```
!pip install rasterio
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/pub
Collecting rasterio
  Downloading rasterio-1.2.10-cp37-cp37m-manylinux1_x86_64.whl (19.3 MB)
     |████████████████████████████████| 19.3 MB 7.5 MB/s
Collecting click-plugins
  Downloading click_plugins-1.1.1-py2.py3-none-any.whl (7.5 kB)
Collecting cligj>=0.5
  Downloading cligj-0.7.2-py3-none-any.whl (7.1 kB)
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages (fro
Requirement already satisfied: certifi in /usr/local/lib/python3.7/dist-packages (from r
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from ras
Collecting affine
  Downloading affine-2.3.1-py2.py3-none-any.whl (16 kB)
Requirement already satisfied: attrs in /usr/local/lib/python3.7/dist-packages (from ras
Requirement already satisfied: click>=4.0 in /usr/local/lib/python3.7/dist-packages (fro
Collecting snuggs>=1.4.1
  Downloading snuggs-1.4.7-py3-none-any.whl (5.4 kB)
Requirement already satisfied: pyparsing>=2.1.6 in /usr/local/lib/python3.7/dist-package
Installing collected packages: snuggs, cligj, click-plugins, affine, rasterio
Successfully installed affine-2.3.1 click-plugins-1.1.1 cligj-0.7.2 rasterio-1.2.10 snug
```

```
import rasterio as rio
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
from pandas import DataFrame
import statsmodels.graphics.api as smg


from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

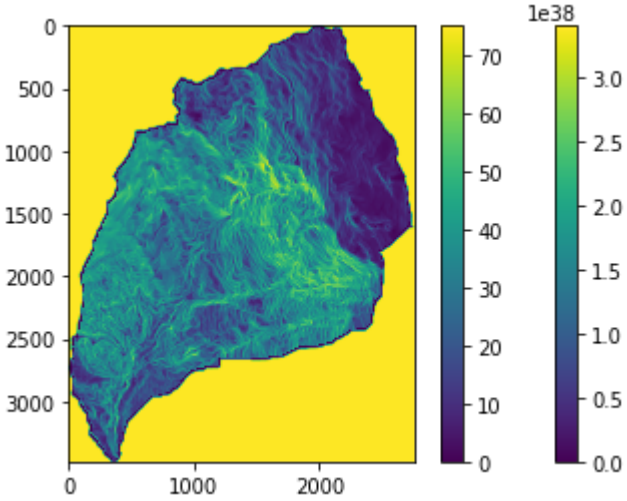
```
raster = rio.open('/content/drive/MyDrive/4. UNIVERSIDAD NACIONAL/Cartografia geotecnica/Cart
pendiente=raster.read(1)
plt.imshow(pendiente)
plt.colorbar();

raster_mask = rio.open('/content/drive/MyDrive/4. UNIVERSIDAD NACIONAL/Cartografia geotecnica
msk=raster_mask.read_masks(1)
msk=np.where(msk==255,1,np.nan)
pendiente=msk*pendiente

pendiente=np.where(pendiente<0,np.nan,pendiente)
plt.imshow(pendiente)
plt.colorbar();
```

```
pendiente_vector=pendiente.ravel()  # para pasarlo a un vector
pendiente_vector_MenM=pendiente_vector[~np.isnan(pendiente_vector)] # para eliminar  del vect
pendiente_vector_MenM.shape # otra forma de saber las dimensiones
```

```
(5554845,)
```

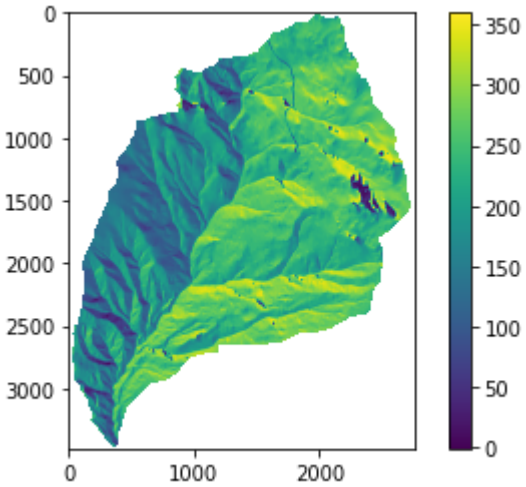



```
raster = rio.open('/content/drive/MyDrive/4. UNIVERSIDAD NACIONAL/Cartografia geotecnica/Cart
aspecto=raster.read(1)

