

TAREA 1 LAB



September 26, 2022

```
[1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
import sklearn
import scipy

%matplotlib inline
```

0.1 1 Lectura de datos

```
[2]: datos_junaeb = pd.read_csv('../datos tarea 1/junaeb.csv')
datos_junaeb.dropna(inplace=True)
```

```
[3]: datos_junaeb.reset_index(drop=True, inplace=True)
datos_junaeb
```

```
[3]:
```

	vive_padre	vive_madre	n_personas	n_habitaciones	cercania_juegos	\
0	0	1	3.0	4.0	1.0	
1	0	1	5.0	3.0	1.0	
2	1	1	5.0	3.0	1.0	
3	1	1	4.0	2.0	1.0	
4	1	1	5.0	3.0	2.0	
...	
6374	1	1	4.0	2.0	1.0	
6375	1	1	4.0	2.0	1.0	
6376	0	1	3.0	2.0	1.0	
6377	1	1	4.0	2.0	1.0	
6378	0	1	5.0	3.0	1.0	

	cercania_servicios	edad_primer_parto	area	educm	educp
0	1.0	25.0	1	0	0
1	1.0	23.0	1	13	13
2	1.0	19.0	1	12	17
3	1.0	27.0	1	6	13
4	1.0	20.0	1	13	16
...

6374	1.0	24.0	1	15	13
6375	1.0	22.0	1	15	15
6376	1.0	40.0	1	15	0
6377	1.0	29.0	1	13	13
6378	1.0	30.0	1	15	20

[6379 rows x 10 columns]

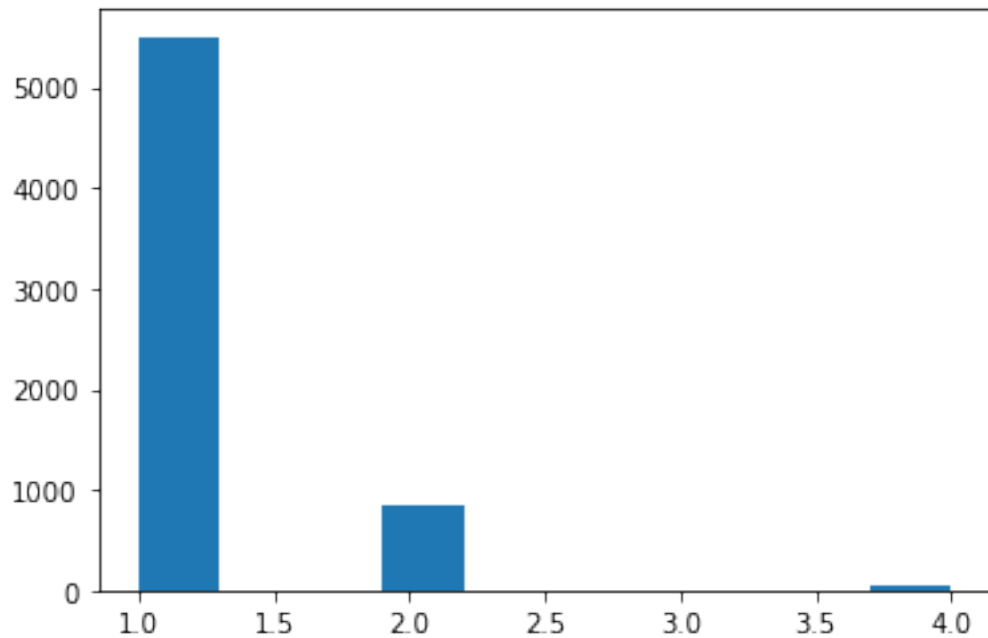
0.2 2 Limpieza de datos

Verificamos si existen outliers en cada variable que esté restringida a ciertos valores determinados

```
[4]: plt.hist(datos_junaeb['cercania_servicios'])
      datos_junaeb.cercania_servicios.value_counts()
```



