

# Tarea4\_Munoz

December 19, 2022

## 1 TAREA 4: Laboratorio de métodos aplicados avanzados

1.0.1 Autor: Sebastián Muñoz

1.0.2 Fecha: 09/12/22

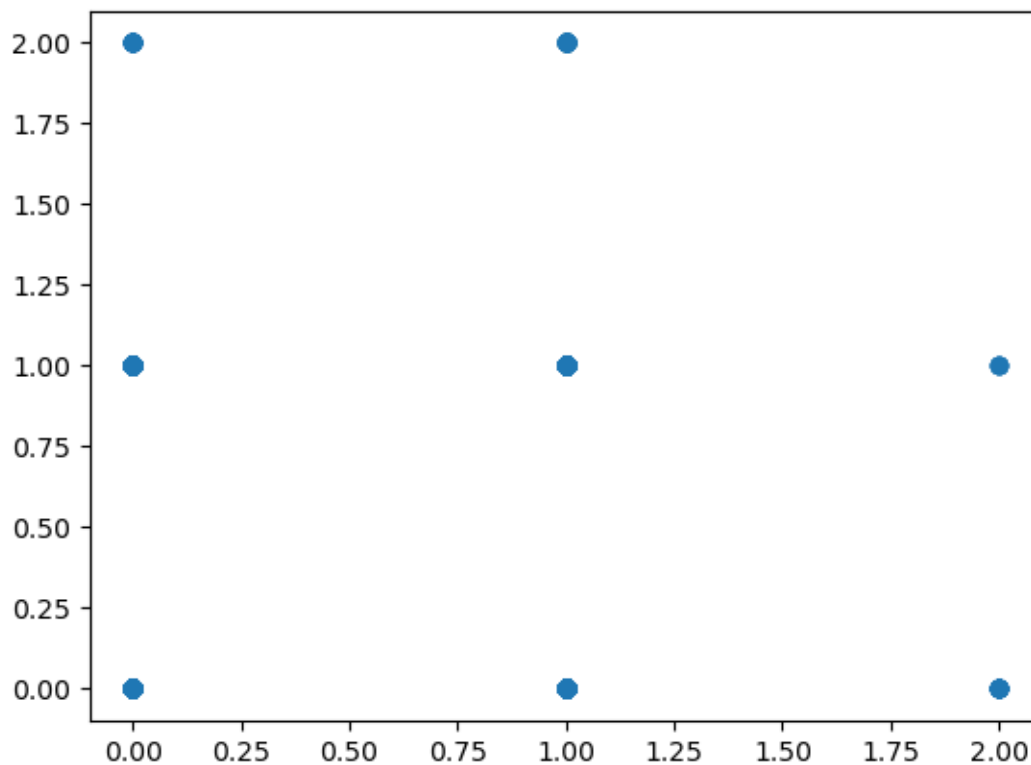
1.0.3 Pregunta 1

```
[1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
import statsmodels.formula.api as smf
import sklearn
import scipy
from scipy.linalg import eigh, cholesky
from scipy.stats import norm
import linearmodels.panel as lmp
from pylab import plot, show, axis, subplot, xlabel, ylabel, grid
import semopy
import seaborn as sns
from factor_analyzer import FactorAnalyzer
from sklearn.decomposition import PCA
```

```
[2]: juna2 = pd.read_csv('../data/junaeb2.csv')
juna2.dropna(inplace=True)
juna2.reset_index(drop=True, inplace=True)
```

```
[3]: plt.scatter(juna2['vive_padre'], juna2['vive_madre'])
```

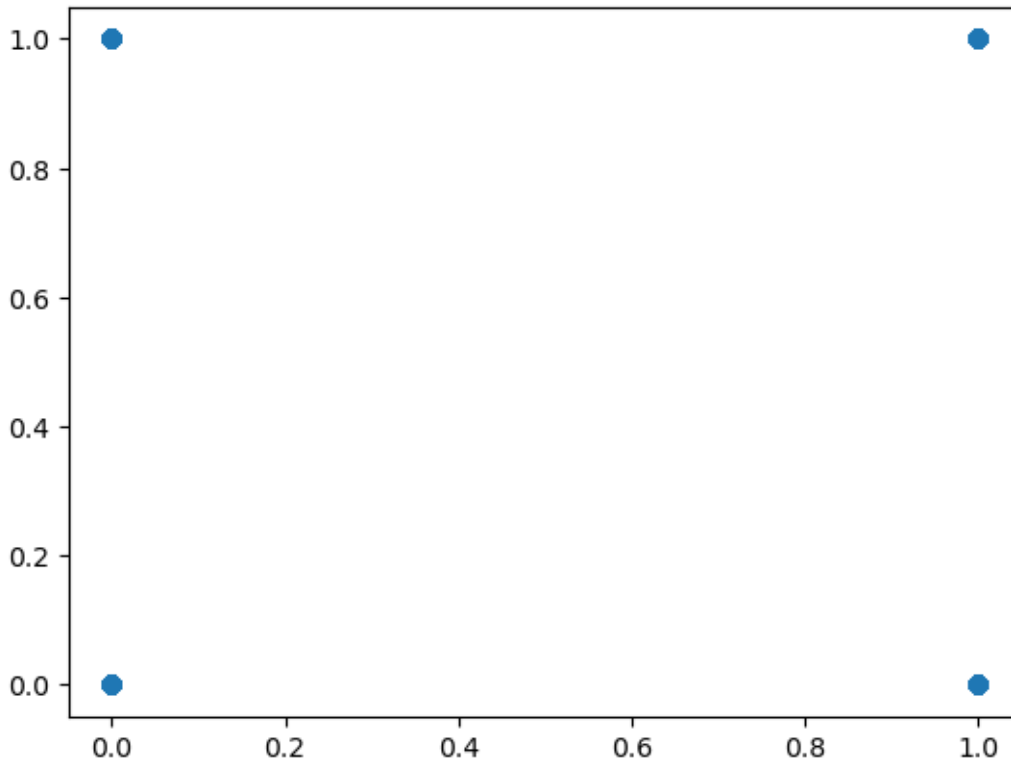
```
[3]: <matplotlib.collections.PathCollection at 0x18b4a2d5ca0>
```



Si se grafican las variables dummy **vive\_padre** y **vive\_madre** se observan outliers, donde existen muestras cuyo valor es 2 y claramente se encuentra fuera del rango 0-1 de las variables binarias; se asumirá que esto se produjo debido a un error de tipeo y con el objetivo de no descartar muestras se reemplazarán dichos valores por '1'.

```
[4]: juna2['vive_madre']=juna2['vive_madre'].replace(2,1)
juna2['vive_padre']=juna2['vive_padre'].replace(2,1)
plt.scatter(juna2['vive_padre'],juna2['vive_madre'])
```

```
[4]: <matplotlib.collections.PathCollection at 0x18b4a53f820>
```

