Programming and frameworks for ML

Pandas Additional Resources

About Me

Big Data Consultant at Indra / Big Data Lecturer

- More than 20 years of experience in different environments, technologies, customers, countries ...
- Passionate data and technology
- Enthusiastic Big Data world and NoSQL

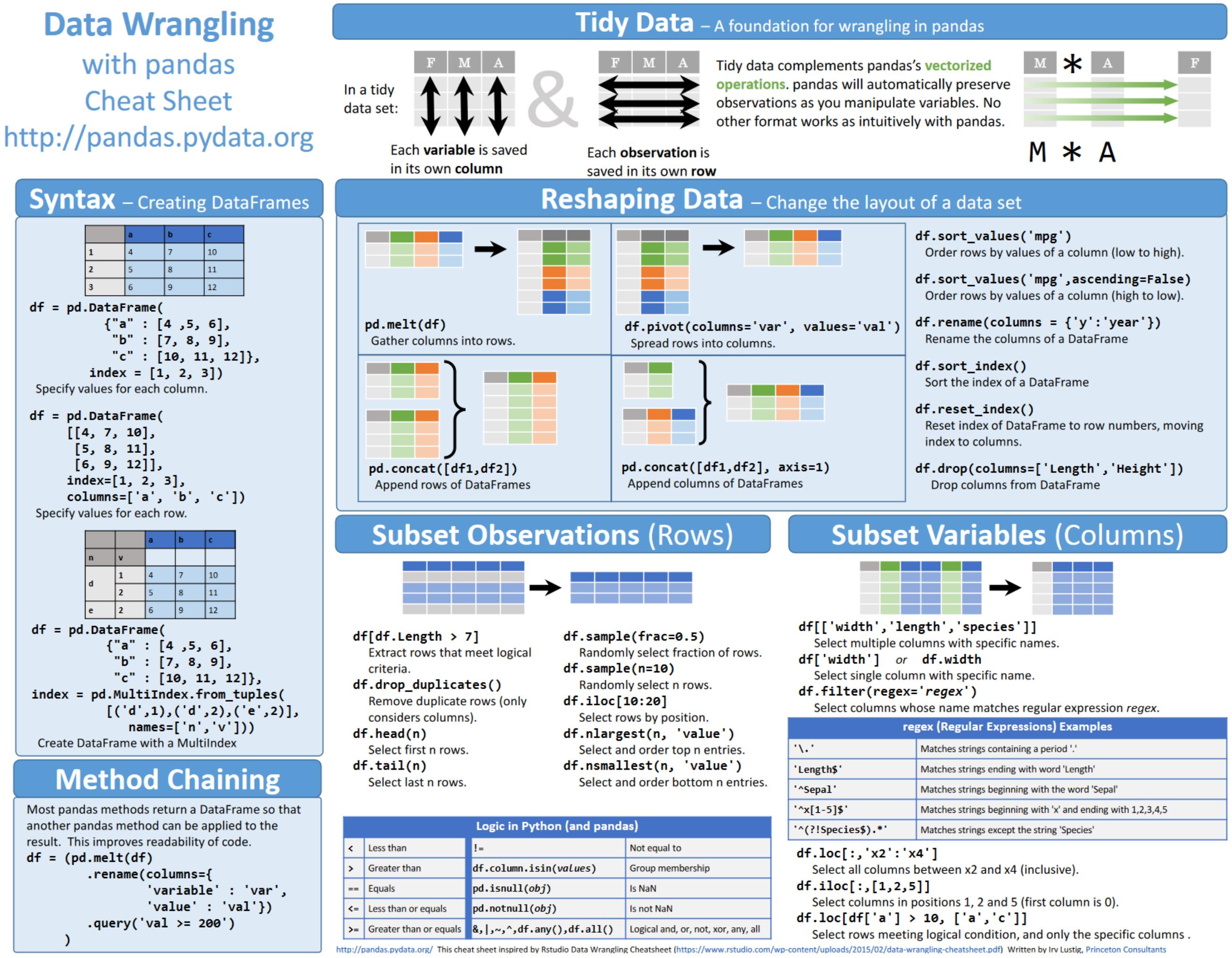
Daniel Villanueva Jiménez

BigData Developer / Lecturer

INDRA • Universidad Pontificia de Salamanca

IMMUNE
CODING INSTITUTE

Pandas Cheat Sheet



Data Wrangling
with pandas
Cheat Sheet
http://pandas.pydata.org

Tidy Data – A foundation for wrangling in pandas

In a tidy data set:

F M A

Each **variable** is saved in its own **column**

&

F M A

Each **observation** is saved in its own **row**

Tidy data complements pandas's vectorized operations. pandas will automatically preserve observations as you manipulate variables. No other format works as intuitively with pandas.

M * A F

M * A

Syntax – Creating DataFrames

	a	b	c
1	4	7	10
2	5	8	11
3	6	9	12

```
df = pd.DataFrame(
          {"a" : [4 ,5, 6],
           "b" : [7, 8, 9],
           "c" : [10, 11, 12]},
        index = [1, 2, 3])
```
Specify values for each column.

```
df = pd.DataFrame(
     [[4, 7, 10],
      [5, 8, 11],
      [6, 9, 12]],
     index=[1, 2, 3],
     columns=['a', 'b', 'c'])
```
Specify values for each row.

		a	b	c
n	v			
d	1	4	7	10
	2	5	8	11
e	2	6	9	12

```
df = pd.DataFrame(
          {"a" : [4 ,5, 6],
           "b" : [7, 8, 9],
           "c" : [10, 11, 12]},
index = pd.MultiIndex.from_tuples(
          [('d',1),('d',2),('e',2)],
             names=['n','v']))
```
Create DataFrame with a MultiIndex

Method Chaining

Most pandas methods return a DataFrame so that another pandas method can be applied to the result. This improves readability of code.

```
df = (pd.melt(df)
        .rename(columns={
                'variable' : 'var',
                'value' : 'val'})
        .query('val >= 200')
     )
```

Reshaping Data – Change the layout of a data set

`pd.melt(df)`
Gather columns into rows.

`df.pivot(columns='var', values='val')`
Spread rows into columns.

`pd.concat([df1,df2])`
Append rows of DataFrames

`pd.concat([df1,df2], axis=1)`
Append columns of DataFrames

`df.sort_values('mpg')`
Order rows by values of a column (low to high).

`df.sort_values('mpg',ascending=False)`
Order rows by values of a column (high to low).

`df.rename(columns = {'y':'year'})`
Rename the columns of a DataFrame

`df.sort_index()`
Sort the index of a DataFrame

`df.reset_index()`
Reset index of DataFrame to row numbers, moving index to columns.