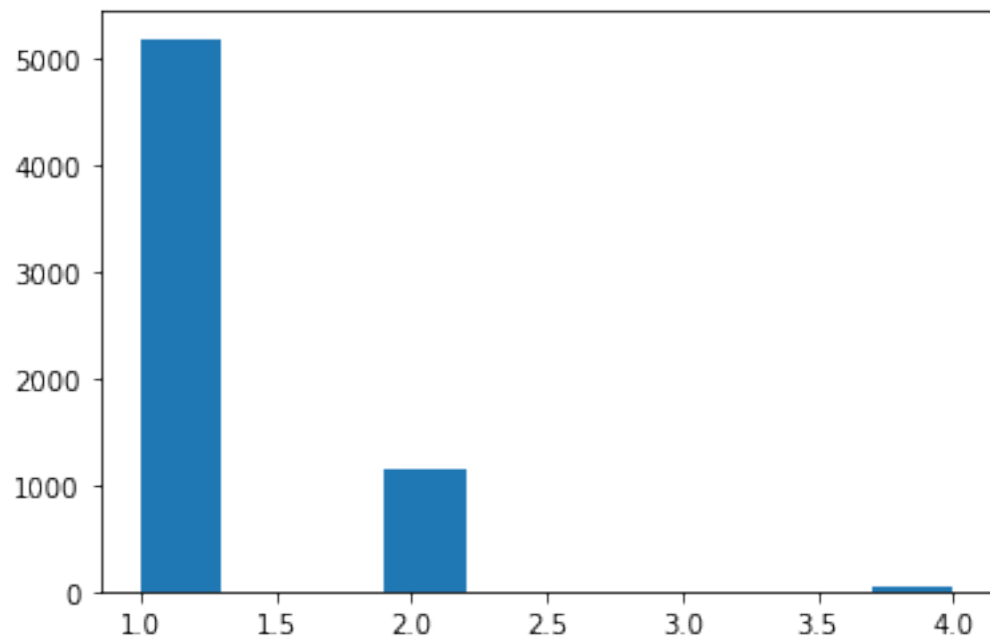
```
[4]: 1.0    5503
      2.0     834
      4.0      42
      Name: cercania_servicios, dtype: int64
```



```
[5]: plt.hist(datos_junaeb['cercania_juegos'])
      datos_junaeb.cercania_juegos.value_counts()
```

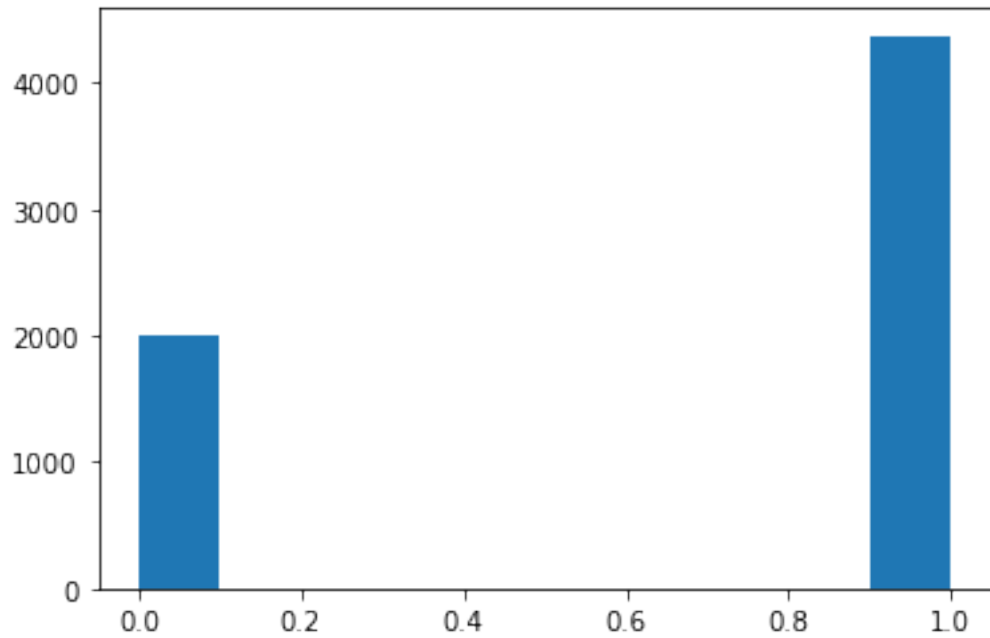
```
[5]: 1.0    5184
      2.0   1154
      4.0     41
```