raster_mask = rio.open('/content/drive/MyDrive/4. UNIVERSIDAD NACIONAL/Cartografia geotecnica
msk=raster_mask.read_masks(1)
msk=np.where(msk==255,1,np.nan)
aspecto=msk*aspecto

aspecto=np.where(aspecto<-100,np.nan,aspecto)
aspecto_vector=aspecto.ravel()
aspecto_vector_MenM=aspecto_vector[~np.isnan(aspecto_vector)]
plt.imshow(aspecto)
plt.colorbar()
aspecto_vector_MenM.shape
```

```
(5554845,)
```

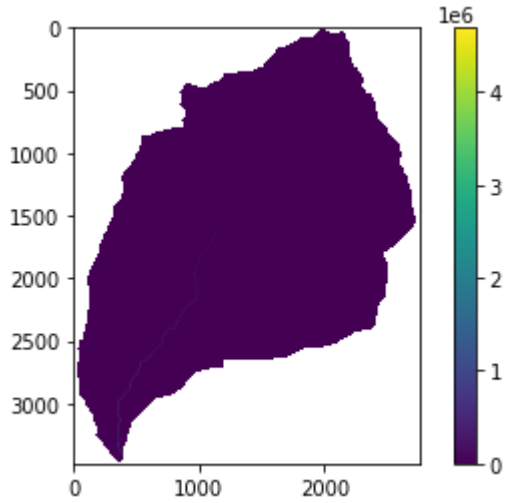


```
raster = rio.open('/content/drive/MyDrive/4. UNIVERSIDAD NACIONAL/Cartografia geotecnica/Cart
flujo=raster.read(1)

raster_mask = rio.open('/content/drive/MyDrive/4. UNIVERSIDAD NACIONAL/Cartografia geotecnica
msk=raster_mask.read_masks(1)
msk=np.where(msk==255,1,np.nan)
flujo=msk*flujo

flujo=np.where(flujo<0,np.nan,flujo)
flujo_vector=flujo.ravel()
flujo_vector_MenM=flujo_vector[~np.isnan(flujo_vector)]
plt.imshow(flujo)
plt.colorbar()
flujo.shape
flujo_vector_MenM.shape
```

```
(5554845,)
```

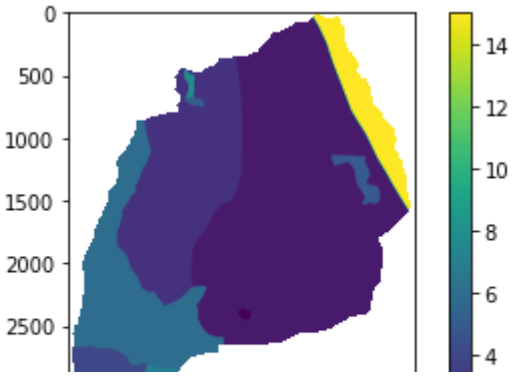



```
raster = rio.open('/content/drive/MyDrive/4. UNIVERSIDAD NACIONAL/Cartografia geotecnica/Cart
geologia=raster.read(1)

raster_mask = rio.open('/content/drive/MyDrive/4. UNIVERSIDAD NACIONAL/Cartografia geotecnica
msk=raster_mask.read_masks(1)
msk=np.where(msk==255,1,np.nan)
geologia=msk*geologia

geologia=np.where(geologia<0,np.nan,geologia)
geologia_vector=geologia.ravel()
geologia_vector_MenM=geologia_vector[~np.isnan(geologia_vector)]
plt.imshow(geologia)
plt.colorbar()
geologia_vector_MenM.shape
```

