

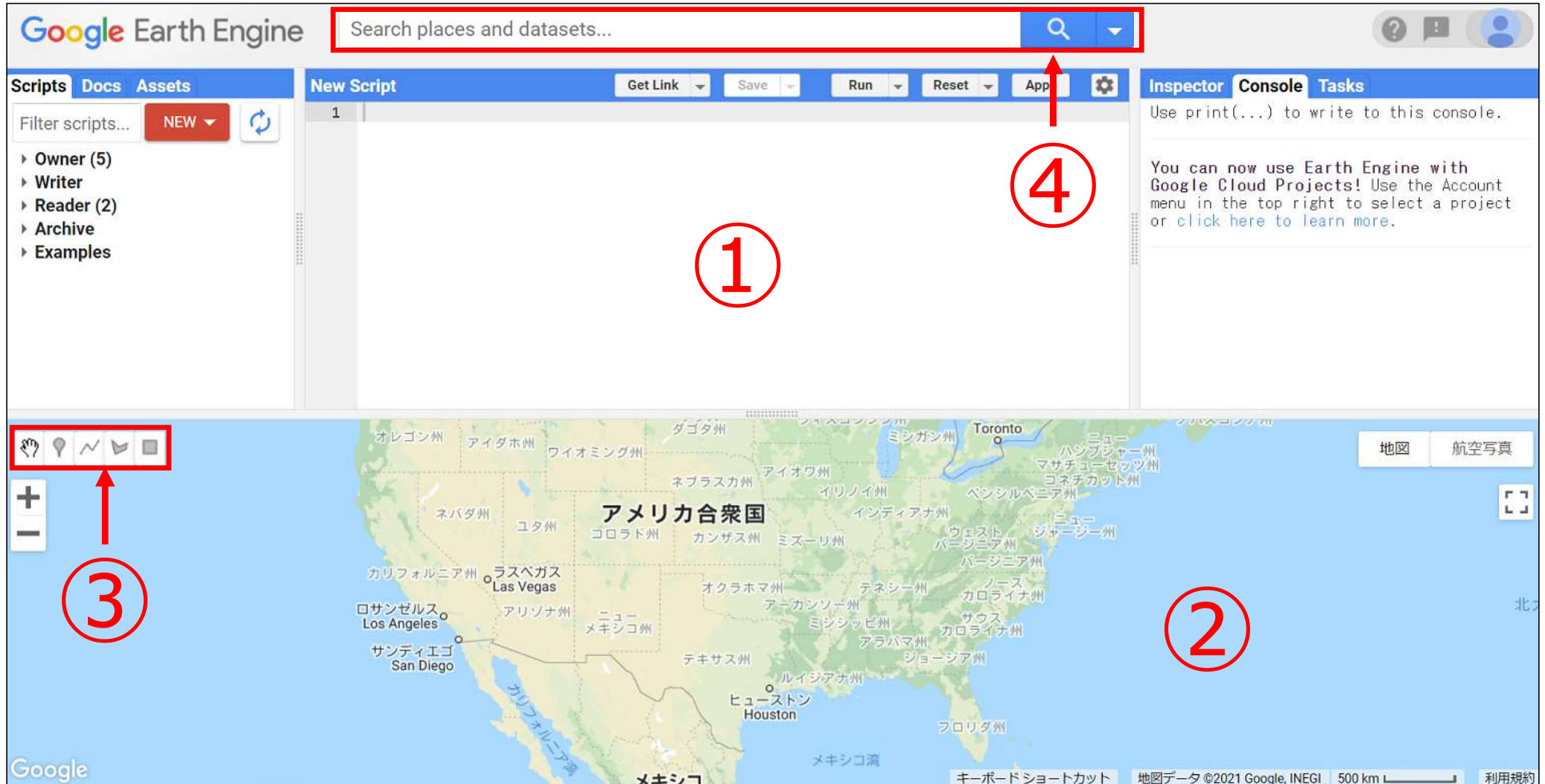
Introduction of the Basic Handling of Satellite Data (Nighttime Light Data)

Section Composition

1. Google Earth Engine (GEE) Main Screen
2. Prerequisite Knowledge for Programming in GEE
3. How to Handle Administrative Boundaries Data
4. How to Handle Nighttime Light Data
5. Additional Tips
6. Reference Materials for Studying Nighttime Light Data

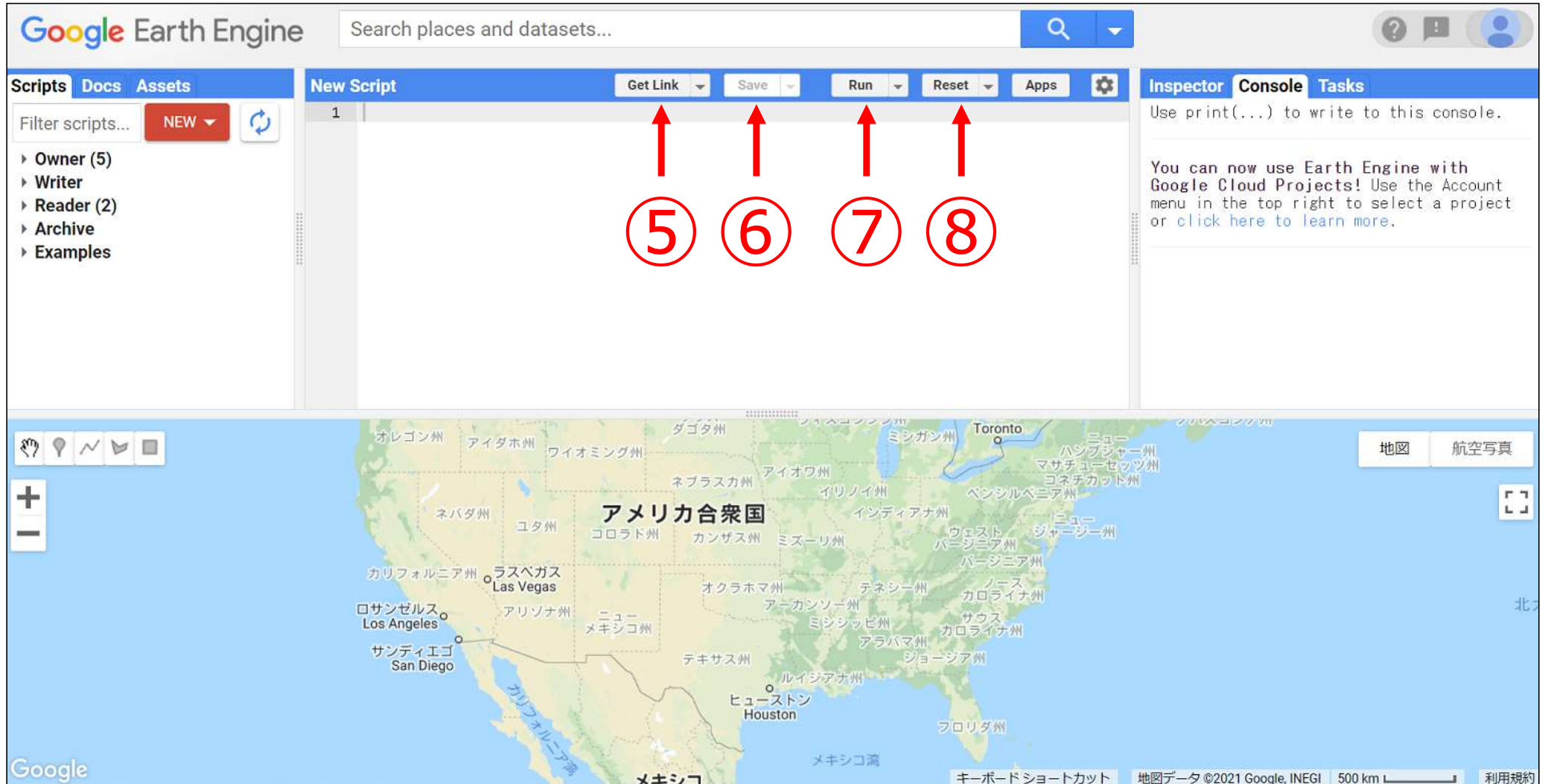
Google Earth Engine Main Screen

Google Earth Engine (GEE) Main Screen 1/2



- ① Code editor: Where you write the code
- ② Map: Where the satellite data is displayed
- ③ Geometry tool: Tool used to draw points, lines, and figures on a map
- ④ Search field: Where you enter text to search to find satellite data stored in GEE

Google Earth Engine (GEE) Main Screen 2/2



- ⑤ Get Link: Generates a link for sharing the code with other people
- ⑥ Save: Saves the program/script written
- ⑦ Run: Runs the program/script written (alternatively, press Ctrl + Enter.)
- ⑧ Reset: Deletes the program/script written

Prerequisite Knowledge for Programming in GEE

Programming Language Used in GEE

- **JavaScript** is adopted as GEE's programming language. JavaScript is not a statistical programming language like R, but rather a language used to develop websites and web apps.
- On the other hand, JavaScript used in GEE is not exactly the same as JavaScript used in website and web app development. It is important to note that it is **only based on JavaScript**.
- GEE provides a variety of JavaScript-based GEE-specific codes. One example is **Map.addLayer**, which displays data on a map and is a code introduced in this document.

How to write codes

- The basic description to use for GEE programming is as follows: “**var**” creates new variables.

var variable name = number, data, etc.

- For example, if you want to create a variable “abc ” containing numerical data “5,” you would write the code as follows:

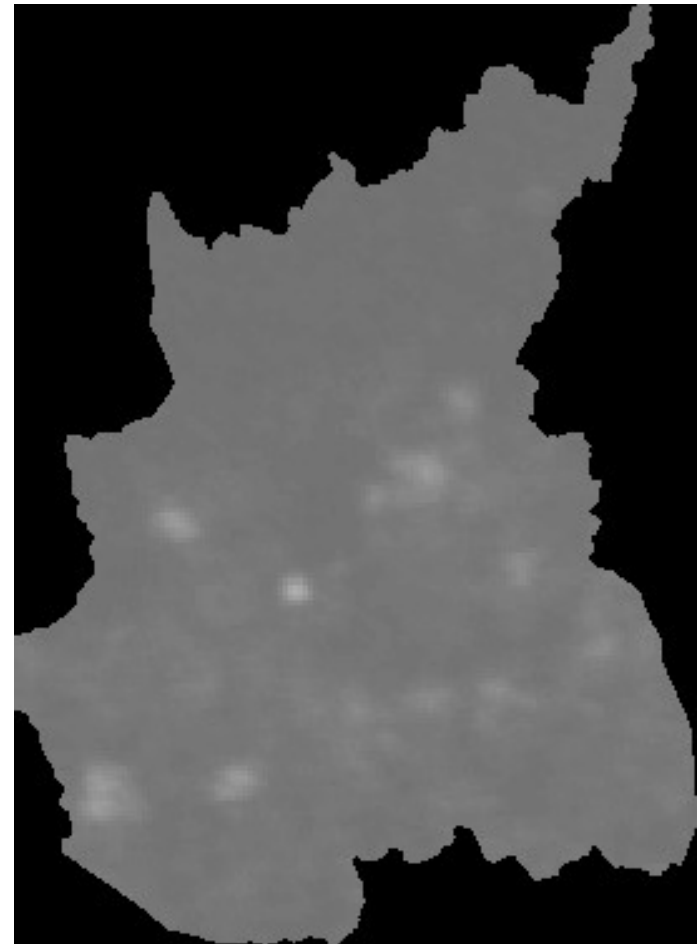
var abc = 5

- This document introduces a variety of codes. Bear in mind that **a code does not work even if the spelling is incorrect**. Particularly beginners often produce errors due to spelling mistakes. Therefore, if an error occurs, it is recommended that you check the spelling first.

