



Data Elixir is a weekly newsletter of curated data science news and resources from around the web.

Free for data lovers.

[CLICK HERE TO SUBSCRIBE - FREE](#)

No spam, ever. We'll never share your email address and you can opt out at any time.

Data Elixir recommended 454 articles, tutorials and resources in 2016, from hundreds of writers and publications.

These were our favorites.

Business

[A Guide to Building a High Functioning Data Science Department](#)

MultiThreaded, Stitch Fix • March 2016

Insightful article about common problems faced by teams of data scientists and engineers and how Stitch Fix managed to build A Better Way to structure a data science department.

[Doing Data Science Right — Your Most Common Questions Answered](#)

First Round • April 2016

Building a data science team? Jeremy Stanley and Daniel Tunkelang discuss why, when, where and how.

[The Competitive Landscape for Machine Intelligence](#)

Harvard Business Review • November 2016

Shivon Zilis and James Cham offer insights into how the "Stack" of machine intelligence building blocks is maturing and what all businesses need to do NOW in order to survive and outlast their competitors.

[The Great A.I. Awakening](#)

New York Times • December 2016

How Google used artificial intelligence to transform Google Translate and how machine learning is poised to reinvent computing itself.

Data Stories

SPEAK, MEMORY

The Verge • October 2016

When her best friend died, she trained a bot on hundreds of his messages and used AI to keep him alive.

How the Circle Line rogue train was caught with data

Singapore data.gov • November 2016

For months, a train line suffered from mysterious disruptions and created confusion and distress. Here's how a team of data scientists saved the day.

Tools and Techniques

Practical advice for analysis of large, complex data sets

Unofficial Google Data Science Blog • October 2016

A How-To for approaching large datasets.

Modern Pandas

Tom Augspurger • March-May 2016

Fantastic 7-part series on writing idiomatic pandas code.

28 Jupyter Notebook tips, tricks and shortcuts

Dataquest • October 2016

Nice collection of tips and tricks for working with Jupyter.

Top-down learning path: Machine Learning for Software Engineers

Nam Vu • October 2016

A complete daily plan for studying to become a machine learning engineer.

Machine learning algorithms

Artem Golubin • November 2016

Simple and clean examples of machine learning algorithms for people who want to learn how they work. All algorithms are implemented in Python, using numpy, scipy and autograd.

Approaching (Almost) Any Machine Learning Problem

Kaggle • July 2016

Python tutorial based on the author's experience with more than 100 machine learning competitions. Includes discussion of a generalized workflow and parameter optimization.

[An Introduction to Deep Learning](#)

Algorithmia • November 2016

Nice intro to deep learning, including links to online courses, free books, and popular projects.

[Neural Network Playground](#)

Daniel Smilkov and Shan Carter • April 2016

Awesome! Tinker with a real neural network in your browser.

[The Neural Network Zoo](#)

The Asimov Institute • September 2016

A "mostly complete chart of architectures." Both the diagrams and descriptions are fantastic.

Career

[2016 Data Science Salary Survey](#)

O'Reilly Media • September 2016

This is a great report that's based on a survey of nearly 1000 respondents. Free registration required.

[Building a data science portfolio](#)

Dataquest • June-August 2016

[Part 1: Storytelling with data](#)

[Part 2: Making a data science blog](#)

[Part 3: Machine learning project](#)

[Part 4: The key to building a data science portfolio that will get you a job](#)

Data Visualization

[Visualizations That Really Work](#)

Harvard Business Review • June 2016

Not long ago, the ability to create smart data visualizations was a nice-to-have management skill. Not anymore. Here's how visual communication is quickly becoming an essential skill for decision-makers.

[39 studies about human perception in 30 minutes](#)

Kennedy Elliott • May 2016

Great exploration of how visual perception affects data visualizations.



Data Science Cheat Sheet

Pandas

KEY

We'll use shorthand in this cheat sheet

df - A pandas DataFrame object

s - A pandas Series object

IMPORTS

Import these to start

```
import pandas as pd
```

```
import numpy as np
```

IMPORTING DATA

pd.read_csv(filename) - From a CSV file

pd.read_table(filename) - From a delimited text file (like TSV)

pd.read_excel(filename) - From an Excel file

pd.read_sql(query, connection_object) -

Read from a SQL table/database

pd.read_json(json_string) - Read from a JSON formatted string, URL or file.

pd.read_html(url) - Parses an html URL, string or file and extracts tables to a list of dataframes

pd.read_clipboard() - Takes the contents of your clipboard and passes it to **read_table()**

pd.DataFrame(dict) - From a dict, keys for columns names, values for data as lists