```
(5554845,)
```




```
np.shape(pendiente)
```

```
(3482, 2763)
```

```
np.shape(aspecto)
```

```
(3482, 2763)
```

```
np.shape(flujo)
```

```
(3482, 2763)
```

```
np.shape(geologia)
```

```
(3482, 2763)
```

```
raster = rio.open('/content/drive/MyDrive/4. UNIVERSIDAD NACIONAL/Cartografia geotecnica/Cart
inventario=raster.read(1)
raster_mask = rio.open('/content/drive/MyDrive/4. UNIVERSIDAD NACIONAL/Cartografia geotecnica
msk=raster_mask.read_masks(1)
msk=np.where(msk==255,1,np.nan)
inventario=msk*inventario
inventario_vector=inventario.ravel()
inventario_vector_MenM=inventario_vector[~np.isnan(inventario_vector)]
plt.imshow(inventario)
plt.colorbar()
inventario_vector_MenM.shape
```

```
(5554845,)
```


```
np.shape(inventario)
```

```
(3482, 2763)
```


```
d={'inventario':inventario_vector_MenM,'pendiente':pendiente_vector_MenM,'flujo_acum':flujo_v
df = pd.DataFrame(d)
print(list(df.columns))
```

```
['inventario', 'pendiente', 'flujo_acum', 'aspecto', 'geologia']
```

```
df.head()
```

	inventario	pendiente	flujo_acum	aspecto	geologia
0	-2.147484e+09	10.148775	1.0	218.836258	15.0
1	-2.147484e+09	9.004342	1.0	218.494370	15.0
2	-2.147484e+09	9.214328	1.0	220.666763	15.0
3	-2.147484e+09	9.395398	1.0	222.757767	15.0
4	-2.147484e+09	9.551983	1.0	224.808960	15.0

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5554845 entries, 0 to 5554844
Data columns (total 5 columns):
 #   Column      Dtype  
---  ------      -----  
 0   inventario  float64
 1   pendiente   float64
 2   flujo_acum  float64
 3   aspecto     float64
 4   geologia    float64
dtypes: float64(5)
memory usage: 211.9 MB
```

```
df1=df[(df["inventario"]==1) | (df["inventario"]==0).sample(frac=.1)]
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 32 entries, 799648 to 5343495
Data columns (total 5 columns):
```

```
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   inventario  32 non-null     float64
 1   pendiente   32 non-null     float64
 2   flujo_acum  32 non-null     float64
 3   aspecto     32 non-null     float64
 4   geologia    32 non-null     float64
dtypes: float64(5)
memory usage: 1.5 KB
```

```
resumen=df1.describe().T
print(resumen)
```

```
             count        mean        std        min         25%         50%  \
inventario    32.0    1.000000   0.000000   1.000000    1.000000    1.000000   
pendiente     32.0   30.881745   9.989842  10.981749   24.061012   30.022455   
flujo_acum    32.0   49.937500  68.161070   0.000000    7.500000   26.500000   
aspecto       32.0  177.716631  57.809234  85.605873  129.389715  174.594040   
geologia      32.0    3.968750   1.674946   2.000000    3.000000    3.000000   

                    75%         max  
inventario     1.000000    1.000000  
pendiente     38.332056   49.843197  
flujo_acum    60.250000  330.000000  
aspecto      224.301723  270.946014  
geologia       6.000000    6.000000  
```

```
matriz=df.drop(['inventario'],axis=1) # función para eliminar una columna (axis=1)
matriz.head()
```

	pendiente	flujo_acum	aspecto	geologia
0	10.148775	1.0	218.836258	15.0
1	9.004342	1.0	218.494370	15.0
2	9.214328	1.0	220.666763	15.0
3	9.395398	1.0	222.757767	15.0
4	9.551983	1.0	224.808960	15.0