How to Handle Administrative Boundaries Data

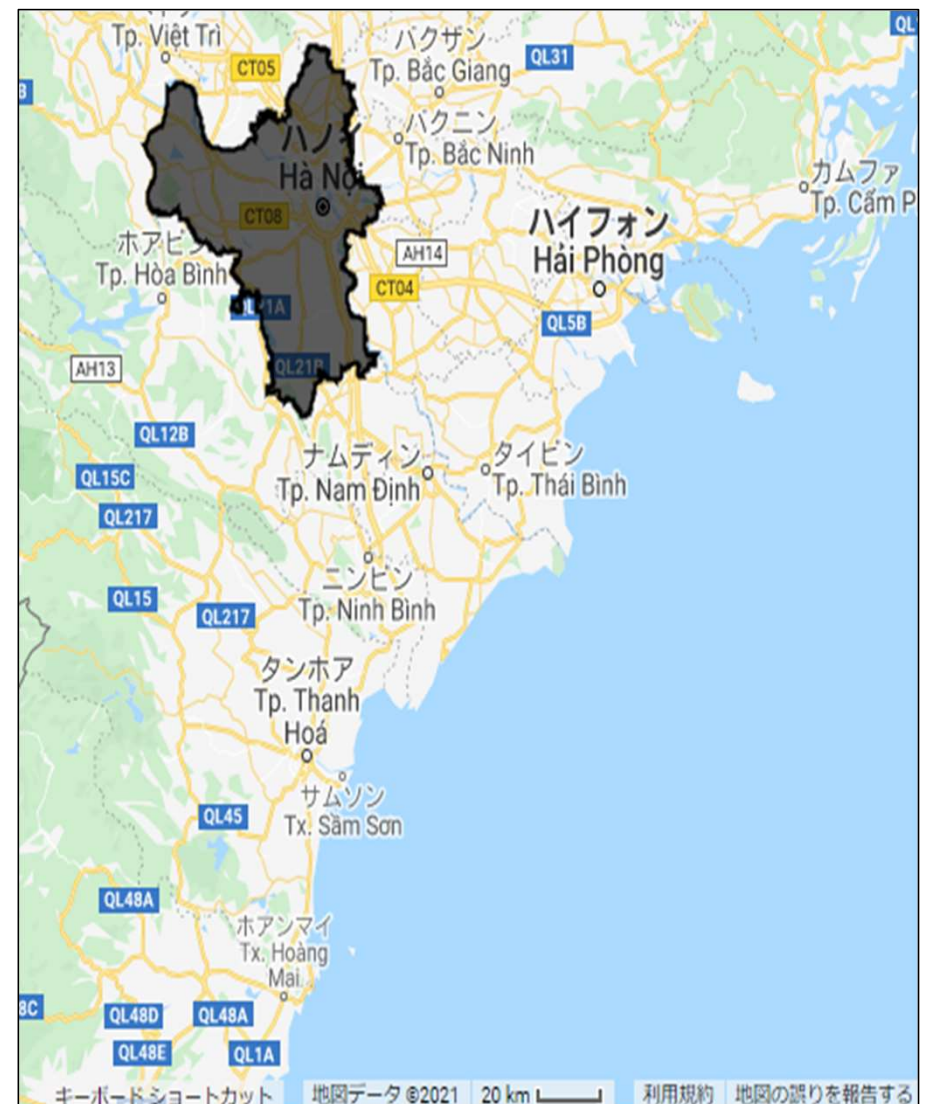
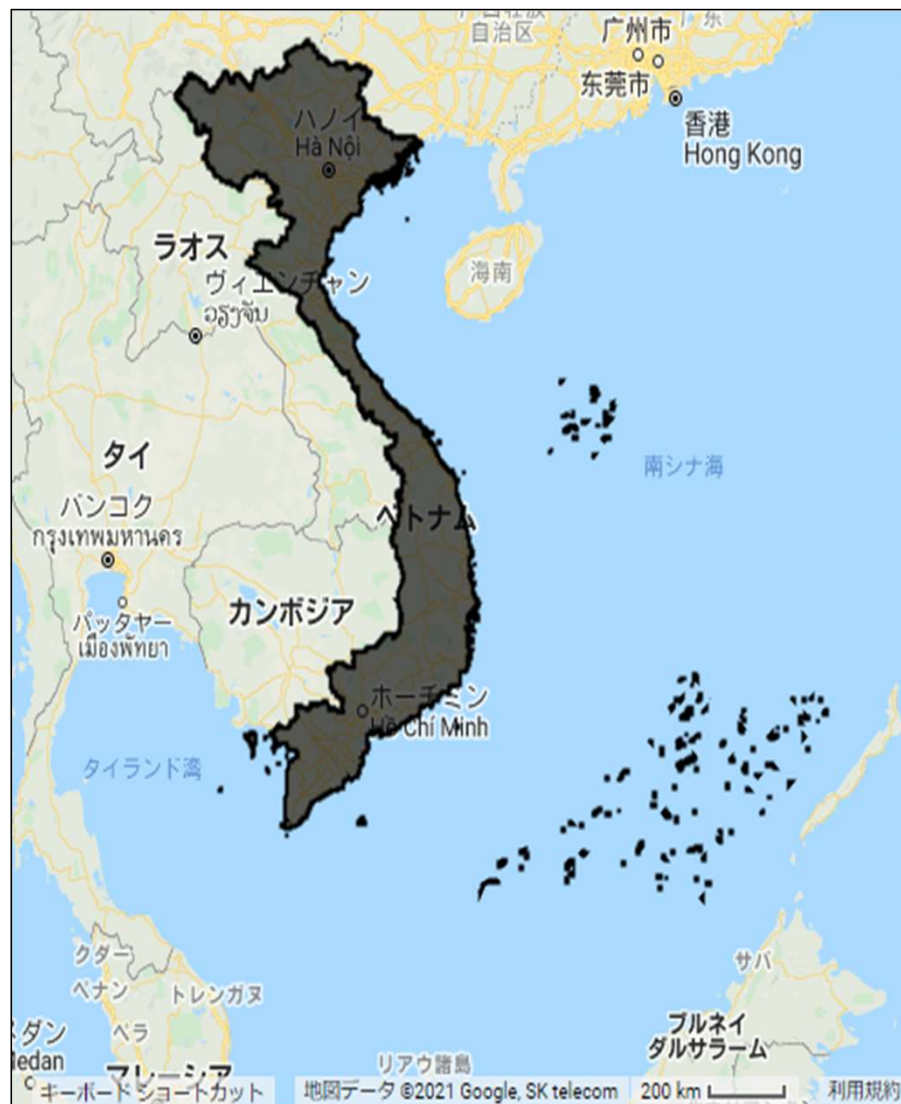
Necessity for Handling Administrative Boundaries Data

- When analyzing satellite data, it is often the case that you need to handle only a specific country or region. Therefore, it is useful to know how to upload administrative boundaries data to GEE and import the data from GEE, and how to map the data.
- For instance, one example of satellite data analysis conducted by the JICA Evaluation Department is the analysis of nighttime light data in Rattanakiri Province, Cambodia.



Final Goal of This Section

- Upload the administrative boundaries shapefiles of Vietnam and visualize (1) country level and (2) province level (Hanoi) on the map.

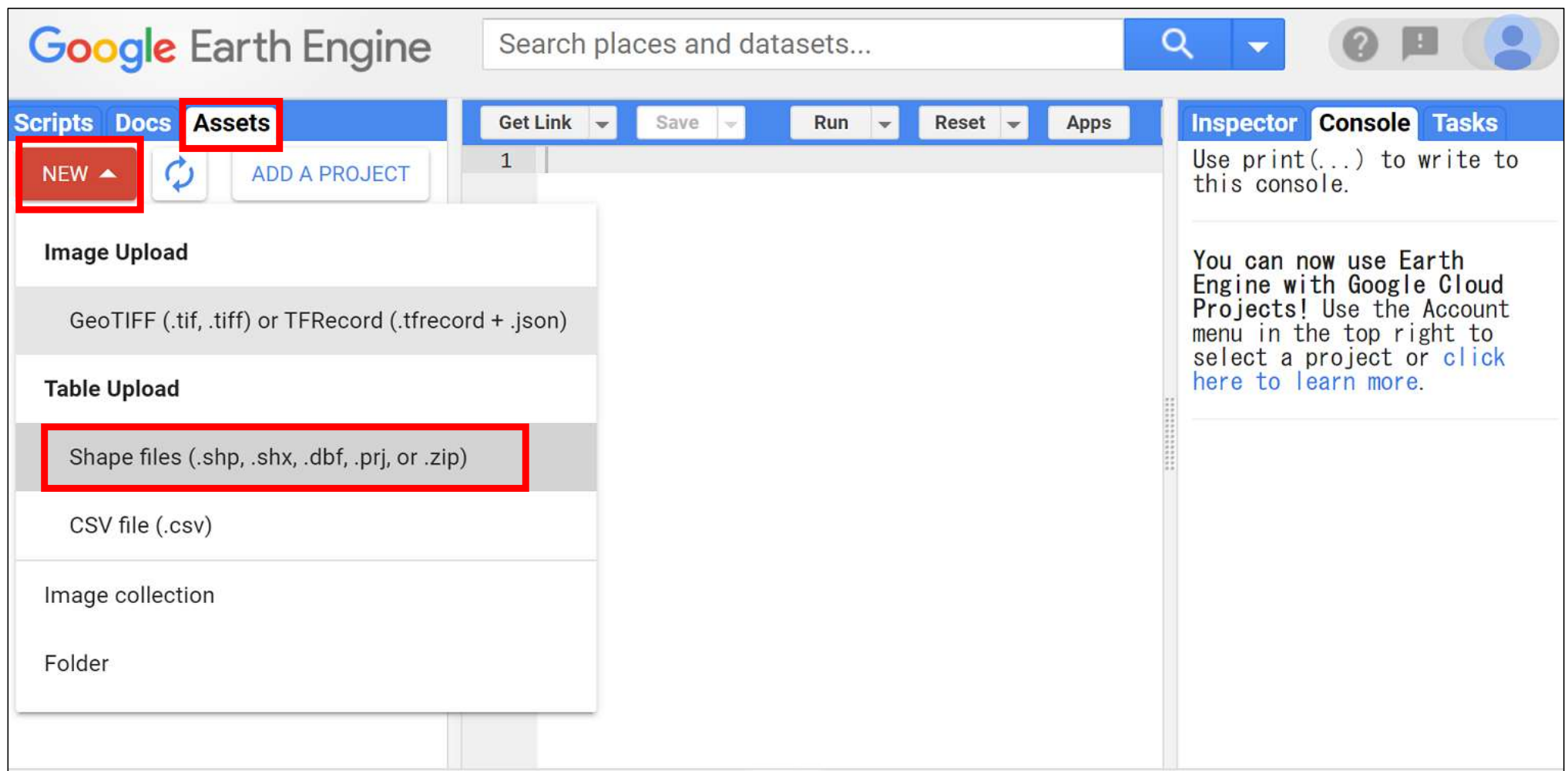


- ① Download the target administrative boundaries data from the Humanitarian Data Exchange (<https://data.humdata.org/>), operated by the United Nations Office for the Coordination of Humanitarian Affairs (OCHA).
 - For the purpose of this document, download the administrative boundaries data for Vietnam from the following URL (<https://data.humdata.org/dataset/viet-nam-administrative-boundaries-polygon-polyline>).
 - The file you should download is as follows, containing level 0 (country), level 1 (province), and level 2 (district) shapefiles:

vnm_adm_gov_20201027_shp.zip

Uploading Administrative Boundaries Data (Country Level) 2/4

- ② Click **Assets** in the left pane of GEE to display the **NEW** button and click it. A drop-down list appears. Click **Shape files** located in the middle.



Uploading Administrative Boundaries Data (Country Level) 3/4

- ③ The figure to the right is displayed in the center pane. Click **SELECT** and select the files you downloaded earlier. You should select four files with the extensions: **.dbf**, **.prj**, **.shp**, and **.shx**.
- ④ When you select a file, the **Asset Name** is automatically entered. Edit the name to any name you like.
- ⑤ Click **UPLOAD** button at the bottom to begin uploading administrative boundaries data.

The screenshot shows a web interface for uploading a new shapefile asset. It includes a 'Source files' section with a 'SELECT' button, a list of four files (vnm_admbnda_adm0_gov_20200103.dbf, .prj, .shp, .shx), an 'Asset ID' section with an 'Asset Name' field containing 'vietnam_lv10', a 'Properties' section with metadata information and buttons to add start time, end time, or properties, and an 'Advanced options' section with fields for character encoding (UTF-8), maximum error (1.0), and a checkbox for splitting large geometries. At the bottom, there is a 'CANCEL' button and a blue 'UPLOAD' button.

Upload a new shapefile asset

Source files ③

SELECT

Please drag and drop or select files for this asset.
Allowed extensions: shp, zip, dbf, prj, shx, cpq, fix, qix, sbn or shp.xml.

vnm_admbnda_adm0_gov_20200103.dbf	🗑️
vnm_admbnda_adm0_gov_20200103.prj	🗑️
vnm_admbnda_adm0_gov_20200103.shp	🗑️
vnm_admbnda_adm0_gov_20200103.shx	🗑️

Asset ID

users/devjpc consultant / ④

Asset Name
vietnam_lv10

Properties

Metadata properties about the asset which can be edited during asset upload and after ingestion. The "system:time_start" property is used as the primary date of the asset.

[Add start time](#) [Add end time](#) [Add property](#)

Advanced options

Character encoding
UTF-8 🔍 ⓘ

Maximum error
1.0 ⓘ

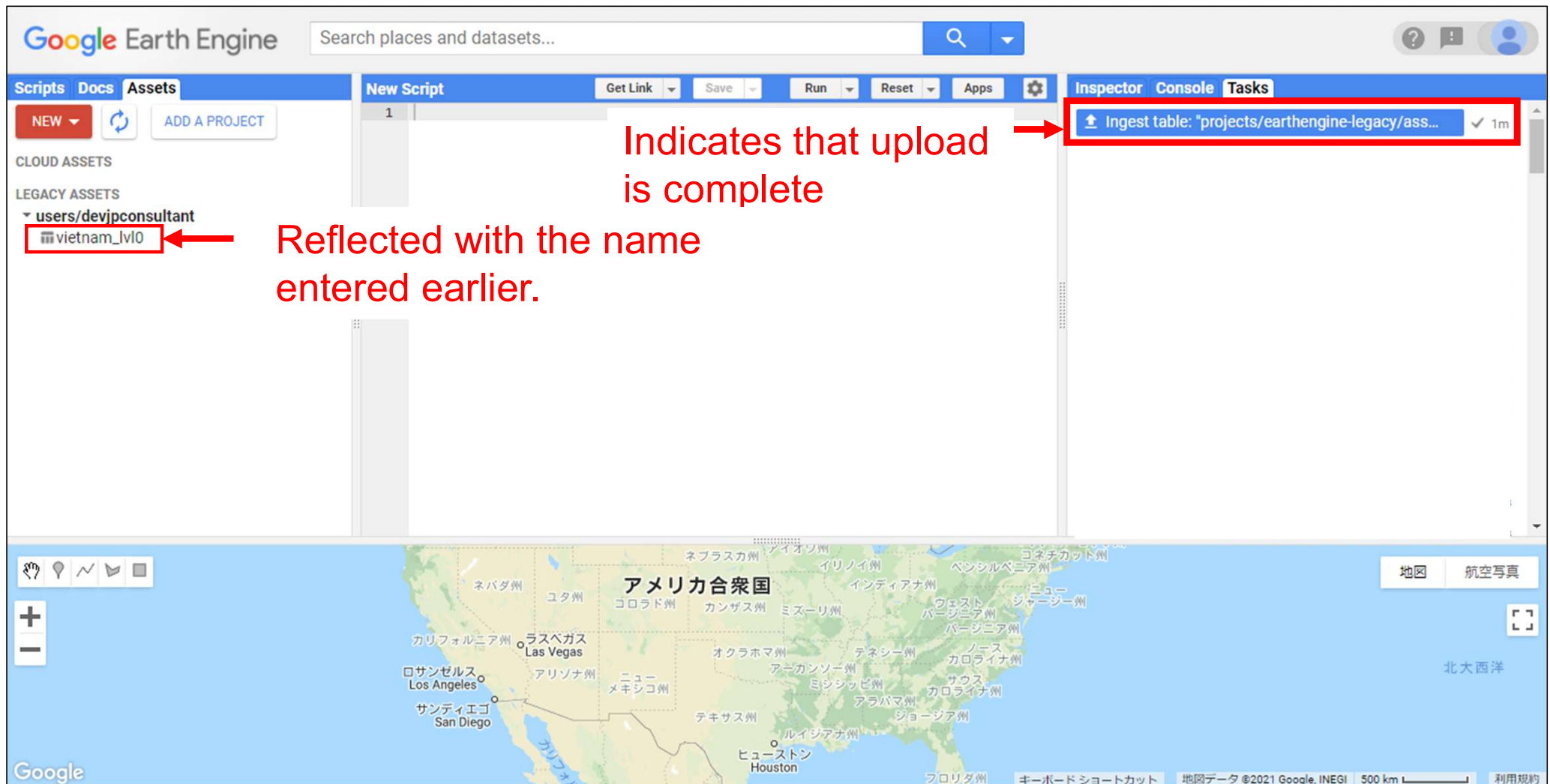
☐ Split large geometries ⓘ

[Learn more](#) about how uploaded files are processed.

[CANCEL](#) ⑤ **UPLOAD**

Uploading Administrative Boundaries Data (Country Level) 4/4

- ⑥ You can check the upload status in **Tasks** in the right pane. Once the upload is complete, the data is reflected in **Assets**.



Importing Administrative Boundaries Data (Country Level) 1/2


- ① Click the administrative boundaries data uploaded to **Assets** to view its basic information. You can import administrative boundaries data from **IMPORT** at the bottom right of the screen.

