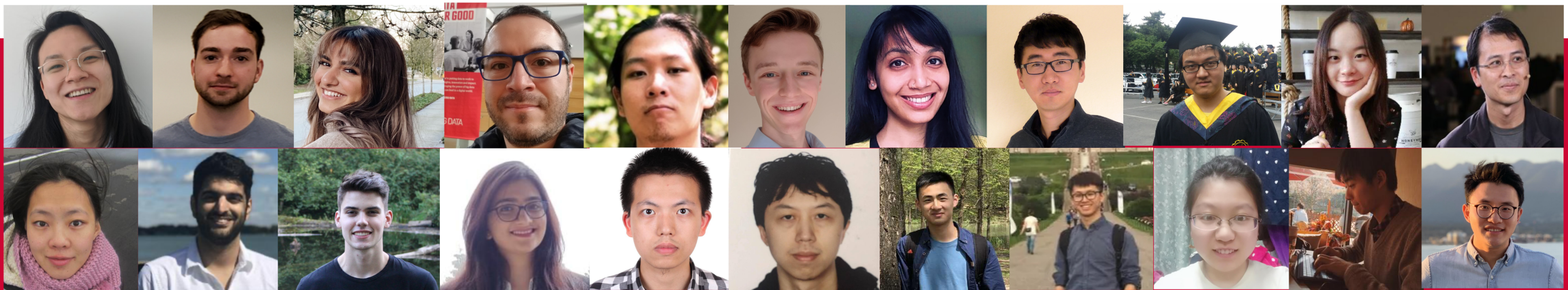
DataPrep - The easiest way to prepare data in Python

Jiannan Wang

Simon Fraser University

Jan 6, 2021, Databricks

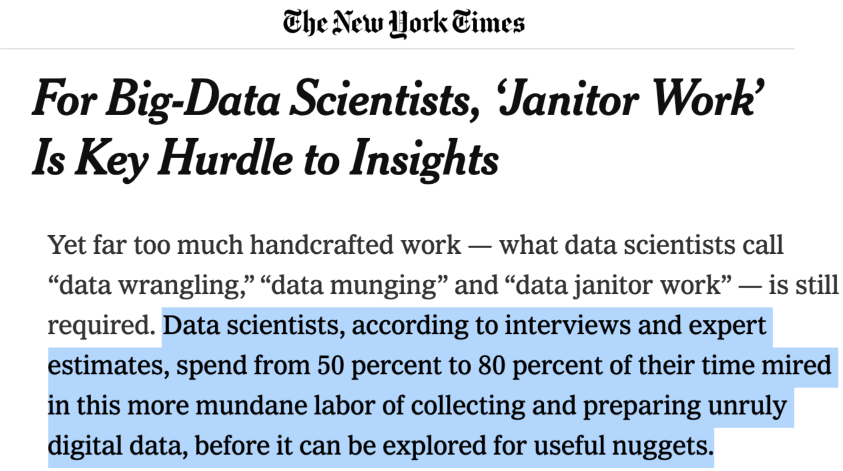


Talk Outline

1. DataPrep Overview
2. Dive into DataPrep
    - DataPrep.EDA
    - DataPrep.Connector
3. Future Direction

Data Preparation Is Still the Bottleneck!!!

2014



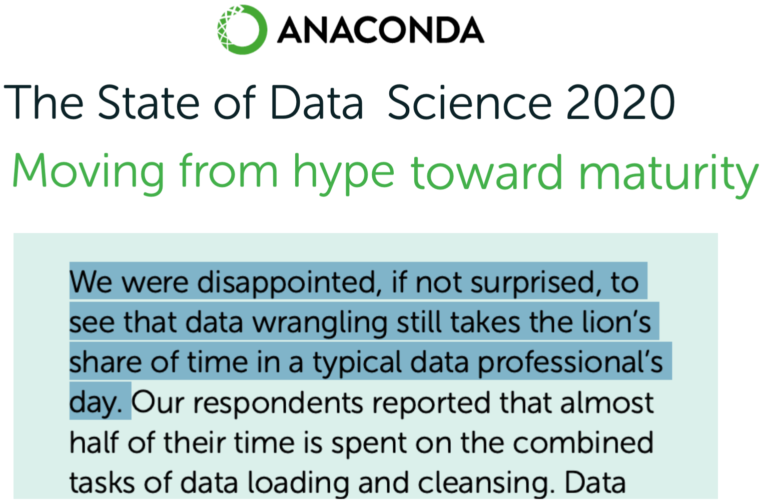
The New York Times

For Big-Data Scientists, 'Janitor Work' Is Key Hurdle to Insights

Yet far too much handcrafted work — what data scientists call "data wrangling," "data munging" and "data janitor work" — is still required. Data scientists, according to interviews and expert estimates, spend from 50 percent to 80 percent of their time mired in this more mundane labor of collecting and preparing unruly digital data, before it can be explored for useful nuggets.

https://www.nytimes.com/2014/08/18/technology/for-big-data-scientists-hurdle-to-insights-is-janitor-work.html

2020



ANACONDA

The State of Data Science 2020

Moving from hype toward maturity

We were disappointed, if not surprised, to see that data wrangling still takes the lion's share of time in a typical data professional's day. Our respondents reported that almost half of their time is spent on the combined tasks of data loading and cleansing. Data

https://www.anaconda.com/state-of-data-science-2020

Why Is Data Preparation Hard?

Collection

Cleaning

Integration

Analysis

How much time is spent on preparation?

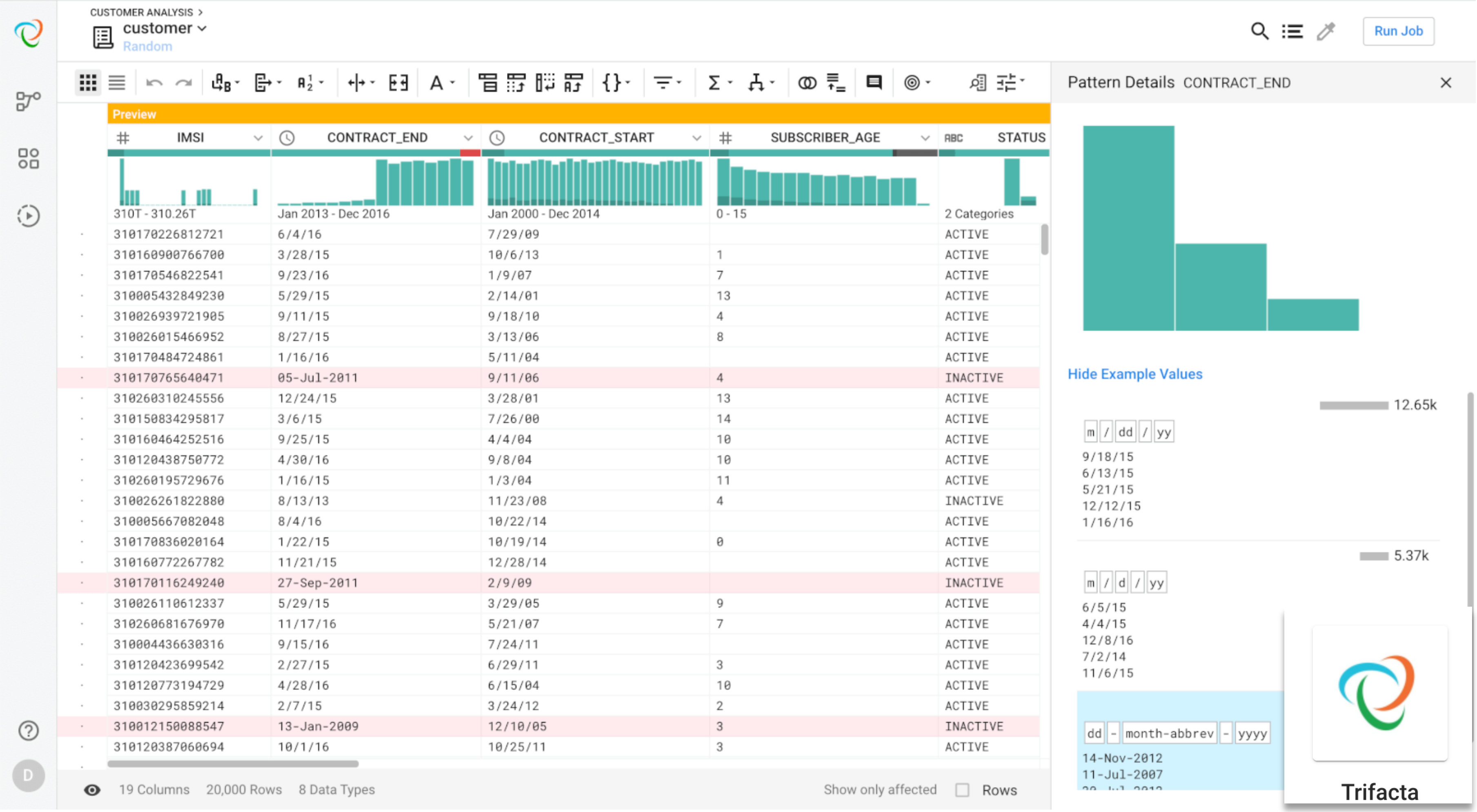
1. **Too many small problems** (e.g., standardize date, dedup address, etc)
2. Humans have **different levels of expertise** (in data science and programming)
3. **Domain specific** (finance, social science, healthcare, economics, etc.)

Human-in-the-loop Data Preparation

Three Directions

- Spreadsheet GUI
- Workflow GUI
- Notebook GUI

Spreadsheet GUI



IMSI	CONTRACT_END	CONTRACT_START	SUBSCRIBER_AGE	STATUS
310T - 310.26T	Jan 2013 - Dec 2016	Jan 2000 - Dec 2014	0 - 15	2 Categories
310170226812721	6/4/16	7/29/09		ACTIVE
310160900766700	3/28/15	10/6/13	1	ACTIVE
310170546822541	9/23/16	1/9/07	7	ACTIVE
310005432849230	5/29/15	2/14/01	13	ACTIVE
310026939721905	9/11/15	9/18/10	4	ACTIVE
310026015466952	8/27/15	3/13/06	8	ACTIVE
310170484724861	1/16/16	5/11/04		ACTIVE
310170765640471	05-Jul-2011	9/11/06	4	INACTIVE
310260310245556	12/24/15	3/28/01	13	ACTIVE
310150834295817	3/6/15	7/26/00	14	ACTIVE
310160464252516	9/25/15	4/4/04	10	ACTIVE
310120438750772	4/30/16	9/8/04	10	ACTIVE
310260195729676	1/16/15	1/3/04	11	ACTIVE
310026261822880	8/13/13	11/23/08	4	INACTIVE
310005667082048	8/4/16	10/22/14		ACTIVE
310170836020164	1/22/15	10/19/14	0	ACTIVE
310160772267782	11/21/15	12/28/14		ACTIVE
310170116249240	27-Sep-2011	2/9/09		INACTIVE
310026110612337	5/29/15	3/29/05	9	ACTIVE
310260681676970	11/17/16	5/21/07	7	ACTIVE
310004436630316	9/15/16	7/24/11		ACTIVE
310120423699542	2/27/15	6/29/11	3	ACTIVE
310120773194729	4/28/16	6/15/04	10	ACTIVE
310030295859214	2/7/15	3/24/12	2	ACTIVE
310012150088547	13-Jan-2009	12/10/05	3	INACTIVE
310120387060694	10/1/16	10/25/11	3	ACTIVE

