Data Versioner

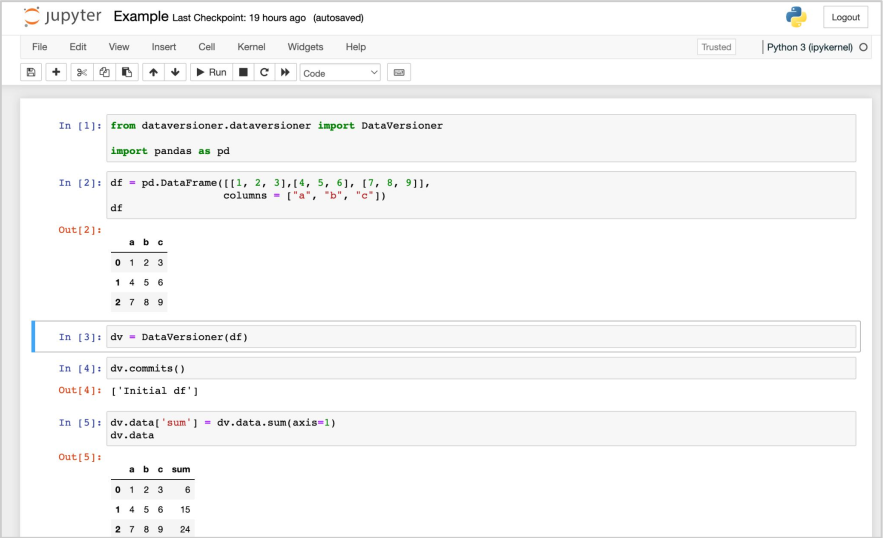
Castle Leonard
March 8, 2022
DATA 515 - Project Presentation

Background & Use Cases

Problem: How can we avoid interrupting and stemming our creativity in experimental and exploratory phases of data science without losing track of variations and findings? Especially in the wild west that is Jupyter notebooks?

Solution: Data versioner is a python package that offers git-like tracking of your pandas dataframe so that your experimental variations and EDA findings are preserved, even as your notebook evolves.

Demo



```
In [1]: from dataversioner.dataversioner import DataVersioner

        import pandas as pd
```

```
In [2]: df = pd.DataFrame([[1, 2, 3],[4, 5, 6], [7, 8, 9]],
                          columns = ["a", "b", "c"])
        df
```

```
Out[2]:
           a  b  c
        0  1  2  3
        1  4  5  6
        2  7  8  9
```

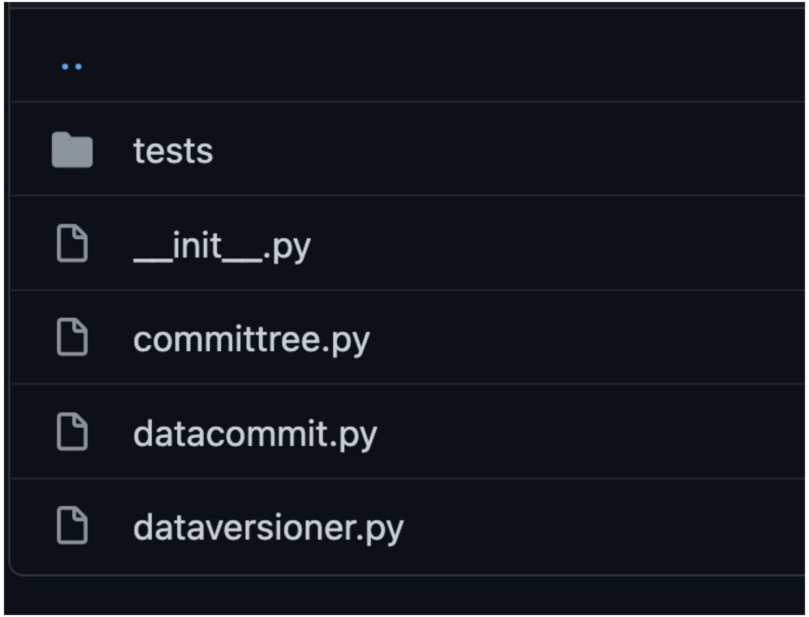
```
In [3]: dv = DataVersioner(df)
```

```
In [4]: dv.commits()
Out[4]: ['Initial df']
```

```
In [5]: dv.data['sum'] = dv.data.sum(axis=1)
        dv.data
```

```
Out[5]:
           a  b  c  sum
        0  1  2  3    6
        1  4  5  6   15
        2  7  8  9   24
```

Design & Project Structure



```
..
tests
__init__.py
committree.py
datacommit.py
dataversioner.py
```

Software Engineering Lessons & Future Work

Lessons

1. **Trade-offs everywhere -** In many choices I encountered trade offs between user experience, code complexity, development time, and constraints to the future of the package.
2. **Start with MVP -** It is so easy to get swept up with all the aspirations and anticipations of needs for sophisticated functionalities, but premature optimization

Future

1. **Extensible multi-inheritance -** My dream implementation of this package is a wrapper class of the initializing dataframe type. This would be much less obtrusive during data manipulation (data_versioner.data is no longer necessary) and would enable the package to be extended relatively easily to provide seamless user experience for all supported dataframe types.
2. **Automatic dataframe modification logging and diffing**