

6 - Parte EDA & Limpieza - Sección NPCGP

December 14, 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_excel('Encuesta_Multiproposito_Suba.xlsx')
```

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

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

Hay en total 24536 encuestados en la Localidad de Suba

2 Sexta Sección (NPCGP)

2.1 NPCGP1

2.1.1 1. ¿Dónde o con quién permanece ...durante la mayor parte del tiempo entre semana?

1. Asiste a un hogar comunitario, jardín, centro de desarrollo infantil o colegio
2. Con su padre o madre en la casa
3. Con su padre o madre en el trabajo
4. Con empleado/a o niño/a en la casa
5. Al cuidado de un/a pariente de 18 años o más
6. Al cuidado de un/a pariente menor de 18 años
7. En casa solo/a
8. Otro

Datos: 1021

```
[6]: data['NPCGP1'].count()
```

```
[6]: 1021
```

```
[10]: data['NPCGP1'].value_counts()
```

```
[10]: Con su padre o madre en la casa
572
Asiste a un hogar comunitario, jardín, centro de desarrollo infantil o colegio
239
Al cuidado de un/a pariente de 18 años o más
122
Con empleado/a o niñero/a en la casa
43
Con su padre o madre en el trabajo
20
Otro
12
Al cuidado de un/a pariente menor de 18 años
11
En casa solo/a
2
Name: NPCGP1, dtype: int64
```

```
[9]: data['NPCGP1'] = data['NPCGP1'].replace([1,2,3,4,5,6,7,8],['Asiste a un hogar_
↳comunitario, jardín, centro de desarrollo infantil o colegio',
'Con su padre o madre en la casa',
'Con su padre o madre en el trabajo',
'Con empleado/a o niñero/a en la casa',
'Al cuidado de un/a pariente de 18 años o más',
'Al cuidado de un/a pariente menor de 18 años',
'En casa solo/a',
'Otro'])
```

```
[11]: data['NPCFP36'].describe().apply("{0:.2f}".format)
```

```
-----
ValueError                                Traceback (most recent call last)
Cell In [11], line 1
----> 1 data['NPCFP36'].describe().apply("{0:.2f}".format)

File ~\Desktop\Otros Proyectos\Encuesta_
↳Multiproposito\venv\lib\site-packages\pandas\core\series.py:4774, in Series.
↳apply(self, func, convert_dtype, args, **kwargs)
    4664 def apply(
    4665     self,
    4666     func: AggFuncType,
    (...)
    4669     **kwargs,
    4670 ) -> DataFrame | Series:
    4671     """
```

```

4672     Invoke function on values of Series.
4673     (...)
4772     dtype: float64
4773     """
-> 4774     return SeriesApply(self, func, convert_dtype, args, kwargs).apply()

```

```

File ~\Desktop\Otros Proyectos\Encuesta
↳ Multiproposito\venv\lib\site-packages\pandas\core\apply.py:1100, in
↳ SeriesApply.apply(self)
    1097     return self.apply_str()
    1099 # self.f is Callable
-> 1100 return self.apply_standard()

```

```

File ~\Desktop\Otros Proyectos\Encuesta
↳ Multiproposito\venv\lib\site-packages\pandas\core\apply.py:1151, in
↳ SeriesApply.apply_standard(self)
    1149     else:
    1150         values = obj.astype(object)._values
-> 1151         mapped = lib.map_infer(
    1152             values,
    1153             f,
    1154             convert=self.convert_dtype,
    1155         )
    1157 if len(mapped) and isinstance(mapped[0], ABCSeries):
    1158     # GH#43986 Need to do list(mapped) in order to get treated as nested
    1159     # See also GH#25959 regarding EA support
    1160     return obj._constructor_expanddim(list(mapped), index=obj.index)

```

```

File ~\Desktop\Otros Proyectos\Encuesta
↳ Multiproposito\venv\lib\site-packages\pandas\_libs\lib.pyx:2919, in pandas.
↳ _libs.lib.map_infer()

```

ValueError: Unknown format code 'f' for object of type 'str'

[]:

```

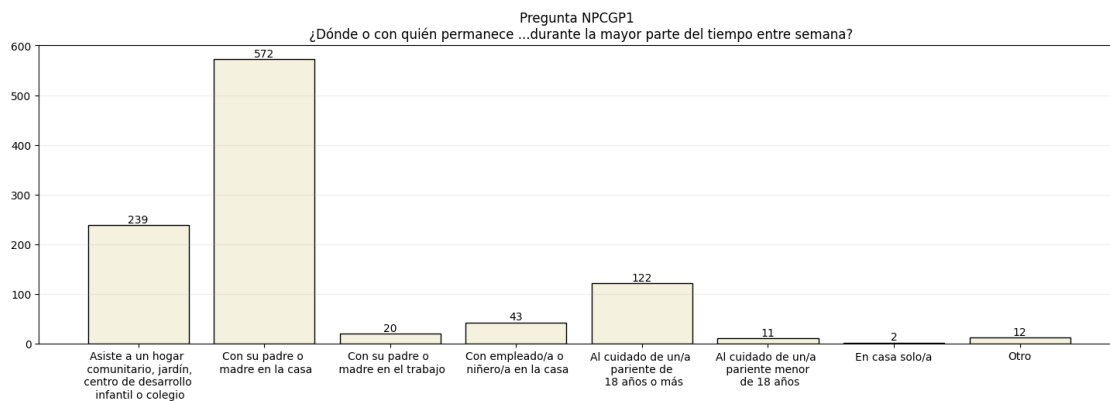
[8]: plt.figure(figsize=(18,5))
bars = plt.bar(data['NPCGP1'].value_counts().index.tolist(), data['NPCGP1'].
↳ value_counts().tolist(), edgecolor = 'k', color = '#f4f1de')
plt.xticks([1,2,3,4,5,6,7,8], ['Asiste a un hogar \n comunitario, jardín,\n
↳ centro de desarrollo \n infantil o colegio',
'Con su padre o \n madre en la casa',
'Con su padre o \n madre en el trabajo',
'Con empleado/a o \n niño/a en la casa',
'Al cuidado de un/a \n pariente de \n 18 años o más',
'Al cuidado de un/a \n pariente menor \n de 18 años',

```

```

'En casa solo/a',
'Otro'])
plt.title('Pregunta NPCGP1 \n ¿Dónde o con quién permanece ...durante la mayor_
↳ parte del tiempo entre semana?')
plt.bar_label(bars)
plt.grid(alpha = 0.2, axis = 'y')
plt.show()

```



2.2 NPCGP1A

2.2.1 1a. Asiste a un hogar comunitario, jardín, centro de desarrollo infantil o colegio ¿Cuántas horas al semana?

Datos: 239

```
[9]: data['NPCGP1A'].count()
```

[9]: 239

```
[10]: data['NPCGP1A'].describe().apply("{0:.2f}".format)
```

```

[10]: count      239.00
      mean       30.65
      std        8.25
      min       21.00
      25%       24.00
      50%       30.00
      75%       40.00
      max       55.00
      Name: NPCGP1A, dtype: object

