1ª Emisión

Módulo 05

Manipulación y visualización de datos con Python

Mtro. Ricardo Daniel Alanis Tamez

DGTIC

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
Dirección General de Cómputo y de Tecnologías de información y Comunicación
Dirección de Docencia en TIC

Educación Continua
1971 - 2021

Introducción a Pandas

Manipulación y visualización de datos con Python

Ricardo Alanís

Presentación

En este tema aprenderemos sobre las herramientas que contiene Python para el análisis de datos para luego entrar a detalle sobre Pandas - *la* librería base para el análisis de datos y sus características.

Objetivo

El participante identificará la librería Pandas dentro del ambiente de Python Científico. Así como los demás elementos que conforman dicho ambiente y el contexto en que se desarrolló. Para luego hablar de la importancia de los elementos que plantea como herramienta computacional.

Agenda de hoy

1. **Ambiente de Python Científico**
2. **Investigación sobre las librerías disponibles y exposición**
3. **Autor y datos de Pandas, Cómo contribuir**
4. **Elementos que conforman la librería**
5. **Tipos de datos que expone la librería**

Primero, una anécdota

Los llamamos "hechos de tránsito" y no "accidentes" ya que un accidente es un hecho que es fortuito, que ocurre por azar o casualidad y de forma inesperada. No se puede prevenir. Por el contrario, al utilizar el término "siniestro de tránsito" o "siniestro vial" o "hecho de tránsito" son hechos que podemos evitar y cuyos factores podemos identificar.

AVENIDAS **CONFLICTIVAS**

Cinco vías rápidas, sin semáforos, son las que se encuentran en la cima de reporte de percances

AVENIDAS	ACCIDENTES
Garza Sada	1241
Gonzalitos	1084
Leones	897
Constitución	745
Morones Prieto	704
Madero	606
Lincoln	596
Ruiz Cortines	509
Alfonso Reyes	431
Bernardo Reyes	307
Venustiano Carranza	293
Cuauhtémoc	225
Pablo González	209
Pino Suárez	178
Juárez	163

Excel vs DataFrame

B	C	D	E	F
ast name	First name	Birthday	Country	Date of purchase
avidson	Michael	04/03/1986	United States	10/12/2016
ito	Jim	09/01/1994	United Kingdom	02/02/2016
ohnson	Tom	23/08/1972	France	02/11/2016
ewis	Peter	18/10/1979	Germany	22/11/2016
oenig	Edward	13/05/1983	Argentina	26/03/2015
reston	Jack	16/06/1991	United States	06/11/2016
mith	David	11/03/1965	Canada	15/11/2016
rown	Luis	03/09/1997	Australia	03/07/2015
iller	Thomas	07/01/1980	Germany	07/11/2016
illiams	Bill	26/07/1960	United States	20/11/2015
emini	Alexia	12/09/1995	Canada	11/03/2017
ond	James	25/02/1975	United Kingdom	12/08/2017
urgle	Patricia	01/12/1990	United States	18/01/2015

```
In [13]: # importing libraries

         from __future__ import print_function
         from ipywidgets import interact, interactive, fixed, inte
         from IPython.core.display import display, HTML

         import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
         import plotly.express as px
         import folium
         import plotly.graph_objects as go
         import seaborn as sns
         import ipywidgets as widgets

In [14]: # loading data right from the source:
         death_df = pd.read_csv('https://raw.githubusercontent.com
         confirmed_df = pd.read_csv('https://raw.githubusercontent
         recovered_df = pd.read_csv('https://raw.githubusercontent
         country_df = pd.read_csv('https://raw.githubusercontent.c

In [15]: confirmed_df.head()
```

Las herramientas en Python para DS

NumPy
Base N-dimensional array package

SciPy library
Fundamental library for scientific computing

Matplotlib
Comprehensive 2-D plotting

IPython
Enhanced interactive console

SymPy
Symbolic mathematics

pandas
Data structures & analysis

Ejercicio 1.1 - Hagamos grupos para entender los paquetes

NumPy
Base N-dimensional array package

SciPy library
Fundamental library for scientific computing

Matplotlib
Comprehensive 2-D plotting

IPython
Enhanced interactive console

SymPy
Symbolic mathematics

pandas
Data structures & analysis

Localiza:

1. La utilidad principal
2. Documentación
3. Repo e issues

23 minutos y presentación

Introducción a Pandas

pandas

Data structures & analysis

History [edit]

Developer Wes McKinney started working on pandas in 2008 while at AQR Capital Management out of the need for a high performance, flexible tool to perform quantitative analysis on financial data. Before leaving AQR he was able to convince management to allow him to open source the library.

