

Lab 4: AI-Assisted Programming for Google Earth Engine

Complete ChatGPT Integration Tutorial for Environmental Analysis

AI-Powered Coding Earth Engine Integration Educational Best Practices

Table of Contents

- 1. Learning Objectives
- 2. AI in Education & Programming
- 3. Introduction to ChatGPT for Coding
- 4. GEE-Specific AI Applications
- 5. Prompt Engineering Best Practices
- 6. Step-by-Step Tutorial
- 7. Debugging with AI
- 8. Professional Development

Learning Objectives

Technical Skills

- Generate GEE code using natural language prompts
- Debug JavaScript errors with AI
- Understand complex Earth Engine functions
- Optimize code for better performance

Learning Outcomes

- Build confidence in programming
- Develop efficient coding workflows
- Master prompt engineering techniques
- Apply ethical AI practices

Lab Overview

This lab transforms your approach to Google Earth Engine programming by integrating ChatGPT as your coding assistant. You'll learn to leverage AI for writing, understanding, and debugging GEE scripts while maintaining scientific rigor and developing genuine programming expertise. By the end, you'll have a powerful workflow that accelerates your environmental analysis capabilities.

AI in Education & Programming

Benefits of AI Coding Assistants

Accelerated Learning

AI assistants provide instant explanations, helping you understand complex concepts and syntax without getting stuck on implementation details.

