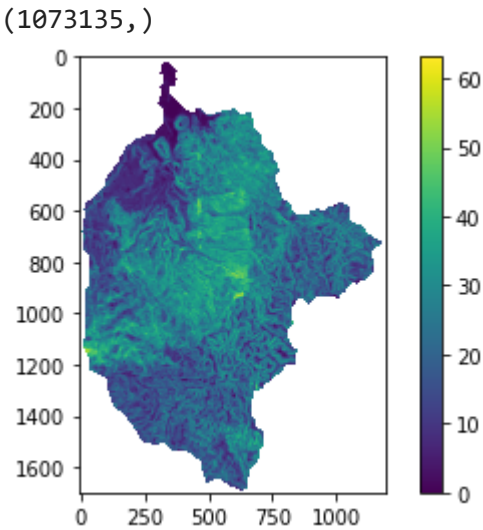


```
!pip install rasterio
```

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

```
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
```

```
raster = rio.open ('/content/drive/MyDrive/Materias/Cartografía geotécnica /Documentos_CartografíaGeotécnica/DATOS/Pendiente
pendiente=raster.read(1)
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 vector los datos NaN
pendiente_vector_MenM.shape # otra forma de saber las dimensiones
```



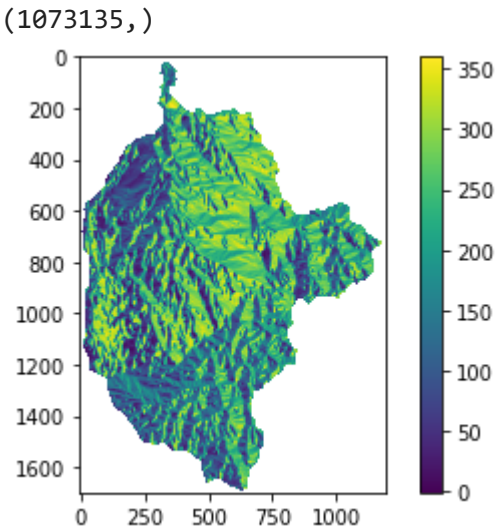
```
type(pendiente)

numpy.ndarray
```

```
np.shape(pendiente)

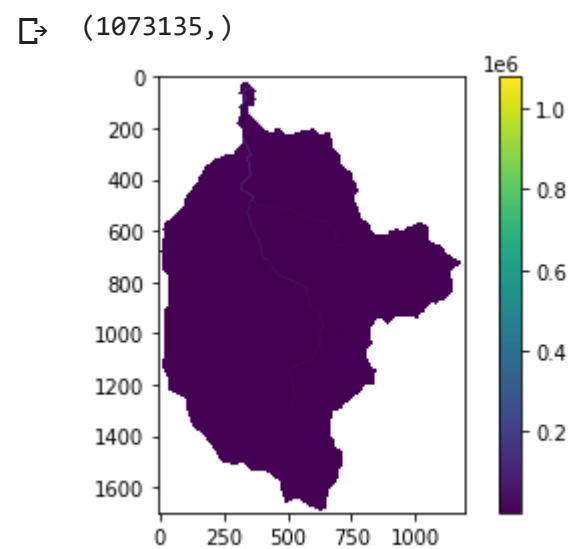
(1699, 1195)
```

```
raster = rio.open('/content/drive/MyDrive/Materias/Cartografía geotécnica /Documentos_CartografíaGeotécnica/DATOS/Aspecto.tif')
aspecto=raster.read(1)
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
```

