Compute LST and NDVI in NYC for Landsat 8

Introduction

In this notebook, we compute LST for the New York City using *Landsat 8* data that is clipped to NYC boundaries. To compute LST, we follow the procedure outlined in Using McConnell's 2022 paper adapted for Landsat 8. The steps are

- Convert Band 6 to Top of Atmosphere (TOA) spectral radiance using two constants from Landsat metadata
- Compute brightness temperature using TOA and 2 constants from metadata
- Compute NDVI using

$$\frac{\mathrm{NIR} - \mathrm{Red}}{\mathrm{NIR} + \mathrm{Red}} = \frac{\mathrm{Band}\,5 - \mathrm{Band}\,4}{\mathrm{Band}\,5 + \mathrm{Band}\,4} \tag{1}$$

- Convert NDVI to vegetation fraction
- Compute emissivity using the formula specified in McConnell's paper
- Compute LST using brightness temperature, emissivity, and some constants also specified in McConnell's paper

The contents of this notebook are the following.

- In Section 1, we define functions to compute the outlined procedure
- In Section 2 we test the procedure on a single Landsat file and plot the result.
- In Section 3 we compute LST and NDVI for all the Landsat files and export results.

Results

LST and NDVI are computed in this notebook and exported for later use. A heatmap of LST/NDVI can be generated in Section 2.

Data

The data imported into this notebook is clipped Landsat data located in `02-data/landsat_clipped_nyc/`. These are raster files saved in `.tif` format. We import the files relevant to LST calculations with Landsat 8, which are bands 4, 5, and 10.

Data is exported to the folders `02-data/ndvi_clipped_nyc` and `02-data/lst_clipped_nyc`. The exported filenames contain the parameter computed with the original Landsat filename. For example, the file `ndvi_LC08_L1GT_109212_20150719_20170406_01_T2.tif` contains the NDVI calculation in NYC for the Landsat file listed.

In [2]:

```
import os
DIR_PARENT = os.path.abspath(os.path.join(os.getcwd(), os.pardir))
DIR_SCRIPTS = DIR_PARENT + "/01-scripts"

"""Push the directory to load helper scripts from"""
import sys
sys.path.append(DIR_SCRIPTS)

import helpers

import rioxarray as rxr
import xarray as xr
import numpy as np

# For plotting
import matplotlib.pyplot as plt

# Import directories
DIR_DATA = DIR_PARENT + "/02-data"
DIR_RAW_CLIPPED_NYC = DIR_PARENT + "/02-data/landsat_clipped_nyc"

# Export directories
DIR_NDVI_CLIPPED_NYC = DIR_PARENT + "/02-data/ndvi_clipped_nyc"
DIR_LST_CLIPPED_NYC = DIR_PARENT + "/02-data/lst_clipped_nyc"
DIR_FIGS = DIR_PARENT + "/03-figs"
```

