Importation des bibliothèques

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Chargement des données

In [3]:
```
data = pd.read_excel("Exam.xlsx",index_col=0)
print(data)
data.shape
```

```
         PVP  AGR   CMI   TRA   LOG   EDU   ACS   ANC   DEF   DET  DIV
année
1872    18.0  0.5   0.1   6.7   0.5   2.1   2.0   0.0  26.4  41.5  2.1
1880    14.1  0.8   0.1  15.3   1.9   3.7   0.5   0.0  29.8  31.3  2.5
1890    13.6  0.7   0.7   6.8   0.6   7.1   0.7   0.0  33.8  34.4  1.7
1900    14.3  1.7   1.7   6.9   1.2   7.4   0.8   0.0  37.7  26.2  2.2
1903    10.3  1.5   1.4   9.3   0.6   8.5   0.9   0.0  38.4  27.2  3.0
1906    13.4  1.4   0.5   8.1   0.7   8.6   1.8   0.0  38.5  25.3  1.9
1909    13.5  1.1   0.5   9.0   1.6   9.0   3.4   0.0  36.8  23.5  2.6
1912    12.9  1.4   0.3   9.4   0.6   9.3   4.3   0.0  41.1  19.4  1.3
1920    12.3  0.3   0.1  11.9   2.4   3.7   1.7   1.9  42.4  23.1  0.2
1923     7.6  1.2   3.2   5.1   0.6   5.6   1.8  10.0  29.0  35.0  0.9
1926    10.5  0.3   0.4   4.5   1.8   6.6   2.1  10.1  19.9  41.6  2.3
1939    10.0  0.6   0.6   9.0   1.0   8.1   3.2  11.8  28.0  25.8  2.0
1932    10.6  0.8   0.3   8.9   3.0  10.0   6.4  13.4  27.4  19.2  0.0
1936     8.8  2.6   1.4   7.8   1.4  12.4   6.2  11.3  29.3  18.5  0.4
1938    10.1  1.1   1.2   5.9   1.4   9.5   6.0   5.9  40.7  18.2  0.0
1947    15.6  1.6  10.0  11.4   7.6   8.8   4.8   3.4  32.2   4.6  0.0
1950    11.2  1.3  16.5  12.4  15.8   8.1   4.9   3.4  20.7   4.2  1.5
1953    12.9  1.5   7.0   7.9  12.1   8.1   5.3   3.9  36.1   5.2  0.0
1956    10.9  5.3   9.7   7.6   9.6   9.4   8.5   4.6  28.2   6.2  0.0
1959    13.1  4.4   7.3   5.7   9.8  12.5   8.0   5.0  26.7   7.5  0.0
1962    12.8  4.7   7.5   6.6   6.8  15.7   9.7   5.3  24.5   6.4  0.1
1965    12.4  4.3   8.4   9.1   6.0  19.5  10.6   4.7  19.8   3.5  1.8
1968    11.4  6.0   9.5   5.9   5.0  21.1  10.7   4.2  20.0   4.4  1.9
1971    12.8  2.8   7.1   8.5   4.0  23.8  11.3   3.7  18.8   7.2  0.0
```

Out[3]: (24, 11)

In [4]:
```
data.index
```

Out[4]:
```
Int64Index([1872, 1880, 1890, 1900, 1903, 1906, 1909, 1912, 1920, 1923, 1926,
            1939, 1932, 1936, 1938, 1947, 1950, 1953, 1956, 1959, 1962, 1965,
            1968, 1971],
           dtype='int64', name='année')
```

Calculer et commenter brièvement les statistiques sommaires (moyenne, écart-type, min, max) les variables ?

In [7]:
```
info = data.describe()
print(info)
```

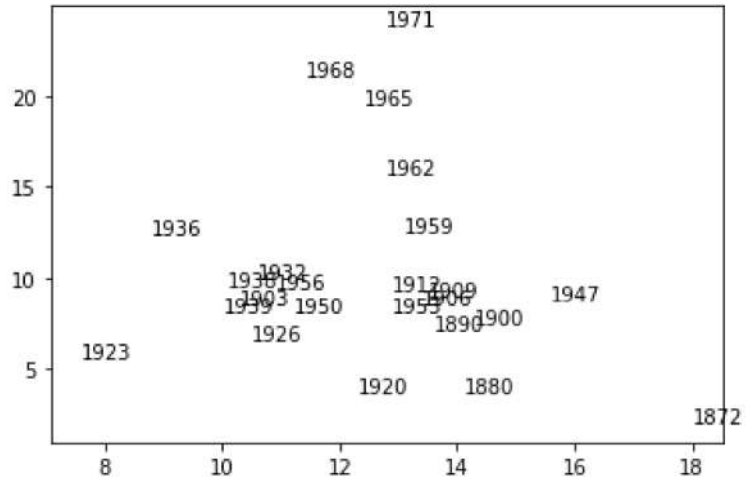
```
              PVP        AGR        CMI        TRA        LOG        EDU  \
count   24.000000  24.000000  24.000000  24.000000  24.000000  24.000000
mean    12.212500   1.995833   3.979167   8.320833   4.000000   9.941667
std      2.238267   1.681221   4.550679   2.520866   4.242436   5.335600
min      7.600000   0.300000   0.100000   4.500000   0.500000   2.100000
25%     10.575000   0.800000   0.475000   6.675000   0.925000   7.325000
50%     12.600000   1.400000   1.400000   8.000000   1.850000   8.700000
75%     13.425000   2.650000   7.350000   9.150000   6.200000  10.600000
max     18.000000   6.000000  16.500000  15.300000  15.800000  23.800000

              ACS        ANC        DEF        DET        DIV
count   24.000000  24.000000  24.000000  24.000000  24.000000
mean     4.816667   4.275000  30.258333  19.141667   1.183333
std      3.482087   4.244203   7.466733  12.455972   1.047841
min      0.500000   0.000000  18.800000   3.500000   0.000000
25%      1.800000   0.000000  25.925000   6.350000   0.000000
50%      4.550000   3.800000  29.150000  19.300000   1.400000
75%      6.800000   5.450000  37.025000  26.450000   2.025000
max     11.300000  13.400000  42.400000  41.600000   3.000000
```

