

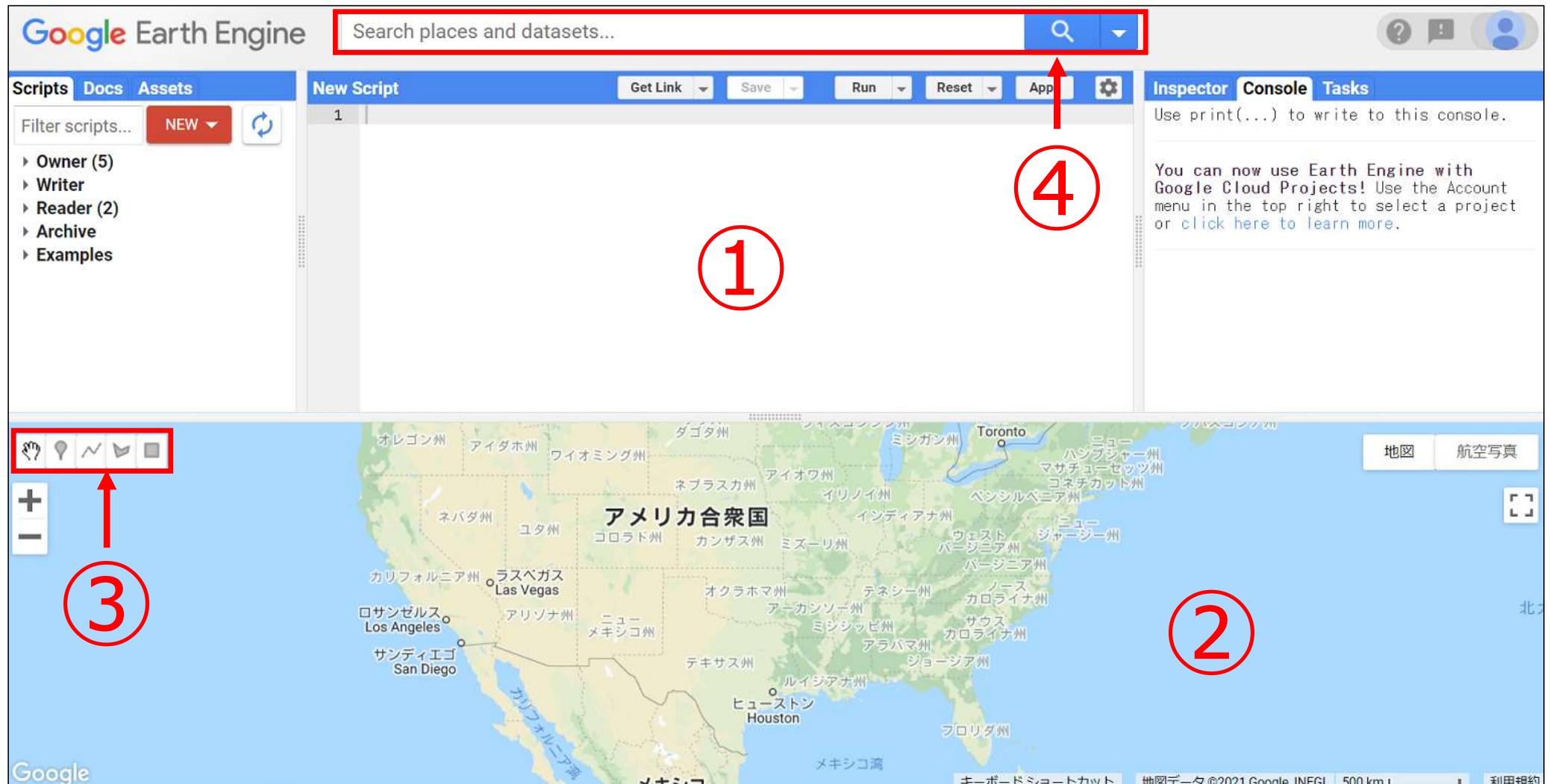
# Introduction of the Basic Handling of Satellite Data (Nighttime Light Data)