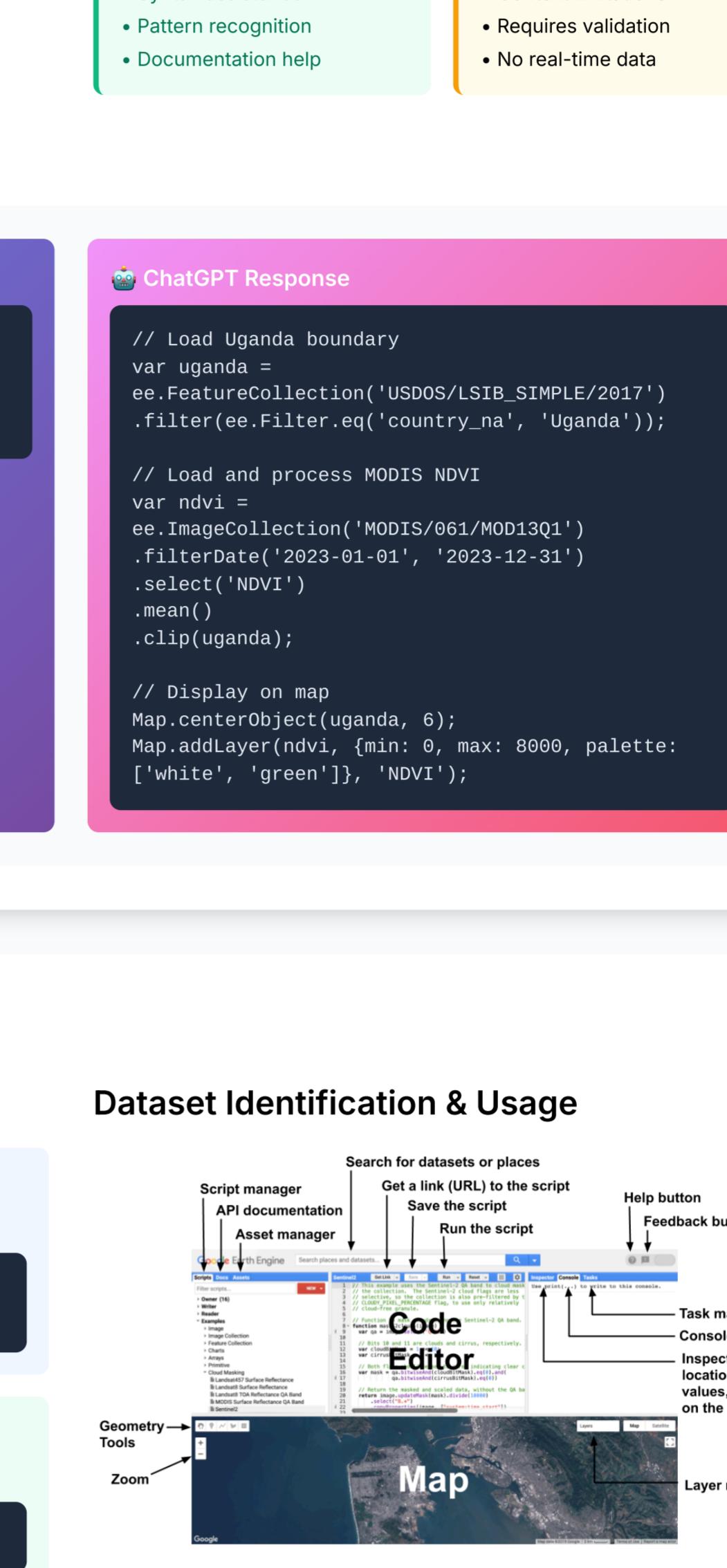
Confidence Building

Reduces programming anxiety by providing a supportive learning environment where you can experiment and make mistakes safely.

Focused Problem-Solving

Allows you to focus on scientific questions rather than getting bogged down in syntax and implementation details.

Educational Technology Integration



Modern Learning Paradigm

AI tools are transforming education by providing personalized, interactive learning experiences that adapt to individual learning styles and pace.

Responsible AI Use in Academic Settings



Ethical Guidelines

- Always cite AI assistance
- Understand generated code
- Verify all outputs
- Use as learning tool



Academic Integrity

- Supplement, don't replace learning
- Develop genuine understanding
- Practice independent coding
- Build core competencies



Best Practices

- Start with simple tasks
- Gradually increase complexity
- Always test and validate
- Document your process