Name: cercania_juegos, dtype: int64



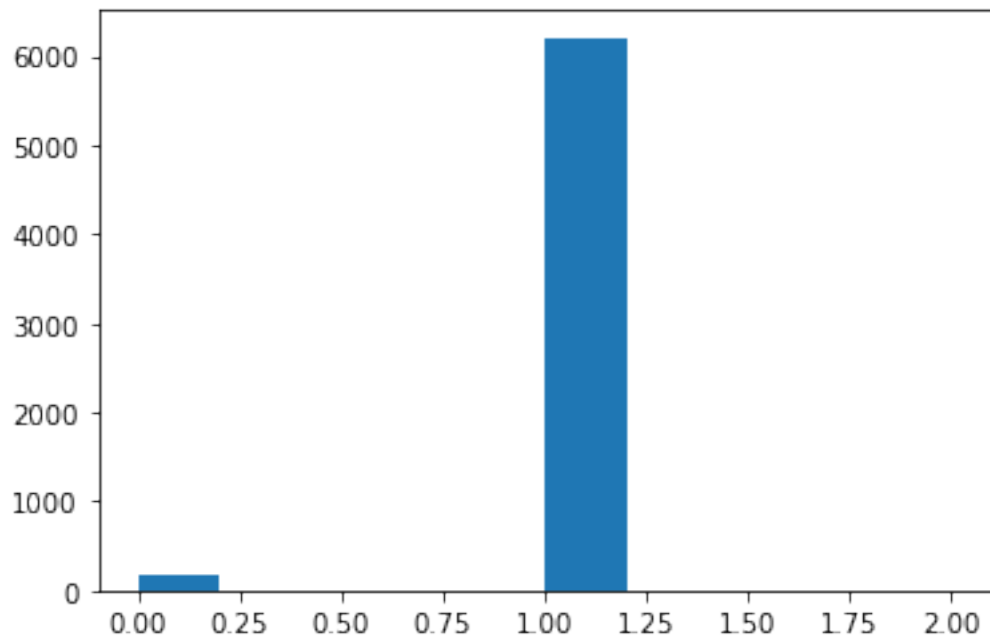
```
[6]: plt.hist(datos_junaeb['vive_padre'])  
datos_junaeb.vive_padre.value_counts()
```

```
[6]: 1    4373  
     0    2006  
     Name: vive_padre, dtype: int64
```



```
[7]: plt.hist(datos_junaeb['vive_madre'])  
datos_junaeb.vive_madre.value_counts()
```

```
[7]: 1    6208  
     0    166  
     2     5  
     Name: vive_madre, dtype: int64
```