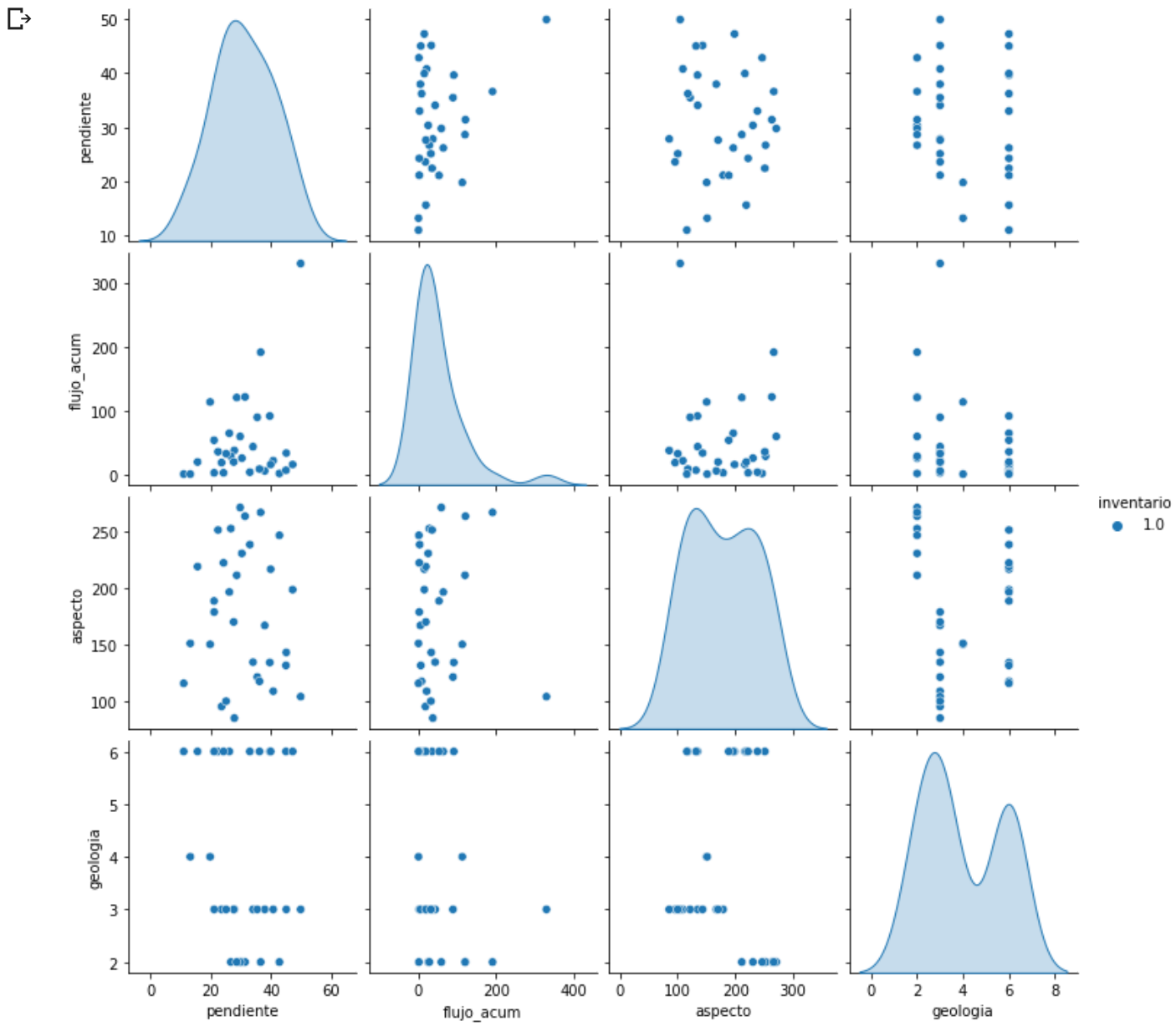
```
matriz_cont=matriz.drop(['geologia'],axis=1)
matriz_cont.head()
```