Avant d'analyser les résultats proprement dits d'une A.C.P., il est bon d'en regarder les résultats préliminaires. Tout d'abord, pour chaque variable considérée, son minimum, son maximum, sa moyenne et son écart-type. Cela permet d'avoir une première connaissance des données étudiées et, le cas échéant, de décider si l'A.C.P. doit être réduite on non

Quelles sont les années qui se ressemblent ? (proximité entre les individus)

In [24]:
```
fig, ax =plt.subplots()
ax.plot(data.PVP,data.EDU,"wo")
for v in data.index:
    ax.text(data.PVP[v],data.EDU[v],v)
plt.show()
```



Sur quelles variables sont fondées les ressemblances /

dissemblances

Quelles sont les relations entre les variables ? Conclusion sur la dispersion des variables ?

In [12]:
```
mat_corr = np.corrcoef(data.values, rowvar=False)
print(mat_corr)
```

```
[[ 1.         -0.08456147 -0.0100045   0.23274025  0.04175802 -0.15003967
  -0.13140258 -0.68690117  0.10105014  0.03355634  0.1493243 ]
 [-0.08456147  1.          0.6021499  -0.27583846  0.43329087  0.73132063
   0.80568293  0.04428292 -0.44836614 -0.6949172  -0.27720187]
 [-0.0100045   0.6021499   1.          0.09623124  0.88787619  0.46825867
   0.61521586  0.0132535  -0.53029486 -0.80308323 -0.33306608]
 [ 0.23274025 -0.27583846  0.09623124  1.          0.17001655 -0.21315407
  -0.20307154 -0.31322317  0.15797549 -0.14834035  0.11436882]
 [ 0.04175802  0.43329087  0.88787619  0.17001655  1.          0.23216256
   0.48697912  0.03469904 -0.37220648 -0.75970613 -0.42711483]
 [-0.15003967  0.73132063  0.46825867 -0.21315407  0.23216256  1.
   0.87497789  0.15696654 -0.52405752 -0.67024982 -0.24864596]
 [-0.13140258  0.80568293  0.61521586 -0.20307154  0.48697912  0.87497789
   1.          0.28819417 -0.56715016 -0.80819753 -0.52959488]
 [-0.68690117  0.04428292  0.0132535  -0.31322317  0.03469904  0.15696654
   0.28819417  1.         -0.41685323 -0.0493663  -0.37746819]
 [ 0.10105014 -0.44836614 -0.53029486  0.15797549 -0.37220648 -0.52405752
  -0.56715016 -0.41685323  1.          0.26163585  0.02041298]
 [ 0.03355634 -0.6949172  -0.80308323 -0.14834035 -0.75970613 -0.67024982
  -0.80819753 -0.0493663   0.26163585  1.          0.55393211]
 [ 0.1493243  -0.27720187 -0.33306608  0.11436882 -0.42711483 -0.24864596
  -0.52959488 -0.37746819  0.02041298  0.55393211  1.        ]]
```

La matrice de corrélation va nous permettre de réaliser le résumé d'information. De cette matrice, on va extraire, à l'aide de ces vecteurs propres, les facteurs que l'on recherche, en petit nombre (2 ou 3). Ces facteurs vont permettre de réaliser les projections désirées du nuage dans cet espace de petite dimension, en déformant le moins possible la configuration globale des individus selon l'ensemble des variables initiales qui sont remplacées par les facteurs.

les corrélations linéaires sont positives (ce qui signifie que toutes les variables varient, en moyenne, dans le même sens), certaines étant très fortes (0.98 et 0.95), d'autres moyennes (0.65 et 0.51), d'autres enfin plutôt faibles (0.40 et 0.23).

