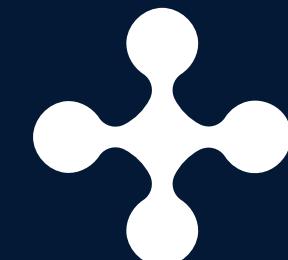


A STEP-BY-STEP GUIDE FOR
GEOSPATIAL ANALYSIS IN ALICANTE

LAND SURFACE
TEMPERATURE & NDVI
ANALYSIS IN ALICANTE,
SPAIN (2014-2023)

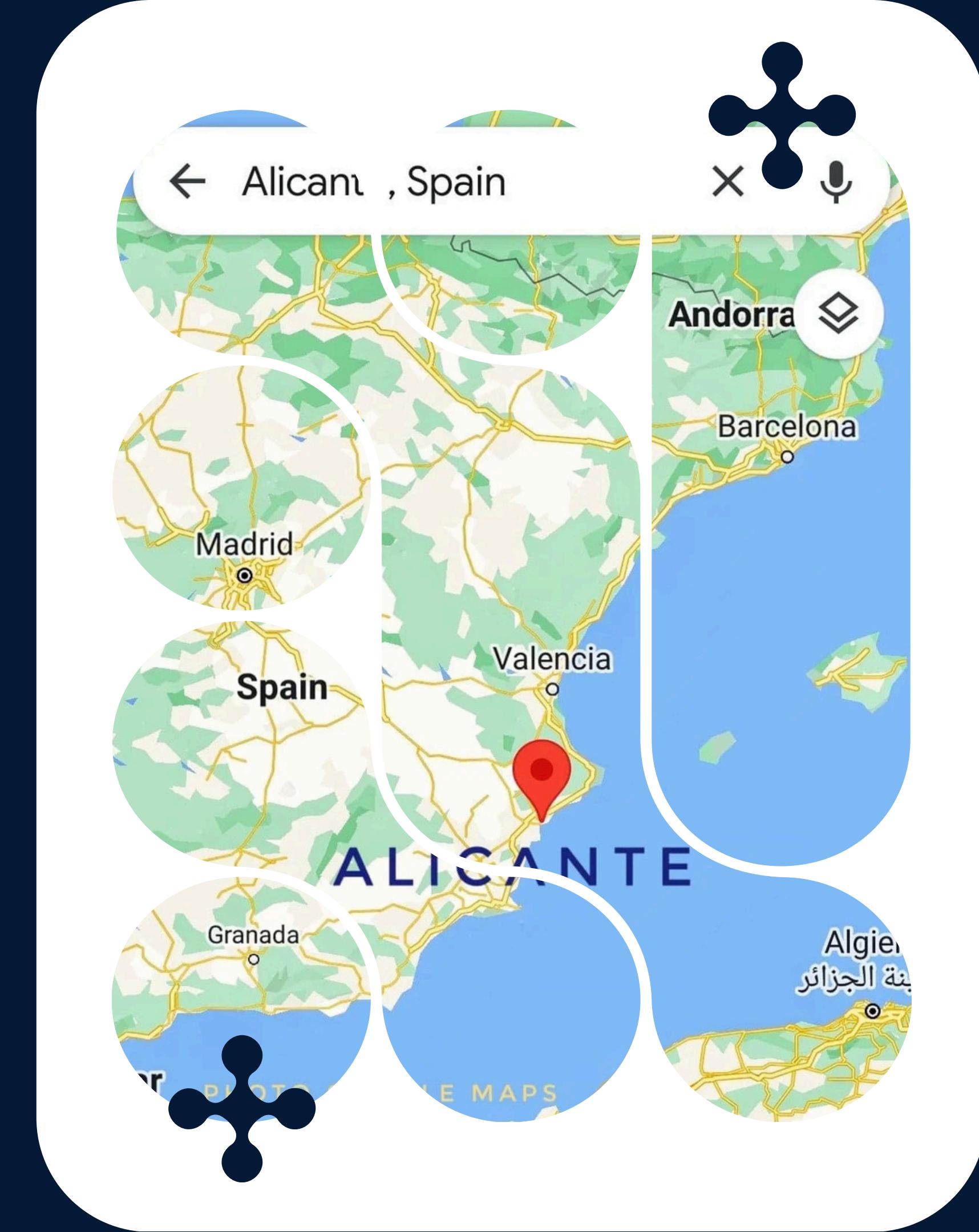
A GOOGLE EARTH ENGINE (GEE) PROJECT
MAHMUD -JULY 26, 2025

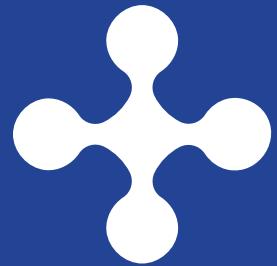


Project Objectives for GEE Analysis

Exploring Land Surface Temperature and NDVI in Alicante

The **primary goals** of this project are to utilize Google Earth Engine to analyze Land Surface Temperature and NDVI data over a 10-year period in Alicante, Spain, providing valuable insights into environmental changes.





Overview of Alicante, Spain

This section highlights key aspects of Alicante's geography and climate.

Mediterranean Climate

Alicante enjoys a **warm Mediterranean climate** ideal for diverse ecosystems.

Urban Development

The city's growth has impacted **land use patterns**, affecting environmental studies.

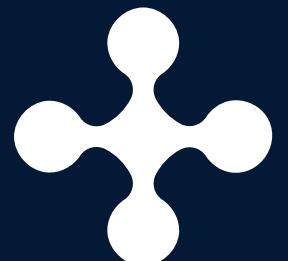
Rich Biodiversity

Alicante is home to various **flora and fauna**, making it a valuable study area.