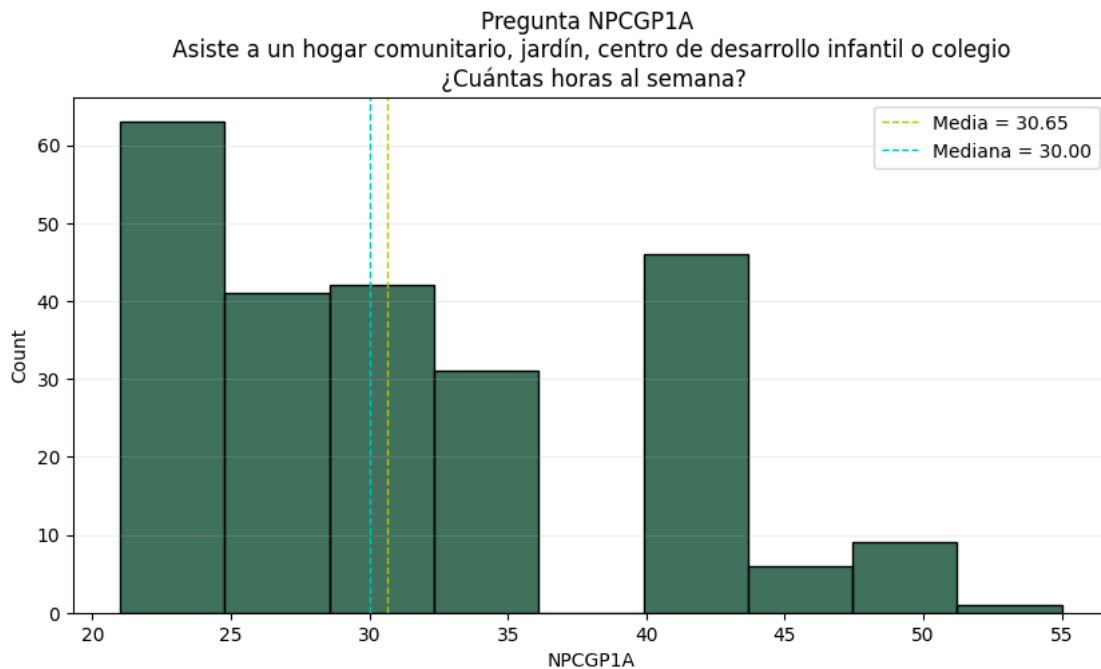
```

```

[11]: plt.figure(figsize=(10,5))
      sns.histplot(data = data, x = 'NPCGP1A', color = '#004225')

```

```
plt.title('Pregunta NPCGP1A \n Asiste a un hogar comunitario, jardín, centro de desarrollo infantil o colegio \n ¿Cuántas horas al semana?')
plt.grid(alpha = 0.2, axis = 'y')
plt.axvline(data['NPCGP1A'].mean(), color='y', linestyle='dashed', linewidth=1, label = f'Media = {data["NPCGP1A"].mean():.2f}')
plt.axvline(data['NPCGP1A'].median(), color='c', linestyle='dashed', linewidth=1, label = f'Mediana = {data["NPCGP1A"].median():.2f}')
#plt.xticks(range(0,3100000,250000),[f'{(i / 1000000):.2f} M' for i in range(0,3100000,250000)], fontsize=7)
plt.legend()
plt.show()
```



2.3 NPCGP2

2.3.1 2. ¿Cuál es la razón principal por la cual ...no asiste a un hogar comunitario, jardín, centro de desarrollo infantil o colegio?

1. No hay una institución cercana
2. Es muy costoso
3. No encontró cupo
4. Prefiere que no asista
5. Tiene un/a familiar en la casa que lo/a cuida
6. Considera que no está en edad de asistir
7. Solo asiste algunas horas o algunos días de la semana
8. Otra

Datos: 782

```
[12]: data['NPCGP2'].count()
```

```
[12]: 782
```

```
[13]: data['NPCGP2'].value_counts()
```

```
[13]: 6.0    251
      4.0    186
      5.0    146
      8.0     82
      3.0     39
      2.0     35
      7.0     30
      1.0     13
      Name: NPCGP2, dtype: int64
```

```
[13]: data['NPCGP2'] = data['NPCGP2'].replace([1,2,3,4,5,6,7,8],['No hay una_
      ↪institución cercana',
      'Es muy costoso',
      'No encontró cupo',
      'Prefiere que no asista',
      'Tiene un/a familiar en la casa que lo/a cuida',
      'Considera que no está en edad de asistir',
      'Solo asiste algunas horas o algunos días de la semana',
      'Otra'])
```

```
[14]: data['NPCGP2'].describe().apply("{0:.2f}".format)
```

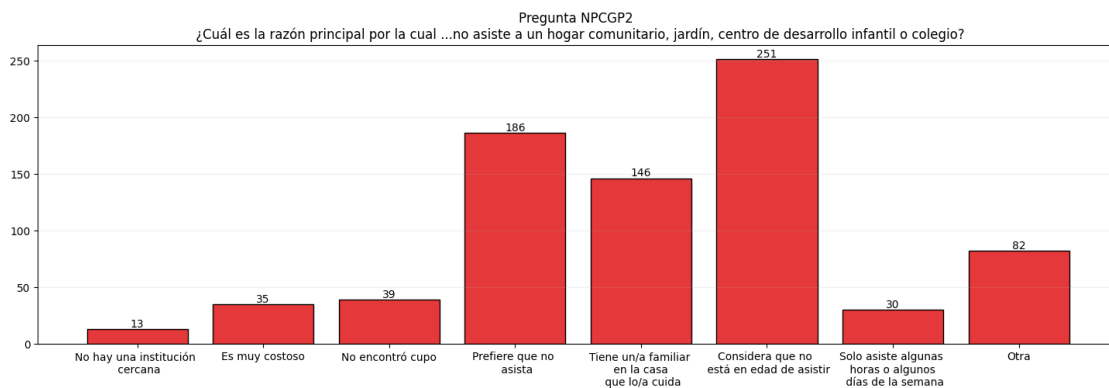
```
[14]: count      782.00
      mean        5.17
      std         1.58
      min         1.00
      25%         4.00
      50%         5.00
      75%         6.00
      max         8.00
      Name: NPCGP2, dtype: object
```

```
[14]: data['NPCGP2'].value_counts()
```

```
[14]: Considera que no está en edad de asistir      251
      Prefiere que no asista                        186
      Tiene un/a familiar en la casa que lo/a cuida  146
      Otra                                           82
      No encontró cupo                             39
      Es muy costoso                               35
```

Solo asiste algunas horas o algunos días de la semana 30
 No hay una institución cercana 13
 Name: NPCGP2, dtype: int64

```
[16]: plt.figure(figsize=(18,5))
bars = plt.bar(data['NPCGP2'].value_counts().index.tolist(),data['NPCGP2'].
    ↪value_counts().tolist(), edgecolor = 'k', color = '#e5383b')
plt.xticks([1,2,3,4,5,6,7,8], ['No hay una institución \n cercana',
    'Es muy costoso',
    'No encontró cupo',
    'Prefiere que no \n asista',
    'Tiene un/a familiar \n en la casa \n que lo/a cuida',
    'Considera que no \n está en edad de asistir',
    'Solo asiste algunas \n horas o algunos \n días de la semana',
    'Otra'])
plt.title('Pregunta NPCGP2 \n ¿Cuál es la razón principal por la cual ...no
    ↪asiste a un hogar comunitario, jardín, centro de desarrollo infantil o
    ↪colegio?')
plt.bar_label(bars)
plt.grid(alpha = 0.2, axis = 'y')
plt.show()
```



2.4 NPCGP2B

2.4.1 2b. Solo asiste algunas horas o algunos días de la semana ¿Cuántas horas a la semana?

Datos: 30

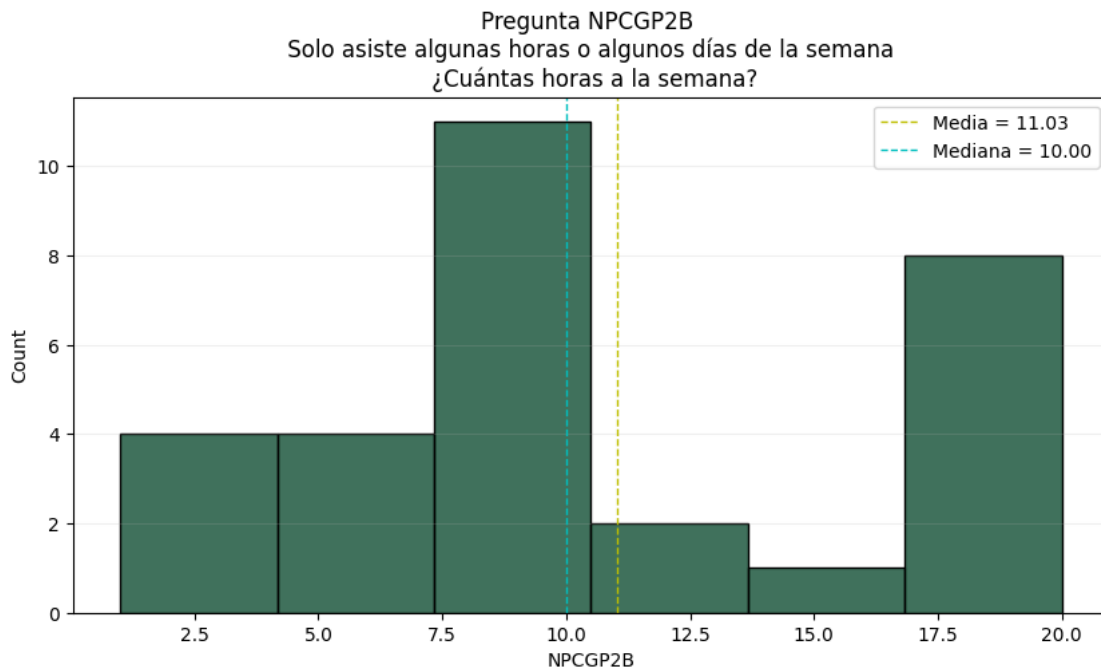
```
[17]: data['NPCGP2B'].count()
```

[17]: 30