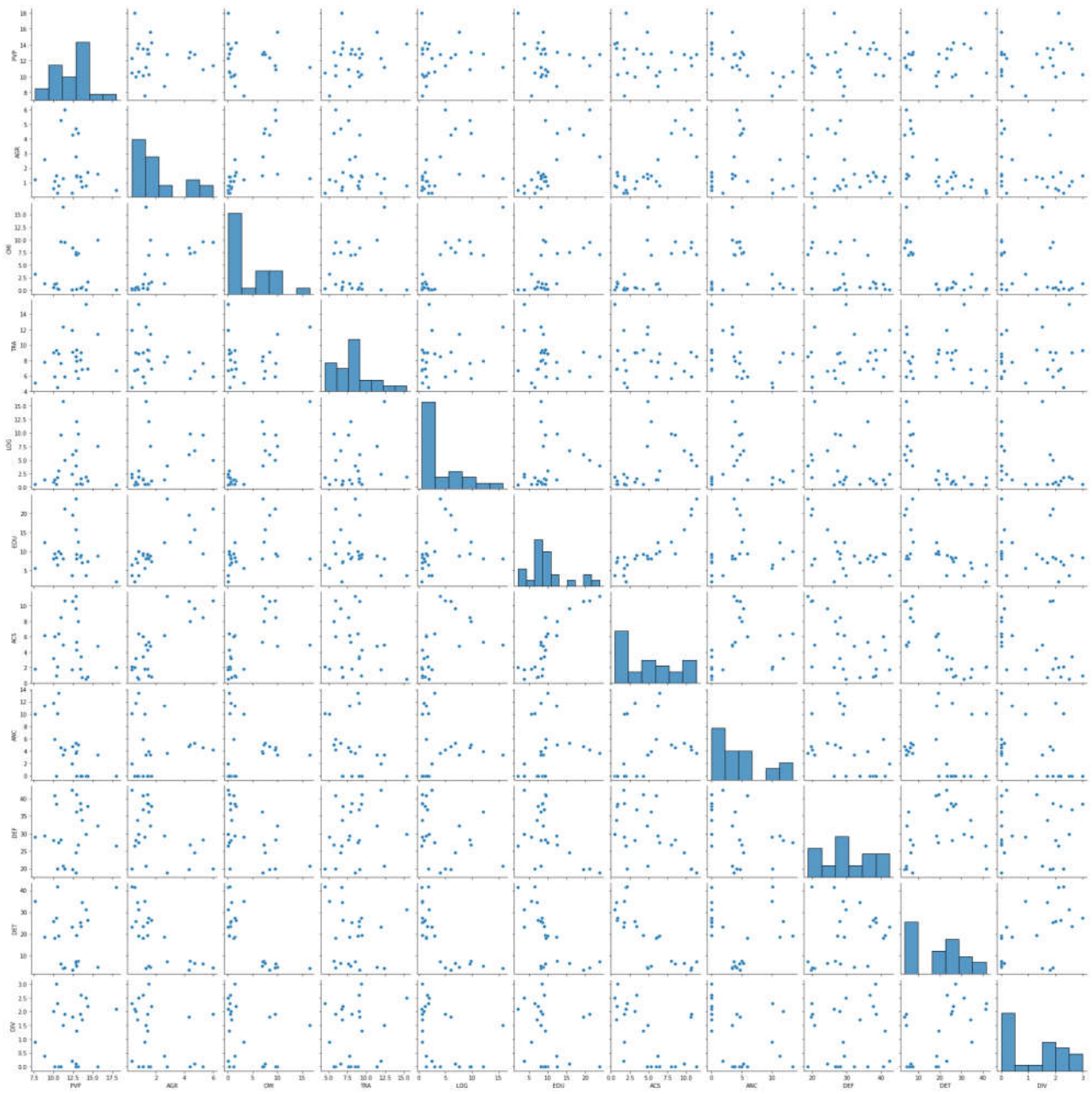
Que vaut l'inertie totale (2 Méthodes)? Interprétation

In [ ]:
```
```

La représentation graphique des individus

In [5]:
```
sns.pairplot(data)
```

```
plt.show()
```



Centrage et réduction

In [13]:

```
moyenne = np.mean(data.values, axis=0)
ecart_type = np.std(data.values, axis=0)

D = (data.values - moyenne) / ecart_type
print(D)
```

```
[[ 2.64131934 -0.90886658 -0.870771   -0.65679575 -0.84274168 -1.5012978
  -0.82629967 -1.0289201  -0.52785037  1.83359542  0.89362993]
 [ 0.8614238  -0.72658694 -0.870771    2.82810508 -0.50564501 -1.19497561
  -1.26634092 -1.0289201  -0.06270361  0.99709866  1.28357753]
 [ 0.63323207 -0.78734682 -0.73608677 -0.61627364 -0.81866335 -0.54404097
  -1.20766875 -1.0289201   0.48452787  1.25132807  0.50368232]
 [ 0.9527005  -0.17974799 -0.51161304 -0.57575154 -0.67419335 -0.48660556
  -1.17833267 -1.0289201   1.01807857  0.57885029  0.99111683]
 [-0.87283339 -0.30126775 -0.57895516  0.39677892 -0.81866335 -0.27600905
```

```
 -1.14899659 -1.0289201   1.11384408  0.66085977  1.77101204]
[ 0.54195537 -0.36202764 -0.78098151 -0.08948631 -0.79458502 -0.25686392
 -0.88497184 -1.0289201   1.12752486  0.50504175  0.69865612]
[ 0.58759372 -0.54430729 -0.78098151  0.27521261 -0.57788001 -0.18028337
 -0.41559451 -1.0289201   0.89495148  0.35742468  1.38106443]
[ 0.31376364 -0.36202764 -0.82587626  0.43730102 -0.81866335 -0.12284796
 -0.15156976 -1.0289201   1.48322533  0.02118578  0.11373472]
[ 0.03993355 -1.03038635 -0.870771    1.45035359 -0.38525334 -1.19497561
 -0.91430792 -0.57162228  1.66107556  0.32462088 -0.95862119]
[-2.10506876 -0.4835474  -0.17490245 -1.30514939 -0.81866335 -0.83121802
 -0.88497184  1.37791055 -0.1721499   1.30053376 -0.27621289]
[-0.78155669 -1.03038635 -0.80342889 -1.54828201 -0.52972334 -0.63976665
 -0.79696359  1.40197885 -1.41710152  1.84179637  1.08860373]
[-1.00974843 -0.8481067  -0.75853414  0.27521261 -0.72235002 -0.3525896
 -0.47426668  1.81114006 -0.30895777  0.54604649  0.79614303]
[-0.73591835 -0.72658694 -0.82587626  0.23469051 -0.24078334  0.011168
  0.46448798  2.19623297 -0.3910425   0.00478389 -1.153595  ]
[-1.55740859  0.36709096 -0.57895516 -0.21105262 -0.62603668  0.47065128
  0.40581582  1.69079853 -0.13110754 -0.05262275 -0.76364739]
[-0.96411008 -0.54430729 -0.6238499  -0.98097257 -0.62603668 -0.08455769
  0.34714365  0.39110998  1.42850218 -0.0772256  -1.153595  ]
[ 1.54599901 -0.24050787  1.35151891  1.24774308  0.86682002 -0.21857364
 -0.00488935 -0.21059768  0.26563528 -1.19255461 -1.153595  ]
[-0.46208826 -0.42278752  2.81059814  1.65296411  2.84124339 -0.3525896
  0.02444674 -0.21059768 -1.30765523 -1.2253584   0.30870852]
[ 0.31376364 -0.30126775  0.67809772 -0.17053052  1.95034504 -0.3525896
  0.14179107 -0.09025615  0.79918598 -1.14334891 -1.153595  ]
[-0.59900331  2.0076078   1.28417679 -0.29209682  1.3483867  -0.10370282
  1.08054573  0.078222   -0.2815962  -1.06133943 -1.153595  ]
[ 0.40504033  1.46076886  0.74543984 -1.06201677  1.39654336  0.48979641
  0.93386531  0.17449522 -0.486808   -0.9547271  -1.153595  ]
[ 0.26812529  1.64304851  0.79033459 -0.69731785  0.67419335  1.10244079
  1.43257872  0.24670014 -0.78778532 -1.04493753 -1.0561081 ]
[ 0.0855719   1.40000897  0.99236094  0.31573472  0.48156668  1.82995598
  1.69660347  0.1022903  -1.43078231 -1.28276504  0.60116922]
[-0.37081157  2.43292699  1.23928204 -0.98097257  0.24078334  2.13627817
  1.72593955 -0.01805123 -1.40342074 -1.2089565   0.69865612]
[ 0.26812529  0.48861073  0.70054509  0.0726021   0.          2.65319686
  1.90195605 -0.13839276 -1.56759018 -0.97932994 -1.153595  ]]
```