Introduction to ChatGPT for Coding

Understanding Large Language Models

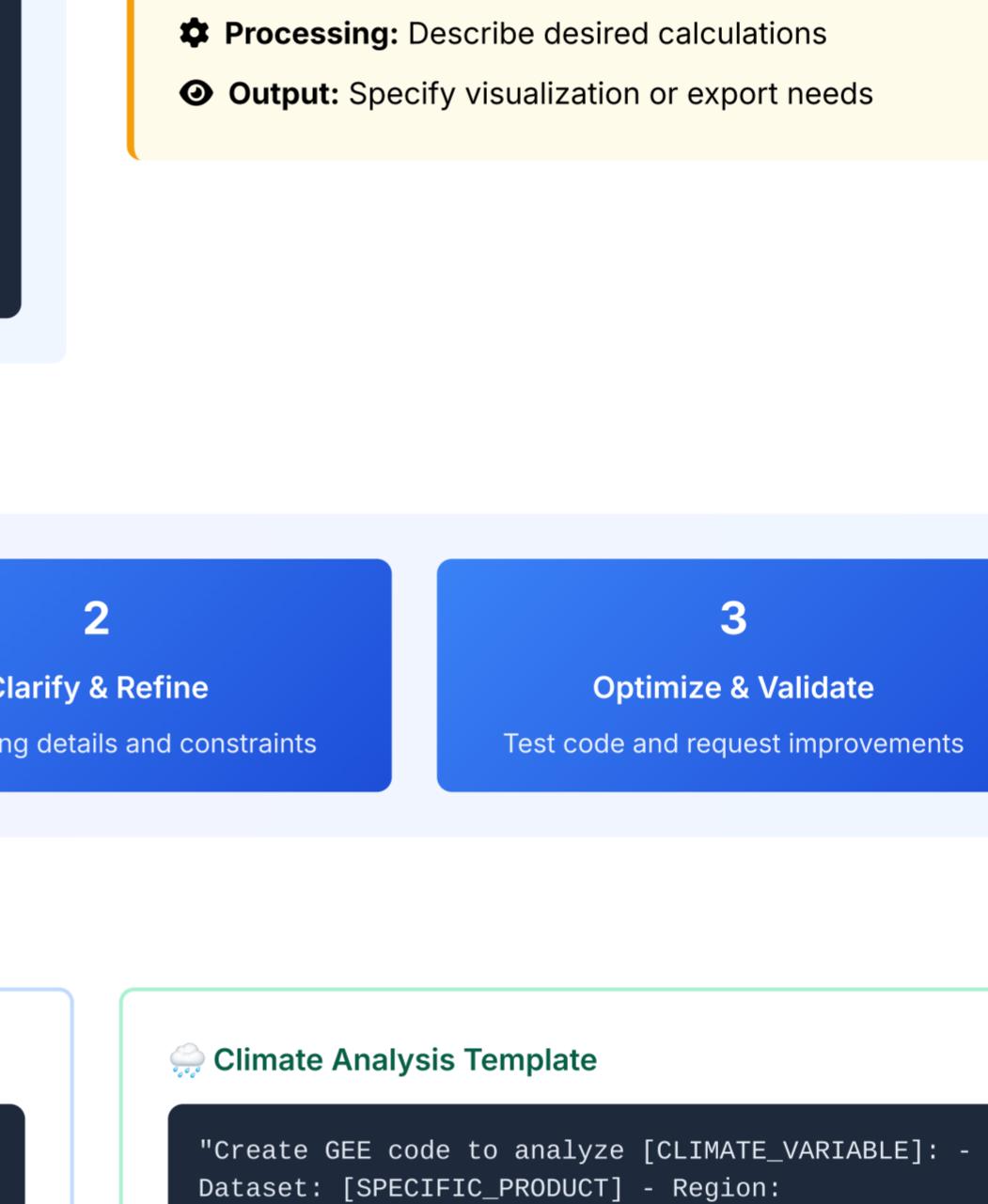
How ChatGPT Works for Programming

ChatGPT is trained on vast amounts of code and documentation, allowing it to understand programming patterns, syntax, and best practices across multiple languages including JavaScript for Google Earth Engine.

Key Capabilities:

- Code generation from natural language
- Function and algorithm explanation
- Error diagnosis and debugging
- Code optimization suggestions
- Documentation and commenting

Capabilities and Limitations



Conversational Programming Approach

Your Prompt

```
"I need to load MODIS NDVI data for Uganda in 2023, calculate the mean, and display it with a color palette. Can you write the GEE JavaScript code?"
```

ChatGPT Response

```
// Load Uganda boundary var uganda = ee.FeatureCollection('USDOS/LSTB_SIMPLE/2017'); // Load MODIS NDVI data for Uganda var modis = ee.ImageCollection('MODIS/06/MOD13Q1').filterDate('2023-01-01', '2023-12-31').select('NDVI').map(function(image){ return image.clip(uganda); }); // Display on map Map.setCenter(uganda, 6); Map.addLayer(ndvi, {min: 0, max: 8000, palette: ['white', 'green']}, 'NDVI');
```

GEE-Specific AI Applications

JavaScript Syntax Assistance

Function Syntax Help

Ask ChatGPT for specific function syntax.