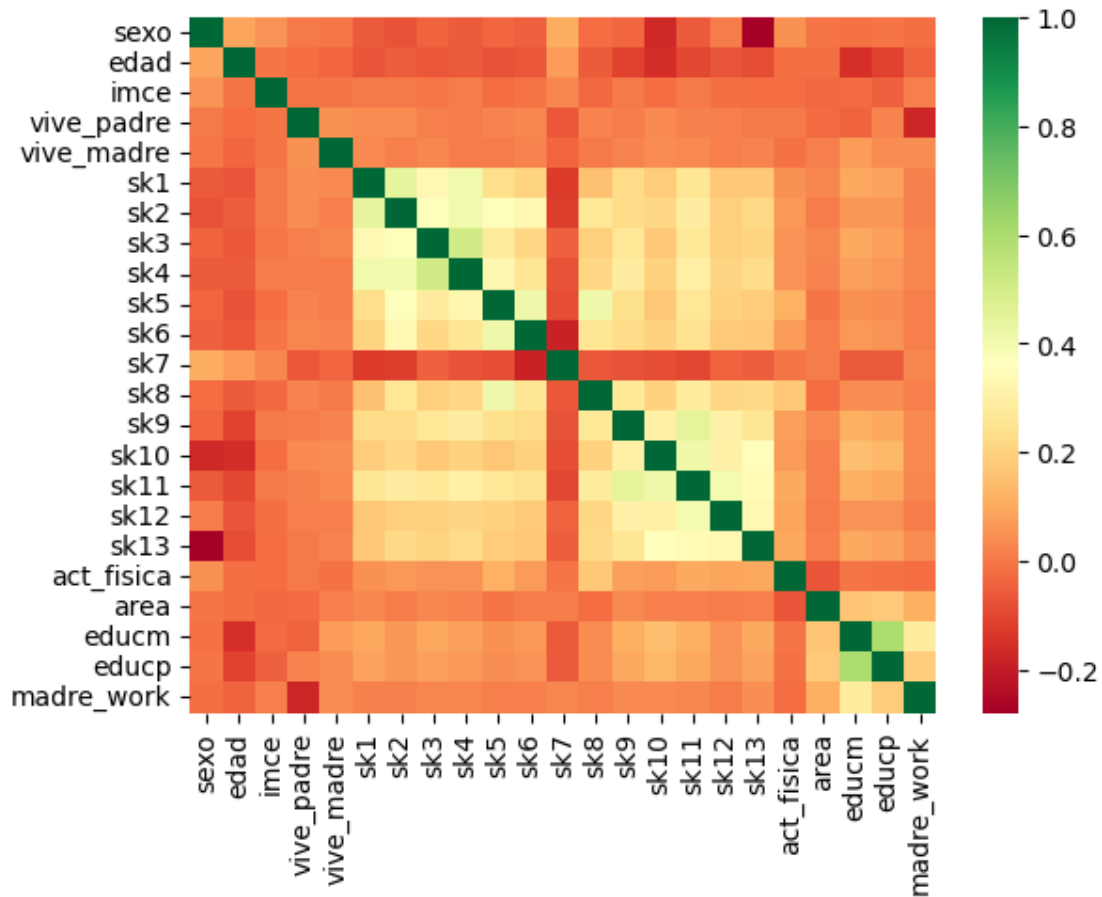


Se recodifican las variables que tienen un sentido positivo de tal forma que mientras mayor sea el puntaje más positiva es la respuesta ('5: siempre'), a excepción de sk7 la cual a mayor agresividad del niño mas bajo queda el puntaje ('1: siempre').

```
[5]: for i in range(1,14,1):
      if i != 7:
          juna2['sk%s'%i]=juna2['sk%s'%i].replace(1,'cinco')
          juna2['sk%s'%i]=juna2['sk%s'%i].replace(2,'cuatro')
          juna2['sk%s'%i]=juna2['sk%s'%i].replace(4,2)
          juna2['sk%s'%i]=juna2['sk%s'%i].replace(5,1)
          juna2['sk%s'%i]=juna2['sk%s'%i].replace('cinco',5)
          juna2['sk%s'%i]=juna2['sk%s'%i].replace('cuatro',4)

sks=juna2[['sk1','sk2','sk3','sk4','sk5','sk6','sk7','sk8','sk9','sk10','sk11','sk12','sk13']]
sns.heatmap(juna2.corr(), cmap='RdYlGn')
```

```
[5]: <AxesSubplot:>
```



Las variables observadas que presentan entre si una mayor correlación positiva son: educp y educm, por ejemplo, mientras mayor escolaridad tenga la madre, el padre también tendrá más años de escolaridad y viceversa.

Para las medidas, sk3 y sk4 son las que mayor correlación positiva presentan, ya que tienen relación con cómo y cuánto los niños expresan sus sentimientos. sk5 y sk8 se correlacionan positivamente en torno a la disposición de participación que tienen los niños a jugar con otros. La correlación positiva entre sk9 y sk11 se basa principalmente en el grado de curiosidad y de cuestionamiento que tiene el niño con su entorno, haciendo que el infante tienda a hacerle preguntas a sus padres.

```
[6]: desc=juna2.describe()
desc
```

```
[6]:
```

	sexo	edad	imce	vive_padre	vive_madre \
count	57357.000000	57357.000000	57357.000000	57357.000000	57357.000000
mean	0.535349	81.880032	1.018703	0.721080	0.974946
std	0.498753	3.767887	1.367474	0.448472	0.156289
min	0.000000	62.000000	-5.020000	0.000000	0.000000
25%	0.000000	80.000000	0.110000	0.000000	1.000000

50%	1.000000	81.000000	0.980000	1.000000	1.000000
75%	1.000000	83.000000	1.930000	1.000000	1.000000
max	1.000000	107.000000	5.040000	1.000000	1.000000

	sk1	sk2	sk3	sk4	sk5 \
count	57357.000000	57357.000000	57357.000000	57357.000000	57357.000000
mean	4.893561	4.614450	4.746500	4.753177	4.736301
std	0.375685	0.646275	0.572033	0.562731	0.558155
min	1.000000	1.000000	1.000000	1.000000	1.000000
25%	5.000000	4.000000	5.000000	5.000000	5.000000
50%	5.000000	5.000000	5.000000	5.000000	5.000000
75%	5.000000	5.000000	5.000000	5.000000	5.000000
max	5.000000	5.000000	5.000000	5.000000	5.000000

	...	sk9	sk10	sk11	sk12 \
count	...	57357.000000	57357.000000	57357.000000	57357.000000
mean	...	4.677424	4.154349	4.623028	4.510696
std	...	0.651805	0.933007	0.658345	0.786647
min	...	1.000000	1.000000	1.000000	1.000000
25%	...	5.000000	4.000000	4.000000	4.000000
50%	...	5.000000	4.000000	5.000000	5.000000
75%	...	5.000000	5.000000	5.000000	5.000000
max	...	5.000000	5.000000	5.000000	5.000000

	sk13	act_fisica	area	educm	educp \
count	57357.000000	57357.000000	57357.000000	57357.000000	57357.000000
mean	4.316526	2.559461	0.907213	13.084175	12.988807
std	0.977397	1.070257	0.290137	3.321205	3.420867
min	1.000000	1.000000	0.000000	0.000000	0.000000
25%	4.000000	2.000000	1.000000	12.000000	11.000000
50%	5.000000	2.000000	1.000000	13.000000	13.000000
75%	5.000000	3.000000	1.000000	15.000000	14.000000
max	5.000000	5.000000	1.000000	22.000000	22.000000

	madre_work
count	57357.000000
mean	0.102498
std	0.941063
min	-1.000000
25%	-1.000000
50%	0.000000
75%	1.000000
max	1.000000

[8 rows x 23 columns]


La pregunta de cuestionario sk que obtuvo un puntaje más alto en promedio fue sk1: muestra

afecto con los padres, con un puntaje promedio cercano a 5 (el mayor puntaje), del mismo modo esta pregunta fue la que obtuvo menor varianza en sus respuestas, con una varianza de  $0.0376^2 = 0.00141$