`df.drop(columns=['Length','Height'])`
Drop columns from DataFrame

Subset Observations (Rows)

`df[df.Length > 7]`
Extract rows that meet logical criteria.

`df.drop_duplicates()`
Remove duplicate rows (only considers columns).

`df.head(n)`
Select first n rows.

`df.tail(n)`
Select last n rows.

`df.sample(frac=0.5)`
Randomly select fraction of rows.

`df.sample(n=10)`
Randomly select n rows.

`df.iloc[10:20]`
Select rows by position.

`df.nlargest(n, 'value')`
Select and order top n entries.

`df.nsmallest(n, 'value')`
Select and order bottom n entries.

Logic in Python (and pandas)			
<	Less than	!=	Not equal to
>	Greater than	`df.column.isin(values)`	Group membership
==	Equals	`pd.isnull(obj)`	Is NaN
<=	Less than or equals	`pd.notnull(obj)`	Is not NaN
>=	Greater than or equals	`&,|,~,^,df.any(),df.all()`	Logical and, or, not, xor, any, all

Subset Variables (Columns)

`df[['width','length','species']]`
Select multiple columns with specific names.

`df['width']` *or* `df.width`
Select single column with specific name.

`df.filter(regex='regex')`
Select columns whose name matches regular expression *regex*.

regex (Regular Expressions) Examples	
`'\.'`	Matches strings containing a period '.'
`'Length$'`	Matches strings ending with word 'Length'
`'^Sepal'`	Matches strings beginning with the word 'Sepal'
`'^x[1-5]$'`	Matches strings beginning with 'x' and ending with 1,2,3,4,5
`'^(?!Species$).*'`	Matches strings except the string 'Species'

`df.loc[:,'x2':'x4']`
Select all columns between x2 and x4 (inclusive).

`df.iloc[:,[1,2,5]]`
Select columns in positions 1, 2 and 5 (first column is 0).

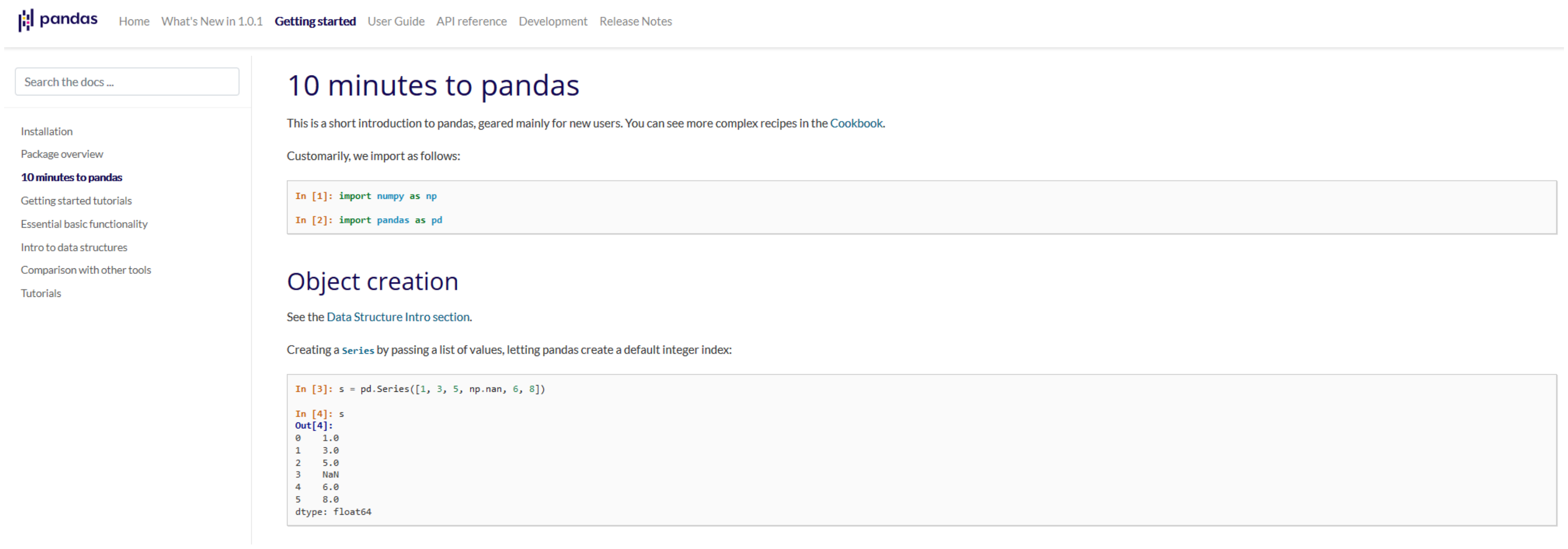
`df.loc[df['a'] > 10, ['a','c']]`
Select rows meeting logical condition, and only the specific columns .

http://pandas.pydata.org/ This cheat sheet inspired by Rstudio Data Wrangling Cheatsheet (https://www.rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf) Written by Irv Lustig, Princeton Consultants

https://pandas.pydata.org/Pandas_Cheat_Sheet.pdf

IMMUNE CODING INSTITUTE

10 minutes to pandas



10 minutes to pandas

This is a short introduction to pandas, geared mainly for new users. You can see more complex recipes in the Cookbook.

Customarily, we import as follows:

```
In [1]: import numpy as np

In [2]: import pandas as pd
```

Object creation

See the Data Structure Intro section.

Creating a `Series` by passing a list of values, letting pandas create a default integer index:

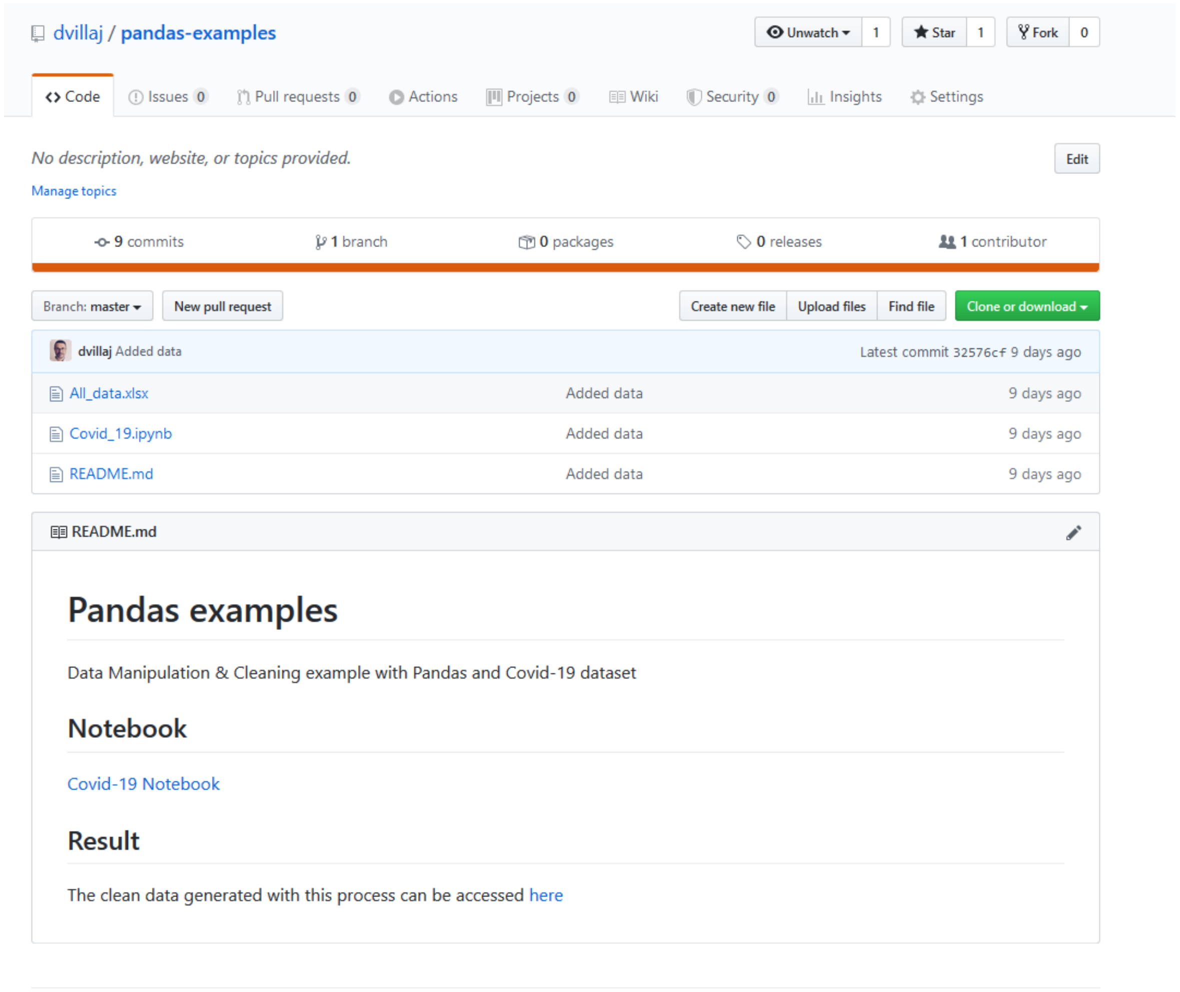
```
In [3]: s = pd.Series([1, 3, 5, np.nan, 6, 8])

In [4]: s
Out[4]:
0    1.0
1    3.0
2    5.0
3    NaN
4    6.0
5    8.0
dtype: float64
```

https://pandas.pydata.org/pandas-docs/stable/getting_started/10min.html

IMMUNE CODING INSTITUTE

Examples


dvillaj / pandas-examples

Unwatch 1 | Star 1 | Fork 0

No description, website, or topics provided.

Manage topics

9 commits | 1 branch | 0 packages | 0 releases | 1 contributor

dvillaj Added data — Latest commit 32576cf 9 days ago

All_data.xlsx	Added data	9 days ago
Covid_19.ipynb	Added data	9 days ago
README.md	Added data	9 days ago