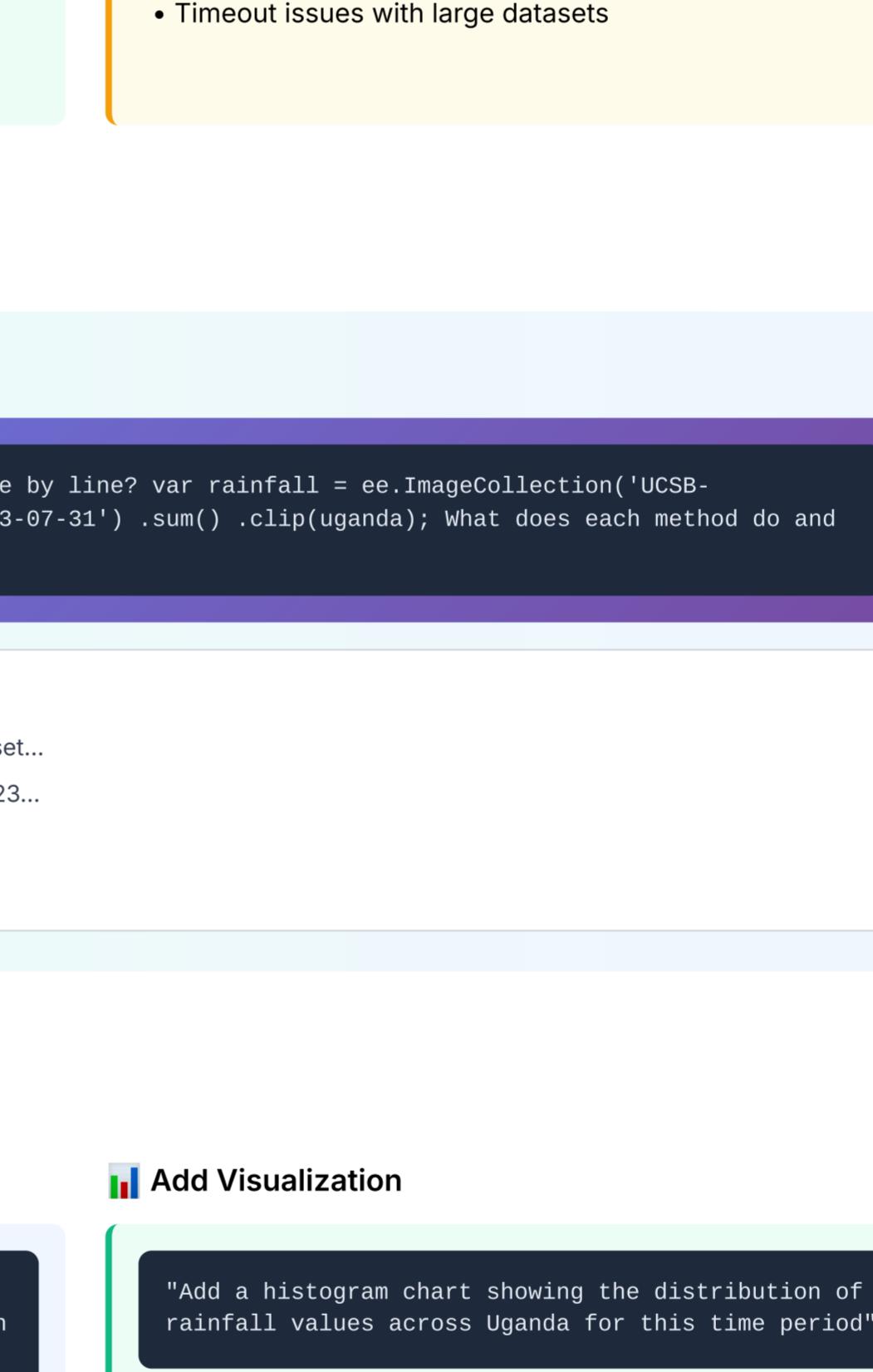
```
"How do I use ee.ImageCollection.filterDate() in Google Earth Engine?"
```

Parameter Explanation

Get detailed parameter explanations.

```
"Explain the parameters for Map.addLayer() in GEE"
```

Dataset Identification & Usage



Common GEE Tasks with AI Assistance

Data Loading

Get help with ImageCollection and FeatureCollection syntax.

Filtering

Learn temporal and spatial filtering techniques.

