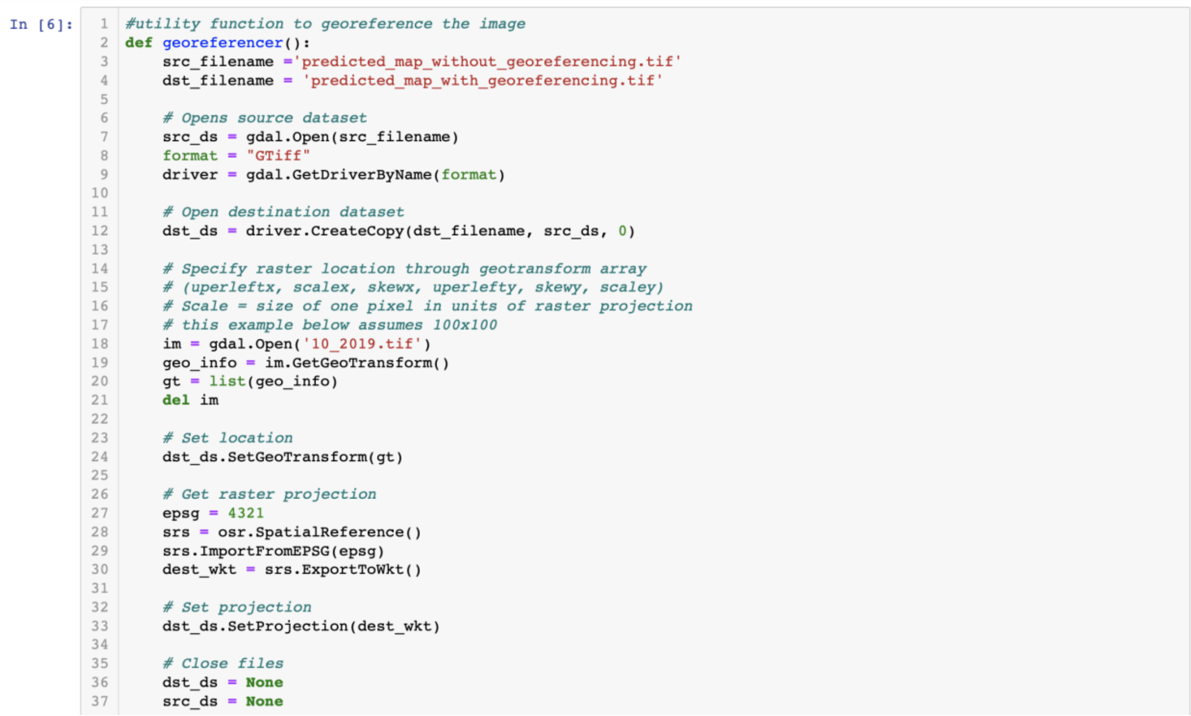
Let’s move to the *georeferencer* function now.



```
#utility function to georeference the image
def georeferencer():
    src_filename ='predicted_map_without_georeferencing.tif'
    dst_filename = 'predicted_map_with_georeferencing.tif'

    # Opens source dataset
    src_ds = gdal.Open(src_filename)
    format = "GTiff"
    driver = gdal.GetDriverByName(format)

    # Open destination dataset
    dst_ds = driver.CreateCopy(dst_filename, src_ds, 0)

    # Specify raster location through geotransform array
    # (uperleftx, scalex, skewx, uperlefty, skewy, scaley)
    # Scale = size of one pixel in units of raster projection
    # this example below assumes 100x100
    im = gdal.Open('10_2019.tif')
    geo_info = im.GetGeoTransform()
    gt = list(geo_info)
    del im

    # Set location
    dst_ds.SetGeoTransform(gt)

    # Get raster projection
    epsg = 4321
    srs = osr.SpatialReference()
    srs.ImportFromEPSG(epsg)
    dest_wkt = srs.ExportToWkt()

    # Set projection
    dst_ds.SetProjection(dest_wkt)

    # Close files
    dst_ds = None
    src_ds = None
```

In *src_filename*, we have to put the name of the image file whose georeferencing we want to do.

In *dst_filename,* we have to put the name of the image file we want to generate by georeferencing.
Our desired fomat in *GTiff,* so we will put that in *format*.

In *im*, we have to put the name of the image whose georeferencing you want to copy in the output. In this case, it can be either of the 6 images.

Rest of the work will be done automatically and output image with georeferencing will be saved in the system will name as *dst_filename*.