WHAT IS GOOGLE EARTH ENGINE (GEE)

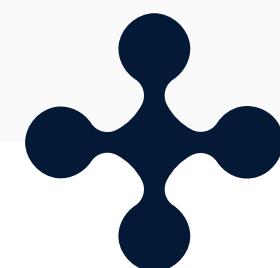
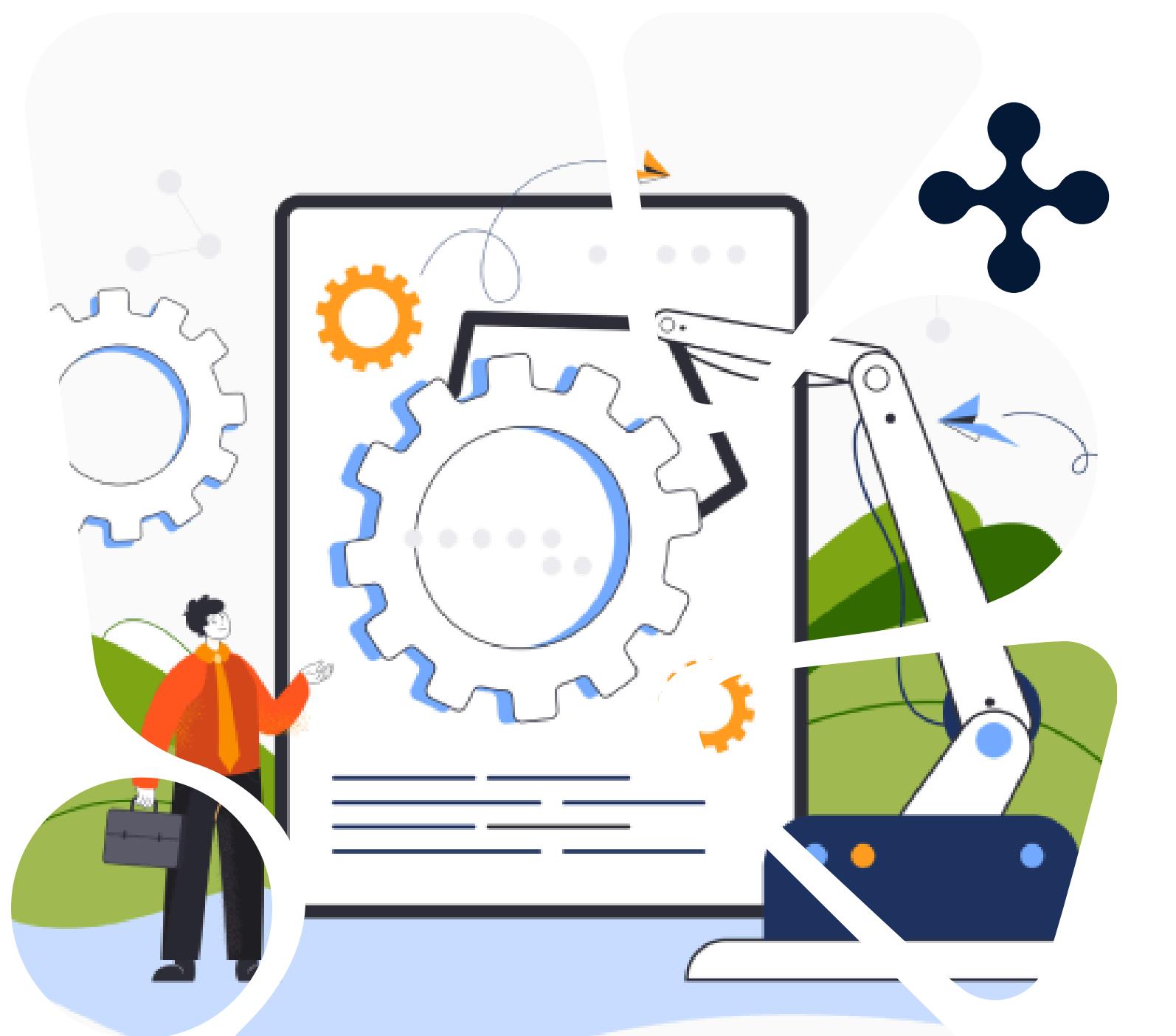
- CLOUD-BASED PLATFORM FOR PLANETARY-SCALE SATELLITE DATA ANALYSIS
- USED FOR REMOTE SENSING AND ENVIRONMENTAL MONITORING AND CLIMATE AND AGRICULTURE
- ENABLES FAST PROCESSING AND VISUALIZATION



Workflow Steps in GEE

**Access, process, and download
environmental data efficiently**

This section outlines the essential **workflow steps** for effectively accessing, processing, and downloading data using Google Earth Engine. By following these steps, you can streamline your analysis of Land Surface Temperature and NDVI for Alicante, Spain, over the past decade.



SOFTWARE IN A WHOLE

Google Earth Engine ? mymemeberlink User icon

Search places and datasets...

Scripts Docs Assets NEW Filter scripts... Refresh

Owner (1)
users/maahmoudsm1/DataViz
Alicante-Spain
Spain

Writer
No accessible repositories. Click Refresh to check again.

Reader
No accessible repositories. Click Refresh to check again.

Archive
No accessible repositories. Click Refresh to check again.

Examples

New Script *

```
1 // Load the full Spain Level 2 shapefile
2 var spain = ee.FeatureCollection("projects/mymemeberlink/assets/gadm41_ESP_2");
3
4 // Filter for Alicante
5 var alicante = spain.filter(ee.Filter.eq('NAME_1', 'Alicante'));
6
7 // Add to map
8 Map.addLayer(alicante, {}, 'Alicante');
9 Map.centerObject(alicante, 9);
10
11 // =====
12 // Load Spain Level 2 shapefile and filter for Alicante
13 // =====
14 var spain = ee.FeatureCollection("projects/mymemeberlink/assets/gadm41_ESP_2");
15 var roi = spain.filter(ee.Filter.eq('NAME_1', 'Alicante'));
16
17 Map.addLayer(roi, {}, 'Alicante');
18 Map.centerObject(roi, 9);
19
20 // =====
21 // Function to Apply Scaling Factors for Landsat 8 SR Collection 2, Tier 1
22 // =====
23 function applyScaleFactors(image) {
24
```

Inspector Console Tasks

Use print(...) to write to this console.

