Plotting and Programming in Python

Software Carpentry @ Dartmouth

Simon Stone

Dartmouth Library, Research Data Services

About Research Data Services

Research Data Management

Data Management Plans (DMPs) for sponsored projects

Finding and using 3rd party data

Collection and cleaning of data

Organization and documentation

Publishing and Repositories

Data Analysis/Visualization

Textual, numeric, spatial data

Reproducible research workflows

Scripting in R: tidyverse core package (i.e. ggplot, dplyr, tydr, tibble, etc.)

Scripting in Python: NumPy, SciPy, Pandas, Scikit-learn, Matplotlib, Seaborn, (OpenCV, PyTorch, TensorFlow, Tesseract, NLTK, etc.)

Computational Scholarship

Computational project planning

Collections as Data

Storytelling with data and visualizations

Text and data mining

Digital Humanities support

Computational Pedagogy

Work with us

ResearchDataHelp@groups.dartmouth.edu

Jeremy Mikecz
Research Data Science Specialist
jeremy.m.mikecz@dartmouth.edu
dartgo.org/jeremyappts

Simon Stone
Research Data Science Specialist
simon.stone@dartmouth.edu
dartgo.org/meetwithsimon

Lora Leligdon
Head of Research Data Services
lora.c.leligdon@dartmouth.edu
dartgo.org/lora

Agenda

- 🐍 Built-in Functions and Help
- 📚 Libraries
- # Reading Tabular Data
- 🐼 Pandas DataFrames
- 📈 Plotting
- 🔁 Looping
- 🌈 Programming Style

Built-in Functions and Help

Built-in Functions					
A abs() aiter() all() any() anext() ascii() B bin() bool() breakpoint() bytearray() bytes() C callable() chr() classmethod() compile() complex()	D delattr() dict() dir() divmod() E enumerate() eval() exec() F filter() float() format() frozenset() G getattr() globals()	H hasattr() hash() help() hex() I id() input() int() isinstance() issubclass() iter() L len() list() locals()	M map() max() memoryview() min() N next() O object() oct() open() ord() P pow() print() property()	R range() repr() reversed() round() S set() setattr() slice() sorted() staticmethod() str() sum() super() T tuple() type()	V vars() Z zip() _ __import__()

Built-in Functions and Help

Hands-on

Libraries

- 💪 Most of the power of a programming language is in its libraries
- 📚 A **library** is a collection of files (called **modules**) that contains functions for use by other programs
- ε May also contain data values (e.g., numerical constants) and other things
- 🍱 Library's contents are supposed to be related
- 🐍 The Python standard library is an extensive suite of modules that comes with Python itself ("batteries included")
- 🌎 Many additional libraries are available from PyPI (the Python Package Index)

Libraries

Hands-on

Reading Tabular Data

- Tabular data is often stored in **C**omma-**S**eparated **V**alue files
- Pandas is a widely-used Python library for statistics, particularly on tabular data
- Reads CSV files and imports them into a structure called DataFrame