	pendiente	flujo_acum	aspecto
0	10.148775	1.0	218.836258

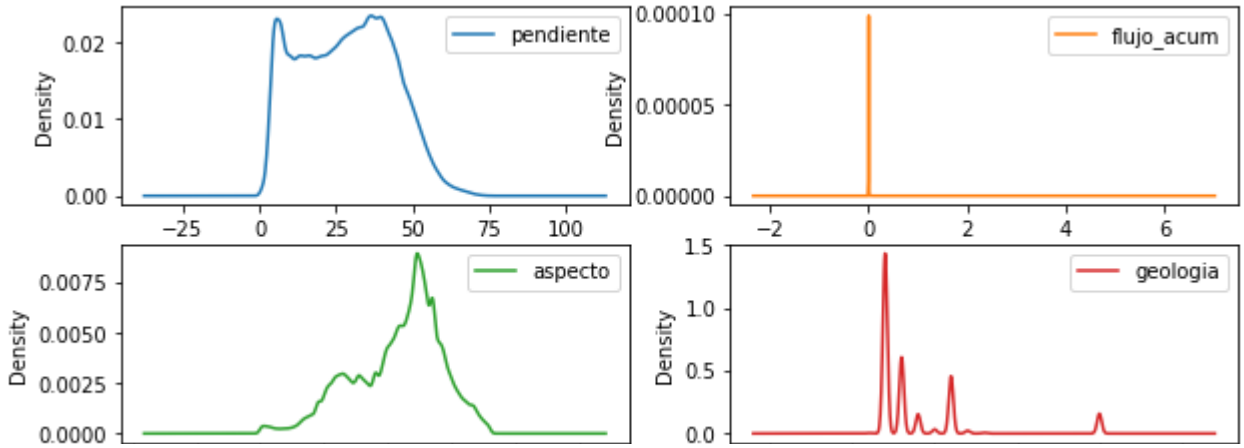
```
pd.plotting.scatter_matrix(matriz_cont, alpha = 0.3, figsize = (14,10), diagonal='kde');
```

2	9.214328	1.0	220.666763

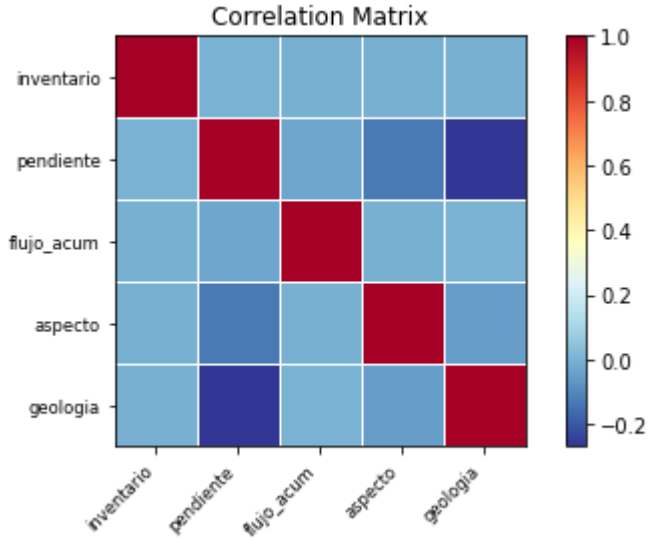
```
sns.pairplot(df1, hue='inventario');
```




```
matriz.plot(kind='density', subplots=True, layout=(2, 2), sharex=False, figsize=(10, 4));
```

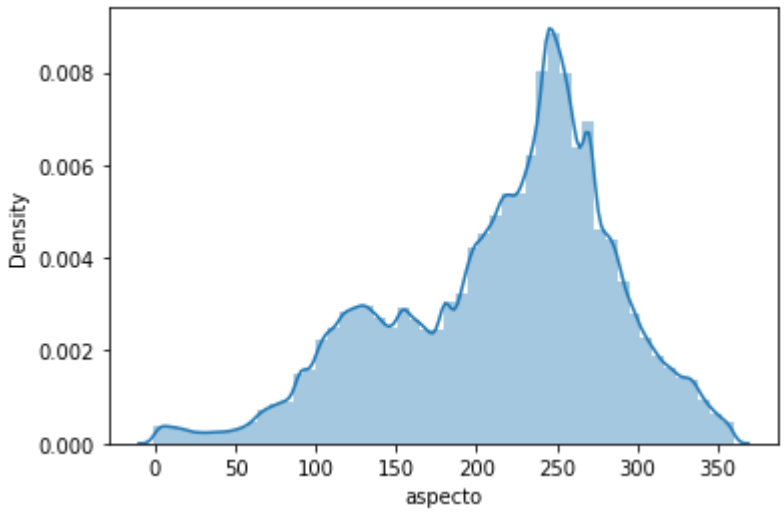



```
MatCorre=DataFrame(df.corr())
smg.plot_corr(MatCorre, xnames=list(MatCorre.columns)) ;
```