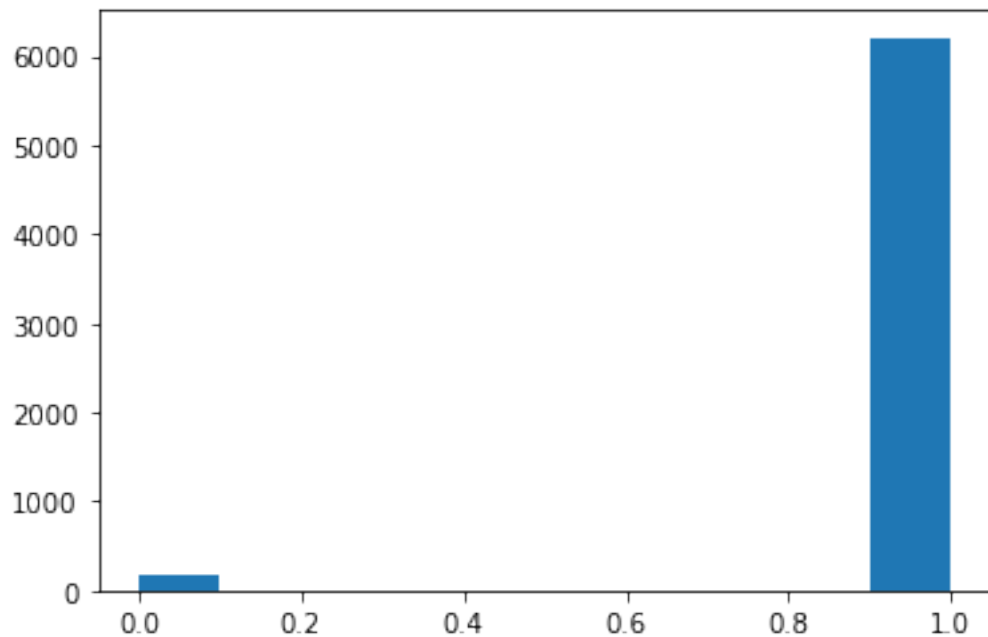
En este ultimo histograma, nos damos cuenta que existen valores fuera de rango(vive_madre=2) para la variable vive_madre, ya que solo puede tomar valores 0 o 1 y se procede a eliminar dichos datos erroneos

```
[8]: #Eliminar datos fuera de rango para la variable vive_madre

datos_junaeb.drop(datos_junaeb[datos_junaeb['vive_madre']==2].index,inplace_
    ↪=True)

plt.hist(datos_junaeb['vive_madre'])
datos_junaeb.vive_madre.value_counts()
```

```
[8]: 1    6208
     0     166
     Name: vive_madre, dtype: int64
```



```
[9]: datos_junaeb
```

```
[9]:
```

	vive_padre	vive_madre	n_personas	n_habitaciones	cercania_juegos	\
0	0	1	3.0	4.0	1.0	
1	0	1	5.0	3.0	1.0	
2	1	1	5.0	3.0	1.0	
3	1	1	4.0	2.0	1.0	
4	1	1	5.0	3.0	2.0	

```

...
6374      1      1      4.0      2.0      1.0
6375      1      1      4.0      2.0      1.0
6376      0      1      3.0      2.0      1.0
6377      1      1      4.0      2.0      1.0
6378      0      1      5.0      3.0      1.0

```

```

      cercania_servicios  edad_primer_parto  area  educm  educp
0      1.0      25.0      1      0      0
1      1.0      23.0      1     13     13
2      1.0      19.0      1     12     17
3      1.0      27.0      1      6     13
4      1.0      20.0      1     13     16
...
6374      1.0      24.0      1     15     13
6375      1.0      22.0      1     15     15
6376      1.0      40.0      1     15      0
6377      1.0      29.0      1     13     13
6378      1.0      30.0      1     15     20

```

[6374 rows x 10 columns]

0.3 OLS

La variable dependiente en estudio es `vive_padre` y todas las demas son las posibles variables explicativas sobre que el padre se encuentre o no viviendo en el hogar

```

[10]: y=datos_junaeb['vive_padre']
      X=datos_junaeb[['vive_madre', 'n_personas', 'n_habitaciones', 'cercania_juegos', 'cercania_servicios']]

      X=sm.add_constant(X)
      model = sm.OLS(y, X)
      results = model.fit()
      print(results.summary())

```

```

                        OLS Regression Results
=====
Dep. Variable:          vive_padre      R-squared:          0.169
Model:                  OLS             Adj. R-squared:      0.167
Method:                 Least Squares    F-statistic:         143.3
Date:                   Thu, 15 Sep 2022  Prob (F-statistic):    2.85e-247
Time:                   12:10:43          Log-Likelihood:      -3567.3
No. Observations:       6374             AIC:                7155.
Df Residuals:           6364             BIC:                7222.
Df Model:                9
Covariance Type:        nonrobust
=====
=====


