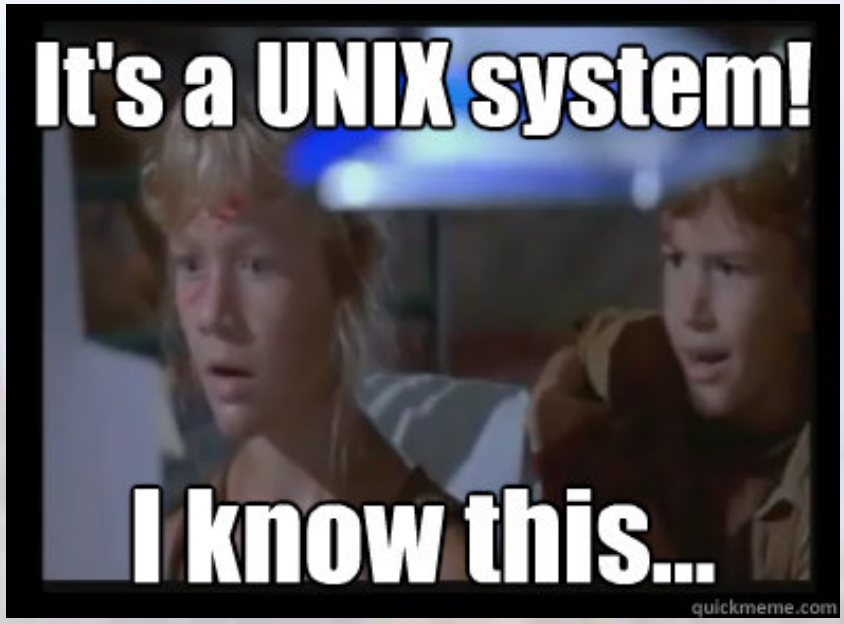
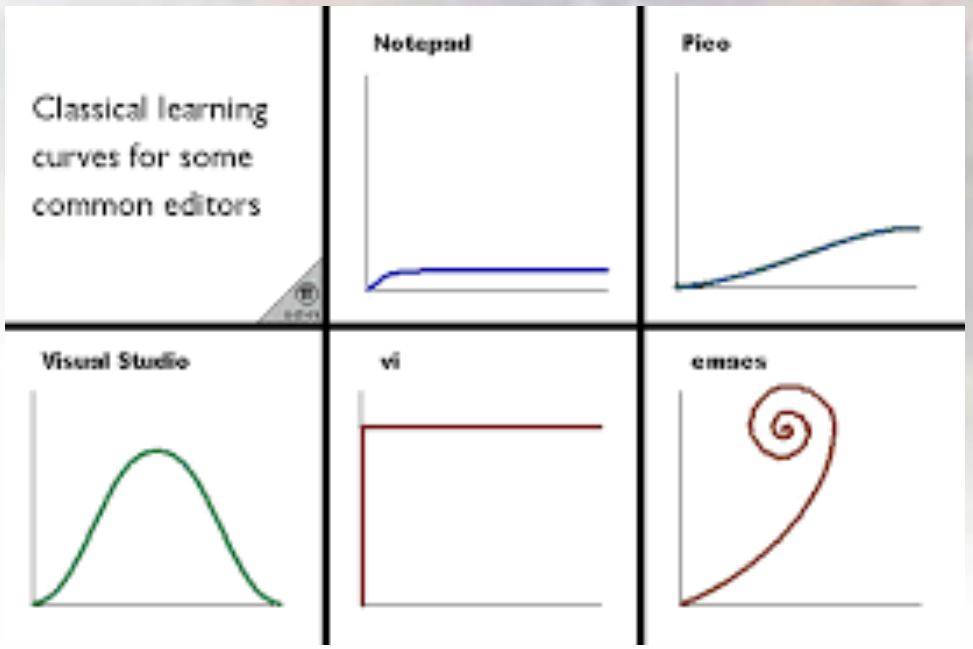
Unix/Linux


It's a UNIX system!
I know this...
quickmeme.com


Classical learning curves for some common editors
Notepad
Pico
Visual Studio
vi
emacs