# 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

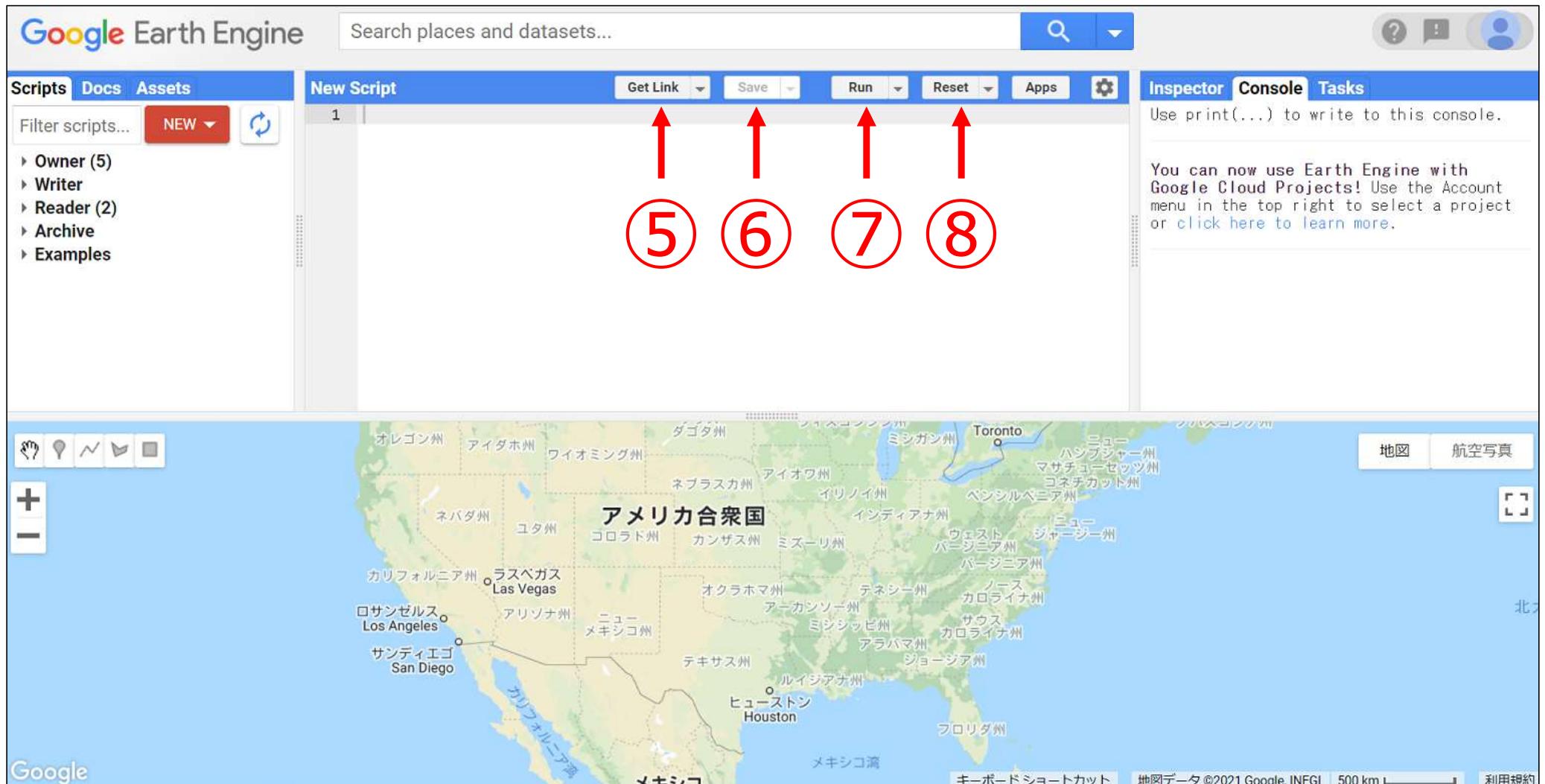
# Google Earth Engine Main Screen

# 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:

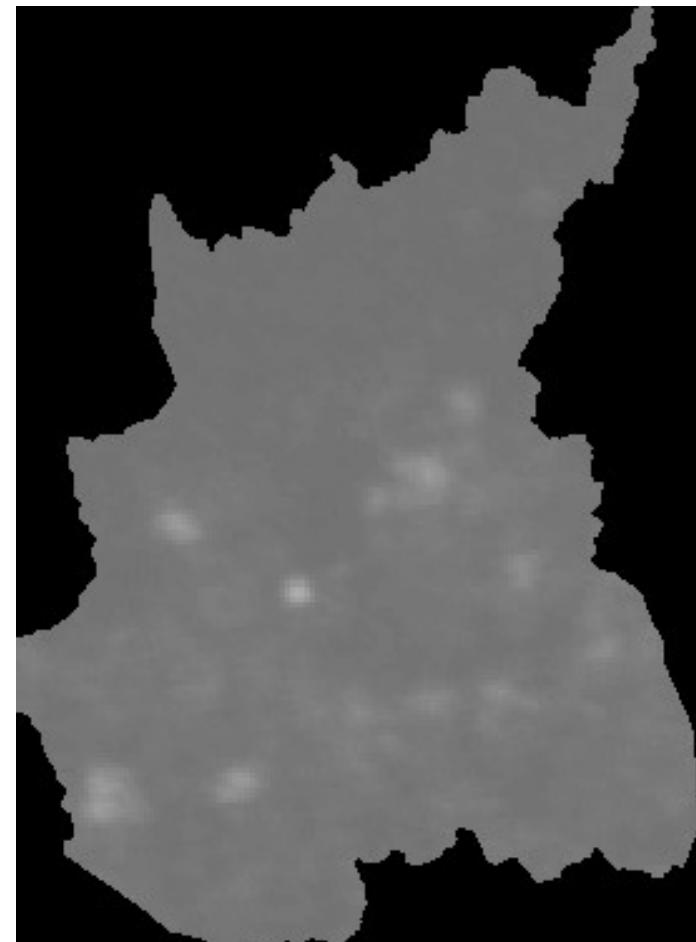
var abc = 5

- 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

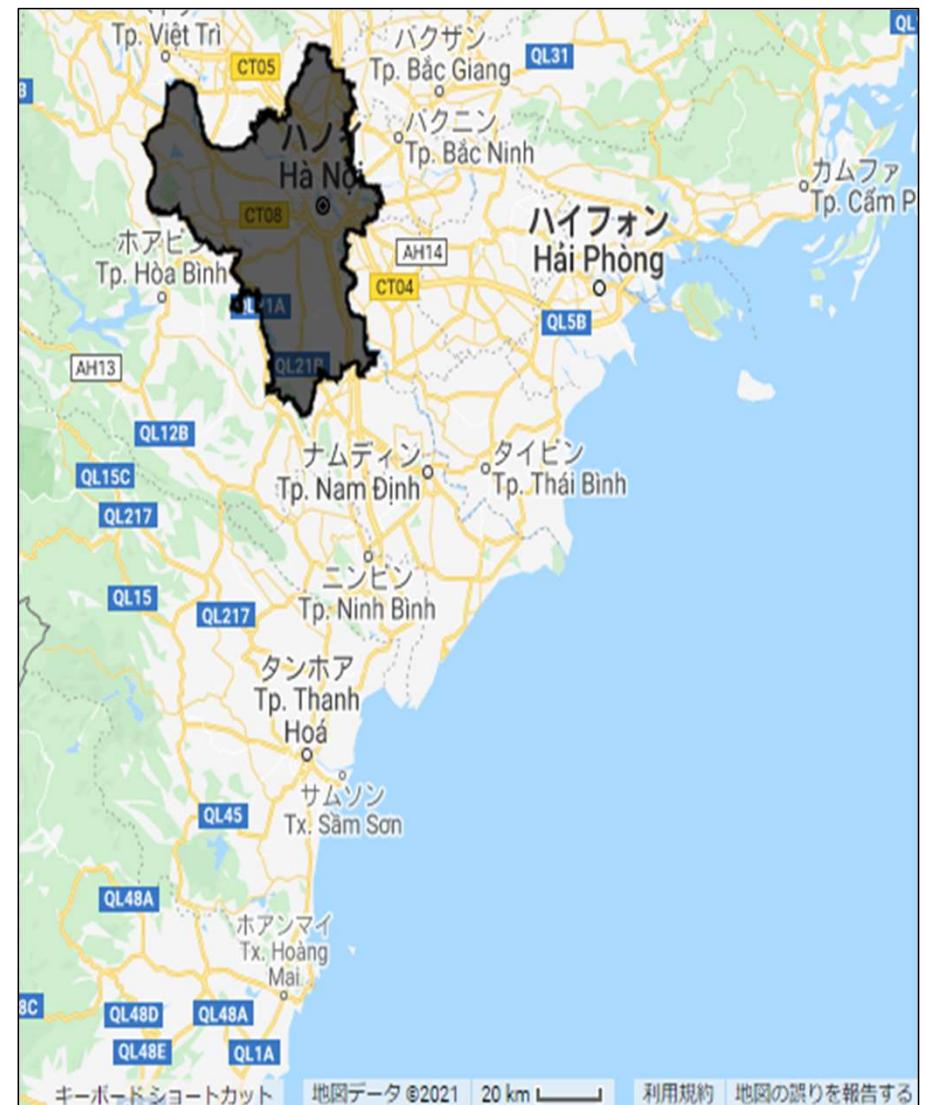
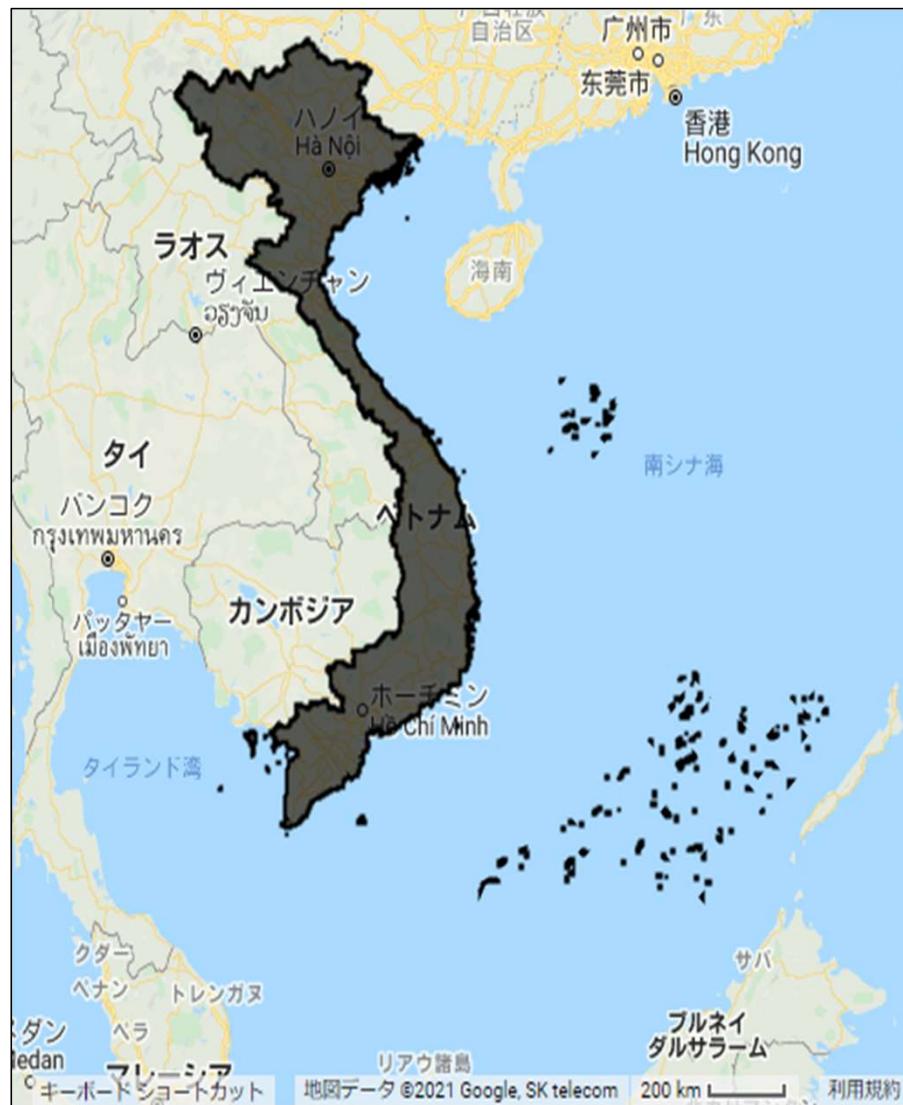
# Necessity for Handling Administrative Boundaries Data

- 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.



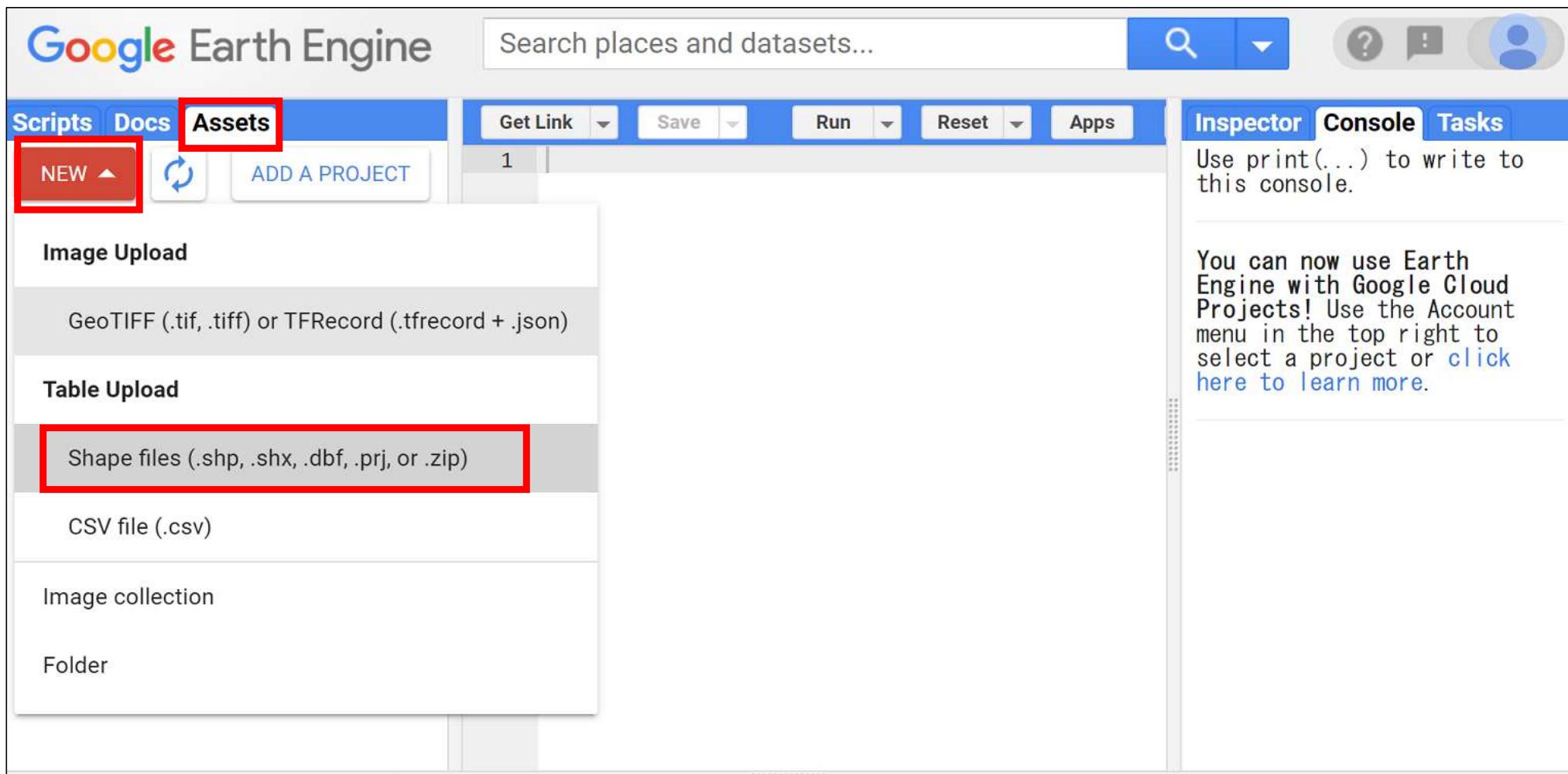
# Uploading Administrative Boundaries Data (Country Level) 1/4