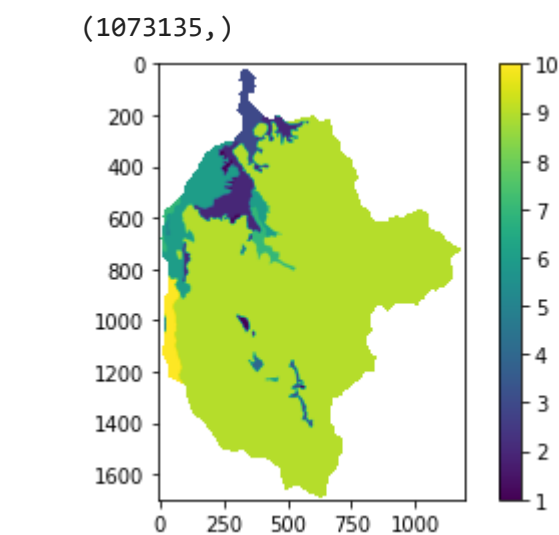


```
np.shape(flujo)

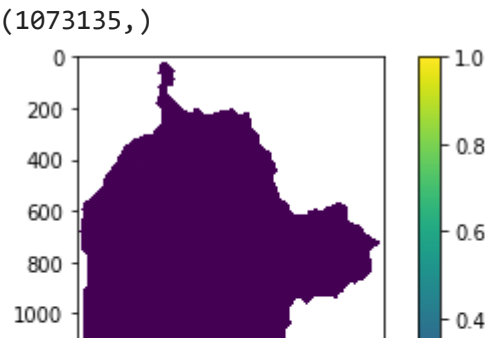
raster = rio.open('/content/drive/MyDrive/Materias/Cartografía geotécnica /Documentos_CartografíaGeotécnica/DATOS/Flow_Accu.1
flujo=raster.read(1)
raster_mask = rio.open('/content/drive/MyDrive/Materias/Cartografía geotécnica /Documentos_CartografíaGeotécnica/DATOS/Pendi
msk=raster_mask.read_masks(1)
msk=np.where(msk==255,1,np.nan)
flujo=msk*flujo
flujo_vector=flujo.ravel()
flujo_vector_MenM=flujo_vector[~np.isnan(flujo_vector)]
plt.imshow(flujo)
plt.colorbar()
flujo_vector_MenM.shape
```



```
raster = rio.open('/content/drive/MyDrive/Materias/Cartografía geotécnica /Documentos_CartografíaGeotécnica/DATOS/Geología.t:
geologia=raster.read(1)
raster_mask = rio.open('/content/drive/MyDrive/Materias/Cartografía geotécnica /Documentos_CartografíaGeotécnica/DATOS/Pendi
msk=raster_mask.read_masks(1)
msk=np.where(msk==255,1,np.nan)
geologia=msk*geologia
geologia_vector=geologia.ravel()
geologia_vector_MenM=geologia_vector[~np.isnan(geologia_vector)]
plt.imshow(geologia)
plt.colorbar()
geologia_vector_MenM.shape
```




```
raster = rio.open('/content/drive/MyDrive/Materias/Cartografía geotécnica /Documentos_CartografíaGeotécnica/DATOS/inventario
inventario=raster.read(1)
raster_mask = rio.open('/content/drive/MyDrive/Materias/Cartografía geotécnica /Documentos_CartografíaGeotécnica/DATOS/Pendi
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
```



```
d={'inventario':inventario_vector_MenM,'pendiente':pendiente_vector_MenM,'flujo_acum':flujo_vector_MenM,'aspecto':aspecto_vector_MenM,'geologia':geologia_vector_MenM}
df = pd.DataFrame(d)
print(list(df.columns))
```

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

```
df.head()
```

	inventario	pendiente	flujo_acum	aspecto	geologia	
0	0.0	0.746838	2.0	162.738892	3.0	
1	0.0	0.707836	3.0	195.004776	3.0	
2	0.0	0.493800	1.0	199.883652	3.0	
3	0.0	0.864414	1.0	144.305893	3.0	
4	0.0	0.870724	4.0	157.589264	3.0	

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1073135 entries, 0 to 1073134
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   inventario  1073135 non-null float64
1   pendiente  1073135 non-null float32
2   flujo_acum 1073135 non-null float64
3   aspecto    1073135 non-null float32
4   geologia   1073135 non-null float64
dtypes: float32(2), float64(3)
memory usage: 32.7 MB
```

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

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 107337 entries, 0 to 1073130
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   inventario  107337 non-null float64
1   pendiente  107337 non-null float32
2   flujo_acum 107337 non-null float64
3   aspecto    107337 non-null float32
4   geologia   107337 non-null float64
dtypes: float32(2), float64(3)
memory usage: 4.1 MB
```