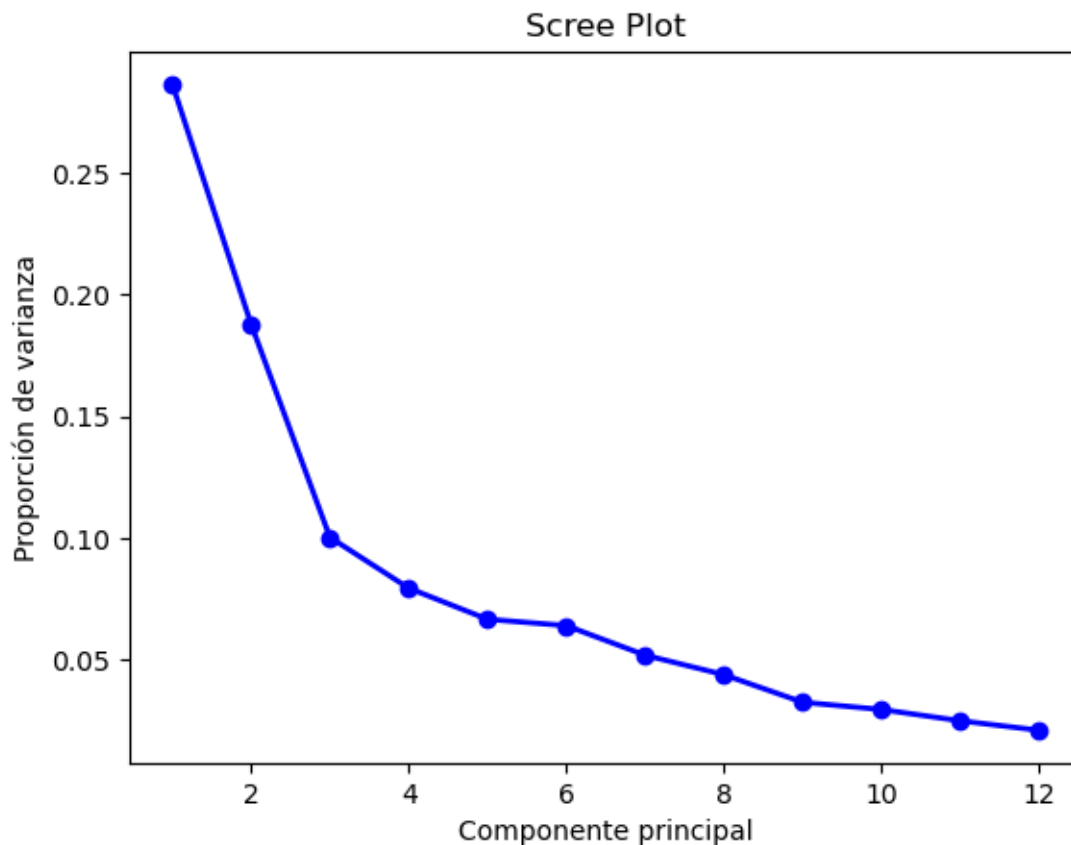
#### 1.0.4 Pregunta 2

```
[7]: pca = PCA(n_components='mle')
pca_features = pca.fit_transform(sks)
print(pca.explained_variance_ratio_)
```

```
[0.28626422 0.18771356 0.10019573 0.07933285 0.0665471  0.06393448
 0.05180848 0.04378094 0.03246624 0.02949163 0.02484546 0.02097125]
```

Se estimaron un total 12 componentes principales por medio de máxima verosimilitud, o sea el máximo de componentes que es posible estimar, por consiguiente, cada uno de estos componentes entregan información significativa estadísticamente. 

```
[8]: PC_values = np.arange(pca.n_components_) + 1
plt.plot(PC_values, pca.explained_variance_ratio_, 'o-', linewidth=2,
         color='blue')
plt.title('Scree Plot')
plt.xlabel('Componente principal')
plt.ylabel('Proporción de varianza')
plt.show()
```



```
[9]: pca_df = pd.DataFrame(data=pca_features,columns=['PC1', 'PC2', 'PC3', 'PC4', 'PC5', 'PC6', 'PC7', 'PC8', 'PC9', 'PC10', 'PC11', 'PC12'])
pca_df.describe().apply(lambda s: s.apply('{0:.3f}'.format))
```

```
[9]:
```

	PC1	PC2	PC3	PC4	PC5	PC6 \
count	57357.000	57357.000	57357.000	57357.000	57357.000	57357.000
mean	-0.000	0.000	0.000	-0.000	0.000	-0.000
std	1.467	1.188	0.868	0.772	0.707	0.693
min	-1.903	-3.220	-4.116	-3.465	-3.925	-3.932
25%	-1.092	-0.725	-0.470	-0.245	-0.294	-0.400
50%	-0.271	0.187	0.005	-0.090	0.094	0.031
75%	0.785	0.814	0.454	0.515	0.339	0.273
max	10.919	4.584	6.318	3.695	5.215	4.582

	PC7	PC8	PC9	PC10	PC11	PC12
count	57357.000	57357.000	57357.000	57357.000	57357.000	57357.000
mean	0.000	0.000	-0.000	0.000	-0.000	0.000
std	0.624	0.574	0.494	0.471	0.432	0.397
min	-4.605	-4.376	-4.205	-3.554	-2.456	-3.108
25%	-0.315	-0.289	-0.212	-0.192	-0.218	-0.056

50%	0.063	0.028	-0.019	0.039	0.049	0.020
75%	0.313	0.195	0.207	0.120	0.114	0.041
max	4.669	5.222	4.466	3.779	3.834	3.286

