


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

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



NVCBP11B

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

Datos: 24536

```
In [80]: data['NVCBP11B'].count()
Out[80]: 24536
```

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

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

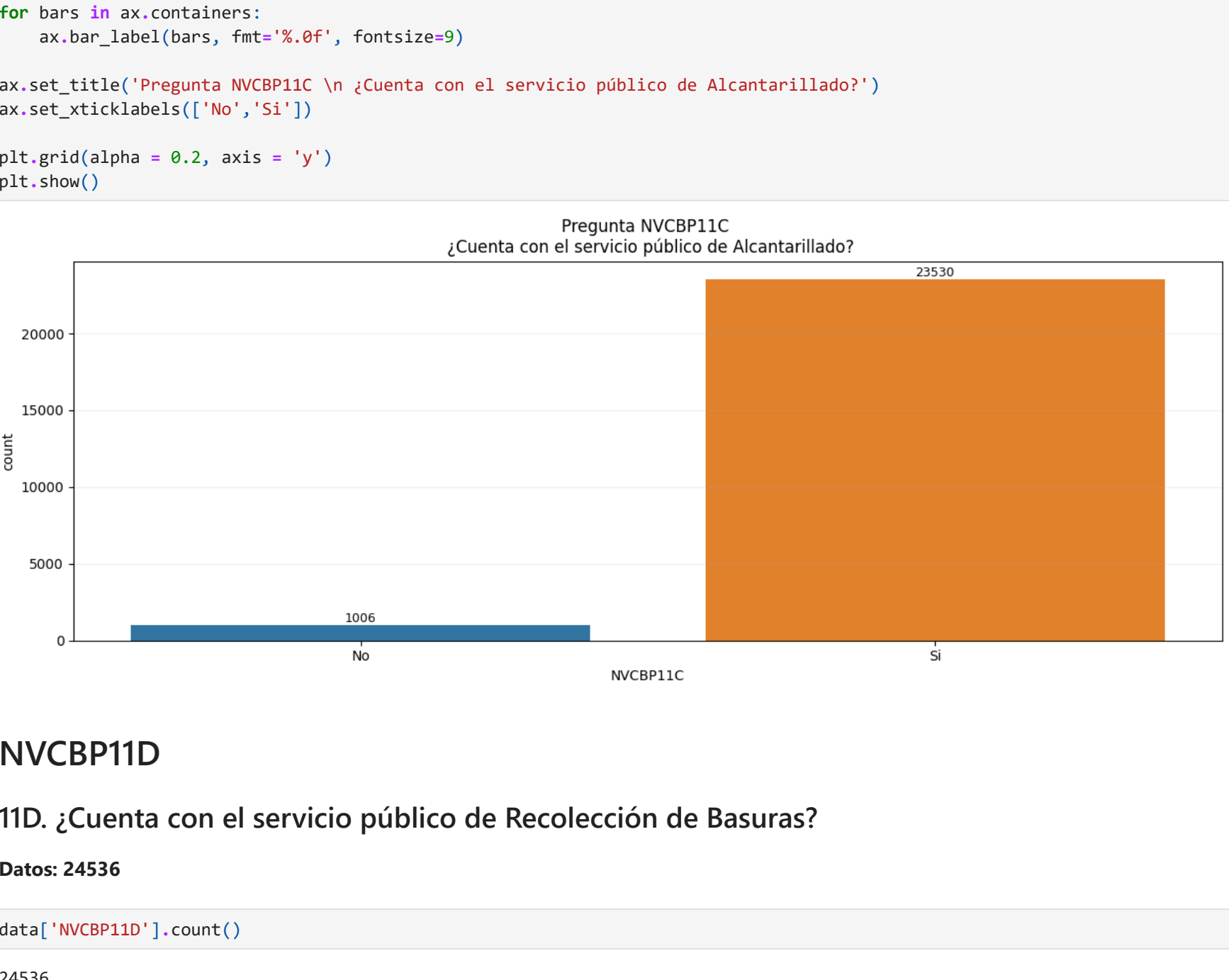
```
Out[82]: 1    24323
0         213
Name: NVCBP11B, dtype: int64
```

```
In [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()
```



