



# Pandas Cheat Sheet: Data Wrangling in Python

March 8th, 2017 in Python



Karlijn Willems



By now, you'll already know the [Pandas](#) library is one of the most preferred tools for data manipulation and analysis, and you'll have explored the fast, flexible, and expressive Pandas data structures, maybe with the help of DataCamp's [Pandas Basics cheat sheet](#).

Yet, there is still much functionality that is built into this package to explore, especially when you get hands-on with the data: you'll need to reshape or rearrange your data, iterate over DataFrames, visualize your data, and much more. And this might be even more difficult than "just" mastering the basics.

That's why today's post introduces a new, more advanced Pandas cheat sheet.

(Do you want to learn more? Start our [Pandas Foundations](#) course for free now or try out our [Pandas](#)

learners and start one of our interactive tutorials today!

[Learn R](#)

[Learn Python](#)

**Python For Data Science Cheat Sheet**  
Pandas  
Learn Python for Data Science interactively at [datacamp.com](#)

**Reshaping Data**  
**Pivot**  
Spread rows into columns  
`>>> df3 = df2.pivot(index="Date", columns="Type", values="Value")`

**Pivot Table**  
Spread rows into columns  
`>>> df4 = pd.pivot_table(df2,`

**Advanced Indexing**  
**Selecting**  
`>>> df2.loc[:, df2>1].any()`  
`>>> df2.loc[:, df2>1].all()`  
`>>> df2.loc[:, df2.isnull().any()]`  
`>>> df2.loc[:, df2.isnull().all()]`  
**Indexing With in**  
`>>> df1[df1.Country.isin(df2.Type)]`  
`>>> df2.iloc[1:(df2.Type=="a").any()]`  
`>>> df2.ix[df2.Type=="a", "a"]`  
**Where**  
`>>> a.where(a > 0)`  
**Query**  
`>>> df2.query("a > 0")`  
**Setting/Resetting Index**  
`>>> df.set_index("Country")`  
`>>> df = df.reset_index()`  
`>>> df = df.reset_index(drop=True)`  
**Reindexing**  
`>>> a2 = a.reindex(["a", "b", "a", "a", "b"])`

**Combining Data**  
**Merge**  
`>>> pd.merge(data1, data2, how="left", on="X")`  
`>>> pd.merge(data1, data2, how="right", on="X")`  
`>>> pd.merge(data1, data2, how="inner", on="X")`

**Melt**  
Gather columns into rows  
`>>> pd.melt(df2, id_vars="Date", value_vars=["Type", "Value"], var_name="Measure", value_name="Value")`

**Duplicate Data**  
`>>> a3.unique()`  
`>>> df2.duplicated("Type")`  
`>>> df2.drop_duplicates("Type", keep="last")`  
`>>> df1.index.duplicated()`

**Grouping Data**  
**Aggregation**  
`>>> df2.groupby(["Date", "Type"]).mean()`  
`>>> df4.groupby(level=0).sum()`  
`>>> df4.groupby(level=0).agg("a": lambda x: sum(x)/len(x), "b": np.max)`  
**Transformation**  
`>>> df.groupby(level=0).transform(lambda x: x**2)`  
`>>> df4.groupby(level=0).transform(coutinho)`

**Iteration**  
Column/index, Series pairs  
Row/index, Series pairs  
`>>> df.iterrows()`  
`>>> df.itercolumns()`

**Missing Data**  
Drop Null values  
Fill Null values with a predetermined value  
Replace values with others  
`>>> df.dropna()`  
`>>> df2.fillna(df2.mean())`  
`>>> df2.replace("a", "b")`

**Dates**  
**Horizontal/Vertical**  
`>>> pd.concat([a2, a1], axis=1, keys=["Date", "Two"])`  
`>>> pd.concat([data1, data2], axis=0, join="inner")`  
`>>> df2["Date"] = pd.to_datetime(df2["Date"])`  
`>>> df2["Date"] = pd.date_range("2000-1-1", periods=5, freq="D")`  
`>>> dates = [datetime(2012, 5, 1), datetime(2012, 5, 2)]`  
`>>> index = pd.date_range(dates, freq="BMS")`

**Visualization**  
Also see Matplotlib  
`>>> import matplotlib.pyplot as plt`  
`>>> a.plot()`  
`>>> df2.plot()`  
`>>> plt.show()`

**Check Out Pandas Cheat Sheet**

The Pandas cheat sheet will guide you through some more advanced indexing techniques, DataFrame iteration, handling missing values or duplicate data, grouping and combining data, data functionality, and data visualization.

In short, everything that you need to complete your data manipulation with Python!

Don't miss out on our other cheat sheets for data science that cover [Matplotlib](#), [SciPy](#), [Numpy](#), and the [Python basics](#).

---

learners and start one of our interactive tutorials today!

## Up Next

[Learn R](#)

[Learn Python](#)



### New Course! Supervised Learning in R: Classification

R Programming

73 views

This beginner-level introduction to machine learning covers four of the most common classification algorithms. You will come away with a b...

September 27th, 2017 in Blog

---



learners and start one of our interactive tutorials today!

## New Course: Case Studies in Statistical Thinking!

[Learn R](#)

[Learn Python](#)

Python

1,135 views

Hone your applied data science skills in Python by doing a variety of case studies across multiple disciplines. Use data science to solve ...

September 20th, 2017 in Blog



## Jupyter Notebook Cheat Sheet

Python

19,126 views

This Jupyter Notebook cheat sheet will help you to find your way around the well-known Jupyter Notebook App, a subproject of Project Jupyter...

September 19th, 2017 in Blog