- 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.

Upload a new shapefile asset

Source files **SELECT** ③

Please drag and drop or select files for this asset.  
Allowed extensions: shp, zip, dbf, prj, shx, cpg, 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/devjpconsultant/ ④ Asset Name vietnam\_lvl0

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 ⑤

# 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**.

The screenshot shows the Google Earth Engine interface. The top navigation bar includes 'Google Earth Engine', a search bar, and user account options. Below the search bar is a menu bar with 'Scripts', 'Docs', and 'Assets' selected. A sub-menu under 'Assets' shows 'NEW', 'ADD A PROJECT', and a dropdown for 'CLOUD ASSETS' and 'LEGACY ASSETS'. Under 'LEGACY ASSETS', there is a list with a red box around the item 'users/devjconsultant/vietnam\_lv10'. In the center, a 'New Script' editor window is open with a single line of code '1'. At the top of the script editor are buttons for 'Get Link', 'Save', 'Run', 'Reset', 'Apps', and a gear icon. To the right of the script editor is a 'Inspector' panel, a 'Console' panel, and a 'Tasks' panel. The 'Tasks' panel has a red box around its header and contains a single task entry: 'Ingest table: "projects/earthengine-legacy/ass...' with a checkmark and the text '1m'. At the bottom of the interface is a map of the United States with state boundaries and labels in Japanese. The map includes labels for states like 'ネバダ州', 'ユタ州', 'アリゾナ州', 'カリフォルニア州', and 'テキサス州'. Cities like 'Las Vegas', 'Los Angeles', 'San Diego', and 'Houston' are also labeled. The map is overlaid with a grid of dashed lines representing administrative boundaries.

Indicates that upload is complete

Reflected with the name entered earlier.

# 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.

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

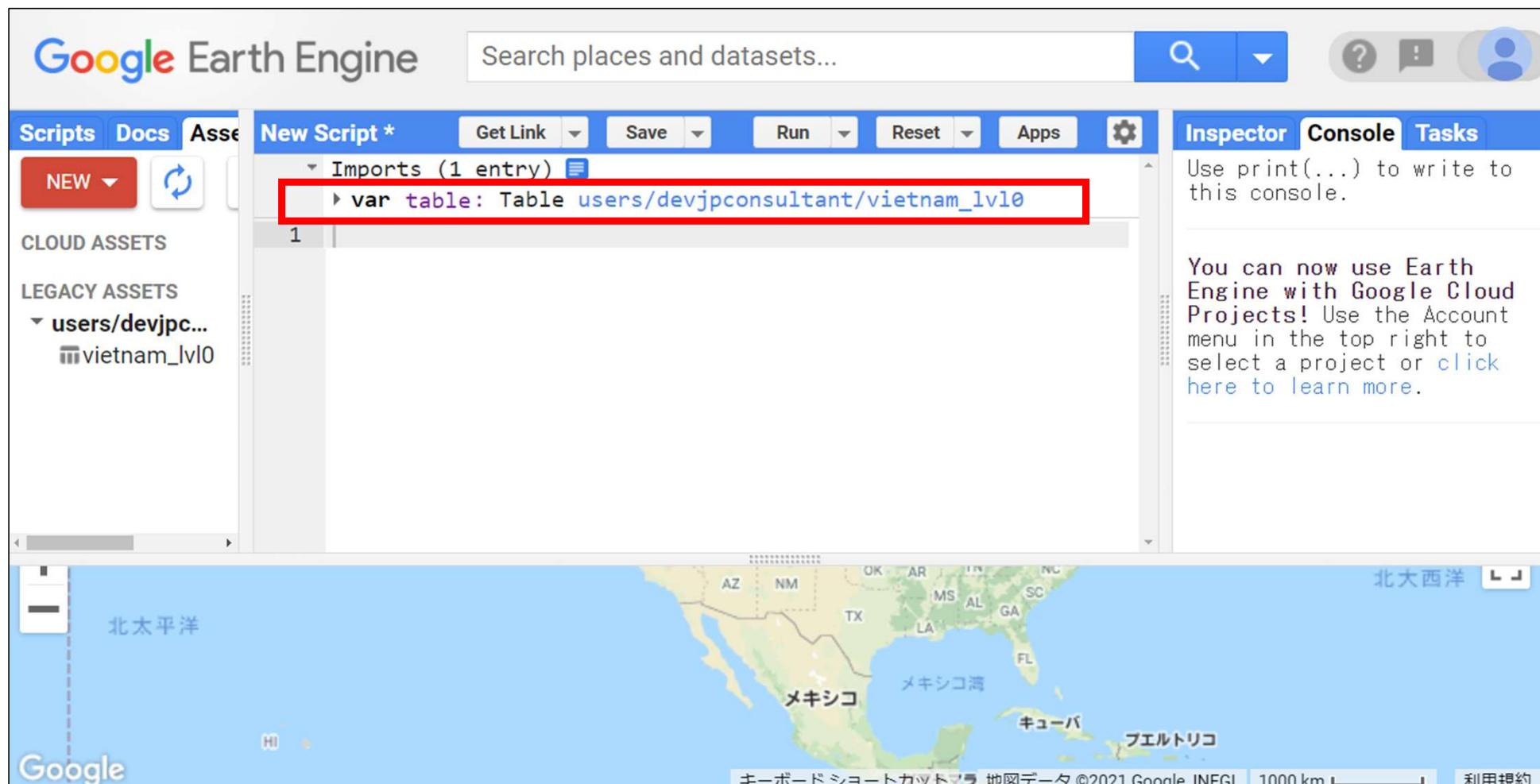
Last modified 2021-08-28 06:21:46 UTC

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

Google Earth Engine

Search places and datasets... ? ! User

Scripts Docs Assets New Script \*

Get Link Save Run Reset Apps gear

NEW refresh

CLOUD ASSETS

LEGACY ASSETS

users/devjp... viетnam\_lv10

Imports (1 entry) ↗  
var table: Table users/devjpconsultant/vietnam\_lv10

1 Map.centerObject(table, 4); ← Code that moves the map to the specified location  
2  
3 Map.addLayer(table); ← Data is visualized onto the map as Layer 1.  
4  
5 Map.addLayer(table, {}, "ベトナム"); ← Data is projected onto the map as Vietnam.  
6  
7

Layers lock 地図 航空写真

ベトナム  
Layer 1

You can change the transparency of the map by moving the bar.

バングラデシュ  
コルカタ  
Kolkata  
ムンバイ  
Mumbai  
ハイデラバード  
Hyderabad  
バンガロール  
Bengaluru  
チェンナイ  
Chennai  
ベンガル湾  
パンガル湾  
ベトナム  
カンボジア  
タイ  
ラオス  
フィリピン  
香港  
Hong Kong  
南シナ海  
アンダマン海  
スリランカ  
スヌー海  
キーボードショートカット 地図データ ©2021 Google, SK telecom 500 km 利用規約

# 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, cpg, 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\_lvl1** 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 shows the Google Earth Engine interface. The top navigation bar includes 'Google Earth Engine', a search bar, and user account options. Below the search bar is a toolbar with 'Get Link', 'Save', 'Run', 'Reset', 'Apps', and a settings gear icon. The main menu bar has tabs for 'Scripts', 'Docs', and 'Assets'. A 'New Script' tab is active. On the left, there's a sidebar for 'CLOUD ASSETS' and 'LEGACY ASSETS' under 'users/devjpconsultant', showing 'vietnam\_lv10' and 'vietnam\_lv1'. The main workspace displays a map of North America with state/province boundaries labeled in Japanese. The bottom right corner of the map area contains text in Japanese. The bottom of the screen shows standard browser controls and a footer with copyright information.