### 1.0.5 Pregunta 3

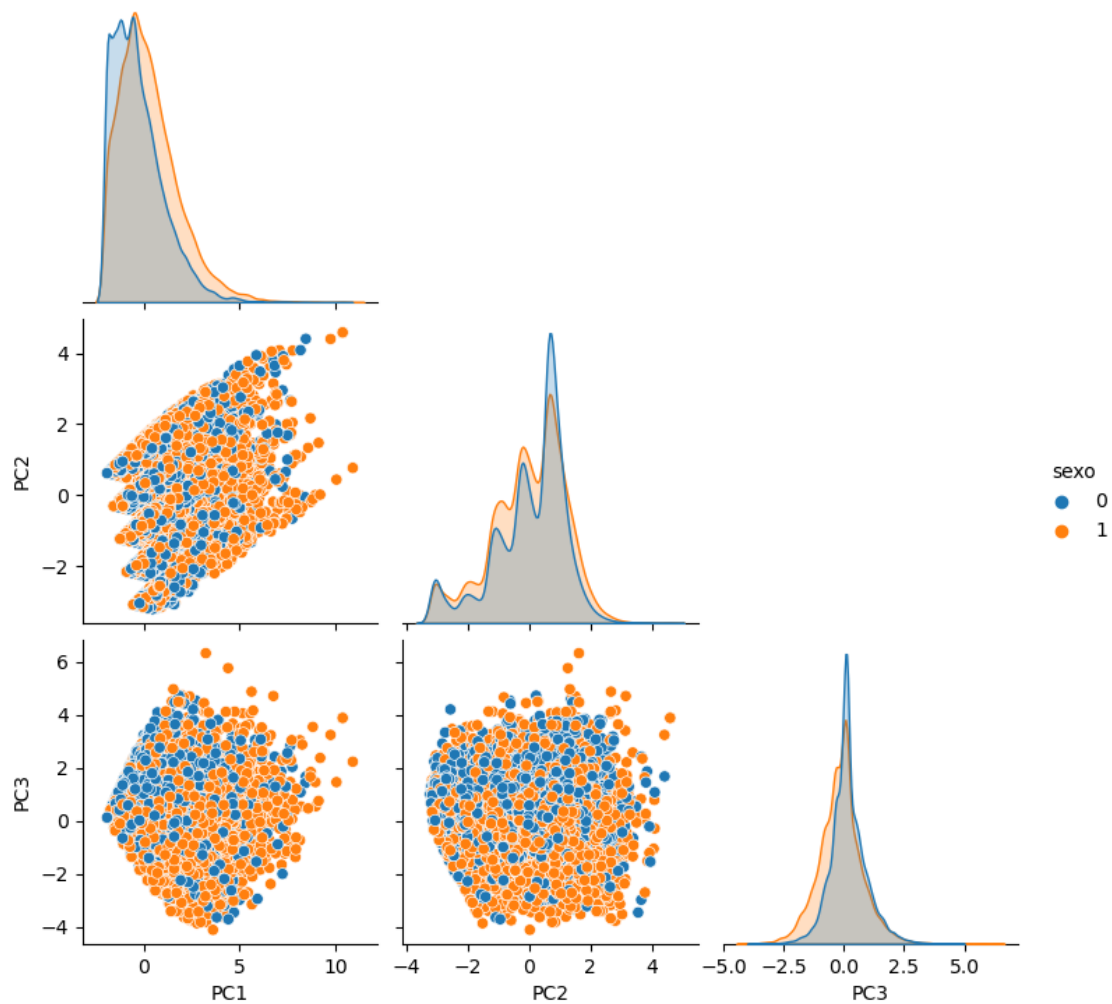
Los 3 primeros componentes principales son los siguientes:

```
[10]: print(pca.explained_variance_ratio_[0:3])
```

```
[0.28626422 0.18771356 0.10019573]
```

```
[11]: subset=pca_df[['PC1', 'PC2', 'PC3']].copy()
subset['sexo']=juna2['sexo']
sns.pairplot(subset, hue='sexo', corner=True)
```

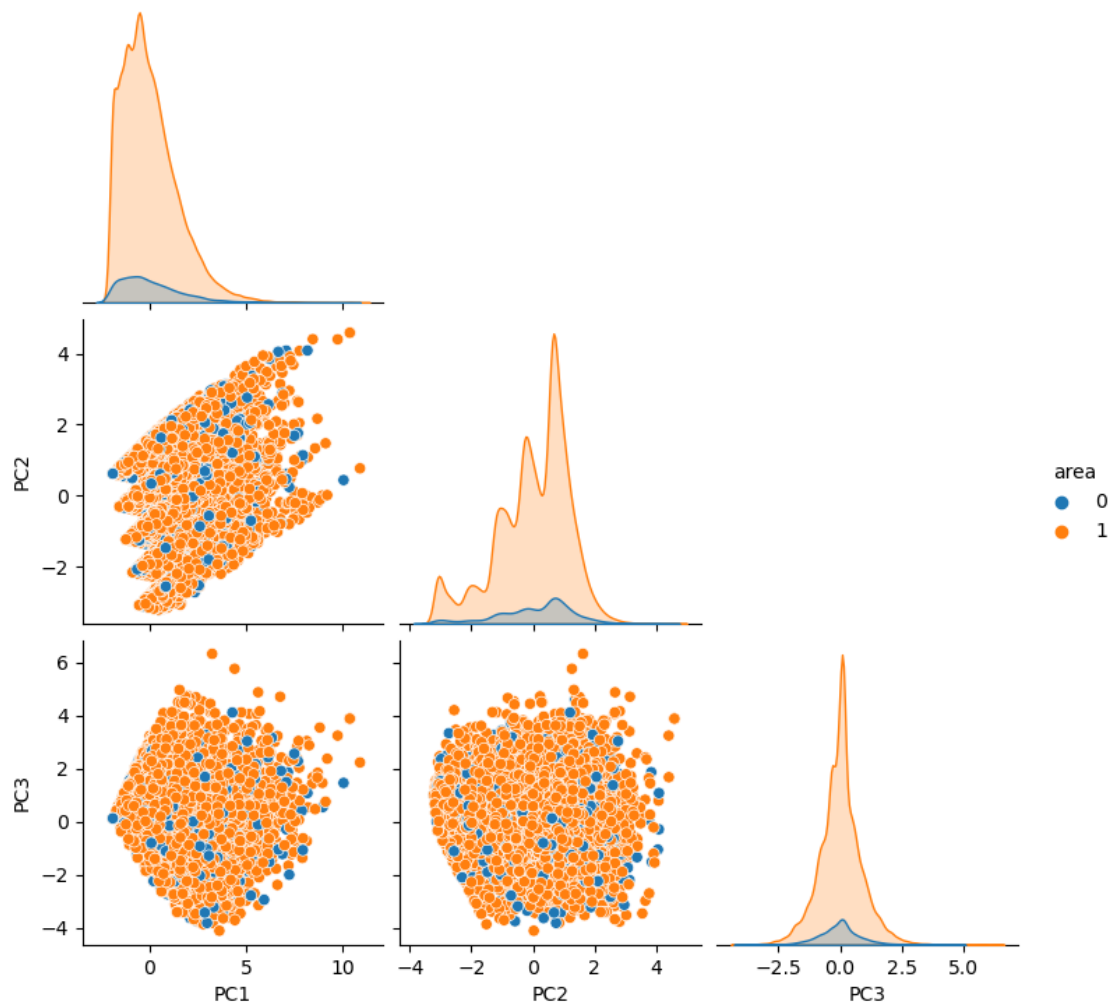
```
[11]: <seaborn.axisgrid.PairGrid at 0x18b2a37dc40>
```