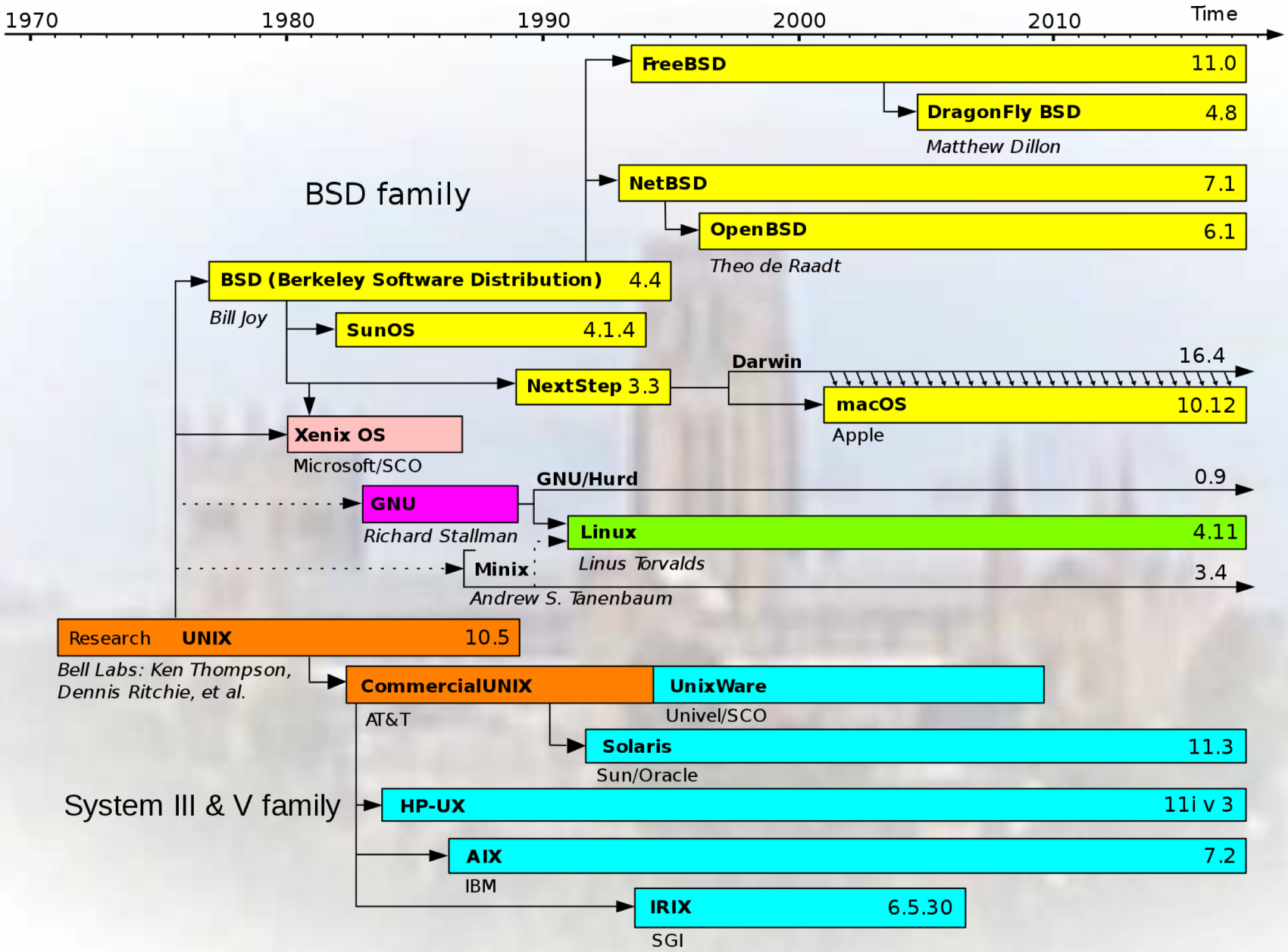
Brief Unix history

In the mid 60s the large major frames that used the Multics (Multiplexed Information and Computing Service) OS were becoming very complex and costly for the development of large programmes. Richie and Thompson, who were Multics programmers at Bell Labs, built a simpler, smaller (~8 kb) system that they would call Unix – the name was "a kind of treacherous pun on Multics". This led onto the development of a new language C.

Dennis Ritchie (standing) with Ken Thompson at work in 1972.