Annual Variation of LST in Alicante (MODIS)

Temperature (°C)

Day of Year

Mean LST Over Time (2014–2023)

LST_Day_1km

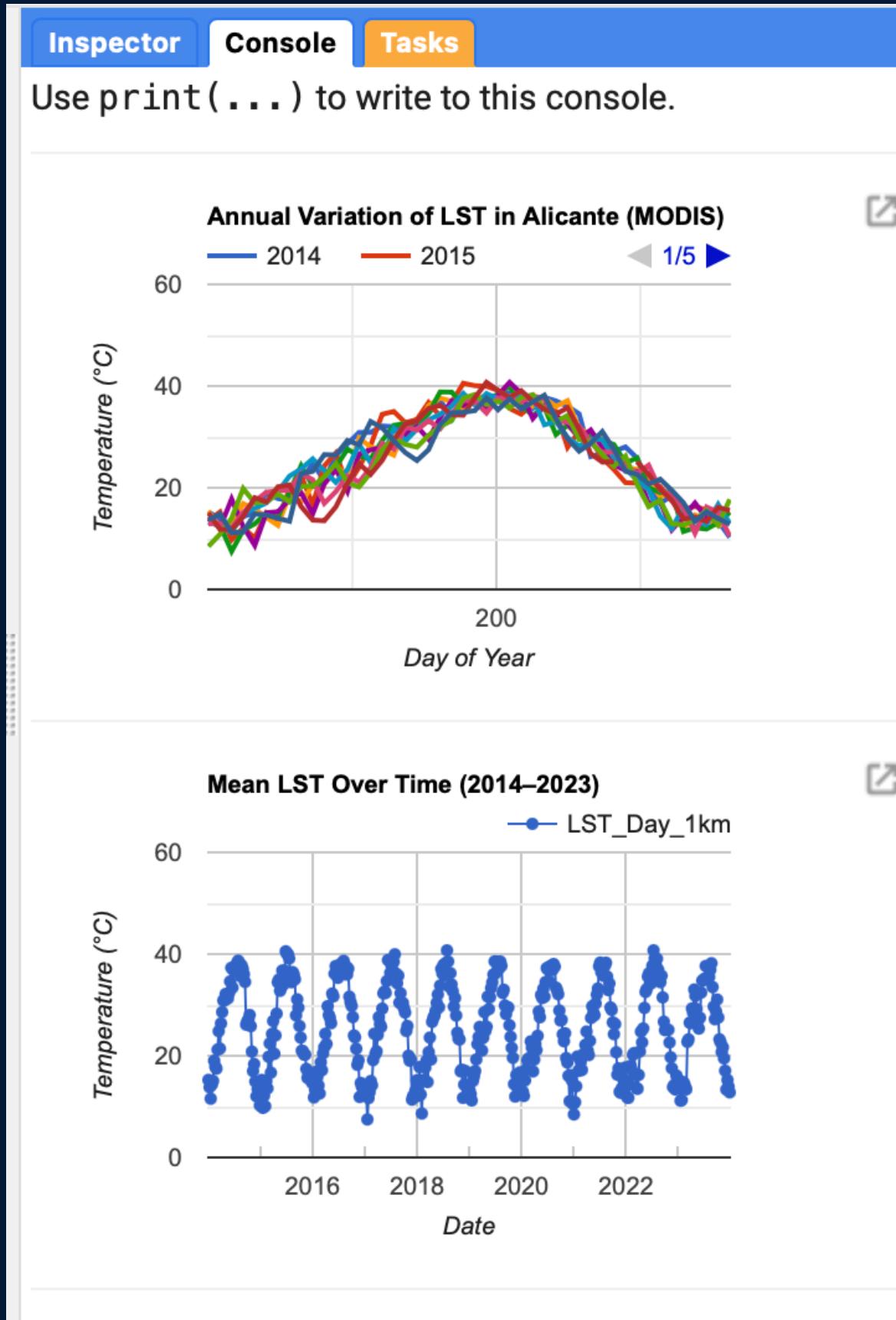
Land Surface Temperature - Alicante (2014–2023)

Layers

- 23.57
- 26.79
- 27.86
- 28.93
- 30.00
- 31.07
- 32.14
- 33.21
- 34.29
- 35.36
- 36.43
- 37.50
- 38.57
- 39.64
- 40.71
- 41.79

The screenshot displays the Google Earth Engine (GEE) web interface. The top navigation bar includes 'Google Earth Engine', a search bar, and user authentication. The left sidebar shows ownership details and a list of available scripts, assets, and examples. The central area features a 'New Script' editor with a code snippet for filtering a Spain Level 2 shapefile to focus on the Alicante region. To the right of the script editor are two charts: one showing the 'Annual Variation of LST in Alicante (MODIS)' from 2014 to 2015, and another showing the 'Mean LST Over Time (2014–2023)'. Below these charts is a map titled 'Land Surface Temperature - Alicante (2014–2023)', which displays a heatmap of LST values across the region. A legend on the right side of the map lists temperature ranges from 23.57°C to 41.79°C, with colors ranging from blue (lowest) to red (highest). The map also shows various geographical features and place names like Daimiel, Tomelloso, Hellín, and Santa Pola.

RIGHT PANEL (Console/Inspector/Tasks)



Tabs:

- Inspector: Click anywhere on the map to get pixel values (not shown here).
- Console (Active): Displays logs and charts from `print()` or `ui.Chart`.
 - Example shown:
 - Annual Variation of LST in Alicante (MODIS) – line graph of seasonal LST trends from 2014–2015.
 - Mean LST Over Time (2014–2023) – time series of monthly or yearly LST values.
- Tasks: Where export jobs (e.g., downloading images/tables) are queued and monitored.