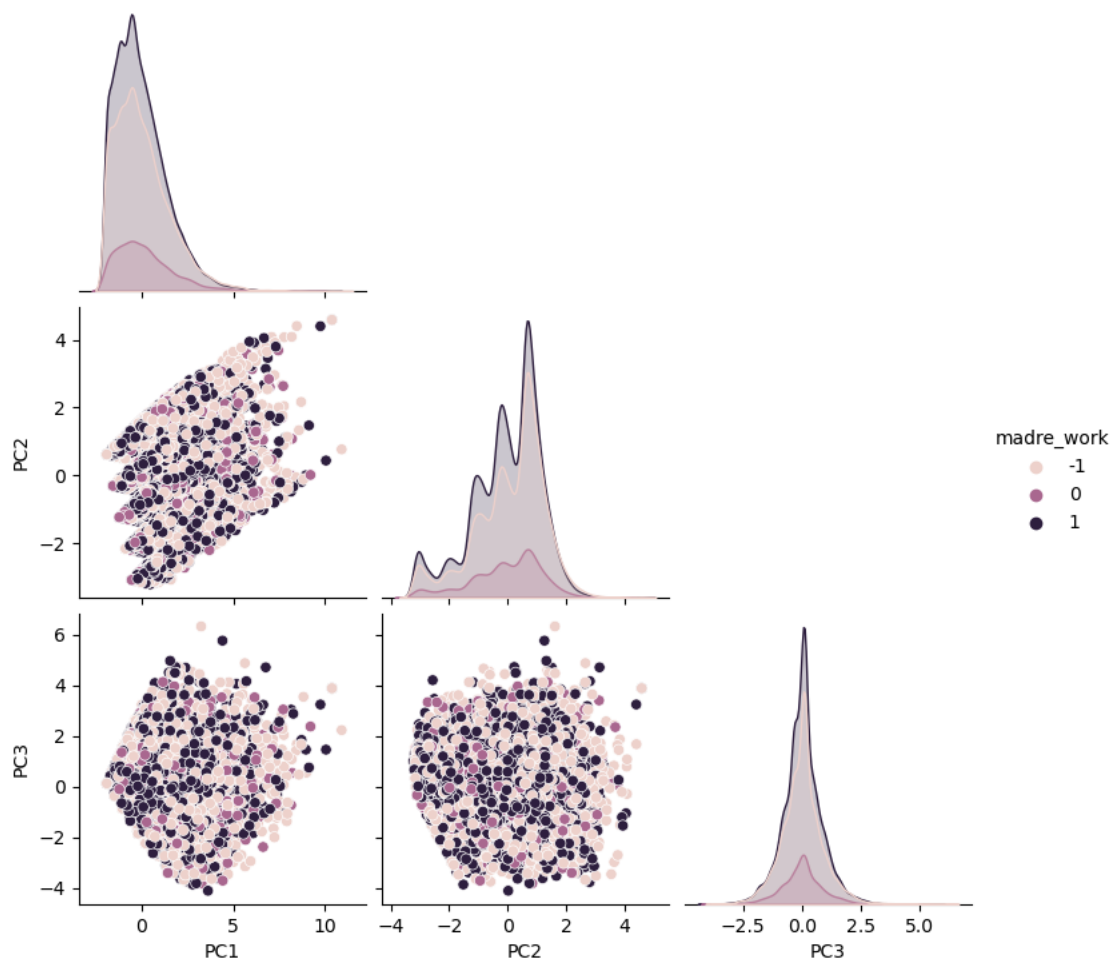
```
[12]: subset=pca_df[['PC1', 'PC2', 'PC3']].copy()
subset['area']=juna2['area']
sns.pairplot(subset, hue='area', corner=True)
```

```
[12]: <seaborn.axisgrid.PairGrid at 0x18b4eb23040>
```



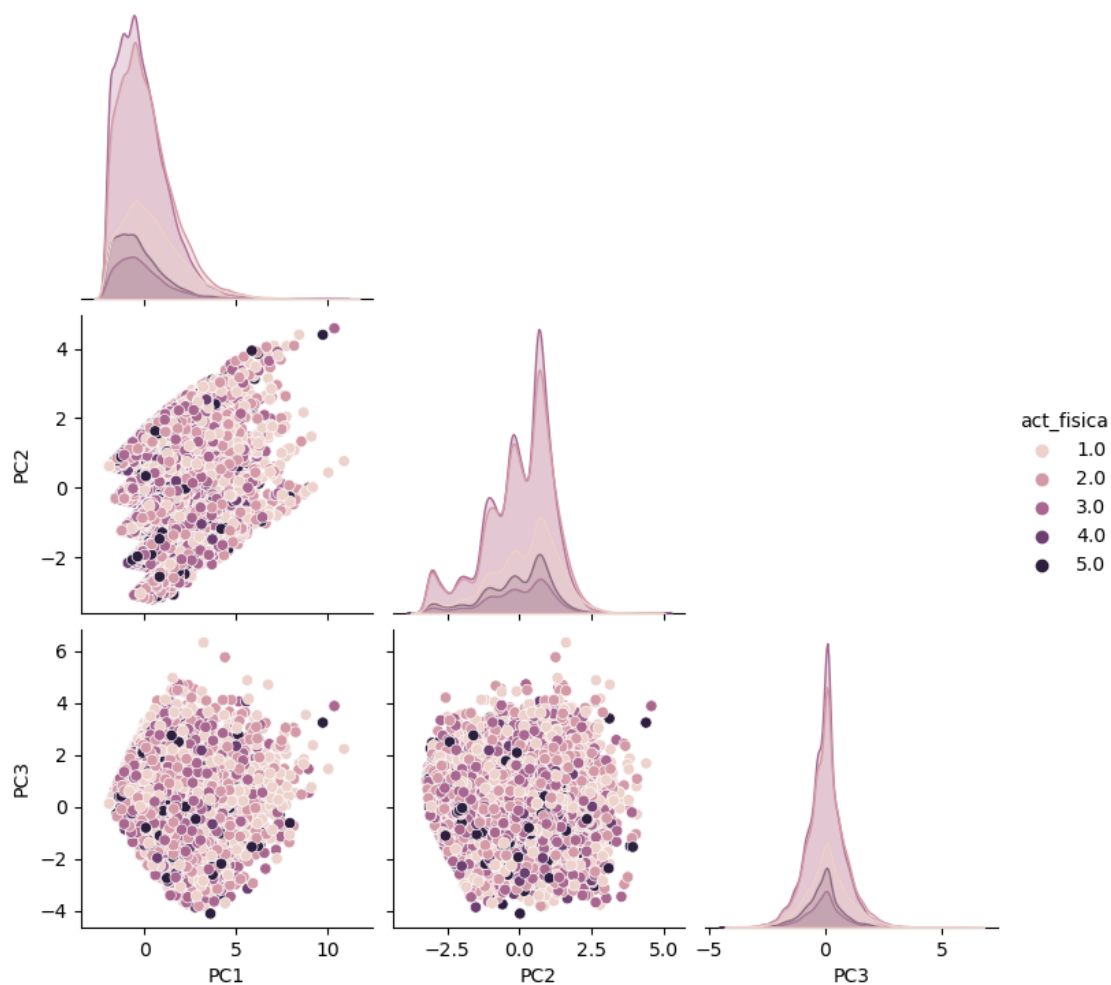
```
[13]: subset=pca_df[['PC1', 'PC2', 'PC3']].copy()
subset['madre_work']=juna2['madre_work']
sns.pairplot(subset, hue='madre_work', corner=True)
```

```
[13]: <seaborn.axisgrid.PairGrid at 0x18b4eba7e80>
```



```
[14]: subset=pca_df[['PC1', 'PC2', 'PC3']].copy()
subset['act_fisica'] = juna2['act_fisica']
sns.pairplot(subset, hue='act_fisica', corner=True)
```

```
[14]: <seaborn.axisgrid.PairGrid at 0x18b4fa99580>
```



```
[15]: pca_vectors = pd.DataFrame(data = pca.components_)
      pca_vectors.head()
```

```
[15]:
```

	0	1	2	3	4	5	6	\
0	-0.104286	-0.229217	-0.169568	-0.185835	-0.185243	-0.251850	0.340735	
1	-0.006915	-0.030275	-0.062554	-0.053272	-0.039041	0.031414	-0.926864	
2	-0.083371	-0.266831	-0.183221	-0.190528	-0.297768	-0.334258	-0.114556	
3	-0.001878	0.023796	0.032604	0.026872	0.043287	-0.003485	-0.035722	
4	-0.151066	-0.289580	-0.313595	-0.306029	0.016044	-0.173908	-0.037906	
	7	8	9	10	11	12		
0	-0.292287	-0.240529	-0.405828	-0.288448	-0.292894	-0.427178		
1	-0.099300	-0.085178	-0.163943	-0.100178	-0.152161	-0.231332		
2	-0.478687	-0.078622	0.389056	0.001042	0.090213	0.493595		
3	0.096996	-0.102579	-0.698533	-0.145107	-0.016440	0.682113		
4	0.754956	-0.089218	0.184764	-0.090541	-0.207857	0.077011		