Indicates that upload is complete

↑ Ingest table: "projects/earthengi... 3m

Reflected with the name entered earlier.

# 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.

Table: vietnam\_lv11

DESCRIPTION FEATURES PROPERTIES

No description.

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

Table ID  users/devjpconsultant/vietnam\_lv11

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.”

The screenshot shows the Google Earth Engine interface. The top navigation bar includes "Google Earth Engine", a search bar, and various icons. The left sidebar has tabs for "Scripts", "Docs", and "Assets", with "Assets" selected. A red box highlights the "NEW" button. Below the sidebar, there's a "CLOUD ASSETS" section and a "LEGACY ASSETS" section containing a folder named "users/devjpconsultant/vietnam\_lv10". The main workspace is titled "New Script \*". It shows an "Imports (1 entry)" section with a single entry: "var table: Table users/devjpconsultant/vietnam\_lv11". This line is also highlighted with a red box. The bottom half of the screen displays a map of the United States with state boundaries. The map is labeled in Japanese, with state names like "ネバダ州", "ユタ州", "カリフォルニア州", "アリゾナ州", "ニューメキシコ州", "テキサス州", "ルイジアナ州", "フロリダ州", etc. The map also shows major cities like "Las Vegas", "Los Angeles", "San Diego", and "Houston". The bottom right corner of the map includes Japanese text: "地図データ ©2021 Google, INEGI 500 km 利用規約".

# 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

Google Earth Engine

Search places and datasets... ? ! User

Scripts Docs Assets New ADD

New Script \*

Imports (1 entry) Get Link Save Run Reset Apps gear

```
var table: Table users/devjpconsultant/vietnam_lvl1
Map.centerObject(table, 4);
Map.addLayer(table, {}, "ベトナム_地域レベル");
```

CLOUD ASSETS

LEGACY ASSETS

users/devjpconsultant  
vietnam\_lvl1

Code that moves the map to the specified location

Data is projected on the map as Vietnam\_Province Level

Layers lock map satellite

インド バングラデシュ ミャンマー(ビルマ) ラオス タイ カンボジア フィリピン ベトナム

ムンバイ Mumbai コルカタ Kolkata ハイデラバード Hyderabad チェンノイ Chennai バンガロール Bengaluru クアラルンプール Kuala Lumpur ホーチミン Ho Chi Minh City ハノイ Hanoi ホーチミン Ho Chi Minh City フィリピン海 南シナ海 スールー海

Administrative boundaries data for all provinces are displayed.

You can change the transparency of the map by moving the bar.

- 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

Google Earth Engine

Search places and datasets... ? ! 👤

Scripts Docs Assets New Add A

New Script \*

Imports (1 entry) Get Link Save Run Reset Apps ⚙️

```
1 var table: Table = users/devjpconsultant/vietnam_lvl1
2
3 Map.centerObject(table, 4);
4
5 var HaNoi = table.filter(ee.Filter.eq("ADM1_EN", "Ha Noi"));
6
7
8
9 Map.addLayer(HaNoi, {}, "Ha Noi");
10
```

New variable **HaNoi** stores the specified and extracted province.

Specify/Extract **Ha Noi** of **ADM1\_EN** for **table**

Layers Map Aerial Photo

Ha Noi

ベトナム\_地域レベル

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.



# How to Handle Nighttime Light Data

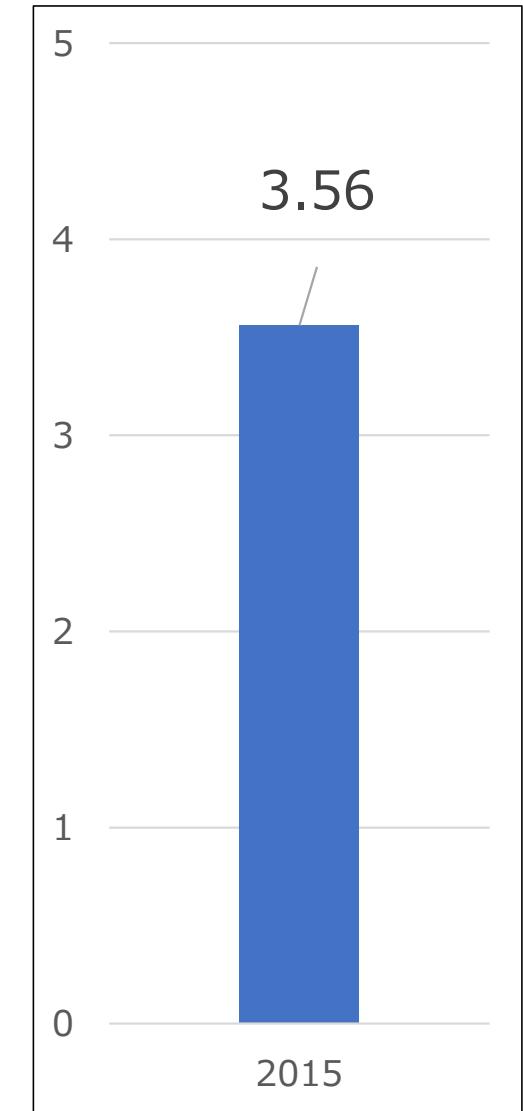
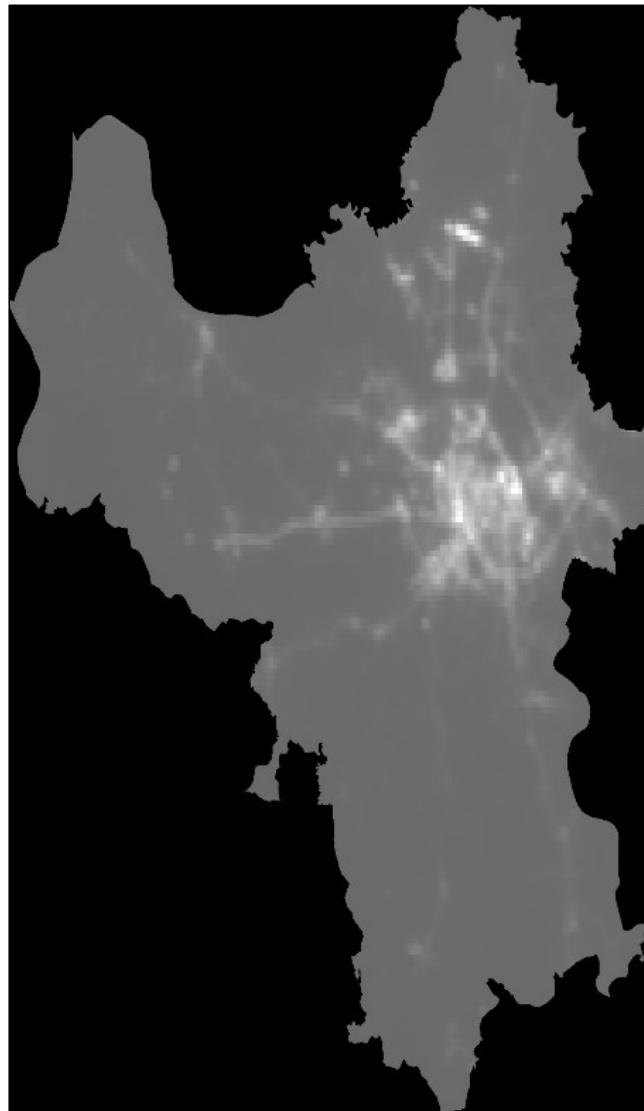
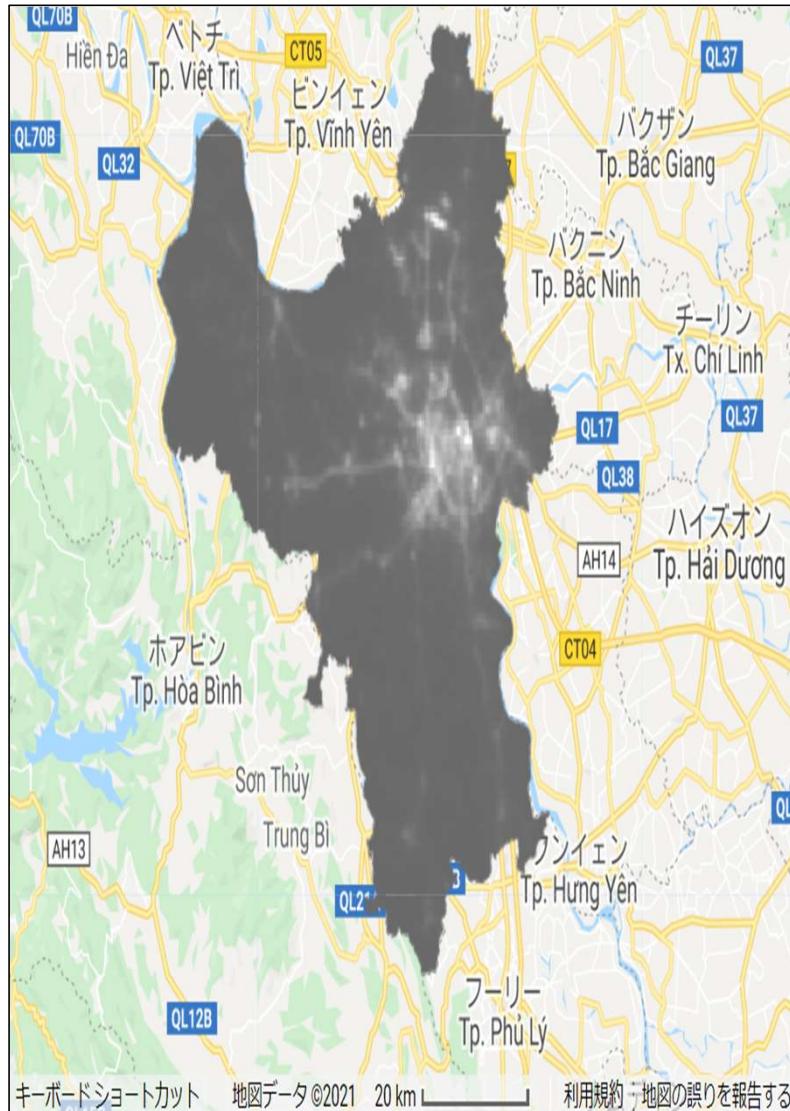
# Nighttime Light Data Stored in GEE