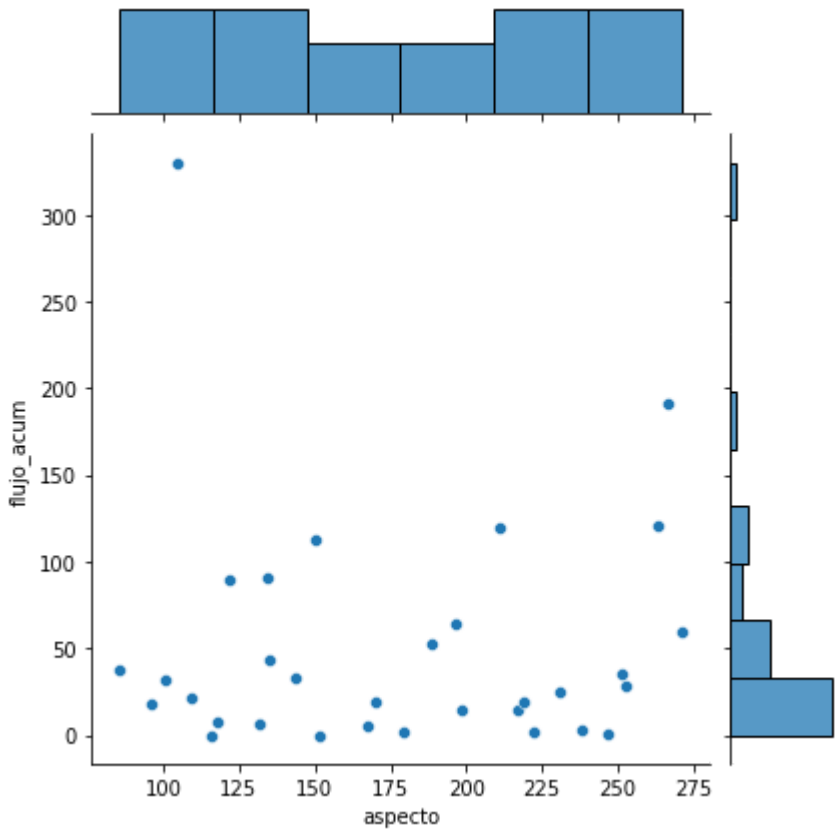



```
sns.distplot(df['aspecto']);
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarnin
  warnings.warn(msg, FutureWarning)
```

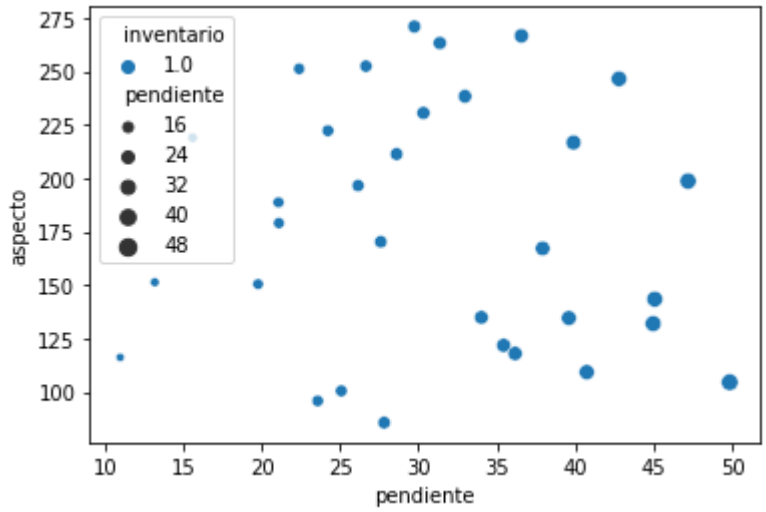


```
sns.jointplot(x='aspecto', y='flujo_acum', data=df1, kind='scatter');
```



