

Supplementary material

Calculating geometries in QGIS field calculator

Area: \$area

Length: \$length

Orientation: Degrees(azimuth(start_point(\$geometry), end_point(\$geometry)))

Calculating basic fracture parameters in Python

#Import key modules including geopandas, numpy and pandas

Import geopandas as gpd

Import numpy as np

Import pandas as pd

You may need install geopandas using either:

#Anaconda install

conda install -c conda-forge geopandas

or

#pip install

pip install geopandas

#Import fracture trace shape file and analysis window shapefile as Geopandas dataframes.

(Terms in italics and capitals need to be changed).

df=gpd.read_file(r"*LINK TO SHAPE FILE LOCATION OF FRACTURE TRACES*")

aoi=gpd.read_file(r"*LINK TO SHAPE FILE LOCATION OF AREAS*")

#Group dataframe by each analysis window.

df2=df[df['*ANALYSIS_WINDOW*']== '*ANALYSIS_WINDOW_REFERENCE*']

aoi2=aoi[aoi['*ANALYSIS_WINDOW*']== '*ANALYSIS_WINDOW_REFERENCE*']

For making individual density and spacing calculations

#Make python array of lengths and area from shapefile geometries.

df2_length=np.array(df2.geometry.length)

aoi2_area=np.array(aoi2.geometry.area)

#Calculate sum of lengths.

```
df2_length_sum=np.sum(df2.length)
```

#Finally, calculate fracture density based total fracture length and area.

```
density=df2_length_sum/aoi2_area
```

#In addition, as before calculate spacing, which is the reciprocal of Density.

```
spacing=aoi2_area /df2_length_sum
```

For making density and spacing calculations for multiple groups of data

#Add length from the geometry column to a new column

```
df['length']=df.geometry.length
```

#Group dataframe based on digitisation areas

```
df_group=df.groupby(df. ANALYSIS_WINDOW)
```

#Calculate the sum lengths for each group and convert to pandas dataframe

```
df_group_lengths=pd.DataFrame(np.sum(df_group.length))
```

#Add Location column to dataframe

```
df_group_lengths['Location']=df_group_lengths.index
```

#Merge main data frame with area dataframe based on location

```
df3=pd.merge(df_group_lengths, aoi, how='inner', on='ANALYSIS_WINDOW')
```

#Calculate density and intensity in new a columns

```
df3['density']=df3.length/df3.area
```

```
df3['spacing']=df3.area/df3.length
```

#Export as excel file

```
df3.to_excel(r"LINK TO LOCATION TO CREATE EXCEL FILE IN")
```