```

	coef	std err	t	P> t	[0.025

const	0.1166	0.053	2.186	0.029	0.012
0.221					
vive_madre	0.1214	0.034	3.575	0.000	0.055
0.188					
n_personas	0.0546	0.005	12.073	0.000	0.046
0.063					
n_habitaciones	-0.0409	0.007	-6.151	0.000	-0.054
-0.028					
cercania_juegos	-0.0104	0.013	-0.799	0.424	-0.036
0.015					
cercania_servicios	0.0149	0.014	1.050	0.294	-0.013
0.043					
edad_primer_parto	0.0095	0.001	8.903	0.000	0.007
0.012					
area	-0.0777	0.018	-4.238	0.000	-0.114
-0.042					
educm	-0.0160	0.001	-10.983	0.000	-0.019
-0.013					
educp	0.0334	0.001	31.197	0.000	0.031
0.036					
=====					
Omnibus:	762.716	Durbin-Watson:		1.981	
Prob(Omnibus):	0.000	Jarque-Bera (JB):		825.499	
Skew:	-0.831	Prob(JB):		5.56e-180	
Kurtosis:	2.410	Cond. No.		315.	
=====					

Warnings:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Para realizar las pruebas de hipotesis de significancia individual utilizamos el valor p, puesto que el tamaño de la muestra no afecta a los resultados. La tabla de resultados nos  muestra que existen 2 de 9 variables no significativas, las cuales corresponden a cercania de juegos y cercania de servicios, el resto de las variables son significativas en su mayoría con 99% de confianza.

En cuanto a la interpretación de las variables explicativas, se tiene que por ejemplo la probabilidad de que el padre viva en el hogar aumenta en un 3,34% cuando aumenta en una unidad el año educacional del padre. Lo mismo pasa con la variable vive_madre, donde podemos notar que la probabilidad de que el padre en el hogar aumenta en 12.14% cuando la madre vive en el hogar.

[11]: `## 3. Probit`

```
model = sm.Probit(y, X)
```

```
probit_model = model.fit()
print(probit_model.summary())

mfx = probit_model.get_margeff()
print(mfx.summary())
```

Optimization terminated successfully.

Current function value: 0.538584

Iterations 5

Probit Regression Results

```
=====
Dep. Variable:          vive_padre    No. Observations:          6374
Model:                  Probit        Df Residuals:              6364
Method:                 MLE          Df Model:                  9
Date:                  Thu, 15 Sep 2022    Pseudo R-squ.:            0.1353
Time:                  12:10:43          Log-Likelihood:           -3432.9
converged:              True            LL-Null:                  -3969.9
Covariance Type:        nonrobust        LLR p-value:              2.017e-225
=====
```

```
=====
                                coef    std err          z      P>|z|      [0.025
0.975]
-----
const                -1.2186      0.174     -6.999     0.000     -1.560
-0.877
vive_madre           0.3393      0.108      3.131     0.002      0.127
0.552
n_personas           0.1716      0.015     11.479     0.000      0.142
0.201
n_habitaciones      -0.1245      0.021     -5.862     0.000     -0.166
-0.083
cercania_juegos     -0.0342      0.042     -0.807     0.420     -0.117
0.049
cercania_servicios   0.0412      0.046      0.889     0.374     -0.050
0.132
edad_primer_parto    0.0336      0.004      9.332     0.000      0.027
0.041
area                -0.2388      0.061     -3.923     0.000     -0.358
-0.119
educm               -0.0511      0.005    -10.302     0.000     -0.061
-0.041
educp                0.0974      0.004     26.975     0.000      0.090
0.104
=====
```

Probit Marginal Effects


```

=====
Dep. Variable:          vive_padre
Method:                dydx
At:                    overall
=====
=====

```

	dy/dx	std err	z	P> z	[0.025
0.975]					

vive_madre	0.1041	0.033	3.137	0.002	0.039
0.169					
n_personas	0.0526	0.004	11.759	0.000	0.044
0.061					
n_habitaciones	-0.0382	0.006	-5.901	0.000	-0.051
-0.026					
cercania_juegos	-0.0105	0.013	-0.807	0.420	-0.036
0.015					
cercania_servicios	0.0126	0.014	0.889	0.374	-0.015
0.040					
edad_primer_parto	0.0103	0.001	9.483	0.000	0.008
0.012					
area	-0.0732	0.019	-3.936	0.000	-0.110
-0.037					
educm	-0.0157	0.001	-10.489	0.000	-0.019
-0.013					
educp	0.0299	0.001	32.313	0.000	0.028
0.032					
=====					
=====					