Another AQR employee, Chang She, joined the effort in 2012 as the second major contributor to the library.

In 2015, pandas signed on as a fiscally sponsored project of NumFOCUS, a 501(c)(3) nonprofit charity in the United States.[12]

Pandas

pandas-dev / pandas

Sponsor | Unwatch | Unstar 29.8k | Fork 12.4k

Code | Issues 3.5k | Pull requests 198 | Actions | Projects 6 | Wiki | Releases 131

master | 15 branches | 131 tags

mroeschke TST: Add tests for old issues 2 (#41493) a246270 16 hours ago 26,831 commits

.github	DEPS: bump pyarrow version to 0.17.0 #38870 (#41476)	3 days ago
LICENSES	DEPS/CLN: remove distutils usage (#41207)	14 days ago
asv_bench	[ArrowStringArray] PERF: use pa.compute.match_substring_regex for ...	2 days ago
ci	DEPS: bump pyarrow version to 0.17.0 #38870 (#41476)	3 days ago
doc	Deprecate passing args as positional in DataFrame/Series.interpolate ...	16 hours ago
flake8	CI add end-of-file-fixer (#36826)	8 months ago
pandas	TST: Add tests for old issues 2 (#41493)	16 hours ago
scripts	CI: Fix changed flake8 error message after upgrade (#41462)	6 days ago
web	WEB: Fix maintainers grid not displaying correctly (GH41438) (#41447)	5 days ago
.devcontainer.json	BUILD: make tests discoverable in .devcontainer.json (#34929)	11 months ago

Flexible and powerful data analysis / manipulation library for Python, providing labeled data structures similar to R data.frame objects, statistical functions, and much more

pandas.pydata.org

python | flexible | pandas | alignment | data-analysis

BSD-3-Clause License

Pandas 1.2.4 Latest
on Apr 12

Sponsor this project

numfocus NumFOCUS

¿Quién ya tiene configurado su git y github?

Un poco de contexto, y porqué es importante compartir en git/github

https://drivendata.github.io/cookiecutter-data-science/

Contribuyendo a Pandas

pandas-dev / pandas ✓

Sponsor | Unwatch | Unstar 29.8k | Fork 12.4k

Code | Issues 3.5k | Pull requests 198 | Actions | Projects 6 | Wiki | Releases 131 | ...

master | 15 branches | 131 tags

mroeschke TST: Add tests for old issues 2 (#41493) ✓ a246270 16 hours ago 26,831 commits

.github	DEPS: bump pyarrow version to 0.17.0 #38870 (#41476)	3 days ago
LICENSES	DEPS/CLN: remove distutils usage (#41207)	14 days ago
asv_bench	[ArrowStringArray] PERF: use pa.compute.match_substring_regex for ...	2 days ago
ci	DEPS: bump pyarrow version to 0.17.0 #38870 (#41476)	3 days ago
doc	Deprecate passing args as positional in DataFrame/Series.interpolate ...	16 hours ago
flake8	CI add end-of-file-fixer (#36826)	8 months ago
pandas	TST: Add tests for old issues 2 (#41493)	16 hours ago
scripts	CI: Fix changed flake8 error message after upgrade (#41462)	6 days ago
web	WEB: Fix maintainers grid not displaying correctly (GH41438) (#41447)	5 days ago
.devcontainer.json	BUILD: make tests discoverable in .devcontainer.json (#34929)	11 months ago