1. Import data and define functions

In [6]:

```
landsat_filenames = []
helpers.get_filenames(DIR_RAW_CLIPPED_NYC, landsat_filenames)
landsat_filenames.sort()

# Filter clipped Landsat files by band
band4_filenames = [x for x in landsat_filenames if ("B4" in x and "LC08" in x)]
band5_filenames = [x for x in landsat_filenames if ("B5" in x and "LC08" in x)]
band10_filenames = [x for x in landsat_filenames \
                    if ("B10" in x and "LC08" in x)]

band4_filenames.sort()
band5_filenames.sort()
band10_filenames.sort()

print("Number of B4 files = ", len(band4_filenames))
print("Number of B5 files = ", len(band5_filenames))
print("Number of B10 files = ", len(band10_filenames))

def get_original_landsat_name(name):
    return "_".join(name.split("/")[-1].split(".")[0].split("_")[2:-1])

def import_txt_as_dict(name):
    d = {}
    with open(name) as f:
        for line in f:
            if len(line.split("="))>1:
                (key, val) = line.split("=")
                d[key] = val.split("\n")[0]
    return d
```

```python
def open_mtl_file(landsat_filename):
    dirname = "/".join(landsat_filename.split("/")[:-1])
    fileroot = get_original_landsat_name(landsat_filename)
    return import_txt_as_dict(dirname+"/"+fileroot+"_MTL"+".txt")

def open_masked_band(filename, mask_value):
    band = rxr.open_rasterio(filename)
    data_xarray = xr.DataArray(np.ma.masked_where(band==mask_value, \
                                                  band).squeeze(),
        dims=["y","x"],
        coords={"x":band["x"].data,
               "y":band["y"].data,
               "band":1,
               "spatial_ref":0},
        attrs=band.attrs)
    data_xarray.rio.write_crs(band.rio.crs, inplace=True)
    return data_xarray

def compute_toa_radiance(band, meta):
    Ml = float(meta["    RADIANCE_MULT_BAND_10 "])
    Al = float(meta["    RADIANCE_ADD_BAND_10 "])
    return Ml*band + Al

def compute_brightness_temp(toa, meta):
    K1 = float(meta["    K1_CONSTANT_BAND_10 "])
    K2 = float(meta["    K2_CONSTANT_BAND_10 "])
    return K2/np.log(K1/toa + 1)

def compute_ndvi(nir, red, meta):
    nir_toa = compute_toa_radiance(nir, meta)
    red_toa = compute_toa_radiance(red, meta)
    return (nir_toa-red_toa)/(nir_toa+red_toa)

def compute_vegetation_fraction(ndvi):
    ndvi_min = np.min(ndvi)
    ndvi_max = np.max(ndvi)
    return (ndvi-ndvi_min/(ndvi_max-ndvi_min))**2

def compute_emissivity(p):
    return 0.004*p + 0.986481

def compute_lst(w, bt, emiss):
    Q = 1.438*10**(-2)
    return bt/(1 + w*bt*np.log(emiss)/Q)
```

```
Number of B4 files =  26
Number of B5 files =  26
Number of B10 files =  26
```

2. Test the computation for a single date

In [7]:

```python
"""Test the program by selecting a single file"""
select = 0 # OPTIONS: Integer between 0 and (Number of Landsat files) - 1.

# Open band 10 and its metadata file
band6_test = open_masked_band(band10_filenames[select], 0.0)
mtl_test = open_mtl_file(band10_filenames[select])
```