```
[18]: data['NPCGP2B'].describe().apply("{0:.2f}".format)
```

```
[18]: count    30.00
      mean     11.03
      std      6.10
      min      1.00
      25%      6.50
      50%     10.00
      75%     16.50
      max     20.00
      Name: NPCGP2B, dtype: object
```

```
[19]: plt.figure(figsize=(10,5))
      sns.histplot(data = data, x = 'NPCGP2B', color = '#004225')
      plt.title('Pregunta NPCGP2B \n Solo asiste algunas horas o algunos días de la
      ↪semana \n ¿Cuántas horas a la semana?')
      plt.grid(alpha = 0.2, axis = 'y')
      plt.axvline(data['NPCGP2B'].mean(), color='y', linestyle='dashed', linewidth=1,
      ↪label = f'Media = {data["NPCGP2B"].mean():.2f}')
      plt.axvline(data['NPCGP2B'].median(), color='c', linestyle='dashed',
      ↪linewidth=1, label = f'Mediana = {data["NPCGP2B"].median():.2f}')
      #plt.xticks(range(0,310000,250000),[f'{(i / 100000):.2f} M' for i in
      ↪range(0,310000,250000)], fontsize=7)
      plt.legend()
      plt.show()
```



2.5 NPCGP3

2.5.1 3. ¿A qué tipo de establecimiento (hogar comunitario, jardín, centro de desarrollo infantil o colegio) asiste?

1. Hogar comunitario de Bienestar Familiar
2. Hogar infantil o jardín de Bienestar Familiar
3. Jardín infantil de la Secretaría Distrital de Integración Social (sólo para Bogotá)
4. Centro de Desarrollo Infantil Oficial (CDI)
5. Jardín o colegio oficial
6. Jardín o colegio privado

Datos: 239

```
[20]: data['NPCGP3'].count()
```

```
[20]: 239
```

```
[21]: data['NPCGP3'].value_counts()
```

```
[21]: 6.0    144
      3.0     25
      1.0     25
      2.0     23
      5.0     21
      4.0      1
      Name: NPCGP3, dtype: int64
```

```
[15]: data['NPCGP3'] = data['NPCGP3'].replace([1,2,3,4,5,6], ['Hogar comunitario de Bienestar Familiar',
'Hogar infantil o jardín de Bienestar Familiar',
'Jardín infantil de la Secretaría Distrital de Integración Social',
'Centro de Desarrollo Infantil Oficial (CDI)',
'Jardín o colegio oficial',
'Jardín o colegio privado'])
```

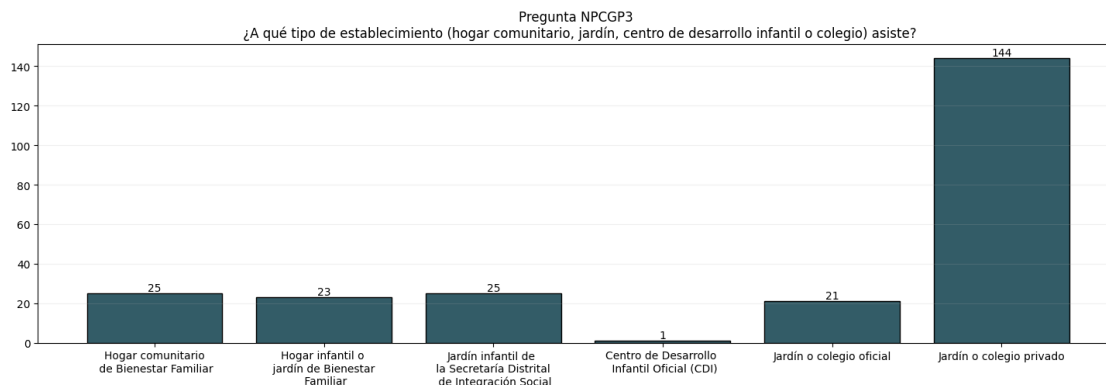
```
[22]: data['NPCGP3'].describe().apply("{0:.2f}".format)
```

```
[22]: count      239.00
      mean       4.68
      std       1.87
      min       1.00
      25%       3.00
      50%       6.00
      75%       6.00
      max       6.00
      Name: NPCGP3, dtype: object
```

```
[16]: data['NPCGP3'].value_counts()
```

```
[16]: Jardín o colegio privado 144
      Jardín infantil de la Secretaría Distrital de Integración Social 25
      Hogar comunitario de Bienestar Familiar 25
      Hogar infantil o jardín de Bienestar Familiar 23
      Jardín o colegio oficial 21
      Centro de Desarrollo Infantil Oficial (CDI) 1
      Name: NPCGP3, dtype: int64
```

```
[24]: plt.figure(figsize=(18,5))
      bars = plt.bar(data['NPCGP3'].value_counts().index.tolist(),data['NPCGP3'].
      ↪value_counts().tolist(), edgecolor = 'k', color = '#335c67')
      plt.xticks([1,2,3,4,5,6], ['Hogar comunitario\n de Bienestar Familiar',
      'Hogar infantil o \n jardín de Bienestar \n Familiar',
      'Jardín infantil de \n la Secretaría Distrital \n de Integración Social',
      'Centro de Desarrollo \n Infantil Oficial (CDI)',
      'Jardín o colegio oficial',
      'Jardín o colegio privado'])
      plt.title('Pregunta NPCGP3 \n ¿A qué tipo de establecimiento (hogar
      ↪comunitario, jardín, centro de desarrollo infantil o colegio) asiste?')
      plt.bar_label(bars)
      plt.grid(alpha = 0.2, axis = 'y')
      plt.show()
```



2.6 NPCGP4

2.6.1 4. ¿Quién lleva usualmente a ...al establecimiento que asiste?

1. El padre
2. La madre
3. Otra persona de 18 años y más
4. Otra persona menor de 18 años
5. Transporte escolar
6. Nadie, va solo(a)

Datos 239

```
[25]: data['NPCGP4'].count()
```

```
[25]: 239
```

```
[26]: data['NPCGP4'].value_counts()
```

```
[26]: 2.0    128
      1.0    50
      5.0    36
      3.0    24
      4.0     1
      Name: NPCGP4, dtype: int64
```

```
[17]: data['NPCGP4'] = data['NPCGP4'].replace([1,2,3,4,5,6], ['El padre',
'La madre',
'Otra persona de 18 años y más',
'Otra persona menor de 18 años',
'Transporte escolar','Naide, va solo(a)'])
```