Flexible and powerful data analysis / manipulation library for Python, providing labeled data structures similar to R data.frame objects, statistical functions, and much more

pandas.pydata.org

python | flexible | pandas | alignment | data-analysis

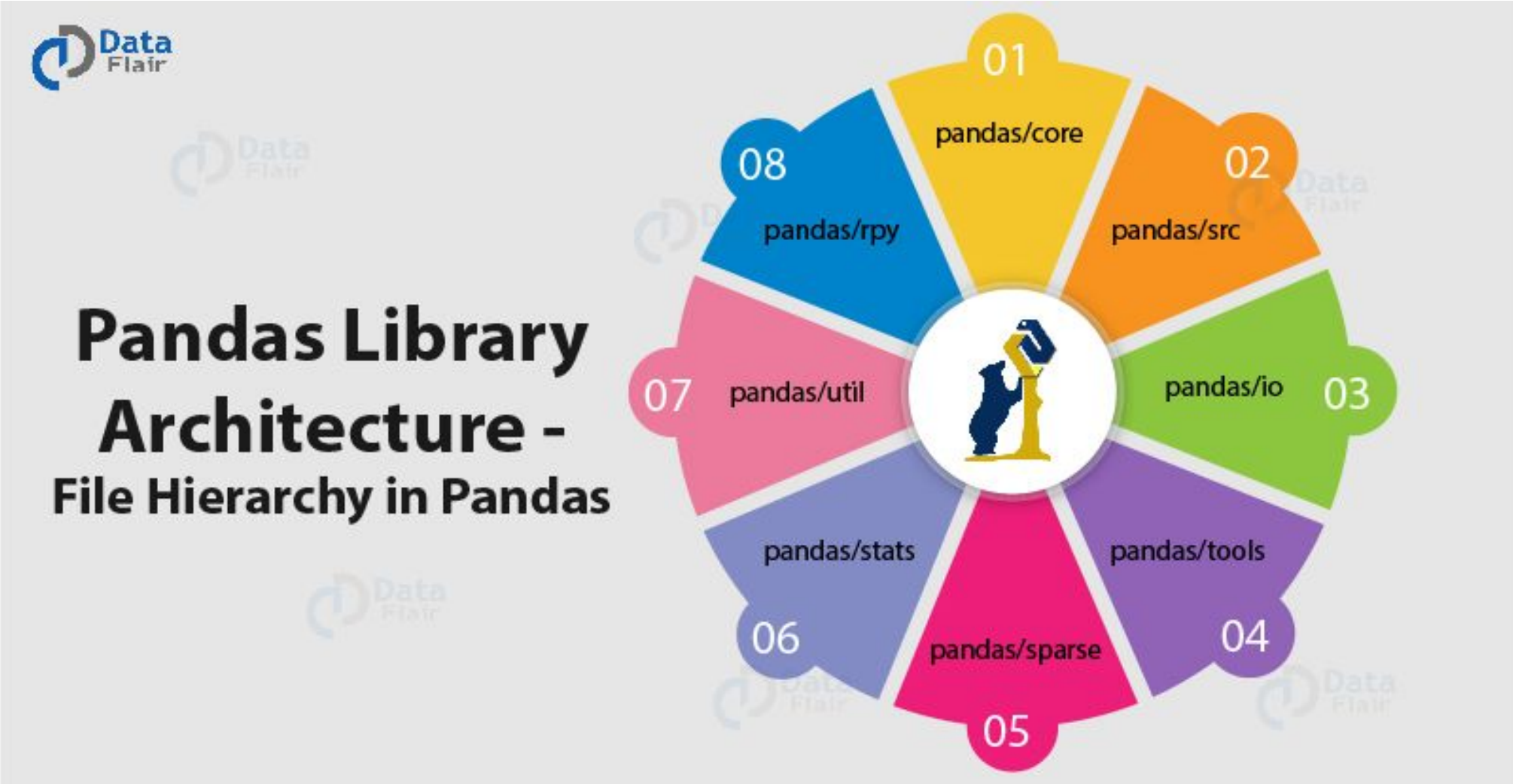
BSD-3-Clause License

Pandas 1.2.4 Latest
on Apr 12

Sponsor this project

numfocus NumFOCUS

Pandas

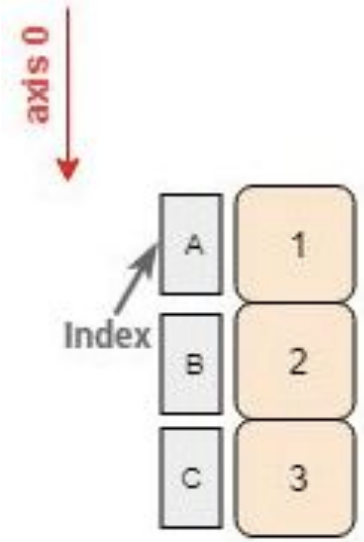

Data Flair
Pandas Library Architecture -
File Hierarchy in Pandas
01
pandas/core
02
pandas/src
03
pandas/io
04
pandas/tools
05
pandas/sparse
06
pandas/stats
07
pandas/util
08
pandas/rpy

