**Cheat Sheet: The pandas DataFrame**

# Cheat Sheet: The Pandas DataFrame

## Preliminaries

Start by importing these Python modules:

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt # for charts

Check which version of pandas you are using:

print(pd.\_\_version\_\_)

This cheat sheet was written for pandas version 0.25. It assumes you are using Python 3.

## The Conceptual Model

Pandas provides two important data types: the DataFrame and the Series.

* **DataFrame:** A two-dimensional table of data with column and row indexes, similar to a spreadsheet. Columns are made up of Series objects. A DataFrame has two indexes:
  + Column index (df\_cols): typically a list of strings (variable names) or integers.
  + Row index (df\_idx): can be integers, strings (for case names), or DateTimeIndex/PeriodIndex for time series data.
* **Series:** An ordered, one-dimensional array of data with an index. All data in a Series is of the same data type. Series arithmetic is vectorized after aligning the Series index for each operand.

### Examples of Series Arithmetic:

srs1 = pd.Series(range(0, 4)) # 0, 1, 2, 3

srs2 = pd.Series(range(1, 5)) # 1, 2, 3, 4

srs3 = srs1 + srs2 # 1, 3, 5, 7

srs4 = pd.Series([1, 2, 3], index=[0, 1, 2])

srs5 = pd.Series([1, 2, 3], index=[2, 1, 0])

srs6 = srs4 + srs5 # 4, 4, 4

srs7 = pd.Series([1, 2, 3], index=[1, 2, 3])

srs8 = pd.Series([1, 2, 3], index=[0, 1, 2])

srs9 = srs7 + srs8 # NAN, 3, 5, NAN

## Get Your Data into a DataFrame

### Instantiate a DataFrame:

df = pd.DataFrame() # the empty DataFrame

df = pd.DataFrame(python\_dict)

df = pd.DataFrame(np\_matrix)

### Load a DataFrame from a CSV File:

df = pd.read\_csv('data\_file.csv', header=0, index\_col=0, quotechar='"', sep=':', na\_values=['na', '-', '.', ''])

### Get Your Data from Inline Python CSV Text:

from io import StringIO

data = """, Animal, Cuteness, Desirable

A, dog, 8.7, True

B, cat, 9.5, False"""

df = pd.read\_csv(StringIO(data), header=0, index\_col=0, skipinitialspace=True)

### Other Loading Options:

df = pd.read\_html(url/html\_str)

df = pd.read\_json(json\_path/json\_str)

df = pd.read\_sql(sql\_query, db\_connection)

df = pd.read\_excel('excelfilename.xlsx')

df\_copy = pd.read\_clipboard() # e.g., from ExcelFilename.xlsx copy

## Fake Up Some Random Data – Useful for Testing

df\_rand = (pd.DataFrame(np.random.rand(1100, 6), columns=list('ABCDEF')) - 0.5).cumsum()

df\_rand['Group'] = [np.random.choice(list('abcd')) for \_ in range(len(df\_rand))]

df\_rand['Date'] = pd.date\_range('1/1/2017', periods=len(df\_rand), freq='D')

## Saving a DataFrame

### Saving to a CSV File:

df.to\_csv('output\_file.csv', encoding='utf-8')

### Saving to an Excel Workbook:

writer = pd.ExcelWriter('output\_file.xlsx')

df.to\_excel(writer, 'Sheet1')

writer.save()

### Saving to a Python Object:

dict\_data = df.to\_dict() # to dictionary

numpy\_data = df.values # to a numpy matrix

html\_str = df.to\_html()

df.to\_json()

df.to\_sql()

df.to\_clipboard() # then paste into Excel

This cheat sheet covers essential operations and concepts related to pandas DataFrames. Whether you're loading data, performing calculations, or saving results, pandas provides powerful tools for working with tabular data efficiently in Python.