```
[27]: data['NPCGP4'].describe().apply("{0:.2f}".format)
```

```
[27]: count    239.00
      mean     2.35
      std     1.25
      min     1.00
      25%     2.00
      50%     2.00
      75%     3.00
      max     5.00
      Name: NPCGP4, dtype: object
```

```
[18]: data['NPCGP4'].value_counts()
```

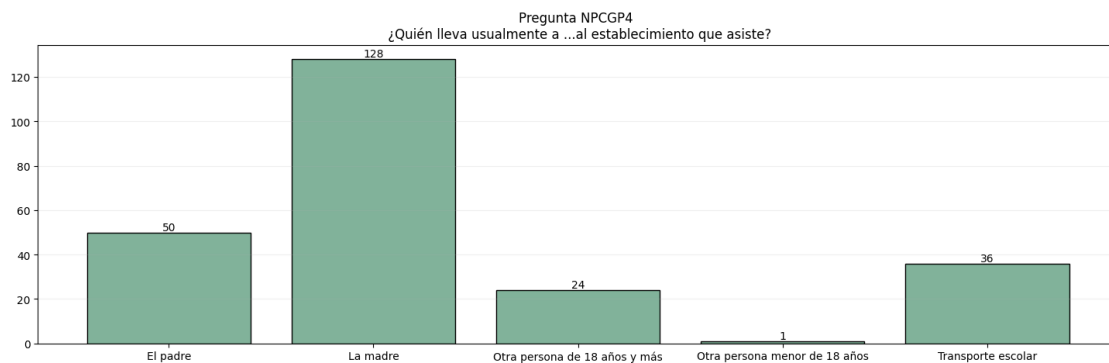
```
[18]: La madre          128
      El padre         50
      Transporte escolar  36
      Otra persona de 18 años y más  24
      Otra persona menor de 18 años  1
      Name: NPCGP4, dtype: int64
```

```
[29]: plt.figure(figsize=(18,5))
      bars = plt.bar(data['NPCGP4'].value_counts().index.tolist(),data['NPCGP4'].
      ↪value_counts().tolist(), edgecolor = 'k', color = '#81b29a')
      plt.xticks([1,2,3,4,5], ['El padre',
      'La madre',
      'Otra persona de 18 años y más',
```

```

'Otra persona menor de 18 años',
'Transporte escolar'])
plt.title('Pregunta NPCGP4 \n ¿Quién lleva usualmente a ...al establecimiento_
que asiste?')
plt.bar_label(bars)
plt.grid(alpha = 0.2, axis = 'y')
plt.show()

```



2.7 NPCGP5

2.7.1 5. ¿Quién recoge usualmente a ...en la institución a la que asiste?

1. El padre
2. La madre
3. Otra persona de 18 años y más
4. Otra persona menor de 18 años
5. Transporte escolar
6. Nadie, va solo/a

Datos: 239

```
[30]: data['NPCGP5'].count()
```

```
[30]: 239
```

```
[31]: data['NPCGP5'].value_counts()
```

```

[31]: 2.0    137
      3.0     35
      1.0     33
      5.0     32
      4.0      2
      Name: NPCGP5, dtype: int64

```

```
[19]: data['NPCGP5'] = data['NPCGP5'].replace([1,2,3,4,5,6], ['El padre',
'La madre',
'Otra persona de 18 años y más',
'Otra persona menor de 18 años',
'Transporte escolar','Naide, va solo(a)'])
```

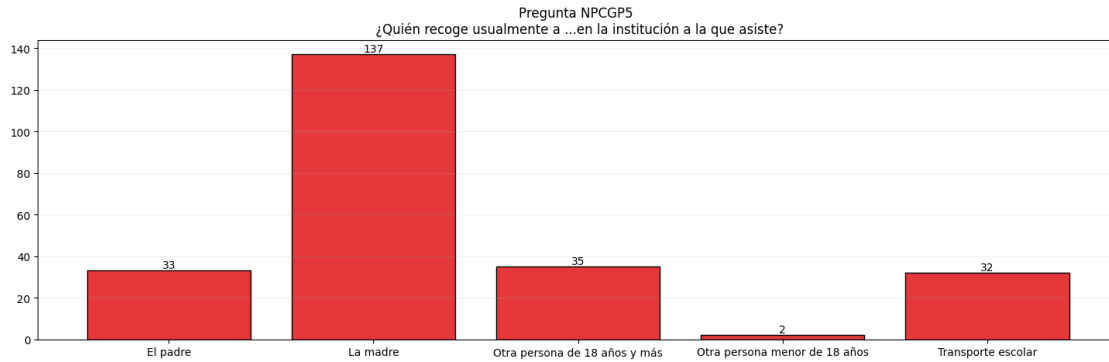
```
[32]: data['NPCGP5'].describe().apply("{0:.2f}".format)
```

```
[32]: count      239.00
mean         2.43
std          1.16
min          1.00
25%          2.00
50%          2.00
75%          3.00
max          5.00
Name: NPCGP5, dtype: object
```

```
[20]: data['NPCGP5'].value_counts()
```

```
[20]: La madre                137
Otra persona de 18 años y más    35
El padre                    33
Transporte escolar            32
Otra persona menor de 18 años     2
Name: NPCGP5, dtype: int64
```

```
[34]: plt.figure(figsize=(18,5))
bars = plt.bar(data['NPCGP5'].value_counts().index.tolist(),data['NPCGP5'].
↳value_counts().tolist(), edgecolor = 'k', color = '#e5383b')
plt.xticks([1,2,3,4,5], ['El padre',
'La madre',
'Otra persona de 18 años y más',
'Otra persona menor de 18 años',
'Transporte escolar'])
plt.title('Pregunta NPCGP5 \n ¿Quién recoge usualmente a ...en la institución a
↳la que asiste?')
plt.bar_label(bars)
plt.grid(alpha = 0.2, axis = 'y')
plt.show()
```



2.8 NPCGP6B

2.8.1 6b. ¿Cuánto tiempo se demora ... en su viaje de ida al establecimiento al que asiste? Minutos

Datos: 239

```
[35]: data['NPCGP6B'].count()
```

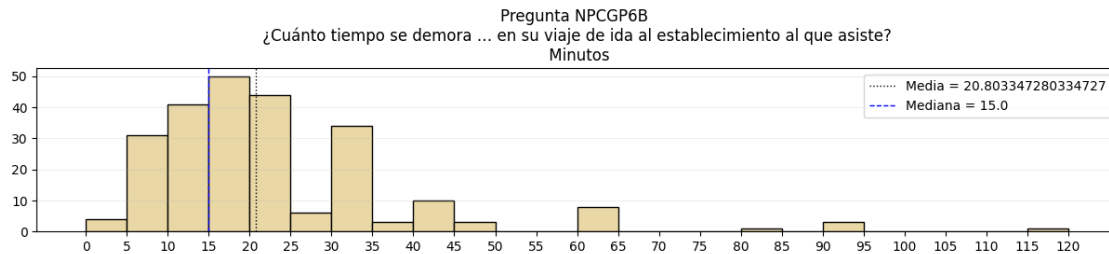
```
[35]: 239
```

```
[36]: data['NPCGP6B'].describe().apply("{0:.2f}".format)
```

```
[36]: count      239.00
      mean       20.80
      std        16.46
      min         0.00
      25%        10.00
      50%        15.00
      75%        30.00
      max       120.00
      Name: NPCGP6B, dtype: object
```

```
[37]: plt.figure(figsize=(15,5))
      plt.subplot(2,1,1)
      counts, edges, bars = plt.hist(data['NPCGP6B'] , bins = 24, edgecolor = 'black', color = '#e9d8a6')
      #ticklabels = [i for i in range(5)]
      plt.xticks(range(0,121,5))
      #plt.bar_label(bars)
      plt.title('Pregunta NPCGP6B \n ¿Cuánto tiempo se demora ... en su viaje de ida al establecimiento al que asiste? \n Minutos')
      #plt.xlim([-1,10])
      plt.axvline(data['NPCGP6B'].mean(), color='k', linestyle=':', linewidth=1, label = f'Media = {data["NPCGP6B"].mean()}')
```

