Lab 08: Advanced Raster Analysis Using Landsat 7-Band Data in PostGIS

You will deepen your understanding of raster operations in PostGIS using Landsat imagery from the below aspects:

1. Compute additional spectral indices
2. Apply reclassification and pixel counts

Total counts: 9pts

**Please submit the lab in word or PDF format, with pasted (don’t screenshot) commands in blue color and screenshotted results.**

**In-class exercise 1:** UseST\_SummaryStats() to calculate count, sum, mean, stddev, min, max for b5 and b4 from ch12.landsat (1pt).

SELECT

(stats\_b4).\* AS b4,

(stats\_b5).\* AS b5

FROM (

SELECT

ST\_SummaryStats(rast, 4) AS stats\_b4,

ST\_SummaryStats(rast, 5) AS stats\_b5

FROM ch12.landsat

) AS subquery;

Paste your screenshot:

In-class exercise 2: Use ST\_Union() to combine all tiles into one single tile for ch12.ndvi\_raster and name it as ch12.ndvi\_raster\_union. (1pt)

CREATE TABLE ch12.ndvi\_raster\_union AS

SELECT

ST\_Union(rast) AS rast

FROM ch12.ndvi\_raster;

Paste your screenshot:

**Task 1:** You need to classify the **ch12.barren\_index** using the [ST\_Reclass](https://postgis.net/docs/RT_ST_Reclass.html) function in PostGIS. This is useful to group continuous BI values into meaningful land categories, such as **sparse**, **moderate**, or **dense** bare soil. (1pt)

Hints:

1. ST\_Reclass reassigns raster values from a **range** into **class values**.
2. '-1.0-0.1:1, 0.1-0.3:2, 0.3-1.0:3' is a reclassification rule. For example, pixels with BI between 0.1 and 0.3 will be given the value 2.
3. '32BF' means the output raster will be in 32-bit float format.

CREATE TABLE ch12.barren\_classified AS

SELECT rid,

ST\_Reclass(

rast,

1,

'-1.0-0.1:1, 0.1-0.3:2, 0.3-1.0:3',

'32BF'

) AS classified\_rast

FROM ch12.barren\_index;

Paste your screenshot:

Demo answer:

A screenshot of a computer

Description automatically generated

**Task 2.** Water strongly absorbs in Near-Infrared (NIR) and reflects in the Green band. A common water index is the **Normalized Difference Water Index (NDWI)**: **NDWI = (Green - NIR) / (Green + NIR)**. Use Band 3 (Green) and Band 5 (NIR) from ch12.landsat. Please use these two bands to calculate the NDWI (name it: ch12.ndwi\_index). (1pt)

**DROP TABLE IF EXISTS ch12.ndwi;**

**CREATE TABLE ch12.ndwi AS**

**SELECT**

**rid,**

**ST\_MapAlgebra(rast, 5, rast, 3,**

**'([rast1] - [rast2]) / ([rast1] + [rast2])') AS rast**

**FROM ch12.landsat;**

Paste your screenshot:

**Demo answer:**

**A map of a river

Description automatically generated**

Task 3. After generating the NDWI raster, identify water pixels by thresholding NDWI (e.g., NDWI > 0). Higher NDWI values indicate a higher likelihood of water presence. Positive values are typically associated with water bodies, while negative or zero values are associated with soil and land-based vegetation. (When you load NDWI data into QGIS, it may take a while). (3 pts)

Hints:

1. '-1.0-0.0:1,0.0-1.0:2' is the reclassification rule.

DROP TABLE IF EXISTS ch12.ndwi\_water\_mask;

CREATE TABLE ch12.ndwi\_water\_mask AS

SELECT rid,

ST\_Reclass(rast, 1, '-1.0-0.0:1, 0.0-1.0:2', '32BF') AS classified\_rast

FROM ch12.ndwi;

Paste your screenshot:

Demo answer:

A black and white image of a river

Description automatically generated

**Task 4.** You have generated an NDWI raster and reclassified it into a binary raster (ch12.ndwi\_water\_mask), where:

1 represents water pixels; 2 represents non-water pixels (1pt)

Your task is to calculate the total number of water pixels.

1. Use the ST\_ValueCount() function to summarize pixel values from the raster.
2. Focus on counting only those pixels where the value equals 1 (indicating water).
3. Wrap the summary in a subquery to filter and sum water pixels.

SELECT SUM((stats).count) AS water\_pixel\_count

FROM (

SELECT ST\_ValueCount(classified\_rast) AS stats

FROM ch12.ndwi\_water\_mask

) AS counts

WHERE (stats).value = 1;

Paste your screenshot:

**Task 5.** Combining ch12.barren\_index, ch12.ndvi\_raster, ch12.ndwi\_index into a single raster with 3 bands (ch12.landsat\_indices) by using ST\_AddBand(). (1pt)

Hint:

1. When you are combing multiple raster layers, it is important to match each raster tile correctly by using JOIN … ON …

2. You might need to use ~= for spatial alignment (and the condition may be like JOIN … ON … AND)

**Task 6.** Use ST\_SummaryStats() to calculate min, max, mean, and stddev for NDVI and NDWI from ch12.landsat\_indices. (1pt)

SELECT

(stats\_ndvi).min AS ndvi\_min,

(stats\_ndvi).max AS ndvi\_max,

(stats\_ndvi).mean AS ndvi\_mean,

(stats\_ndvi).stddev AS ndvi\_sd,

(stats\_ndwi).min AS ndwi\_min,

(stats\_ndwi).max ndwi\_max,

(stats\_ndwi).mean AS ndwi\_mean,

(stats\_ndwi).stddev AS ndwi\_sd

FROM (

SELECT

ST\_SummaryStats(rast, 1) AS stats\_ndvi,

ST\_SummaryStats(rast, 3) AS stats\_ndwi

FROM ch12.landsat\_indices

) AS subquery;

Paste your screenshot:

Demo answer:

A screenshot of a computer

Description automatically generated

**Task 7.** If you open the table of ch12.landsat\_indices, you will find 30 separate rows, which is because each raster band is being imported row by row and tile by tile rather than a single raster. Please use ST\_Union() to combine tiles and do the statistical summary for NDWI. (1pt)

DROP TABLE IF EXISTS ch12.landsat\_indices\_union;

CREATE TABLE ch12.landsat\_indices\_union AS

SELECT

ST\_Union(rast) AS rast

FROM ch12.landsat\_indices;

SELECT

(stats\_ndvi).min AS ndvi\_min,

(stats\_ndvi).max AS ndvi\_max,

(stats\_ndvi).mean AS ndvi\_mean,

(stats\_ndvi).stddev AS ndvi\_sd,

(stats\_ndwi).min AS ndwi\_min,

(stats\_ndwi).max ndwi\_max,

(stats\_ndwi).mean AS ndwi\_mean,

(stats\_ndwi).stddev AS ndwi\_sd

FROM (

SELECT

ST\_SummaryStats(rast, 1) AS stats\_ndvi,

ST\_SummaryStats(rast, 3) AS stats\_ndwi

FROM ch12.landsat\_indices\_union

) AS subquery;

Paste your screenshot:

Demo answer