The screenshot shows a web interface for a data table named 'vietnam_lv10'. On the left, there is a map of Vietnam with a red square indicating a specific location. Below the map, the 'Table ID' is 'users/devjpconsultant/vietnam_lv10', the 'Date' is 'Start date: NA' and 'End date: NA', the 'File Size' is '1.42MB', the 'Number of Features' is '1', and the 'Last modified' date is '2021-08-28 06:21:46 UTC'. On the right, there are tabs for 'DESCRIPTION', 'FEATURES', and 'PROPERTIES'. The 'DESCRIPTION' tab is active, showing 'No description.' At the bottom right, there are four buttons: 'IMPORT', 'DELETE', 'SHARE', and 'CLOSE'. The 'IMPORT' button is highlighted with a red box and a red arrow pointing to it from the text 'Click here to import the data.'

Table: vietnam_lv10

DESCRIPTION FEATURES PROPERTIES

No description.

Table ID 
users/devjpconsultant/vietnam_lv10

Date
Start date: NA
End date: NA

File Size 1.42MB

Number of Features 1

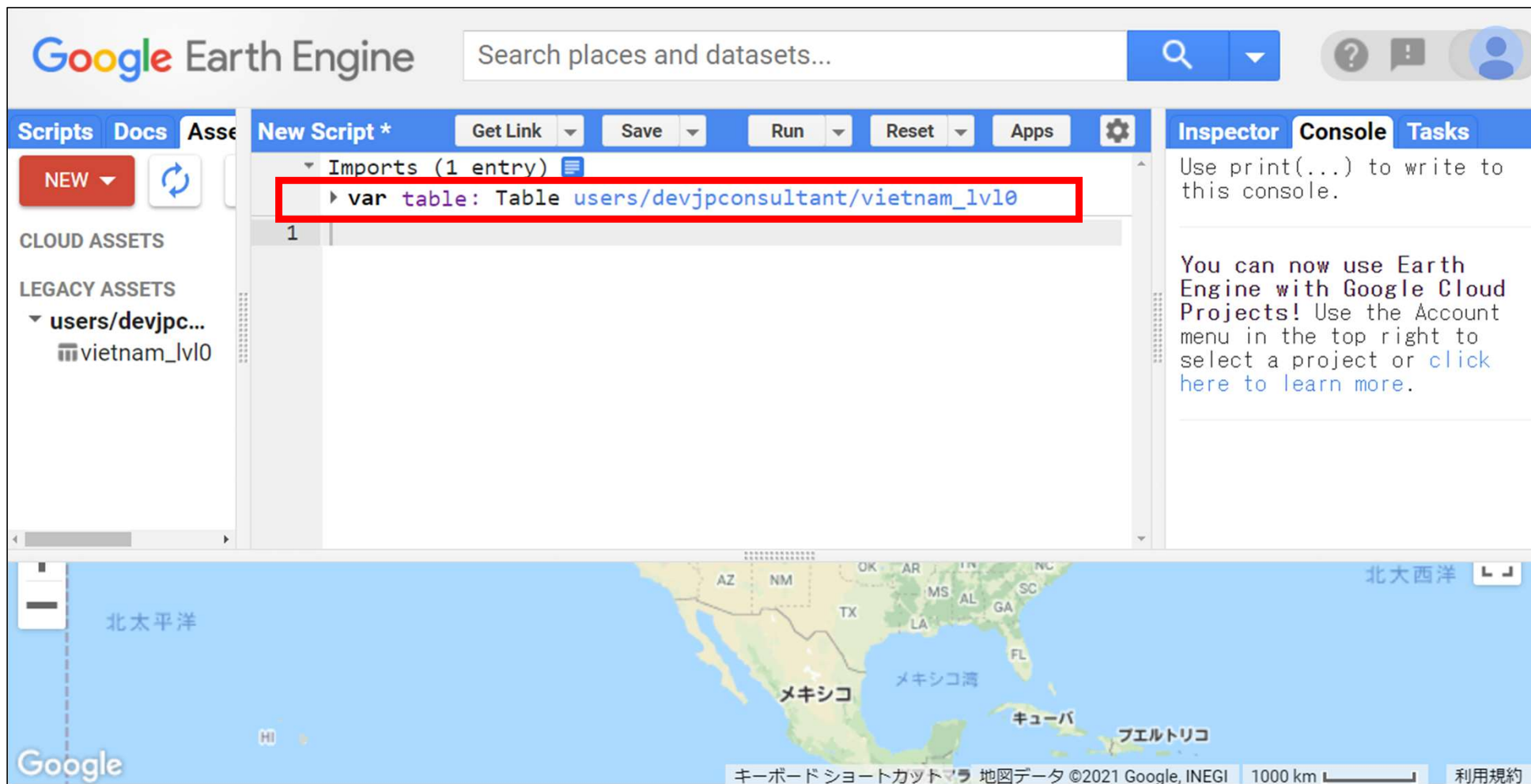
Last modified 2021-08-28 06:21:46 UTC

Click here to import the data.

IMPORT DELETE SHARE CLOSE

Importing Administrative Boundaries Data (Country Level) 2/2

- ② The imported administrative boundaries data is displayed in the center pane with the variable **table**. Click **table** if you want to rename the variable. In this document, the variable name is unchanged from “table.”



Visualizing Administrative Boundaries Data (Country Level) onto Map 1/2

- GEE provides **Map.addLayer()**; as the code to visualize an image onto the map.
- By using this code, you can visualize the image stored in the specified variable name as a layer on the map.
- Write **Map.addLayer(variable name, {}, “layer name”)**; to name the layer projected onto the map. If no layer name is given, a **layer number** is displayed.
 - The next slide shows the actual code.

Projecting Administrative Boundaries Data (Country Level) onto Map 2/2

The screenshot displays the Google Earth Engine web interface. At the top, the 'Google Earth Engine' logo is on the left, and a search bar with the text 'Search places and datasets...' is in the center. To the right of the search bar are icons for help, notifications, and a user profile. Below the search bar is a blue header bar with tabs for 'Scripts', 'Docs', and 'Assets'. The 'Scripts' tab is active, showing a 'New Script *' window. This window contains a code editor with the following script:

```
Imports (1 entry)
var table: Table users/devjpconsultant/vietnam_lv10

1 Map.centerObject(table, 4);
2
3 Map.addLayer(table);
4
5 Map.addLayer(table, {}, "ベトナム");
```

Red arrows point from explanatory text to specific lines of code:

- Line 1: `Map.centerObject(table, 4);` ← Code that moves the map to the specified location
- Line 3: `Map.addLayer(table);` ← Data is visualized onto the map as **Layer 1**.
- Line 5: `Map.addLayer(table, {}, "ベトナム");` ← Data is projected onto the map as **Vietnam**.

Below the script editor is a map of Southeast Asia. Vietnam is highlighted in black. The map includes labels for various countries and cities in Japanese and English. A 'Layers' panel is open on the right side of the map, showing two layers: 'ベトナム' (Vietnam) and 'Layer 1'. Both layers have a checkmark and a transparency slider. A red arrow points to the transparency slider for 'Layer 1'. A text box with a red border contains the text: 'You can change the transparency of the map by moving the bar.'

At the bottom of the map, there is a scale bar showing '500 km' and a copyright notice: '地図データ ©2021 Google, SK telecom'. The Google logo is in the bottom left corner.

Uploading Administrative Boundaries Data (Province Level) 1/2

- One step needs to be added after the last step of projecting the administrative boundaries data (country level) onto the map.
- ① In the right pane, change the file to be uploaded from **adm0** to **adm1**.
- ② When you select a file, the **Asset Name** is automatically entered. Edit the name to any name you like.
- ③ Click **UPLOAD** button at the bottom to begin uploading administrative boundaries data.

Upload a new shapefile asset

Source files

SELECT

①

Please drag and drop or select files for this asset.
Allowed extensions: shp, zip, dbf, prj, shx, cpq, fix, qix, sbn or shp.xml.

vnm_admbnda_adm1_gov_20201027.dbf	🗑
vnm_admbnda_adm1_gov_20201027.prj	🗑
vnm_admbnda_adm1_gov_20201027.shp	🗑
vnm_admbnda_adm1_gov_20201027.shx	🗑

Asset ID

users/devjpconsultant/

Asset Name

vietnam_lv1

②

Properties

Metadata properties about the asset which can be edited during asset upload and after ingestion. The "system:time_start" property is used as the primary date of the asset.

Add start time

Add end time

Add property

Advanced options

Character encoding

UTF-8

Maximum error

1.0

☐ Split large geometries

[Learn more](#) about how uploaded files are processed.

CANCEL

UPLOAD

③

Uploading Administrative Boundaries Data (Province Level) 2/2

- ④ You can check the upload status in **Tasks** in the right pane, and once the upload is complete, the data is reflected in **Assets**.

The screenshot displays the Google Earth Engine web interface. At the top, the 'Assets' tab is selected in the left sidebar. Under 'LEGACY ASSETS', the folder 'users/devjpconsultant' is expanded, showing two assets: 'vietnam_lv0' and 'vietnam_lv1'. The asset 'vietnam_lv1' is highlighted with a red box and a red arrow pointing to it from the text 'Reflected with the name entered earlier.'.

In the top right pane, the 'Tasks' tab is selected. It shows a single task: 'Ingest table: "projects/earthengi...' with a green checkmark and '3m' indicating completion. A red box highlights this task, and a red arrow points to it from the text 'Indicates that upload is complete'.

The bottom of the interface shows a map of North America with labels in Japanese for various states and cities. The Google logo is in the bottom left corner.

Importing Administrative Boundaries Data (Province Level) 1/2

- ① Click the administrative boundaries data uploaded to **Assets** to view its basic information. You can import the administrative boundaries data from **IMPORT** at the bottom right of the screen.


The screenshot shows a web interface for a data asset named 'Table: vietnam_lv1'. The interface has three tabs: 'DESCRIPTION', 'FEATURES', and 'PROPERTIES'. The 'DESCRIPTION' tab is active, showing 'No description.' and a map of Vietnam with province-level boundaries. A red arrow points to the map with the text 'The province level image differs from that of the country level.' Below the map, there is a 'Table ID' field with a copy icon, showing 'users/devjpconsultant/vietnam_lv1'. Below that, there are fields for 'Date' (Start date: NA, End date: NA), 'File Size' (3.01MB), 'Number of Features' (63), and 'Last modified' (2021-08-28 14:45:03 UTC). At the bottom right, there are four buttons: 'IMPORT', 'DELETE', 'SHARE', and 'CLOSE'. A red arrow points to the 'IMPORT' button with the text 'Click here to import data.'

Table: vietnam_lv1

DESCRIPTION FEATURES PROPERTIES

No description.

← The province level image differs from that of the country level.

Table ID 
users/devjpconsultant/vietnam_lv1

Date
Start date: NA
End date: NA

File Size 3.01MB

Number of Features 63

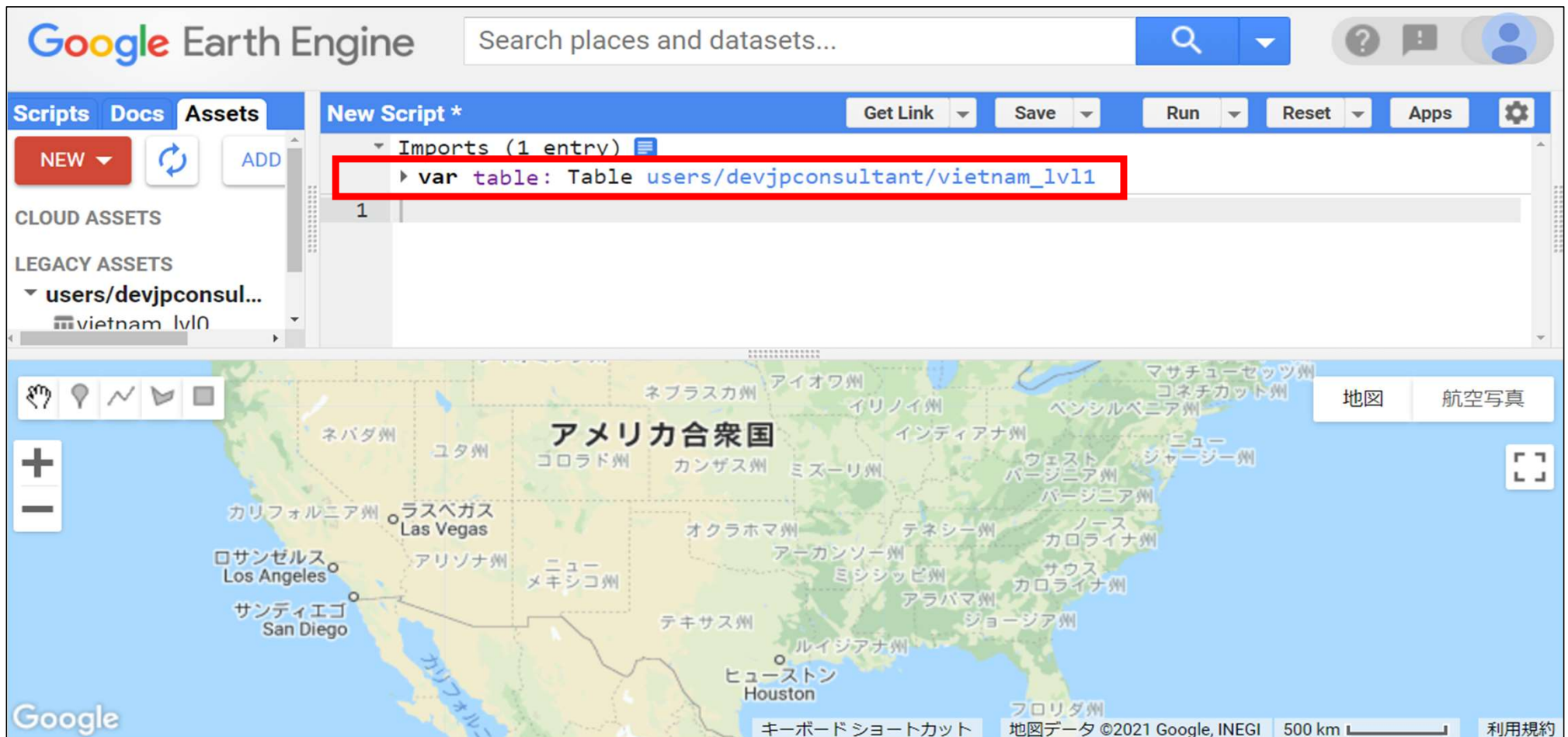
Last modified 2021-08-28 14:45:03 UTC

Click here to import data.

IMPORT DELETE SHARE CLOSE

Importing Administrative Boundaries Data (Province Level) 2/2

- ② The imported administrative boundaries data is displayed in the center pane with the variable **table**. Click **table** if you want to rename the variable. In this document, the variable name is unchanged from “table.”