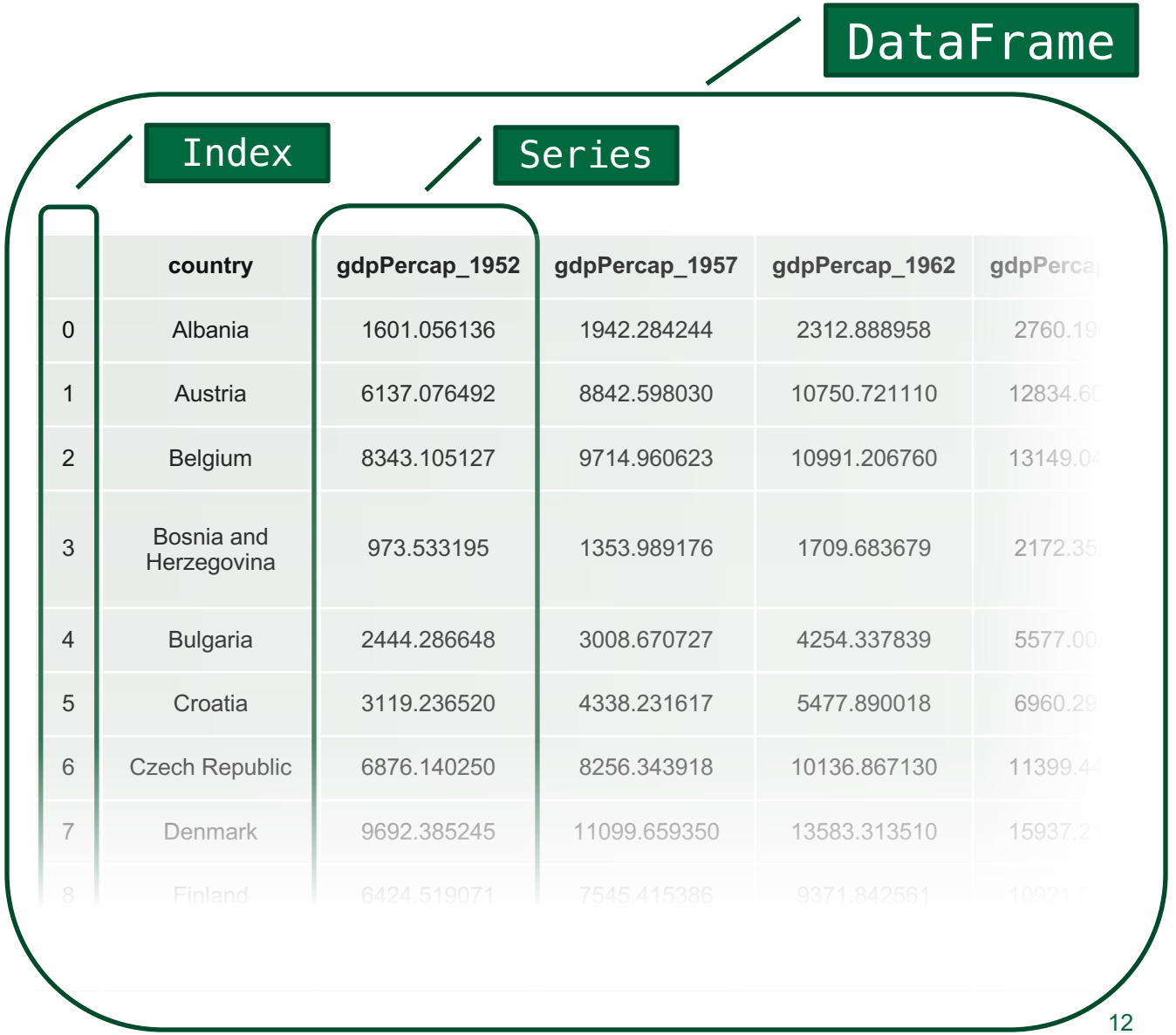
```
country,gdpPercap_1952,gdpPercap_1957,gdpPercap_1962,gdpP
Albania,1601.056136,1942.284244,2312.888958,2760.196931,3
Austria,6137.076492,8842.59803,10750.72111,12834.6024,166
Belgium,8343.105127,9714.960623,10991.20676,13149.04119,1
Bosnia and Herzegovina,973.5331948,1353.989176,1709.68367
Bulgaria,2444.286648,3008.670727,4254.337839,5577.0028,65
Croatia,3119.23652,4338.231617,5477.890018,6960.297861,91
Czech Republic,6876.14025,8256.343918,10136.86713,11399.4
Denmark,9692.385245,11099.65935,13583.31351,15937.21123,1
Finland,6424.519071,7545.415386,9371.842561,10921.63626,1
France,7029.809327,8662.834898,10560.48553,12999.91766,16
Germany,7144.114393,10187.82665,12902.46291,14745.62561,1
Greece,3530.690067,4916.299889,6017.190733,8513.097016,12
Hungary,5263.673816,6040.180011,7550.359877,9326.64467,10
Iceland,7267.688428,9244.001412,10350.15906,13319.89568,1
Ireland,5210.280328,5599.077872,6631.597314,7655.568963,9
Italy,4931.404155,6248.656232,8243.58234,10022.40131,1226
Montenegro,2647.585601,3682.259903,4649.593785,5907.85093
```