19 Columns 20,000 Rows 8 Data Types

Show only affected □ Rows

m / dd / yy

9/18/15
6/13/15
5/21/15
12/12/15
1/16/16

m / d / yy

6/5/15
4/4/15
12/8/16
7/2/14
11/6/15

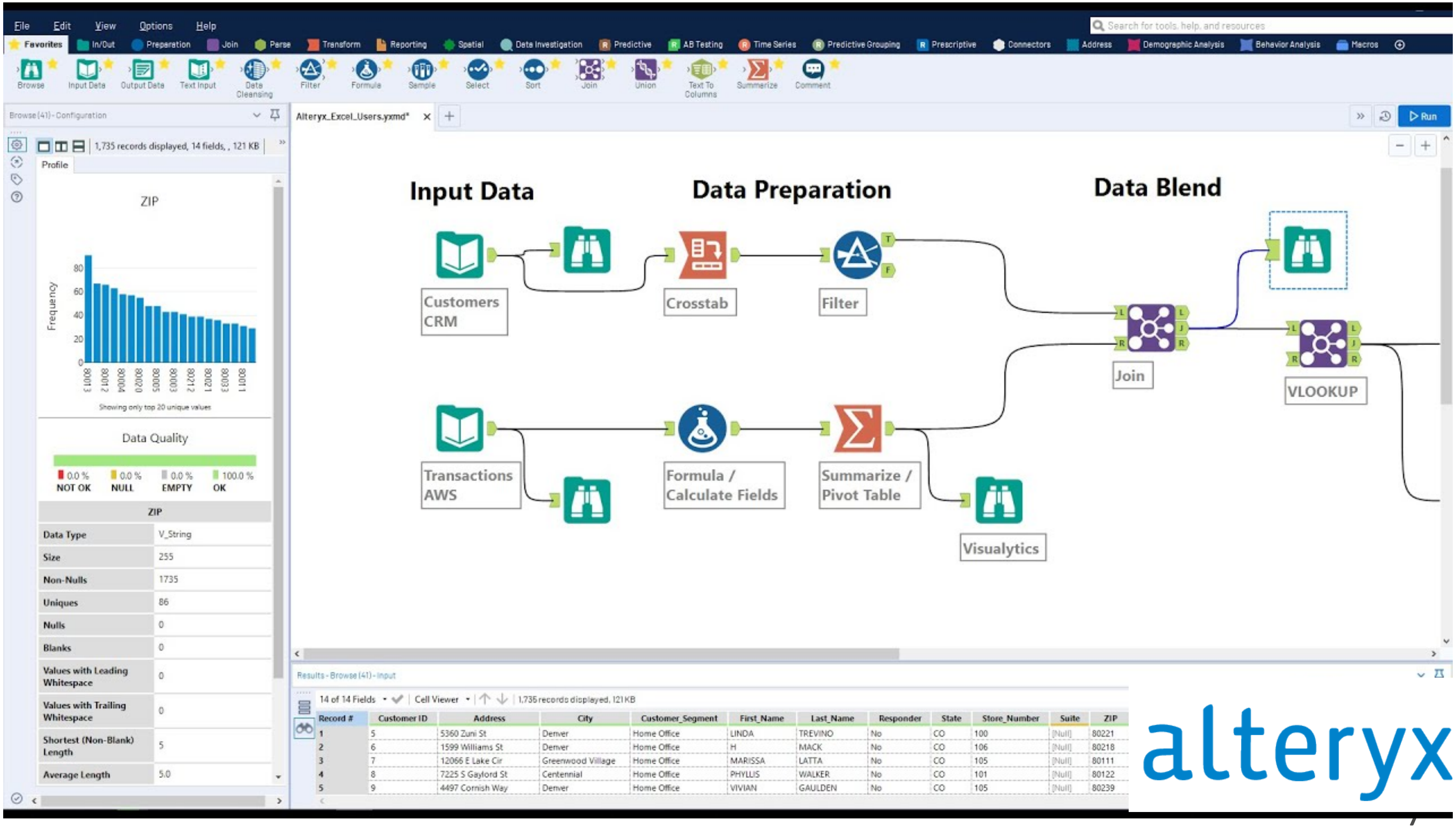
dd - month-abbrev - yyyy

14-Nov-2012
11-Jul-2007

Trifacta

SFU

Workflow GUI



1,735 records displayed, 14 fields, , 121 KB

0.0 %	0.0 %	0.0 %	100.0 %
NOT OK	NULL	EMPTY	OK

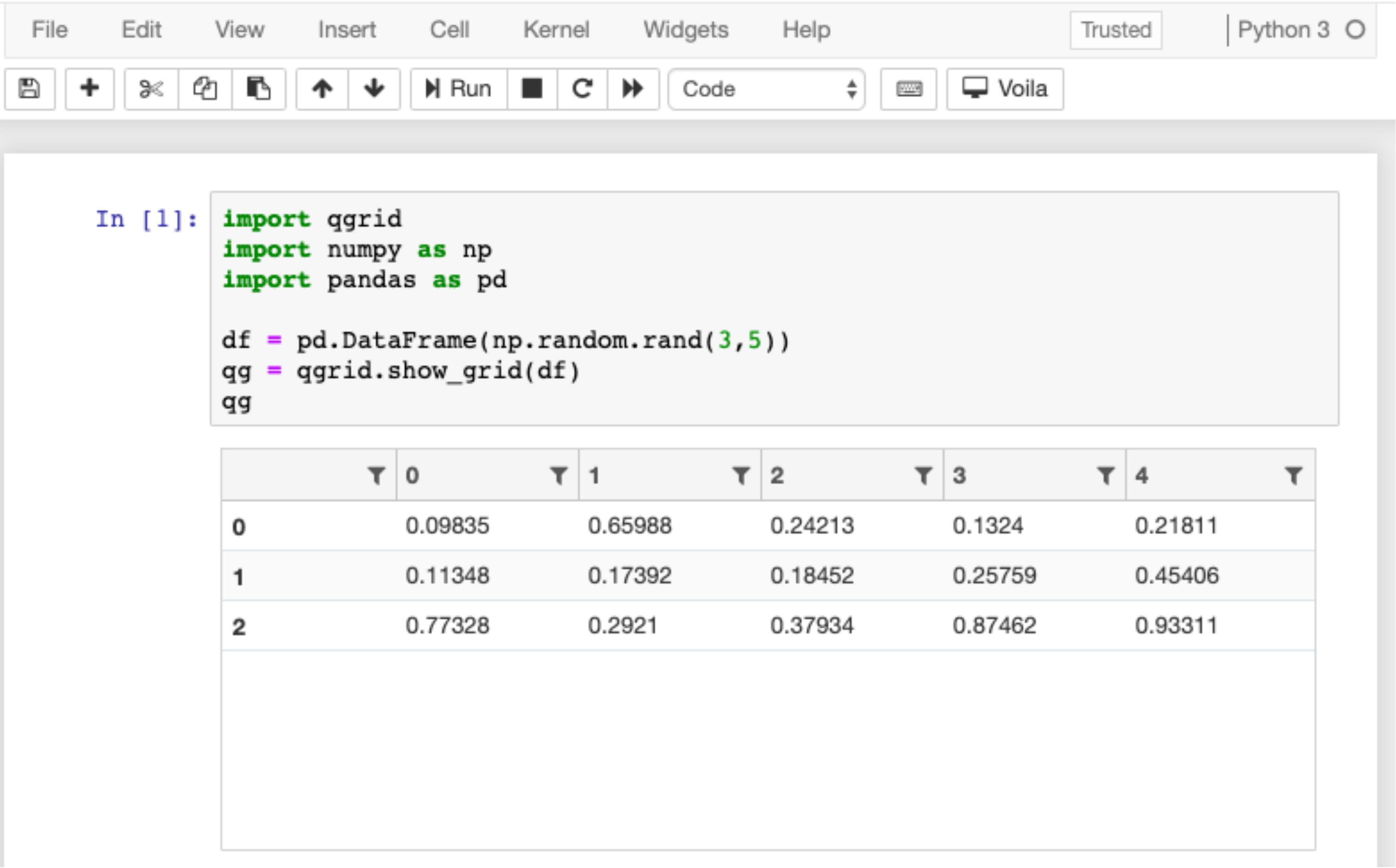
ZIP	
Data Type	V_String
Size	255
Non-Nulls	1735
Uniques	86
Nulls	0
Blanks	0
Values with Leading Whitespace	0
Values with Trailing Whitespace	0
Shortest (Non-Blank) Length	5
Average Length	5.0

1,735 records displayed, 121 KB

Record #	Customer ID	Address	City	Customer_Segment	First_Name	Last_Name	Responder	State	Store_Number	Suite	ZIP
1	5	5360 Zuni St	Denver	Home Office	LINDA	TREVINO	No	CO	100	[Null]	80221
2	6	1599 Williams St	Denver	Home Office	H	MACK	No	CO	106	[Null]	80218
3	7	12066 E Lake Cir	Greenwood Village	Home Office	MARISSA	LATTA	No	CO	105	[Null]	80111
4	8	7225 S Gaylord St	Centennial	Home Office	PHYLLIS	WALKER	No	CO	101	[Null]	80122
5	9	4497 Cornish Way	Denver	Home Office	VIVIAN	GAULDEN	No	CO	105	[Null]	80239

alteryx

Notebook GUI



In [1]:
```
import qgrid
import numpy as np
import pandas as pd

df = pd.DataFrame(np.random.rand(3,5))
qg = qgrid.show_grid(df)
qg
```

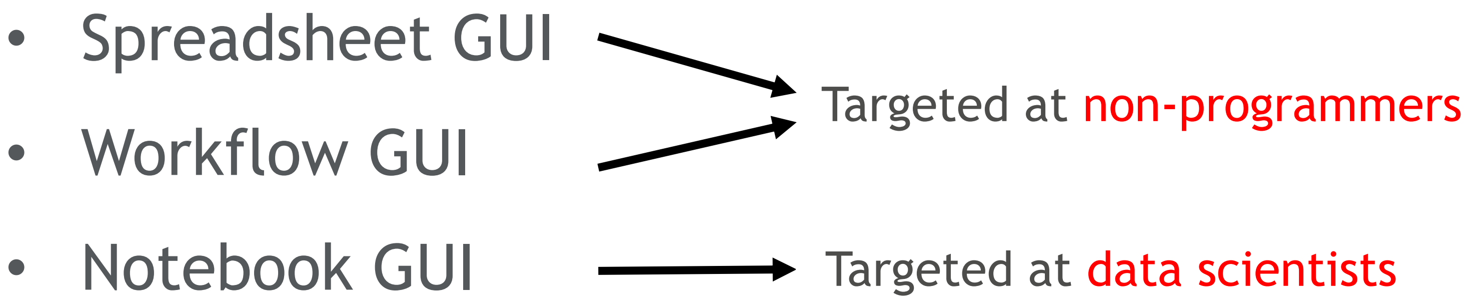
	0	1	2	3	4
0	0.09835	0.65988	0.24213	0.1324	0.21811
1	0.11348	0.17392	0.18452	0.25759	0.45406
2	0.77328	0.2921	0.37934	0.87462	0.93311

Which Direction To Go?

> “Data Prep Market was valued at USD 3.29 Billion in 2019 and is projected to reach USD 18.11 Billion by 2027, growing at a CAGR of 25.64% from 2020 to 2027”

Source: https://www.verifiedmarketresearch.com/product/data-prep-market/

Three Directions



- Spreadsheet GUI
- Workflow GUI
- Notebook GUI

Targeted at non-programmers

Targeted at data scientists

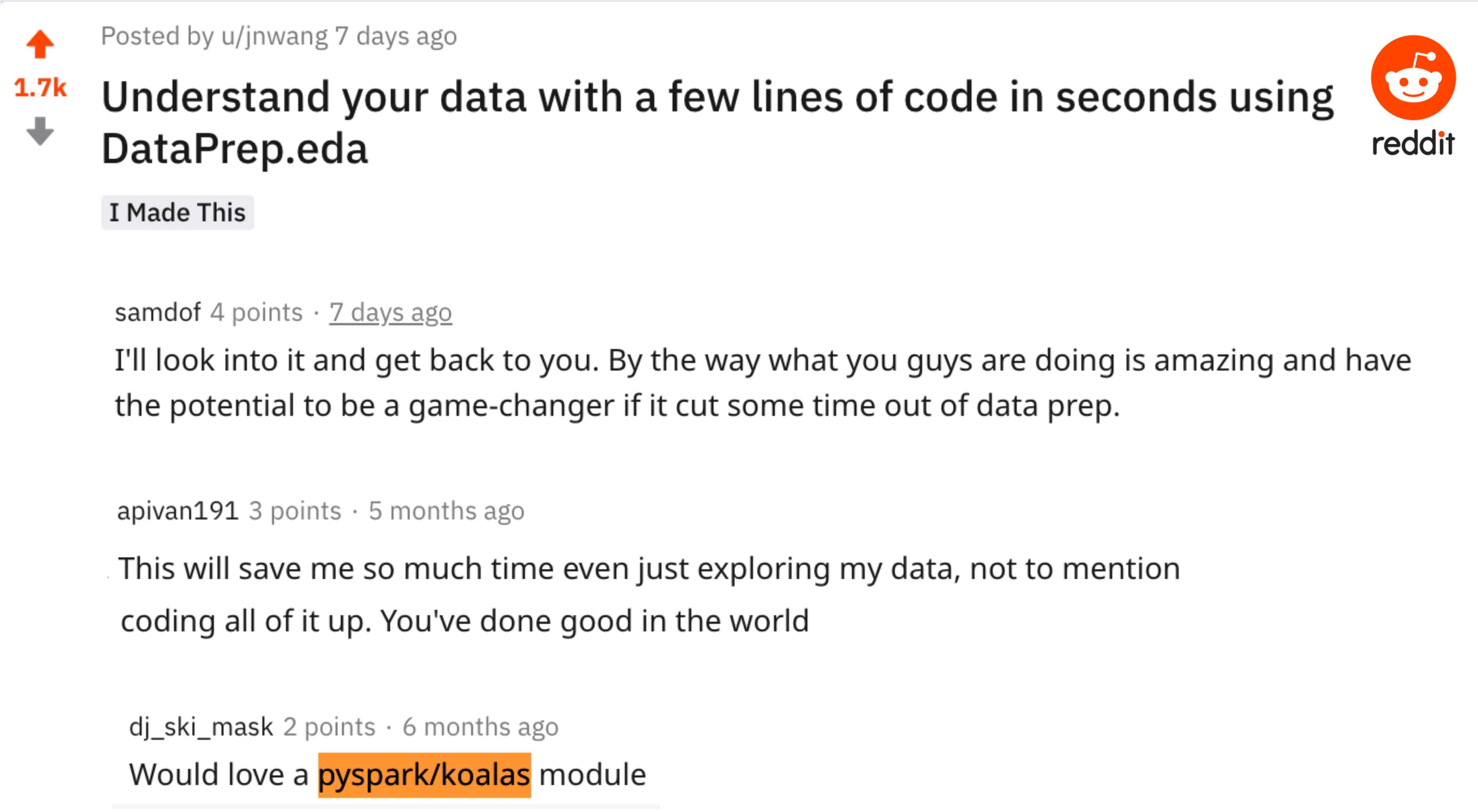
Our Vision

Machine Learning Made Easy	scikit learn
Deep Learning Made Easy	PyTorch
Big Data Made Easy	Koalas
Visualization Made Easy	matplotlib
Data Preparation Made Easy	dataprep