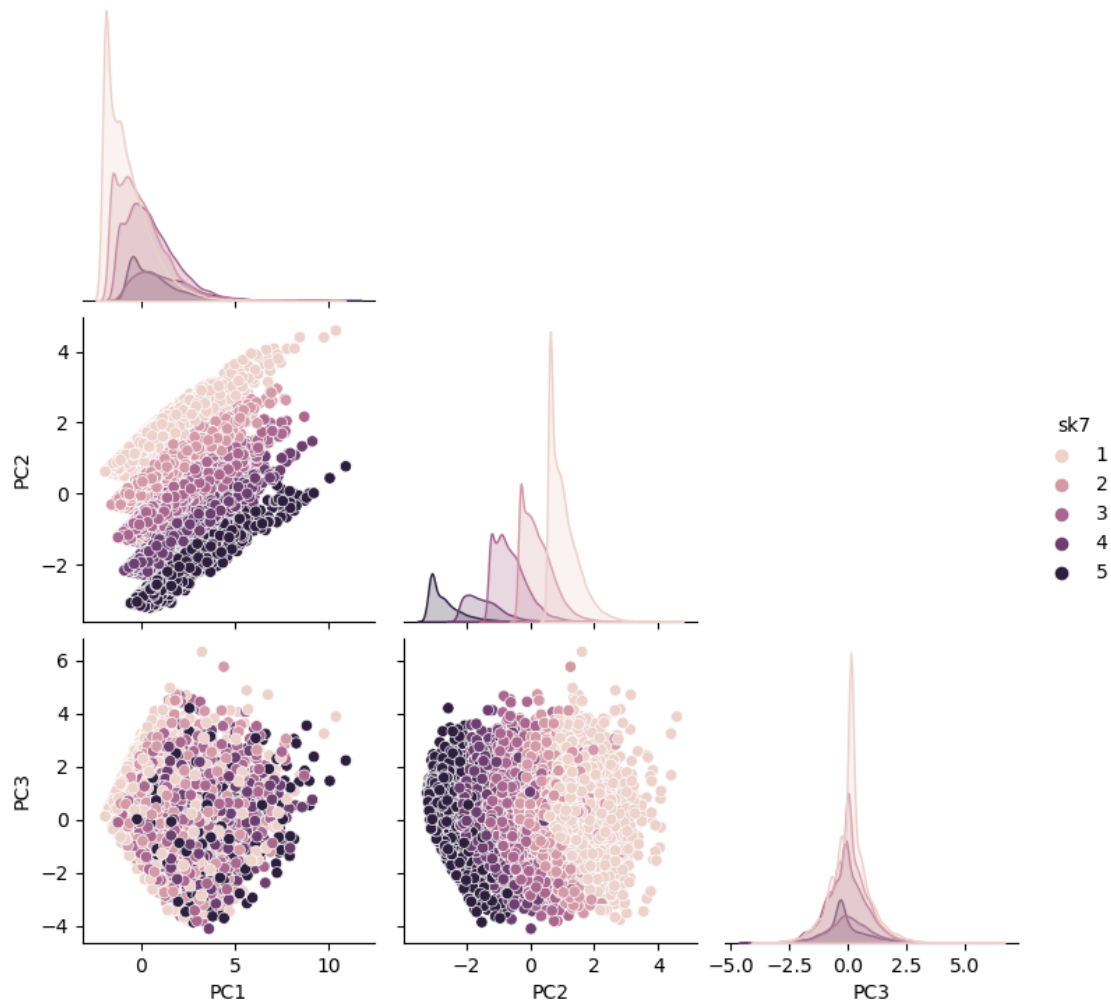
Como fue posible examinar, usando las variables sexo, area, madre\_work y act\_fisica no fue posible observar diferencias significativas entre grupos para los tres primeros componentes principales, es decir, ninguno de estos componentes pudo capturar información de las variables antes mencionadas; probablemente esto es así debido a que dichas variables usadas tienen muy poca correlación con las variables sk que entregan la mayor información para cada uno de los 3 componentes, donde en base a la matriz de los pesos relativos de cada una de las variables sk en los componentes principales:

- 1º componente principal -> sk13
- 2º componente principal -> sk7 y sk10
- 3º componente principal -> sk13 y sk8

Por ejemplo, clasificando a partir de sk7 se nota que la información de esta variable es en gran parte capturada por el componente principal PC2 y es posible observar grupos en base al puntaje de 0 a 5 que posee dicha variable sk7, como se presenta en el siguiente gráfico:

```
[16]: subset=pca_df[['PC1', 'PC2', 'PC3']].copy()
subset['sk7']=juna2['sk7']
sns.pairplot(subset, hue='sk7', corner=True)
```

```
[16]: <seaborn.axisgrid.PairGrid at 0x18b56947310>
```



#### 1.0.6 Pregunta 4

A partir del mismo set de variables sk1-sk13 realice un EFA. En particular determine el numero optimo de factores y las variables que se asocian a cada factor. Tambien discuta si existen variables que no son informativas (Hint: para realizar un EFA, todas las variables deben estar representatadas en el mismo sentido logico. Si una carateristica es negativa debe ser invertida en la escala, de tal forma que todas las variables representen aspectos positivos).



```
[17]: fa = FactorAnalyzer(rotation='promax')
fa_features=fa.fit_transform(sks)
fa.loadings_
```

```
[17]: array([[ 0.01513615,  0.60602878, -0.03702337],
             [-0.03606024,  0.49666637,  0.23058715],
             [ 0.02403653,  0.64646172, -0.0437862 ],
             [ 0.00377166,  0.73995746, -0.03352797],
```

```

[-0.14794493, -0.02660843,  0.86280532],
[ 0.00580677,  0.04159328,  0.51848018],
[-0.01566549, -0.04014277, -0.1397605 ],
[ 0.14636207, -0.10733803,  0.51462899],
[ 0.48339265,  0.08125916,  0.05198111],
[ 0.6197227 , -0.0344817 , -0.03268303],
[ 0.69684439,  0.03003392,  0.00592717],
[ 0.56945833, -0.02303978, -0.00404897],
[ 0.52731831,  0.01951838, -0.01165986]])

```

A partir de los loadings provenientes del análisis factorial, se han estimado de un total de 3 factores como cantidad óptima. Se observa que los pesos relativos de sk7 para los factores 1, 2 y 3 fueron -0.015, -0.040 y -0.1397 respectivamente; de esto se puede concluir de que cuán agresiva es una persona es lo que menos información aporta en los 3 factores.

```

[18]: correlacion=sks.corr(method='spearman')['sk7']
      correlacion

```

```

[18]: sk1      -0.141484
      sk2      -0.148371
      sk3      -0.066243
      sk4      -0.096674
      sk5      -0.105019
      sk6      -0.197593
      sk7       1.000000
      sk8      -0.087732
      sk9      -0.094519
      sk10     -0.111258
      sk11     -0.118664
      sk12     -0.065862
      sk13     -0.078280
      Name: sk7, dtype: float64

```

Notar también que la correlación entre sk7 y las demás es insignificante, debido a esto se descarta incluir esta variable en los factores.

