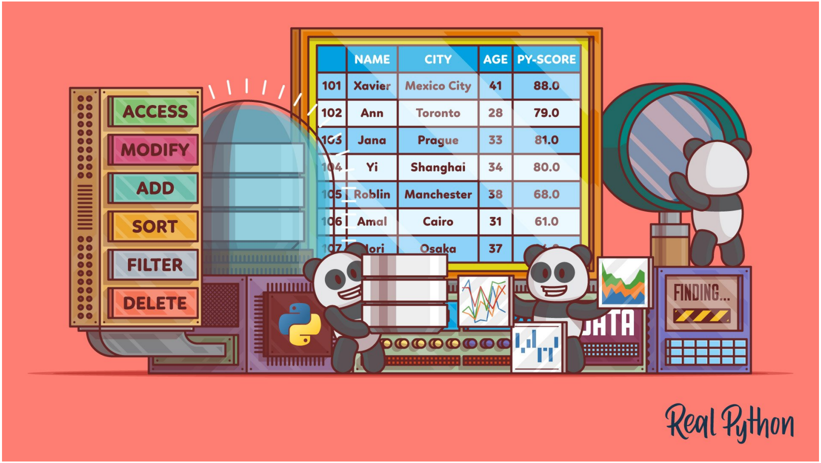
Pandas : transition douce d’Excel vers Python

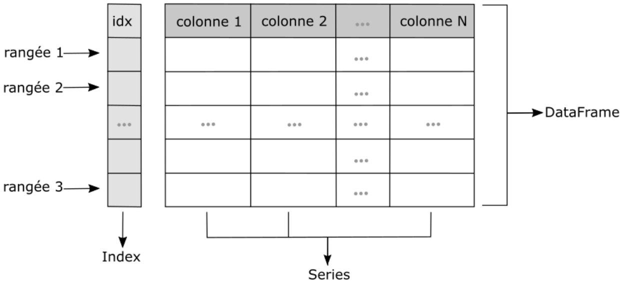


	NAME	CITY	AGE	PY-SCORE
101	Xavier	Mexico City	41	88.0
102	Ann	Toronto	28	79.0
103	Jana	Prague	33	81.0
104	Yi	Shanghai	34	80.0
105	Roblin	Manchester	38	68.0
106	Amal	Cairo	31	61.0
107	Nori	Osaka	37	[illegible]

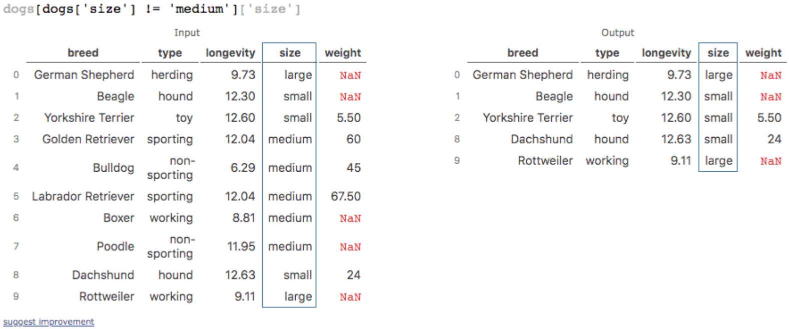
- Facilite le traitement des données sous forme tableau
- Usage comme base de données
- Montée en taille avec *Dask*
- Brique de base dans beaucoup d'autres bibliothèques
- Permet de faire facilement statistiques/visualisations exploratoires

Structure d’un tableau

Structure d'un tableau *Pandas*



Afficher et manipuler les tableaux



```
dogs[dogs['size'] != 'medium']['size']
```

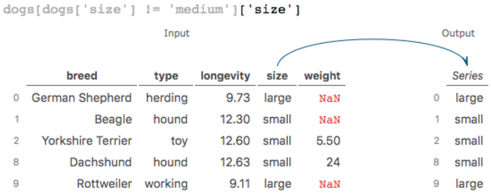
Input

	breed	type	longevity	size	weight
0	German Shepherd	herding	9.73	large	NaN
1	Beagle	hound	12.30	small	NaN
2	Yorkshire Terrier	toy	12.60	small	5.50
3	Golden Retriever	sporting	12.04	medium	60
4	Bulldog	non-sporting	6.29	medium	45
5	Labrador Retriever	sporting	12.04	medium	67.50
6	Boxer	working	8.81	medium	NaN
7	Poodle	non-sporting	11.95	medium	NaN
8	Dachshund	hound	12.63	small	24
9	Rottweiler	working	9.11	large	NaN

Output

	breed	type	longevity	size	weight
0	German Shepherd	herding	9.73	large	NaN
1	Beagle	hound	12.30	small	NaN
2	Yorkshire Terrier	toy	12.60	small	5.50
8	Dachshund	hound	12.63	small	24
9	Rottweiler	working	9.11	large	NaN

suggest improvement



```
dogs[dogs['size'] != 'medium']['size']
```

Input

	breed	type	longevity	size	weight
0	German Shepherd	herding	9.73	large	NaN
1	Beagle	hound	12.30	small	NaN
2	Yorkshire Terrier	toy	12.60	small	5.50
8	Dachshund	hound	12.63	small	24
9	Rottweiler	working	9.11	large	NaN