"Because Bell Labs' parent, AT&T, was a regulated telephone monopoly, it was prohibited from competing in the computer industry, and so had no pecuniary interest in Unix. This allowed Ritchie and Thompson to distribute Unix free of charge to universities and research institutions, which loved its clean, economical design."

Dennis Ritchie obituary in
the Guardian, 13th October 2011

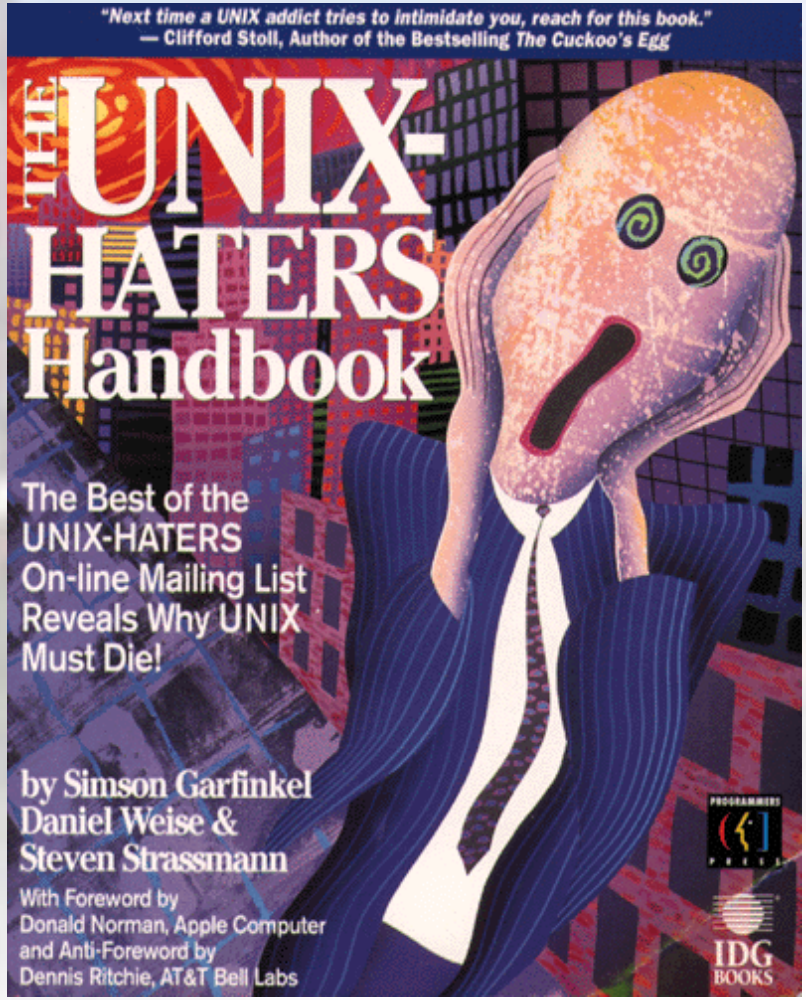


Major Linux distributions: Ubuntu, Fedora, Scientific Linux, Debian, Mint, …

Unix/Linux

"Two of the most famous products of Berkeley are LSD and Unix.
I don't think that this is a coincidence" - Anonymous

"I have a natural revulsion to any operating system that shows so little planning as to have named its commands after digestive noises (awk, grep, fsck, nroff)."



Case Sensitivity

In Windows-based operating systems, the names of most objects (such as files and directories) are case preserving, but case insensitive. That means you can use uppercase and lowercase characters when naming such objects, but these operating systems do not distinguish between names based on case alone. For example, you cannot have two files in the same directory named **Galaxies.txt** and **galaxies.txt** because the system regards the names to be identical for the purposes of identifying files.

Unix/Linux, by contrast, is fully case sensitive, and so these distinguish between object names when the only difference between those names is the case of characters used in the object names.

OSX is like Windows case preserving, but case insensitive.

Linux / Unix 101 commands

Linux / Unix ls command

Lists the files in a directory

ls *[options] [file]*

Examples

ls -a will list all files including hidden files
(files with names beginning with a dot)

ls -l gives a long listing of all files, e.g.