README.md

Pandas examples

Data Manipulation & Cleaning example with Pandas and Covid-19 dataset

Notebook

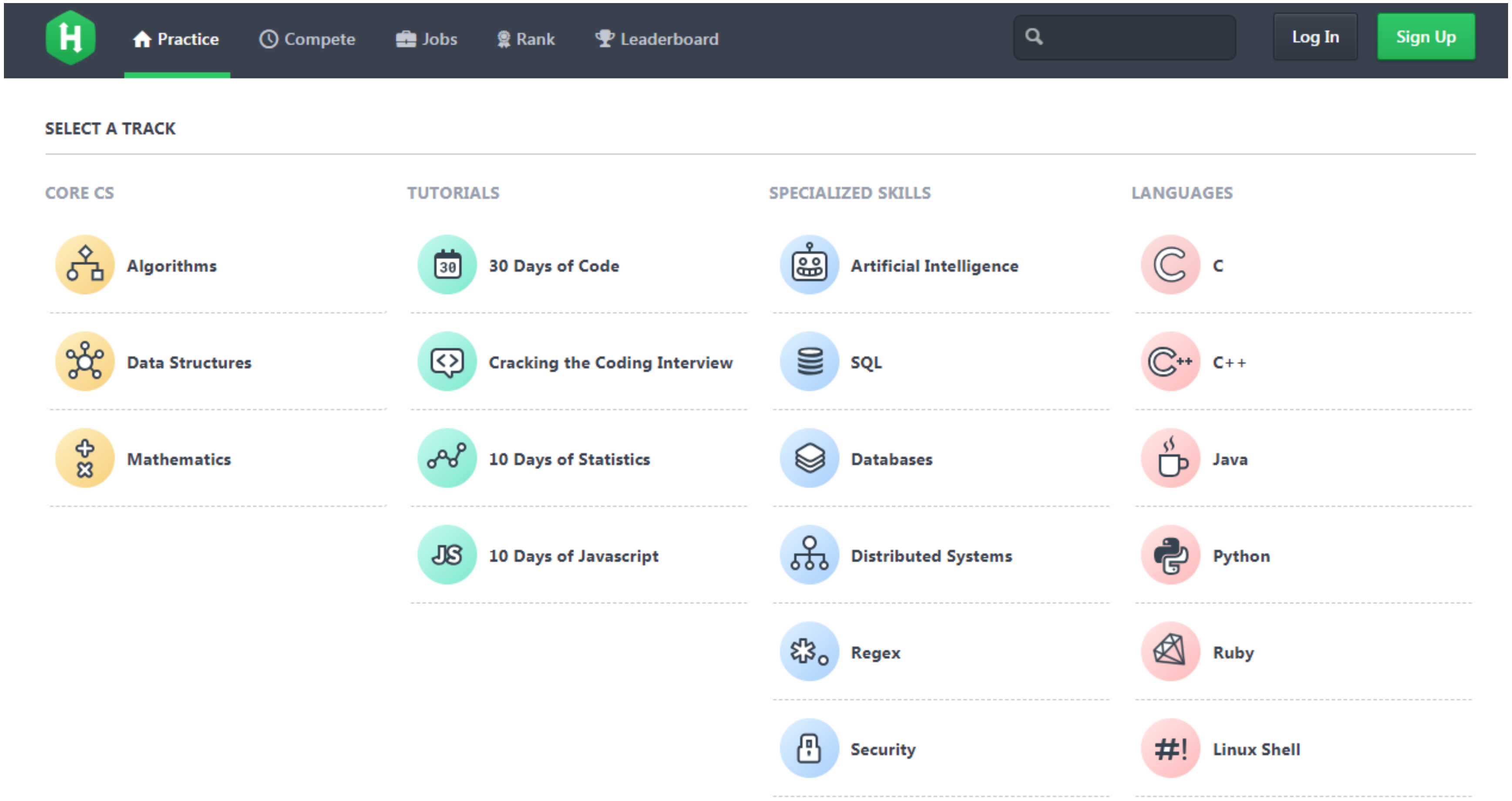
Covid-19 Notebook

Result

The clean data generated with this process can be accessed here

https://github.com/dvillaj/pandas-examples

Practice

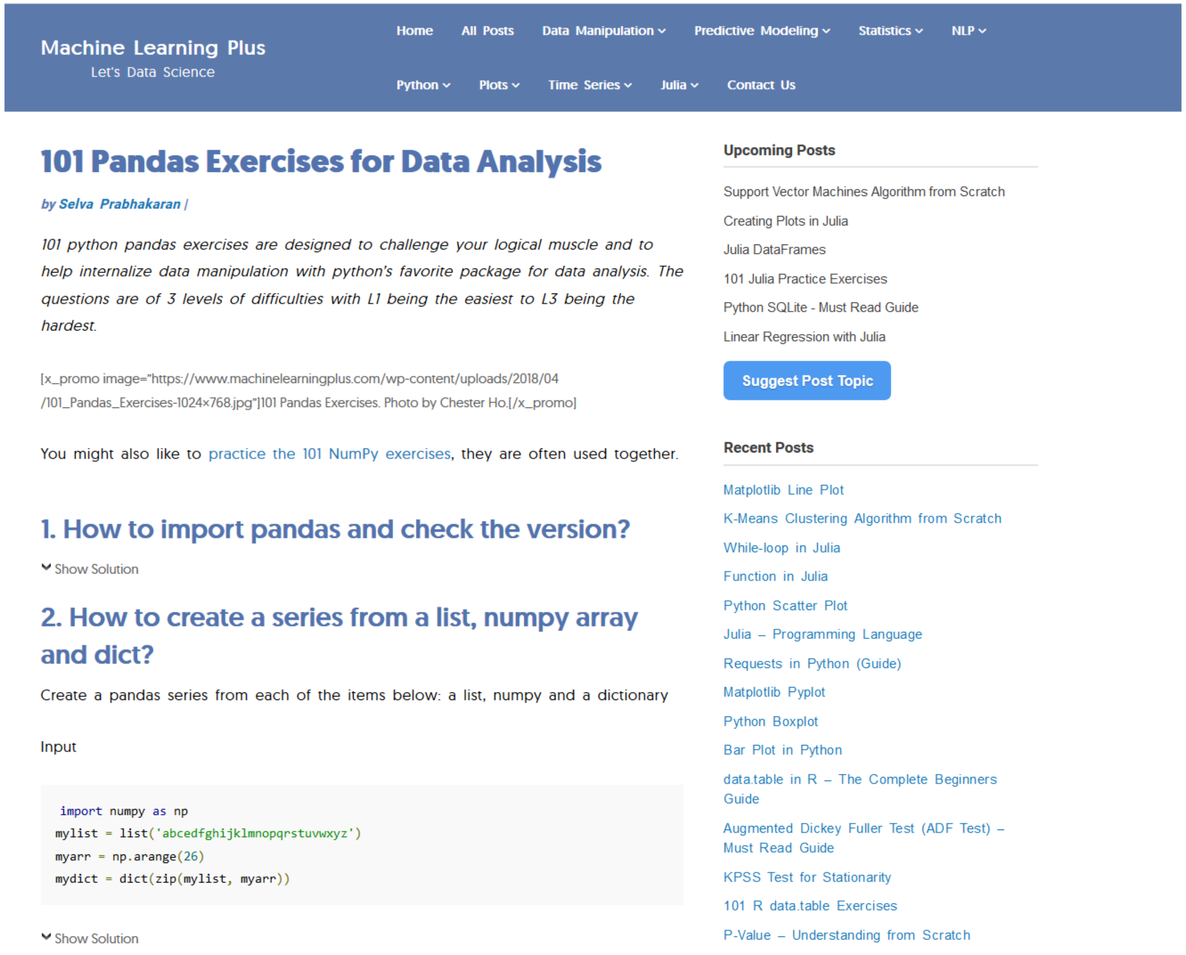


SELECT A TRACK

CORE CS	TUTORIALS	SPECIALIZED SKILLS	LANGUAGES
Algorithms	30 Days of Code	Artificial Intelligence	C
Data Structures	Cracking the Coding Interview	SQL	C++
Mathematics	10 Days of Statistics	Databases	Java
	10 Days of Javascript	Distributed Systems	Python
		Regex	Ruby
		Security	Linux Shell

https://www.hackerrank.com

Practice



101 Pandas Exercises for Data Analysis

by Selva Prabhakaran |

101 python pandas exercises are designed to challenge your logical muscle and to help internalize data manipulation with python's favorite package for data analysis. The questions are of 3 levels of difficulties with L1 being the easiest to L3 being the hardest.

[x_promo image="https://www.machinelearningplus.com/wp-content/uploads/2018/04/101_Pandas_Exercises-1024×768.jpg"]101 Pandas Exercises. Photo by Chester Ho.[/x_promo]

You might also like to practice the 101 NumPy exercises, they are often used together.