DataPrep Components

May 2019 - Now	DataPrep.EDA	Simplify Exploratory Data Analysis
Nov 2019 - Now	DataPrep.Connector	Simplify Web Data Collection
Sept 2020 - Now	DataPrep.Clean	Simplify Data Cleaning
Planning	DataPrep.Feature	Simplify Feature Engineering
Planning	DataPrep.Integrate	Simplify Data Integration

User Feedback



1.7k

Posted by u/jnwang 7 days ago

Understand your data with a few lines of code in seconds using DataPrep.eda

I Made This

samdof 4 points · 7 days ago

I'll look into it and get back to you. By the way what you guys are doing is amazing and have the potential to be a game-changer if it cut some time out of data prep.

apivan191 3 points · 5 months ago

This will save me so much time even just exploring my data, not to mention coding all of it up. You've done good in the world

dj_ski_mask 2 points · 6 months ago

Would love a pyspark/koalas module

https://www.reddit.com/r/Python/comments/hlqnim/understand_your_data_with_a_few_lines_of_code_in/

Talk Outline

1. DataPrep Overview
2. Dive into DataPrep
   - DataPrep.EDA
   - DataPrep.Connector
3. Future Direction

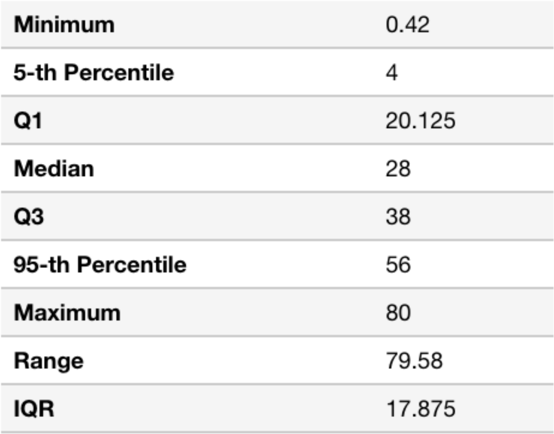
DataPrep.EDA

Task-Centric Exploratory Data Analysis

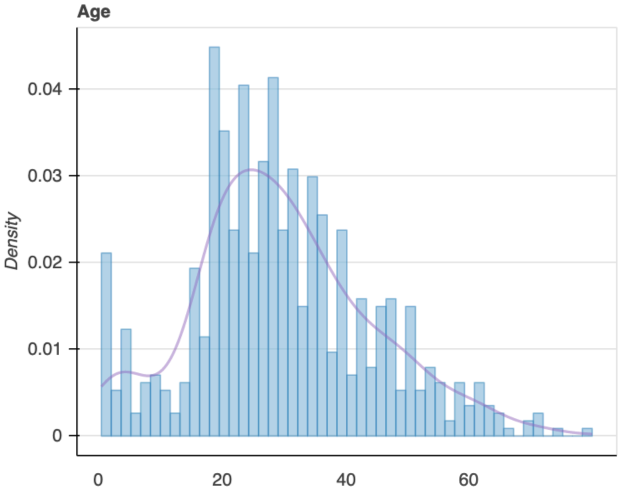
Exploratory Data Analysis (EDA)

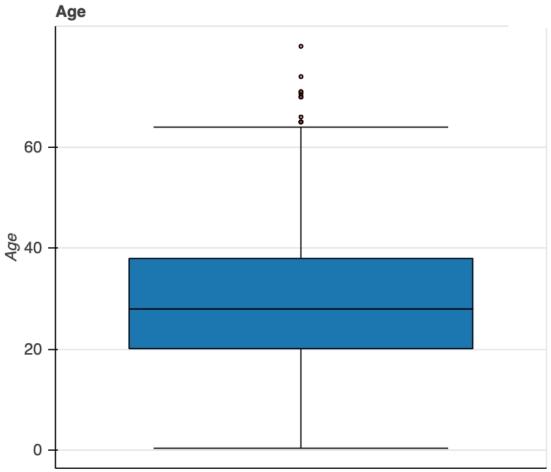
Understand data and discover insights
via data visualization, data summarization, etc.

Understand “Age” column



Minimum	0.42
5-th Percentile	4
Q1	20.125
Median	28
Q3	38
95-th Percentile	56
Maximum	80
Range	79.58
IQR	17.875





Current EDA Solutions in Python

Solution 1: Pandas + Matplotlib

☹ Hard to Use

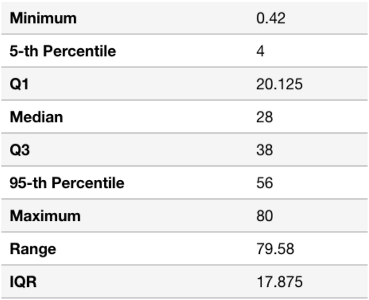
- Beginner: Need to know how to write plotting code
- Expert: Need to write lengthy and repetitive code

Understand “Age” column

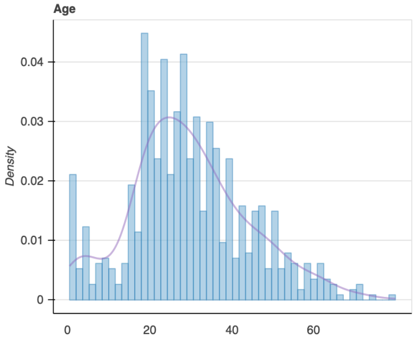
Write Code

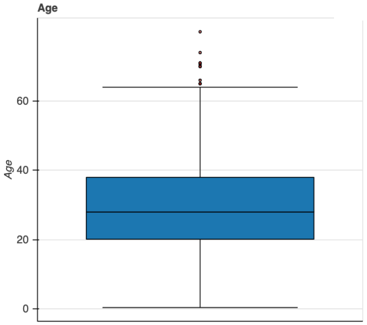
Write Code

Write Code



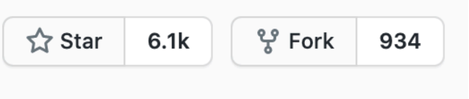
Minimum	0.42
5-th Percentile	4
Q1	20.125
Median	28
Q3	38
95-th Percentile	56
Maximum	80
Range	79.58
IQR	17.875





Current EDA Solutions in Python

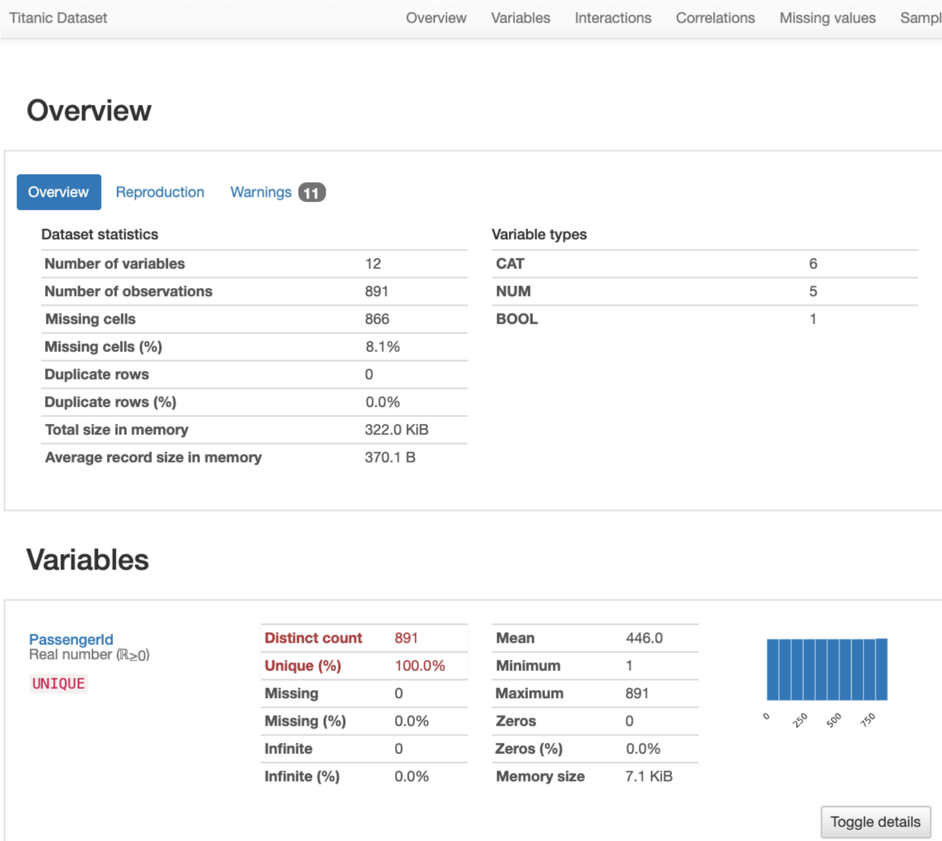
Solution 2: Pandas-profiling



☹ Slow

☹ Hard to Customize

```
profile = ProfileReport(df, title="Pandas Profiling Report")
```



Overview

Dataset statistics	
Number of variables	12
Number of observations	891
Missing cells	866
Missing cells (%)	8.1%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	322.0 KiB
Average record size in memory	370.1 B

Variable types	
CAT	6
NUM	5
BOOL	1

Variables

PassengerId
Real number ($\mathbb{R}_{\geq 0}$)
UNIQUE

Distinct count	891	Mean	446.0
Unique (%)	100.0%	Minimum	1
Missing	0	Maximum	891
Missing (%)	0.0%	Zeros	0
Infinite	0	Zeros (%)	0.0%
Infinite (%)	0.0%	Memory size	7.1 KiB

DataPrep.EDA Design Goals

EDA Solutions	Easy to Use	Interactive Speed	Easy to Customize
1. Pandas + Matplotlib	☹	☺	☺
2. Pandas-profiling	☺	☹	☹
3. DataPrep.EDA	☺	☺	☺