-rw-r--r-- 1 jrl jrl 12501 2009-09-29 10:25 known_hosts

ls -lrt lists files in reverse date order

Linux / Unix cd, pwd commands

Changes the directory, i.e. cd *[directory]*

cd .. go to parent directory

cd - go back to the previous directory

cd ~ or cd $home go to the user's home directory

Examples:

cd text

cd ../2mass

pwd prints working directory

Linux / Unix cp commands

Copies a file(s) from one location to another.

cp [OPTION]... SOURCE DEST
cp [OPTION]... SOURCE... DIRECTORY

Valuable options

-f force overwriting destination file if it exists
-p preserves file attributes
-r copy directories recursively

cp -pr /home/jrl/pretty_pictures/ /home/jrl/backup

Note what the following gives instead
cp -pr /home/jrl/pretty_pictures /home/jrl/backup

Linux / Unix chmod command

Changes permissions of a file.

chmod [OPTION]... MODE[,MODE]... FILE...

chmod +r file read access is enabled for everyone

(r=read, w=write, x=execute
+ adds, – removes, = exact mode
u=user, g=group, o=other, a=all)

chmod ug+rw mydir	Add the read and write permissions to the user and group
chmod 755	read and execute access for everyone and also write access for the owner of the file

Linux / Unix grep command

grep = *global / regular expression / print*

Searches for text strings in files.

grep [options] PATTERN [FILE...]

Example: grep unix *.html
search all .htm files in the current directory
for any reference of unix

Many options including

-i ignore case
-v inverted match

Linux / Unix sort command

Sorts the lines in a text file.

sort [options] [+fields] filename

Many options including
- -r sorts in reverse order
- -k column position to sort on
- -n sort on numeric value rather than string

Unix/Linux 101

More key commands:

- man – display online help
- cat – displays the content of a file
- df – reports amount of free disk space, e.g. df -h
- diff – compares two files
- find – finds files with given conditions, e.g. find . -name “*.c”
- lpr – sends files to the printer; lprm to remove file from queue
- more – display files one screenful at a time
- mv – move or rename files and directories
- ps – list the processes that are running, e.g. ps ax
- rm – deletes one or more files; there is not an undelete!
- tail – print the last lines of a file
- quota – displays user's disk usage and limits
- kill – cancels a job, e.g. kill -9 xxxx
- history – lists previous commands
- top – lists the jobs running

Command line recall

TAB key for file completion and <Control>D to list options

tcsh shell (/bin/tsch/)

Defaults

e.g. setenv PATH *[or LD_LIBRARY_PATH]* in $home/.login
aliases set in $home/.cshrc

In my .cshrc file I have
alias jj ‘ps –ef | grep jrl’

Shell commands/features in /bin/tcsh
(other shells like /bin/sh similar)

cmd > file	Sends output of cmd to file
cmd >! file	As above but overwrites file
cmd >> file	Appends output to file
cmd >& file	Sends both standard output and standard error to file
cmd < file	Takes input for cmd from file
cmd1 | cmd2	Uses output from cmd1 as input to cmd2
	ls –l | sort –k 5 -n
cmd &	Runs process in the background
cmd1;cmd2	Puts two commands on the same line
cmd1$$cmd2	run cmd2 if cmd1 is successful

Essential Generic Tools

gzip/gunzip – compress/uncompress files

(s)ftp – transfers files to and from computers elsewhere
(mostly superseded by scp)

ssh – secure remote login programme
(can set up password-less login easily)