Les vecteurs et valeurs propres Interprétation

In [14]:

```
pr=np.linalg.eig(mat_corr)
print(pr)
```

```
(array([4.96126622e+00, 2.05925063e+00, 1.28441008e+00, 9.95109455e-01,
       7.01965248e-01, 5.68187830e-01, 2.05382534e-01, 1.27684939e-01,
       1.22067828e-04, 6.32792010e-02, 3.33417921e-02]), array([[-0.07811213,  0.5128916
5,  0.30518515,  0.12343131,  0.39868573,
        -0.53870405,  0.05357074,  0.31852602,  0.12931254,  0.22256511,
         0.05224375],
       [ 0.36807485,  0.00297434,  0.32045008,  0.15398522, -0.0416238 ,
         0.28662782,  0.77262646, -0.00719907,  0.09499487,  0.08775076,
         0.20123769],
       [ 0.37256132,  0.23947417, -0.12005011, -0.26298568,  0.17632764,
         0.27109337, -0.19230603, -0.27685014,  0.25841642,  0.5909088 ,
        -0.29345149],
       [-0.06102323,  0.4398959 , -0.33340227, -0.28513581, -0.64486558,
        -0.29918533,  0.27391348, -0.09493472,  0.1415265 , -0.00781884,
         0.02850784],
       [ 0.32245446,  0.28343253, -0.33399551, -0.20860806,  0.32286063,
         0.22327311, -0.08605183,  0.24289737,  0.23115688, -0.53847252,
         0.31102849],
```

```
[ 0.35400138, -0.09636943,  0.3696934 ,  0.11560134, -0.38393755,
 -0.1429112 , -0.47811768, -0.06598577,  0.29202314,  0.04507357,
  0.47672596],
[ 0.4190553 , -0.07216862,  0.14260101,  0.15297398, -0.12509738,
 -0.22907305, -0.00358361,  0.10237004,  0.19284744, -0.3875001 ,
 -0.71378254],
[ 0.1281888 , -0.56358537, -0.32952809, -0.20466024,  0.0194974 ,
 -0.2607074 ,  0.13810503,  0.51492059,  0.23827556,  0.31874241,
  0.06728976],
[-0.27423216,  0.15182101, -0.23422617,  0.6306401 , -0.18970648,
  0.36178349, -0.08230109,  0.29272849,  0.40877765,  0.12424421,
 -0.08545872],
[-0.39958805, -0.209196  ,  0.14709757, -0.17638081,  0.20459768,
 -0.07717622,  0.13709762, -0.39692895,  0.6968175 , -0.18464819,
 -0.00205585],
[-0.24430753,  0.08250209,  0.47429427, -0.50951388, -0.21644583,
  0.37024073, -0.08667076,  0.48022088,  0.0459918 , -0.01616556,
 -0.15658962]]))
```

Triage et pourcentage des valeurs propres

In [17]:

```
t=sorted(pr[0],reverse=True)
v=t/sum(t)
w=np.cumsum(v)
t=[t,v,w]
print(pd.DataFrame(np.transpose(t)))
```

```
           0         1         2
0   4.961266  0.451024  0.451024
1   2.059251  0.187205  0.638229
2   1.284410  0.116765  0.754993
3   0.995109  0.090464  0.845458
4   0.701965  0.063815  0.909273
5   0.568188  0.051653  0.960926
6   0.205383  0.018671  0.979597
7   0.127685  0.011608  0.991205
8   0.063279  0.005753  0.996958
9   0.033342  0.003031  0.999989
10  0.000122  0.000011  1.000000
```

Combien d'axes factoriels pensez-vous qu'il faille retenir ? justifier ?