TOP TOOLBAR & CENTER PANEL (Script Editor)

The screenshot shows the Google Earth Engine (GEE) Script Editor interface. At the top is a toolbar with five buttons: "Get Link", "Save", "Run", "Reset", and "Apps". To the right of the toolbar is a gear icon for settings. Below the toolbar is a "New Script *" tab. The main area contains GEE JavaScript code. The code includes functions for loading shapefiles, filtering data for Alicante, adding layers to the map, applying scaling factors for Landsat 8 SR Collection 2, Tier 1, creating a cloud mask, and calculating LST for a given year. The code uses ee.FeatureCollection, ee.Filter.eq, Map.addLayer, Map.centerObject, image.select, image.multiply, image.addBands, image.updateMask, ee.Date.fromYMD, and ee.Date.

```
1 // Load the full Spain Level 2 shapefile
2 var spain = ee.FeatureCollection("projects/mymemeberlink/assets/gadm41_ESP_2");
3
4 // Filter for Alicante
5 var alicante = spain.filter(ee.Filter.eq('NAME_1', 'Alicante'));
6
7 // Add to map
8 Map.addLayer(alicante, {}, 'Alicante');
9 Map.centerObject(alicante, 9);
10
11 // =====
12 // Load Spain Level 2 shapefile and filter for Alicante
13 // =====
14 var spain = ee.FeatureCollection("projects/mymemeberlink/assets/gadm41_ESP_2");
15 var roi = spain.filter(ee.Filter.eq('NAME_1', 'Alicante'));
16
17 Map.addLayer(roi, {}, 'Alicante');
18 Map.centerObject(roi, 9);
19
20 // =====
21 // Function to Apply Scaling Factors for Landsat 8 SR Collection 2, Tier 1
22 // =====
23 function applyScaleFactors(image) {
24   var opticalBands = image.select('SR_B_').multiply(0.0000275).add(-0.2);
25   var thermalBands = image.select('ST_B.*').multiply(0.00341802).add(149.0);
26   return image.addBands(opticalBands, null, true)
27     .addBands(thermalBands, null, true);
28 }
29
30 // =====
31 // Function to Mask Clouds and Cloud Shadows
32 // =====
33 function cloudMask(image) {
34   var cloudShadowBitmask = (1 << 3);
35   var cloudBitmask = (1 << 5);
36   var qa = image.select('QA_PIXEL');
37   var mask = qa.bitwiseAnd(cloudShadowBitmask).eq(0)
38     .and(qa.bitwiseAnd(cloudBitmask).eq(0));
39   return image.updateMask(mask);
40 }
41
42 // =====
43 // Function to Calculate LST for a Given Year
44 // =====
45 function calculateLST(year) {
46   var startDate = ee.Date.fromYMD(year, 6, 1);
47   var endDate = ee.Date.fromYMD(year, 9, 21);
```

TOP TOOLBAR ("Get Link" / "Save" / "Run" / "Reset" / "Apps" / "Settings")

- **Get Link:** Share your script via a unique link.
- **Save:** Saves the script you're working on.
- **Run:** Executes the script and processes data.
- **Reset:** Clears the results or layers from the map.
- **Apps:** Lets you deploy your GEE script as a web app.
- **Settings:** Toggle code suggestions, autocomplete, and syntax highlighting.

Script Tabs ("New Script*")

- This is the Code Editor, where you write and modify your GEE JavaScript.
- You can have multiple scripts open in tabs like a browser.

LEFT PANEL (Titled "Scripts")

The screenshot shows the Google Earth Engine interface with the left panel titled "Scripts". The top navigation bar includes tabs for "Scripts", "Docs", and "Assets". Below the navigation is a search bar labeled "Filter scripts..." and a red "NEW" button with a dropdown arrow. A refresh icon is also present. The main content area is organized into sections: "Owner (1)" which lists "users/maahmoudsm1/DataViz" containing "Alicante-Spain" and "Spain"; "Writer" which states "No accessible repositories. Click Refresh to check again"; "Reader" which states "No accessible repositories. Click Refresh to check again"; "Archive" which states "No accessible repositories. Click Refresh to check again"; and "Examples".

This is the Scripts panel where your project files are listed.

Owner / Writer / Reader / Archive / Examples

- Owner: Shows your saved scripts and assets (like shapefiles).
 - Here, you can see **Alicante-Spain** and **Spain** shapefiles under **users/maahmoudsm1/DataViz**.
- Writer: Projects where you're a collaborator.
- Reader: Projects you can view but not edit.
- Archive: Older scripts saved automatically.
- Examples: Sample code snippets and datasets from Google Earth Engine.

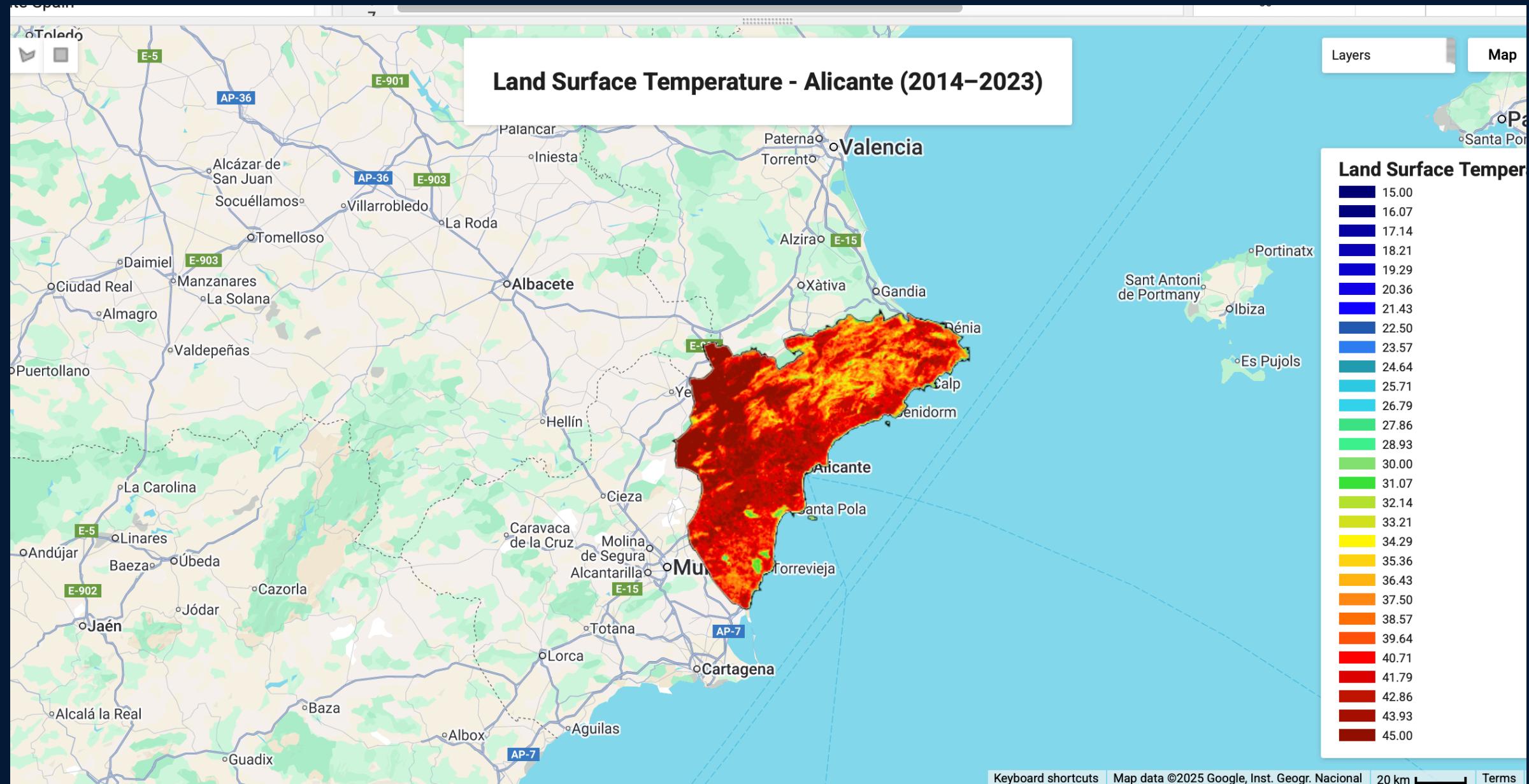
Docs – Gives you access to Earth Engine's documentation, including:

- Code snippets
- API references
- Examples for different use cases (NDVI, LST, change detection)

Assets – Where you upload or manage your personal data like:

- Shapefiles (boundaries)
- Image assets (custom rasters)
- Tables (CSV files, point data, etc.)

Map Viewer & Tools



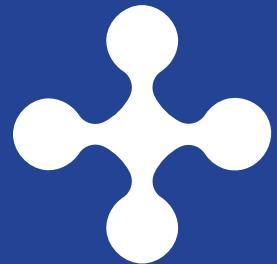
Map Viewer

- Displays the spatial results of your analysis (e.g., NDVI, LST, boundaries).
- The map shows: LST raster heatmap over Alicante (with a color scale from 26–44°C).
- Labels like “Land Surface Temperature – Alicante (2014–2023)”.
- You can zoom, pan, and inspect values using the Inspector.

Map Tools:

- Layers button (right side): Toggle visibility, change opacity, view legend (like the temperature color scale shown).
- Capture / Options (center bottom): Take a screenshot of your map view.
- Zoom / Reset / Drag (left side tools): Navigate the map area.





Code Explanation and Instructions

This section details the step-by-step process for data analysis

Accessing Earth Engine Data

Learn how to **efficiently retrieve** satellite data for your analysis

Processing Land Surface Temperature

Step-by-step guidance on **calculating** Land Surface Temperature using GEE

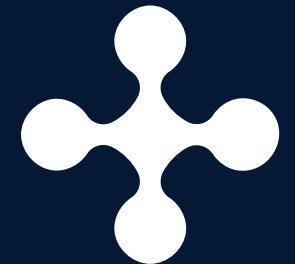
Analyzing NDVI Changes

Instructions for computing and **interpreting NDVI** over the specified timeframe



STEP 1 IMPORTS & ROI

- IMPORT SHAPEFILE
- IDENTIFY (REGION OF INTREST) ROI



STEP 1 IMPORT REGION

```
1 // Load the full Spain Level 2 shapefile
2 var spain = ee.FeatureCollection("projects/mymemeberlink/assets/gadm41_ESP_2");
3
4 // Filter for Alicante
5 var alicante = spain.filter(ee.Filter.eq('NAME_1', 'Alicante'));
```

What's Happening Here?

- We load the Spain Level 2 shapefile in GEE. (If GEE doesn't have the specific city or region, we may need to download the shapefile externally (e.g. from GADM or GeoBoundaries), then upload it manually to Earth Engine Assets.)
- It's loaded as a FeatureCollection in Earth Engine.
- Then we filter it to keep only the Alicante region.

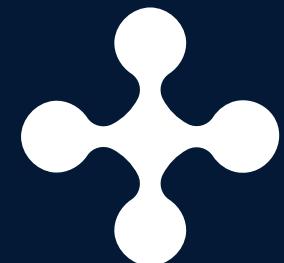
Why?

This defines our Region of Interest (ROI) so we can:

- Focus all analysis only on Alicante
- Clip satellite images to that boundary
- Calculate NDVI/LST within this region only

STEP 2 DATA AND PREPROCESS

- LOAD LANDSAT 8 AND MODIS
- APPLY SCALING FACTORS
- APPLY CLOUD AND SHADOW MASKING



STEP 2 - Add the filtered Alicante to the Map and center the view

```
7 // Add to map  
8 Map.addLayer(alicante, {}, 'Alicante');  
9 Map.centerObject(alicante, 9);
```

Explanation:

- **Map.addLayer(...)**: Adds the filtered Alicante region as a visible boundary layer on the map.
- **{}**: This is the styling object (left empty here, so it uses defaults).
- **'Alicante'**: The name of the layer shown in the Layers panel.
- **Map.centerObject(alicante, 9)**: Automatically centers and zooms the map on Alicante with zoom level 9 (a decent city-level zoom).