EXPORTING DATA

df.to_csv(filename) - Write to a CSV file

df.to_excel(filename) - Write to an Excel file

df.to_sql(table_name, connection_object) - Write to a SQL table

df.to_json(filename) - Write to a file in JSON format

df.to_html(filename) - Save as an HTML table

df.to_clipboard() - Write to the clipboard

CREATE TEST OBJECTS

Useful for testing

pd.DataFrame(np.random.rand(20,5)) - 5 columns and 20 rows of random floats

pd.Series(my_list) - Create a series from an iterable **my_list**

df.index = pd.date_range('1900/1/30', periods=df.shape[0]) - Add a date index

VIEWING/INSPECTING DATA

df.head(n) - First **n** rows of the DataFrame

df.tail(n) - Last **n** rows of the DataFrame

df.shape() - Number of rows and columns

df.info() - Index, Datatype and Memory information

df.describe() - Summary statistics for numerical columns

s.value_counts(dropna=False) - View unique values and counts

df.apply(pd.Series.value_counts) - Unique values and counts for all columns

SELECTION

df[col] - Return column with label **col** as Series

df[[col1, col2]] - Return Columns as a new DataFrame

s.iloc[0] - selection by position

s.loc[0] - selection by index

df.iloc[0,:] - first row

df.iloc[0,0] - first element of first column

DATA CLEANING

df.columns = ['a','b','c'] - Rename columns

pd.isnull() - Checks for null Values, Returns Boolean Array

pd.notnull() - Opposite of **s.isnull()**

df.dropna() - Drop all rows that contain null values

df.dropna(axis=1) - Drop all columns that contain null values

df.dropna(axis=1,thresh=n) - Drop all rows have less than **n** non null values

df.fillna(x) - Replace all null values with **x**

s.fillna(s.mean()) - Replace all null values with the mean (mean can be replaced with almost any function from the statistics section)

s.astype(float) - Convert the datatype of the series to float

s.replace(1, 'one') - Replace all values equal to 1 with 'one'

s.replace([1,3], ['one', 'three']) - Replace all 1 with 'one' and 3 with 'three'

df.rename(columns=lambda x: x + 1) - mass renaming of columns

df.rename(columns={'old_name': 'new_name'}) - selective renaming

df.set_index('column_one') - change the index

df.rename(index=lambda x: x + 1) - mass renaming of index

FILTER, SORT, & GROUPBY

df[df[col] > 0.5] - Rows where the **col** column is greater than 0.5

df[(df[col] > 0.5) & (df[col] < 0.7)] - Rows where 0.7 > col > 0.5

df.sort_values(col1) - Sort values by **col1** in ascending order

df.sort_values(col2,ascending=False) - Sort values by **col2** in descending order

df.sort_values([col1,col2],

ascending=[True,False]) - Sort values by **col1** in ascending order then **col2** in descending order

df.groupby(col1) - Return a groupby object for values from one column

df.groupby([col1,col2]) - Return a groupby object values from multiple columns

df.groupby(col1)[col2].mean() - Return the mean of the values in **col2**, grouped by the values in **col1** (mean can be replaced with almost any function from the statistics section)

df.pivot_table(index=col1,values=[col2,col3],aggfunc=max) - Create a pivot table that groups by **col1** and calculates the mean of **col2** and **col3**

df.groupby(col1).agg(np.mean) - find the average across all columns for every unique column 1 group

data.apply(np.mean) - apply a function across each column

data.apply(np.max, axis=1) - apply a function across each row

JOIN/COMBINE

df1.append(df2) - Add the rows in **df1** to the end of **df2** (columns should be identical)

df.concat([df1, df2],axis=1) - Add the columns in **df1** to the end of **df2** (rows should be identical)

df1.join(df2,on=col1,how='inner') - SQL-style join the columns in **df1** with the columns on **df2** where the rows for **col** have identical values. **how** can be one of 'left', 'right', 'outer', 'inner'

STATISTICS

These can all be applied to a series as well.

df.describe() - Summary statistics for numerical columns

df.mean() - Return the mean of all columns

df.corr() - finds the correlation between columns in a DataFrame.

df.count() - counts the number of non-null values in each DataFrame column.

df.max() - finds the highest value in each column.

df.min() - finds the lowest value in each column.

df.median() - finds the median of each column.

df.std() - finds the standard deviation of each column.



Special Offer - Save up to **25%** off Annual Subscriptions

Subscribe and Save →

Terms and conditions apply. Offer Expires January 31st 2017.

Learn Data Science

Our practical approach teaches you Data Science using interactive coding challenges.

You'll learn how to think like a data scientist.

You'll learn how to solve problems like a data scientist.

You'll work on the same projects a data scientist works on.

And eventually, you'll become a data scientist.



Learn by Doing

Watching videos doesn't help you learn. Instead, you'll be writing code and working with real-life data sets from your browser. We'll check your code, give you hints along the way, and give you help and support as you learn.

Practice your skills

We'll teach you all the skills you need to get a data science job, in one handy path. You'll have a chance to practice every single skill you gain. The more you practice, the better you'll get at thinking like a data scientist.

Build projects

Building projects helps reinforce your learning, and demonstrate your skills to employers. Our guided projects help you explore techniques and master data science skills, so you can build your data science portfolio and get hired.

Learn and work with others

Communication and collaboration are two important skills for data scientists. By sharing your work and collaborating with other students in our Dataquest community, you'll become a well rounded data practitioner.

Start Learning For Free Today

Our free tier includes 20 free missions and projects

Sign Up For Free



4,000 tech eBooks and Videos. All of them just \$5

Whatever you want to learn when it comes to tech, Packt have you covered this December. Their epic offer has returned, and it's bigger than ever before – making it the perfect time to build your eBook library and prepare for 2017.

Go straight to Packt home page to [begin your \\$5 search](#), or check out some of the dedicated bundles below – you can pick up 5 products for \$25!

[\\$5 Big Data bundles](#)

[\\$5 Spark bundles](#)

[\\$5 Python Data Bundles](#)

[\\$5 Machine Learning Bundles](#)

[\\$5 Data Analysis Bundles](#)

As if that wasn't enough, Packt is also offering three free eBooks that **Data Elixir** and **Dataquest** subscribers might just love...

Simply click the links below and register with Packt to claim your free eBook. You'll find it in your account once you log in.

[Learning Python](#)

[Building Machine Learning Systems with Python](#)

[Practical Data Analysis](#)



#packtfivedollar