Otra forma de ver la documentación

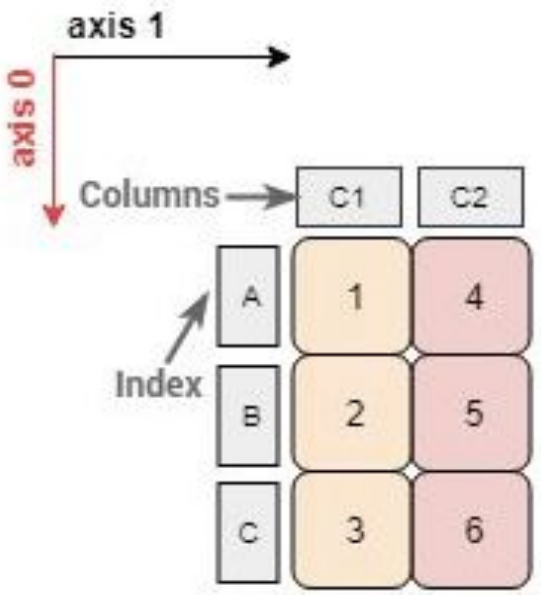
Search the docs ...

Input/output
General functions
Series
DataFrame
pandas arrays
Index objects
Date offsets
Window
GroupBy
Resampling
Style
Plotting
General utility functions
Extensions

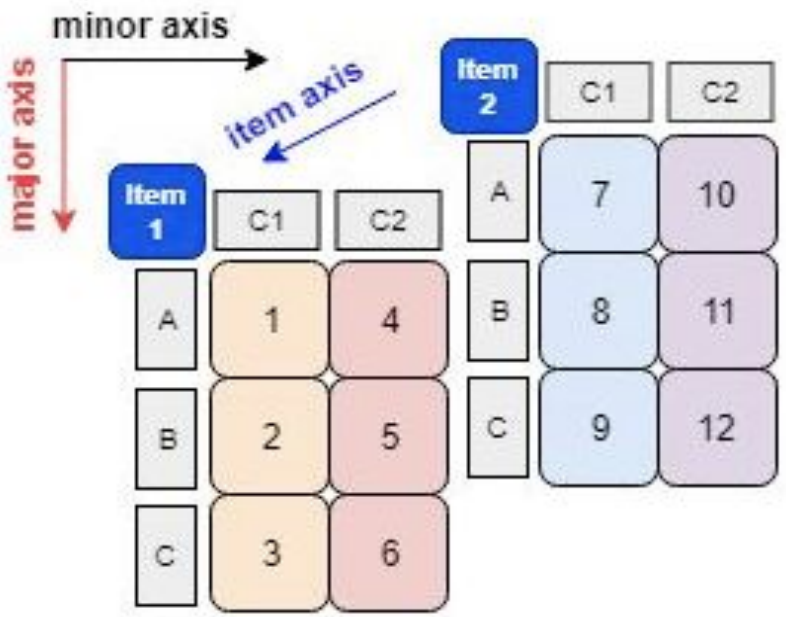
Lo más importante: Las estructuras de datos



Series



DataFrame



Panel

Series

a. Es un arreglo con índices nombrados

Constructing Series from a dictionary with an Index specified

```
>>> d = {'a': 1, 'b': 2, 'c': 3}
>>> ser = pd.Series(data=d, index=['a', 'b', 'c'])
>>> ser
a   1
b   2
c   3
dtype: int64
```

The keys of the dictionary match with the Index values, hence the Index values have no effect.

```
>>> d = {'a': 1, 'b': 2, 'c': 3}
>>> ser = pd.Series(data=d, index=['x', 'y', 'z'])
>>> ser
x   NaN
y   NaN
z   NaN
dtype: float64
```

DataFrame

Estructura de datos de dos dimensiones (La data está acomodada en una forma tabular de filas y columnas)

Características:

a. Las columnas son de tipos variados
b. Tamaño mutable
c. Los ejes están etiquetados (filas y columnas)
d. Se pueden realizar operaciones en filas y columnas

Estructura

Columns

rows

Regd. No	Name	Marks%
1000	Steve	86.29
1001	Mathew	91.63
1002	Jose	72.90
1003	Patty	69.23
1004	Vin	88.30

pandas.DataFrame

pandas.DataFrame

class pandas.**DataFrame**(*data=None, index=None, columns=None, dtype=None, copy=False*) [source]

Two-dimensional, size-mutable, potentially heterogeneous tabular data.