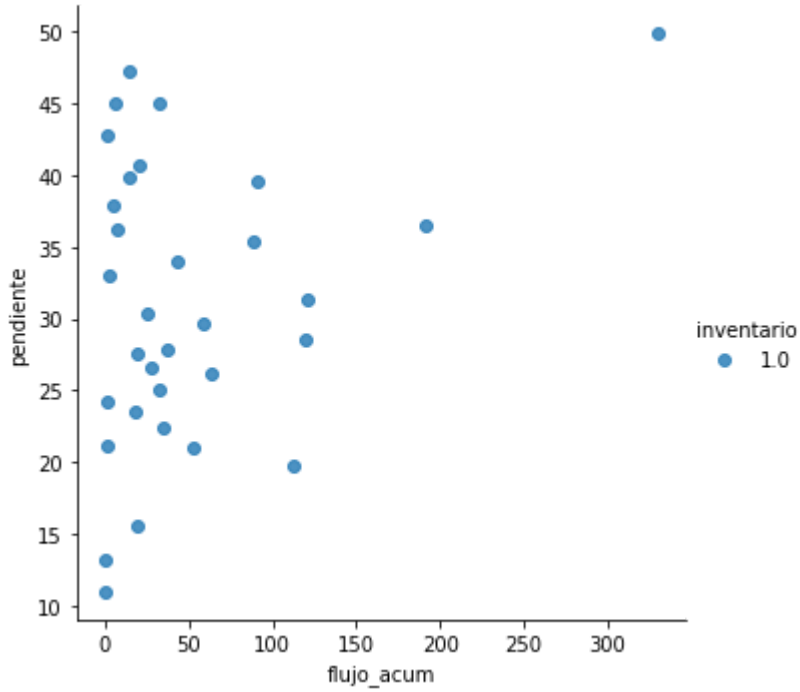

```
sns.jointplot(x='pendiente', y='flujo_acum', data=df1, kind='kde', color='g');
```

```
sns.scatterplot(x="pendiente", y="aspecto", hue="inventario", size='pendiente',data=df1);
```




```
sns.lmplot('flujo_acum', 'pendiente', data=df1, hue='inventario', fit_reg=False);
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: P
  FutureWarning
```




```
media=df.groupby('inventario').mean()
print(media)
```

```
                pendiente   flujo_acum     aspecto  geologia
inventario
-2.147484e+09  28.300629  1844.058365  216.588516  3.749118
 1.000000e+00  30.881745    49.937500  177.716631  3.968750
```

```
#Para contar el numero de celdas con y sin MenM
df['inventario'].value_counts()
```

```
     -2.147484e+09    5554813
      1.000000e+00         32
     Name: inventario, dtype: int64
```

```
landslides=df.inventario.astype(bool)
si_lands=df[landslides]
no_lands=df[~landslides]
```

```
si_lands.count()
```

```
     inventario    5554845
     pendiente     5554845
     flujo_acum    5554845
     aspecto       5554845
     geologia      5554845
     dtype: int64
```

```
no_lands.count()
```

```
     inventario    0
     pendiente     0
     flujo_acum    0
     aspecto       0
     geologia      0
     dtype: int64
```

```
df.boxplot('pendiente', by='inventario', notch=True, widths=0.8, showmeans=True, meanline=Tru
plt.plot([], [], '--', linewidth=1, color='red', label='mean')
plt.plot([], [], '-', linewidth=1, color='blue', label='median')
plt.legend();
```

```
/usr/local/lib/python3.7/dist-packages/matplotlib/cbook/__init__.py:1376: VisibleD
  X = np.atleast_1d(X.T if isinstance(X, np.ndarray) else np.asarray(X))
```


```
from scipy import stats
stats.ttest_ind(no_lands["pendiente"], si_lands["pendiente"])
```

```
Ttest_indResult(statistic=nan, pvalue=nan)
```


```
fig, ax = plt.subplots()
si_lands['pendiente'].plot.kde(ax=ax, label='Sin MenM')
no_lands['pendiente'].plot.kde(ax=ax, label='Con MenM')
ax.set_xlim(0,90)
ax.set_xlabel('Pendiente (°)', color='k', size=12)
ax.set_ylabel('Densidad', color='k', size=12)
ax.legend(loc=1, fontsize=10)
ax.tick_params('y', colors='k', labelsize= 10)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-52-d9737c74e2e4> in <module>
```

```
#se importan todas las librerias a utilizar
from sklearn.decomposition import PCA
from sklearn.preprocessing import scale

#Se importan los archivos
data= pd.read_excel("https://github.com/edieraristizabal/Libro_cartoGeotecnia/blob/master/dat
puntos=data['INVENTARIO']
data.drop('INVENTARIO', axis=1, inplace=True)

# Se debe escalar los datos antes de aplicar PCA
data = pd.DataFrame(scale(data), columns=['DV', 'A', 'CP', 'CT', 'DF', 'GEOLOGIA','RR','R','S