Projecting Administrative Boundaries Data (Province Level) onto Map 1/4

- Likewise administrative boundaries data (country level), use the code **Map.addLayer()** to project the image stored in the specified variable.
- However, note that at this stage, all provinces will be displayed on the map due to not having specified or extracted a specific province yet.
 - The result is shown in the next slide.

Projecting Administrative Boundaries Data (Province Level) onto Map 2/4

The screenshot displays the Google Earth Engine web interface. At the top, the 'Google Earth Engine' logo is on the left, and a search bar with the text 'Search places and datasets...' is on the right. Below the logo, there are tabs for 'Scripts', 'Docs', and 'Assets'. The 'Assets' tab is active, showing a list of assets under 'users/devjpconsul...'. The 'New Script *' editor is open, showing a script with the following code:

```
Imports (1 entry)
var table: Table users/devjpconsultant/vietnam_lv11
1 Map.centerObject(table, 4);
2
3 Map.addLayer(table, {}, "ベトナム_地域レベル");
4
5
```

Red annotations highlight specific parts of the script:

- A red arrow points to line 1, `Map.centerObject(table, 4);`, with the text: "Code that moves the map to the specified location".
- A red arrow points to line 3, `Map.addLayer(table, {}, "ベトナム_地域レベル");`, with the text: "Data is projected on the map as **Vietnam_Province Level**".

The map below the script shows Southeast Asia. A red arrow points to the dark, semi-transparent layer representing administrative boundaries, with the text: "Administrative boundaries data for all provinces are displayed.".

On the right side of the map, there is a 'Layers' panel. It shows a checkbox for 'ベトナム_地域レベル' which is checked. A red arrow points to the transparency slider bar next to it, with the text: "You can change the transparency of the map by moving the bar.".

At the bottom of the map, there is a scale bar showing '500 km' and a copyright notice '地図データ ©2021 Google, INEGI'.

- The result on the previous slide is not sufficient for province level analysis as the target province is not specified/extracted yet.
- Therefore, it is necessary to specify and extract the target province by applying a filter to the variable **table** in which the province level administrative boundaries data is stored, and store it in another variable. This time, you will specify/extract Hanoi (Ha Noi).
- One of the codes for specifying/extracting a province is **variable.name.filter(ee.Filter.eq("attributes", "target province"));**
 - The next slide shows the actual code.

Projecting Administrative Boundaries Data (Province Level) onto Map

4/4

The screenshot displays the Google Earth Engine web interface. The top navigation bar includes the Google Earth Engine logo, a search bar, and icons for help, notifications, and user profile. Below the navigation bar, there are tabs for 'Scripts', 'Docs', and 'Assets'. The 'Assets' tab is active, showing a list of assets under 'CLOUD ASSETS' and 'LEGACY ASSETS'. The 'New Script' editor is open, showing a script with the following code:

```
Imports (1 entry)
var table: Table users/devjpconsultant/vietnam_lv11

1 Map.centerObject(table, 4);
2
3
4
5 var HaNoi = table.filter(ee.Filter.eq("ADM1_EN", "Ha Noi"));
6
7
8
9 Map.addLayer(HaNoi, {}, "Ha Noi");
10
```

Annotations on the script editor:

- A red arrow points to the variable **HaNoi** in line 5, with the text: "New variable **HaNoi** stores the specified and extracted province."
- A red arrow points to the string **"Ha Noi"** in line 5, with the text: "Specify/Extract **Ha Noi** of **ADM1_EN** for **table**".

The map view shows a map of Vietnam with the province of Ha Noi highlighted in black. The 'Layers' panel on the right shows two layers: 'Ha Noi' (checked) and 'ベトナム_地域レベル' (unchecked). A red arrow points to the 'Ha Noi' layer, with the text: "Removing the ✓ mark eliminates the item from the map."

How to Look for Attribute and Province Name Required for Filtering

- Use the **Inspector** function in the right pane and click the province you want to specify or extract. You will then see the data for the province you clicked. Look for the appropriate column and province names.

The screenshot displays the Google Earth Engine web interface. On the left, the 'Assets' pane shows a project named 'users/devjpconsultant' with two assets: 'vietnam_lv10' and 'vietnam_lv11'. The central 'New Script' editor contains a JavaScript script that filters a table for 'Ha Noi' based on the 'ADM1_EN' attribute. The right-hand 'Inspector' panel, highlighted with a red box and a circled '1', shows the details of a selected feature. Within the 'properties' section, the 'ADM1_EN' attribute is highlighted with a red box and a circled '3', showing its value as 'Ha Noi'. A red arrow points from the text 'Use this attribute and province name found here.' to this specific attribute. Below the script editor, a map of Vietnam is shown with a red arrow and a circled '2' pointing to the 'Ha Noi' region, with the text 'Click' next to it. The bottom of the interface includes a map scale bar and copyright information.

Google Earth Engine

Search places and datasets...

Scripts Docs Assets

NEW ADD A PROJECT

CLOUD ASSETS

LEGACY ASSETS

users/devjpconsultant

vietnam_lv10

vietnam_lv11

New Script *

Get Link Save Run Reset Apps

Inspector Console Tasks

Imports (1 entry)

var table: Table users/devjpconsultant/vietnam_lv11

```
1 Map.centerObject(table, 4);
2
3 Map.addLayer(table, {}, "ベトナム_地域レベル");
4
5 var HaNoi = table.filter(ee.Filter.eq("ADM1_EN", "Ha Noi"));
6
7 Map.addLayer(HaNoi, {}, "Ha Noi");
8
9
```

Point (105.734, 20.902) at 1Km/px

Objects

ベトナム_地域レベル: FeatureCollection (1 element)

0: Feature 00000000000000000017 (Polygon, 15 p...

type: Feature

id: 00000000000000000017

geometry: Polygon, 4668 vertices

properties: Object (15 properties)

ADM0_EN: Viet Nam

ADM0_PCODE: VN

ADM0_VI: Việt Nam

ADM1ALT1EN:

ADM1ALT1VI:

ADM1ALT2EN:

ADM1ALT2VI:

ADM1_EN: Ha Noi

ADM1_PCODE: VN101

ADM1_REF:

ADM1_VI: TP. Hà Nội

Use this attribute and province name found here.

Click

Layers

地図 航空写真

Google

キーボードショートカット 地図データ ©2021 20 km 利用規約

How to Handle Nighttime Light Data

Nighttime Light Data Stored in GEE

- The third in the list is to be used in future ex-post evaluations.

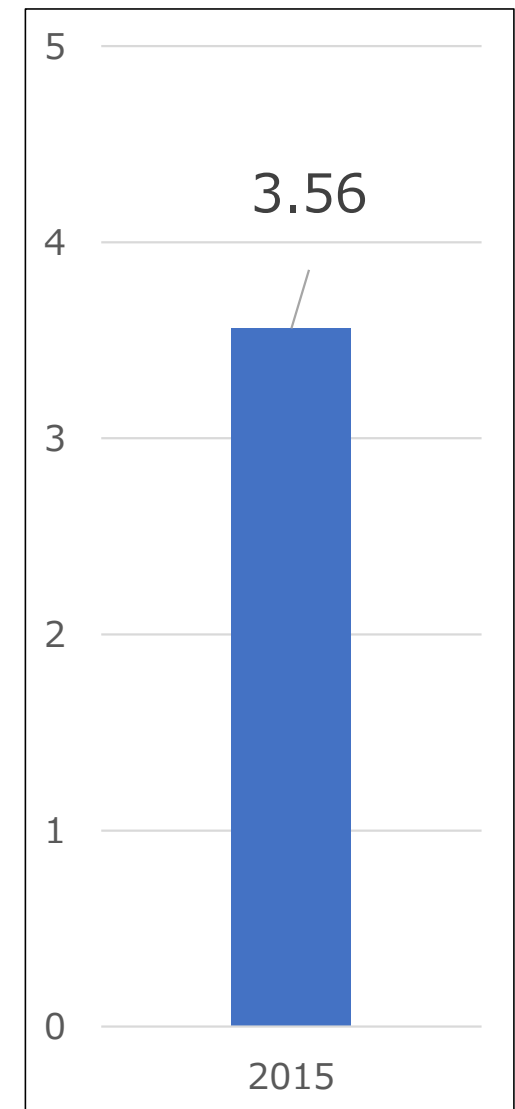
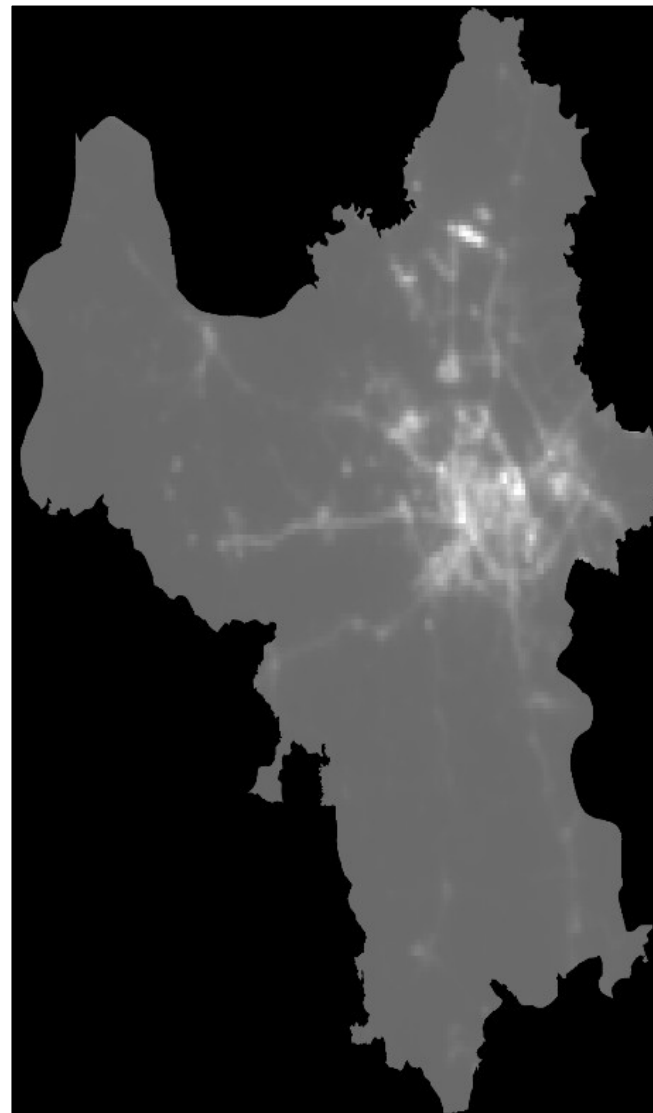
This document deals with the **third** dataset.

Please study the other datasets on your own.

No	Name	Period	Nighttime light intensity range (pt)
1	DMSP OLS: Nighttime Lights Time Series Version 4	1992 - 2014	0-63
2	DMSP OLS: Global Radiance-Calibrated Nighttime Lights Version 4	1996 - 2011	0-6060.6
3	VIIRS Nighttime Day/Night Band Composites Version 1	April 2012 - present	-1.5-340573
4	VIIRS Stray Light Corrected Nighttime Day/Night Band Composites Version 1	2014 - present	-1.5-193565

Final Goal of This Section

- (1) Display of nighttime light data and (2) exported nighttime light data for Hanoi in 2015, as uploaded in the previous section.



Importing Nighttime Light Data 1/5

- Type “Nighttime” in the search field.
- Click **VIIRS Nighttime Day/Night Band Composites Version 1** from among the displayed satellite data.

The screenshot shows a search interface with a search bar at the top containing the text "Nighttime". A red arrow points to the search bar with the text "Type 'Nighttime.'". Below the search bar, there are sections for "PLACES", "RASTERS", and "TABLES". Under the "RASTERS" section, several satellite data entries are listed. The entry "VIIRS Nighttime Day/Night Band Composites Version 1" is highlighted with a red rectangular box. To the right of this entry is a blue link that says "import »". A red arrow points from the text "Relevant satellite data is displayed automatically." to the highlighted entry. Other entries in the "RASTERS" section include "DMSP OLS: Global Radiance-Calibrated Nighttime Lights Version 4, Defense Meteorologica...", "DMSP OLS: Nighttime Lights Time Series Version 4, Defense Meteorological Program Oper...", "VIIRS Stray Light Corrected Nighttime Day/Night Band Composites Version 1", and several "YCEO Surface Urban Heat Islands" entries. A "more »" link is at the bottom of the "RASTERS" section.

Nighttime ← Type “Nighttime.”


Relevant satellite data is displayed automatically.

VIIRS Nighttime Day/Night Band Composites Version 1 [import »](#)

Importing Nighttime Light Data 2/5


- The information of the clicked nighttime light data is displayed.
- Data summary, year of use, band, code required for data import, etc. are provided. What you should check for is the band and the code required for data import.

VIIRS Nighttime Day/Night Band Composites Version 1



Dataset Availability
2012-04-01T00:00:00 - 2021-05-01T00:00:00

Dataset Provider
[Earth Observation Group, Payne Institute for Public Policy, Colorado School of Mines](#)

Collection Snippet 

```
ee.ImageCollection("NOAA/VIIRS/DNB/MONTHLY_V1/VCMCFG")
```

[See example](#)

Band information

DESCRIPTION	BANDS	TERMS OF USE
<p>Monthly average radiance composite images using nighttime data from the Visible Infrared Imaging Radiometer Suite (VIIRS) Day/Night Band (DNB).</p> <p>As these data are composited monthly, there are many areas of the globe where it is impossible to get good quality data coverage for that month. This can be due to cloud cover, especially in the tropical regions, or due to solar illumination, as happens toward the poles in their respective summer months. Therefore it is recommended that users of these data utilize the 'cf_cvg' band and not assume a value of zero in the average radiance image means that no lights were observed.</p> <p>Cloud cover is determined using the VIIRS Cloud Mask product (VCM). In addition, data near the edges of the swath are not included in the composites (aggregation zones 29-32). Version 1 has NOT been filtered to screen out lights from aurora, fires, boats, and other temporal lights. This separation is under development and will be included in a later version of this time series. Also in development is a method to separate lights from background (non-lights) ...</p> <p>Prior ... impacted by stray light, lightning, lunar illumination, and cloud-cover.</p>		


Code required for data import

[CLOSE](#) [IMPORT](#)

Importing Nighttime Light Data 3/5


- The band used in this document is **avg_rad**.

VIIRS Nighttime Day/Night Band Composites Version 1



Dataset Availability
2012-04-01T00:00:00 - 2021-05-01T00:00:00

Dataset Provider
[Earth Observation Group, Payne Institute for Public Policy, Colorado School of Mines](#)

Collection Snippet 

```
ee.ImageCollection("NOAA/VIIRS/DNB/MONTHLY_V1/VCMSFG")
```

[See example](#)

DESCRIPTION **BANDS** TERMS OF USE

Resolution
463.83 meters

Bands Table

Name	Description	Min*	Max*	Units
avg_rad	Average DNB radiance values.	-1.5	340572.84	nanoWatts/cm2/sr
cf_cvg	Cloud-free coverages; the total number of observations that went into each pixel. This band can be used to identify areas with low numbers of observations where the quality is reduced.	0	58	


* = Values are estimated

[CLOSE](#) [IMPORT](#)

Importing Nighttime Light Data 4/5


- After checking the band, copy the code required for data import.
- Click **CLOSE** after copying.