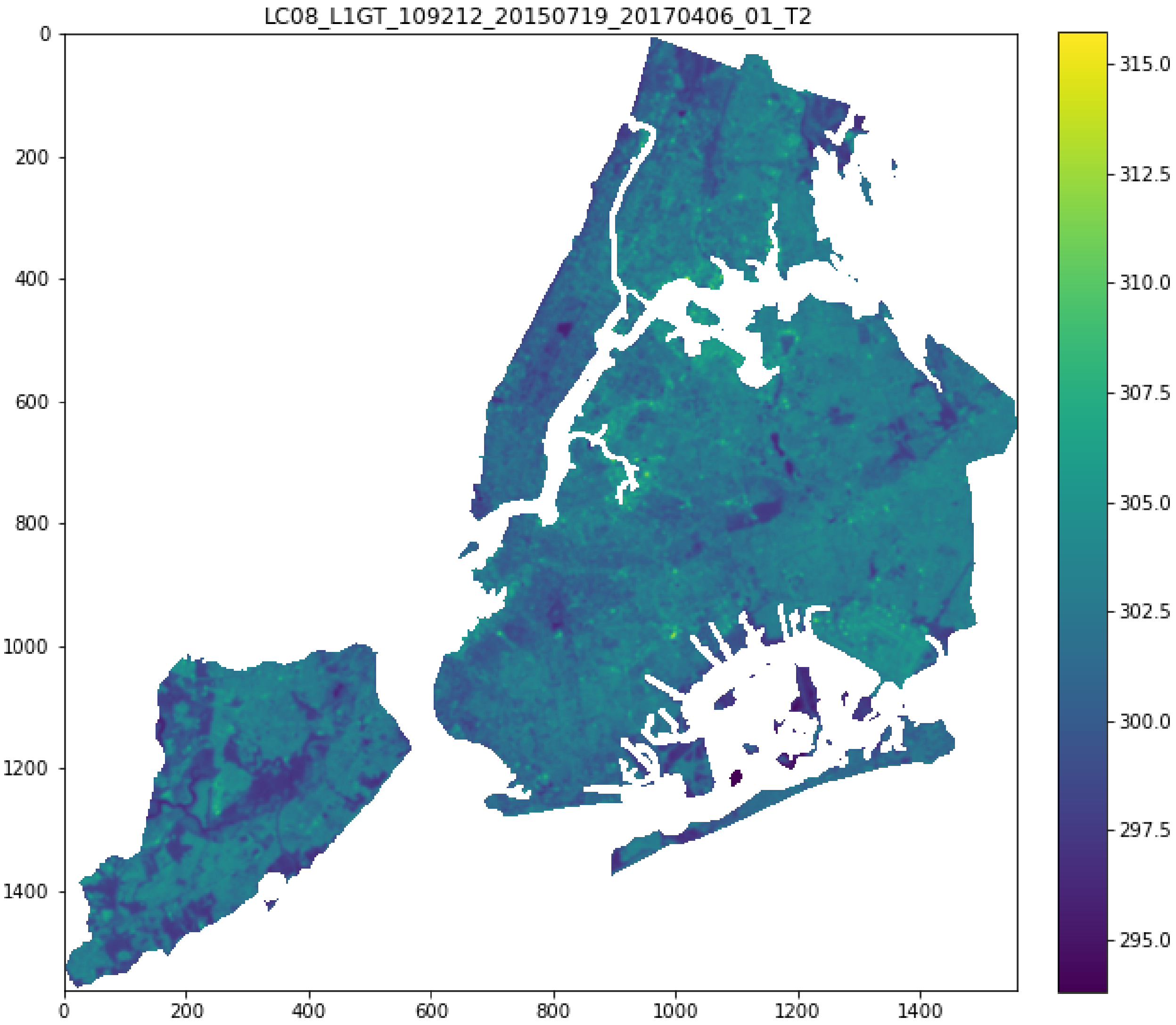
```
# Open nir and red
band5_test = open_masked_band(band5_filenames[select], 0.0)
band4_test = open_masked_band(band4_filenames[select], 0.0)

print(band4_filenames[select])
print(band5_filenames[select])
print(band10_filenames[select])

# Begin LST equations
toa = compute_toa_radiance(band6_test, mtl_test)
bt = compute_brightness_temp(toa, mtl_test)

ndvi = compute_ndvi(band5_test, band4_test, mtl_test)

vf = compute_vegetation_fraction(ndvi)
emiss = compute_emissivity(vf)
w = 10.895*10**-6 # Middle of Band 10 wavelengths, which range 10.6-11.19 µm
lst = compute_lst(w, bt, emiss)

exportname_lst = DIR_LST_CLIPPED_NYC + "/lst_" + \
    get_original_landsat_name(band10_filenames[select])
exportname_ndvi = DIR_NDVI_CLIPPED_NYC + "/ndvi_" + \
    get_original_landsat_name(band10_filenames[select])

# Test export ######################################################
# NOTE: Running this will overwrite any data already exported
# lst.rio.to_raster(exportname_lst, driver="GTiff")
# ndvi.rio.to_raster(exportname_ndvi, driver="GTiff")
```

```
/home/aderrasc/Documents/japa_final/02-data/landsat_clipped_nyc/clipped_nyc_LC08
_L1GT_109212_20150719_20170406_01_T2_B4.TIF
/home/aderrasc/Documents/japa_final/02-data/landsat_clipped_nyc/clipped_nyc_LC08
_L1GT_109212_20150719_20170406_01_T2_B5.TIF
/home/aderrasc/Documents/japa_final/02-data/landsat_clipped_nyc/clipped_nyc_LC08
_L1GT_109212_20150719_20170406_01_T2_B10.TIF
```