Key Idea

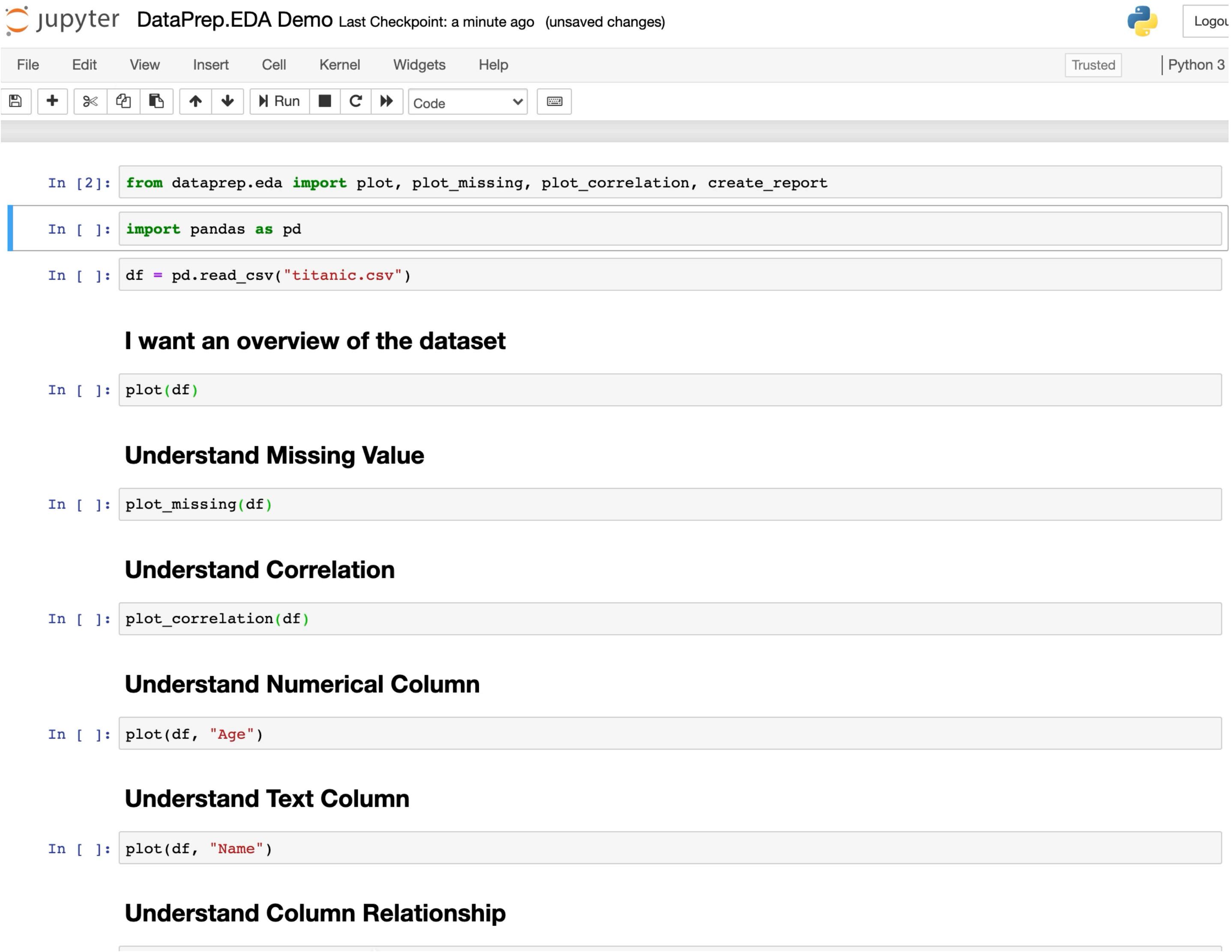
Task-Centric API Design

- Declarative
- Support both coarse-grained and fine-grained EDA tasks

Example

- plot(df): “I want to see an overview of the dataset”
- plot_missing(df): “I want to understand the missing values of the dataset”
- plot(df, x): “I want to understand the column x”
- plot(df, x, y): “I want to understand the relationship between x and y”
- …

DataPrep.EDA (Demo)



jupyter
DataPrep.EDA Demo Last Checkpoint: a minute ago (unsaved changes)
Logou
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
In [2]: from dataprep.eda import plot, plot_missing, plot_correlation, create_report
```

```
In [ ]: import pandas as pd
```

```
In [ ]: df = pd.read_csv("titanic.csv")
```

I want an overview of the dataset

```
In [ ]: plot(df)
```

Understand Missing Value

```
In [ ]: plot_missing(df)
```

Understand Correlation

```
In [ ]: plot_correlation(df)
```

Understand Numerical Column

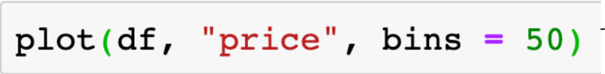
```
In [ ]: plot(df, "Age")
```

Understand Text Column

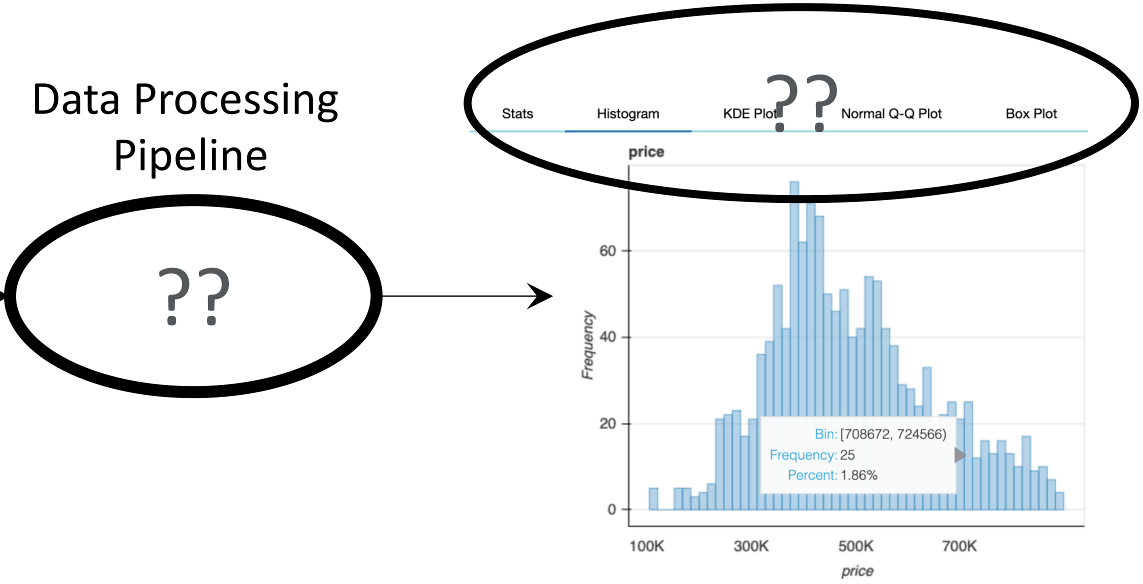
```
In [ ]: plot(df, "Name")
```

Understand Column Relationship

Under the Hood



```
plot(df, "price", bins = 50)
```



Data Processing Pipeline

??

Mapping Rules

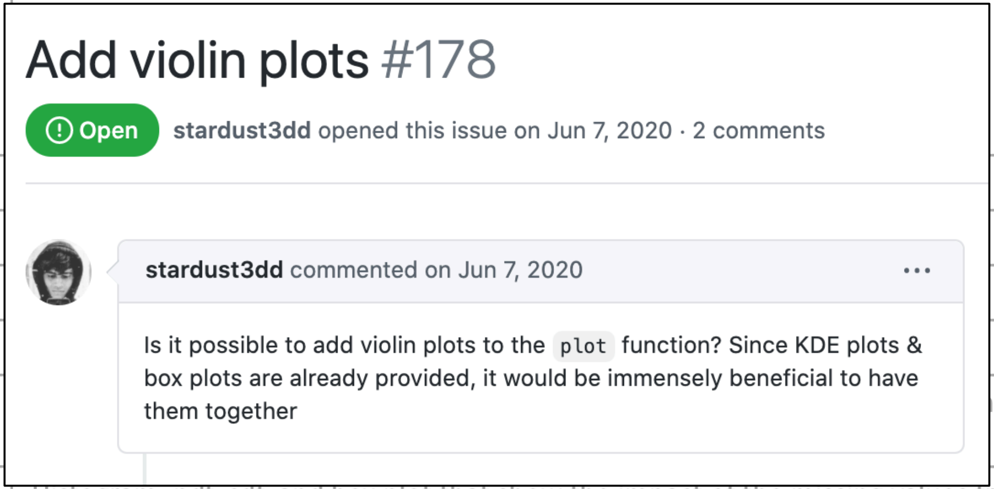
??

Bin: [708672, 724566)
Frequency: 25
Percent: 1.86%

Mapping Rules

N = Numerical, C = Categorical

Task-Centric API Design	Corresponding Stats/Plots
plot(df)	Dataset statistics, histogram or bar chart for each column
plot(df, col_1)	(1) col_1 = N ⟶ Column statistics, histogram, kde plot, qq-normal plot, box plot (2) col_1 = C ⟶ Column statistics, bar chart, pie chart, word cloud, word frequencies
plot(df, col_1, col_2)	
plot_correlation(df)	
plot_correlation(df, col_1)	
plot_correlation(df, col_1, col_2)	
plot_missing(df)	
plot_missing(df, col_1)	all other
plot_missing(df, col_1, col_2)	Histogram, pdf, cdf, and box plot that show the impact of the missing values from col_1 on col_2



Add violin plots #178

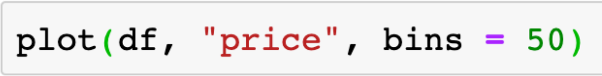
Open stardust3dd opened this issue on Jun 7, 2020 · 2 comments

stardust3dd commented on Jun 7, 2020

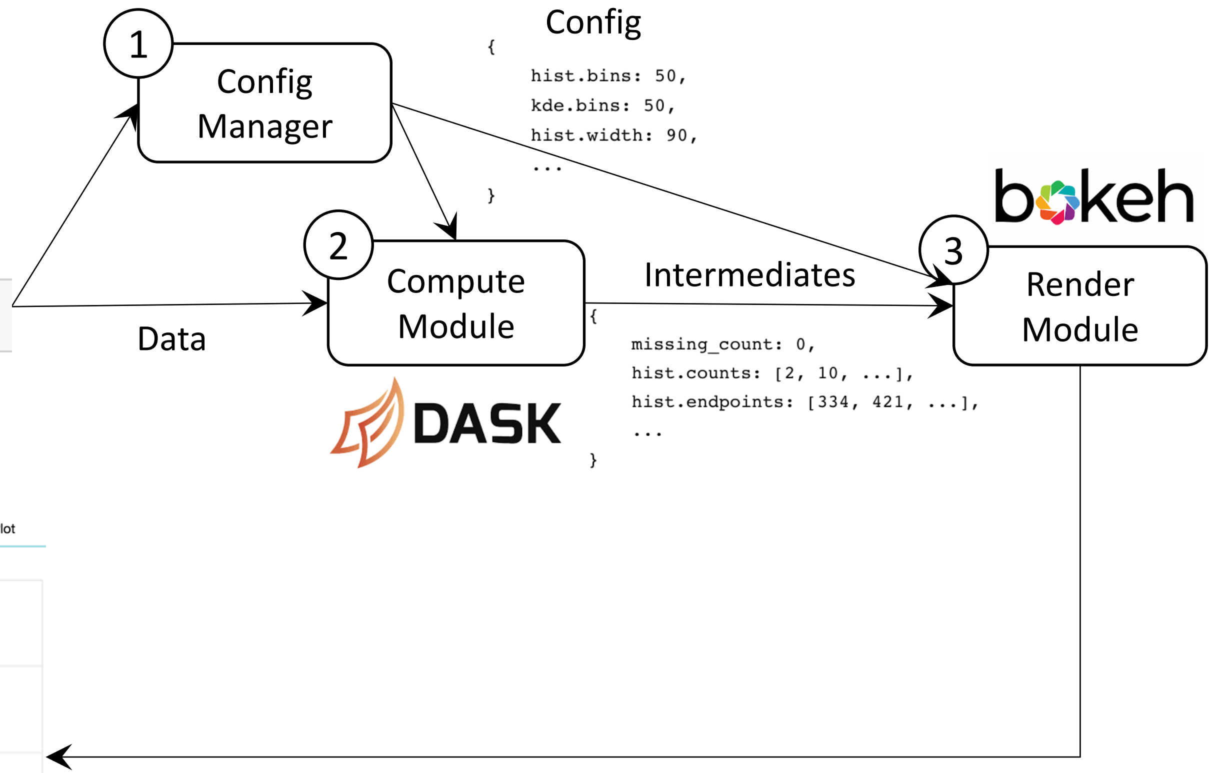
Is it possible to add violin plots to the `plot` function? Since KDE plots & box plots are already provided, it would be immensely beneficial to have them together

[1] https://www.data-to-viz.com/
[2] Exploratory data analysis with R
[3] Missingno: a missing data visualization suite
…

Data Processing Pipeline

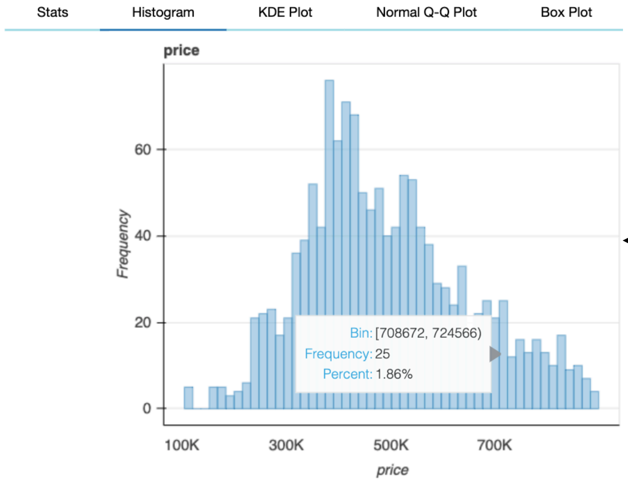