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

	count	mean	std	min	25%	50% \
inventario	107337.0	0.000224	0.014951	0.0	0.000000	0.000000
pendiente	107337.0	22.647333	10.638272	0.0	14.440104	23.205252
flujo_acum	107337.0	1115.124822	23020.459752	1.0	3.000000	6.000000
aspecto	107337.0	188.683899	124.304214	-1.0	58.194107	220.642776
geologia	107337.0	8.341290	1.830089	1.0	9.000000	9.000000
	75%	max				
inventario	0.000000	1.000000e+00				
pendiente	30.857691	6.255898e+01				
flujo_acum	13.000000	1.076870e+06				
aspecto	306.488983	3.599980e+02				
geologia	9.000000	1.000000e+01				

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

	pendiente	flujo_acum	aspecto	geologia
0	0.746838	2.0	162.738892	3.0
1	0.707836	3.0	195.004776	3.0
2	0.493800	1.0	199.883652	3.0
3	0.864414	1.0	144.305893	3.0
4	0.870724	4.0	157.589264	3.0



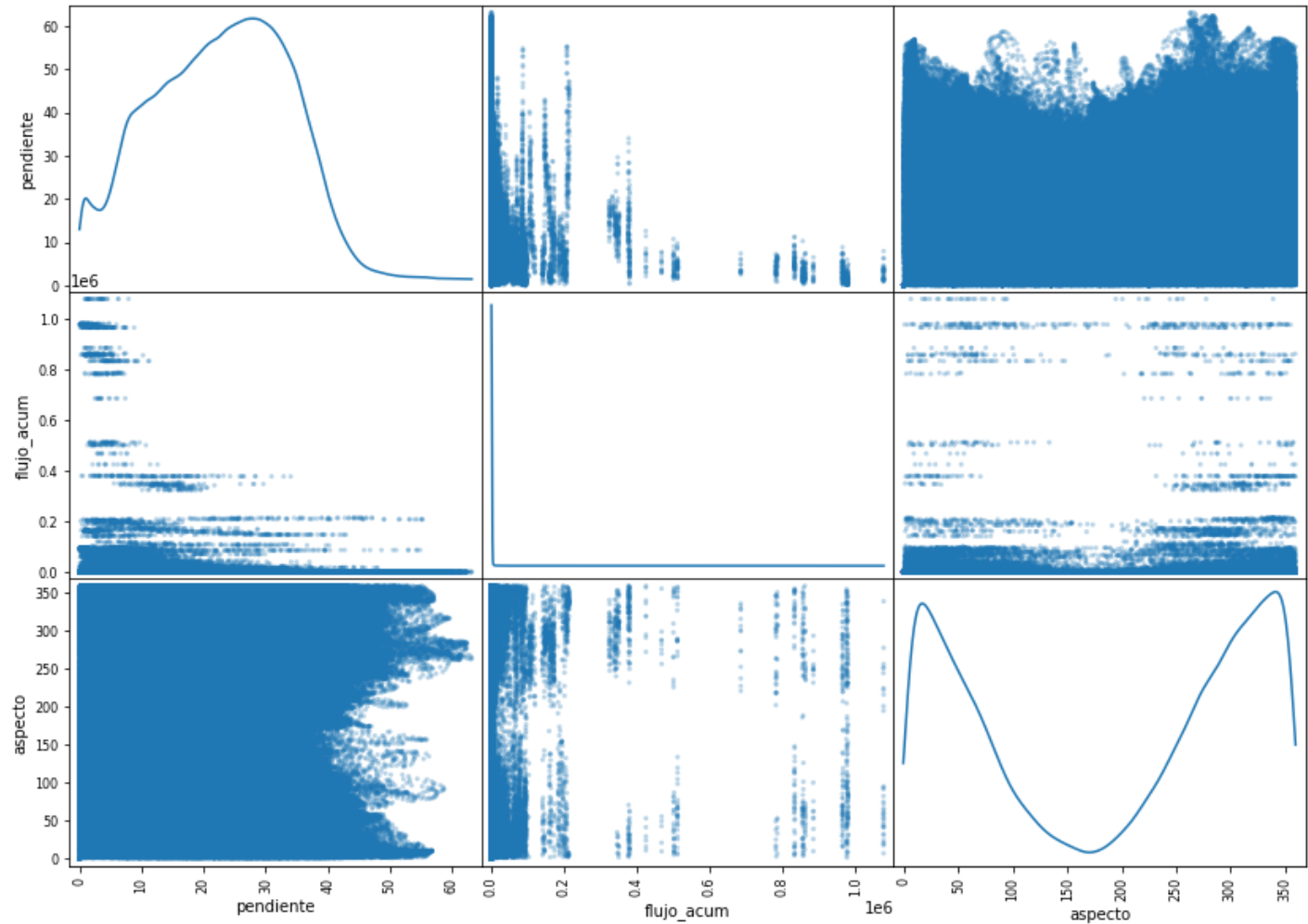
```
matriz_cont=matriz.drop(['geologia'],axis=1)
matriz_cont.head()
```