```
plt.axvline(data['NPCGP6B'].median(), color='b', linestyle='dashed', linewidth=1, label = f'Mediana = {data["NPCGP6B"].median()}')
plt.legend()
plt.grid(alpha = 0.2, axis = 'y')
plt.show()
```



2.9 NPCGP14

2.9.1 13. ¿Recibe o toma ...desayuno o almuerzo en el lugar donde permanece la mayor parte del tiempo ENTRE SEMANA?

1. Sí
2. No

Datos: 1021

```
[38]: data['NPCGP14'].count()
```

```
[38]: 1021
```

```
[22]: data['NPCGP14'].value_counts()
```

```
[22]: Si      940
      No      81
      Name: NPCGP14, dtype: int64
```

```
[21]: data['NPCGP14'] = data['NPCGP14'].replace([0, 1], ['No', 'Si'])
```

```
[40]: data['NPCGP14'].describe()
```

```
[40]: count      1021.000000
      mean        1.079334
      std         0.270392
      min         1.000000
      25%         1.000000
      50%         1.000000
      75%         1.000000
      max         2.000000
```

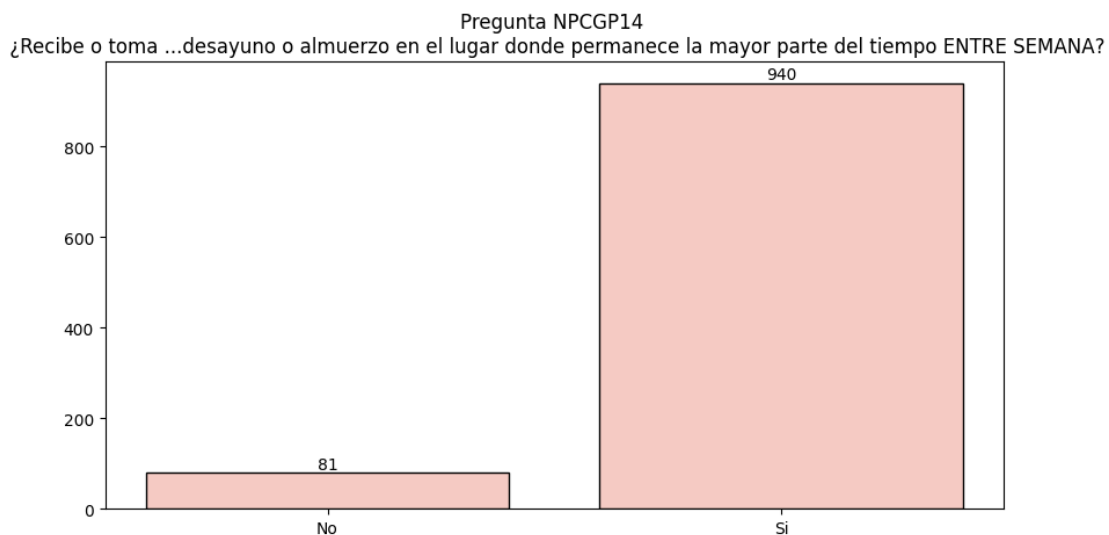
Name: NPCGP14, dtype: float64

```
[41]: data = data.replace({'NPCGP14':2},0)
```

```
[42]: data['NPCGP14'].value_counts()
```

```
[42]: 1.0    940  
      0.0     81  
      Name: NPCGP14, dtype: int64
```

```
[43]: plt.figure(figsize=(10,5))  
      bars = plt.bar(data['NPCGP14'].value_counts().index.tolist(),data['NPCGP14'].  
      ↪value_counts().tolist(), edgecolor = 'black', color = '#f5cac3')  
      plt.xticks([0,1],['No','Si'])  
      plt.title('Pregunta NPCGP14 \n ¿Recibe o toma ...desayuno o almuerzo en el_  
      ↪lugar donde permanece la mayor parte del tiempo ENTRE SEMANA?')  
      plt.bar_label(bars)  
      plt.show()
```



2.10 NPCGP15

2.10.1 14. ¿Este hogar paga por el desayuno o almuerzo que ...recibe?

1. Sí, completamente
2. Sí, un pago simbólico
3. No paga
4. No paga, lo recibe o lo lleva del hogar

Datos: 940


```
[44]: data['NPCGP15'].count()
```

```
[44]: 940
```

```
[45]: data['NPCGP15'].value_counts()
```

```
[45]: 1.0    440
      4.0    380
      3.0    116
      2.0     4
      Name: NPCGP15, dtype: int64
```

```
[23]: data['NPCGP15'] = data['NPCGP15'].replace([1, 2, 3, 4], ['Sí, completamente',
'Sí, un pago simbólico',
'No paga',
'No paga, lo recibe o lo lleva del hogar'])
```

```
[46]: data['NPCGP15'].describe().apply("{0:.2f}".format)
```

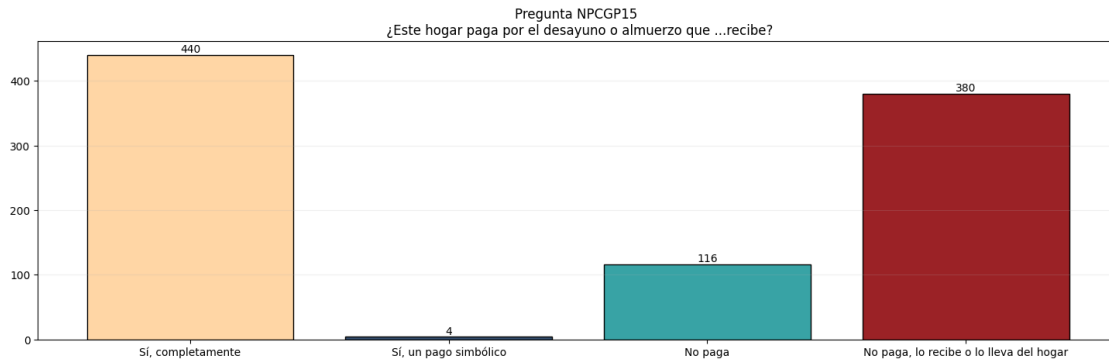
```
[46]: count      940.00
      mean        2.46
      std         1.41
      min         1.00
      25%         1.00
      50%         3.00
      75%         4.00
      max         4.00
      Name: NPCGP15, dtype: object
```

```
[24]: data['NPCGP15'].value_counts()
```

```
[24]: Sí, completamente      440
      No paga, lo recibe o lo lleva del hogar    380
      No paga      116
      Sí, un pago simbólico      4
      Name: NPCGP15, dtype: int64
```

```
[48]: plt.figure(figsize=(18,5))
      bars = plt.bar(data['NPCGP15'].value_counts().index.tolist(),data['NPCGP15'].
      ↪value_counts().tolist(), edgecolor = 'k', color =_
      ↪['#ffd6a5', '#9b2226', '#38a3a5', '#355070'])
      plt.xticks([1,2,3,4], ['Sí, completamente',
'Sí, un pago simbólico',
'No paga',
'No paga, lo recibe o lo lleva del hogar'])
      plt.title('Pregunta NPCGP15 \n ¿Este hogar paga por el desayuno o almuerzo que .
      ↪..recibe?')
      plt.bar_label(bars)
```

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



2.11 NPCGP15A

2.11.1 14a. Sí, un pago simbólico. Valor MENSUAL

Datos: 4

```
[49]: data['NPCGP15A'].count()
```

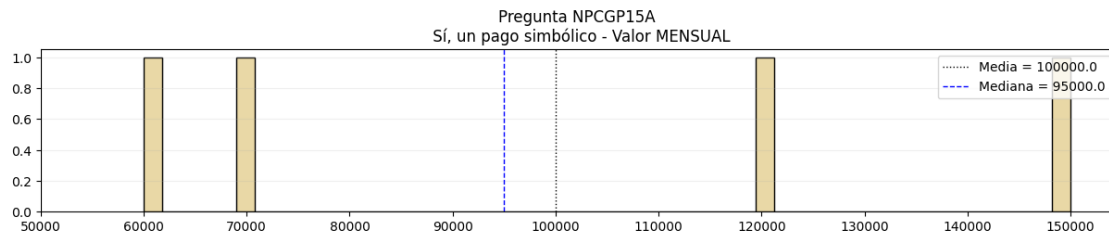
```
[49]: 4
```