```
plot(df, "price", bins = 50)
```



```
{
    hist.bins: 50,
    kde.bins: 50,
    hist.width: 90,
    ...
}
```

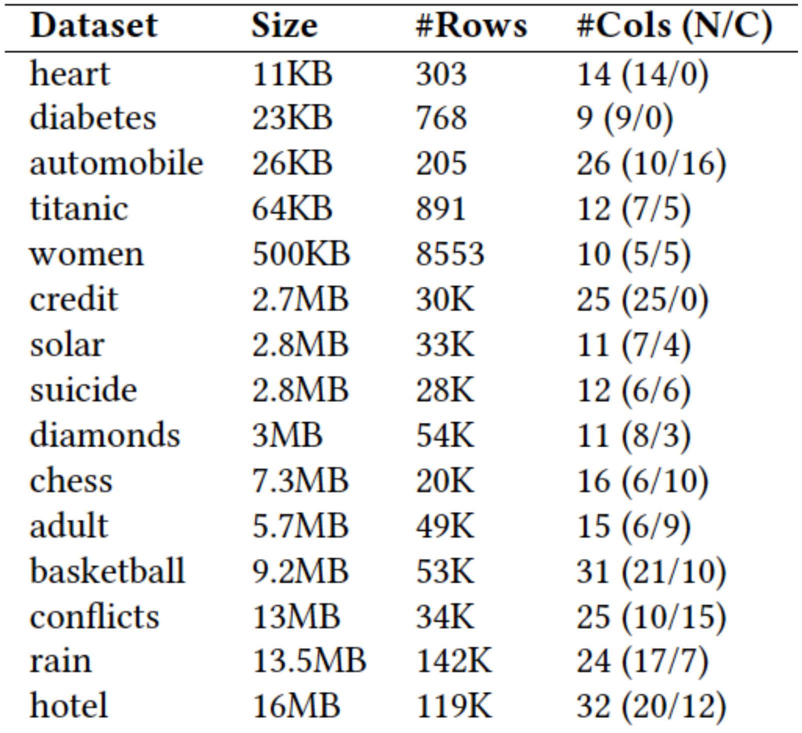
```
{
    missing_count: 0,
    hist.counts: [2, 10, ...],
    hist.endpoints: [334, 421, ...],
    ...
}
```



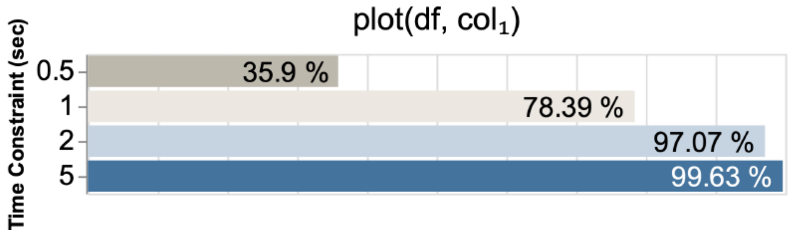
Bin: [708672, 724566)
Frequency: 25
Percent: 1.86%

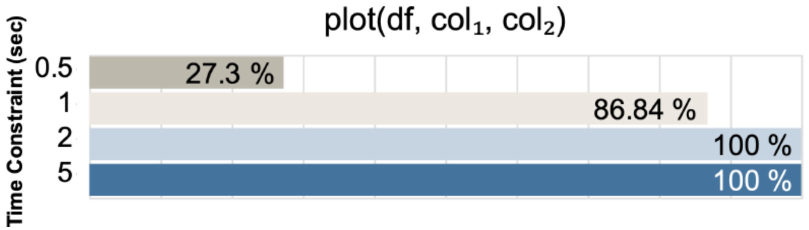
Interactive Speed

Ubuntu 16.04 Linux server with 64 GB memory and 8 Intel E7-4830 cores



Dataset	Size	#Rows	#Cols (N/C)
heart	11KB	303	14 (14/0)
diabetes	23KB	768	9 (9/0)
automobile	26KB	205	26 (10/16)
titanic	64KB	891	12 (7/5)
women	500KB	8553	10 (5/5)
credit	2.7MB	30K	25 (25/0)
solar	2.8MB	33K	11 (7/4)
suicide	2.8MB	28K	12 (6/6)
diamonds	3MB	54K	11 (8/3)
chess	7.3MB	20K	16 (6/10)
adult	5.7MB	49K	15 (6/9)
basketball	9.2MB	53K	31 (21/10)
conflicts	13MB	34K	25 (10/15)
rain	13.5MB	142K	24 (17/7)
hotel	16MB	119K	32 (20/12)



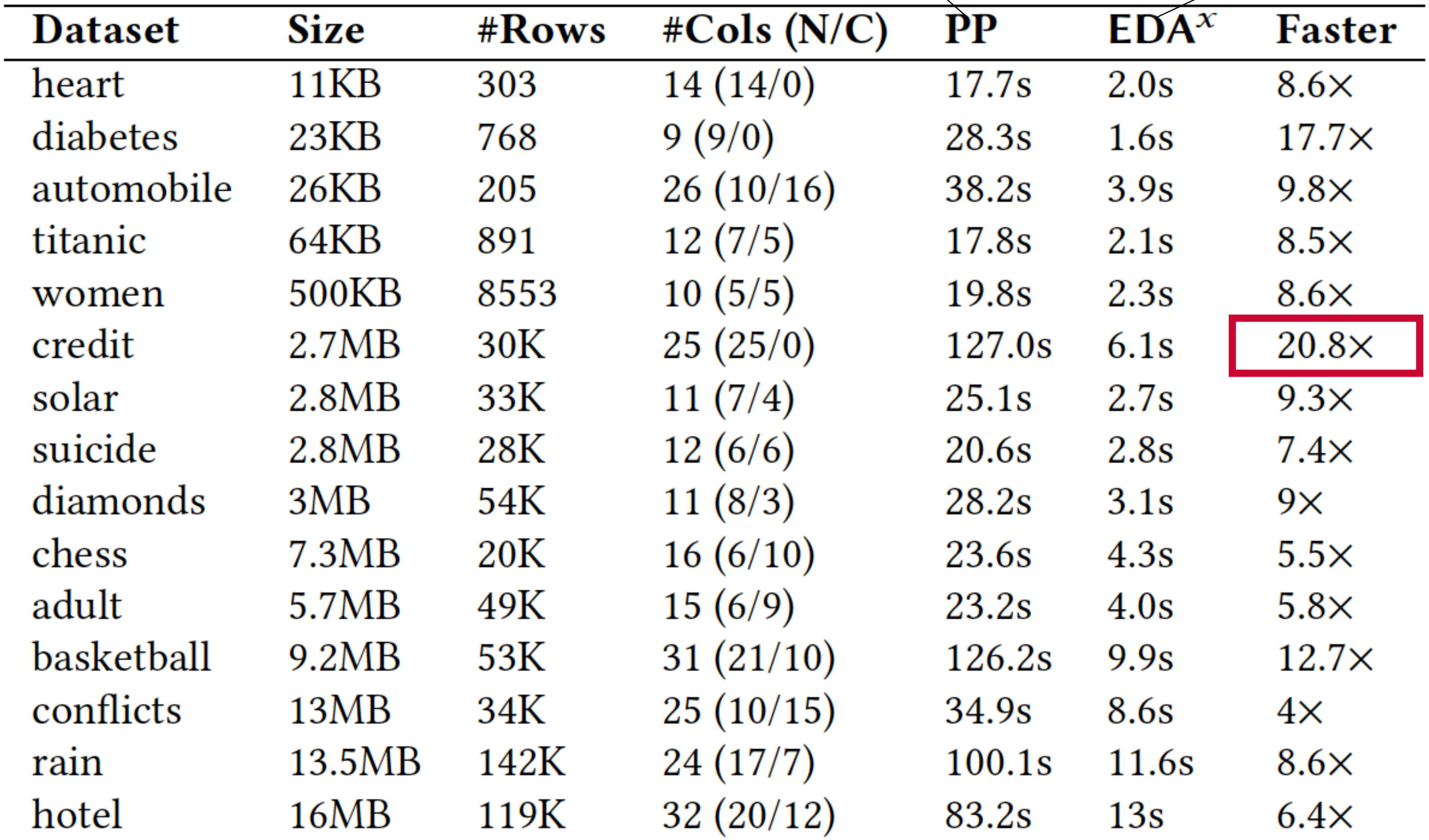


Efficiency Comparison

DataPrep.EDA vs Pandas-Profiling

Pandas-Profiling

DataPrep.EDA



Dataset	Size	#Rows	#Cols (N/C)	PP	EDA^x	Faster
heart	11KB	303	14 (14/0)	17.7s	2.0s	8.6×
diabetes	23KB	768	9 (9/0)	28.3s	1.6s	17.7×
automobile	26KB	205	26 (10/16)	38.2s	3.9s	9.8×
titanic	64KB	891	12 (7/5)	17.8s	2.1s	8.5×
women	500KB	8553	10 (5/5)	19.8s	2.3s	8.6×
credit	2.7MB	30K	25 (25/0)	127.0s	6.1s	20.8×
solar	2.8MB	33K	11 (7/4)	25.1s	2.7s	9.3×
suicide	2.8MB	28K	12 (6/6)	20.6s	2.8s	7.4×
diamonds	3MB	54K	11 (8/3)	28.2s	3.1s	9×
chess	7.3MB	20K	16 (6/10)	23.6s	4.3s	5.5×
adult	5.7MB	49K	15 (6/9)	23.2s	4.0s	5.8×
basketball	9.2MB	53K	31 (21/10)	126.2s	9.9s	12.7×
conflicts	13MB	34K	25 (10/15)	34.9s	8.6s	4×
rain	13.5MB	142K	24 (17/7)	100.1s	11.6s	8.6×
hotel	16MB	119K	32 (20/12)	83.2s	13s	6.4×

Easy to Customize (Available Soon)

How to Guide