1. How to import pandas and check the version?

Show Solution

2. How to create a series from a list, numpy array and dict?

Create a pandas series from each of the items below: a list, numpy and a dictionary

Input

```
 import numpy as np
mylist = list('abcedfghijklmnopqrstuvwxyz')
myarr = np.arange(26)
mydict = dict(zip(mylist, myarr))
```

Show Solution

Upcoming Posts

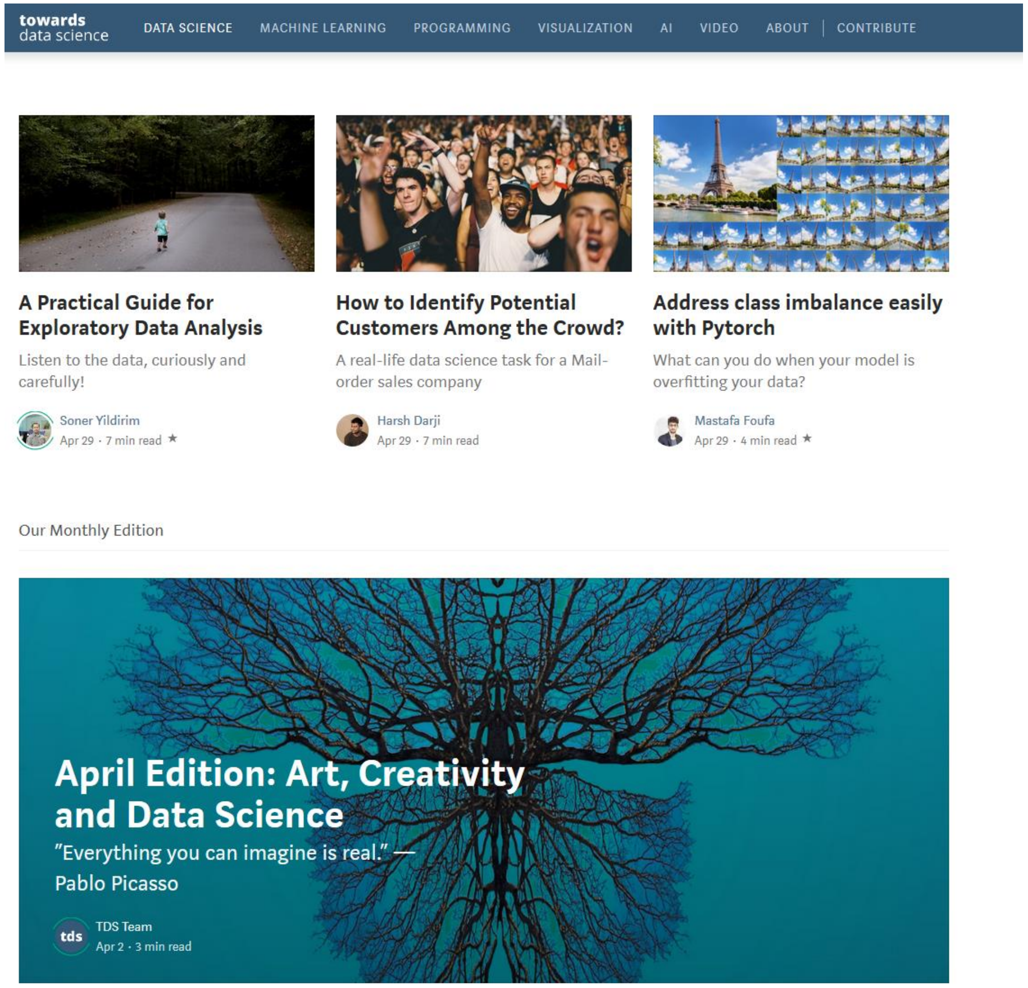
Support Vector Machines Algorithm from Scratch
Creating Plots in Julia
Julia DataFrames
101 Julia Practice Exercises
Python SQLite - Must Read Guide
Linear Regression with Julia

Recent Posts

Matplotlib Line Plot
K-Means Clustering Algorithm from Scratch
While-loop in Julia
Function in Julia
Python Scatter Plot
Julia – Programming Language
Requests in Python (Guide)
Matplotlib Pyplot
Python Boxplot
Bar Plot in Python
data.table in R – The Complete Beginners Guide
Augmented Dickey Fuller Test (ADF Test) – Must Read Guide
KPSS Test for Stationarity
101 R data.table Exercises
P-Value – Understanding from Scratch

https://www.machinelearningplus.com/python/101-pandas-exercises-python/

Articles



towards data science

DATA SCIENCE · MACHINE LEARNING · PROGRAMMING · VISUALIZATION · AI · VIDEO · ABOUT | CONTRIBUTE

A Practical Guide for Exploratory Data Analysis

Listen to the data, curiously and carefully!

Soner Yildirim
Apr 29 · 7 min read ★

How to Identify Potential Customers Among the Crowd?

A real-life data science task for a Mail-order sales company

Harsh Darji
Apr 29 · 7 min read

Address class imbalance easily with Pytorch

What can you do when your model is overfitting your data?

Mastafa Foufa
Apr 29 · 4 min read ★

Our Monthly Edition

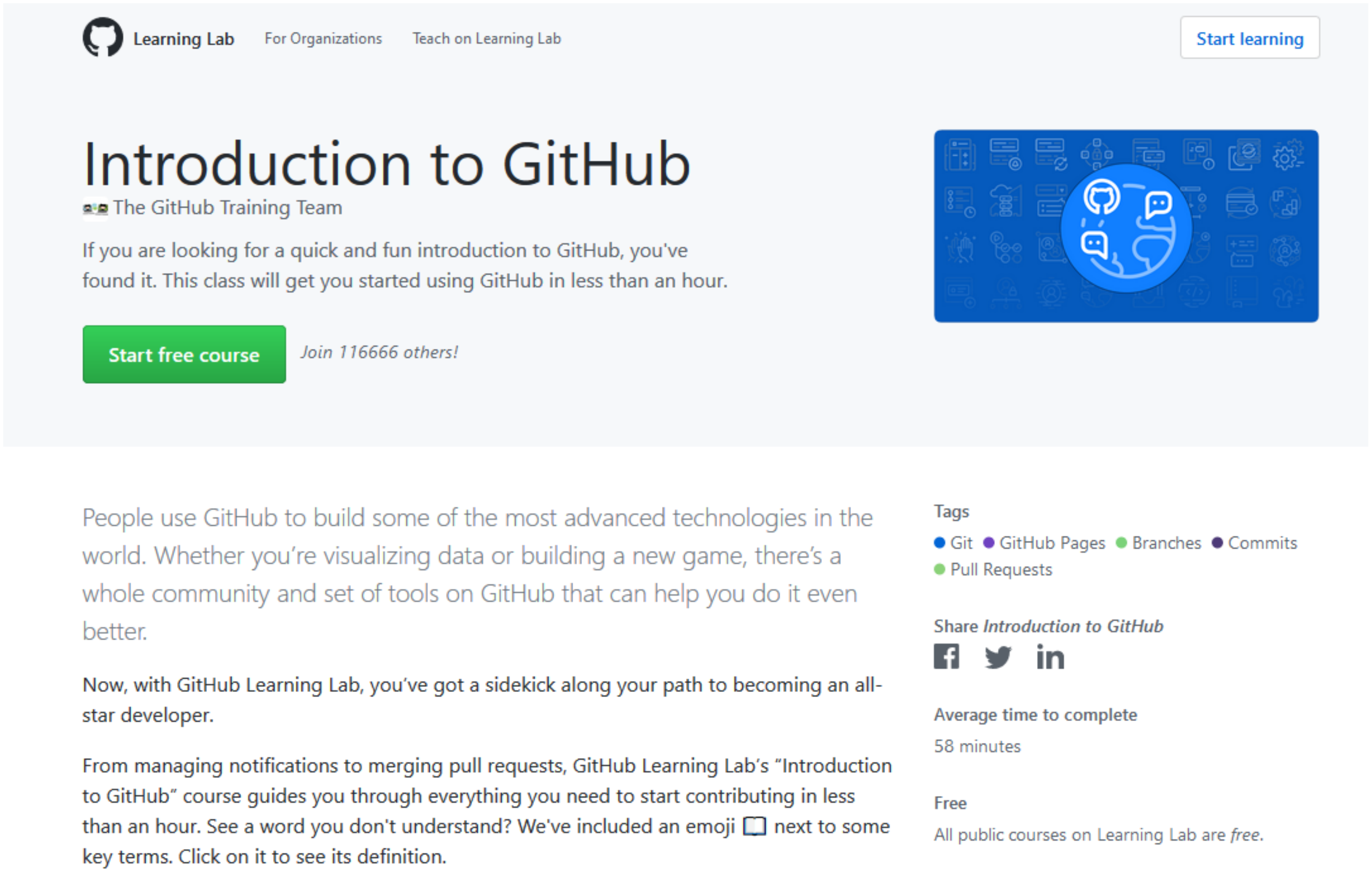
April Edition: Art, Creativity and Data Science

