The Planetary Data Reader (*pdr*): a Python toolkit for reading planetary data. C. C. Million[1], M. St. Clair[2], K. Michael Aye[3], and Jordan Padams[4]; [1]Million Concepts (chase@millionconcepts.com), [2]Million Concepts, [3]University of Colorado Boulder, [4]Jet Propulsion Laboratory, California Institute of Technology.

Introduction: We are developing the Planetary Data Reader (*pdr*), an open-source, Python-based tool for the ingestion of planetary observational data into planetary science workflows. This project seeks to solve a well-known pain point for planetary data users: simply figuring out how to read the data. This is a particular problem for data archived under the Version 3 standards of the Planetary Data System (PDS); the PDS is in the process of migrating archives to the stricter PDS4 standard, but the timeline is uncertain. *pdr* provides a stopgap solution and future-proofs research workflows while also supplementing the migration effort.

The software is currently available in an alpha stage with functional support for a large fraction of PDS3-compliant image and table data, as well as all PDS4-compliant data. We are actively using it on a number of current and recently completed projects and are ready to accept more users and community feedback. [1]

Basic Design and Functionality: The *pdr* tool is being designed to have almost no learning curve for users with basic Python proficiency. You must simply import the module and then pass an observational *or* metadata file path to a `pdr.read()` function. *pdr* returns an object with *both* the metadata *and* observational data as attributes in standard Python types (e.g. list, numpy.ndarray, pandas.DataFrame) or as *pvl* objects [2]. The structure of the object is the same regardless of whether an observational data or metadata file is passed, or what standard the data were archived under. The user will not be required to specify the data format type; the software infers this.

Direction: Over the next several years --- with support from the NASA PDART program --- we will build out *pdr* with functionality and a thorough test suite to cover almost all data (including tables and images) and metadata formats currently archived by the PDS, as well as some formats common in planetary science but not typically archived in the PDS (e.g. ISIS cube, JP2, GeoTiff).

References:

[1] https://github.com/millionconcepts/pdr
[2] https://github.com/planetarypy/pvl

```
In [0]: import pdr

In [1]: datafile = 'thispath/file.IMG'
In [2]: datalbl = 'thispath/file.LBL'

In [3]: data = pdr.read(datafile)

In [4]: data.keys()
Out[4]:  ['LABEL','HEADER','IMAGE']
In [5]: data.IMAGE.shape
Out[5]: [400, 300]
In [6]: data.IMAGE.mean()
Out[6]: 728.6

In [7]: data = pdr.read(lblfile)

In [8]: data.keys()
Out[8]: ['LABEL','HEADER','IMAGE']
In [9]: data.IMAGE.mean()
Out[9]: 728.6

In [10]: data.LABEL['TARGET']
Out[10]: 'MARS'
```

```
In [0]: import pdr

In [1]: datafile = 'thispath/file2.csv'
In [2]: datalbl = 'thispath/file2.xml'

In [3]: data = pdr.read(datafile)

In [4]: data.keys()
Out[4]: ['LABEL','TABLE']
In [5]: data.TABLE.keys()
Out[5]: ['Time','Temp','Temp_err']

In [6]: data.TABLE['Temp'].min()
Out[6]: 23.4
```

Figure. These are mock-up workflows for reading data and deriving simple information about those data. The top frame gives an example for reading an image file archived under PDS3. The bottom frame is for a table file in PDS4. This very closely represents the behavior of the current alpha-stage version of *pdr*.