VIIRS Nighttime Day/Night Band Composites Version 1



Dataset Availability
2012-04-01T00:00:00 - 2021-05-01T00:00:00

Dataset Provider
[Earth Observation Group, Payne Institute for Public Policy, Colorado School of Mines](#)

Collection Snippet 

```
ee.ImageCollection("NOAA/VIIRS/DNB/MONTHLY_V1/VCMCFG")
```

[See example](#)

DESCRIPTION BANDS TERMS OF USE

Monthly average radiance composite images using nighttime data from the Visible Infrared Imaging Radiometer Suite (VIIRS) Day/Night Band (DNB).

As these data are composited monthly, there are many areas of the globe where it is impossible to get good quality data coverage for that month. This can be due to cloud cover, especially in the tropical regions, or due to solar illumination, as happens toward the poles in their respective summer months. Therefore it is recommended that users of these data utilize the 'cf_cvg' band and not assume a value of zero in the average radiance image means that no lights were observed.

Cloud cover is determined using the VIIRS Cloud Mask product (VCM). In addition, data near the edges of the swath are not included in the composites (aggregation zones 29-32). Version 1 has NOT been filtered to screen out lights from aurora, fires, boats, and other temporal lights. This separation is under development and will be included in a later version of this time series. Also in development is a method to separate lights from background (non-light) values.

DNB data is filtered to exclude data impacted by stray light, lightning, lunar illumination, and cloud-cover.

CLOSE IMPORT

← Copy

Importing Nighttime Light Data 5/5

- You will be returned to the main screen. Paste the code into the code editor.
- Define the variable and store the code. In this document, variable name is **data**.

The screenshot shows the Google Earth Engine web interface. At the top, the search bar contains "Nighttime". Below the search bar, the "New Script *" button is visible. The script editor shows the following code:

```

Imports (1 entry)
var table: Table users/devjpconsultant/vietnam_lv11
1 var HaNoi = table.filter(ee.Filter.eq("ADM1_EN", "Ha Noi"));
2
3 Map.centerObject(HaNoi, 9);
4 Map.addLayer(HaNoi, {}, "Ha Noi");
5
6 var data = ee.ImageCollection("NOAA/VIIRS/DNB/MONTHLY_V1/VCMSFCG...");
7
8

```

A red box highlights line 6: `var data = ee.ImageCollection("NOAA/VIIRS/DNB/MONTHLY_V1/VCMSFCG...");`. Below the script editor, a red text overlay reads: **var variable name = code for nighttime light data**. The map view shows Hanoi, Vietnam, with various labels in Japanese and English. The map includes a scale bar (20 km) and a copyright notice (©2021).

Viewing the Information on Nighttime Light Data 1/2

- You can use **print(variable name)** to view nighttime light data.
- Information is displayed in **Console**.

The screenshot displays the Google Earth Engine web interface. At the top, the 'Nighttime' search bar is active. Below the search bar, a toolbar contains buttons for 'New Script', 'Get Link', 'Save', 'Run', 'Reset', and 'Apps'. The script editor on the left contains the following code:

```
1 var HaNoi = table.filter(ee.Filter.eq("ADM1_EN", "Ha Noi"));
2
3 Map.centerObject(HaNoi, 9);
4 Map.addLayer(HaNoi, {}, "Ha Noi");
5
6 var data = ee.ImageCollection("NOAA/VIIRS/DNB/MONTHLY_V1/VCMLCFG")
7
8 print(data);
9
10
```

The line `print(data);` is highlighted with a red rectangle. To the right of the script editor, the 'Inspector' and 'Console' tabs are visible. The 'Console' tab is active, showing the message: 'Use print(...) to write to this console.' Below this, an entry is displayed: 'ImageCollection NOAA/VIIRS/DNB/MONTHLY...' with a 'JSON' link. A red arrow points from the text 'Information is displayed in Console.' to the console output.

The bottom portion of the screenshot shows a map of Vietnam, centered on Hanoi (ハノイ). The map displays various geographical features, including rivers, roads, and administrative boundaries. Labels for nearby locations such as 'Tp. Bắc Ninh', 'Tp. Chí Linh', 'Tp. Hải Dương', and 'Tp. Hải Phòng' are visible. The map interface includes standard navigation controls like zoom in/out buttons and a scale bar at the bottom.

Viewing the Information on Nighttime Light Data 2/2

- Click an item to open the information. If you open **features**, you can see that the data from and after April 2012 is stored.
- Only the data from January to December 2015 is used in this document.

The screenshot displays the Google Earth Engine web interface. The top navigation bar includes the Google Earth Engine logo, a search bar with the text 'Nighttime', and icons for help, notifications, and user profile. Below the navigation bar is a toolbar with buttons for 'Get Link', 'Save', 'Run', 'Reset', and 'Apps'. The main workspace is divided into two panels. The left panel, titled 'Imports (1 entry)', contains a code editor with the following JavaScript code:

```
1 var table: Table users/devjpconsultant/vi
2
3 Map.centerObject(HaNoi, 9);
4 Map.addLayer(HaNoi, {}, "Ha Noi");
5
6 var data = ee.ImageCollection("NOAA/VIIRS/DNB/
7
8 print(data);
9
10
```

The right panel, titled 'Inspector', shows the details of the selected data. It displays the following information:

- ImageCollection NOAA/VIIRS/DNB/MONTHLY_V1/VCMCFG (109 elements) JSON
- type: ImageCollection
- id: NOAA/VIIRS/DNB/MONTHLY_V1/VCMCFG
- version: 1633091423442399
- bands: []
- features: List (109 elements)

A red arrow points to the 'features' list, with the text 'Year of data' written in red above it. The list shows the following features:

- 0: Image NOAA/VIIRS/DNB/MONTHLY_V1/VCMCFG/20120401 (2 bands)
- 1: Image NOAA/VIIRS/DNB/MONTHLY_V1/VCMCFG/20120501 (2 bands)
- 2: Image NOAA/VIIRS/DNB/MONTHLY_V1/VCMCFG/20120601 (2 bands)
- 3: Image NOAA/VIIRS/DNB/MONTHLY_V1/VCMCFG/20120701 (2 bands)
- 4: Image NOAA/VIIRS/DNB/MONTHLY_V1/VCMCFG/20120801 (2 bands)
- 5: Image NOAA/VIIRS/DNB/MONTHLY_V1/VCMCFG/20120901 (2 bands)
- 6: Image NOAA/VIIRS/DNB/MONTHLY_V1/VCMCFG/20121001 (2 bands)
- 7: Image NOAA/VIIRS/DNB/MONTHLY_V1/VCMCFG/20121101 (2 bands)
- 8: Image NOAA/VIIRS/DNB/MONTHLY_V1/VCMCFG/20121201 (2 bands)
- 9: Image NOAA/VIIRS/DNB/MONTHLY_V1/VCMCFG/20130101 (2 bands)
- 10: Image NOAA/VIIRS/DNB/MONTHLY_V1/VCMCFG/20130201 (2 bands)
- 11: Image NOAA/VIIRS/DNB/MONTHLY_V1/VCMCFG/20130301 (2 bands)
- 12: Image NOAA/VIIRS/DNB/MONTHLY_V1/VCMCFG/20130401 (2 bands)
- 13: Image NOAA/VIIRS/DNB/MONTHLY_V1/VCMCFG/20130501 (2 bands)
- 14: Image NOAA/VIIRS/DNB/MONTHLY_V1/VCMCFG/20130601 (2 bands)
- 15: Image NOAA/VIIRS/DNB/MONTHLY_V1/VCMCFG/20130701 (2 bands)
- 16: Image NOAA/VIIRS/DNB/MONTHLY_V1/VCMCFG/20130801 (2 bands)

Extracting Target Year Data

- Apply the date filter `.filterDate("first date", "last date")` to the nighttime light data.
- If you check `print()`, you can see that data only for 2015 is extracted.

The screenshot displays the Google Earth Engine web interface. On the left, the 'New Script' editor shows a JavaScript script. The script filters a table for 'Ha Noi' and then filters an image collection for the year 2015. A red arrow points to the semicolon at the end of the `.filterDate` line. On the right, the 'Console' tab shows the output of the `print` statement, which is an `ImageCollection` for 2015. A red arrow points to the year '2015' in the console output.

```
Google Earth Engine Nighttime
```

New Script * Get Link Save Run Reset Apps

```
Imports (1 entry)
  var table: Table users/devjpconsultant/vietnam_lv11
1 var HaNoi = table.filter(ee.Filter.eq("ADM1_EN", "Ha Noi"));
2
3 Map.centerObject(HaNoi, 9);
4 Map.addLayer(HaNoi, {}, "Ha Noi");
5
6 var data = ee.ImageCollection("NOAA/VIIRS/DNB/MONTHLY_V1/VCMSFG")
7   .filterDate("2015-01-01", "2015-12-31");
8
9 print(data);
10
```

Inspector Console Tasks

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

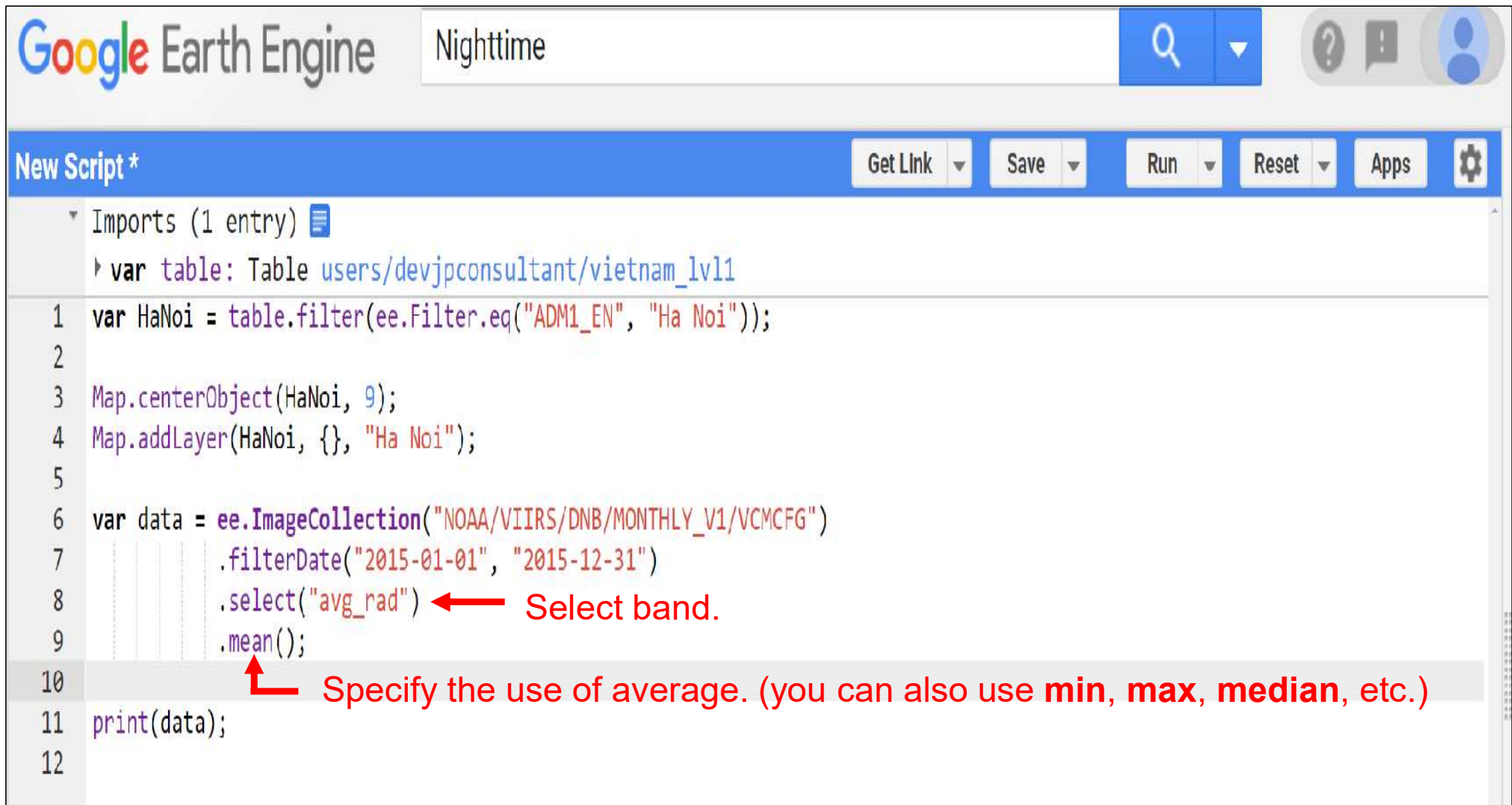
```
ImageCollection NOAA/VIIRS/DNB/MONTHLY_V1/VCMSFG (12 elements) JSON
  type: ImageCollection
  id: NOAA/VIIRS/DNB/MONTHLY_V1/VCMSFG
  version: 1633091423442399
  bands: []
  features: List (12 elements)
    0: Image NOAA/VIIRS/DNB/MONTHLY_V1/VCMSFG/20150101 (2 ...
    1: Image NOAA/VIIRS/DNB/MONTHLY_V1/VCMSFG/20150201 (2 ...
    2: Image NOAA/VIIRS/DNB/MONTHLY_V1/VCMSFG/20150301 (2 ...
    3: Image NOAA/VIIRS/DNB/MONTHLY_V1/VCMSFG/20150401 (2 ...
    4: Image NOAA/VIIRS/DNB/MONTHLY_V1/VCMSFG/20150501 (2 ...
    5: Image NOAA/VIIRS/DNB/MONTHLY_V1/VCMSFG/20150601 (2 ...
    6: Image NOAA/VIIRS/DNB/MONTHLY_V1/VCMSFG/20150701 (2 ...
    7: Image NOAA/VIIRS/DNB/MONTHLY_V1/VCMSFG/20150801 (2 ...
    8: Image NOAA/VIIRS/DNB/MONTHLY_V1/VCMSFG/20150901 (2 ...
    9: Image NOAA/VIIRS/DNB/MONTHLY_V1/VCMSFG/20151001 (2 ...
    10: Image NOAA/VIIRS/DNB/MONTHLY_V1/VCMSFG/20151101 (2 ...
    11: Image NOAA/VIIRS/DNB/MONTHLY_V1/VCMSFG/20151201 (2 ...
  properties: Object (23 properties)
```

Don't forget to put the semicolon.

2015 only

Selecting the Target Band and Calculating the Average

- Specify the band to use for the selection filter.
- Specify the use of the 2015 annual average.