- 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 containing "Nighttime". Below the search bar, a list of satellite data is displayed under the heading "PLACES". A red arrow points to the search bar with the text "Type ‘Nighttime.’". Another red arrow points to the "RASTERS" section with the text "Relevant satellite data is displayed automatically.". The "RASTERS" section contains several items, with the third item, "VIIRS Nighttime Day/Night Band Composites Version 1", highlighted with a red box and an "import »" button to its right. The "RASTERS" section also includes a "more »" link at the bottom. Below the "RASTERS" section, there are sections for "TABLES" and a map view.

Nighttime ← Type “Nighttime.”

PLACES

RASTERS

DMSP OLS: Global Radiance-Calibrated **Nighttime** Lights Version 4, Defense Meteorologica...

DMSP OLS: **Nighttime** Lights Time Series Version 4, Defense Meteorological Program Oper...

**VIIRS Nighttime Day/Night Band Composites Version 1** import »

VIIRS Stray Light Corrected **Nighttime** Day/Night Band Composites Version 1

YCEO Surface Urban Heat Islands: Spatially-Averaged Monthly Composites of Daytime and...

YCEO Surface Urban Heat Islands: Spatially-Averaged Daytime and **Nighttime** Intensity for ...

YCEO Surface Urban Heat Islands: Pixel-Level Annual Daytime and **Nighttime** Intensity

YCEO Surface Urban Heat Islands: Pixel-Level Composites of Yearly Summertime Daytime ...

more »

TABLES

# 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**

**Band information**

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_cvrg' 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-temporal).		

**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")`

**Code required for data import**

Prior impacted by stray light, lightning, lunar illumination, and cloud-cover.

**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 [\[copy\]](#)

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

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/cm <sup>2</sup> /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\_cvrg' 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.

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

**CLOSE** **IMPORT**

# 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**.

Google Earth Engine      Nighttime

New Script \*

```
Imports (1 entry) ▾  
var table: Table users/devjpconsultant/vietnam_lvl1  
  
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  
8
```

**var variable name = code for nighttime light data**



Layers      地図      航空写真

QL32 CT01  
QL18  
QL23  
QL17 QL37  
QL38  
AH14 AH14 QL10  
QL5B  
MOC CHAU TOWN VÂN HỒ  
ホアビン Tp. Hòa Bình  
ソンタイ Tx. Sơn Tây  
ハノイ Hà Nội  
バクニン Tp. Bắc Ninh  
チーリン Tx. Chí Linh  
ハイズオン Tp. Hải Dương  
ハイフォン Hải Phòng  
ウオンビ Tp. Uông Bí  
Tp.  
Sơn Thủy キーボードショートカット 地図データ ©2021 20 km 利用規約 地図の誤りを報告する

# 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 shows the Google Earth Engine interface. On the left, the "New Script" panel contains the following code:

```
Imports (1 entry)
var table: Table users/devjpconsultant/vietnam_lv11
var HaNoi = table.filter(ee.Filter.eq("ADM1_EN", "Ha Noi"));
Map.centerObject(HaNoi, 9);
Map.addLayer(HaNoi, {}, "Ha Noi");
var data = ee.ImageCollection("NOAA/VIIRS/DNB/MONTHLY_V1/VCMCFG")
print(data);
```

A red box highlights the `print(data);` line. On the right, the "Console" tab is selected in the "Inspector" panel, showing the output: "Use `print(...)` to write to this console." Below it, a message from the "ImageCollection" entry in the "Tasks" list is shown: "NOAA/VIIRS/DNB/MONTHLY\_V1/VCMCFG JSON". A red arrow points from the "Information is displayed in Console." text to the "Console" tab. The main map view shows Hanoi and its surroundings with a dark gray polygon highlighting the city area.

Information is displayed in  
Console.

# 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 shows the Google Earth Engine code editor interface. The title bar says "Google Earth Engine" and "Nighttime". The toolbar includes "Get Link", "Save", "Run", "Reset", and "Apps". The code editor pane contains the following script:

```
Get Link Save Run Reset Apps
Imports (1 entry)
var table: Table users/devjpconsultant/vi
1 var HaNoi = table.filter(ee.Filter.eq("ADM1_EN
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 "Console" tab is selected in the top navigation bar. It displays the output of the script, which includes a list of 109 monthly images from April 2012 to April 2013. A red arrow points to the "version" field of the first image object, with the text "Year of data" written above it.

Inspector Console Tasks

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

▼ 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)

▶ 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 shows the Google Earth Engine interface with a script editor and an inspector panel.

**New Script \***

```
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
9 print(data);
10
```

A red arrow points upwards from the line `print(data);` in the script to the **Console** tab in the inspector panel.

**Inspector** **Console** **Tasks**

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

**ImageCollection** NOAA/VIIRS/DNB/MONTHLY\_V1/VCMCFG (12 el... JSON

- type:** ImageCollection  
**id:** NOAA/VIIRS/DNB/MONTHLY\_V1/VCMCFG  
**version:** 1633091423442399  
**bands:** []
- features:** List (12 elements)
  - 0: Image NOAA/VIIRS/DNB/MONTHLY\_V1/VCMCFG/20150101 (2 ...)
  - 1: Image NOAA/VIIRS/DNB/MONTHLY\_V1/VCMCFG/20150201 (2 ...)
  - 2: Image NOAA/VIIRS/DNB/MONTHLY\_V1/VCMCFG/20150301 (2 ...)
  - 3: Image NOAA/VIIRS/DNB/MONTHLY\_V1/VCMCFG/20150401 (2 ...)
  - 4: Image NOAA/VIIRS/DNB/MONTHLY\_V1/VCMCFG/20150501 (2 ...)
  - 5: Image NOAA/VIIRS/DNB/MONTHLY\_V1/VCMCFG/20150601 (2 ...)
  - 6: Image NOAA/VIIRS/DNB/MONTHLY\_V1/VCMCFG/20150701 (2 ...)
  - 7: Image NOAA/VIIRS/DNB/MONTHLY\_V1/VCMCFG/20150801 (2 ...)
  - 8: Image NOAA/VIIRS/DNB/MONTHLY\_V1/VCMCFG/20150901 (2 ...)
  - 9: Image NOAA/VIIRS/DNB/MONTHLY\_V1/VCMCFG/20151001 (2 ...)
  - 10: Image NOAA/VIIRS/DNB/MONTHLY\_V1/VCMCFG/20151101 (2 ...)
  - 11: Image NOAA/VIIRS/DNB/MONTHLY\_V1/VCMCFG/20151201 (2 ...)
- properties:** Object (23 properties)

A red arrow points downwards from the **2015 only** text in the **Console** tab to the **properties** object in the **Inspector** panel.

**Don't forget to put the semicolon.**

# 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 with a "New Script\*" tab. The code editor contains the following script:

```
Imports (1 entry)
var table: Table users/devjpconsultant/vietnam_lvl1

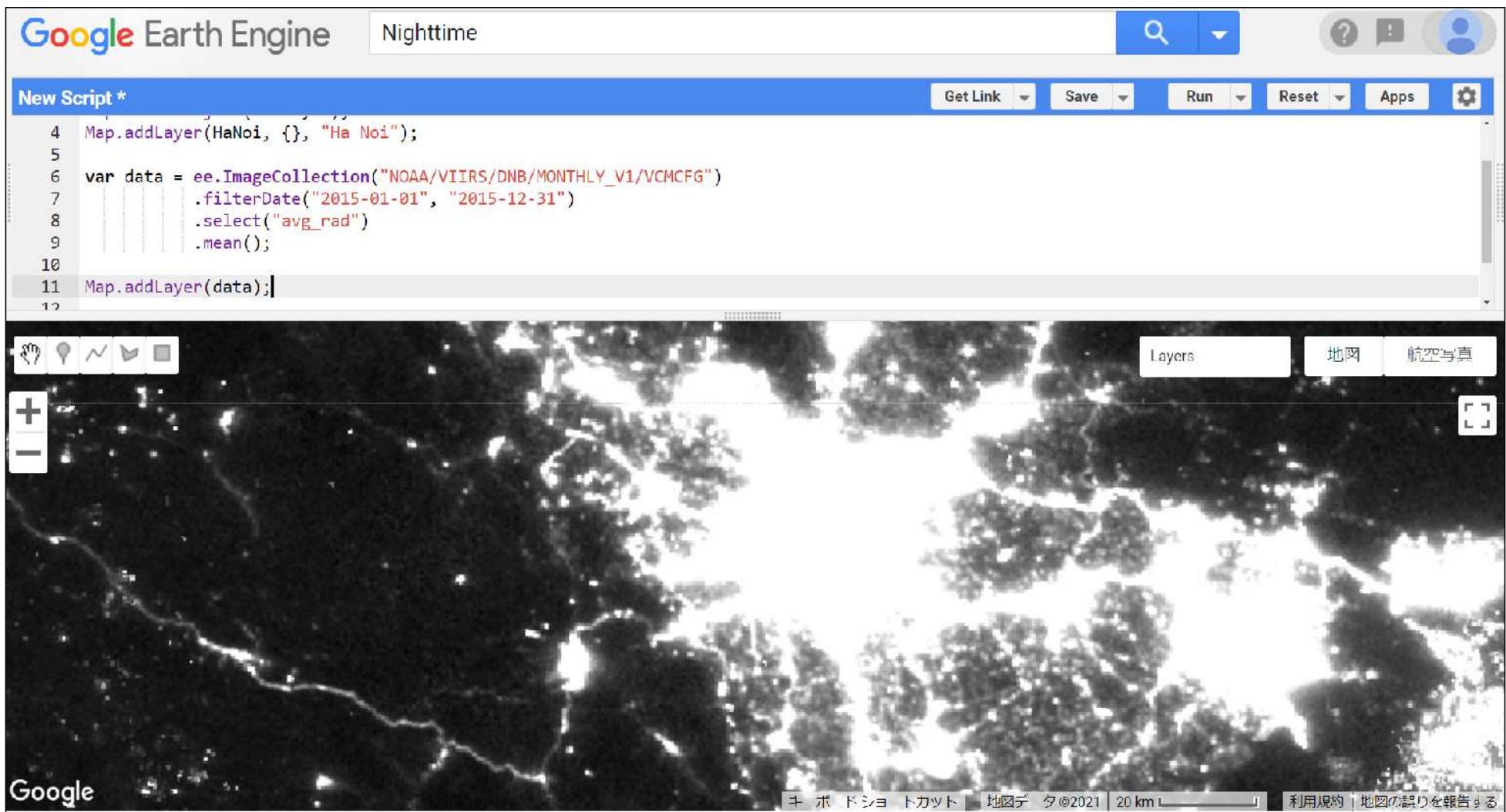
1 var HaNoi = table.filter(ee.Filter.eq("ADM1_EN", "Ha Noi"));
2
3 Map.centerObject(HaNoi, 9);
4 Map.addLayer(HaNoi, {}, "Ha Noi");
5
6 var data = ee.ImageCollection("NOAA/VIIRS/DNB/MONTHLY_V1/VCMCFG")
7   .filterDate("2015-01-01", "2015-12-31")
8   .select("avg_rad") ← Select band.
9   .mean();
10
11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

Annotations in red highlight specific parts of the code:

- An arrow points to the `.select("avg_rad")` line with the text "Select band."
- An arrow points to the `.mean();` line with the text "Specify the use of average. (you can also use min, max, median, etc.)"

# 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.



The screenshot shows the Google Earth Engine interface. At the top, there's a search bar with the text "Nighttime". Below it is a script editor titled "New Script \*". The script code is:

```
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 Map.addLayer(data);
12
```

The main area displays a grayscale map of the world, where bright white and yellow areas represent nighttime light intensity. The map includes various controls at the bottom: a toolbar with icons for selection, measurement, and zoom; a "Layers" button; a "地図" (Map) button; an "航空写真" (Aerial Photo) button; and zoom controls (+, -, fit).

# Specifying Projection Regions

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

Google Earth Engine

Nighttime

New Script \*

```
Map.centerObject(HaNoi, 9);
Map.addLayer(HaNoi, {}, "Ha Noi");

var data = ee.ImageCollection("NOAA/VIIRS/DNB/MONTHLY_V1/VCMCFG")
  .filterDate("2015-01-01", "2015-12-31")
  .select("avg_rad")
  .mean();

Map.addLayer(data.clip(HaNoi));| ← variable name.clip(variable name)
```

variable name.clip(variable name)

The map displays several regions, including Ha Noi (Hanoi), Tp. Việt Trì, Tp. Vĩnh Yên, Tp. Bắc Giang, Tp. Bắc Ninh, Tx. Chí Linh, Tp. Uông Bí, and Hải Phòng. Major roads are labeled with codes like QL32B, QL70B, QL32, QL43, AH14, QL17, QL38, CT04, QL10, QL37, QL5B, and QL37. The script editor shows a snippet of code for filtering a VIIRS DNB monthly collection from January 2015 to December 2015, selecting the 'avg\_rad' band, calculating the mean, and adding the result as a layer clipped to the Ha Noi geometry.

# 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**.

The screenshot shows the Google Earth Engine script editor interface. The title bar says "Google Earth Engine" and the search bar contains "Nighttime". The main area is titled "New Script \*". The code is as follows:

```
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");
```

Annotations in red text provide additional information:

- "Variable for parameters" points to the line `var visParams = {min: 0, max: 60, palette:[ "#666666", "#FFFFFF" ]};`
- "<Settings>" points to the values `min: 0` and `max: 60`.
- "Minimum: 0" and "Maximum: 60" are listed below the settings.
- "Projection color: gray, white" is listed below the settings.
- "<Projection mechanism>" points to the line `Map.addLayer(data.clip(HaNoi), visParams, "Night Light");`
- "The minimum value is projected in gray, and as the value increases, the projection becomes whiter." is described below the mechanism.
- An arrow points from the text "Embed between variable and layer names." to the comma after `visParams` in the final line of code.

Embed between variable and layer names.

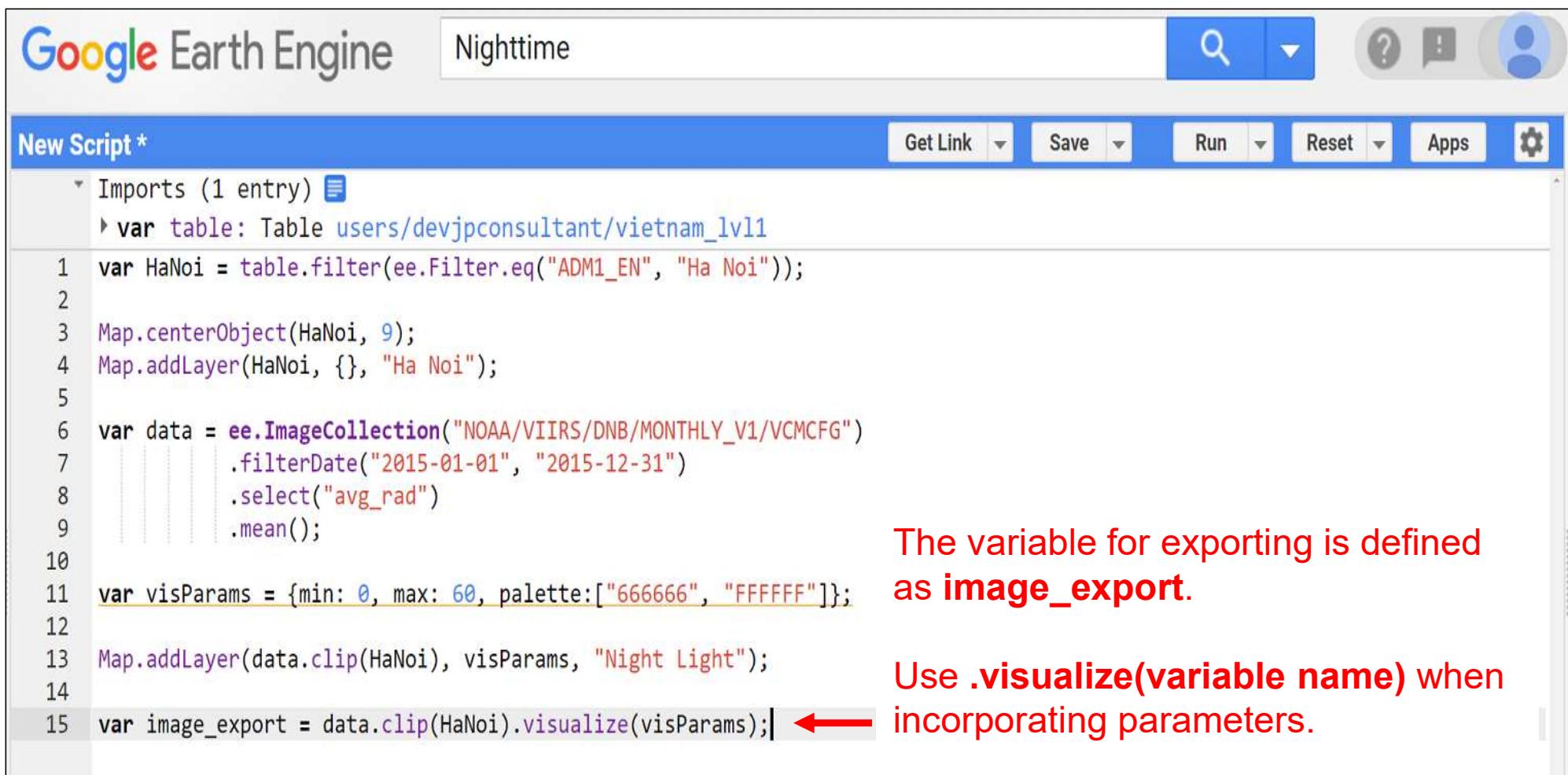
# Projection of Nighttime Light Data for 2015 in Hanoi

- 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).