Chaque ligne du tableau ci-dessus correspond à une variable virtuelle dont la colonne Valeurs Propres fournit la variance (en fait, chaque valeur propre représente la variance du facteur correspondant).

La colonne Pourcentage correspond au pourcentage de variance de chaque ligne par rapport au total. La colonne Somme Cumulé représente le cumul de ces pourcentages .

On constate alors que si on prend juste les trois premiers valeurs propres, on garde un pourcentage de presque 75% d'information total.

L'objectif (résumé pertinent des données en petite dimension) est donc atteint.

Vecteurs Propres

In [18]:
```
pr[1]
```

Out[18]:
```
array([[-0.07811213,  0.51289165,  0.30518515,  0.12343131,  0.39868573,
        -0.53870405,  0.05357074,  0.31852602,  0.12931254,  0.22256511,
         0.05224375],
       [ 0.36807485,  0.00297434,  0.32045008,  0.15398522, -0.0416238 ,
         0.28662782,  0.77262646, -0.00719907,  0.09499487,  0.08775076,
         0.20123769],
       [ 0.37256132,  0.23947417, -0.12005011, -0.26298568,  0.17632764,
         0.27109337, -0.19230603, -0.27685014,  0.25841642,  0.5909088 ,
        -0.29345149],
       [-0.06102323,  0.4398959 , -0.33340227, -0.28513581, -0.64486558,
        -0.29918533,  0.27391348, -0.09493472,  0.1415265 , -0.00781884,
         0.02850784],
       [ 0.32245446,  0.28343253, -0.33399551, -0.20860806,  0.32286063,
         0.22327311, -0.08605183,  0.24289737,  0.23115688, -0.53847252,
         0.31102849],
       [ 0.35400138, -0.09636943,  0.3696934 ,  0.11560134, -0.38393755,
        -0.1429112 , -0.47811768, -0.06598577,  0.29202314,  0.04507357,
         0.47672596],
       [ 0.4190553 , -0.07216862,  0.14260101,  0.15297398, -0.12509738,
        -0.22907305, -0.00358361,  0.10237004,  0.19284744, -0.3875001 ,
        -0.71378254],
       [ 0.1281888 , -0.56358537, -0.32952809, -0.20466024,  0.0194974 ,
        -0.2607074 ,  0.13810503,  0.51492059,  0.23827556,  0.31874241,
         0.06728976],
       [-0.27423216,  0.15182101, -0.23422617,  0.6306401 , -0.18970648,
         0.36178349, -0.08230109,  0.29272849,  0.40877765,  0.12424421,
        -0.08545872],
       [-0.39958805, -0.209196  ,  0.14709757, -0.17638081,  0.20459768,
        -0.07717622,  0.13709762, -0.39692895,  0.6968175 , -0.18464819,
        -0.00205585],
       [-0.24430753,  0.08250209,  0.47429427, -0.50951388, -0.21644583,
         0.37024073, -0.08667076,  0.48022088,  0.0459918 , -0.01616556,
        -0.15658962]])
```