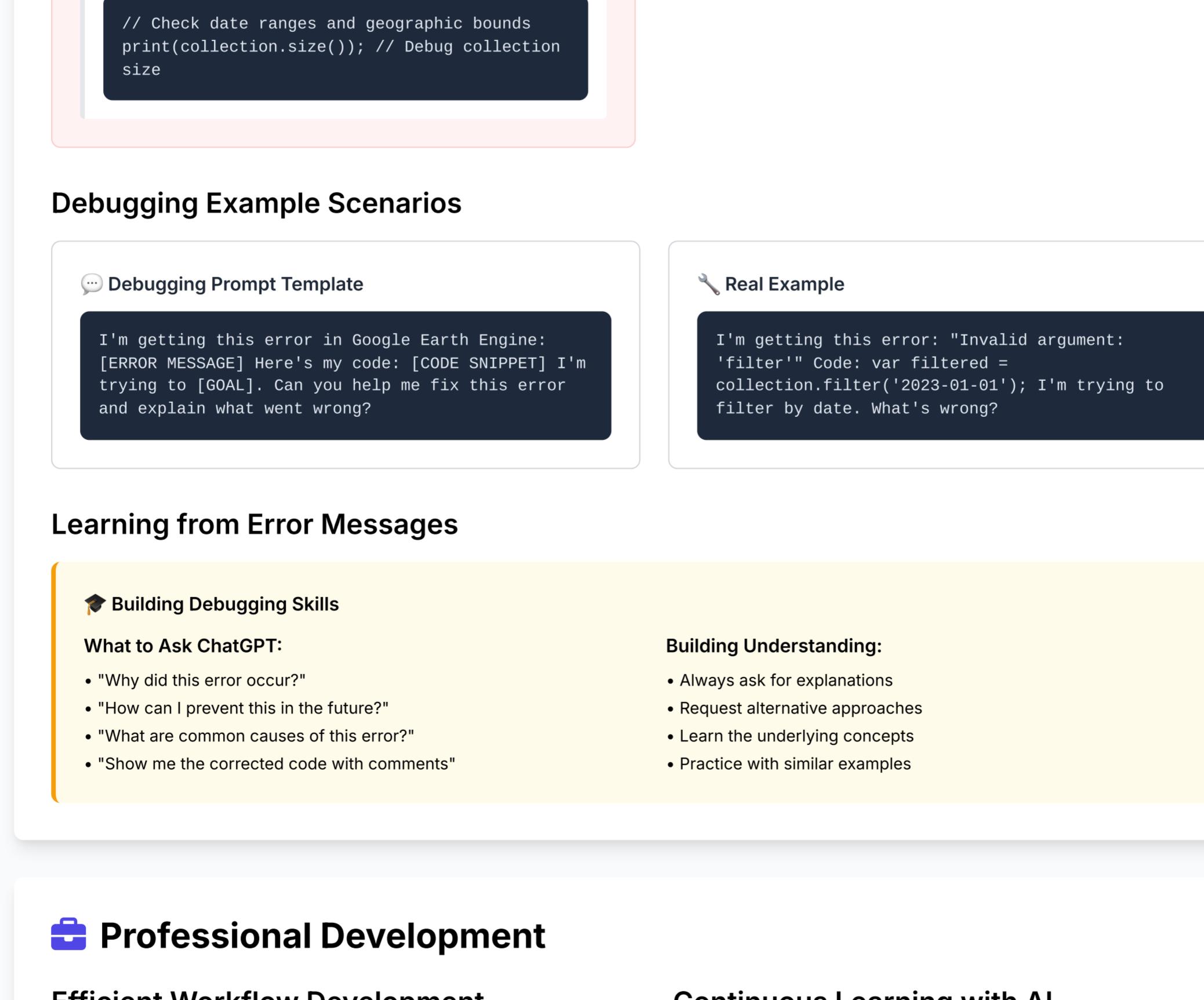
Processing

Master calculations, aggregations, and transformations.

Export

Configure exports to Drive, Assets, or Cloud Storage.

Prompt Engineering Best Practices



Clear and Specific Prompt Construction

Specificity Principle

The more specific your prompt, the better the response. Include context, constraints, and desired outcomes.

Vague Prompt

```
"Help me with NDVI in GEE"
```

Specific Prompt

```
"Write Google Earth Engine JavaScript code to: 1. Load [DATASET] for [LOCATION] 2. Filter for [TIME_PERIOD] 3. Calculate [METRIC] (e.g., mean NDVI) 4. Clip into [BOUNDARY] 5. Visualize with [COLOR_PALETTE] 6. Export as [FORMAT] to [DESTINATION]"
```

Providing Context and Requirements

Essential Elements

Geographic Area: Specify country, region, or coordinates.

Time Period: Define start and end dates.

Dataset: Mention specific satellite products.

Processing: Describe desired calculations.

Output: Specify visualization or export needs.