## Se implementa el análisi PCA con la libreria sklearn de python
n = len(data.columns)
pca = PCA(n_components=n)
pca = pca.fit(data)
pca_samples = pca.transform(data)

#Se puede graficar cuanto aporta a la varianza cada componente generado
plt.plot(pca.explained_variance_ratio_)
plt.xlabel('Number of components')
plt.ylabel('Explained variance')
plt.show()

#graficamos el acumulado de varianza explicada en las nuevas dimensiones
plt.plot(np.cumsum(pca.explained_variance_ratio_))
plt.xlabel('number of components')
plt.ylabel('cumulative explained variance')
plt.show()

#Para identificar cada variable como se relaciona con las componentes utilizamos las figuras
# 0,1 denota el componente principl 1 y 2 (PC1 and PC2); para otros componentes se modifica
xvector = pca.components_[0]
yvector = pca.components_[1]

xs = pca.transform(data)[:,0] # Componente principal 1
ys = pca.transform(data)[:,1] # Componente principal 2

mask1=np.ma.masked_where(puntos < 1,xs )
mask2=np.ma.masked_where(puntos < 1,ys )

## Para visualizar las proyecciones de cada variable en los componentes se utiliza la siguien
for i in range(len(xvector)):
# arrows project features (ie columns from csv) as vectors onto PC axes
    plt.arrow(0, 0, xvector[i]*max(xs), yvector[i]*max(ys),
              color='r', width=0.0005, head_width=0.0025)
    plt.text(xvector[i]*max(xs)*1.2, yvector[i]*max(ys)*1.2,
             list(data.columns.values)[i], color='r')
```

```
plt.scatter(xs, ys, s=70,marker='x',c='blue', label='MenM')
plt.scatter(mask1,mask2,facecolors='black', edgecolors='black', s=70, alpha=0.5, label='No Me
plt.tick_params('y', colors='k', labelsize=12, length=2)
plt.tick_params('x', colors='k', labelsize= 12, length=2)
plt.xlabel("Componente Principal 1", fontsize=16)
plt.ylabel("Componente Principal 2", fontsize=16)
plt.legend(fontsize=14)
plt.show()
```

```
---------------------------------------------------------------------------
HTTPError                                 Traceback (most recent call last)
<ipython-input-54-f4aa5d4546f4> in <module>
      4
      5 #Se importan los archivos
----> 6 data=
pd.read_excel("https://github.com/edieraristizabal/Libro_cartoGeotecnia/blob/maste
raw=true", sheet_name='PUNTOS')
      7 puntos=data['INVENTARIO']
      8 data.drop('INVENTARIO', axis=1, inplace=True)

                                  12 frames
/usr/lib/python3.7/urllib/request.py in http_error_default(self, req, fp, code,
msg, hdrs)
    647 class HTTPDefaultErrorHandler(BaseHandler):
    648     def http_error_default(self, req, fp, code, msg, hdrs):
--> 649         raise HTTPError(req.full_url, code, msg, hdrs, fp)
    650
    651 class HTTPRedirectHandler(BaseHandler):
```

Colab paid products - Cancel contracts here

0s completed at 4:47 PM