The screenshot shows the Google Earth Engine interface with a search bar containing "Nighttime". The main area is a "New Script" editor with the following code:

```
Imports (1 entry)
var table: Table users/devjpconsultant/vietnam_lvl1

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");
14
15 var image_export = data.clip(HaNoi).visualize(visParams);
```

Annotations in red text are overlaid on the code:

- "The variable for exporting is defined as **image\_export**.
- "Use **.visualize(variable name)** when incorporating parameters." An arrow points to the line `var image_export = data.clip(HaNoi).visualize(visParams);`.

# 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 | Settings | Help | Search | More
```

```
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
```

The screenshot shows the Google Earth Engine interface with a script editor. The script is written in JavaScript and performs the following steps:

- Selects the "avg\_rad" band from the data.
- Calculates the mean of the selected band.
- Creates a visualization configuration object (`visParams`) with a color palette ranging from dark gray (666666) to white (FFFFFF).
- Adds a layer to the map using the visualized data and the configuration object.
- Creates a variable (`image_export`) representing the visualized data.
- Exports the data as an image to Google Drive using the `Export.image.toDrive` function.

A red box highlights the export code (lines 17-24). To the right of the code, red annotations explain the parameters of the `Export.image.toDrive` function:

- `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.

Google Earth Engine

UntitledFile

Search places and datasets...

Get Link Save Run Reset Apps Inspector Console Tasks

```
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");
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
```

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.

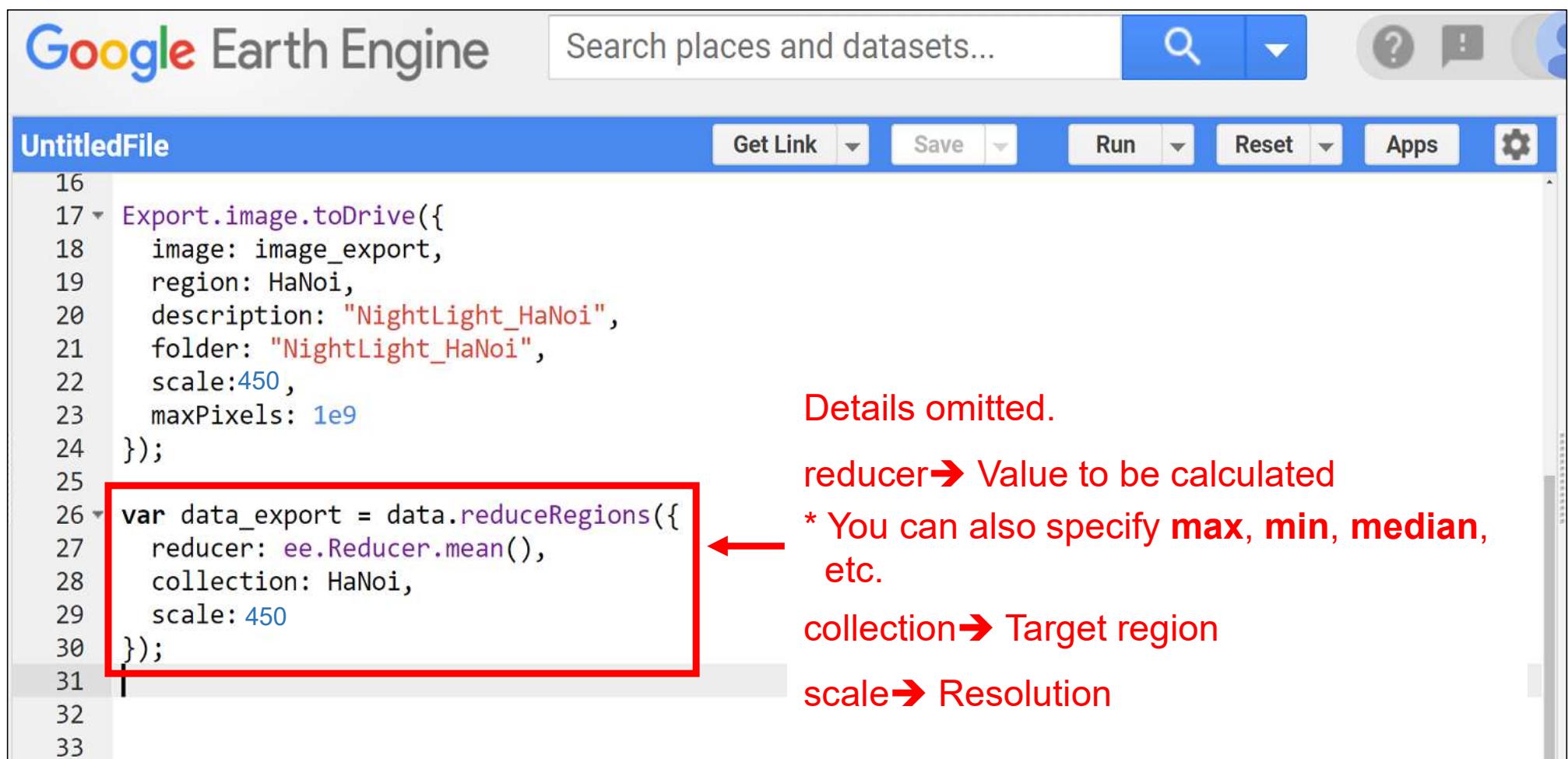
Search or cancel multiple tasks in the Task Manager [Task Manager](#).

 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 title bar says "UntitledFile". The toolbar includes "Get Link", "Save", "Run", "Reset", and "Apps". A search bar at the top right says "Search places and datasets..." with a magnifying glass icon. The main area contains the following code:

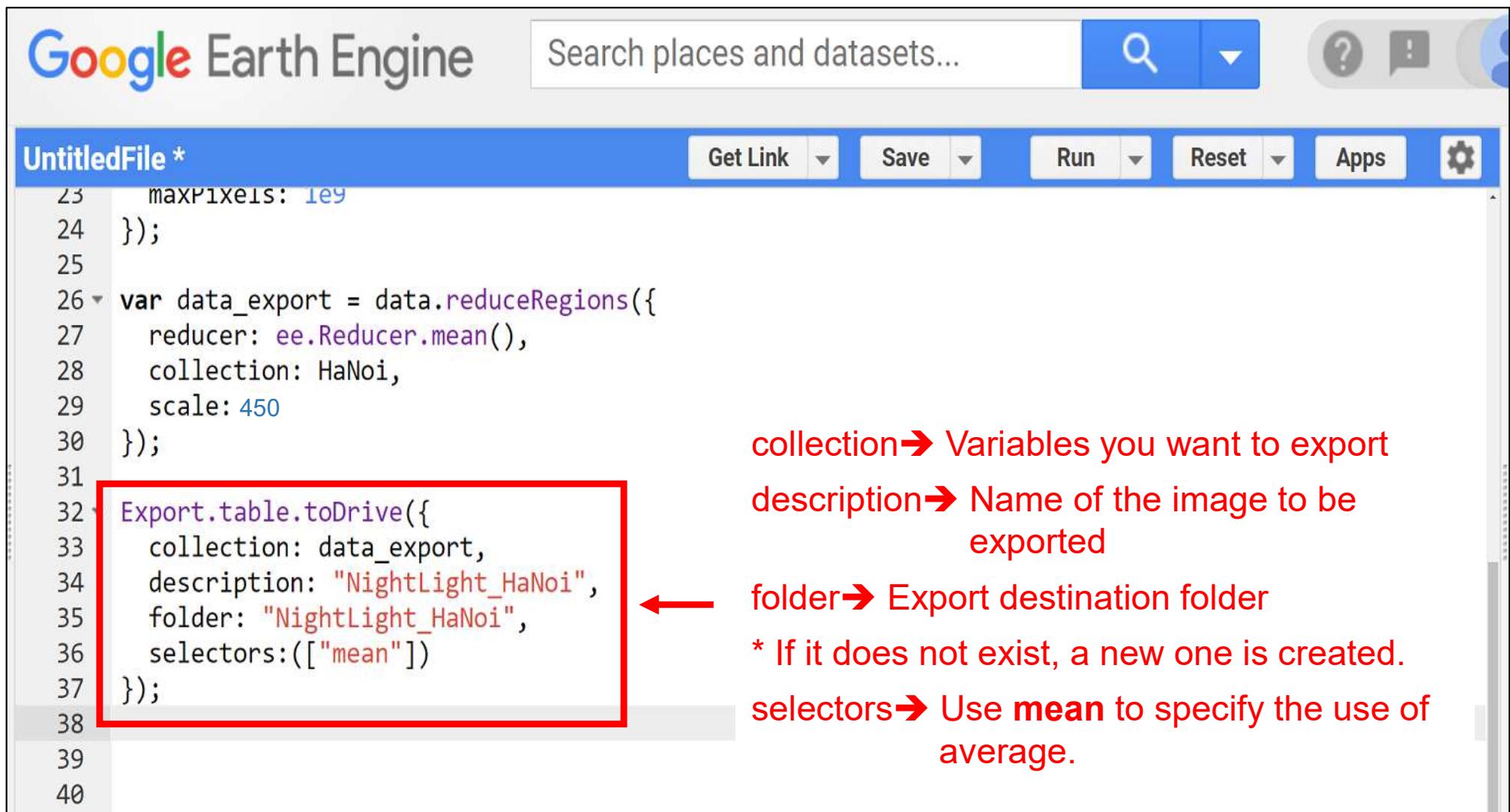
```
16
17 -> Export.image.toDrive({
18   image: image_export,
19   region: HaNoi,
20   description: "NightLight_HaNoi",
21   folder: "NightLight_HaNoi",
22   scale: 450,
23   maxPixels: 1e9
24 });
25
26 -> var data_export = data.reduceRegions({
27   reducer: ee.Reducer.mean(),
28   collection: HaNoi,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

A red box highlights the code from line 26 to 30. A red arrow points from the explanatory text for "reducer" to this highlighted code. The explanatory text is as follows:

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 code editor interface. The title bar says "UntitledFile \*". The code area contains 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
```

A red box highlights the last four lines of code (32-37). A red arrow points from the explanatory text for "collection" to this highlighted block.

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 interface. On the left, the code editor displays a script for exporting nighttime light data. The script includes code for setting maxPixels, defining a reducer, specifying a collection (HaNoi), and setting a scale. It then uses the Export.table.toDrive method to create a table in Google Drive with a specific description, folder, and selector. On the right, the Task Manager is open, showing two tasks listed under 'UNSUBMITTED TASKS'. Both tasks are named 'NightLight\_HaNoi' and have a 'RUN' button next to them. A red arrow points from the text 'Values are ready for export. Click RUN.' to the first 'RUN' button. The 'Tasks' tab in the top navigation bar is highlighted with a red box.