Reading Tabular Data

Hands-on

Pandas DataFrames

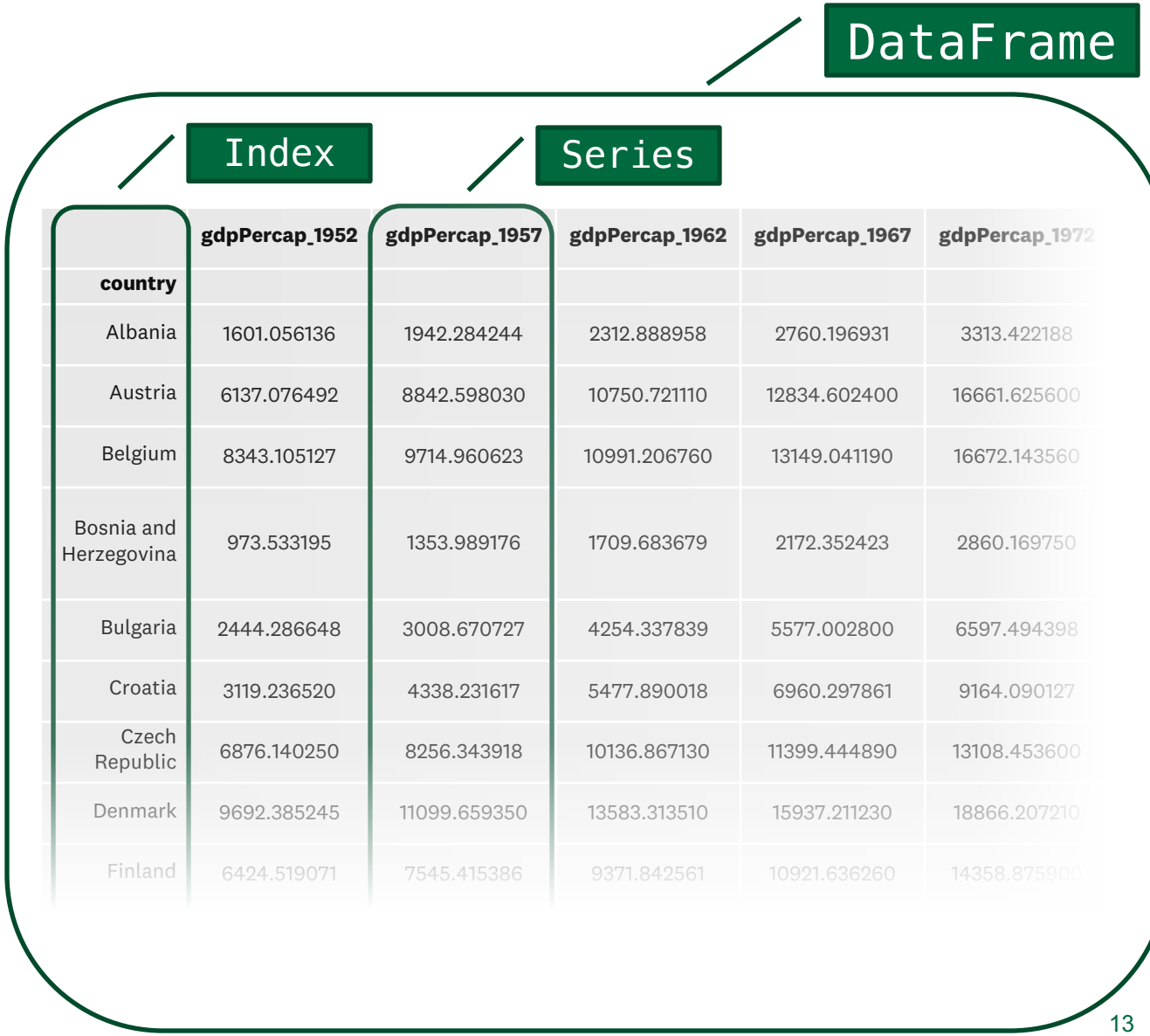
- A `DataFrame` represents a 2D table as a collection of columns
- Each column is represented by a `Series`
- Every `Series` in a `DataFrame` uses the same `Index`
- Try to use a meaningful `Index`



	country	gdpPercap_1952	gdpPercap_1957	gdpPercap_1962	gdpPerca
0	Albania	1601.056136	1942.284244	2312.888958	2760.19
1	Austria	6137.076492	8842.598030	10750.721110	12834.6
2	Belgium	8343.105127	9714.960623	10991.206760	13149.0
3	Bosnia and Herzegovina	973.533195	1353.989176	1709.683679	2172.35
4	Bulgaria	2444.286648	3008.670727	4254.337839	5577.00
5	Croatia	3119.236520	4338.231617	5477.890018	6960.29
6	Czech Republic	6876.140250	8256.343918	10136.867130	11399.4
7	Denmark	9692.385245	11099.659350	13583.313510	15937.2
8	Finland	6424.519071	7545.415386	9371.842561	10921.6