```
In [*]:  plot(df, "age")
In [ ]:
In [ ]:
In [ ]:
In [ ]:
In [ ]:
In [ ]:
In [ ]:
In [ ]:
In [ ]:
```

DataPrep.EDA Takeaways

Innovation

The **first** task-centric EDA system in Python

Achieve three design goals

Easy to use
Interactive speed
Easy to customize

DataPrep.Connector

A Unified API Wrapper
to Simplify Web Data Collection

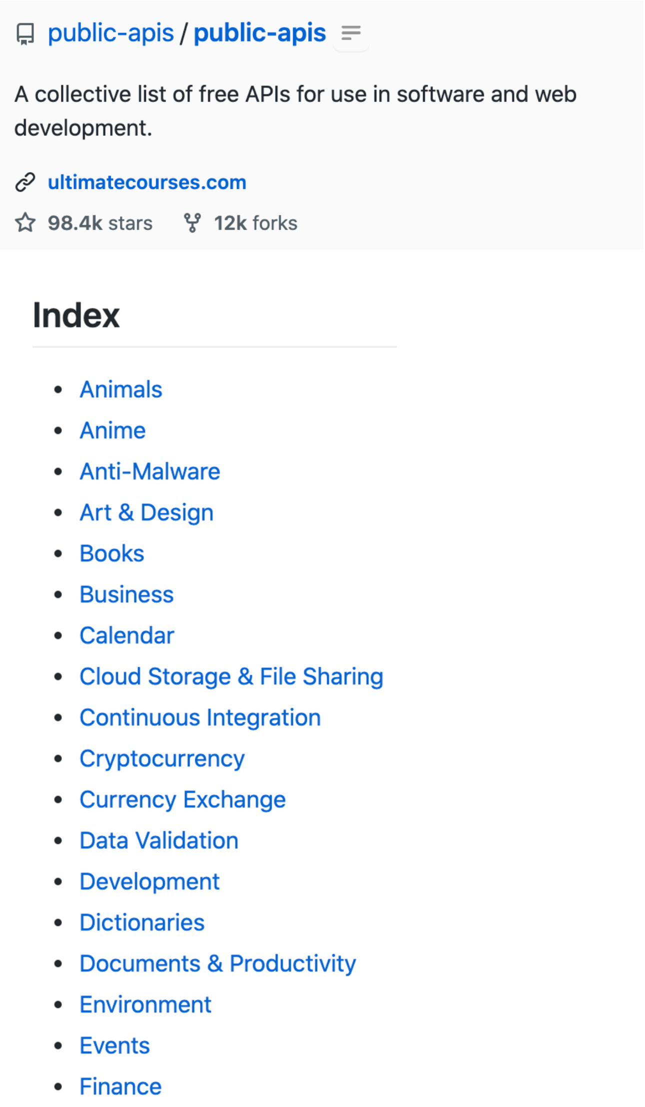
Data Collection Through Restful APIs

Social Data

Business Data

Event Data

Publication Data



public-apis / **public-apis**

A collective list of free APIs for use in software and web development.

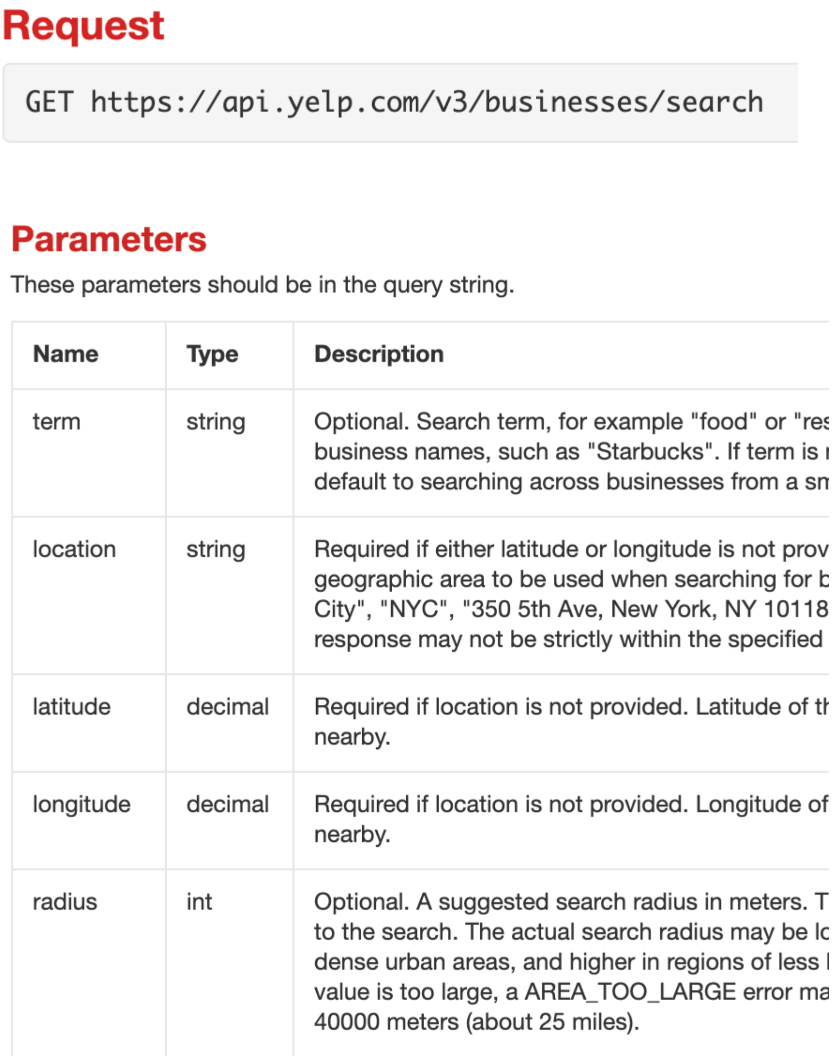
ultimatecourses.com

☆ **98.4k** stars **12k** forks

Index

- Animals
- Anime
- Anti-Malware
- Art & Design
- Books
- Business
- Calendar
- Cloud Storage & File Sharing
- Continuous Integration
- Cryptocurrency
- Currency Exchange
- Data Validation
- Development
- Dictionaries
- Documents & Productivity
- Environment
- Events
- Finance

Restful API Example



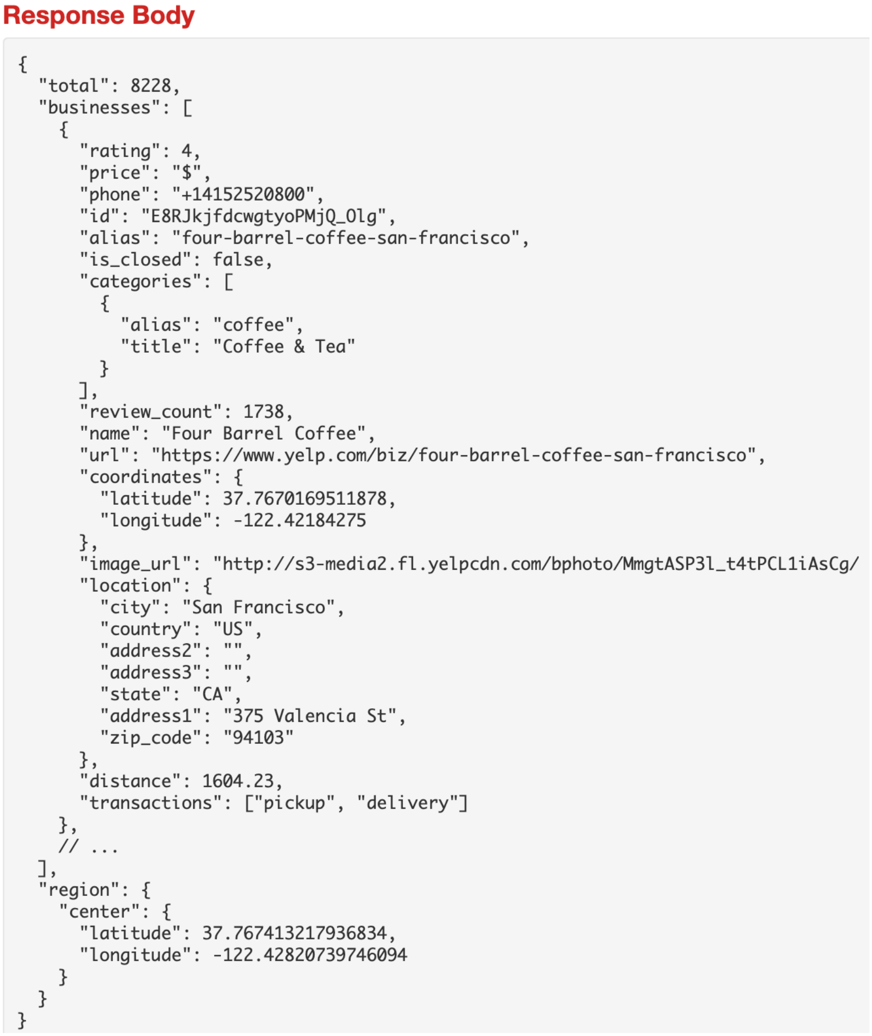
Request

```
GET https://api.yelp.com/v3/businesses/search
```

Parameters

These parameters should be in the query string.

Name	Type	Description
term	string	Optional. Search term, for example "food" or "res business names, such as "Starbucks". If term is default to searching across businesses from a sm
location	string	Required if either latitude or longitude is not prov geographic area to be used when searching for b City", "NYC", "350 5th Ave, New York, NY 10118 response may not be strictly within the specified
latitude	decimal	Required if location is not provided. Latitude of th nearby.
longitude	decimal	Required if location is not provided. Longitude of nearby.
radius	int	Optional. A suggested search radius in meters. T to the search. The actual search radius may be lo dense urban areas, and higher in regions of less value is too large, a AREA_TOO_LARGE error ma 40000 meters (about 25 miles).

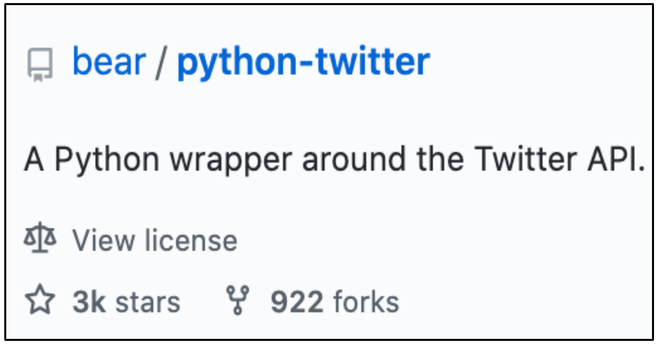


Response Body