scp – secure remote copy

make - to maintain groups of programs

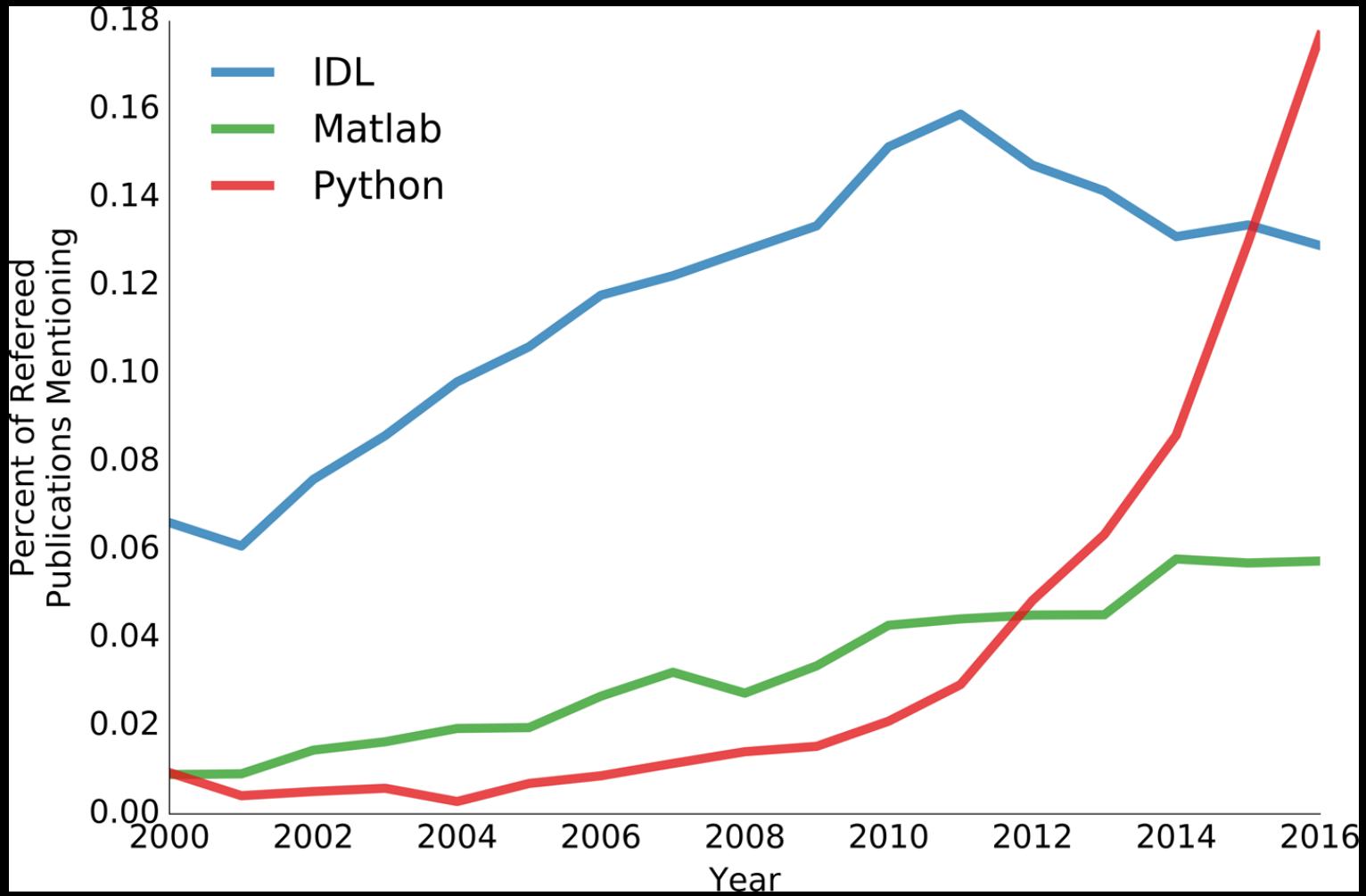
cron - for running commands at a set time

(g)awk - pattern scanning utility, e.g. swap columns of a table

tar - (tape) archive program to un/pack sets of files

wget - non-interactive download of files from the Web

rsync - provides fast incremental file transfer, great for backups


IDL
Matlab
Python
Percent of Refereed Publications Mentioning
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
2000
2002
2004
2006
2008
2010
2012
2014
2016
Year