```
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
```

Search places and datasets... ? ! ...

Get Link Save Run Reset Apps ...

Inspector Console **Tasks**

Manage tasks.

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

UNSUBMITTED TASKS

NightLight_HaNoi	<span>RUN</span>
NightLight_HaNoi	<span>RUN</span>

Values are ready for export.  
Click RUN.

# 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 [Task Manager](#).

UNSUBMITTED TASKS

NightLight\_HaNoi **RUN**

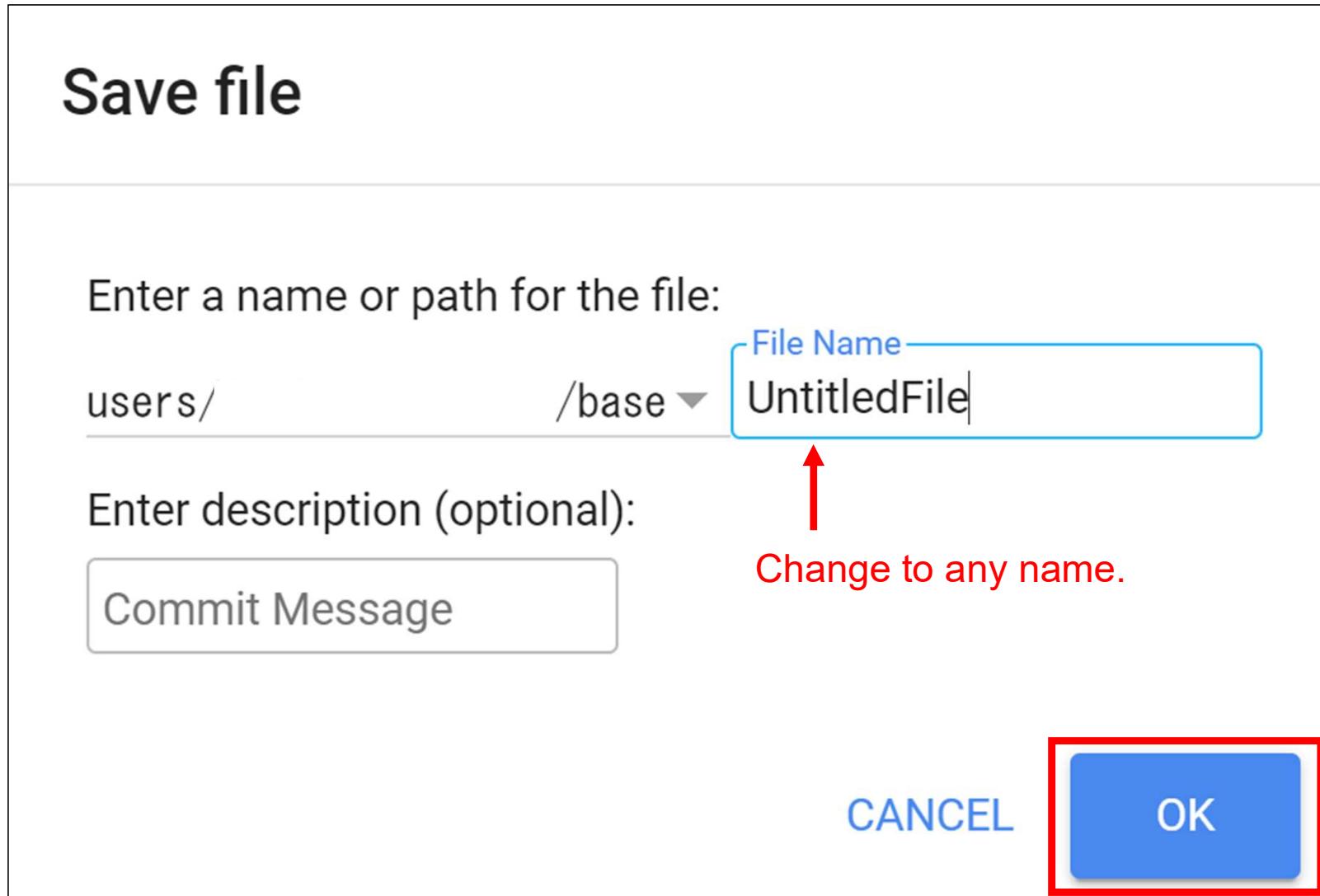
NightLight\_HaNoi <1m

↑

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

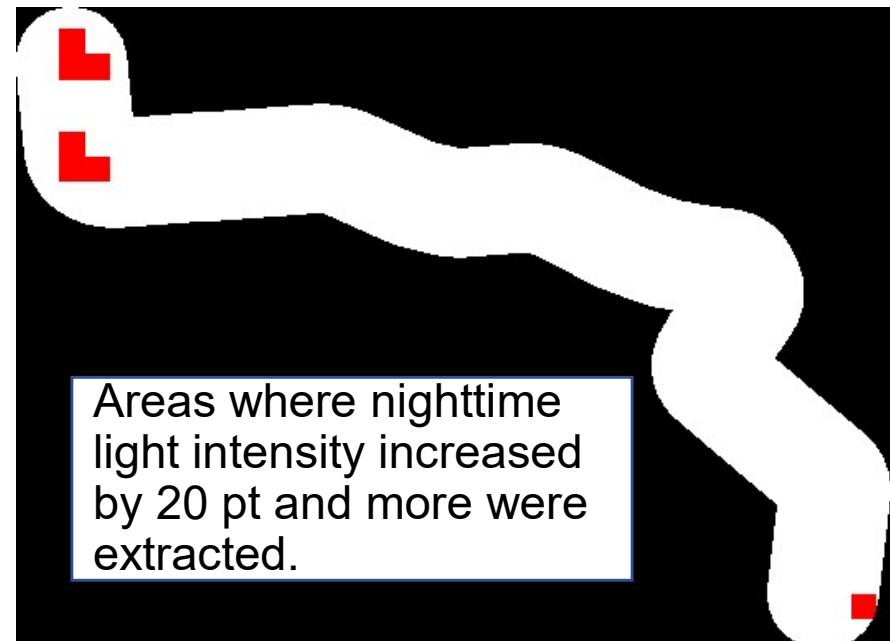
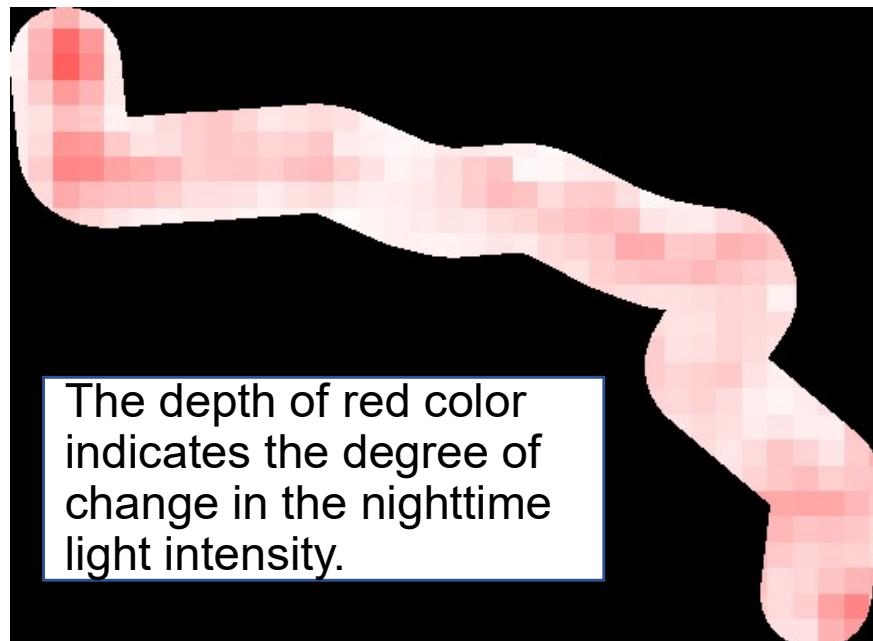
# Last Step (Saving the Code)

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



# 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/Open-Nighttime-Lights/)
- Cardille et al. (2022) “Cloud-Based Remote Sensing with Google Earth Engine: Fundamentals and Applications”
  - [eefabook.org](http://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.