	pendiente	flujo_acum	aspecto
0	0.746838	2.0	162.738892
1	0.707836	3.0	195.004776
2	0.493800	1.0	199.883652
3	0.864414	1.0	144.305893
4	0.870724	4.0	157.589264

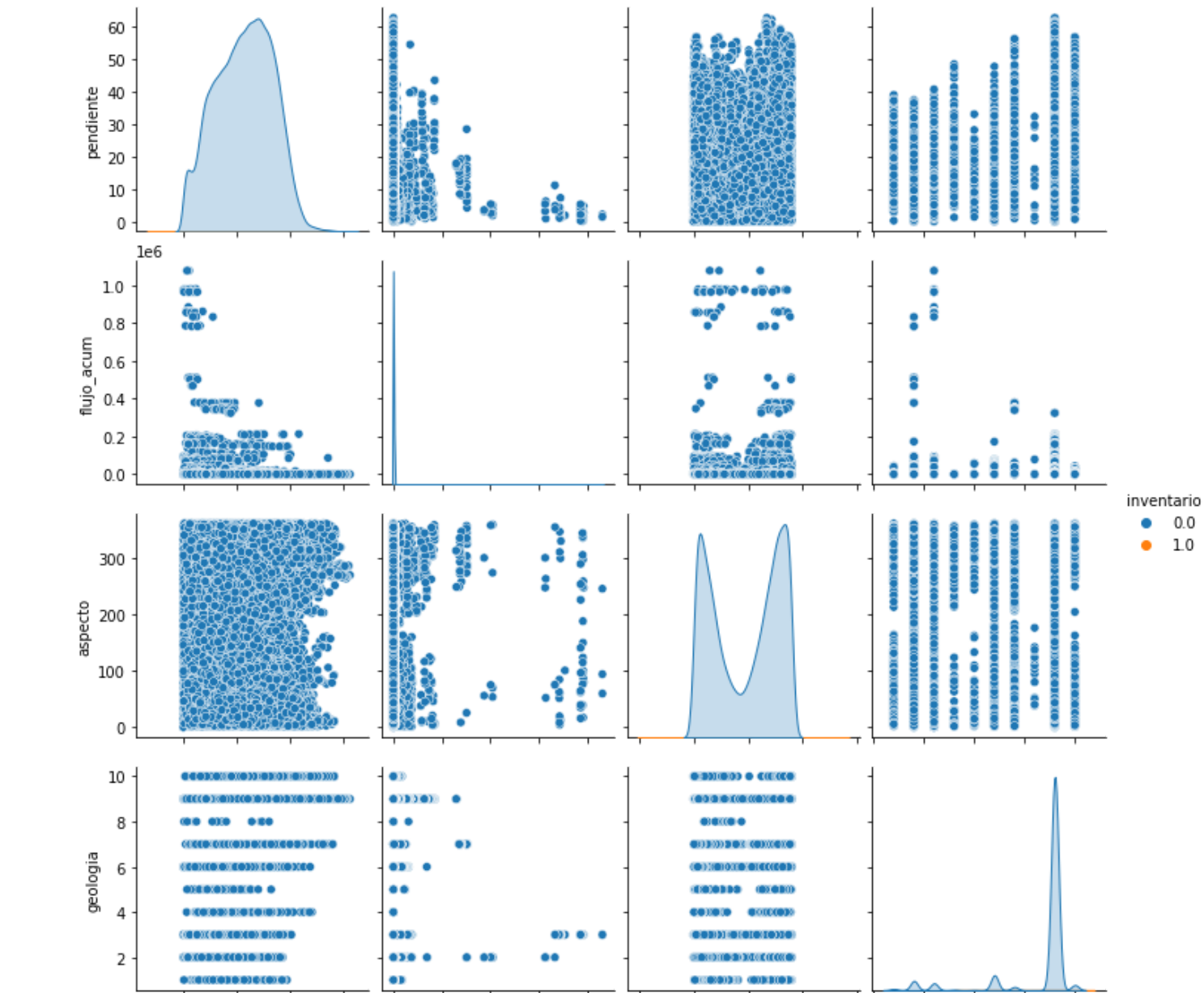


Análisis de todas las variables

