Python and Julia Interface for Data Retriever

Abstract

Data Retriever automates the tasks of finding, downloading, and cleaning up publicly available data, and then stores them in a local database or as .csv files. Simply put, it's a package manager for data. This allows data analysts to spend a majority of their time in analysing rather than in cleaning up or managing data.

The Data Retriever is written in Python. It is equipped with a command line interface (CLI) and can also be used through an associated R package that wraps this CLI.

Describe the Need of the Project

Python is presently one of the most popular choices for data analysis and hosts many well-known data analysis modules like Pandas, NumPy, and SciPy. Current Data Retriever lacks Python support and for adding Python support for Data Retriever, creating a Python interface is necessary. A Python package will be easy to install and will provide large amounts of ecological data collections. This would offer a data analyst the ability to directly derive meaning out of data without any hassle. As the Data Retriever has a Python codebase, creating a Python interface would complement the code intelligibility.

To improve the outreach to the community and user base, another well-known language, Julia, would be a genuine choice for implementing Data Retriever features. R already has a package wrapper for the Data Retriever in place. Adding a Julia package wrapper to the CLI would facilitate Julia users to benefit from the data retriever features.

Technical Details

Implementing a python interface for the Data Retriever framework will start with restructuring the design. Currently Data Retriever only supports CLI for python. So

there are a lot of extra imports that have to be taken care of. The list of all imports that are currently visible on import of retriever are shown.
41 Retriever imports are:

CITATION	REPO_URL	getmtime	platform
COPYRIGHT	SCRIPT_LIST	imp	print_function
DATA_DIR	SCRIPT_SEARCH_PATH	io	pwd
DATA_SEARCH_PATHS	SCRIPT_WRITE_PATH	isfile	sample_script
DATA_WRITE_PATH	VERSION	join	set_proxy
ENCODING	absolute_import	lib	str
ENGINE_LIST	compile_json	open_csvw	sys
HOME_DIR	csv	open_fr	to_str
MASTER_BRANCH	current_platform	open_fw	
MODULE_LIST	dir	os	
REPOSITORY	exists	parse_version	

In the first phase. Restructuring of design will be done. This will be done in three sections. In the first section all the constants that have been defined in the imports will be skimmed. The second section will reduce the redundancy of imports and will check for direct library imports. In the third section important functions will be kept and consistency checks will be performed.
In the second phase I will be working on building functions for the package. Functions like *fetch, install, json* functions will be implemented. The *fetch* function will allow the user to download raw data available. All of these function will be using existing Data Retriever functions. Added functionality parameters for functions will also be implemented.
In the third phase I will be working on building the Julia package wrapper for Data Retriever. The Julia wrapper would include all functions. The Julia package wrapper will be integrated with native language support having all the functionalities of Data Retriever.

Schedule of Deliverables

May 1st - May 28th, Community Bonding Period

- Get accustomed to the general community environment and coding practices.
- Try to get all community members and users personal insight about the project idea and its implementation.
- Put more thought into the implementation details of my proposal and keep improving it.
- Start a blog on Data Retriever about the project to help my mentor track my progress.
- Getting a head start and polishing Data Retriever by working on design issues.

May 29th - June 3rd

Restructure Design:

- Changing files structure of retriever.
- There are currently 41 functions that can be accessed on importing retriever. Including some which are not required in the interface. The focus will be on removing such imports.
- Start by targeting the Constants, 14 imports. Matching them against their usage among the scripts and removing their imports from retriever.

June 5th - June 9th

Restructure Design:

- Compete the placement of constants in appropriate positions.
- Keeping imports for citation, ls, defaults.
- Performing Unit and Integration testing.

June 12th - June 16th

Restructure Design:

- Removing direct import of libraries io, csv, sys, str, pwd, platform. So that they don't show up in retriever import.

- Checking the import of sub-functions of libraries and making sure they do not show up in retriever import.

June 19th - June 23th, End of Phase 1

Testing New Design:

- Making sure if all the tasks mentioned in the previous weeks have been performed to completion.

- Completing the work of removing redundant and unnecessary import from retriever.

- Performing Unit and Integration testing for the complete package.

June 26th - June 30th, Begin of Phase 2

In this phase I will be making the python interface. This will be the most important week with respect to the implementation perspective. The equivalent functions of the CLI functions like download, install will be made.

Building Interface:

- Work on *fetch* function. The function will download the raw data files.

- Constructing parameters for specifying the location to save the downloaded data files.

- Test this new function against all the scripts in data retriever code base.

July 3rd - July 7th

Building Interface:

- Work on *install* function. The function will install data files in required engine (*mysql, postgres, sqlite, MS access)*.

- Adding options for downloading data files and installing them into respective engine.

- Testing the new *install* functionality against all the scripts in data retriever code base.