Por lo tanto, en base a estos pesos relativos, las variables asociadas a cada factor serán los siguientes, ordenados por importancia en cada factor:

- Factor 1: sk11 , sk10 , sk12 , sk13 , sk9
- Factor 2: sk4 , sk3 , sk1 , sk2
- Factor 3: sk5 , sk6 , sk8

El factor 1 incluye las medidas que representan la curiosidad que presenta el niño con su entorno y su deseo de aprender cosas desconocidas.

El factor 2 incorpora las medidas que representan la capacidad que tiene el niño para poder expresar sus sentimientos y emociones.

El factor 3 abarca las medidas que representan la capacidad y competencia del niño para poder

integrarse y convivir con sus pares.

### 1.0.7 Pregunta 5

Con los resultados obtenidos en la Pregunta 4, proponga un CFA donde cada variable solo se asocia con un factor. Entregue un nombre a cada factor que representa el concepto comun entre todas las variables. Reporte la importancia de cada medida (variable) a cada factor e indique la correlacion entre factores.

En base al analisis exploratorio de Semopy, el cual entrega una aproximación del CFA, entrega 4 factores y las variables que deberian asociarse a ellas, sin embargo el analisis factorial estimó solamente 3 factores, por lo que este modelo será utilizado como guía para poder corresponder cada medida a solo un factor en base también a los loadings calculados anteriormente.

```
[19]: print(semopy.efa.explore_cfa_model(sks, pval=0.05))

eta1 =~ sk11 + sk9 + sk10 + sk12
eta2 =~ sk7 + sk6
eta3 =~ sk4 + sk2 + sk11 + sk5 + sk3 + sk9 + sk1 + sk6 + sk8 + sk12
eta4 =~ sk11 + sk12 + sk13
```

```
[20]: # modelo
mod = """
# measurement model
curiosidad =~ sk11 + sk10 + sk12 + sk13 + sk9
expresion =~ sk4 + sk3 + sk1 + sk2
sociabilidad =~ sk5 + sk6 + sk8
        """

model = semopy.Model(mod)
out=model.fit(sks)
print(out)
```

```
Name of objective: MLW
Optimization method: SLSQP
Optimization successful.
Optimization terminated successfully
Objective value: 0.141
Number of iterations: 31
Params: 1.092 0.897 1.054 0.799 0.912 0.557 1.061 1.020 1.143 0.151 0.204 0.093
0.701 0.161 0.367 0.198 0.243 0.479 0.434 0.597 0.278 0.156 0.112 0.103 0.229
0.111 0.161
```

A continuación se indica la correlación entre los 3 factores, los cuales presentan entre ellos una correlación positiva y medianamente alta, sobre todo el factor 2 con el factor 3.

```
[21]: fa_df = pd.DataFrame(data=fa_features,columns=['eta1', 'eta2', 'eta3'])
fa_df.corr().apply(lambda s: s.apply('{0:.3f}'.format))
```

```
[21]:      eta1  eta2  eta3
eta1  1.000  0.685  0.712
eta2  0.685  1.000  0.770
eta3  0.712  0.770  1.000
```

```
[22]: model.inspect(mode='list', what="names", std_est=True)
```

```
[22]:      lval  op      rval Estimate Est. Std Std. Err      z-value \
0      sk11  ~      curiosidad 1.000000 0.727161      -      -
1      sk10  ~      curiosidad 1.092047 0.560333 0.009842 110.961026
2      sk12  ~      curiosidad 0.897266 0.546049 0.008266 108.55233
3      sk13  ~      curiosidad 1.054243 0.516302 0.010197 103.391819
4      sk9   ~      curiosidad 0.799316 0.587033 0.006931 115.317855
5      sk4   ~      expresion 1.000000 0.700944      -      -
6      sk3   ~      expresion 0.911509 0.628559 0.007533 121.003814
7      sk1   ~      expresion 0.557056 0.584975 0.004871 114.373053
8      sk2   ~      expresion 1.060769 0.647445 0.008579 123.651137
9      sk5   ~      sociabilidad 1.000000 0.718253      -      -
10     sk6   ~      sociabilidad 1.020400 0.559841 0.010303 99.03764
11     sk8   ~      sociabilidad 1.143207 0.552164 0.011647 98.15514
12     expresion ~~      expresion 0.155601 1.000000 0.00186 83.634572
13     expresion ~~      curiosidad 0.112247 0.594400 0.001335 84.100534
14     expresion ~~      sociabilidad 0.103197 0.652552 0.001197 86.177623
15     curiosidad ~~      curiosidad 0.229183 1.000000 0.002642 86.749036
16     curiosidad ~~      sociabilidad 0.111137 0.579059 0.001378 80.670626
17     sociabilidad ~~      sociabilidad 0.160727 1.000000 0.002066 77.790427
18     sk5   ~~      sk5 0.150827 0.484113 0.001572 95.952923
19     sk11  ~~      sk11 0.204249 0.471236 0.001827 111.787721
20     sk1   ~~      sk1 0.092818 0.657805 0.000647 143.555157
21     sk13  ~~      sk13 0.700836 0.733432 0.004645 150.877939
22     sk4   ~~      sk4 0.161098 0.508678 0.001332 120.952759
23     sk6   ~~      sk6 0.366599 0.686579 0.002654 138.105704
24     sk3   ~~      sk3 0.197941 0.604914 0.001447 136.786615
25     sk2   ~~      sk2 0.242596 0.580814 0.00182 133.279071
26     sk8   ~~      sk8 0.478915 0.695115 0.003436 139.401432
27     sk12  ~~      sk12 0.434304 0.701831 0.002941 147.688201
28     sk10  ~~      sk10 0.597192 0.686027 0.004091 145.974358
29     sk9   ~~      sk9 0.278481 0.655393 0.001956 142.398651
```

```
      p-value
0      -
1      0.0
2      0.0
3      0.0
4      0.0
5      -
6      0.0
```



```

7      0.0
8      0.0
9      -
10     0.0
11     0.0
12     0.0
13     0.0
14     0.0
15     0.0
16     0.0
17     0.0
18     0.0
19     0.0
20     0.0
21     0.0
22     0.0
23     0.0
24     0.0
25     0.0
26     0.0
27     0.0
28     0.0
29     0.0

```

```
[23]: semopy.calc_stats(model)
```

```

[23]:          DoF  DoF Baseline          chi2  chi2 p-value  chi2 Baseline          CFI  \
Value    51          66  8083.209374          0.0  154026.164836  0.947829

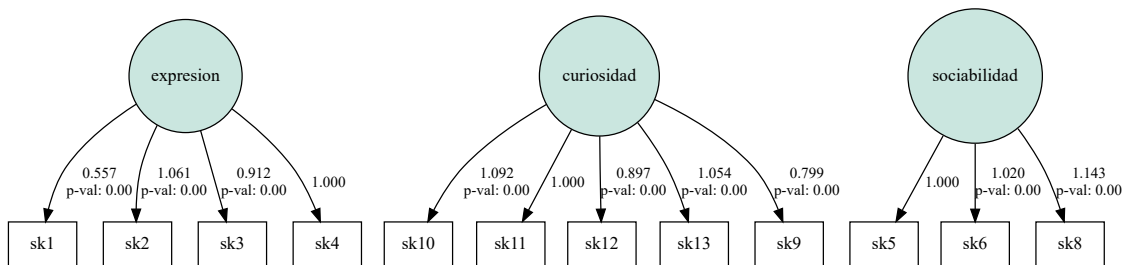
          GFI      AGFI      NFI      TLI      RMSEA      AIC  \
Value  0.947521  0.932085  0.947521  0.932485  0.052401  53.718144

          BIC      LogLik
Value  295.558499  0.140928

