

(3) 第 14~17 行对影像波段进行缩放计算。影像进行乘运算时会丢失大部分的基本属性，为了保留日期的属性 `system: time_start`，这里先用一个变量存储这个属性值，然后缩放后又将这个属性重新赋值回去。

(4) 第 20~29 行处理影像数据。

(5) 第 31~49 行加载显示 roi，以及对应的影像数据。

第五步：将影像数据转换为本地的 numpy 数组 Array 格式并且绘制显示。

```

1.  # 获取影像数据的经纬度，这里需要注意的是这里要加入一个新的波段，
   否则生成 list 之后没法还原成想要的影像
2.  def getImgData(image):
3.      image = image.addBands(ee.Image.pixelLonLat())
4.      data = image.reduceRegion(
5.          reducer=ee.Reducer.toList(),
6.          geometry=roi,
7.          scale=10,
8.          maxPixels=1e13
9.      )
10.     ndvi=np.array(ee.Array(data.get("NDVI")).getInfo())
11.     lat  = np.array(ee.Array(data.get("latitude")).getInfo())
12.     lon  = np.array(ee.Array(data.get("longitude")).getInfo())
13.     return lat, lon, ndvi
14.  # 将读取的信息转换为 Array
15.  def toArray(lats,lons,data):
16.      uniqueLats = np.unique(lats)
17.      uniqueLons = np.unique(Lons)
18.      ncols = len(uniqueLons)
19.      nrows = len(uniqueLats)
20.      arr = np.zeros([nrows, ncols], np.float32)
21.      counter =0
22.      for y in range(0, nrows):
23.          for x in range(0, ncols):
24.              if lats[counter] == uniqueLats[y] \
25.                  and lons[counter] == uniqueLons[x] \
26.                  and counter < len(lats)-1:
27.                  counter+=1

```