July 10th - July 14th

Building Interface:

- Adding support for *update* function. The update function will update all the scripts in the current Data Retriever codebase.

- The *update* function will provide the user with options of downloading the functions again and running an update or only updating the scripts which have become outdated.

July 17th - July 21st, End of Phase 2

Buffer Period:

- Completing pending work if any. Checking documentation for all the added functions. Adding documentation if missing.

- Complete testing of the data retriever module.

July 24th - July 28th, Begin of Phase 3

In this phase the python interface will be completed by the end of first week. In the second week I will start off with the Julia interface. Which will take another 2 weeks to finish before `the final evaluation.

Completing Interface:

- Work on adding *json* functionality as in CLI.

- Working with each script and meeting the requirements necessary. Adding documentation for all the added functions.

- Testing the added functions. Unit and Integration testing of the complete package.

July 31st - August 4th

Adding Julia Interface:

- Work on adding wrapper functions for *download, fetch, update* functionality.

- Testing and adding documentation for the added functions.

August 7th - August 11th

Adding Julia Interface:

- Work on adding wrapper functions for reset, citations and other newly added functions during the timeline.

- Testing and adding documentation for the functions.

August 14th - August 18th

- Creating man page for Julia wrapper.

- Adding DESCRIPTION, NAMESPACE, LICENSE and README for the wrapper.

- Performing thorough testing of the Julia wrapper.

August 21st - August 25th, Final Week

- Solving issues that have come up after the finalizing work for respective phases.

- Checking Documentation for any errors and mismatches.

- Submit sample code to Mentors.

August 28th - August 29th, Submit final work

- Mentors review student code samples and determine if the students have successfully completed their Google Summer of Code 2017 project.

Future works

Achieving Milestone 2.0 set by Data Retriever as soon as possible. Put effort in solving any issues related to the package created under this project. Keep building stuff and working in Data Retriever.

Development Experience

Developed multi platform application to generate infographic representation of crime rates in India. I have extensive experience in working with database management software's like SQL, Raptor, Mysql, Mongo DB, and Oracle. Follow the link for project details: *"Topological Crime Analysis in India*" .

Developed 3D models for Virtual Reality simulations while working under a mentor for the Italian Mars Society. Used Blender graphic software for creating 3D models of space station.

Internship at FOSSEE India open source work was a purely Python oriented internship. I had designed python code snippets and detailed exercises for the book *Advanced Engineering Chemistry*. This content is currently available and being used by student community. This required quite a lot of commitment, improved my dexterity, and helped me hone the skills required for being in a Python community.

I have submitted multiple contributions using Python in FOSSEE India and Mozilla Foundation. In Mozilla Bugzilla I solved a good first bug and strengthened my Python skills in the process.

On Retriever, I have worked on a wide variety of topics. Be it adding core functionalities, adding dataset support, adding test for scripts, or just finding bugs in script files. All of these are mentioned here.

As you can see, I have developed the required Git skills during the process and now am comfortable with using Git on a daily basis.

Link to my repository of competitive programming codes written in Python: here.

Other Experiences

Teamwork

I believe that I have a good aggregate of experience in working as a part of a team. Sports also add to the diversity of this experience as I have been my school sports captain and football team captain for a good 2 years. During my time as the sports captain, I had organised numerous sports events successfully. I have worked as a team in organizing college tech events as well. I am also the member of my college's official Computer Science Club IECSE, under which, numerous events and workshops have been organized by us.

I have also worked under a mentor in the Italian Mars Society and have built 3D models for Virtual Reality (VR) simulations. A team of 4 worked together for 3 months in the summer and for a month during winter. All the work was well coordinated and efficiently performed. We also implemented the AGILE model for software development while working on the project.

Context

I'd be elated on the acceptance of this proposal, as it would allow me to make more advanced contributions to the open source community. To me, the idea of contributing to the open source and helping the non-technical people through coding and my problem-solving skills is both an exciting and a fulfilling experience. I enjoy working on Python modules, and I would love to share my resources with those in need.

Why this project?

In the field of Computer Science, Data Analysis has always been of foremost interest to me. Having worked towards mining data in the past, the concept of Data Retriever intrigued me instinctively and has driven me towards making data analysis easier for future analysts. The special thing about the Data Retriever is that it's a very welcoming and encouraging community.

Personal Details

Name: Shivam Negi

Affiliation (school/degree): Manipal Institute of Technology, Manipal - 6th semester, Computer Science & Engineering

Location (where you are): Manipal, Karnataka, India.

Email: shivamnegi2019@gmail.com

Phone: +91-9663594592

Github Id: ShivamNegi

Project(s) you're working on or want to: Data Retriever Org

Project title: Python and Julia interfaces for the Data Retriever