Pandas DataFrames

- A `DataFrame` represents a 2D table as a collection of columns
- Each column is represented by a `Series`
- Every `Series` in a `DataFrame` uses the same `Index`
- Try to use a meaningful `Index`



country	gdpPercap_1952	gdpPercap_1957	gdpPercap_1962	gdpPercap_1967	gdpPercap_1972
Albania	1601.056136	1942.284244	2312.888958	2760.196931	3313.422188
Austria	6137.076492	8842.598030	10750.721110	12834.602400	16661.625600
Belgium	8343.105127	9714.960623	10991.206760	13149.041190	16672.143560
Bosnia and Herzegovina	973.533195	1353.989176	1709.683679	2172.352423	2860.169750
Bulgaria	2444.286648	3008.670727	4254.337839	5577.002800	6597.494398
Croatia	3119.236520	4338.231617	5477.890018	6960.297861	9164.090127
Czech Republic	6876.140250	8256.343918	10136.867130	11399.444890	13108.453600
Denmark	9692.385245	11099.659350	13583.313510	15937.211230	18866.207210
Finland	6424.519071	7545.415386	9371.842561	10921.636260	14358.875900

Pandas DataFrames

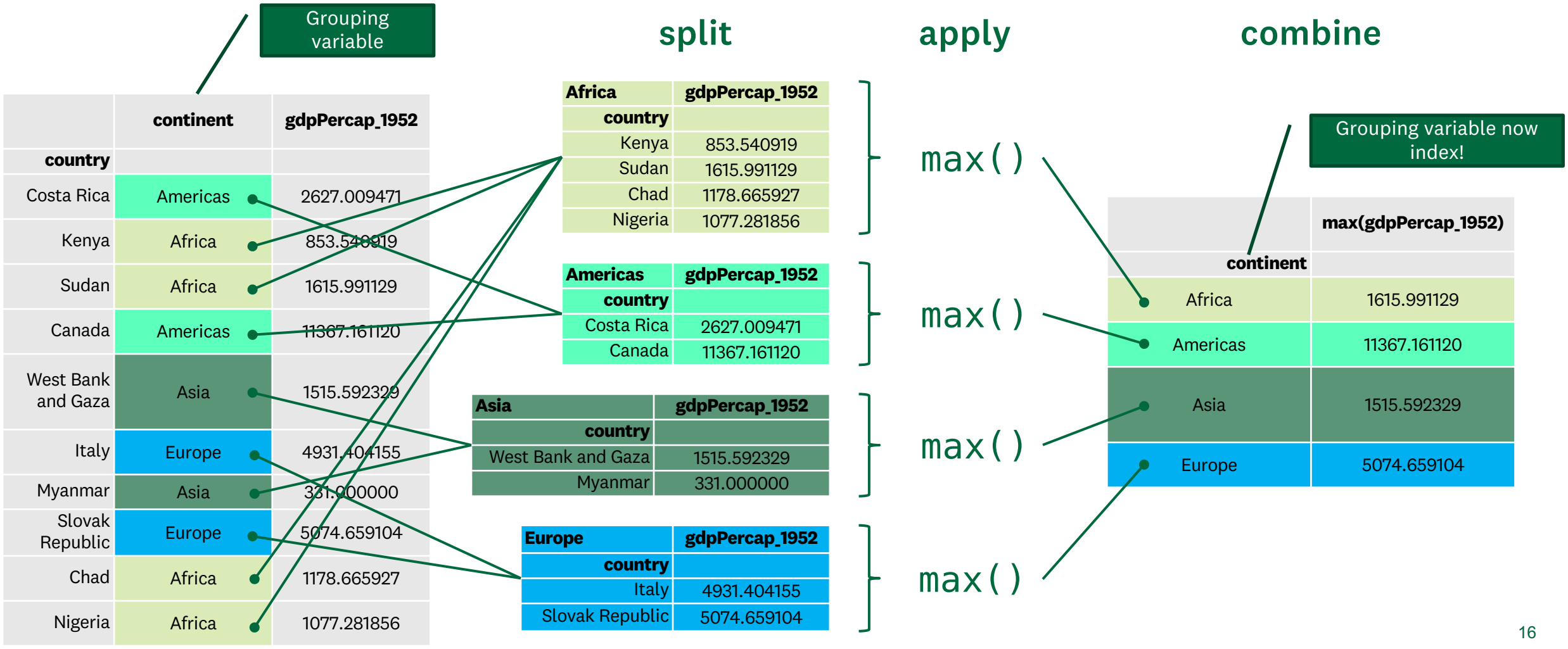
Hands-on

Pandas DataFrames: split-apply-combine

- We often want to do apply operations to subgroups within a dataset
- For example:
  Find the maximum GDP in 1952 on **each** continent
- We can use a grouping variable to
  - Split the dataset into groups
  - Apply our operation
  - Combine the results into a new table