NVCBP11C

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

Datos: 24536

```
In [89]: data['NVCBP11C'].count()
Out[89]: 24536
```

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

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

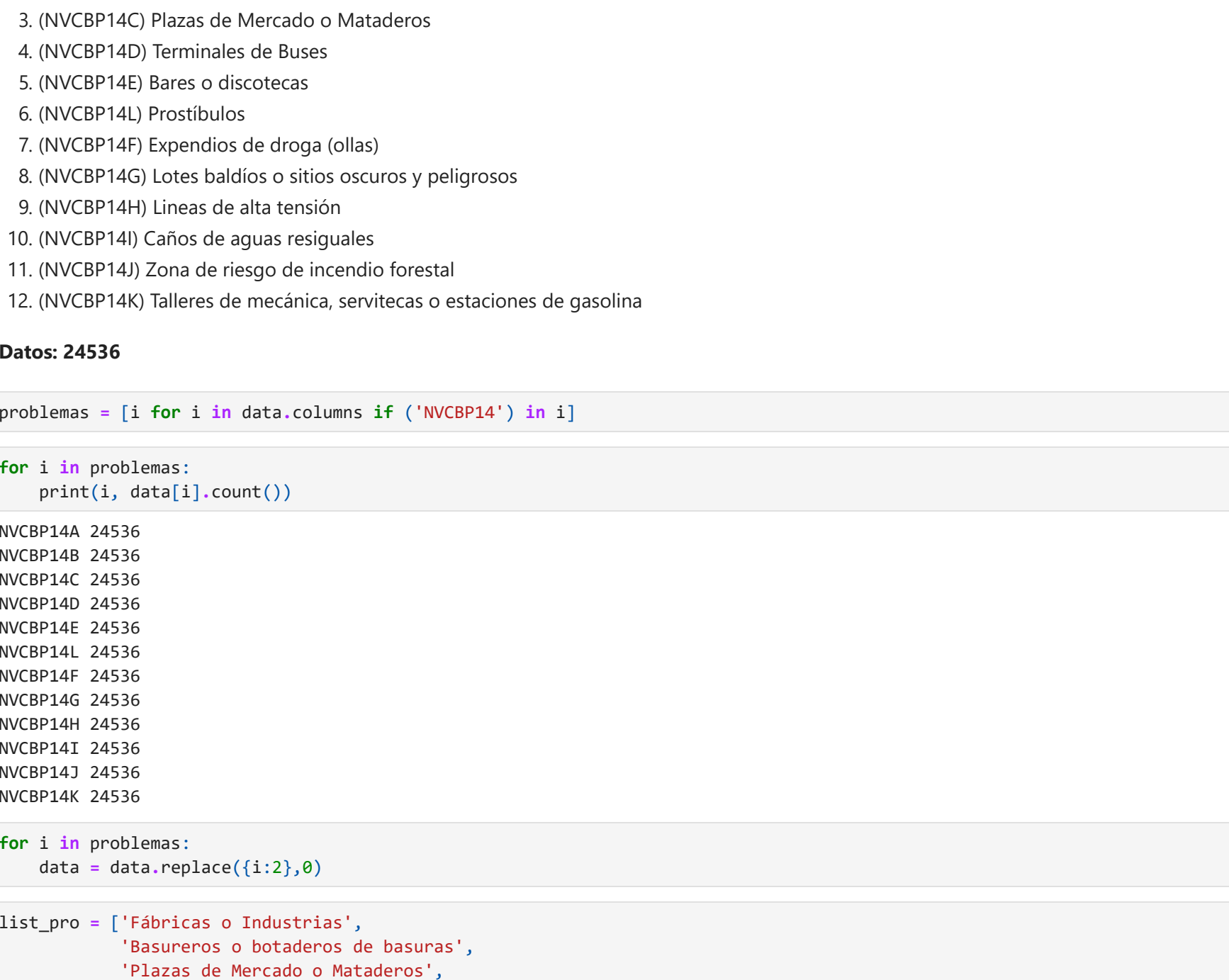
```
Out[91]: 1    21350
0       1006
Name: NVCBP11C, dtype: int64
```

```
In [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()
```