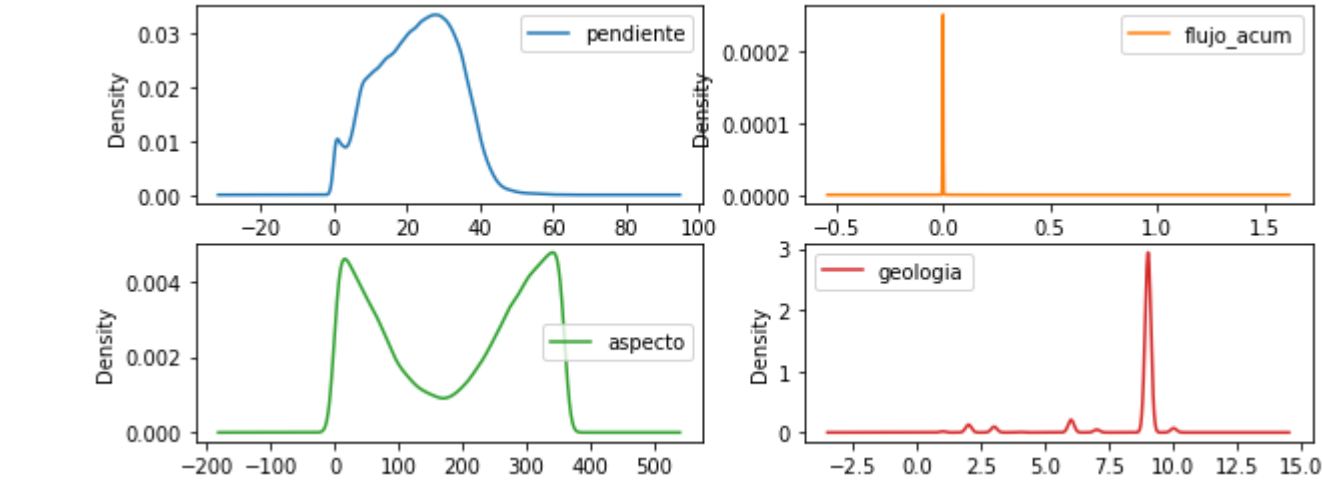
```
pd.plotting.scatter_matrix(matriz_cont, alpha = 0.3, figsize = (14,10), diagonal='kde');
```



```
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)) ;
```