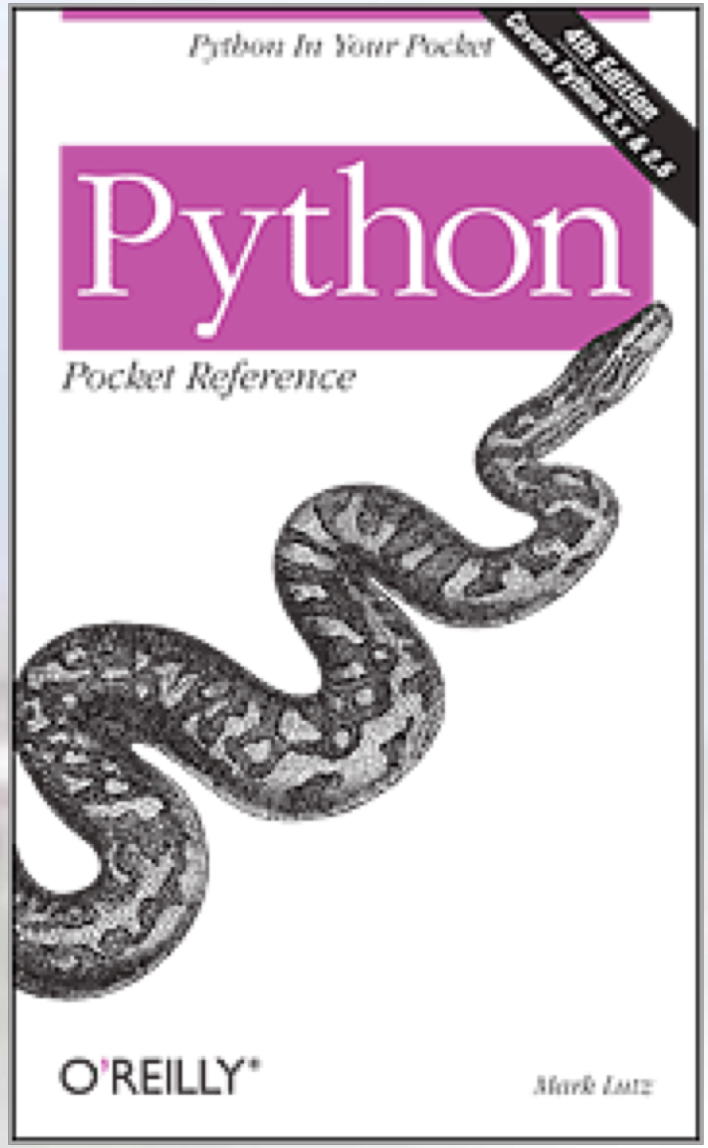
Python

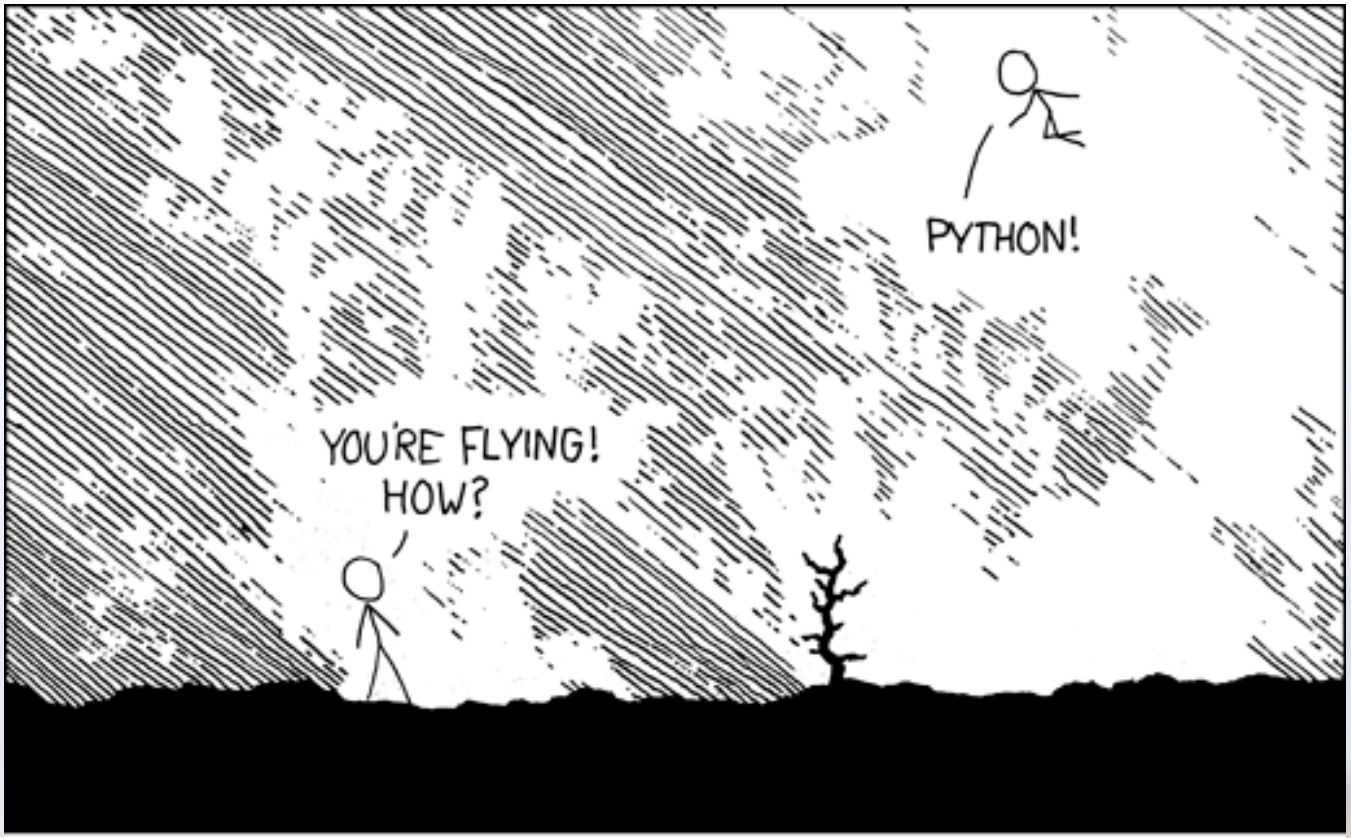
Python is a fun and extremely easy-to-use programming language that combines remarkable power with very clear syntax.