```
[50]: data['NPCGP15A'].describe().apply("{0:.2f}".format)
```

```
[50]: count          4.00
      mean       100000.00
      std        42426.41
      min        60000.00
      25%        67500.00
      50%        95000.00
      75%       127500.00
      max       150000.00
      Name: NPCGP15A, dtype: object
```

```
[51]: plt.figure(figsize=(15,5))
      plt.subplot(2,1,1)
      counts, edges, bars = plt.hist(data['NPCGP15A'] , bins = 50, edgecolor = 'black', color = '#e9d8a6')
      #ticklabels = [i for i in range(5)]
      plt.xticks(range(50000,160000,10000))
      #plt.bar_label(bars)
      plt.title('Pregunta NPCGP15A \n Sí, un pago simbólico - Valor MENSUAL')
      #plt.xlim([-1,10])
```

```
plt.axvline(data['NPCGP15A'].mean(), color='k', linestyle=':', linewidth=1,
↳label = f'Media = {data["NPCGP15A"].mean()}')
plt.axvline(data['NPCGP15A'].median(), color='b', linestyle='dashed',
↳linewidth=1, label = f'Mediana = {data["NPCGP15A"].median()}')
plt.legend()
plt.grid(alpha = 0.2, axis = 'y')
plt.show()
```



2.12 NPCGP15B

2.12.1 14b. ¿Si tuviera que comprar la alimentación en otra parte, cuanto pagaría al MES por lo que recibe? Valor MENSUAL estimado

Datos: 120

```
[52]: data['NPCGP15B'].count()
```

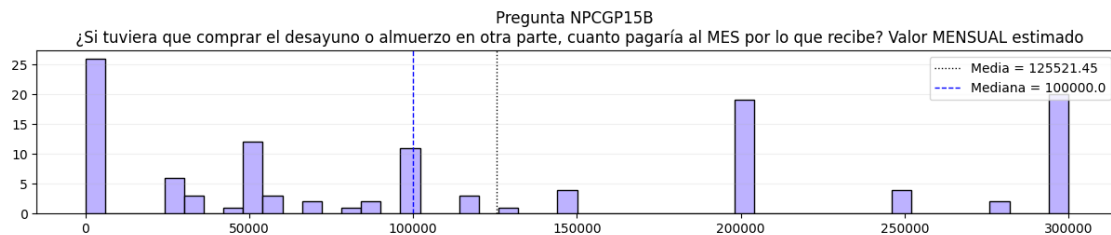
```
[52]: 120
```

```
[53]: data['NPCGP15B'].describe().apply("{0:.2f}".format)
```

```
[53]: count      120.00
      mean      125521.45
      std       109257.39
      min         99.00
      25%       30000.00
      50%      100000.00
      75%      200000.00
      max      300000.00
      Name: NPCGP15B, dtype: object
```

```
[54]: plt.figure(figsize=(15,5))
      plt.subplot(2,1,1)
      counts, edges, bars = plt.hist(data['NPCGP15B'] , bins = 50, edgecolor =
↳'black', color = '#bdb2ff')
      #ticklabels = [i for i in range(5)]
      #plt.xticks(range(50000,160000,10000))
      #plt.bar_label(bars)
```

```
plt.title('Pregunta NPCGP15B \n ¿Si tuviera que comprar el desayuno o almuerzo_
↳en otra parte, cuanto pagaría al MES por lo que recibe? Valor MENSUAL_
↳estimado')
#plt.xlim([-1,10])
plt.axvline(data['NPCGP15B'].mean(), color='k', linestyle=':', linewidth=1,
↳label = f'Media = {data["NPCGP15B"].mean()}')
plt.axvline(data['NPCGP15B'].median(), color='b', linestyle='dashed',
↳linewidth=1, label = f'Mediana = {data["NPCGP15B"].median()}')
plt.legend()
plt.grid(alpha = 0.2, axis = 'y')
plt.show()
```



2.13 NPCGP16

2.13.1 15. ¿Recibe o toma... refrigerio en el lugar donde permanece la mayor parte del tiempo ENTRE SEMANA?

1. Sí
2. No

Datos: 1021

```
[27]: data['NPCGP16'].count()
```

```
[27]: 1021
```

```
[26]: data['NPCGP16'].value_counts()
```

```
[26]: Si      852
      No     169
      Name: NPCGP16, dtype: int64
```

```
[25]: data['NPCGP16'] = data['NPCGP16'].replace([0,1], ['No', 'Si'])
```

```
[57]: data['NPCGP16'].describe()
```

```
[57]: count      1021.000000
      mean        1.165524
      std         0.371835
```

```

min          1.000000
25%          1.000000
50%          1.000000
75%          1.000000
max          2.000000
Name: NPCGP16, dtype: float64

```

```
[58]: data = data.replace({'NPCGP16':2},0)
```

```
[59]: data['NPCGP16'].value_counts()
```

```

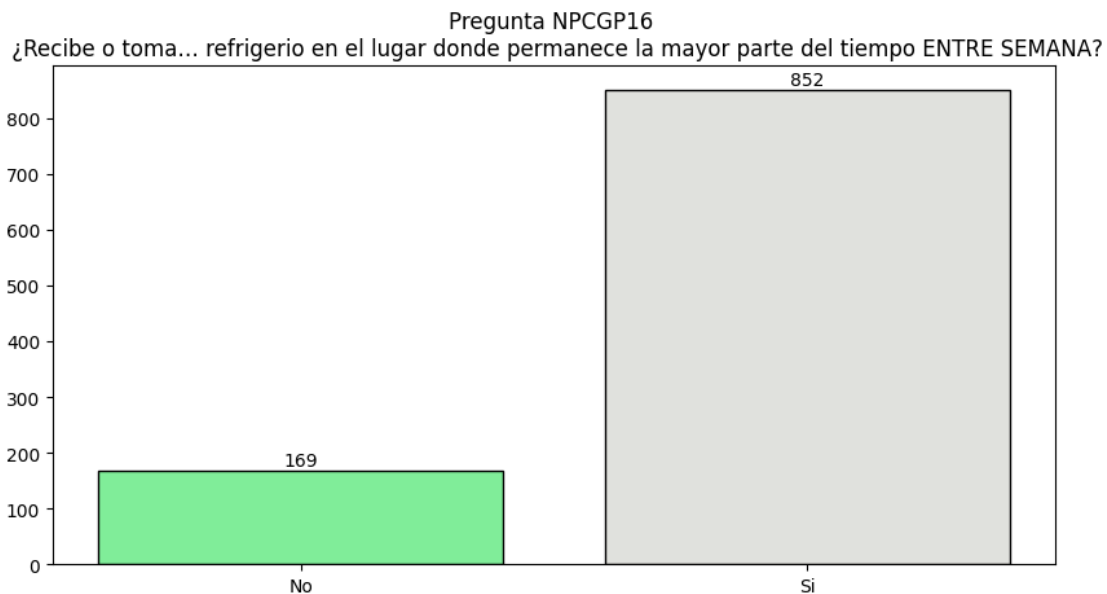
[59]: 1.0      852
      0.0      169
      Name: NPCGP16, dtype: int64

```

