This is a sample of property insurance premiums (sales income) from a major US company.

The project has to be done using these tools:

* OS: Ubuntu
* Server Side Scripting: Python 2.7+ or 3.5+ and Pandas for the API and report
* Server Framework: Flask or SimpleHTTPServer

**Minimum Requirements**

* Build a Data Pipeline/ETL process that takes the CSVs as input and saves into a database at a detailed level while also calculating summarized views. These summarized views could follow star schema or any other that you think will allow for easy querying using different pivots/dimensions. The Data Pipeline can be manually triggered by running a script (include instructions of how to do it!) or automated somehow.
* Build an API (REST, Graph or other design pattern) that provides:
  + Detailed information using different parameters (like agency, month, year, state, etc)
  + Summarized information using different parameters (like agency, month, year, state, etc)
  + An XLS, XLSX or CSV report with Premium info by Agency and Product Line using date range as parameters
* The Data Pipeline/ETL process and also the logic for generating the report must be done using Pandas
* Deployment to AWS

Additionally we will use other technologies :

* Tableau : I will build a dashboard and host it on tableausoftware.com
* Host mysql on AWS RDS and put the mysql db there
* AWS S3 will host the csv file, then python lambda (serverless function) will import it into mysql, then AWS QS will pull sql queries directly to show visualizations
* Integration or Unit tests (at least one of those). You can use pytest or unittest
* Documented code that follows pep8 and The Zen of Python
* API documentation

We need to do following subtasks in scope of the project

1. Start by developing on a local VM

1. Develop pipeline process that will just simply load data from csv to MySQL

2. Extend pipeline process to generate summarized views

3. Start-up simple flask REST API with just simply returning detailed information

4. Extend REST API to return summarized information

5. Extend REST API to generate reports (XLS, XLSX, CSV)

6. Extends Flask Application to return a report page to the browser using Bokeh library

I would say, we can do it iteratively. I will implement one or two of them and we will deploy, then working on next one or two tasks and deploy again.

But I will do on my local

Once I have working let say first task we will deploy it and test it on the local VM

2. Set up the AWS infrastructure and deploy the database and python onto AWS and run it in the cloud.

Make a python panda visualization from this sample

<https://demo.bokehplots.com/apps/movies>

and a state map starting out with this visual

<https://bokeh.pydata.org/en/latest/docs/gallery/texas.html>

but replacing the texas codes with US codes which I already have Py codes for.