```
{
  "total": 8228,
  "businesses": [
    {
      "rating": 4,
      "price": "$",
      "phone": "+14152520800",
      "id": "E8RJkjfdcwgtyoPMjQ_Olg",
      "alias": "four-barrel-coffee-san-francisco",
      "is_closed": false,
      "categories": [
        {
          "alias": "coffee",
          "title": "Coffee & Tea"
        }
      ],
      "review_count": 1738,
      "name": "Four Barrel Coffee",
      "url": "https://www.yelp.com/biz/four-barrel-coffee-san-francisco",
      "coordinates": {
        "latitude": 37.7670169511878,
        "longitude": -122.42184275
      },
      "image_url": "http://s3-media2.fl.yelpcdn.com/bphoto/MmgtASP3l_t4tPCL1iAsCg/
      "location": {
        "city": "San Francisco",
        "country": "US",
        "address2": "",
        "address3": "",
        "state": "CA",
        "address1": "375 Valencia St",
        "zip_code": "94103"
      },
      "distance": 1604.23,
      "transactions": ["pickup", "delivery"]
    },
    // ...
  ],
  "region": {
    "center": {
      "latitude": 37.767413217936834,
      "longitude": -122.42820739746094
    }
  }
}
```

Restful API Wrapper

Wrap API calls into Easy-to-Use Python Functions

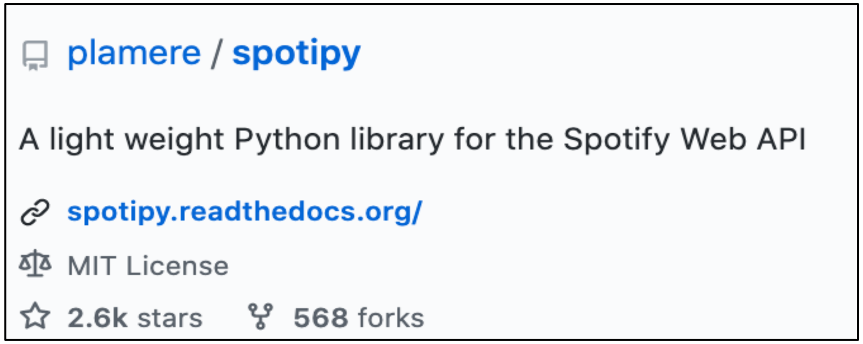


bear / python-twitter

A Python wrapper around the Twitter API.

View license

3k stars 922 forks



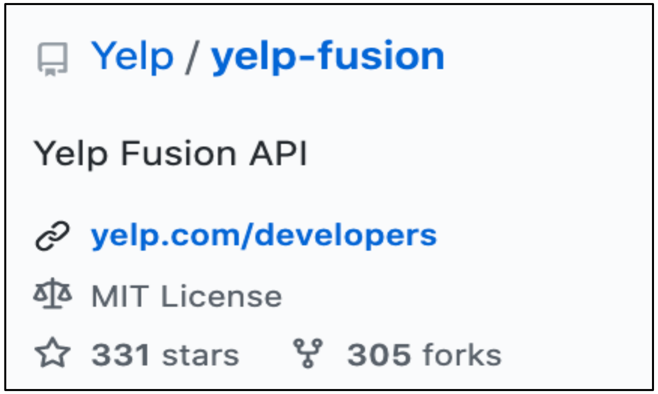
plamere / spotipy

A light weight Python library for the Spotify Web API

spotipy.readthedocs.org/

MIT License

2.6k stars 568 forks



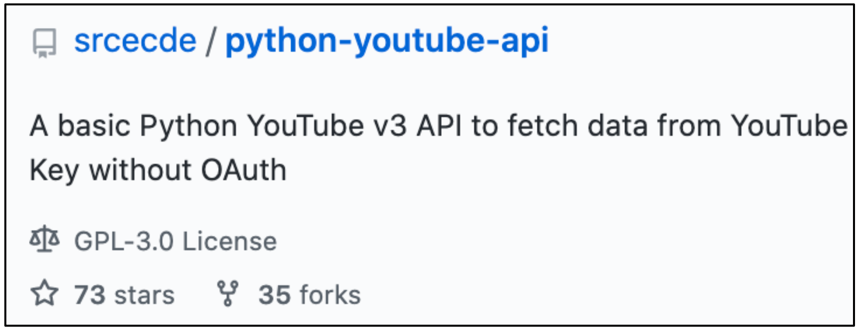
Yelp / yelp-fusion

Yelp Fusion API

yelp.com/developers

MIT License

331 stars 305 forks

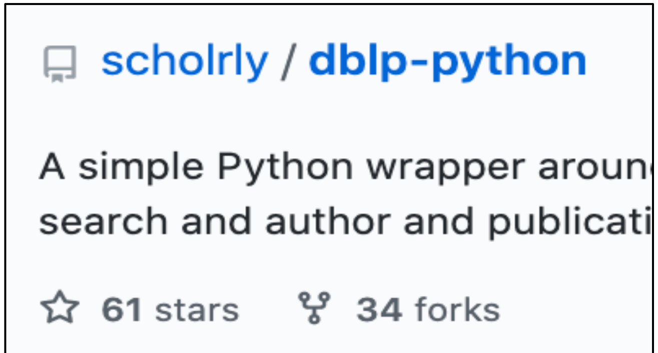


srcecde / python-youtube-api

A basic Python YouTube v3 API to fetch data from YouTube Key without OAuth

GPL-3.0 License

73 stars 35 forks



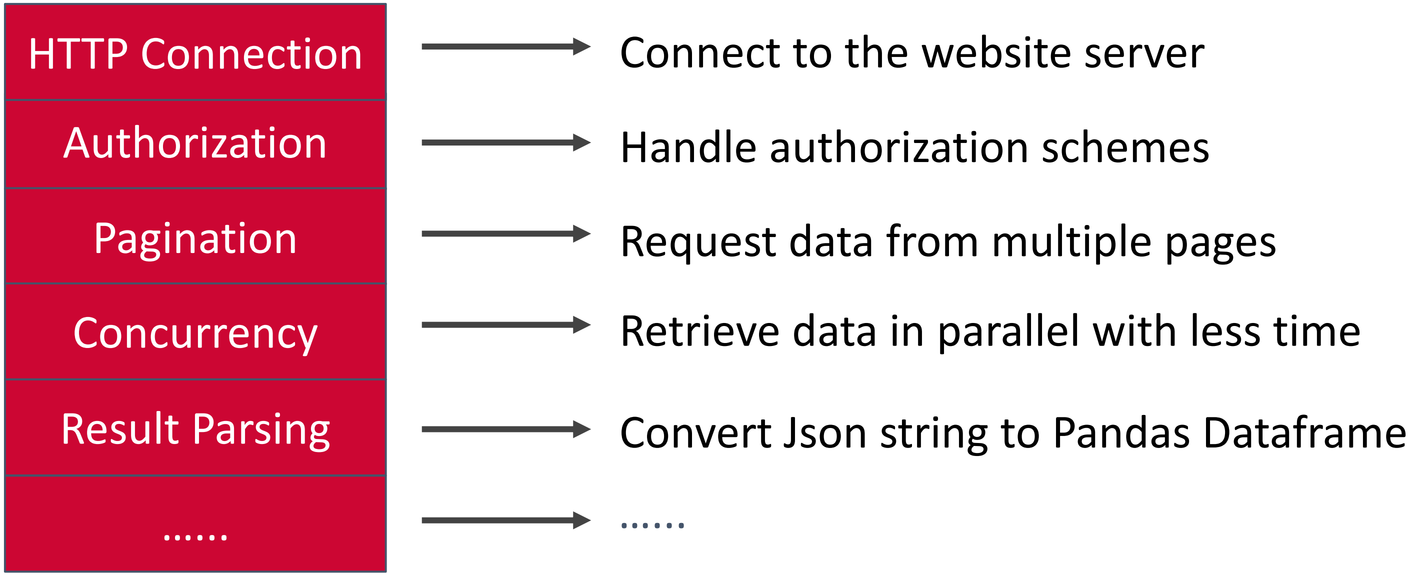
scholrly / dblp-python

A simple Python wrapper aroun
search and author and publicati

61 stars 34 forks

. . .

Build a New API Wrapper is **Tedious**!



HTTP Connection → Connect to the website server

Authorization → Handle authorization schemes

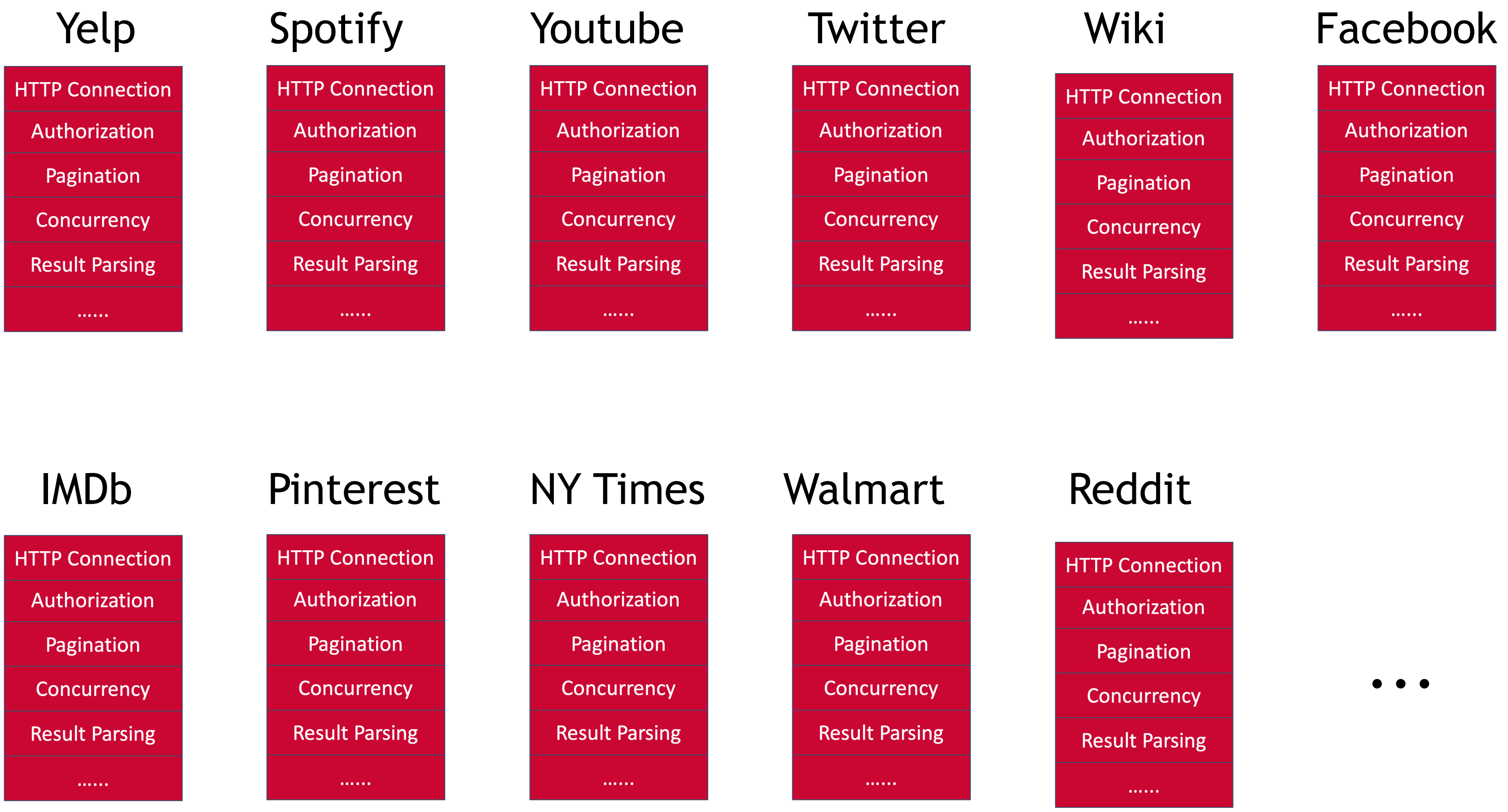
Pagination → Request data from multiple pages

Concurrency → Retrieve data in parallel with less time

Result Parsing → Convert Json string to Pandas Dataframe

…… → ……

If we don’t unify API wrappers, then …

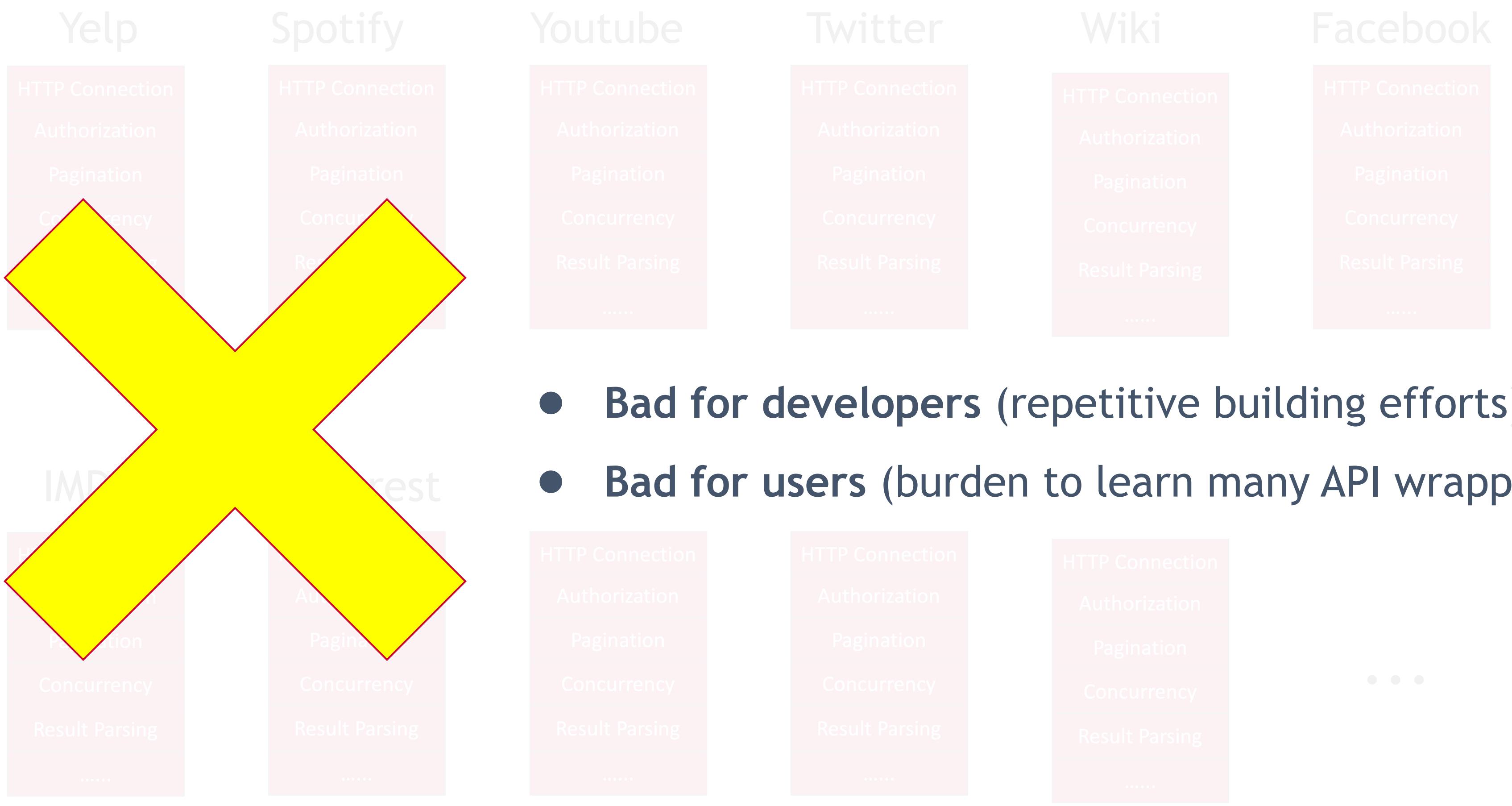


Yelp	Spotify	Youtube	Twitter	Wiki	Facebook
HTTP Connection	HTTP Connection	HTTP Connection	HTTP Connection	HTTP Connection	HTTP Connection
Authorization	Authorization	Authorization	Authorization	Authorization	Authorization
Pagination	Pagination	Pagination	Pagination	Pagination	Pagination
Concurrency	Concurrency	Concurrency	Concurrency	Concurrency	Concurrency
Result Parsing	Result Parsing	Result Parsing	Result Parsing	Result Parsing	Result Parsing
……	……	……	……	……	……

IMDb	Pinterest	NY Times	Walmart	Reddit
HTTP Connection	HTTP Connection	HTTP Connection	HTTP Connection	HTTP Connection
Authorization	Authorization	Authorization	Authorization	Authorization
Pagination	Pagination	Pagination	Pagination	Pagination
Concurrency	Concurrency	Concurrency	Concurrency	Concurrency
Result Parsing	Result Parsing	Result Parsing	Result Parsing	Result Parsing
……	……	……	……	……

• • •

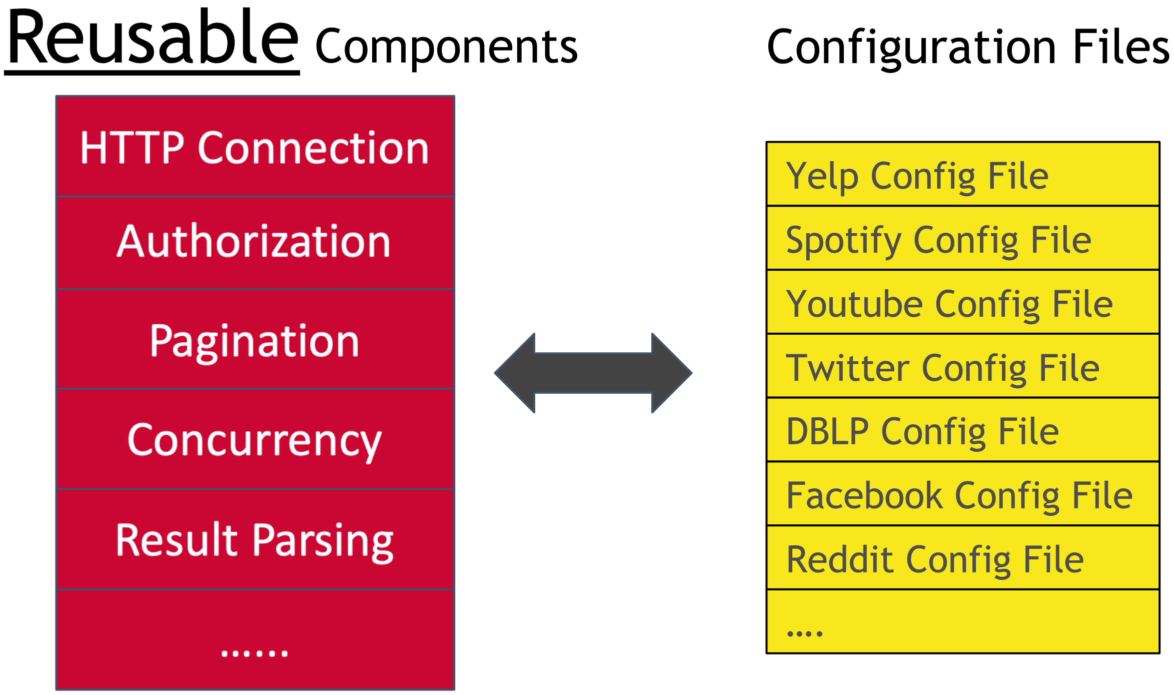
If we don't unify API wrappers, then …



- **Bad for developers** (repetitive building efforts)
- **Bad for users** (burden to learn many API wrappers)

DataPrep.Connector

A Unified API Wrapper



Reusable Components

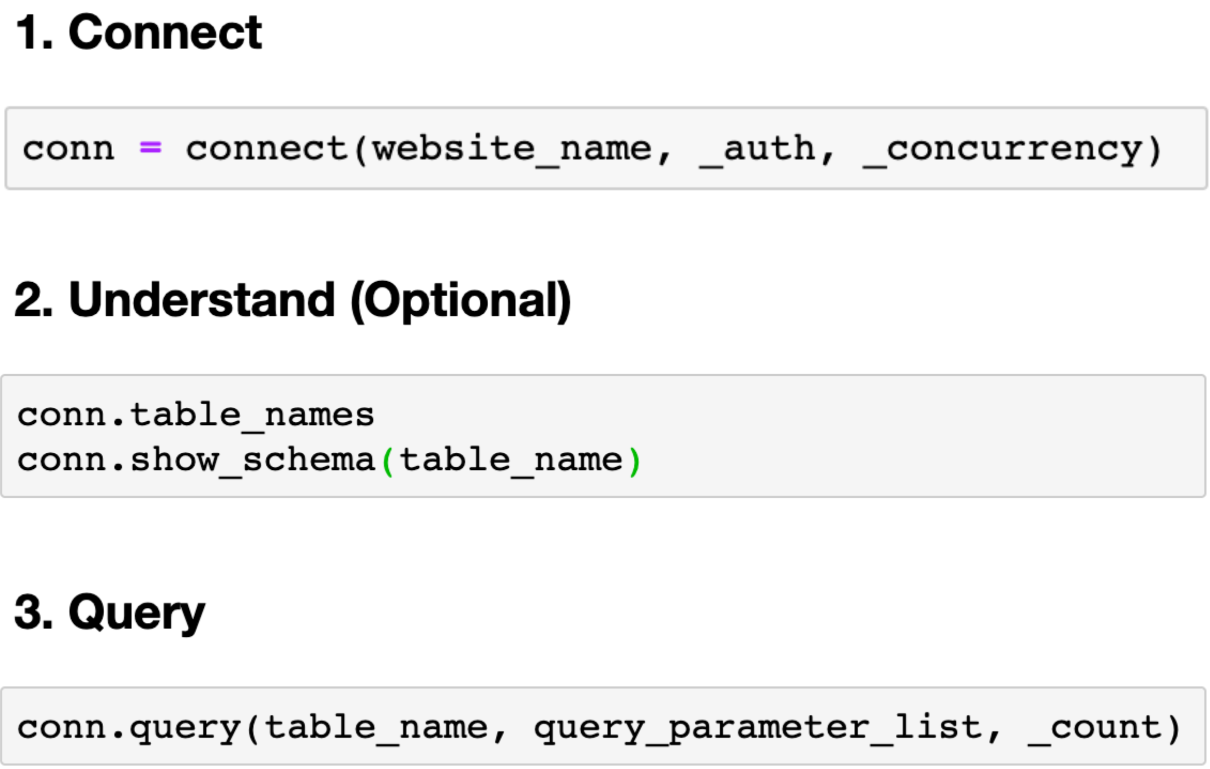
- HTTP Connection
- Authorization
- Pagination
- Concurrency
- Result Parsing
- ……

Configuration Files

- Yelp Config File
- Spotify Config File
- Youtube Config File
- Twitter Config File
- DBLP Config File
- Facebook Config File
- Reddit Config File
- ….

Good for developers (No repetitive building efforts)

The Unified API



1. Connect