"Everything you can imagine is real." — Pablo Picasso

TDS Team
Apr 2 · 3 min read

https://towardsdatascience.com/data-science/home

GitHub



Introduction to GitHub

The GitHub Training Team

If you are looking for a quick and fun introduction to GitHub, you've found it. This class will get you started using GitHub in less than an hour.

Join 116666 others!

People use GitHub to build some of the most advanced technologies in the world. Whether you're visualizing data or building a new game, there's a whole community and set of tools on GitHub that can help you do it even better.

Now, with GitHub Learning Lab, you've got a sidekick along your path to becoming an all-star developer.

From managing notifications to merging pull requests, GitHub Learning Lab's "Introduction to GitHub" course guides you through everything you need to start contributing in less than an hour. See a word you don't understand? We've included an emoji 📖 next to some key terms. Click on it to see its definition.

Tags

● Git ● GitHub Pages ● Branches ● Commits ● Pull Requests

Share *Introduction to GitHub*

Average time to complete

58 minutes

Free

All public courses on Learning Lab are *free*.

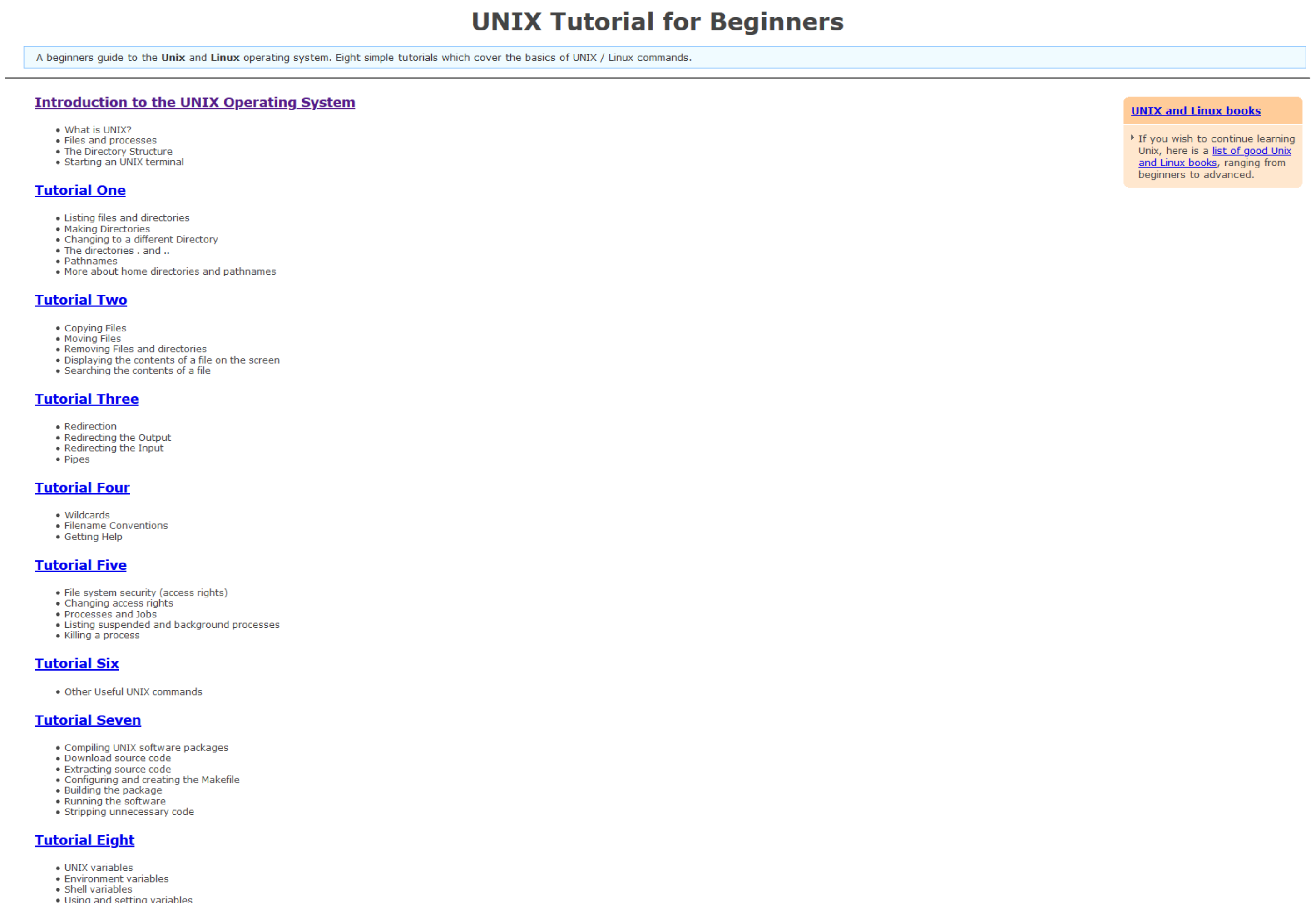
https://lab.github.com/githubtraining/introduction-to-github

IMMUNE CODING INSTITUTE

Git Client

https://www.gitkraken.com/

Unix Tutorial



UNIX Tutorial for Beginners

A beginners guide to the **Unix** and **Linux** operating system. Eight simple tutorials which cover the basics of UNIX / Linux commands.