Python is optimized for software quality, portability, integration, and productivity. It fosters maintainable systems that run just about everywhere (Linux, Mac OS X, Windows) and can be freely mixed with other software components.

Great on-line material at www.python.org

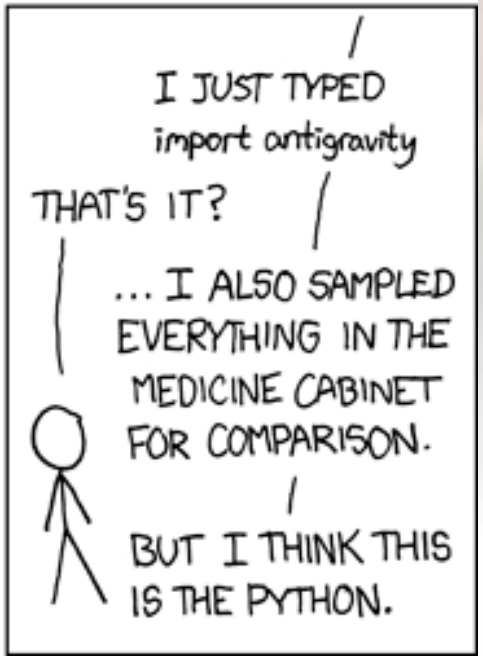
Python supports raising and catching exceptions.




PYTHON!
YOU'RE FLYING! HOW?


I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST print "Hello, world!"


I DUNNO... DYNAMIC TYPING? WHITESPACE?
COME JOIN US! PROGRAMMING IS FUN AGAIN! IT'S A WHOLE NEW WORLD UP HERE!
BUT HOW ARE YOU FLYING?


I JUST TYPED import antigravity
THAT'S IT?
... I ALSO SAMPLED EVERYTHING IN THE MEDICINE CABINET FOR COMPARISON.
BUT I THINK THIS IS THE PYTHON.

XKCD

Python: continued

Great for people who can't really programme, i.e. most of us!

Great for interaction with the OS.

Great for accessing web services (NED, SDSS, 2MASS, SIMBAD, etc)

Great for astronomy data analysis,
e.g. pyfits, numpy, pyraf, astropy, astroML, etc

Great for scientific programming via SciPy (pronounced "Sigh Pie"),
see www.scipy.org

Straightforward to call a Fortran subroutine or a C function,
via F2PY and SWIG respectively

"Using Python for Interactive Data Analysis"
by Perry Greenfield and Robert Jedrzejewski available from
http://stsdas.stsci.edu/perry/pydatatut.pdf

No native Numerical Recipes in Python, convert or use F2PY/SWIG

Many Specialist Python Packages and Notes Available

Python Scientific Lecture Notes
http://scipy-lectures.github.io/

Lists of available packages at for example
"Python for Astronomers" at the IAC
http://www.iac.es/sieinvens/siepedia/pmwiki.php?n=HOWTOs.EmpezandoPython

