

# Developing a Minimal Viable Product for your Final Project.

## Overview

### What is a Minimum Viable Product?

A Minimum Viable Product (MVP) is a lightweight early version of a project. It is literally the simplest thing you can create that demonstrates an idea could work and could solve the problem. For many Geocomputing projects, that could simply be reading a portion of your data (a small study area), applying the most basic analytical method, plotting one chart, and closing the dataset. To better understand an MVP, visit [MVP Explainer](#).

### Understanding your project

Think about ALL of your project's features. Recall your earlier MVP for a GIS. A GIS such as Esri ArcPro or QGIS is full-featured. Your MVP was likely not full-featured, but it was a great example of an MVP. You struggled. You thought about the core features of an MVP. Now, you will apply your newfound knowledge and skills to an MVP of your own in preparation for your final project.

## Specifications

You get to create your own specifications!

Brainstorm the features/capabilities your solution will need. Rate each feature/capability on its value (how useful is it?) and effort (how much time to create it?). List key features and their rated value and effort as High, Medium, Low (H, M, L) for your spec.

Review and refine the spec list. **Cross off features that are too high effort for too low value until you have a list of features that can be implemented in about 1 week. That is your MVP!**

The notebook template has a place where you can add your MVP specifications.

*Hint:* It will likely only be 1-3 items for this reduced MVP.

## Requirements

### Minimum Viable Product: MVP 3

1. Download the **MVP3 Notebook** file from Canvas. Edit it using Jupyter. You will notice this MVP has some existing code to help structure your submission and give you a starting point. We assume you will continue editing this notebook for your final project.
2. **Add Project Title, Description, and Name information:** Edit the markdown cells to include a title and description of your project as well as other requested information.
3. **Structure your code.** Follow principles and best practices. In general, your import statements should be at the top. Followed by function and/or class definitions. Finally, the code that runs everything should be toward the bottom. There are spots here to provide guidance.  
*IMPORTANT NOTE:* You can add, remove, or modify these cells. It is just a suggestion and starting point for those who would like to have some structure.
4. **Rationale:** Write one paragraph that explains why this notebook represents an MVP.
5. Run all cells in your notebook. Export it as a PDF. Save it as a .ipynb file.

Submit the MVP Jupyter Notebook as a .ipynb file and the .pdf file into Canvas.

## Grading

**IMPORTANT NOTE:** This MVP has a change of grading practices. We want you to start your MVP and start working toward your final project. So we are keeping grading clear and simple.

**LOOK AT THE MODIFIED GRADING RUBRIC ON THE NEXT PAGE.**

**Tl;dr:** To achieve the full 5 points for MVP 3. You need to submit a "Quality" Notebook (neat and good presentation) with Markdown questions answered and 75+ lines of Python code. This is the minimum requirement to achieve



## Grading Rubric for MVP 3

The following rubric will be used to grade this MVP exercise. Minimum Viable Products (MVPs) are graded based on two categories: Technique/Concept and Quality.

**5 points** will be awarded if the MVP is graded at "Proficient" level for "Technique & Concept" and the "Competent" level for "Quality." MVP exercises are graded holistically for each category.

### Technique and Concept

	Missing	Unsatisfactory	Competent	Proficient
<b>Technique &amp; Concept</b>	No work.  No questions answered or code.	Work lacks understanding of concepts, materials and skills.  <input checked="" type="checkbox"/> Markdown questions answered <input type="checkbox"/> 25+ lines of Python code. <input type="checkbox"/> 75+ lines of Python code.	Work shows some understanding of skills, concepts, and materials.  <input checked="" type="checkbox"/> Markdown questions answered <input checked="" type="checkbox"/> 25+ lines of Python code. <input type="checkbox"/> 75+ lines of Python code.	Work reflects understanding of skills, concepts, and materials.  <input checked="" type="checkbox"/> Markdown questions answered <input checked="" type="checkbox"/> 25+ lines of Python code. <input checked="" type="checkbox"/> 75+ lines of Python code.
<b>Points</b>	0 points	2 point	3 points	4 points

### Quality

	Unsatisfactory	Competent
<b>Quality</b>	Work is messy and lack of quality detracts from overall presentation.	Work is neat and presentation is solid showing thoughtfulness.
<b>Points</b>	0 points	1 point