The screenshot shows the Google Earth Engine interface. At the top, the "Google Earth Engine" logo is on the left, and a search bar contains the text "Nighttime". To the right of the search bar are icons for help, notifications, and a user profile. Below the search bar is a blue header bar with the text "New Script *". To the right of this header are buttons for "Get Link", "Save", "Run", "Reset", and "Apps", along with a settings gear icon. The main area is a code editor with a light blue background. It shows a script with the following lines:

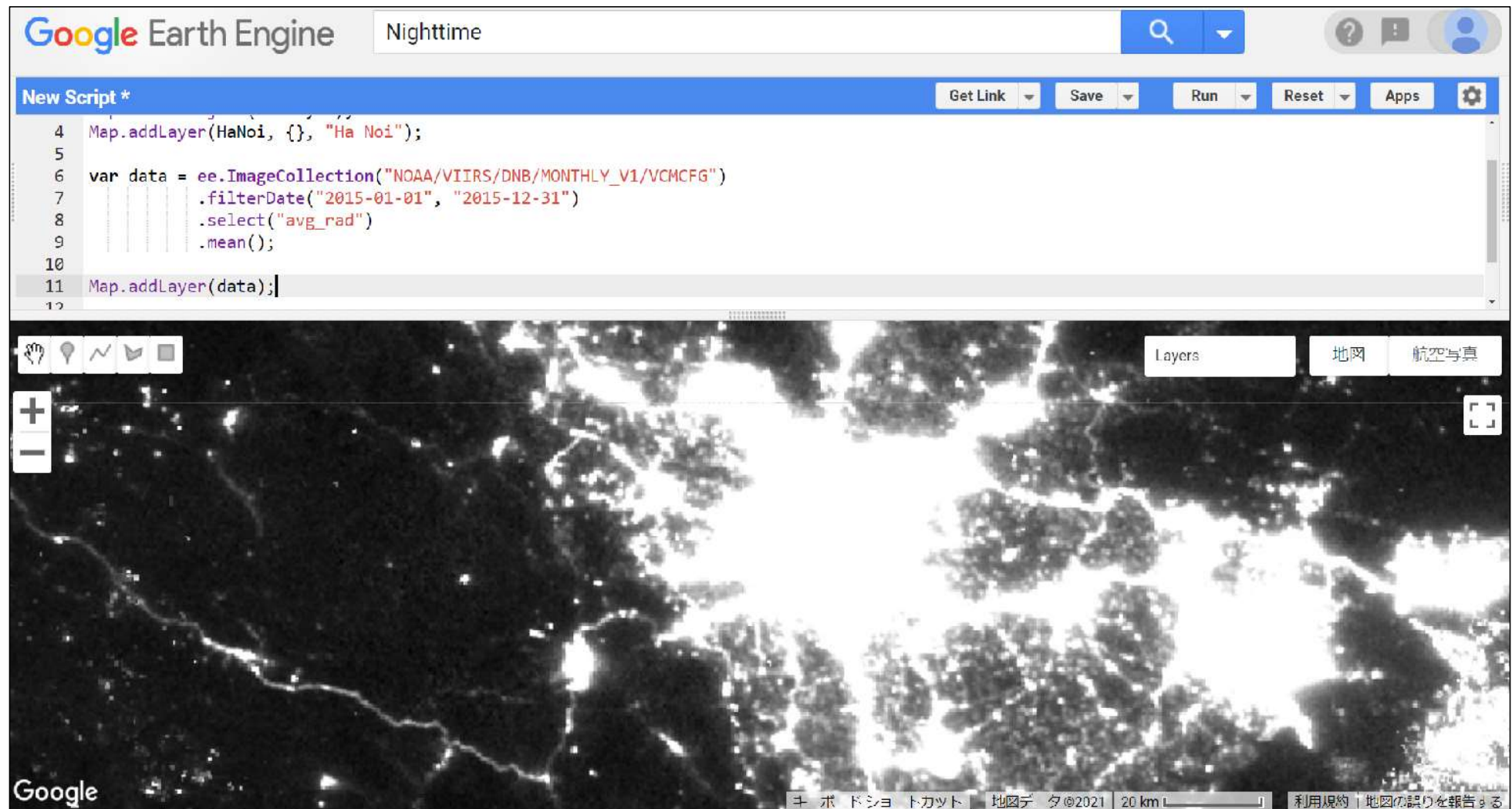
```
Imports (1 entry)
  var table: Table users/devjpconsultant/vietnam_lv11
1 var HaNoi = table.filter(ee.Filter.eq("ADM1_EN", "Ha Noi"));
2
3 Map.centerObject(HaNoi, 9);
4 Map.addLayer(HaNoi, {}, "Ha Noi");
5
6 var data = ee.ImageCollection("NOAA/VIIRS/DNB/MONTHLY_V1/VCMSCFG")
7   .filterDate("2015-01-01", "2015-12-31")
8   .select("avg_rad")
9   .mean();
10
11 print(data);
12
```

Red annotations are present in the code editor:

- A red arrow points from the text "Select band." to the `"avg_rad"` argument in the `.select()` method on line 8.
- A red arrow points from the text "Specify the use of average. (you can also use min, max, median, etc.)" to the `.mean()` method on line 9.

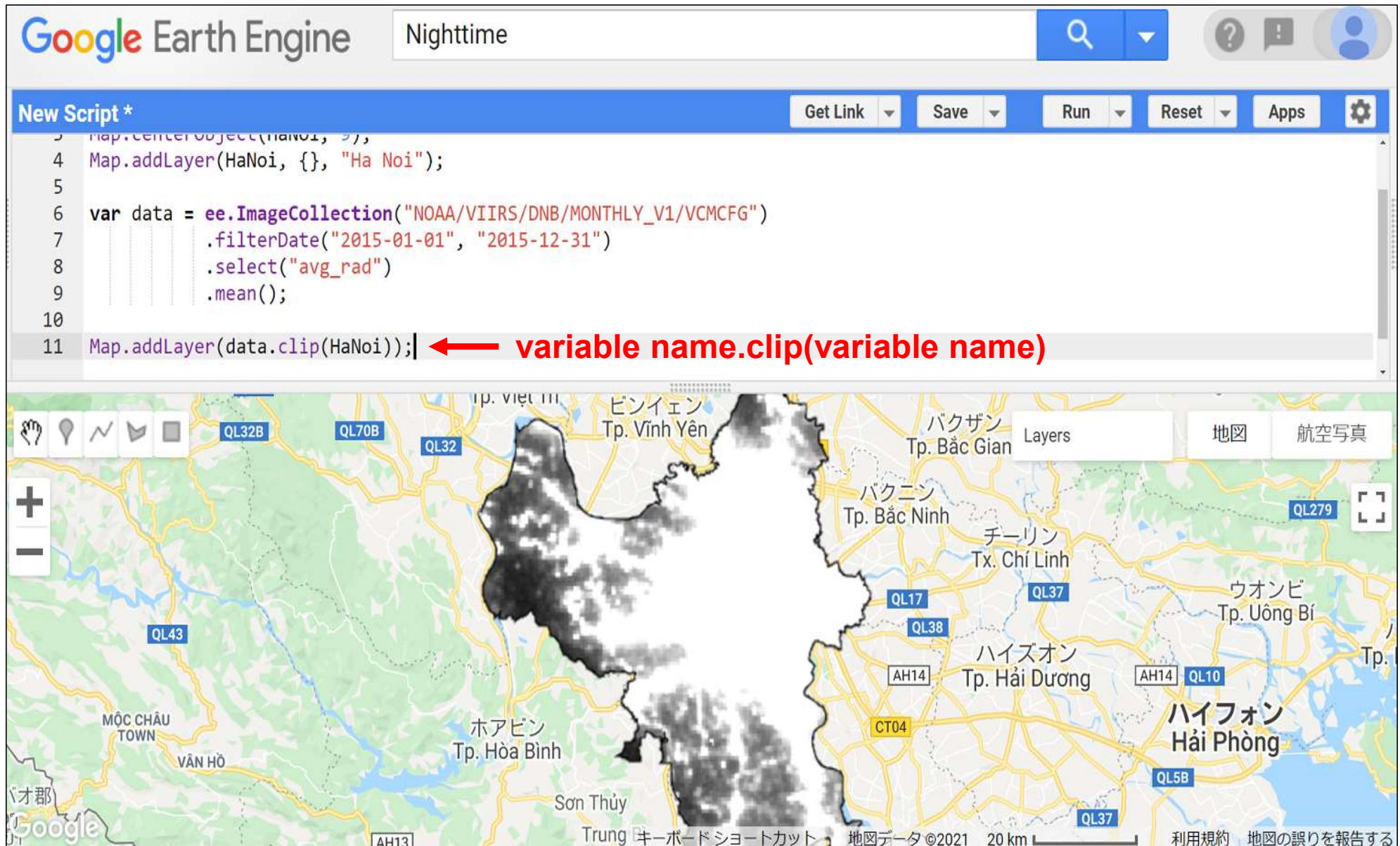
Projecting Nighttime Light Data onto Map

- You can use **Map.addLayer** to project nighttime light data onto the map. However, since the region to be projected is not specified, the nighttime light data of the whole world is projected.



Specifying Projection Regions

- Use **.clip(variable name)** to specify the region to project.



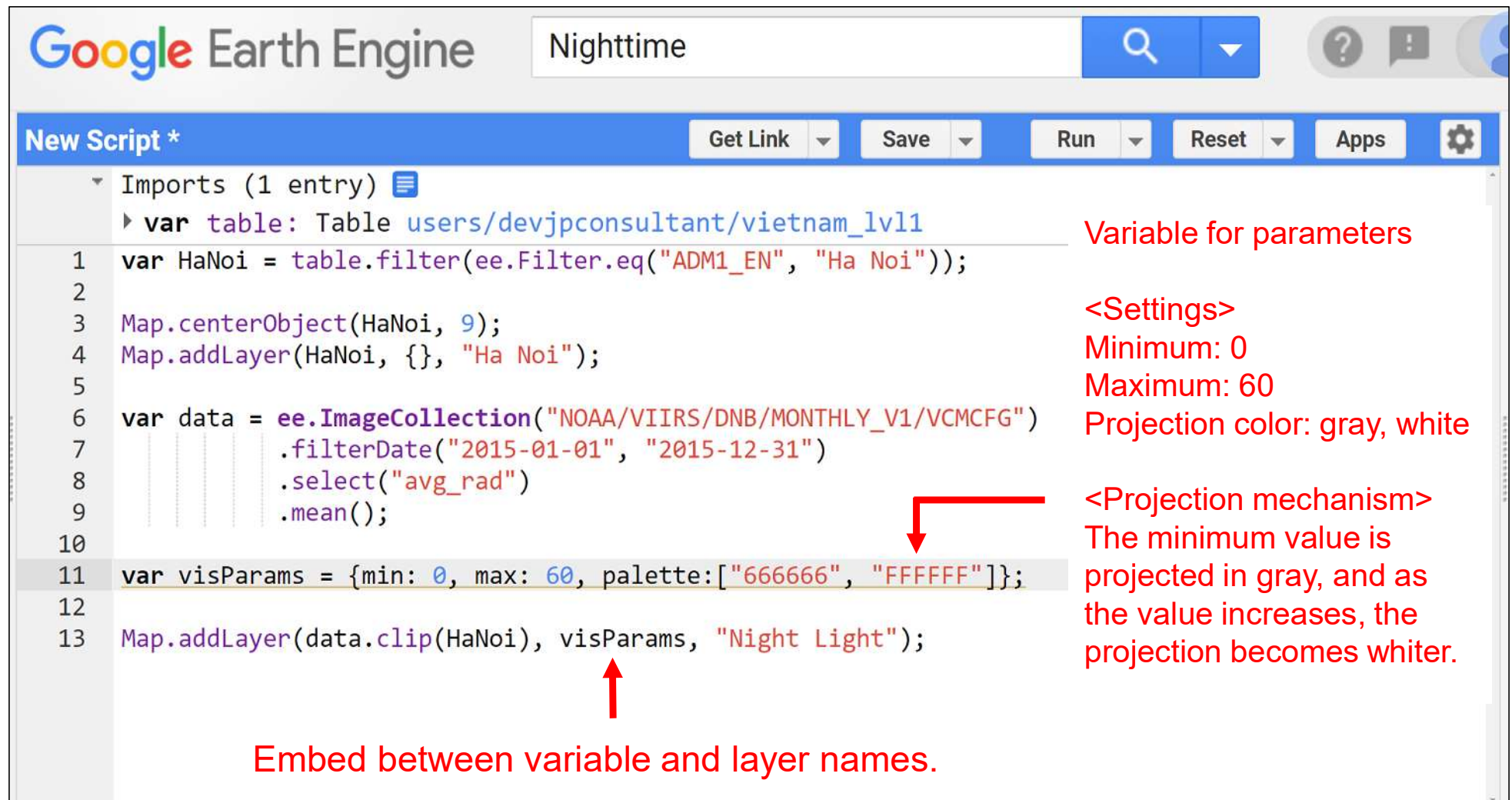
The screenshot displays the Google Earth Engine web interface. At the top, the 'Nighttime' filter is selected. Below the search bar, a toolbar contains buttons for 'New Script *', 'Get Link', 'Save', 'Run', 'Reset', and 'Apps'. The script editor shows the following code:

```
1 Map.centerObject(hanoi, 9);
2
3 Map.addLayer(HaNoi, {}, "Ha Noi");
4
5
6 var data = ee.ImageCollection("NOAA/VIIRS/DNB/MONTHLY_V1/CMCFG")
7   .filterDate("2015-01-01", "2015-12-31")
8   .select("avg_rad")
9   .mean();
10
11 Map.addLayer(data.clip(HaNoi));
```

A red arrow points to the `data.clip(HaNoi)` line, with the text **variable name.clip(variable name)** next to it. The map below shows a satellite image of Vietnam at night, with a dark, irregularly shaped region representing the projected data. The map includes labels for various cities and towns in Vietnamese, such as 'Tp. Vinh Yen', 'Tp. Bac Ninh', 'Tp. Hai Duong', and 'Hải Phòng'. A scale bar at the bottom indicates 20 km.

Setting Projection Parameters

- The reason why the projection view is very bright is because the parameters for the projection are not set yet.
- You must create a variable for the parameters and embed it in **Map.addLayer**.



Google Earth Engine

Nighttime

New Script *

Get Link Save Run Reset Apps

```
Imports (1 entry)
  var table: Table users/devjpconsultant/vietnam_lv11
1 var HaNoi = table.filter(ee.Filter.eq("ADM1_EN", "Ha Noi"));
2
3 Map.centerObject(HaNoi, 9);
4 Map.addLayer(HaNoi, {}, "Ha Noi");
5
6 var data = ee.ImageCollection("NOAA/VIIRS/DNB/MONTHLY_V1/VCMCFG")
7   .filterDate("2015-01-01", "2015-12-31")
8   .select("avg_rad")
9   .mean();
10
11 var visParams = {min: 0, max: 60, palette:["666666", "FFFFFF"]};
12
13 Map.addLayer(data.clip(HaNoi), visParams, "Night Light");
```

Variable for parameters

<Settings>
Minimum: 0
Maximum: 60
Projection color: gray, white

<Projection mechanism>
The minimum value is projected in gray, and as the value increases, the projection becomes whiter.