Al igual que en el caso de OLS, utilizamos el valor p como prueba de hipotesis de significancia individual, ya que se el tamaño de la muestra no afecta en los resultados. Los resultados muestran que existen 2 de 9 variables no significativas, las cuales corresponden a cercania de juegos y cercania de servicio, las demás variables son significativas con un 99% de confianza. En esta oación la estimacion de los betas no representan cambio marginal, dada la forma no lineal. Es por esto que estudiamos los efctos marginales (dy/dx) donde existirán distintas interpretaciones dependiendo de si la variable es discreta o continua. En el caso de las variables continuas por ejemplo: para la variable educm, a medida que aumenta en una unidad los años de escolaridad de la madre, la probabilidad de que el padre viva en la casa disminuye en un 1,57%. Para la variables discreta; se tiene que para el area, si el lugar se encuentra en un sector urbano, disminuye en un 7,32% la probabilidad de que el padre viva en la casa.

```

[12]: # 4. Logit

model = sm.Logit(y, X)
logit_model = model.fit()
print(logit_model.summary())

```

```
mfx = logit_model.get_margeff()
print(mfx.summary())
```

Optimization terminated successfully.
 Current function value: 0.535948
 Iterations 6

Logit Regression Results

```
=====
Dep. Variable:          vive_padre  No. Observations:          6374
Model:                  Logit      Df Residuals:              6364
Method:                 MLE        Df Model:                  9
Date:                  Thu, 15 Sep 2022  Pseudo R-squ.:            0.1395
Time:                  12:10:43      Log-Likelihood:           -3416.1
converged:              True        LL-Null:                 -3969.9
Covariance Type:        nonrobust    LLR p-value:              1.133e-232
=====
```

```
=====
              coef      std err          z      P>|z|      [0.025
0.975]
-----
-----
const          -2.1427      0.298      -7.186      0.000      -2.727
-1.558
vive_madre       0.6072      0.181       3.358      0.001       0.253
0.962
n_personas       0.3123      0.028     11.296      0.000       0.258
0.366
n_habitaciones  -0.2208      0.039     -5.724      0.000      -0.296
-0.145
cercania_juegos -0.0602      0.072     -0.835      0.404      -0.201
0.081
cercania_servicios  0.0703      0.078       0.897      0.370      -0.083
0.224
edad_primer_parto  0.0594      0.006      9.427      0.000       0.047
0.072
area            -0.4016      0.104     -3.845      0.000      -0.606
-0.197
educm           -0.0950      0.009    -10.546      0.000      -0.113
-0.077
educp           0.1673      0.006     26.140      0.000       0.155
0.180
=====
```

Logit Marginal Effects

```
=====
Dep. Variable:          vive_padre
```

Method:	dydx				
At:	overall				
=====					
=====					
	dy/dx	std err	z	P> z	[0.025
0.975]					

vive_madre	0.1082	0.032	3.369	0.001	0.045
0.171					
n_personas	0.0556	0.005	11.652	0.000	0.046
0.065					
n_habitaciones	-0.0393	0.007	-5.770	0.000	-0.053
-0.026					
cercania_juegos	-0.0107	0.013	-0.835	0.404	-0.036
0.014					
cercania_servicios	0.0125	0.014	0.897	0.370	-0.015
0.040					
edad_primer_parto	0.0106	0.001	9.616	0.000	0.008
0.013					
area	-0.0716	0.019	-3.859	0.000	-0.108
-0.035					
educm	-0.0169	0.002	-10.854	0.000	-0.020
-0.014					
educp	0.0298	0.001	32.833	0.000	0.028
0.032					
=====					
=====					

En el caso del Logit, podemos notar que; las variables que son significativas siguen siendo las mismas en comparación al modelo Probit. Esto se debe a que los efectos marginales entre ambos métodos son prácticamente los mismos y solo difieren en la distribución del error que asume cada uno. La interpretación de los efectos marginales es idéntica a la realizada anteriormente para el modelo Probit.