Introduction to the UNIX Operating System

- What is UNIX?
- Files and processes
- The Directory Structure
- Starting an UNIX terminal

Tutorial One

- Listing files and directories
- Making Directories
- Changing to a different Directory
- The directories . and ..
- Pathnames
- More about home directories and pathnames

Tutorial Two

- Copying Files
- Moving Files
- Removing Files and directories
- Displaying the contents of a file on the screen
- Searching the contents of a file

Tutorial Three

- Redirection
- Redirecting the Output
- Redirecting the Input
- Pipes

Tutorial Four

- Wildcards
- Filename Conventions
- Getting Help

Tutorial Five

- File system security (access rights)
- Changing access rights
- Processes and Jobs
- Listing suspended and background processes
- Killing a process

Tutorial Six

- Other Useful UNIX commands

Tutorial Seven

- Compiling UNIX software packages
- Download source code
- Extracting source code
- Configuring and creating the Makefile
- Building the package
- Running the software
- Stripping unnecessary code

Tutorial Eight

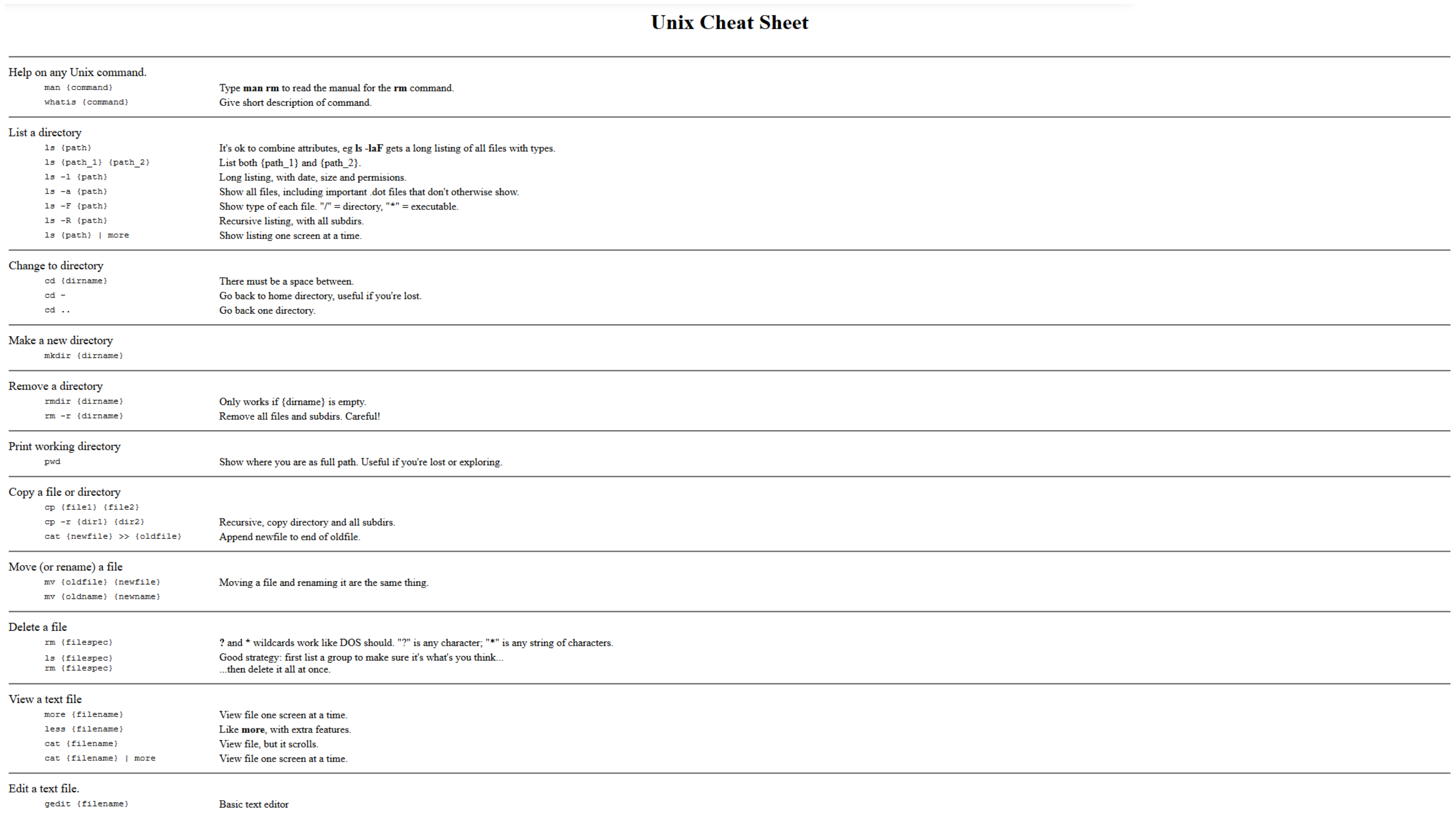
- UNIX variables
- Environment variables
- Shell variables
- Using and setting variables

UNIX and Linux books

If you wish to continue learning Unix, here is a list of good Unix and Linux books, ranging from beginners to advanced.

http://www.ee.surrey.ac.uk/Teaching/Unix/

Unix Cheat Sheet



Unix Cheat Sheet

Help on any Unix command.

`man {command}`	Type **man rm** to read the manual for the **rm** command.
`whatis {command}`	Give short description of command.

List a directory

`ls {path}`	It's ok to combine attributes, eg **ls -laF** gets a long listing of all files with types.
`ls {path_1} {path_2}`	List both {path_1} and {path_2}.
`ls -l {path}`	Long listing, with date, size and permisions.
`ls -a {path}`	Show all files, including important .dot files that don't otherwise show.
`ls -F {path}`	Show type of each file. "/" = directory, "*" = executable.
`ls -R {path}`	Recursive listing, with all subdirs.
`ls {path} | more`	Show listing one screen at a time.