```

```
[24]: semopy.semplot(model, "modelo.png")
```

```
[24]:
```



### 1.0.8 Pregunta 6

```
[25]: data = juna2

mod = """
# measurement model
curiosidad =~ sk11 + sk10 + sk12 + sk13 + sk9
expresion =~ sk4 + sk3 + sk1 + sk2
sociabilidad =~ sk5 + sk6 + sk8
# regressions
curiosidad ~ sexo + edad
expresion ~ edad + vive_padre + vive_madre
sociabilidad ~ edad + imce
act_fisica ~ curiosidad + expresion + sociabilidad + sexo + madre_work + area +
↳imce
"""

model = semopy.Model(mod)
res = model.fit(data)
```

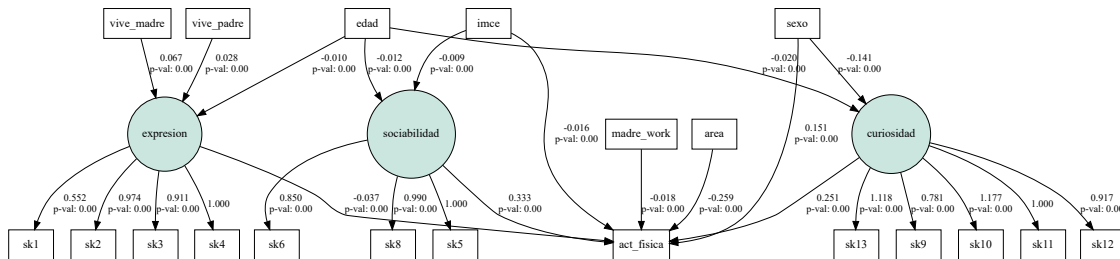
```
[26]: print(model.inspect())
```

	lval	op	rval	Estimate	Std. Err	z-value	p-value
0	curiosidad	~	sexo	-0.141442	0.00453	-31.22108	0.0
1	curiosidad	~	edad	-0.019648	0.0006	-32.749206	0.0
2	expresion	~	edad	-0.010142	0.000519	-19.5286	0.0
3	expresion	~	vive_padre	0.028458	0.00435	6.541891	0.0
4	expresion	~	vive_madre	0.067470	0.012486	5.403727	0.0
5	sociabilidad	~	edad	-0.011963	0.000581	-20.600934	0.0
6	sociabilidad	~	imce	-0.009222	0.001597	-5.774754	0.0
7	sk11	~	curiosidad	1.000000	-	-	-
8	sk10	~	curiosidad	1.176918	0.010844	108.532735	0.0
9	sk12	~	curiosidad	0.916812	0.008949	102.446603	0.0
10	sk13	~	curiosidad	1.117884	0.011073	100.959091	0.0
11	sk9	~	curiosidad	0.780574	0.007465	104.570588	0.0
12	sk4	~	expresion	1.000000	-	-	-
13	sk3	~	expresion	0.910891	0.007978	114.175797	0.0
14	sk1	~	expresion	0.551598	0.005069	108.809942	0.0
15	sk2	~	expresion	0.974372	0.008802	110.704916	0.0
16	sk5	~	sociabilidad	1.000000	-	-	-
17	sk6	~	sociabilidad	0.849839	0.011936	71.198197	0.0
18	sk8	~	sociabilidad	0.990374	0.013846	71.526277	0.0
19	act_fisica	~	curiosidad	0.251499	0.011162	22.532094	0.0
20	act_fisica	~	expresion	-0.037245	0.012601	-2.955702	0.00312
21	act_fisica	~	sociabilidad	0.333101	0.012262	27.164664	0.0
22	act_fisica	~	sexo	0.150607	0.00898	16.770623	0.0
23	act_fisica	~	madre_work	-0.017696	0.00469	-3.772853	0.000161
24	act_fisica	~	area	-0.259007	0.015216	-17.022523	0.0
25	act_fisica	~	imce	-0.015685	0.00322	-4.870769	0.000001

26	sociabilidad	~~	sociabilidad	0.192580	0.003	64.192585	0.0
27	expresion	~~	expresion	0.161266	0.001988	81.115583	0.0
28	curiosidad	~~	curiosidad	0.207824	0.002582	80.489673	0.0
29	sk5	~~	sk5	0.116761	0.002587	45.134224	0.0
30	sk12	~~	sk12	0.434557	0.003019	143.948141	0.0
31	sk11	~~	sk11	0.214154	0.001988	107.71689	0.0
32	sk1	~~	sk1	0.091524	0.000671	136.324375	0.0
33	sk13	~~	sk13	0.681461	0.00469	145.306418	0.0
34	sk4	~~	sk4	0.153543	0.001476	104.050403	0.0
35	sk6	~~	sk6	0.393249	0.002958	132.930881	0.0
36	sk3	~~	sk3	0.191983	0.001526	125.791786	0.0
37	sk2	~~	sk2	0.262896	0.001972	133.320958	0.0
38	act_fisica	~~	act_fisica	1.094130	0.006543	167.210189	0.0
39	sk8	~~	sk8	0.497840	0.003848	129.370288	0.0
40	sk9	~~	sk9	0.291303	0.002054	141.811521	0.0
41	sk10	~~	sk10	0.566876	0.004138	136.996789	0.0

```
[27]: semopy.semplot(model, "semmodelo.png")
```

[27]:



```
[28]: semopy.calc_stats(model)
```

	DoF	DoF Baseline	chi2	chi2 p-value	chi2 Baseline	CFI \
Value	171	197	43424.71842	0.0	169083.393981	0.743889

	GFI	AGFI	NFI	TLI	RMSEA	AIC \
Value	0.743176	0.704126	0.743176	0.704948	0.066409	76.485809

	BIC	LogLik
Value	425.810766	0.757095

De acuerdo con los resultados del modelo propuesto, las variables observadas incluidas junto con los tres factores explican de buena forma a la variable **act\_fisica**, debido a que todos los pesos relativos y coeficientes son distintos de cero con una significancia de 0.01 y el ajuste del modelo tiene un error cuadrático medio bajo: 0.066409

De entre las variables observadas, **area** y **sexo** son las que mayor peso relativo presentan para explicar **act\_fisica**, esto porque de pasar de una zona rural a una zona urbana disminuye en 0.25 puntos la escala de frecuencia de actividad física; lo que tiene sentido lógico si se traslada a un tema cultural

ya que por ejemplo las areas urbanas generalmente presentan mayores tasas de sedentarismo debido a las mejores comodidades que presenta su población a comparación de zonas rurales.

El factor latente con mayor incidencia sobre la actividad fisica fue ‘sociabilidad’, dado que las variables que entregan información a este factor fueron sk5, sk6 y sk8, los cuales al recodificar dichas variables para que a mayor puntaje (de 1 hasta 5) el resultado sea más ‘positivo’ y puesto que al ser mayor este puntaje ayuda a que el niño sea más íntegro en las relaciones interpersonales contribuye a que el factor latente incremente; por eso, en el modelo SEM completo el aumento en una unidad del factor sociabilidad aumenta el puntaje de frecuencia de actividad fisica en 0.33 puntos, lo que tiene sentido lógico pues al compartir y jugar con otros niños se tiende a realizar esfuerzo fisico sobre todo en los infantes.