```

[60]: plt.figure(figsize=(10,5))
      bars = plt.bar(data['NPCGP16'].value_counts().index.tolist(),data['NPCGP16'].
      ↪value_counts().tolist(), edgecolor = 'black', color_
      ↪=['#e0e1dd','#80ed99','#9b2226'])
      plt.xticks([0,1],['No','Si'])
      plt.title('Pregunta NPCGP16 \n ¿Recibe o toma... refrigerio en el lugar donde_
      ↪permanece la mayor parte del tiempo ENTRE SEMANA?')
      plt.bar_label(bars)
      plt.show()

```



2.14 NPCGP17

2.14.1 16. ¿Este hogar paga por refrigerio que ...recibe?

1. Sí, completamente
2. Sí, un pago simbólico
3. No paga
4. No paga, lo recibe o lo lleva del hogar

Datos: 852

```
[61]: data['NPCGP17'].count()
```

```
[61]: 852
```

```
[62]: data['NPCGP17'].value_counts()
```

```
[62]: 1.0    404
      4.0    337
      3.0    108
      2.0     3
      Name: NPCGP17, dtype: int64
```

```
[28]: data['NPCGP17'] = data['NPCGP17'].replace([1,2,3,4], ['Sí, completamente',
'Sí, un pago simbólico',
'No paga',
'No paga, lo recibe o lo lleva del hogar'])
```

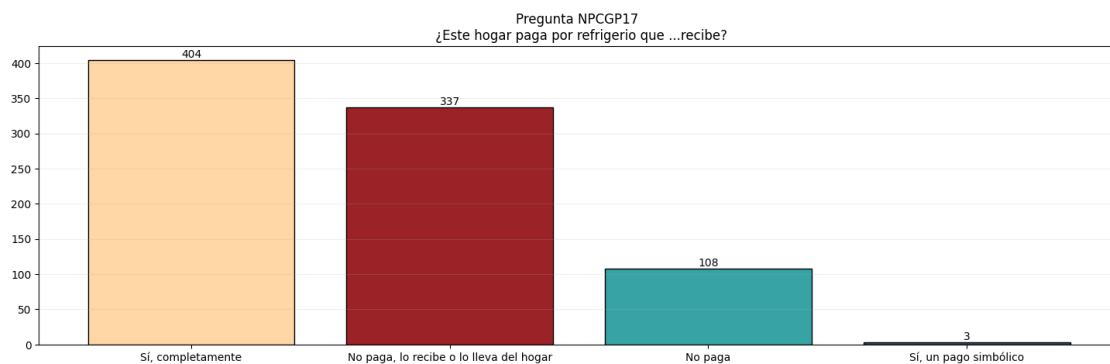
```
[63]: data['NPCGP17'].describe().apply("{0:.2f}".format)
```

```
[63]: count      852.00
      mean        2.44
      std         1.41
      min         1.00
      25%         1.00
      50%         3.00
      75%         4.00
      max         4.00
      Name: NPCGP17, dtype: object
```

```
[29]: data['NPCGP17'].value_counts()
```

```
[29]: Sí, completamente      404
      No paga, lo recibe o lo lleva del hogar  337
      No paga      108
      Sí, un pago simbólico      3
      Name: NPCGP17, dtype: int64
```

```
[31]: plt.figure(figsize=(18,5))
bars = plt.bar(data['NPCGP17'].value_counts().index.tolist(),data['NPCGP17'].
↳value_counts().tolist(), edgecolor = 'k', color =
↳['#ffd6a5','#9b2226','#38a3a5','#355070'])
#plt.xticks([1,2,3,4], ['Sí, completamente',
# 'Sí, un pago simbólico',
# 'No paga',
# 'No paga, lo recibe o lo lleva del hogar'])
plt.title('Pregunta NPCGP17 \n ¿Este hogar paga por refrigerio que ...recibe?')
plt.bar_label(bars)
plt.grid(alpha = 0.2, axis = 'y')
plt.show()
```



2.15 NPCGP17A

2.15.1 16a. Sí, un pago simbólico. Valor MENSUAL

Datos: 3

```
[66]: data['NPCGP17A'].count()
```

```
[66]: 3
```

```
[67]: data['NPCGP17A'].describe().apply("{0:.2f}".format)
```

```
[67]: count          3.00
mean        26699.67
std         25113.67
min           99.00
25%        15049.50
50%        30000.00
75%        40000.00
max         50000.00
Name: NPCGP17A, dtype: object
```

2.16 NPCGP17B

2.16.1 16b. ¿Si tuviera que comprar el refrigerio en otra parte, cuanto pagaría al MES por lo que recibe?

Datos: 111

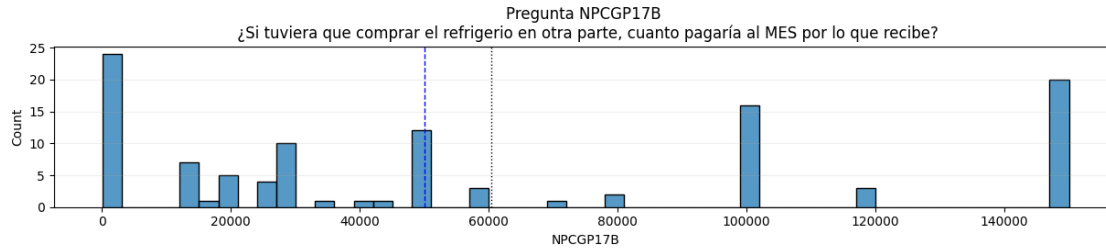
```
[68]: data['NPCGP17B'].count()
```

```
[68]: 111
```

```
[69]: data['NPCGP17B'].describe().apply("{0:.2f}".format)
```

```
[69]: count      111.00
      mean      60480.86
      std      54426.80
      min       99.00
      25%     15000.00
      50%     50000.00
      75%    100000.00
      max     150000.00
      Name: NPCGP17B, dtype: object
```

```
[70]: plt.figure(figsize=(15,5))
      plt.subplot(2,1,1)
      #counts, edges, bars = plt.hist(data['NPCGP17B'] , bins = 50, edgecolor = 'black', color = '#bdb2ff')
      sns.histplot(data = data, x = 'NPCGP17B', bins = 50)
      #sns.kdeplot(data = data, x = 'NPCGP17B')
      #ticklabels = [i for i in range(5)]
      #plt.xticks(range(50000,160000,10000))
      #plt.bar_label(bars)
      plt.title('Pregunta NPCGP17B \n ¿Si tuviera que comprar el refrigerio en otra parte, cuanto pagaría al MES por lo que recibe?')
      #plt.xlim([-1,10])
      plt.axvline(data['NPCGP17B'].mean(), color='k', linestyle=':', linewidth=1, label = f'Media = {data["NPCGP17B"].mean()}')
      plt.axvline(data['NPCGP17B'].median(), color='b', linestyle='dashed', linewidth=1, label = f'Mediana = {data["NPCGP17B"].median()}')
      #plt.legend()
      plt.grid(alpha = 0.2, axis = 'y')
      plt.show()
```

2.17 NPCGP18

2.17.1 17. ¿Quién se encarga principalmente del cuidado de ...después de asistir al hogar comunitario, jardín, centro de desarrollo infantil o en el sitio en el que permanece la mayor parte del tiempo ENTRE SEMANA?

Datos: 1021

```
[71]: data['NPCGP18'].count()
```

```
[71]: 1021
```

```
[33]: data['NPCGP18'].value_counts()
```

```
[33]: Una persona de este hogar      910
      Una mujer de otro hogar      103
      Un hombre de otro hogar       5
      Nadie le cuida                3
      Name: NPCGP18, dtype: int64
```

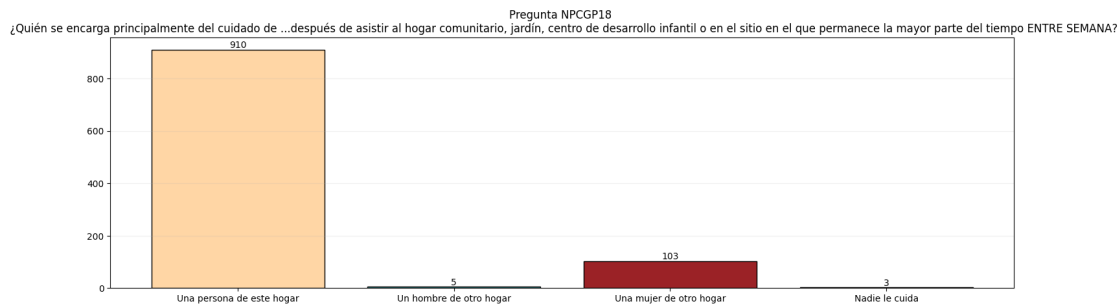
```
[32]: data['NPCGP18'] = data['NPCGP18'].replace([1,2,3,4], ['Una persona de este_
      ↪hogar', 'Un hombre de otro hogar', 'Una mujer de otro hogar', 'Nadie le_
      ↪cuida'])
```