STEP 2 - Apply scaling factors to Landsat 8 bands

```
20 // =====
21 // Function to Apply Scaling Factors for Landsat 8 SR Collection 2, Tier 1
22 // =====
23 function applyScaleFactors(image) {
24   var opticalBands = image.select('SR_B_').multiply(0.0000275).add(-0.2);
25   var thermalBands = image.select('ST_B.*').multiply(0.00341802).add(149.0);
26   return image.addBands(opticalBands, null, true)
27     .addBands(thermalBands, null, true);
28 }
```

Explanation:

- Landsat 8 Collection 2 Level 2 images are delivered as scaled integers, not reflectance/temperature directly.
- This function rescales them to real-world units:
 - Optical bands (**SR_B.**) → Surface Reflectance (0–1)
 - Thermal band (**ST_B10**) → Brightness Temperature in Kelvin
- **multiply(...)** and **add(...)** are based on Landsat documentation.
- **addBands(..., null, true)**: Adds the rescaled bands replacing the originals.



STEP 2 Cloud and shadow mask

```
30 // =====
31 // Function to Mask Clouds and Cloud Shadows
32 // =====
33 function cloudMask(image) {
34   var cloudShadowBitmask = (1 << 3);
35   var cloudBitmask = (1 << 5);
36   var qa = image.select('QA_PIXEL');
37   var mask = qa.bitwiseAnd(cloudShadowBitmask).eq(0)
38     .and(qa.bitwiseAnd(cloudBitmask).eq(0));
39   return image.updateMask(mask);
40 }
```

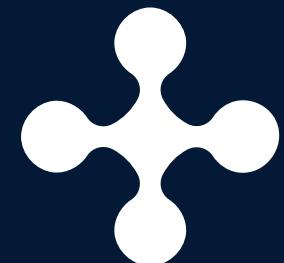
Explanation:

- Removes clouds and shadows using the **QA_PIXEL** quality band.
- **bitwiseAnd(...)** checks specific bits:
 - Bit 3 → Shadow
 - Bit 5 → Cloud
- If both bits are 0, it means clear pixels, so we keep them.
- **updateMask(mask)**: Applies the mask so only clear pixels are used in calculations.



STEP 3 CALCULATE NDVI AND LST

- CALCULATE NDVI VEGETATION
- CALCULATE LST TEMPERATURE
- GENERATE TRUE COLOR IMAGE



STEP 3 - Load & Preprocess Landsat 8 Images

```
42 // =====
43 // Function to Calculate LST for a Given Year
44 // =====
45 function calculateLST(year) {
46   var startDate = ee.Date.fromYMD(year, 6, 1);
47   var endDate = ee.Date.fromYMD(year, 9, 21);
48
49   var image = ee.ImageCollection("LANDSAT/LC08/C02/T1_L2")
50     .filterBounds(roi)
51     .filterDate(startDate, endDate)
52     .map(applyScaleFactors)
53     .map(cloudMask)
54     .median()
55     .clip(roi);
56 }
```

Explanation:

- Loads Landsat 8 Surface Reflectance Tier 1 Level 2 imagery.
- Filters it by location (**roi**, i.e., Alicante) and summer months (June 1 to Sept 21).
- Applies the scale factors and cloud masking.
- Computes the **median image** to get a clean composite.
- Clips it to the shape of Alicante.



STEP 3 - NDVI Calculation for Each Year (Main Loop)

```
57 var ndvi = image.normalizedDifference(['SR_B5', 'SR_B4']).rename('NDVI');
58
59 var ndviMin = ee.Number(ndvi.reduceRegion({
60   reducer: ee.Reducer.min(),
61   geometry: roi,
62   scale: 30,
63   maxPixels: 1e9
64 }).values().get(0));
65
66 var ndviMax = ee.Number(ndvi.reduceRegion({
67   reducer: ee.Reducer.max(),
68   geometry: roi,
69   scale: 30,
70   maxPixels: 1e9
71 }).values().get(0));
72
73 var pv = ((ndvi.subtract(ndviMin)).divide(ndviMax.subtract(ndviMin)))
74   .pow(ee.Number(2))
75   .rename('PV');
76
77 var em = pv.multiply(0.004).add(0.986).rename('EM');
78
79 var thermal = image.select('ST_B10').rename('thermal');
80
81 var lst = thermal.expression(
82   '(TB / (1 + (0.00115 * (TB / 1.438)) * log(em))) - 273.15', {
83     'TB': thermal.select('thermal'),
84     'em': em
85   }).rename('LST Alicante ' + year);
86
87 // Visualizations
88 Map.addLayer(image, {bands: ['SR_B4', 'SR_B3', 'SR_B2'], min: 0.0, max: 0.15}, 'True Color 432 - ' + year);
89 Map.addLayer(ndvi, {min: -1, max: 1, palette: ['blue', 'white', 'green']}, 'NDVI Alicante - ' + year);
90 Map.addLayer(lst, {
91   min: 15, max: 45,
92   palette: [
93     '040274', '040281', '0502a3', '0502b8', '0502ce', '0502e6', '0602ff', '235cb1', '307ef3', '269db1', '30c8e2', '32d3ef',
94     '3be285', '3ff38f', '86e26f', '3ae237', 'b5e22e', 'd6e21f', 'fff705', 'ffd611', 'ffb613', 'ff8b13', 'ff6e08', 'ff500d',
95     'ff0000', 'de0101', 'c21301', 'a71001', '911003'
96   ], 'LST - ' + year);
97
98
```

Calculates:

NDVI: Vegetation health (formula: (NIR-Red)/(NIR+Red))

LST: Land surface temp (requires NDVI-based emissivity, then radiative transfer eqn.)

True Color: Natural-color composite (bands 4,3,2)

Adds all three to the map for visualization.

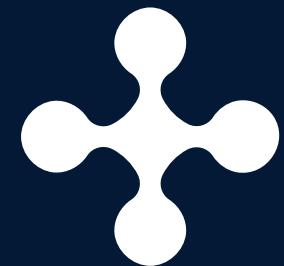
Why:

- Computes vegetation greenness. High NDVI → dense vegetation, low NDVI → bare soil or built-up area.
- This gives you year-by-year maps of vegetation and temperature—good for studying trends and visualizing environmental changes.