0.4 Comentar resultados

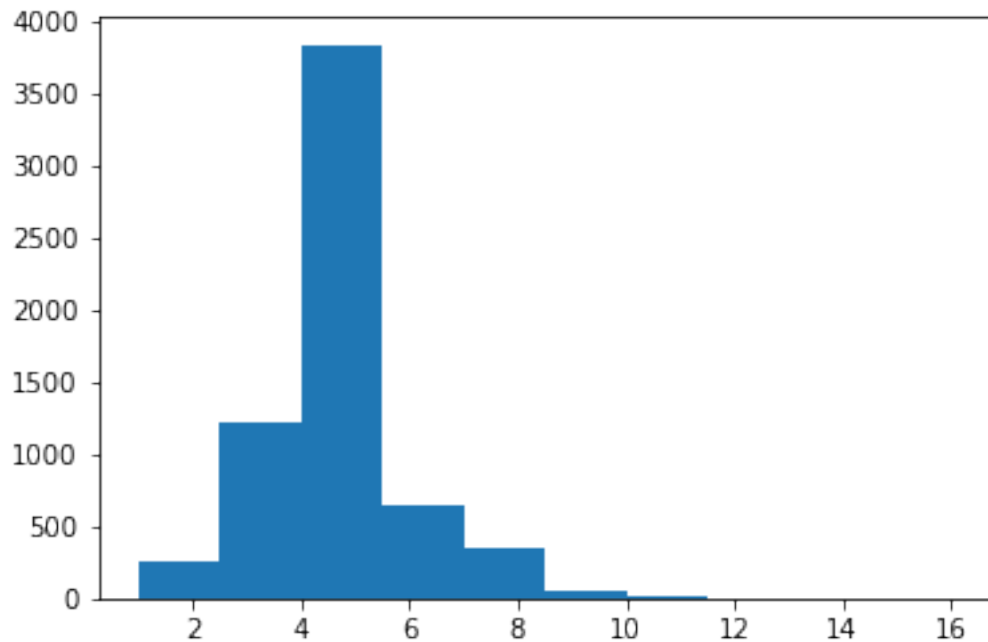
Tal como pudimos observar en cada uno de los modelos, las variables significativas no cambiaron entre estos, sin embargo pudimos notar que el valor predicho de la variable dependiente en el modelo OLS se encuentra fuera del rango de valores especificados para esta variable, es por ello que se utilizan los modelos Logit y Probit para las variables dependientes limitadas (dummy) dada la no linealidad del modelo.

0.5 Poisson

```
[13]: subset=datos_junaeb.loc[datos_junaeb['n_personas']<17]
      y=subset['n_personas']
      X=subset[['vive_padre','vive_madre','n_habitaciones','cercania_juegos','cercania_servicios', 'e
```

```
plt.hist(subset.n_personas)
subset.n_personas.head()
```

```
[13]: 0    3.0
      1    5.0
      2    5.0
      3    4.0
      4    5.0
      Name: n_personas, dtype: float64
```



A simple vista en este grafico podemos notar que no existirá sobredispersión, dado que la mayor concentracion de los datos no se encuentran tan alejados de la media

```
[14]: poisson=sm.GLM(y,X,family=sm.families.Poisson()).fit()
      print(poisson.summary())
```

Generalized Linear Model Regression Results

```
=====
Dep. Variable:          n_personas    No. Observations:          6374
Model:                  GLM           Df Residuals:              6365
Model Family:           Poisson       Df Model:                  8
Link Function:          log           Scale:                    1.0000
Method:                 IRLS          Log-Likelihood:            -11720.
Date:                   Thu, 15 Sep 2022    Deviance:                  2312.2
Time:                   12:10:44           Pearson chi2:              2.55e+03
No. Iterations:         5
```

```

Covariance Type: nonrobust
=====
=====
              coef      std err          z      P>|z|      [0.025
0.975]
-----
vive_padre      0.1168      0.014      8.181      0.000      0.089
0.145
vive_madre      0.6845      0.036     18.850      0.000      0.613
0.756
n_habitaciones  0.1529      0.005     32.270      0.000      0.144
0.162
cercania_juegos  0.0879      0.014      6.363      0.000      0.061
0.115
cercania_servicios  0.0817      0.015      5.522      0.000      0.053
0.111
edad_primer_parto -0.0035      0.001     -3.103      0.002     -0.006
-0.001
area            0.1611      0.021      7.749      0.000      0.120
0.202
educm           0.0026      0.002      1.572      0.116     -0.001
0.006
educp           0.0007      0.001      0.502      0.615     -0.002
0.003
=====
=====