	continent	gdpPercap_1952
country		
Costa Rica	Americas	2627.009471
Kenya	Africa	853.540919
Sudan	Africa	1615.991129
Canada	Americas	11367.161120
West Bank and Gaza	Asia	1515.592329
Italy	Europe	4931.404155
Myanmar	Asia	331.000000
Slovak Republic	Europe	5074.659104
Chad	Africa	1178.665927
Nigeria	Africa	1077.281856

Pandas DataFrames: split-apply-combine



Pandas DataFrames

Hands-on

Plotting

📊 Visualizing data is important for both exploration and reporting

- There are many ways to do this in Python:
  - 🐼 `Pandas` has built-in plotting functionality
  - 🌊 `Seaborn` is a library specializing in statistical graphs based on DataFrames
  - 🖌️ "under the hood" both use the `matplotlib` library
    - General purpose plotting library
    - Interface modeled after MATLAB

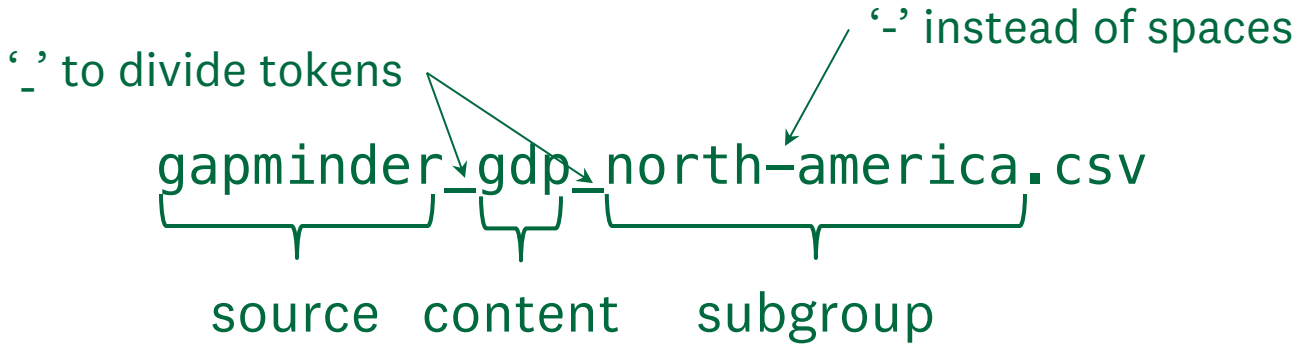
Plotting

Hands-on

Looping over multiple datasets

☝ Processing a set of files is easiest if you follow some conventions:

- 📦 Put all files in the same format (e.g., CSV files)
- 📂 Store them all in the same folder but separately from non-data files
- 🏷 Use a systematic naming scheme, e.g.:



Looping

Hands-on

Programming style

```python
def my_sum(n):
    temp1 = 0
    for temp2 in range(n+1):
        if temp2%2 == 0:
            temp1 += temp2
    return temp2
```

Which one would you rather work with?

```python
def sum_even_nums(n: int) -> int:
    """
    This function calculates the sum of even numbers
    between 0 and a given integer (inclusive).

    Args:
        n (int): an integer value for the upper limit
        of the range to sum

    Returns:
        int: the sum of even numbers between 0 and n
    """
    total = 0
    for num in range(n + 1):
        if num % 2 == 0:
            total += num
    return total
```

Examples generated using ChatGPT

Programming Style

Hands-on

Wrap-up

- How can I use built-in functions and libraries in Python?
- How can I load and manipulate tabular data?
- How can I plot data visualizations?
- How can I process data from several files in a loop?
- What are some good practices around programming style?
- Contact us:
  - Research Computing: rc.dartmouth.edu
  - Research Data Services: www.dartgo.org/rds
- Post-workshop survey: www.dartgo.org/post-carpentry

Thank you