STEP 4 MULTI YEAR ANALYSIS

- PROCESS EACH YEAR 2014 TO 2023
- TRACK TRENDS OVER TIME
- Legends & Map Title
- MODIS LST Time Series and Mean Map



STEP 4 - Multi-Year Analysis Loop, Legends & Map Title

```
138 // =====
139 // Loop for 10 years
140 // =====
141 var years = [2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023];
142 years.forEach(function(year) {
143   calculateLST(year);
144 });
145
146 // =====
147 // LST Legend
148 // =====
149 var minLST = 15;
150 var maxLST = 45;
151 var palette = [
152   '040274', '040281', '0502a3', '0502b8', '0502ce', '0502e6', '0602ff', '235cb1', '307ef3', '269db1', '30c8e2', '32d3ef',
153   '3be285', '3ff38f', '86e26f', '3ae237', 'b5e22e', 'd6e21f', 'fff705', 'ffd611', 'ffb613', 'ff8b13', 'ff6e08', 'ff500d',
154   'ff0000', 'de0101', 'c21301', 'a71001', '911003'
155 ];
156 var step = (maxLST - minLST) / (palette.length - 1);
157
158 var legend = ui.Panel({style: {position: 'bottom-right', padding: '8px 15px', backgroundColor: 'white'}});
159 legend.add(ui.Label({
160   value: 'Land Surface Temperature (°C)',
161   style: {fontWeight: 'bold', fontSize: '16px', margin: '0 0 4px 0'}
162 }));
163
164 for (var i = 0; i < palette.length; i++) {
165   legend.add(ui.Panel({
166     widgets: [
167       ui.Label({style: {backgroundColor: '#' + palette[i], padding: '5px', margin: '0 0 5px 0', width: '30px'}}),
168       ui.Label({value: (minLST + i * step).toFixed(2), style: {fontSize: '10px', margin: '0 0 5px 6px'}})
169     ],
170     layout: ui.Panel.Layout.Flow('horizontal')
171   }));
172 }
173 Map.add(legend);
174
```

Multi-Year Analysis Loop & Legends

Explantion:

- Loops through all years, calling your analysis function for each.
- Builds a visual legend for your temperature palette (helps interpret your LST maps).
- Adds a map title so exported screenshots/maps are labeled.

```
175 // =====
176 // Map Title
177 // =====
178 Map.add(ui.Panel([
179   ui.Label({
180     value: 'Land Surface Temperature - Alicante (2014-2023)',
181     style: {fontWeight: 'bold', fontSize: '20px', padding: '8px'}
182   })
183 ], ui.Panel.Layout.Flow('vertical'), {
184   position: 'top-center'
185 }));
186
```

Map Title

Why:

- So you can monitor change over time essential for trend analysis.
- For clarity, especially when sharing with non-experts.

STEP 4 - MODIS LST Time Series and Mean Map

```
187 // =====
188 // MODIS LST Time Series Chart (2014–2023)
189 // =====
190 var collection = ee.ImageCollection('MODIS/061/MOD11A2')
191   .select("LST_Day_1km")
192   .filterDate('2014-01-01', '2023-12-31')
193   .filterBounds(roi);
194
195 var LSTDay = collection.map(function(img) {
196   return img.multiply(0.02).subtract(273.15)
197   .copyProperties(img, ['system:time_start', 'system:time_end']);
198 });
199
200 // Chart by Day of Year
201 var chart = ui.Chart.image.doySeriesByYear(
202   LSTDay, 'LST_Day_1km', roi, ee.Reducer.mean(), 1000)
203   .setOptions({
204     title: 'Annual Variation of LST in Alicante (MODIS)',
205     vAxis: {title: 'Temperature (°C)'},
206     hAxis: {title: 'Day of Year'}
207   });
208 print(chart);
209
210 // =====
211 // Mean LST Map from MODIS (2014–2023)
212 // =====
213 Map.addLayer(LSTDay.mean().clip(roi), {
214   min: 20.0,
215   max: 40.0,
216   palette: palette
217 }, 'Mean LST MODIS (2014–2023)');
218
```

Explantion:

- Loads MODIS LST (lower res, more frequent: every 8 days)
- Converts to Celsius, builds time series chart, and shows mean LST map for the whole period

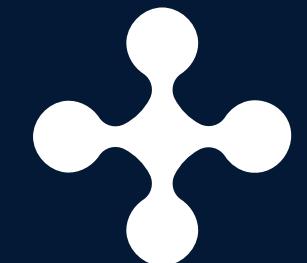
Why:

- Landsat = high spatial resolution, but less frequent.
- MODIS = lower spatial, but better for temporal (trend) analysis.
- Lets you see “big picture” trends and validate your Landsat findings.



STEP 5 EXPORT RESULTS

- CALCULATE NDVI VEGETATION
- CALCULATE LST TEMPERATURE
- GENERATE TRUE COLOR IMAGE



STEP 4 - MODIS LST Time Series and Mean Map

```
99 // ===== EXPORTS: Add here for each year =====
100
101 // Export NDVI
102 Export.image.toDrive({
103   image: ndvi,
104   description: 'NDVI_Alicante_' + year,
105   folder: 'GEE_Exports',
106   fileNamePrefix: 'NDVI_Alicante_' + year,
107   region: roi.geometry(),
108   scale: 30,
109   maxPixels: 1e13,
110   fileFormat: 'GeoTIFF'
111 });
112
113 // Export LST
114 Export.image.toDrive({
115   image: lst,
116   description: 'LST_Alicante_' + year,
117   folder: 'GEE_Exports',
118   fileNamePrefix: 'LST_Alicante_' + year,
119   region: roi.geometry(),
120   scale: 30,
121   maxPixels: 1e13,
122   fileFormat: 'GeoTIFF'
123 });
124
125 // Export True Color (432)
126 Export.image.toDrive({
127   image: image.select(['SR_B4', 'SR_B3', 'SR_B2']),
128   description: 'TrueColor_Alicante_' + year,
129   folder: 'GEE_Exports',
130   fileNamePrefix: 'TrueColor_Alicante_' + year,
131   region: roi.geometry(),
132   scale: 30,
133   maxPixels: 1e13,
134   fileFormat: 'GeoTIFF'
135 });
136
137 }
```