Coordonnées factorielles des individus

In [19]:
```
F = np.dot(D,pr[1])
print(F)
```

```
[[-2.91280420e+00  1.00990121e+00  1.60323472e+00 -4.22838373e-01
   2.03037746e+00 -1.24065099e+00  2.95383581e-01 -2.19223923e-02
   2.51457993e-03 -4.15424096e-02 -3.63955741e-01]
 [-2.77517615e+00  2.00610696e+00 -1.52565367e-01 -1.46816108e+00
  -1.23163578e+00 -7.62694943e-01  9.34778913e-01 -2.48943337e-01
  -9.42180220e-03 -2.40186312e-01  1.44963674e-01]
 [-2.42638730e+00  2.15844307e-01  7.83398673e-01  2.88167224e-01
   6.83513493e-01  1.23092226e-01 -3.20839754e-01 -4.59707371e-01
  -3.34287020e-03  1.90226361e-02  2.29015599e-01]
 [-2.06401831e+00  7.48278942e-01  1.01960306e+00  5.38298846e-01
   4.75653659e-01  6.16602969e-01 -8.46979107e-02  2.63177168e-01
  -1.28177767e-02  3.71779779e-01  2.34095085e-01]
 [-2.25989272e+00  2.21668315e-01  5.97201037e-01 -2.57802942e-01
```

```
 -1.18749115e+00  1.50396487e+00 -1.49638538e-01 -6.70925813e-02
  2.57330057e-02 -3.25152341e-02  6.58012492e-02]
[-1.99268433e+00  6.16961967e-01  6.97827116e-01  7.19322196e-01
 -1.77926882e-01  3.77216510e-01 -1.56853765e-01  6.85855654e-02
 -8.34599967e-03  9.48528395e-02  1.68141276e-01]
[-1.83586709e+00  8.72295294e-01  9.11041797e-01  1.60017440e-01
 -5.39266513e-01  3.01097115e-01 -3.12533211e-01  4.63665600e-01
  2.82811330e-02 -2.21628233e-01 -1.73562294e-01]
[-1.43801914e+00  7.55202838e-01  1.88201393e-01  1.29315118e+00
 -8.07371947e-01  8.73025379e-02 -1.25696575e-01  3.39945828e-02
 -1.85551124e-02 -1.08777095e-01 -2.20488420e-01]
[-2.15000688e+00  9.46190902e-01 -1.74811760e+00  1.05978131e+00
 -6.33579873e-01 -3.22630279e-01  2.90465617e-01 -3.82098794e-01
 -1.89225473e-03 -1.19800024e-01  2.37506313e-02]
[-1.15650894e+00 -2.88275033e+00 -8.54754557e-01 -4.56059716e-01
  5.42596229e-01  8.53057625e-01  6.75082406e-02 -7.19050906e-01
  7.86204062e-04  3.24329121e-01 -6.43702922e-02]
[-1.68789847e+00 -2.60913801e+00  5.27878603e-01 -1.74786653e+00
  1.19973363e+00 -9.04585389e-02 -2.84834903e-01  5.86384260e-02
 -4.16258286e-04 -2.48662277e-01  8.37434694e-02]
[-1.18497750e+00 -1.83371559e+00 -6.03877128e-01 -1.16430963e+00
 -6.83691201e-01 -3.20689844e-01  2.57453937e-02  6.75341173e-01
 -7.24214817e-03  2.32522776e-01 -2.47381711e-02]
[ 2.57681881e-01 -1.97029568e+00 -1.46430814e+00 -3.90288657e-02
 -3.32585253e-01 -1.40920549e+00  7.21504346e-02  4.26049401e-01
  6.52414141e-03 -9.80184065e-02  2.51046586e-02]
[ 6.49041990e-01 -2.30606696e+00 -6.73225793e-01  2.93462336e-01
 -8.23411397e-01 -2.16514592e-01  3.84379209e-01  2.64427238e-02
  2.47931803e-03  9.08933337e-02  1.41043616e-01]
[-4.12841790e-01 -1.35393393e+00 -8.61345113e-01  1.83710249e+00
 -8.16916615e-02  2.74208625e-01 -5.02027238e-01 -5.20537729e-02
 -8.62066859e-03 -8.94803995e-02 -4.02571250e-01]
[ 1.07670239e+00  2.24516967e+00 -1.26990792e+00  2.44480314e-01
  3.75652600e-01 -8.66860714e-01 -1.05664133e-01  1.14717523e-01
  1.28700481e-02  8.41861946e-01 -1.35485216e-02]
[ 2.37415073e+00  2.20142387e+00 -1.89898946e+00 -2.68505233e+00
  2.39096717e-01  8.64981503e-01 -6.31352401e-01 -2.19883069e-01
 -5.52400823e-03 -5.56840633e-02 -1.36946336e-01]
[ 1.19910820e+00  1.14014940e+00 -1.65964837e+00  7.48612691e-01
  9.76887735e-01  4.06632882e-01 -5.27837505e-01  5.29400925e-01
 -6.31082606e-03 -3.75328533e-01  1.97852619e-01]
[ 2.92606984e+00  2.31988409e-01 -5.93938179e-01  4.34173951e-01
  5.19892664e-01  9.34531928e-01  1.11021079e+00 -2.62924610e-01
  4.46481334e-03 -1.40919047e-01 -2.01632945e-01]
[ 2.68281113e+00  1.36952494e-01  7.30504541e-02  7.03313798e-01
  1.21310301e+00  1.73176649e-01  3.91627285e-01  1.87770825e-01
  3.67400633e-03 -2.34601047e-01  3.03330522e-01]
[ 3.05203358e+00 -1.20790157e-01  5.81236708e-01  6.58142512e-01
  4.11788541e-01 -2.45496280e-01  3.97771114e-01 -3.73360075e-02
  1.04019261e-02 -1.37165252e-03  5.70003789e-02]
[ 3.14431941e+00  3.05361658e-01  1.40582608e+00 -7.57530582e-01
 -9.31281989e-01 -2.35628552e-01 -3.66249151e-02  3.14071482e-01
 -8.62386032e-03 -2.64586427e-02 -1.47184951e-01]
[ 3.69813275e+00 -4.72230216e-01  2.28841639e+00 -2.80684165e-01
 -4.89043139e-01  7.28597411e-01  1.91270768e-01  1.03874089e-01
 -1.13511741e-02  2.00583795e-01 -4.82433305e-02]
[ 3.23703092e+00 -1.04575365e-01  1.10376159e+00  3.01307930e-01
 -7.49318964e-01 -1.53363263e+00 -9.22690499e-01 -7.94716644e-01
  4.73558353e-03 -1.40872852e-01  1.23399475e-01]]
```

In [20]:

```
print(pd.DataFrame(F, index=data.index))
```

```
              0         1         2         3         4         5         6   \
année
```