NVCBP11D

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

Datos: 24536

```
In [94]: data['NVCBP11D'].count()
Out[94]: 24536
```

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

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

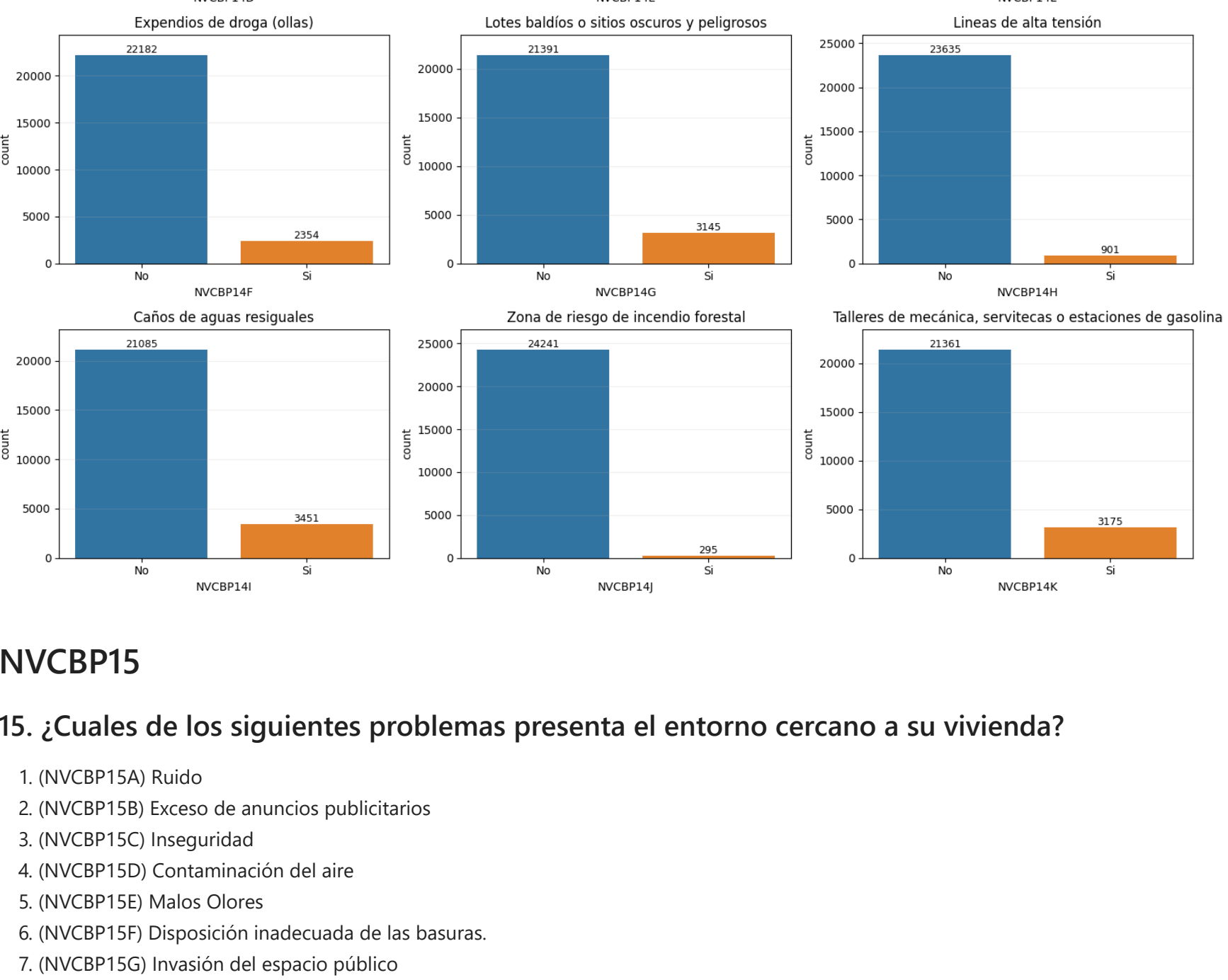
```
Out[96]: 1    24375
0       161
Name: NVCBP11D, dtype: int64
```

```
In [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()
```