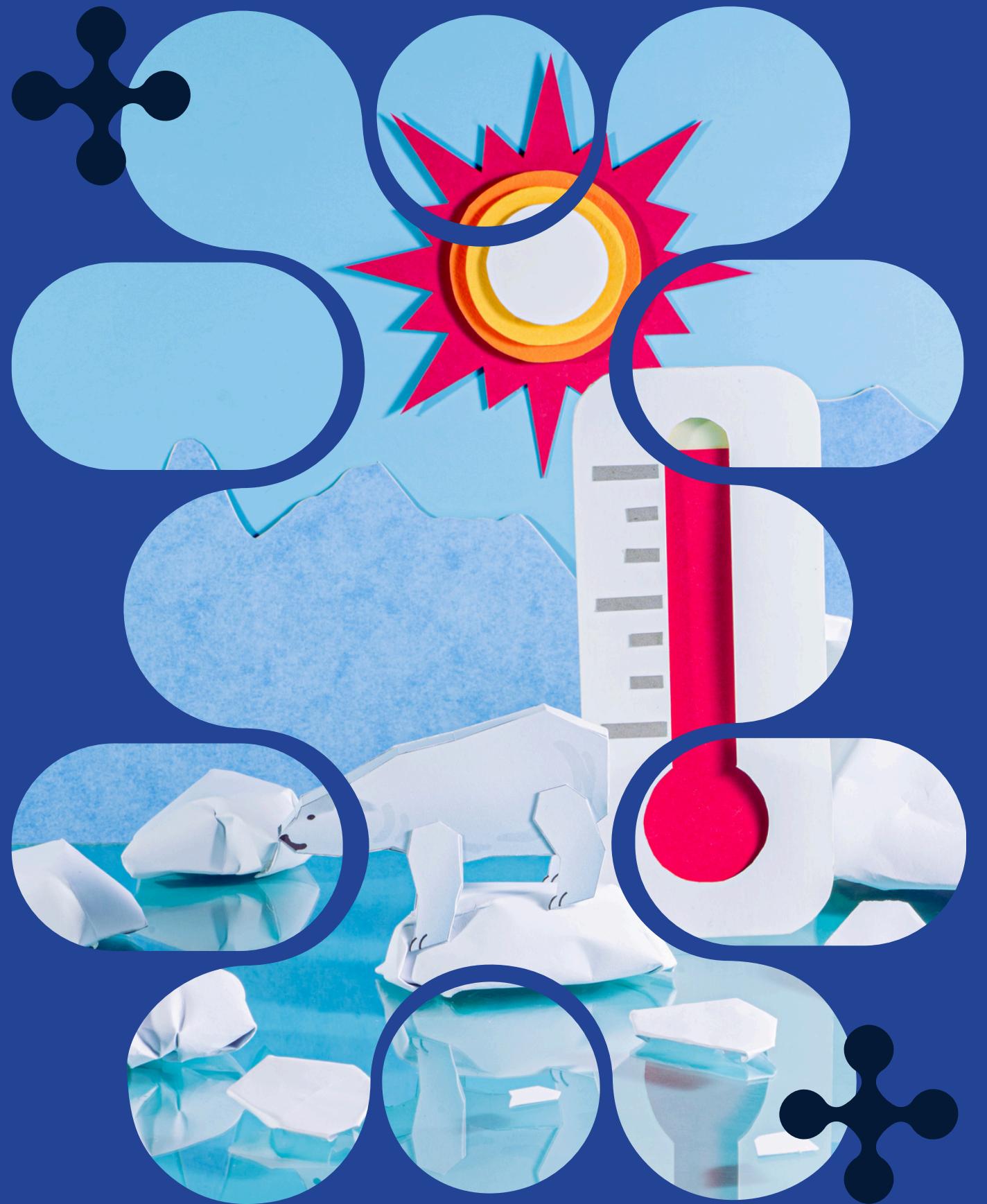
```
219 // ===== EXPORT: MODIS MEAN LST (2014–2023) =====
220 Export.image.toDrive({
221   image: LSTDay.mean().clip(roi),
222   description: 'MODIS_LST_Mean_2014_2023',
223   folder: 'GEE_Exports',
224   fileNamePrefix: 'MODIS_LST_Mean_2014_2023',
225   region: roi.geometry(),
226   scale: 1000,
227   maxPixels: 1e13,
228   fileFormat: 'GeoTIFF'
229 });
230
231 // ===== EXPORT: MODIS LST Time Series Chart Data =====
232 var timeseries = LSTDay.map(function(image) {
233   var date = ee.Date(image.get('system:time_start')).format('YYYY-MM-dd');
234   var mean = image.reduceRegion({
235     reducer: ee.Reducer.mean(),
236     geometry: roi,
237     scale: 1000,
238     maxPixels: 1e13
239   }).get('LST_Day_1km');
240   return ee.Feature(null, {'date': date, 'mean_LST': mean});
241 });
242 var timeseries_fc = ee.FeatureCollection(timeseries);
243
244 Export.table.toDrive({
245   collection: timeseries_fc,
246   description: 'MODIS_LST_Timeseries_Alicante_2014_2023',
247   folder: 'GEE_Exports',
248   fileNamePrefix: 'MODIS_LST_Timeseries_Alicante_2014_2023',
249   fileFormat: 'CSV'
250 });
```

Explantion:

- Sends your selected map layer (NDVI, LST, TrueColor, mean, Time Series Chart Data) to Google Drive as a GeoTIFF.

Why:

- Needed for using your results in QGIS/ArcGIS, reporting, sharing, or further analysis outside GEE.



8 out of 10

Majority of areas show increased temperature trends.

15%

Significant rise in NDVI values observed overall.

3.2°C increase

Average rise in Land Surface Temperature over the decade.

INSIGHTS & APPLICATIONS

- NDVI maps: Identify areas of healthy/dry/degraded vegetation. Spot drought, crop patterns, or urban expansion.
- LST maps: Detect heat islands, track climate/urbanization impacts, support planning for heat resilience.
- Time series: Understand long-term environmental trends and seasonality. Validate against field data or reports.



Why Use GEE for This?

- Handles big data (years, many images) fast
- Cloud-based—no downloads required to process
- Exports standard GIS formats
- Easy to automate, repeat, and share
- All satellite imagery is updated, pre-archived, and accessible

THANK YOU

