Pandas Recipes for New Python Users

SHARCNET: General Interest Webinar

Tyler Collins HPC Analyst, Brock University

SHARCNET: Tyler Collins

Outline: Today's Aim

- Introduce Pandas
- Explain motivation
- Live demo
- Recap
- Question period

Hopefully at the end of this talk, you will use Pandas in your own projects!

This webinar and its materials can be found on GitHub, here: https://github.com/Andesha/sharcnet-pandas

Some Python Commentary

- Python sure is awesome but awesome isn't free!
- Even other languages have this narrative:
 - "There are no zero cost abstractions!" Chandler Carruth, 2019
- "Each abstraction must provide more benefit than cost"
 - From the same talk as above
- What about when you need advanced data structure processing?
 - You suffer presumably...

- Pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language
- Intelligent label-based slicing, fancy indexing, and subsetting of large data sets
- Time series-functionality: date range generation and frequency conversion, moving window statistics, date shifting and lagging. Even create domain-specific time offsets and join time series without losing data

With the support of:



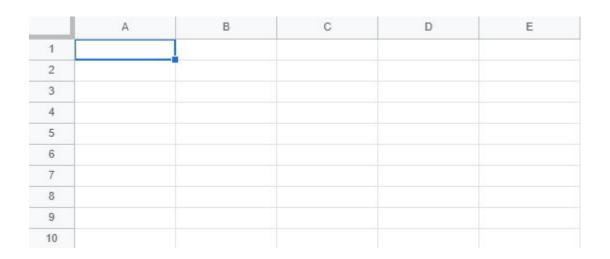












	A	В	C	D	E	F	G	Н	1	J
	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	name	
	18	8	307	130	3504	12	70	1	chevrolet chevel	le malibu
	15	8	350	165	3693	11.5	70	1	buick skylark 320)
	18	8	318	150	3436	11	70	1	plymouth satellite	е
	16	8	304	150	3433	12	70	1	amc rebel sst	
	17	8	302	140	3449	10.5	70	1	ford torino	
	15	8	429	198	4341	10	70	1	ford galaxie 500	
	14	8	454	220	4354	9	70	1	chevrolet impala	
	14	8	440	215	4312	8.5	70	1	plymouth fury iii	
D	14	8	455	225	4425	10	70	1	pontiac catalina	
1	15	8	390	190	3850	8.5	70	1	amc ambassador dpl	
2	15	8	383	170	3563	10	70	1	dodge challenger se	
3	14	8	340	160	3609	8	70	1	plymouth 'cuda 340	
1	15	8	400	150	3761	9.5	70	1	chevrolet monte carlo	
5	14	8	455	225	3086	10	70	1	buick estate wagon (sw)	
5	24	4	113	95	2372	15	70	3	toyota corona mark ii	
7	22	6	198	95	2833	15.5	70	1	plymouth duster	
8	18	6	199	97	2774	15.5	70	1	amc hornet	
9	21	6	200	85	2587	16	70	1	ford maverick	
0	27	4	97	88	2130	14.5	70	3	datsun pl510	

What if I want to do **ALL** of the following tasks:

- Create a new column based on ratio of two columns
- Make sure there are no NaN values in the data
- Convert origin to a string instead of an "enum"
- Slice the data such that you get a view of all ford cars made after 1975 when their horsepower is greater than the original average
- Based on some list of pairs containing *name* as it appears in the dataset, overwrite *origin* with the full origin date of the car

If you're anything like me in the past you would:

- Export to CSV
- Write a quick Python script to ingest the data
- Treat everything as a list, and do the various conditional mappings
- Write it back to CSV
- Done!

What if the data was already in memory and was *massive*? Why write to a file and create dependencies?

Assume the input is a TSV, and the output must be CSV:

```
quick_parse.py > ...
    import pandas as pd

df = pd.read_csv('mpg.tsv', sep='\t')
    df = df.dropna()
    df['disp_div_cyl'] = df['displacement'].div(df['cylinders'])
    df['origin'] = df['origin'].map({1:'NA', 2:'EU', 3:'AS'})
    avg_hp = df['horsepower'].mean()
    sub_df = df[(df['car_name'].str.contains('ford'))]
    sub_df = sub_df[(sub_df['horsepower'] > avg_hp) & (sub_df['model_year'] > 75)]
    sub_df.to_csv('sub_mpg.csv', index=False)
```

Are You Not Entertained?

- Strong and safe IO
- Flexible conditional slicing
- Intuitive mapping

How easy is it to learn and use really?

What about speed?

- Critical code paths are written in Cython or C
- Lots of academic/commercial studies to review if performance is a concern

Live Demo

- We will be working with the MPG data discussed previously
 - Quinlan,R. (1993). Combining Instance-Based and Model-Based Learning. In Proceedings on the Tenth
 International Conference of Machine Learning, 236-243, University of Massachusetts, Amherst. Morgan
 Kaufmann.
- Reference material is on GitHub here:
 - https://github.com/Andesha/sharcnet-pandas
- We will be using Jupyter lab
 - The starting notebook and a completed notebook are on GitHub



Post Demo Discussion

- Hopefully you're convinced and you have some templates to work from
- Things we did not have time to discuss but totally exist and are awesome:
 - Other forms of <u>serialization</u>
 - All sorts of <u>datetime</u> functionality
 - o <u>Grouping</u> and aggregating data together
 - Pivoting
 - Unpivot aka <u>melting</u>
 - Complexity and runtime

Express your desired operation to Google and let Stack Overflow do its work!

Takeaways

- Pandas provides a very strong set of tools that can deal with a multitude of data types
- Wide amount of support on the internet including excellent API documentation
- Many other popular libraries support Pandas

Thanks very much!

tk11br@sharcnet.ca

Questions?