Iterative Prompt Refinement

Initial Prompt

Start with basic requirements

Clarify & Refine

Add missing details and constraints

Optimize & Validate

Test code and request improvements

Example Prompt Templates

Vegetation Analysis Template

```
"Write Google Earth Engine JavaScript code to: 1. Load [DATASET] for [LOCATION] 2. Filter for [TIME_PERIOD] 3. Calculate [METRIC] (e.g., mean NDVI) 4. Clip into [BOUNDARY] 5. Visualize with [COLOR_PALETTE] 6. Export as [FORMAT] to [DESTINATION]"
```

Climate Analysis Template

```
"Create GEE code to analyze [CLIMATE_VARIABLE]: - Dataset: [SPECIFIC_PRODUCT] - Region: [GEOSPATIAL_BOUNDS] - Period: [TEMPORAL_RANGE] - Processing: [AGGREGATION_METHOD] - Quality: [QUALITY_LEVEL] - Requirements: [REQUIREMENTS] - Output: [VISUALIZATION_AND_EXPORT]"
```

Step-by-Step Tutorial

1 Initial Setup

Open Required Platforms

Google Earth Engine

Navigate to [code.earthengine.google.com](#)

ChatGPT

Open [chat.openai.com](#) in a new tab

Create New Script

// In GEE Code Editor: // 1. Click "NEW" button // 2. Name: "Lab4_AI_Assistant" // 3. Save in your workshop folder

2 Generate Your First Code

Copy This Prompt to ChatGPT:

```
write Google Earth Engine JavaScript code that: 1. Loads the CHIRPS Daily rainfall dataset ('UCSB-CHIRPS/DAILY') 2. Filters data for July 2023 (2023-07-01 to 2023-07-31) 3. Calculates the total rainfall over the month 4. Displays the result on the map with a blue color palette (white to dark blue) 5. Centers the map on Uganda with zoom level 6 6. Adds the layer with the name "July 2023 Rainfall" Please include comments explaining each step.
```

Expected Response Structure:
1. Load the CHIRPS Daily rainfall dataset ('UCSB-CHIRPS/DAILY')
2. Filters data for July 2023 (2023-07-01 to 2023-07-31)
3. Calculates the total rainfall over the month
4. Displays the result on the map with a blue color palette (white to dark blue)
5. Centers the map on Uganda with zoom level 6
6. Adds the layer with the name "July 2023 Rainfall"

3 Test and Validate Code

Testing Process

- Copy code from ChatGPT to GEE
- Click "Run" button
- Check console for errors
- Verify map display
- Inspect data values

Common Issues

- Incorrect dataset IDs
- Wrong date formats
- Visualization parameter errors
- Timeout issues with large datasets

4 Get Code Explanations

Ask ChatGPT to Explain Code

Can you explain this Google Earth Engine code line by line? var rainfall = ee.ImageCollection('UCSB-CHIRPS/DAILY') .filterDate('2023-07-01', '2023-07-31').sum(); What does each method do and why is this sequence important?

Expected Explanation Format:
Line 1: Creates an ImageCollection object from the CHIRPS dataset...
Line 2: Filters the collection to only images from July 2023...
Line 3: Sums all daily rainfall values to get total monthly rainfall...
Line 4: Clips the result to Uganda's boundaries...

5 Modify and Extend Code

Request Modifications

```
"Can you modify the previous code to: 1. Change the time period to August 2023 2. Add an export function to save as geotiff 3. Include a print statement showing the date range"
```

Add Visualization

```
"Add a histogram chart showing the distribution of rainfall values across Uganda for this time period"
```

Debugging with AI

Debugging Process

TypeError: map.addlayer is not a function

Problem: Case sensitivity in JavaScript

```
// Wrong: map.addlayer // Correct: Map.addLayer
```

Collection.first: Collection is empty

Problem: No data found for specified filters

```
// Check date ranges and geographic bounds
print(collection.size()); // Debug collection size
```

Error Interpretation & Troubleshooting

Common GEE Errors

TypeError: map.addlayer is not a function

Problem: Case sensitivity in JavaScript

```
// Wrong: map.addlayer // Correct: Map.addLayer
```

Collection.first: Collection is empty

Problem: No data found for specified filters

```
// Check date ranges and geographic bounds
print(collection.size()); // Debug collection size
```

Systematic Debugging Approach

AI-Assisted Debugging Workflow

- Copy Error Message
- Include the exact error text and relevant code snippet

Provide Context

Explain what you're trying to accomplish

Request Explanation

Ask for both the fix and an explanation of why it works

Clear and Specific Prompt Construction

Providing Context and Requirements

Iterative Prompt Refinement

Example Prompt Templates

Skills Acquired

Ethical Considerations in AI-Assisted Programming

Career Applications & Next Steps

Building Advanced Skills

</