Embed between variable and layer names.

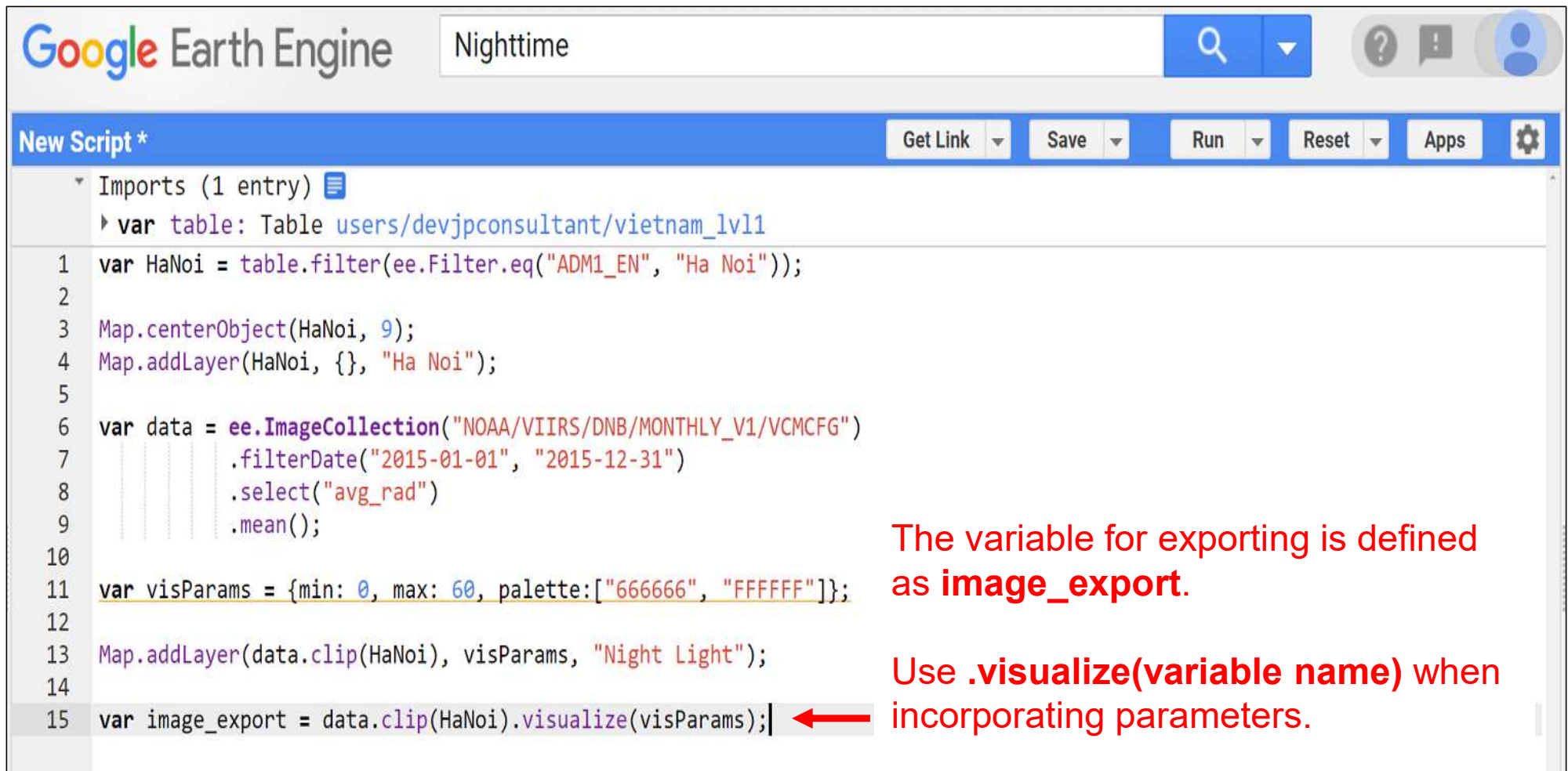
Projection of Nighttime Light Data for 2015 in Hanoi

- The appropriate nighttime light data was projected because the province and parameters were set.
- For the purpose of this document, the maximum value of the parameter is set to 60, but it should be changed accordingly.



Exporting Nighttime Light Data (Image) 1/4

- Export a projection to use for report.
- As a preliminary step, store satellite data incorporating the projection region and parameters in a new variable (for export).



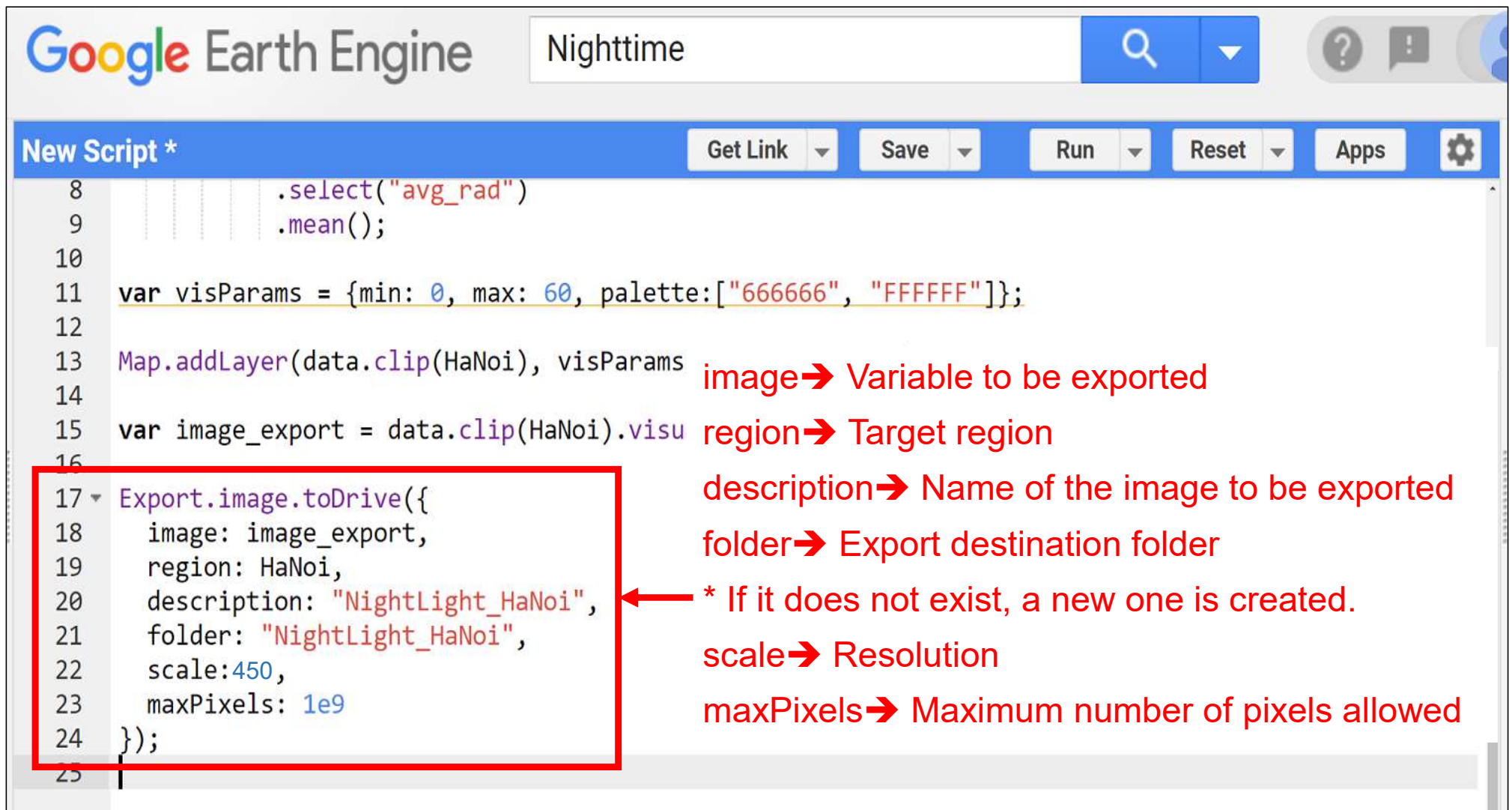
```
Google Earth Engine Nighttime
New Script *
Get Link Save Run Reset Apps
Imports (1 entry)
  var table: Table users/devjpconsultant/vietnam_lv11
1 var HaNoi = table.filter(ee.Filter.eq("ADM1_EN", "Ha Noi"));
2
3 Map.centerObject(HaNoi, 9);
4 Map.addLayer(HaNoi, {}, "Ha Noi");
5
6 var data = ee.ImageCollection("NOAA/VIIRS/DNB/MONTHLY_V1/CMCFG")
7   .filterDate("2015-01-01", "2015-12-31")
8   .select("avg_rad")
9   .mean();
10
11 var visParams = {min: 0, max: 60, palette:["666666", "FFFFFF"]};
12
13 Map.addLayer(data.clip(HaNoi), visParams, "Night Light");
14
15 var image_export = data.clip(HaNoi).visualize(visParams);
```

The variable for exporting is defined as **image_export**.

Use **.visualize(variable name)** when incorporating parameters.

Exporting Nighttime Light Data (Image) 2/4

- The code for exporting is shown below.
- The data is exported to Google Drive.



Google Earth Engine

Nighttime

New Script *

Get Link Save Run Reset Apps

```
8 .select("avg_rad")
9 .mean();
10
11 var visParams = {min: 0, max: 60, palette:["666666", "FFFFFF"]};
12
13 Map.addLayer(data.clip(HaNoi), visParams
14
15 var image_export = data.clip(HaNoi).visu
16
17 Export.image.toDrive({
18   image: image_export,
19   region: HaNoi,
20   description: "NightLight_HaNoi",
21   folder: "NightLight_HaNoi",
22   scale: 450,
23   maxPixels: 1e9
24 });
25
```

image → Variable to be exported

region → Target region

description → Name of the image to be exported

folder → Export destination folder

* If it does not exist, a new one is created.

scale → Resolution

maxPixels → Maximum number of pixels allowed

Exporting Nighttime Light Data (Image) 3/4

- When **Tasks** lights up, click it.
- Image is ready to be exported. Run the export.

The screenshot shows the Google Earth Engine web interface. On the left, a code editor displays a script for exporting nighttime light data. The script filters data for Ha Noi, processes it, and sets up an export task named 'NightLight_HaNoi'. On the right, the 'Tasks' panel is active, showing a list of 'UNSUBMITTED TASKS' with the task 'NightLight_HaNoi' ready to be run. A red arrow points to the 'RUN' button next to the task name.

```
1 var HaNoi = table.filter(ee.Filter.eq("ADM1_EN", "Ha Noi"));
2
3 Map.centerObject(HaNoi, 9);
4 Map.addLayer(HaNoi, {}, "Ha Noi");
5
6 var data = ee.ImageCollection("NOAA/VIIRS/DNB/MONTHLY_V1/VCMSFG")
7   .filterDate("2015-01-01", "2015-12-31")
8   .select("avg_rad")
9   .mean();
10
11 var visParams = {min: 0, max: 60, palette:["666666", "FFFFFF"]};
12
13 Map.addLayer(data.clip(HaNoi), visParams, "Night Light");
14
15 var image_export = data.clip(HaNoi).visualize(visParams);
16
17 Export.image.toDrive({
18   image: image_export,
19   region: HaNoi,
20   description: "NightLight_HaNoi",
21   folder: "NightLight_HaNoi",
22   scale: 450,
23   maxPixels: 1e9
24 });
25
```

Inspector Console **Tasks**

Manage tasks.

Search or cancel multiple tasks in the [Task Manager](#)

UNSUBMITTED TASKS

NightLight_HaNoi **RUN**

Image is ready for export.
Click **RUN**.

Exporting Nighttime Light Data (Image) 4/4

- Click **RUN** when the figure to the left appears. Then export starts (shown right) and the file is saved to Google Drive when export is complete.

Task: Initiate image export

Task name (no spaces) *
NightLight_HaNoi

Coordinate Reference System (CRS)
EPSG:3857

Scale (m/px)
450

DRIVE CLOUD STORAGE EE ASSET

Drive folder
NightLight_HaNoi

Filename *
NightLight_HaNoi

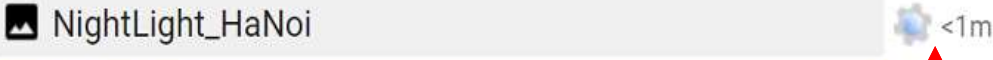
File format *
GEO_TIFF

CANCEL RUN

Inspector Console **Tasks**

Manage tasks.

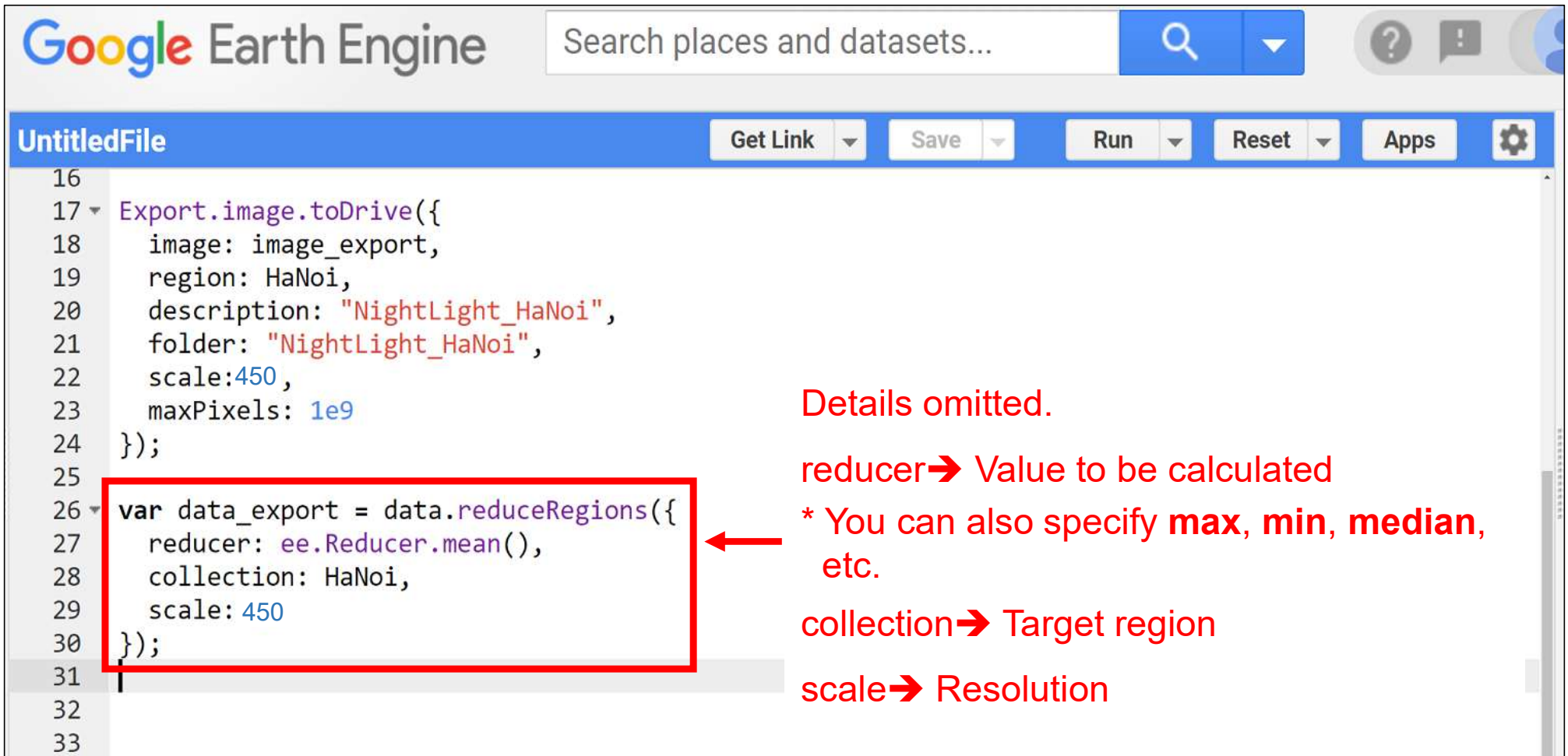
Search or cancel multiple tasks in the [Task Manager](#).