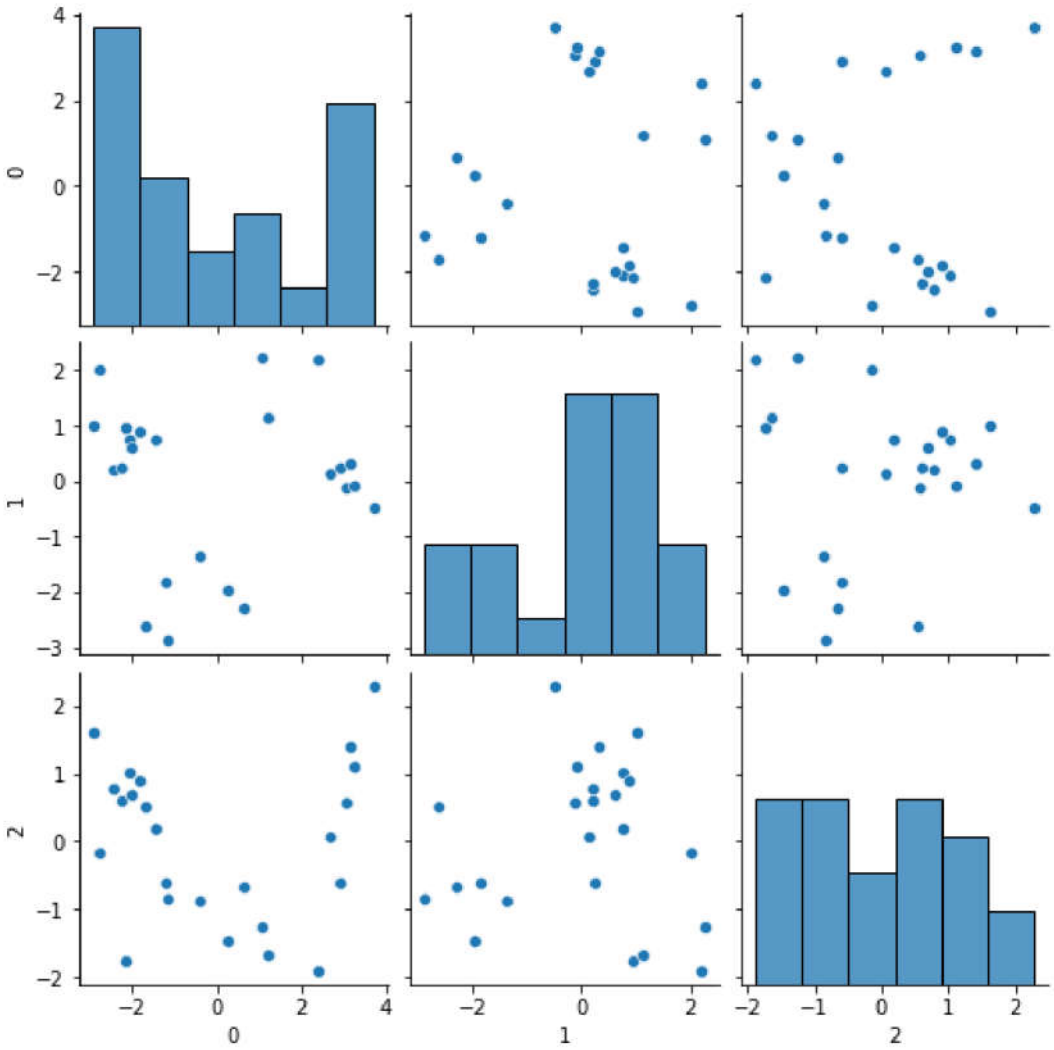
année							
1872	-2.912804	1.009901	1.603235	-0.422838	2.030377	-1.240651	0.295384
1880	-2.775176	2.006107	-0.152565	-1.468161	-1.231636	-0.762695	0.934779
1890	-2.426387	0.215844	0.783399	0.288167	0.683513	0.123092	-0.320840
1900	-2.064018	0.748279	1.019603	0.538299	0.475654	0.616603	-0.084698
1903	-2.259893	0.221668	0.597201	-0.257803	-1.187491	1.503965	-0.149639
1906	-1.992684	0.616962	0.697827	0.719322	-0.177927	0.377217	-0.156854
1909	-1.835867	0.872295	0.911042	0.160017	-0.539267	0.301097	-0.312533
1912	-1.438019	0.755203	0.188201	1.293151	-0.807372	0.087303	-0.125697
1920	-2.150007	0.946191	-1.748118	1.059781	-0.633580	-0.322630	0.290466
1923	-1.156509	-2.882750	-0.854755	-0.456060	0.542596	0.853058	0.067508
1926	-1.687898	-2.609138	0.527879	-1.747867	1.199734	-0.090459	-0.284835
1939	-1.184978	-1.833716	-0.603877	-1.164310	-0.683691	-0.320690	0.025745
1932	0.257682	-1.970296	-1.464308	-0.039029	-0.332585	-1.409205	0.072150
1936	0.649042	-2.306067	-0.673226	0.293462	-0.823411	-0.216515	0.384379
1938	-0.412842	-1.353934	-0.861345	1.837102	-0.081692	0.274209	-0.502027
1947	1.076702	2.245170	-1.269908	0.244480	0.375653	-0.866861	-0.105664
1950	2.374151	2.201424	-1.898989	-2.685052	0.239097	0.864982	-0.631352
1953	1.199108	1.140149	-1.659648	0.748613	0.976888	0.406633	-0.527838
1956	2.926070	0.231988	-0.593938	0.434174	0.519893	0.934532	1.110211
1959	2.682811	0.136952	0.073050	0.703314	1.213103	0.173177	0.391627
1962	3.052034	-0.120790	0.581237	0.658143	0.411789	-0.245496	0.397771
1965	3.144319	0.305362	1.405826	-0.757531	-0.931282	-0.235629	-0.036625
1968	3.698133	-0.472230	2.288416	-0.280684	-0.489043	0.728597	0.191271
1971	3.237031	-0.104575	1.103762	0.301308	-0.749319	-1.533633	-0.922690

année	7	8	9	10
1872	-0.021922	0.002515	-0.041542	-0.363956
1880	-0.248943	-0.009422	-0.240186	0.144964
1890	-0.459707	-0.003343	0.019023	0.229016
1900	0.263177	-0.012818	0.371780	0.234095
1903	-0.067093	0.025733	-0.032515	0.065801
1906	0.068586	-0.008346	0.094853	0.168141
1909	0.463666	0.028281	-0.221628	-0.173562
1912	0.033995	-0.018555	-0.108777	-0.220488
1920	-0.382099	-0.001892	-0.119800	0.023751
1923	-0.719051	0.000786	0.324329	-0.064370
1926	0.058638	-0.000416	-0.248662	0.083743
1939	0.675341	-0.007242	0.232523	-0.024738
1932	0.426049	0.006524	-0.098018	0.025105
1936	0.026443	0.002479	0.090893	0.141044
1938	-0.052054	-0.008621	-0.089480	-0.402571
1947	0.114718	0.012870	0.841862	-0.013549
1950	-0.219883	-0.005524	-0.055684	-0.136946
1953	0.529401	-0.006311	-0.375329	0.197853
1956	-0.262925	0.004465	-0.140919	-0.201633
1959	0.187771	0.003674	-0.234601	0.303331
1962	-0.037336	0.010402	-0.001372	0.057000
1965	0.314071	-0.008624	-0.026459	-0.147185
1968	0.103874	-0.011351	0.200584	-0.048243
1971	-0.794717	0.004736	-0.140873	0.123399