Plot results

In [14]:

```
# Plot the test band
object_to_plot = lst # OPTIONS: lst, ndvi, bandX_test, toa, bt, vf, emiss
#########################################################################
fig = plt.figure(figsize=(10, 10))
im = plt.imshow(object_to_plot)#, cmap="RdYlBu")#,vmin=-1,vmax=1)
plt.colorbar(im, fraction=0.046, pad=0.04)
plt.title(get_original_landsat_name(band10_filenames[select]))
plt.savefig(DIR_FIGS + "/lst_"+get_original_landsat_name(\
                                        band10_filenames[select]) + ".png")
plt.show()
```



3. Export LST and NDVI for all dates

In [15]:

```
for index in range(len(band10_filenames)):
    # Open bands and metadata file
    band6 = open_masked_band(band10_filenames[index], 0.0)
    band4 = open_masked_band(band5_filenames[index], 0.0)
    band3 = open_masked_band(band4_filenames[index], 0.0)
    mtl = open_mtl_file(band10_filenames[index])

    # Begin LST equations
    toa = compute_toa_radiance(band6, mtl)
    bt = compute_brightness_temp(toa, mtl)
    ndvi = compute_ndvi(band4, band3, mtl)
    vf = compute_vegetation_fraction(ndvi)
    emiss = compute_emissivity(vf)
    w = 10.895*10**-6 # Middle of Band 10 wavelengths, 10.6-11.19 µm

    lst = compute_lst(w, bt, emiss)

    suffix = get_original_landsat_name(band10_filenames[index]) + ".tif"
    exportname_lst = DIR_LST_CLIPPED_NYC + "/lst_" + suffix
    exportname_ndvi = DIR_NDVI_CLIPPED_NYC + "/ndvi_" + suffix
    print("Saving ", suffix)
    lst.rio.to_raster(exportname_lst, driver="GTiff")
    ndvi.rio.to_raster(exportname_ndvi, driver="GTiff")
```

```
Saving  LC08_L1GT_109212_20150719_20170406_01_T2.tif
Saving  LC08_L1GT_109212_20150804_20170406_01_T2.tif
Saving  LC08_L1GT_109212_20150820_20170405_01_T2.tif
Saving  LC08_L1GT_109212_20150905_20170404_01_T2.tif
Saving  LC08_L1GT_109212_20150921_20170404_01_T2.tif
Saving  LC08_L1TP_013032_20140731_20170304_01_T1.tif
Saving  LC08_L1TP_013032_20140917_20170303_01_T1.tif
Saving  LC08_L1TP_013032_20150803_20170226_01_T1.tif
Saving  LC08_L1TP_013032_20160720_20170222_01_T1.tif
Saving  LC08_L1TP_013032_20160805_20170222_01_T1.tif
Saving  LC08_L1TP_013032_20170909_20170927_01_T1.tif
Saving  LC08_L1TP_013032_20180710_20180717_01_T1.tif
Saving  LC08_L1TP_013032_20190713_20190719_01_T1.tif
Saving  LC08_L1TP_013032_20190729_20190801_01_T1.tif
Saving  LC08_L1TP_013032_20190830_20190916_01_T1.tif
Saving  LC08_L1TP_014032_20130820_20170309_01_T1.tif
Saving  LC08_L1TP_014032_20140706_20170304_01_T1.tif
Saving  LC08_L1TP_014032_20140807_20170304_01_T1.tif
Saving  LC08_L1TP_014032_20150725_20170226_01_T1.tif
Saving  LC08_L1TP_014032_20150826_20170225_01_T1.tif
Saving  LC08_L1TP_014032_20160727_20170222_01_T1.tif
Saving  LC08_L1TP_014032_20160812_20170222_01_T1.tif
Saving  LC08_L1TP_014032_20160828_20170221_01_T1.tif
Saving  LC08_L1TP_014032_20170730_20170811_01_T1.tif
Saving  LC08_L1TP_014032_20180903_20180912_01_T1.tif
Saving  LC08_L1TP_014032_20190922_20190926_01_T1.tif
```