Output

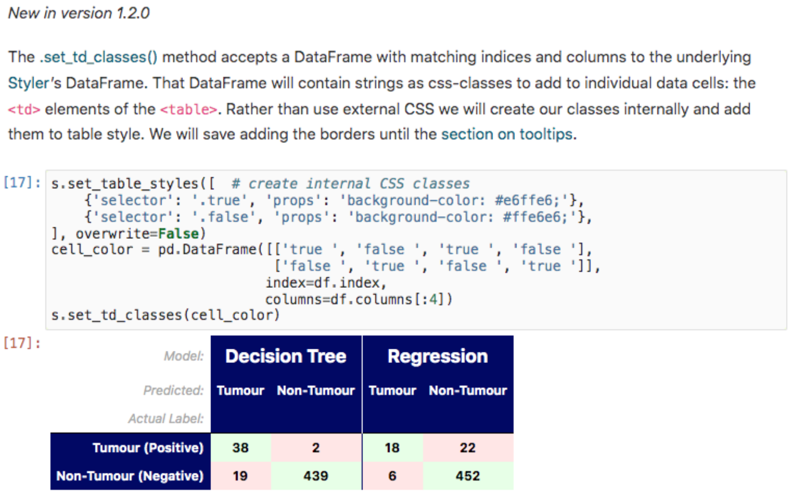
	Series
0	large
1	small
2	small
8	small
9	large

```
https://pandastutor.com/
```

Tellement de possibilités

- Pour voir tout ce qu'il est possible de faire : `https://pandas.pydata.org/pandas-docs/stable/user_guide/cookbook.html`
- Pour **voir** les opérations : `https://pandastutor.com/`

Avec des possibilités d'aller loin...



New in version 1.2.0

The .set_td_classes() method accepts a DataFrame with matching indices and columns to the underlying Styler's DataFrame. That DataFrame will contain strings as css-classes to add to individual data cells: the <td> elements of the <table>. Rather than use external CSS we will create our classes internally and add them to table style. We will save adding the borders until the section on tooltips.

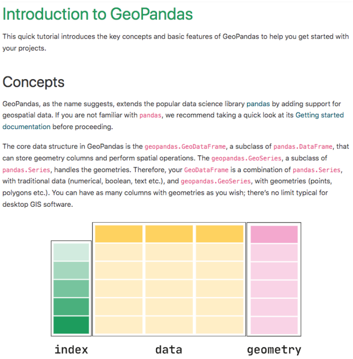
```
[17]: s.set_table_styles([  # create internal CSS classes
          {'selector': '.true', 'props': 'background-color: #e6ffe6;'},
          {'selector': '.false', 'props': 'background-color: #ffe6e6;'},
      ], overwrite=False)
      cell_color = pd.DataFrame([['true ', 'false ', 'true ', 'false '],
                                 ['false ', 'true ', 'false ', 'true ']],
                                index=df.index,
                                columns=df.columns[:4])
      s.set_td_classes(cell_color)
```

[17]:

Model:	Decision Tree		Regression	
Predicted:	Tumour	Non-Tumour	Tumour	Non-Tumour
Actual Label:				
Tumour (Positive)	38	2	18	22
Non-Tumour (Negative)	19	439	6	452

https://pandas.pydata.org/docs/user_guide/style.html

Support d’autres usages : ex. Geopandas



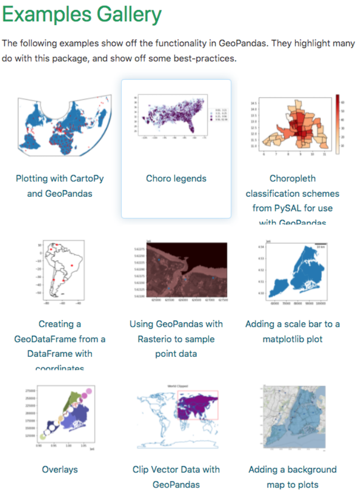
Introduction to GeoPandas

This quick tutorial introduces the key concepts and basic features of GeoPandas to help you get started with your projects.

Concepts

GeoPandas, as the name suggests, extends the popular data science library pandas by adding support for geospatial data. If you are not familiar with `pandas`, we recommend taking a quick look at its Getting started documentation before proceeding.

The core data structure in GeoPandas is the `geopandas.GeoDataFrame`, a subclass of `pandas.DataFrame`, that can store geometry columns and perform spatial operations. The `geopandas.GeoSeries`, a subclass of `pandas.Series`, handles the geometries. Therefore, your `GeoDataFrame` is a combination of `pandas.Series`, with traditional data (numerical, boolean, text etc.), and `geopandas.GeoSeries`, with geometries (points, polygons etc.). You can have as many columns with geometries as you wish; there's no limit typical for desktop GIS software.



Examples Gallery

The following examples show off the functionality in GeoPandas. They highlight many do with this package, and show off some best-practices.

Plotting with CartoPy and GeoPandas

Choro legends

Choropleth classification schemes from PySAL for use with GeoPandas

Creating a GeoDataFrame from a DataFrame with coordinates

Using GeoPandas with Rasterio to sample point data

Adding a scale bar to a matplotlib plot

Overlays

Clip Vector Data with GeoPandas

Adding a background map to plots

De Pandas aux statistiques

Intégration de Pandas dans le *workflow*

- Mise en forme propre de tableaux
- Traitements exploratoires (statistiques et visualisations)
- Traitements plus avancés avec d'autres bibliothèques :
  - Statsmodel, Scikit-learn, Pingouin, etc. pour les statistiques
  - Matplotlib, Seaborn, etc. pour les visualisations