Data structure also contains labeled axes (rows and columns). Arithmetic operations align on both row and column labels. Can be thought of as a dict-like container for Series objects. The primary pandas data structure.

Parameters:

data : *ndarray (structured or homogeneous), Iterable, dict, or DataFrame*

Dict can contain Series, arrays, constants, dataclass or list-like objects. If data is a dict, column order follows insertion-order. *Changed in version 0.25.0:* If data is a list of dicts, column order follows insertion-order.

index : *Index or array-like*

Index to use for resulting frame. Will default to RangeIndex if no indexing information part of input data and no index provided.

columns : *Index or array-like*

Column labels to use for resulting frame. Will default to RangeIndex (0, 1, 2, …, n) if no column labels are provided.

dtype : *dtype, default None*

Data type to force. Only a single dtype is allowed. If None, infer.

copy : *bool, default False*

Copy data from inputs. Only affects DataFrame / 2d ndarray input.

Panel - o no more :((Deprecated)

a. Solía ser la representación multidimensional de datos, ahora **xarray**

xarray

Search the docs ...

FOR USERS

- Getting Started
- User Guide
- Gallery
- Tutorials & Videos
- API Reference
- How do I ...
- Ecosystem

FOR DEVELOPERS/CONTRIBUTORS

- Contributing Guide
- Xarray Internals
- Development Roadmap

Contents

xarray: N-D labeled arrays and datasets in Python

xarray (formerly **xray**) is an open source project and Python package that makes working with labelled multi-dimensional arrays simple, efficient, and fun!

Xarray introduces labels in the form of dimensions, coordinates and attributes on top of raw NumPy-like arrays, which allows for a more intuitive, more concise, and less error-prone developer experience. The package includes a large and growing library of domain-agnostic functions for advanced analytics and visualization with these data structures.

Xarray is inspired by and borrows heavily from pandas, the popular data analysis package focused on labelled tabular data. It is particularly tailored to working with netCDF files, which were the source of xarray's data model, and integrates tightly with dask for parallel computing.

Get in touch

- If you have a question like "How do I concatenate a list of datasets?", ask on GitHub discussions or StackOverflow. Please include a self-contained reproducible example if possible.
- Report bugs, suggest features or view the source code on GitHub.
- For less well defined questions or ideas, or to announce other projects of interest to xarray users, use GitHub discussions or the mailing list.

Pivot table

a. Estamos acostumbrados a verlo como un objeto (Excel), en Pandas es una operación

pandas.DataFrame.pivot

`DataFrame.pivot(index=None, columns=None, values=None)` [source]

Return reshaped DataFrame organized by given index / column values.

Reshape data (produce a "pivot" table) based on column values. Uses unique values from specified *index / columns* to form axes of the resulting DataFrame. This function does not support data aggregation, multiple values will result in a MultiIndex in the columns. See the User Guide for more on reshaping.

Parameters: **index** : *str or object or a list of str, optional*

Column to use to make new frame's index. If None, uses existing index.

Changed in version 1.1.0: Also accept list of index names.

columns : *str or object or a list of str*

Column to use to make new frame's columns.

Changed in version 1.1.0: Also accept list of columns names.

¿Preguntas?

Referencias

- “SciPy.org.” scipy.org. scipy.org
- “pandas documentation.”pandas, 12 Sep, 2021. pandas.pydata.org/docs/
- Willems. Karlijin, “Pandas Cheat Sheet for Data Science in Python.” Datacamp. 17 May, 2021. www.datacamp.com/community/blog/python-pandas-cheat-sheet.

Contacto

Mtro. Ricardo Daniel Alanis Tamez

ricardo@codeandomexico.org

LinkedIn: Ricardo Alanís