```
conn = connect(website_name, _auth, _concurrency)
```

2. Understand (Optional)

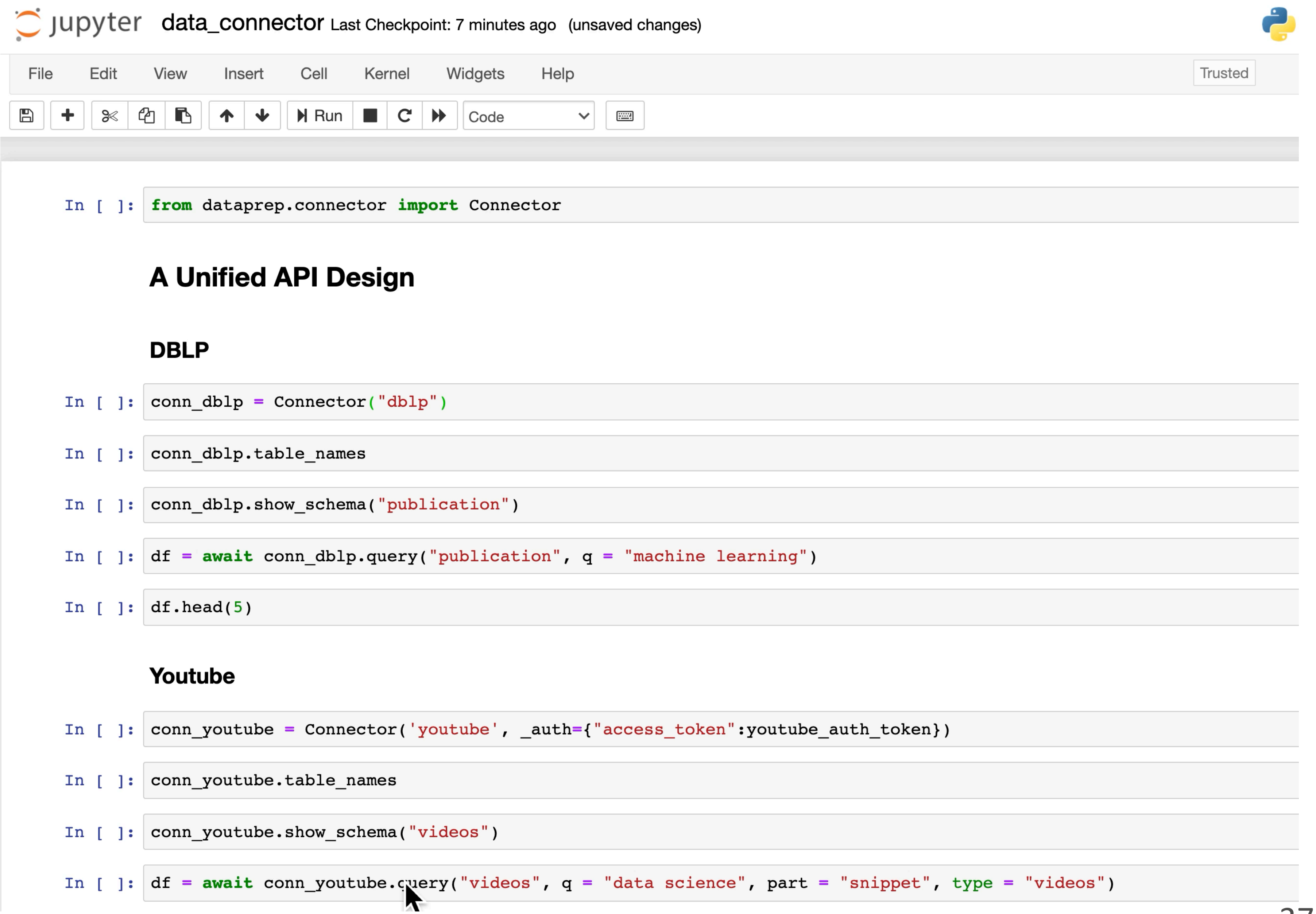
```
conn.table_names
conn.show_schema(table_name)
```

3. Query

```
conn.query(table_name, query_parameter_list, _count)
```

Good for users (No burden to learn many API wrappers)

DataPrep.Connector (Demo)


jupyter
data_connector Last Checkpoint: 7 minutes ago (unsaved changes)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Run
Code

```
In [ ]: from dataprep.connector import Connector
```

A Unified API Design

DBLP

```
In [ ]: conn_dblp = Connector("dblp")

In [ ]: conn_dblp.table_names

In [ ]: conn_dblp.show_schema("publication")

In [ ]: df = await conn_dblp.query("publication", q = "machine learning")

In [ ]: df.head(5)
```

Youtube

```
In [ ]: conn_youtube = Connector('youtube', _auth={"access_token":youtube_auth_token})

In [ ]: conn_youtube.table_names

In [ ]: conn_youtube.show_schema("videos")

In [ ]: df = await conn_youtube.query("videos", q = "data science", part = "snippet", type = "videos")
```

DataPrep.Connector Takeaways

Innovation

The **first** unified API Wrapper in Python

Good For Developers

Speed up wrapper development process

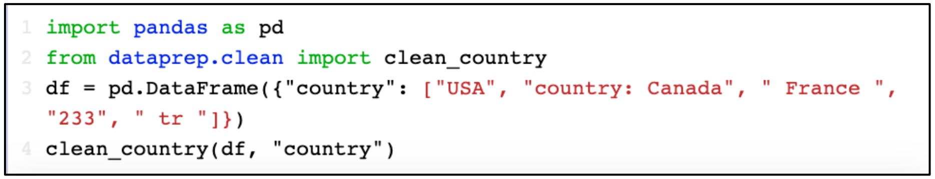
Good For Users

Speed up data collection from Web APIs

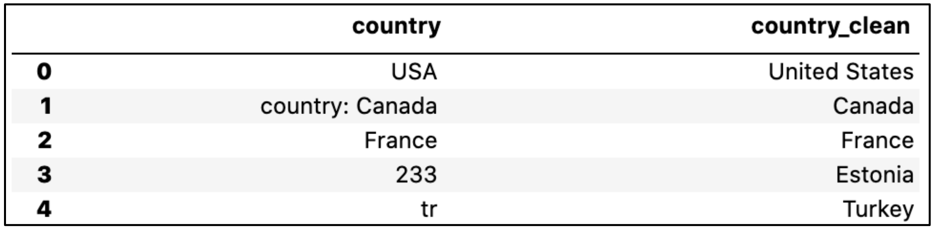
Talk Outline

1. DataPrep Overview
2. Dive into DataPrep
   - DataPrep.EDA
   - DataPrep.Connector
3. Future Direction

Future Direction



```
import pandas as pd
from dataprep.clean import clean_country
df = pd.DataFrame({"country": ["USA", "country: Canada", " France ",
"233", " tr "]})
clean_country(df, "country")
```



	country	country_clean
0	USA	United States
1	country: Canada	Canada
2	France	France
3	233	Estonia
4	tr	Turkey

DataPrep.EDA

- Make plots look attractive
- Understand multiple dataframes (plot_diff, plot_db, …)

DataPrep.Connector

- Speed up read_sql() with arrow and parallel connection

DataPrep.Clean

- **Goal:** Implement 100+ clean_{type}(df, x) functions
- **Example:** clean_email, clean_date, clean_phone, clean_country, etc.
- **Application:** Data Validation, Data Standardization, Semantic Type Detection

SFU

dataprep

The easiest way to prepare data in Python

```
pip install -U dataprep
```

Thank you!

http://dataprep.ai