```
[73]: data['NPCGP18'].describe().apply("{0:.2f}".format)
```

```
[73]: count      1021.00
      mean        1.22
      std         0.62
      min         1.00
      25%         1.00
      50%         1.00
      75%         1.00
      max         4.00
      Name: NPCGP18, dtype: object
```

```
[74]:
```

```
plt.figure(figsize=(18,5))
bars = plt.bar(data['NPCGP18'].value_counts().index.tolist(),data['NPCGP18'].
    ↳value_counts().tolist(), edgecolor = 'k', color =
    ↳['#ffd6a5','#9b2226','#38a3a5','#355070'])
plt.xticks([1,2,3,4], ['Una persona de este hogar', 'Un hombre de otro hogar',
    ↳'Una mujer de otro hogar', 'Nadie le cuida'])
plt.title('Pregunta NPCGP18 \n ¿Quién se encarga principalmente del cuidado de .
    ↳..después de asistir al hogar comunitario, jardín, centro de desarrollo
    ↳infantil o en el sitio en el que permanece la mayor parte del tiempo ENTRE
    ↳SEMANA?')
plt.bar_label(bars)
plt.grid(alpha = 0.2, axis = 'y')
plt.show()
```



2.18 NPCGP19

2.18.1 19. ¿Cuántos años tiene esta persona?

Datos: 108

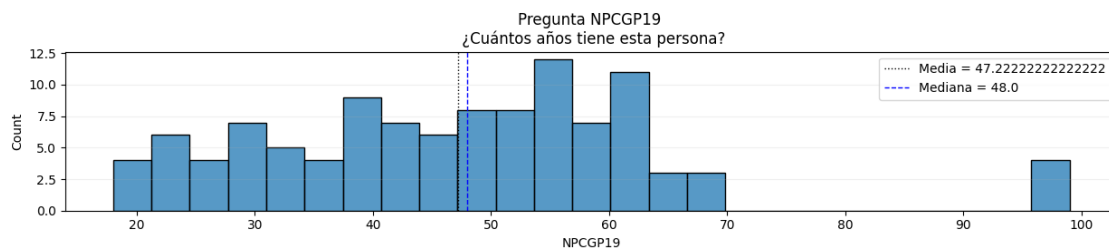
```
[75]: data['NPCGP19'].count()
```

```
[75]: 108
```

```
[76]: data['NPCGP19'].describe().apply("{0:.2f}".format)
```

```
[76]: count      108.00
      mean       47.22
      std        16.75
      min        18.00
      25%        35.00
      50%        48.00
      75%        57.00
      max        99.00
      Name: NPCGP19, dtype: object
```

```
[77]: plt.figure(figsize=(15,5))
plt.subplot(2,1,1)
#counts, edges, bars = plt.hist(data['NPCGP17B'], bins = 50, edgecolor = 'black', color = '#bdb2ff')
sns.histplot(data = data, x = 'NPCGP19', bins = 25)
#sns.kdeplot(data = data, x = 'NPCGP17B')
#ticklabels = [i for i in range(5)]
#plt.xticks(range(50000,160000,10000))
#plt.bar_label(bars)
plt.title('Pregunta NPCGP19 \n ¿Cuántos años tiene esta persona?')
#plt.xlim([-1,10])
plt.axvline(data['NPCGP19'].mean(), color='k', linestyle=':', linewidth=1, label = f'Media = {data["NPCGP19"].mean()}')
plt.axvline(data['NPCGP19'].median(), color='b', linestyle='dashed', linewidth=1, label = f'Mediana = {data["NPCGP19"].median()}')
plt.legend()
plt.grid(alpha = 0.2, axis = 'y')
plt.show()
```



2.19 NPCGP20

2.19.1 19. ¿Cuál es el nivel educativo de esta persona?

1. Algunos años de primaria
2. Toda la primaria
3. Algunos años de secundaria
4. Toda la secundaria
5. Uno o más años de técnica o tecnológica
6. Técnica o tecnológica completa (con o sin título)
7. Algunos años de universidad
8. Universitaria completa (con título)
9. Posgrado
10. Ninguno
11. No sabe

Datos: 108

```
[78]: data['NPCGP20'].count()
```

[78]: 108

```
[35]: data['NPCGP20'].value_counts()
```

```
[35]: Toda la secundaria          39
      Algunos años de secundaria  12
      Técnica o tecnológica completa (con o sin título)  12
      Universitaria completa (con título)  11
      Toda la primaria           8
      99.0                       7
      Algunos años de primaria   6
      Uno o más años de técnica o tecnológica  6
      Algunos años de universidad  3
      Posgrado                   3
      Ninguno                    1
      Name: NPCGP20, dtype: int64
```

```
[34]: data['NPCGP20'] = data['NPCGP20'].replace([1,2,3,4,5,6,7,8,9,10,11],['Algunos_
      ↪ años de primaria',
      'Toda la primaria',
      'Algunos años de secundaria',
      'Toda la secundaria',
      'Uno o más años de técnica o tecnológica',
      'Técnica o tecnológica completa (con o sin título)',
      'Algunos años de universidad',
      'Universitaria completa (con título)',
      'Posgrado',
      'Ninguno',
      'No sabe'])
```

```
[80]: data['NPCGP20'].describe().apply("{0:.2f}".format)
```

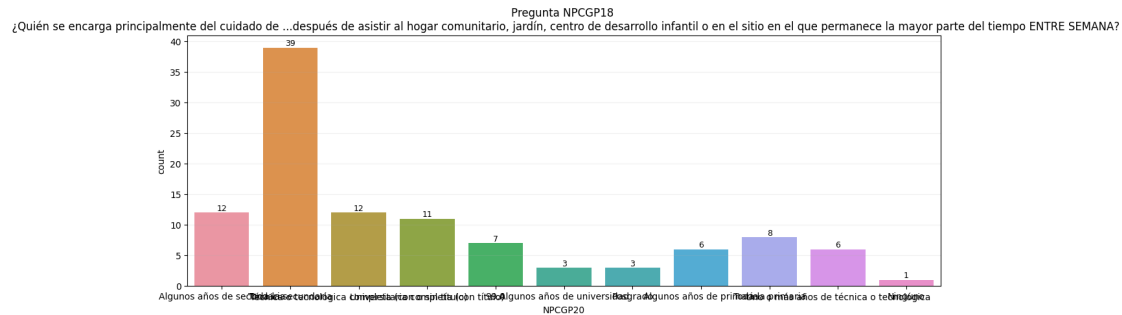
```
[80]: count    108.00
      mean     10.69
      std      23.44
      min       1.00
      25%       4.00
      50%       4.00
      75%       6.00
      max      99.00
      Name: NPCGP20, dtype: object
```

```
[36]: fig, ax = plt.subplots(figsize=(15, 5))
      g = sns.countplot(ax=ax, data = data, x = 'NPCGP20')

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

```
ax.set_title('Pregunta NPCGP18 \n ¿Quién se encarga principalmente del cuidado_
↳de ...después de asistir al hogar comunitario, jardín, centro de desarrollo_
↳infantil o en el sitio en el que permanece la mayor parte del tiempo ENTRE_
↳SEMANA?')
#ax.set_xticklabels(['No', 'Si'])

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



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

```
[ ]:
```