La représentation graphique des individus

In [23]:
```
sns.pairplot(pd.DataFrame(F[:,:3]))
```

Out[23]: <seaborn.axisgrid.PairGrid at 0x1fb04e4cec8>



Interprétation

Par conséquent, les facteurs 0, 1 et 2 présentés ci-dessous résument presque parfaitement la configuration réelle des données qui se trouvent en dimension 4

Utilisation direct de l'ACP

In [21]:
```
from sklearn.decomposition import PCA
acp = PCA()
#coordonnées factorielles
coord = acp.fit_transform(D)
#afficher les nouvelles coordonnées
print(pd.DataFrame(coord,index=data.index))
```

```
              0         1         2         3         4         5         6   \
année
1872  -2.912804 -1.009901  1.603235  0.422838  2.030377  1.240651  0.295384
1880  -2.775176 -2.006107 -0.152565  1.468161 -1.231636  0.762695  0.934779
1890  -2.426387 -0.215844  0.783399 -0.288167  0.683513 -0.123092 -0.320840
1900  -2.064018 -0.748279  1.019603 -0.538299  0.475654 -0.616603 -0.084698
1903  -2.259893 -0.221668  0.597201  0.257803 -1.187491 -1.503965 -0.149639
1906  -1.992684 -0.616962  0.697827 -0.719322 -0.177927 -0.377217 -0.156854
1909  -1.835867 -0.872295  0.911042 -0.160017 -0.539267 -0.301097 -0.312533
```

année							
1912	-1.438019	-0.755203	0.188201	-1.293151	-0.807372	-0.087303	-0.125697
1920	-2.150007	-0.946191	-1.748118	-1.059781	-0.633580	0.322630	0.290466
1923	-1.156509	2.882750	-0.854755	0.456060	0.542596	-0.853058	0.067508
1926	-1.687898	2.609138	0.527879	1.747867	1.199734	0.090459	-0.284835
1939	-1.184978	1.833716	-0.603877	1.164310	-0.683691	0.320690	0.025745
1932	0.257682	1.970296	-1.464308	0.039029	-0.332585	1.409205	0.072150
1936	0.649042	2.306067	-0.673226	-0.293462	-0.823411	0.216515	0.384379
1938	-0.412842	1.353934	-0.861345	-1.837102	-0.081692	-0.274209	-0.502027
1947	1.076702	-2.245170	-1.269908	-0.244480	0.375653	0.866861	-0.105664
1950	2.374151	-2.201424	-1.898989	2.685052	0.239097	-0.864982	-0.631352
1953	1.199108	-1.140149	-1.659648	-0.748613	0.976888	-0.406633	-0.527838
1956	2.926070	-0.231988	-0.593938	-0.434174	0.519893	-0.934532	1.110211
1959	2.682811	-0.136952	0.073050	-0.703314	1.213103	-0.173177	0.391627
1962	3.052034	0.120790	0.581237	-0.658143	0.411789	0.245496	0.397771
1965	3.144319	-0.305362	1.405826	0.757531	-0.931282	0.235629	-0.036625
1968	3.698133	0.472230	2.288416	0.280684	-0.489043	-0.728597	0.191271
1971	3.237031	0.104575	1.103762	-0.301308	-0.749319	1.533633	-0.922690

année	7	8	9	10
1872	0.021922	-0.041542	0.363956	0.002515
1880	0.248943	-0.240186	-0.144964	-0.009422
1890	0.459707	0.019023	-0.229016	-0.003343
1900	-0.263177	0.371780	-0.234095	-0.012818
1903	0.067093	-0.032515	-0.065801	0.025733
1906	-0.068586	0.094853	-0.168141	-0.008346
1909	-0.463666	-0.221628	0.173562	0.028281
1912	-0.033995	-0.108777	0.220488	-0.018555
1920	0.382099	-0.119800	-0.023751	-0.001892
1923	0.719051	0.324329	0.064370	0.000786
1926	-0.058638	-0.248662	-0.083743	-0.000416
1939	-0.675341	0.232523	0.024738	-0.007242
1932	-0.426049	-0.098018	-0.025105	0.006524
1936	-0.026443	0.090893	-0.141044	0.002479
1938	0.052054	-0.089480	0.402571	-0.008621
1947	-0.114718	0.841862	0.013549	0.012870
1950	0.219883	-0.055684	0.136946	-0.005524
1953	-0.529401	-0.375329	-0.197853	-0.006311
1956	0.262925	-0.140919	0.201633	0.004465
1959	-0.187771	-0.234601	-0.303331	0.003674
1962	0.037336	-0.001372	-0.057000	0.010402
1965	-0.314071	-0.026459	0.147185	-0.008624
1968	-0.103874	0.200584	0.048243	-0.011351
1971	0.794717	-0.140873	-0.123399	0.004736

In [ ]: