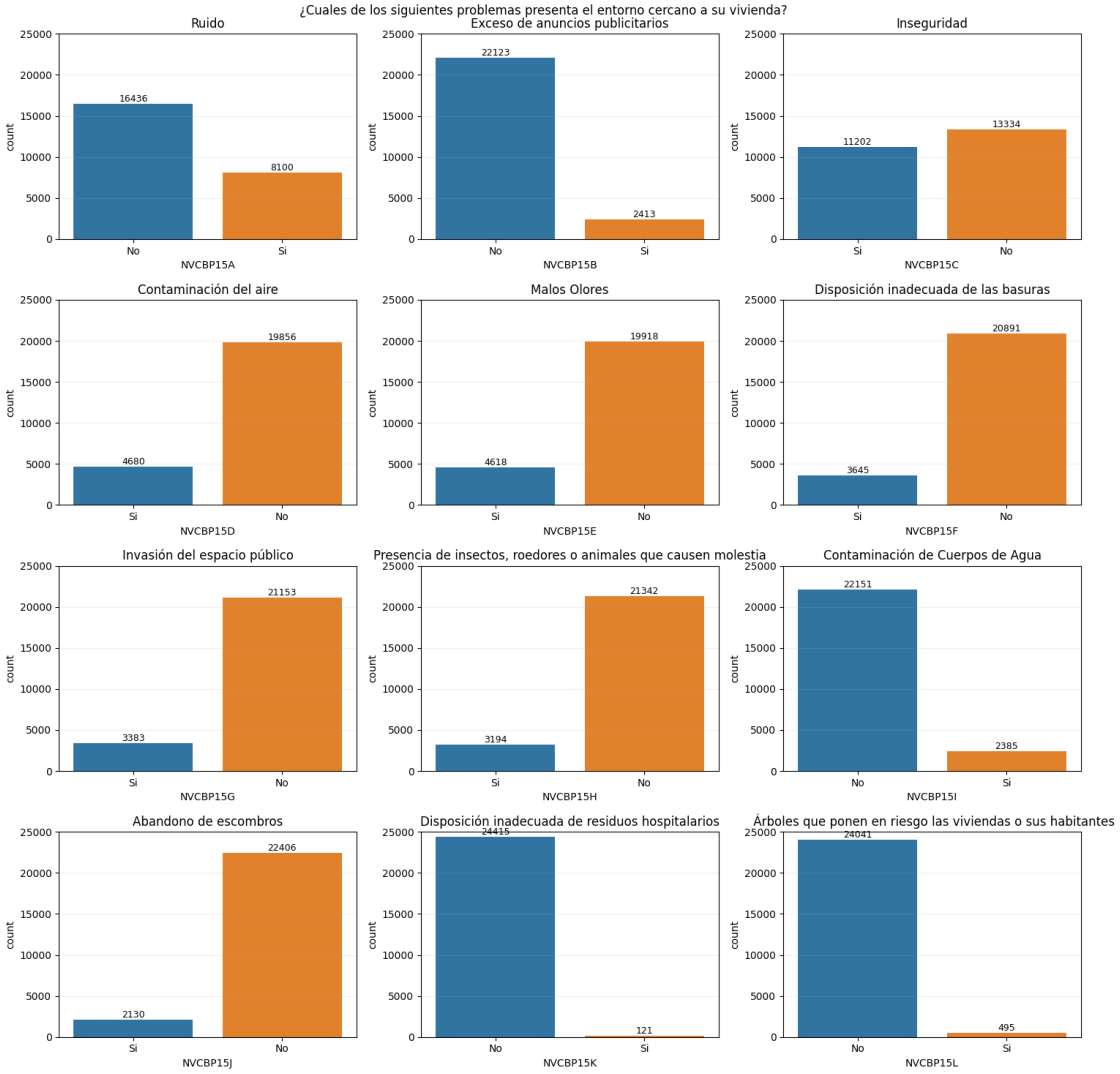
[110]:
```
for i in problemas15:
    data = data.replace({i:2},0)
```

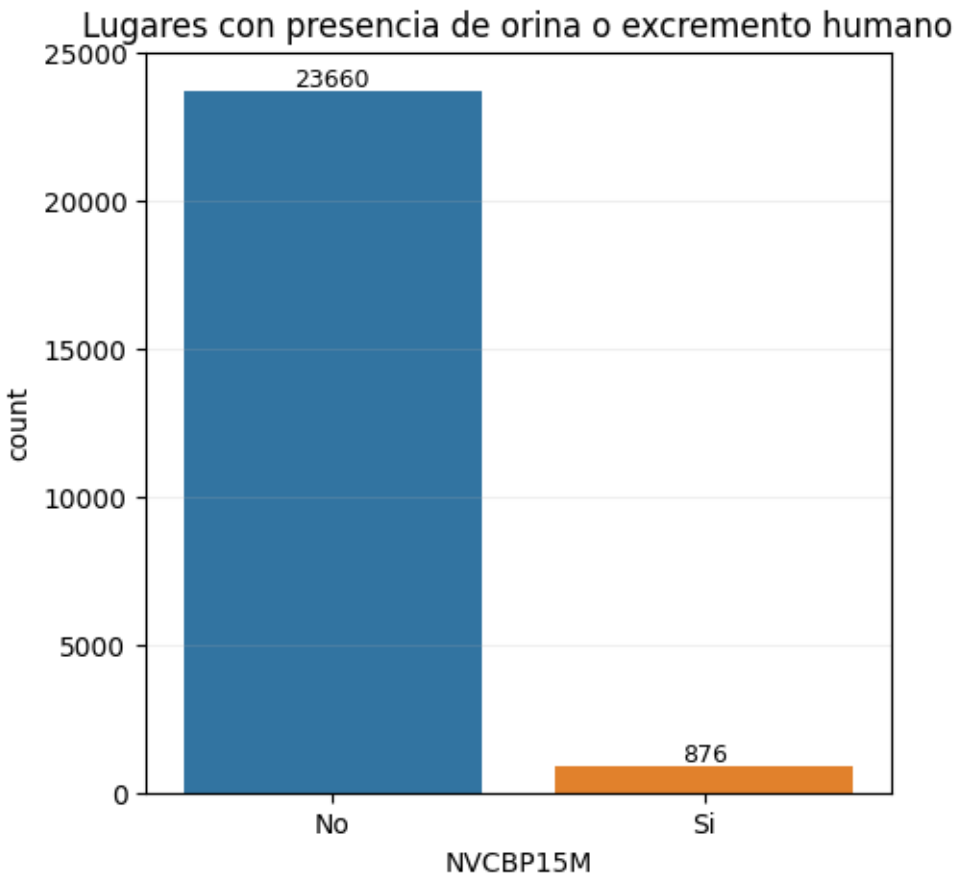
[105]:
```
for i in problemas15:
    data[i] = data[i].replace([0,1],["No","Si"])
```

[106]:
```
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']
```

[107]:
```
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
No
Si
NVCBP15A
Exceso de anuncios publicitarios
22123
2413
No
Si
NVCBP15B
Inseguridad
11202
13334
Si
No
NVCBP15C
Contaminación del aire
4680
19856
Si
No
NVCBP15D
Malos Olores
4618
19918
Si
No
NVCBP15E
Disposición inadecuada de las basuras
3645
20891
Si
No
NVCBP15F
Invasión del espacio público
3383
21153
Si
No
NVCBP15G
Presencia de insectos, roedores o animales que causen molestia
3194
21342
Si
No
NVCBP15H
Contaminación de Cuerpos de Agua
22151
2385
No
Si
NVCBP15I
Abandono de escombros
2130
22406
Si
No
NVCBP15J
Disposición inadecuada de residuos hospitalarios
24415
121
No
Si
NVCBP15K
Árboles que ponen en riesgo las viviendas o sus habitantes
24041
495
No
Si
NVCBP15L
count



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

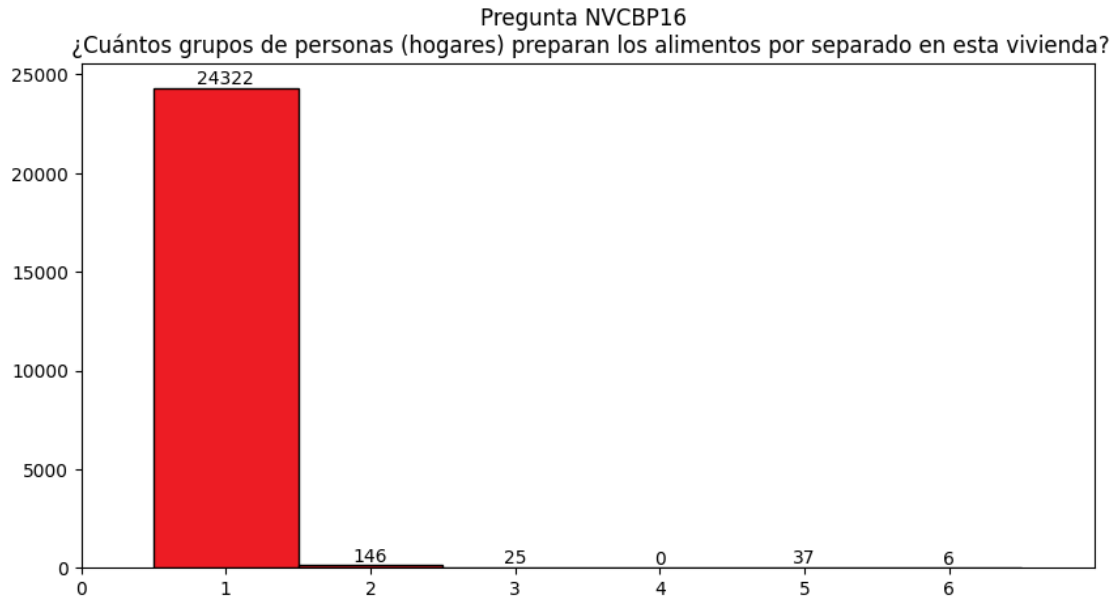
[116]:
```
data['NVCBP16'].count()
```

[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()
```



[108]:
```
data.to_excel('Encuesta_Multiproposito_Suba.xlsx', index = False)
```

[ ]:
```
```