NVCBP14

14. La vivienda está cerca de:

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

Datos: 24536

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

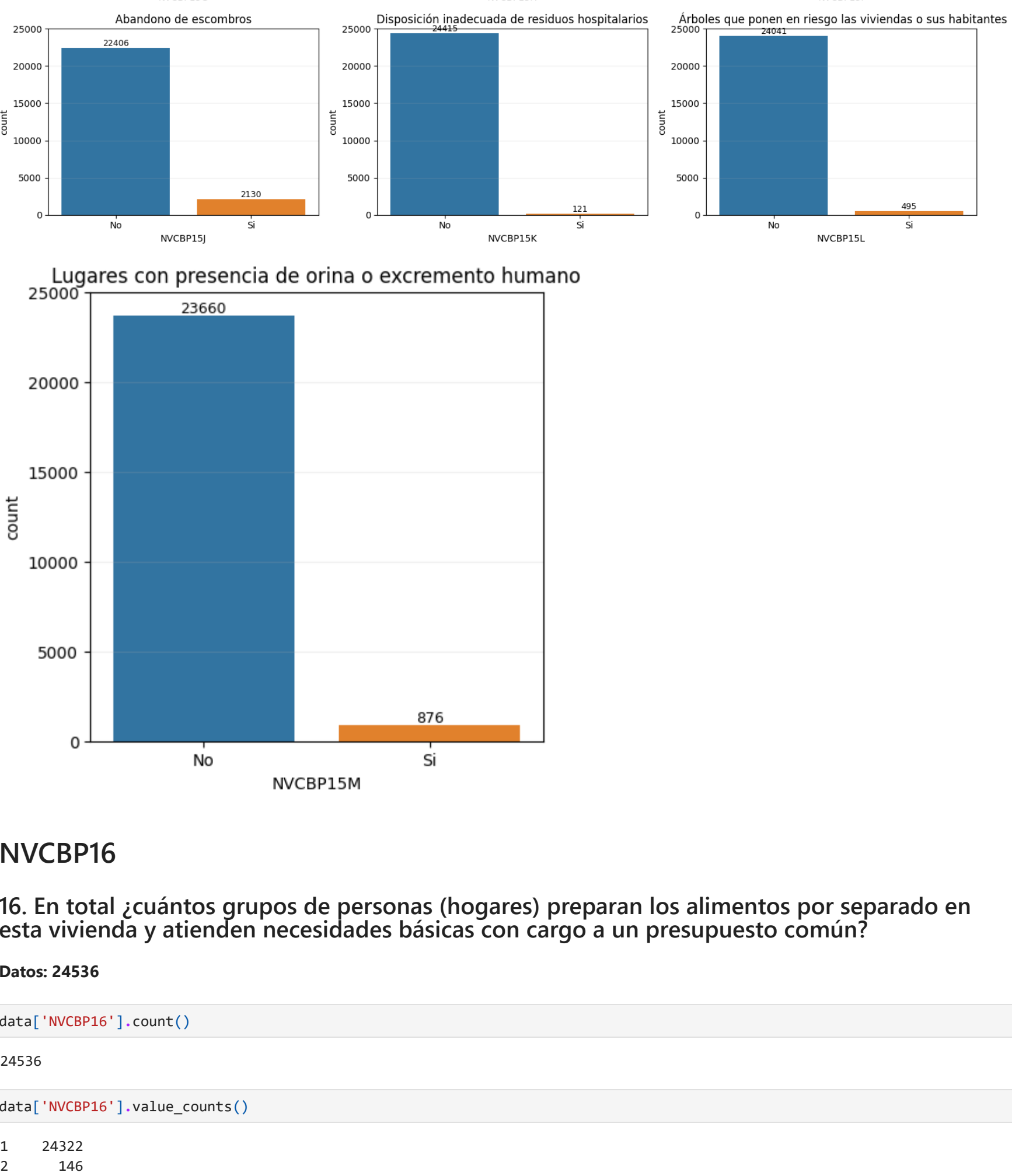
In [101]: for i in problemas:
print(i, data[i].count())
```

```
NVCBP14A 24536
NVCBP14B 24536
NVCBP14C 24536
NVCBP14D 24536
NVCBP14E 24536
NVCBP14F 24536
NVCBP14H 24536
NVCBP14I 24536
NVCBP14J 24536
NVCBP14K 24536
```

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

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

```
In [107]: fig, axes = plt.subplots(4,3, figsize = (15,15), squeeze=False)
fig.subplots_adjust(top=0.9)
ax11 = axes.flatten()
fig.suptitle('La vivienda está cerca de:')
for ax,cols, names in zip(ax11,problemas,list_pro):
    sns.countplot(x = cols, data = data, ax = ax)
    ax.grid(alpha = 0.2, axis = 'y')
    ax.set_title('(' + 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()
```



NVCBP15

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) Árboles que ponen en riesgo las viviendas o sus habitantes
13. (NVCBP15M) Lugares con presencia de orina o excremento humano

Datos: 24536

```
In [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']
```

```
In [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
```

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

```
In [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']
```

```
In [114]: fig, axes = plt.subplots(4,3, figsize = (15,15), squeeze=False)
fig.subplots_adjust(top=0.9)
ax11 = axes.flatten()
fig.suptitle('¿Cuales de los siguientes problemas presenta el entorno cercano a su vivienda?')
for ax,cols, names in zip(ax11,problemas15,list_problemas):
    sns.countplot(x = cols, data = data, ax = ax)
    ax.grid(alpha = 0.2, axis = 'y')
    ax.set_title('(' + 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()
```



NVCBP16

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

```
In [116]: data['NVCBP16'].count()
Out[116]: 24536
```

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

Out[117]: 1    24322
2       146
3        37
4         25
5          6
6          0
Name: NVCBP16, dtype: int64
```

```
In [121]: plt.figure(figsize=(10,5))
counts, edges, bars = plt.hist(data['NVCBP16'], bins = np.arange(1,8) - 0.5, edgecolor = 'black', color = '#ED1C24')
plt.xticks(range(5))
plt.yticks(range(5))
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.xlabel(0,7)
plt.show()
```



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

In [ ]:
```