Programming

Tabular Data Analysis

```
"""
if extrapolate is None:
    extrapolate = self.extrapolate
x = np.asarray(x)
x_shape, x_ndim = x.shape, x.ndim
x = np.ascontiguousarray(x.ravel(), dtype=np

# With periodic extrapolation we map x to the
# [self.t[k], self.t[n]].
if extrapolate == 'periodic':
    n = self.t.size - self.k - 1
    x = self.t[self.k] + (x - self.t[self.k])

    extrapolate = False

out = np.empty((len(x), prod(self.c.shape[1:])),
self._ensure_c_contiguous()
self._evaluate(x, nu, extrapolate, out)
out = out.reshape(x_shape + self.c.shape[1:])
if self.axis != 0:
    # transpose to move the calculated values to t
    l = list(range(out.ndim))
    l = l[x_ndim:x_ndim+self.axis] + l[:x_ndim] + l
    out = out.transpose(l)
return out

def _evaluate(self, xp, nu, extrapolate, out):
    _bspl.evaluate_spline(self.t, self.c.reshape(self.c.s
        self.k, xp, nu, extrapolate, out)

def _ensure_c_contiguous(self):
    """
    c and t may be modified by the user. The Cython code e
    that they are C contiguous.
```

Luna Pianesi

Faculty of Technology, Bielefeld University

UNIVERSITÄT
BIELEFELD

Recap

- Data visualization
  - Matplotlib
  - Whisker plots, line/scatter plots, and histograms
  - NASA’s GISS surface temperature analysis
  - Linear regression
- Numerical data analysis with NumPy
  - N-dimensional arrays
  - Vectorized operations and broadcasting

Tabular data analysis

Pandas Series

Pandas DataFrame

Multi-indexing DataFrames

What is tabular data analysis?

Tabular data is data that comes in the form of ***tables***, meaning it has some rows and some columns.
Analyzing tabular data means to take into account the structure of tabular data, and try to find dependencies among the columns or rows. We want to ***inspect, cleanse, transform and model*** data with the usual goal of discovering useful information.

Source: `https://tabular-data-analysis.github.io/tada2024/`

Structured Data: tables

Extract from file “books.tsv”

title	*isbn*	*pageCount*	*publishedDate*	*authors*	*categories*
Unlocking Android	1933988673	416	2009-04-01	W. Frank Ableson, Charlie Collins, Robi Sen	Open Source, Mobile
Specification by Example	1617290084	-	2011-06-03	Gojko Adzic	Software Engineering
Flex 4 in Action	1935182420	600	2010-11-15	Tariq Ahmed, Dan Orlando, John C. Bland II, Joel Hooks	Internet
Zend Framework in Action	1933988320	432	2008-12-01	Rob Allen, Nick Lo, Steven Brown	Web Development
Flex on Java	1933988797	265	2010-10-15	Bernerd Allmon, Jeremy Anderson	Internet
Griffon in Action	1935182234	375	2012-06-04	Andres Almiray, Danno Ferrin, , James Shingler	Java
OSGi in Depth	193518217X	325	2011-12-12	Alexandre de Castro Alves	Java
Flexible Rails	1933988509	592	2008-01-01	Peter Armstrong	Web Development
Hello! Flex 4	1933988762	258	2009-11-01	Peter Armstrong	Internet
Coffeehouse	1884777384	316	1997-07-01	Levi Asher, Christian Crumlish	Miscellaneous
MongoDB in Action	1935182870	-	2011-12-12	Kyle Banker	Next Generation Databases
Taming Jaguar	1884777686	362	2000-07-01	Michael J. Barlotta, Jason R. Weiss	PowerBuilder
Hibernate in Action	193239415X	400	2004-08-01	Christian Bauer, Gavin King	Java
Java Persistence with Hibernate	1932394885	880	2006-11-01	Christian Bauer, Gavin King	Java
JSTL in Action	1930110529	480	2002-07-01	Shawn Bayern	Internet
iBATIS in Action	1932394826	384	2007-01-01	Clinton Begin, Brandon Goodin, Larry Meadors	Web Development
Designing Hard Software	133046192	350	1997-02-01	Douglas W. Bennett	Object-Oriented Programming
Hibernate Search in Action	1933988649	488	2008-12-21	Emmanuel Bernard, John Griffin	Java
...					

Structured data: tables

Reading tables using the `csv` module

```python
import csv

f = open('books.tsv')
table = list()

for row in csv.reader(f, delimiter = '\t'):

    # ignore rows that are empty or start with '#'
    if not row or row[0].startswith('#'):
        continue

    table.append(row)

# print first row of table
print(table[0])
```

Pandas

Pandas is a library built on top of Python and NumPy that allows easy, fast, and complex analyses of tabular data.
Pandas is very powerful and flexible, and has different useful features than NumPy.

Source: `https://www.nvidia.com/en-us/glossary/pandas-python/`

Pandas

The name Pandas comes from the econometric term “panel data”, which describes data sets that include observations over multiple time periods.

Image: `https://www.abc.net.au/news/2016-09-30/panda-cubs-make-debut/7892968?WT.mc_id=newsmail`

Pandas data structures

Series

- Container for scalar values
- 1D array
- More powerful than a 1D NumPy array
- Allows to freely set index
- Size immutable

DataFrame

- Container for Series
- 2D array/table
- Mutability
  - Rows are immutable
  - Allows insertion of new columns

Tabular data analysis

Pandas Series

Pandas DataFrame

Multi-indexing DataFrames

Pandas Series data structure

See Jupyter Notebook!

- Fancier than NumPy 1D arrays
- Useful to work with different types of indexes

Tabular data analysis

Pandas Series

Pandas DataFrame

Multi-indexing DataFrames

Pandas DataFrame data structure

See Jupyter Notebook!

- Fancier than NumPy 2D arrays
- Container for Series

Tabular data analysis

Pandas Series

Pandas DataFrame

Multi-indexing DataFrames

Multi-indexing DataFrames

See Jupyter Notebook!

- Useful to work with multiple indexes
- A lot of flexibility and new functionalities

Recap

Summary

Tabular data analysis with Pandas:

- Series
  - Creating & indexing
  - Accessing elements and subsets
- DataFrame
  - Creating & indexing
  - Accessing columns, rows, and elements
  - Broadcasting and vectorized operations
  - Reading and writing tables
- Multi-indexing
  - Creating multi-indeces
  - Slicing, grouping, and masking

What comes next?

- Play with census data using Pandas
- Due date for this week’s exercises is ***Wednesday, December 18, 2 pm, 2024***.

Next lecture: Machine Learning ...