 NightLight_HaNoi <1m

Export ongoing mark is displayed.
Export takes about 1 to 5 minutes.

Exporting Nighttime Light Data (Values) 1/4

- Export the nighttime light intensity values for use in the report.
- As a preliminary step, store data obtained by averaging the nighttime light intensity values of entire the target region (Hanoi) in a new variable (for export).



The screenshot shows the Google Earth Engine code editor interface. The top bar includes the Google Earth Engine logo, a search bar with the text "Search places and datasets...", and buttons for "Get Link", "Save", "Run", "Reset", and "Apps". The code editor displays a script for exporting nighttime light data. The script is as follows:

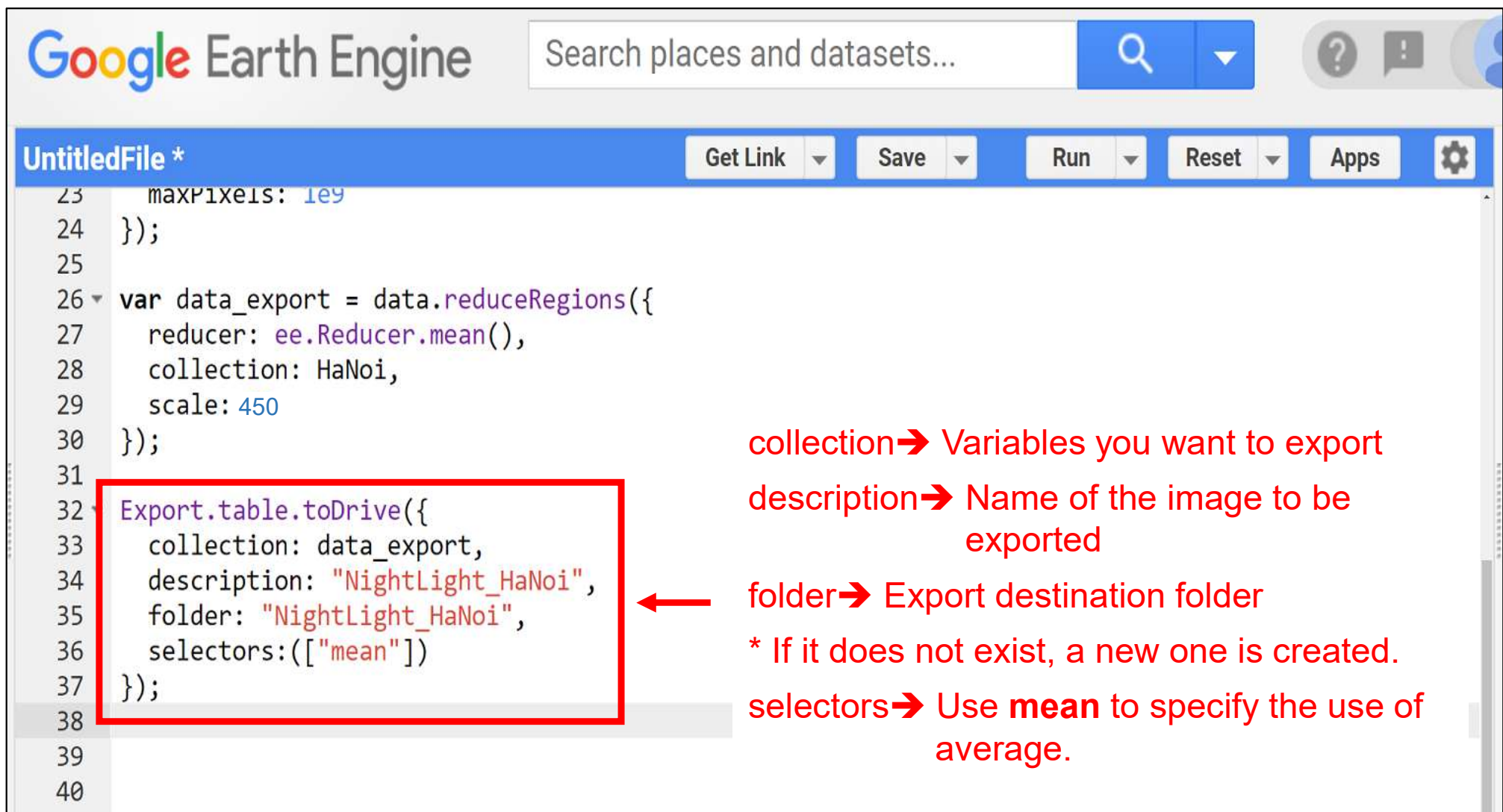
```
16
17 Export.image.toDrive({
18   image: image_export,
19   region: HaNoi,
20   description: "NightLight_HaNoi",
21   folder: "NightLight_HaNoi",
22   scale: 450,
23   maxPixels: 1e9
24 });
25
26 var data_export = data.reduceRegions({
27   reducer: ee.Reducer.mean(),
28   collection: HaNoi,
29   scale: 450
30 });
31
32
33
```

A red box highlights the `reduceRegions` function call on lines 26-30. To the right of the code, there are red annotations:

- Details omitted.
- reducer → Value to be calculated
- * You can also specify **max**, **min**, **median**, etc.
- collection → Target region
- scale → Resolution

Exporting Nighttime Light Data (Values) 2/4

- The code for exporting is shown below.
- The data is exported to Google Drive.



The screenshot shows the Google Earth Engine interface. The code editor displays the following JavaScript code:

```
23   maxPixels: 1e9
24   });
25
26   var data_export = data.reduceRegions({
27     reducer: ee.Reducer.mean(),
28     collection: HaNoi,
29     scale: 450
30   });
31
32   Export.table.toDrive({
33     collection: data_export,
34     description: "NightLight_HaNoi",
35     folder: "NightLight_HaNoi",
36     selectors:(["mean"])
37   });
38
39
40
```

Annotations explaining the `Export.table.toDrive` function parameters:

- `collection` → Variables you want to export
- `description` → Name of the image to be exported
- `folder` → Export destination folder
* If it does not exist, a new one is created.
- `selectors` → Use **mean** to specify the use of average.

Exporting Nighttime Light Data (Values) 3/4

- When **Tasks** lights up, click it.
- Image is ready to be exported. Run the export.

The screenshot shows the Google Earth Engine web interface. The top navigation bar includes the Google Earth Engine logo, a search bar, and icons for help, notifications, and user profile. Below the navigation bar is a toolbar with buttons for 'Get Link', 'Save', 'Run', 'Reset', and 'Apps'. The main workspace is divided into three panels: 'Inspector', 'Console', and 'Tasks'. The 'Tasks' panel is highlighted with a red box. It contains the text 'Manage tasks.' and 'Search or cancel multiple tasks in the [Task Manager](#).' Below this, there is a section titled 'UNSUBMITTED TASKS' which lists two tasks, both named 'NightLight_HaNoi'. Each task has a 'RUN' button next to it. A red arrow points from the text 'Values are ready for export. Click RUN.' to the 'RUN' button of the second task. The left panel shows a code editor with JavaScript code for exporting data.

```
23 maxPixels: 1e9
24 });
25
26 var data_export = data.reduceRegions({
27   reducer: ee.Reducer.mean(),
28   collection: HaNoi,
29   scale: 450
30 });
31
32 Export.table.toDrive({
33   collection: data_export,
34   description: "NightLight_HaNoi",
35   folder: "NightLight_HaNoi",
36   selectors:(["mean"])
37 });
38
39
40
```


Exporting Nighttime Light Data (Values) 4/4

- Click **RUN** when the figure to the left appears. Then export starts (shown right) and the file is saved to Google Drive when export is complete.

Task: Initiate table export

Task name (no spaces) *
NightLight_HaNoi

DRIVE CLOUD STORAGE EE ASSET

Drive folder
NightLight_HaNoi

Filename *
NightLight_HaNoi

File format *
CSV


CANCEL **RUN**



Inspector Console **Tasks**

Manage tasks.

Search or cancel multiple tasks in the [Task Manager](#).

UNSUBMITTED TASKS

 NightLight_HaNoi **RUN**

 NightLight_HaNoi  <1m

Export ongoing mark is displayed.
Export takes about 1 to 5 minutes.

Last Step (Saving the Code)

- When you click **Save**, you are prompted to enter a file name.
- Enter a name of your choice and click **OK**.

Save file

Enter a name or path for the file:

users/ /base ▼ File Name
UntitledFile

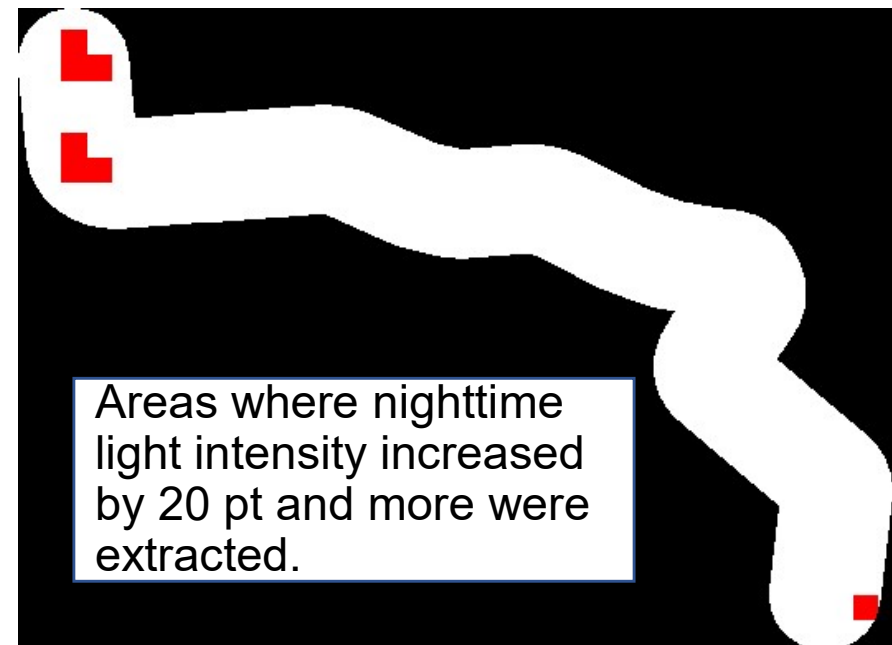
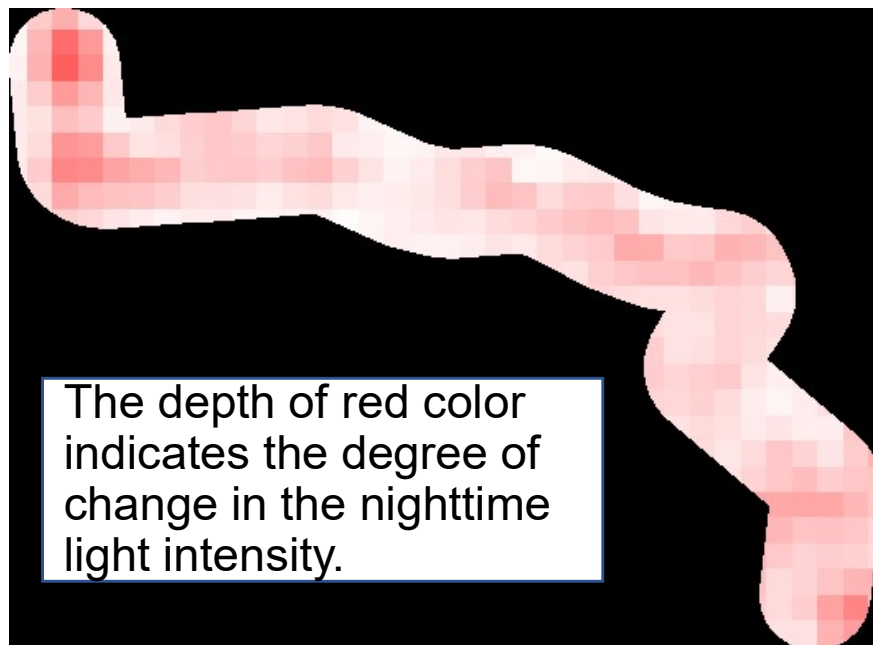
Enter description (optional):
Commit Message

CANCEL OK

Change to any name.

Tracking the Change between Two Points in Time

- This document deals only with data at one point in time. By handling data at two points in time, you can see the change in the target region.
- This will **enable the formulation of (1) detailed evaluation framework and (2) efficient on-site survey plans in the planning stage.**
- The figures below show the change in nighttime light intensity along the Purple Line route constructed with JICA Japanese ODA loans between 2013 and 2018.



Reference Materials for Studying Nighttime Light Data

- The World Bank's Open Nighttime Lights covers a comprehensive overview, history, advantages, challenges, and use of nighttime light data.
 - [Welcome — Open Nighttime Lights \(worldbank.github.io\)](https://worldbank.github.io)
- Cardille et al. (2022) "Cloud-Based Remote Sensing with Google Earth Engine: Fundamentals and Applications"
 - eefabook.org
- Donaldson etc. (2016) "The View from Above: Applications of Satellite Data in Economics": This well-known paper introduces the history, use cases, and issues of satellite data in general.
 - • • etc.