```

```

[15]: print("fitted lambda")
      print(poisson.mu)

```

```

fitted lambda
[4.65988871 4.20261155 4.79037311 ... 3.38602891 3.96899529 4.14113266]

```

0.5.1 Interpretación de coeficientes Poisson

Dado la naturaleza logaritmica de la funcion de distribucion Poisson, podemos notar que la interpretacion de los coeficientes beta que entrega el modelo será del estilo Log-Nivel. Por ejemplo, si interpretamos el efecto de la variable `n_habitaciones` sobre `n_personas`, podemos decir que esta variable aumentará en un 15,2% cuando aumente en una unidad la variable `n_habitaciones`, es decir, cuando aumente en una la cantidad de habitaciones en el hogar. Es logico pensar esto ya que, si una casa tiene mas habitaciones, eventualmente podrian vivir más personas en ella.

A diferencia de la interpretacion para la variable discreta anterior, para interpretar las variables dummy cambia un poco. Si analizamos la variable `vive_madre` podemos ver que, la cantidad de personas que viven en la casa aumenta en un 68,4% cuando la madre efectivamente vive en el hogar. En cambio, al analizar la variable `vive_padre` notamos que la cantidad de personas que viven en la casa aumenta en un 11,68% cuando el padre vive en el hogar. Esto tiene lógica ya por lo general se da que los hijos pequeños dependen mucho de la madre, por lo que no es errado el pensamiento que

si la madre efectivamente vive en la casa, es probable que vivan mas personas en ella, a diferencia de lo que ocurre con el padre.

0.5.2 Test de sobredispersión y posible valor alpha

```
[16]: ov = (2.55*10**3)/6365  
      print(ov)
```

0.40062843676355064

Al realizar el test de sobredispersión, notamos que el cuociente del Pearson chi-cuadrado y Df Residuals nos da como resultado 0.4, el cual es un valor menor a 1, lo que nos indica que no existe sobredispersión. Por este motivo, no seria necesario realizar la aproximacion mediante la distribucion binomial negativa, pues esta se utiliza para relajar la distribucion de Poisson(cuando existe sobredispersión)

```
[17]: aux=((y-poisson.mu)**2-poisson.mu)/poisson.mu  
      auxr=sm.OLS(aux,poisson.mu).fit()  
      print(auxr.params)
```

x1 -0.110029
dtype: float64

Al calcular el valor alpha óptimo de la distribución binomial negativa mediante la regresión lineal que utiliza como único regresor la variable $aux = [(y - \mu)^2 - \mu] / [(y - \mu)^2 - \mu]$, nos entrega un valor de -0,11, el cual al momento de utilizarlo en la construcción del modelo binomial negativo nos despliega un error, puesto que alpha tiene que ser mayor a cero (dada la naturaleza de los logaritmos) es por esta razón que se decidió no utilizar la distribución binomial negativa en este caso.

0.6 Conclusiones finales.

Dado que la distribucion binomial negativa no se realizó, no fue posible contrastar la informacion entregada por ese modelo versus la de Poisson, pero a grandes rasgos podemos decir que, para este caso particular, el mejor modelo estimado fue el de Poisson dado que al no existir sobredispersión, el modelo no tiene ningun problema en cuanto a su funcionamiento. Cabe mencionar que la distribucion binomial negativa es un tipo de distribucion que se utiliza para la relajacion de la distribución Poisson, la cual en este caso no fue necesaria.

```
[ ]:
```