Change to directory

`cd {dirname}`	There must be a space between.
`cd ~`	Go back to home directory, useful if you're lost.
`cd ..`	Go back one directory.

Make a new directory

`mkdir {dirname}`	

Remove a directory

`rmdir {dirname}`	Only works if {dirname} is empty.
`rm -r {dirname}`	Remove all files and subdirs. Careful!

Print working directory

`pwd`	Show where you are as full path. Useful if you're lost or exploring.

Copy a file or directory

`cp {file1} {file2}`	
`cp -r {dir1} {dir2}`	Recursive, copy directory and all subdirs.
`cat {newfile} >> {oldfile}`	Append newfile to end of oldfile.

Move (or rename) a file

`mv {oldfile} {newfile}`	Moving a file and renaming it are the same thing.
`mv {oldname} {newname}`	

Delete a file

`rm {filespec}`	? and * wildcards work like DOS should. "?" is any character; "*" is any string of characters.
`ls {filespec}` `rm {filespec}`	Good strategy: first list a group to make sure it's what's you think... ...then delete it all at once.

View a text file

`more {filename}`	View file one screen at a time.
`less {filename}`	Like **more**, with extra features.
`cat {filename}`	View file, but it scrolls.
`cat {filename} | more`	View file one screen at a time.

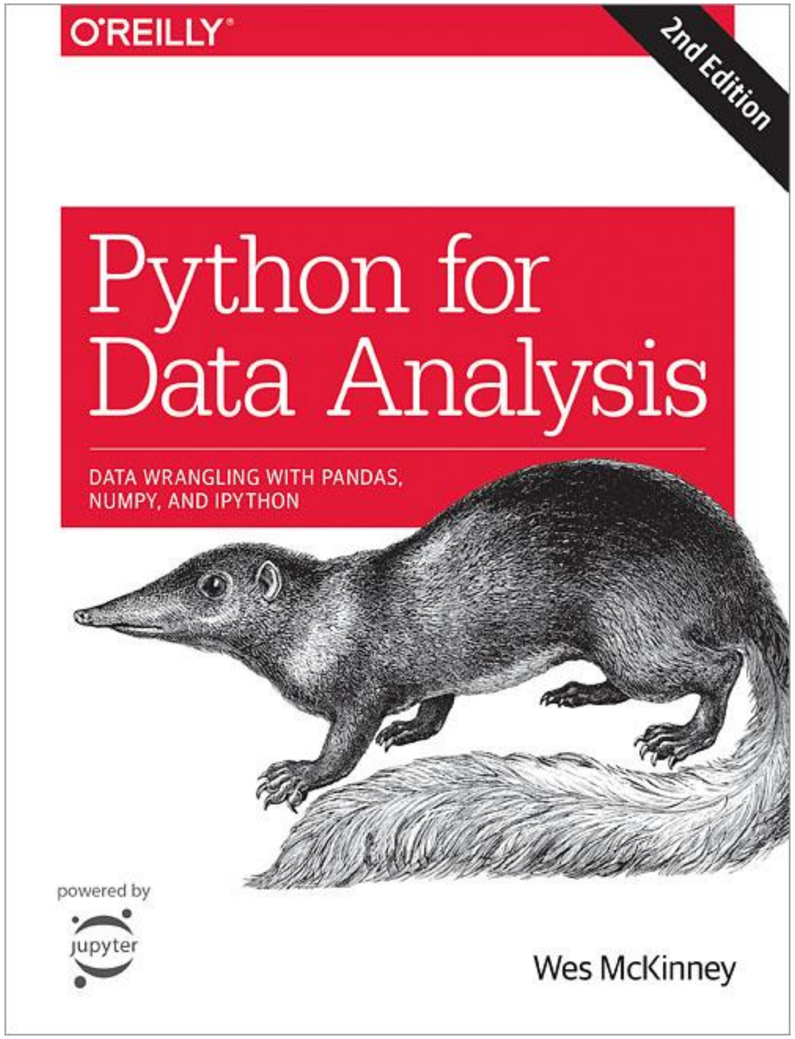
Edit a text file.

`gedit {filename}`	Basic text editor

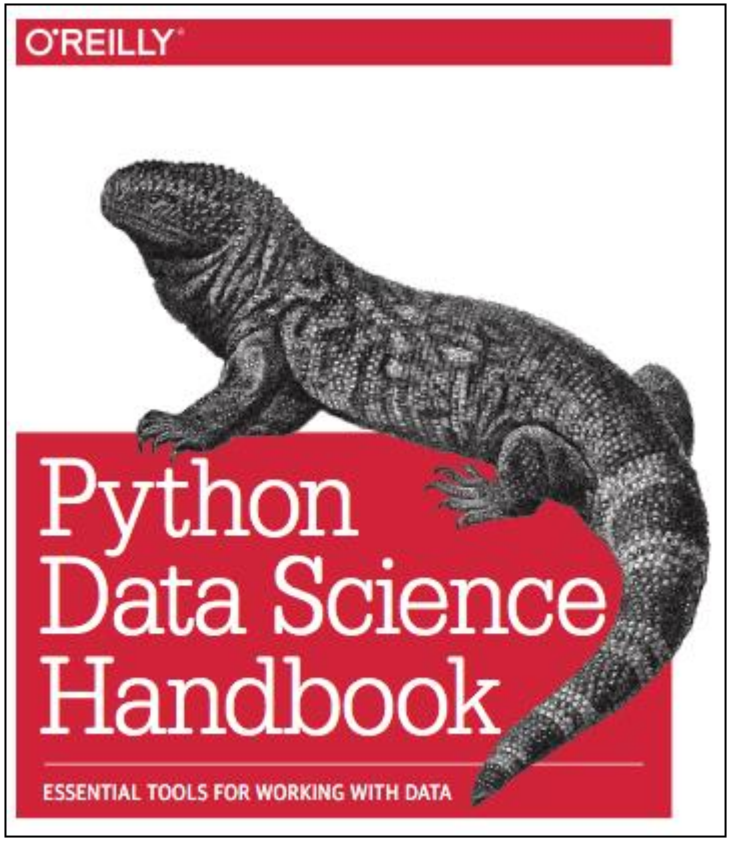
http://www.mathcs.emory.edu/~valerie/courses/fall10/155/resources/unix_cheatsheet.html

IMMUNE
CODING INSTITUTE

Bibliography



Bibliography



- https://github.com/jakevdp/PythonDataScienc eHandbook

THANKS FOR YOUR ATTENTION

Daniel Villanueva Jiménez
daniel.villanueva@immune.institute
@dvillaj