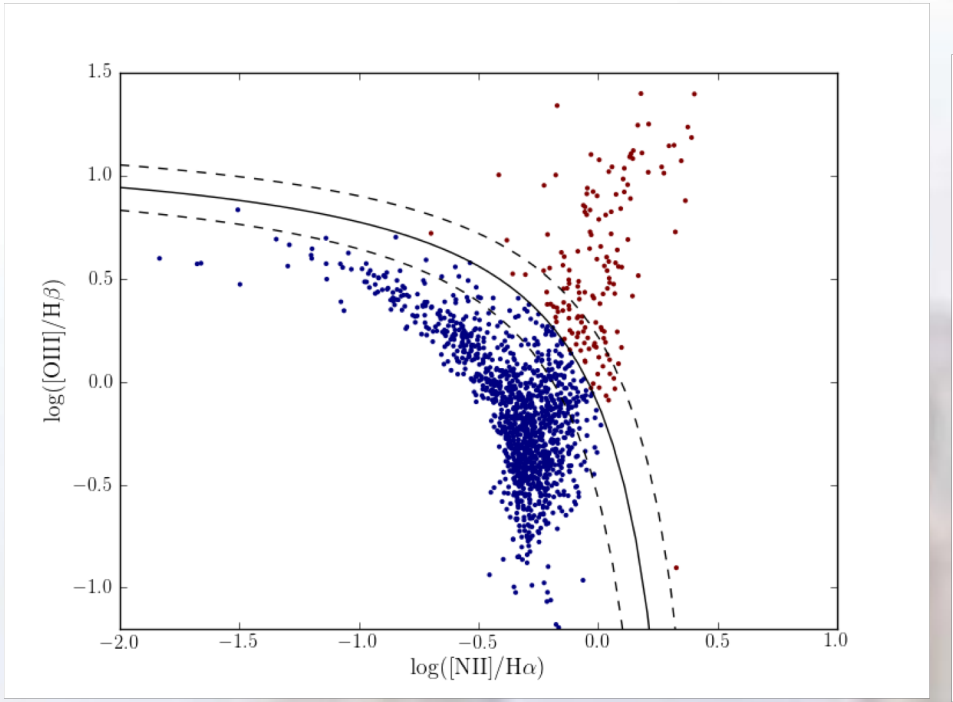
Some examples include

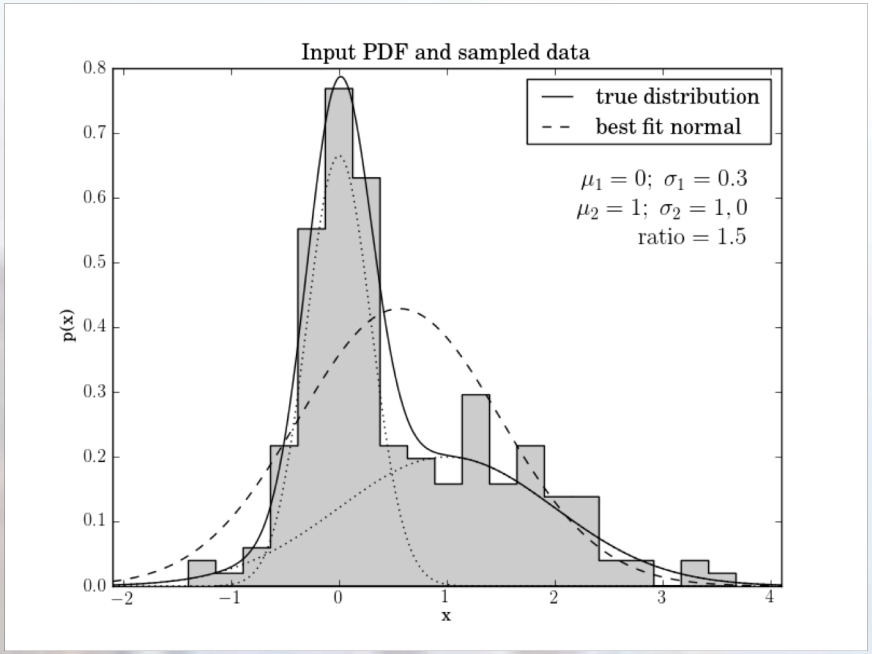
AstroML Machine Learning and Data Mining for Astronomy
http://www.astroml.org/
Check out the videos

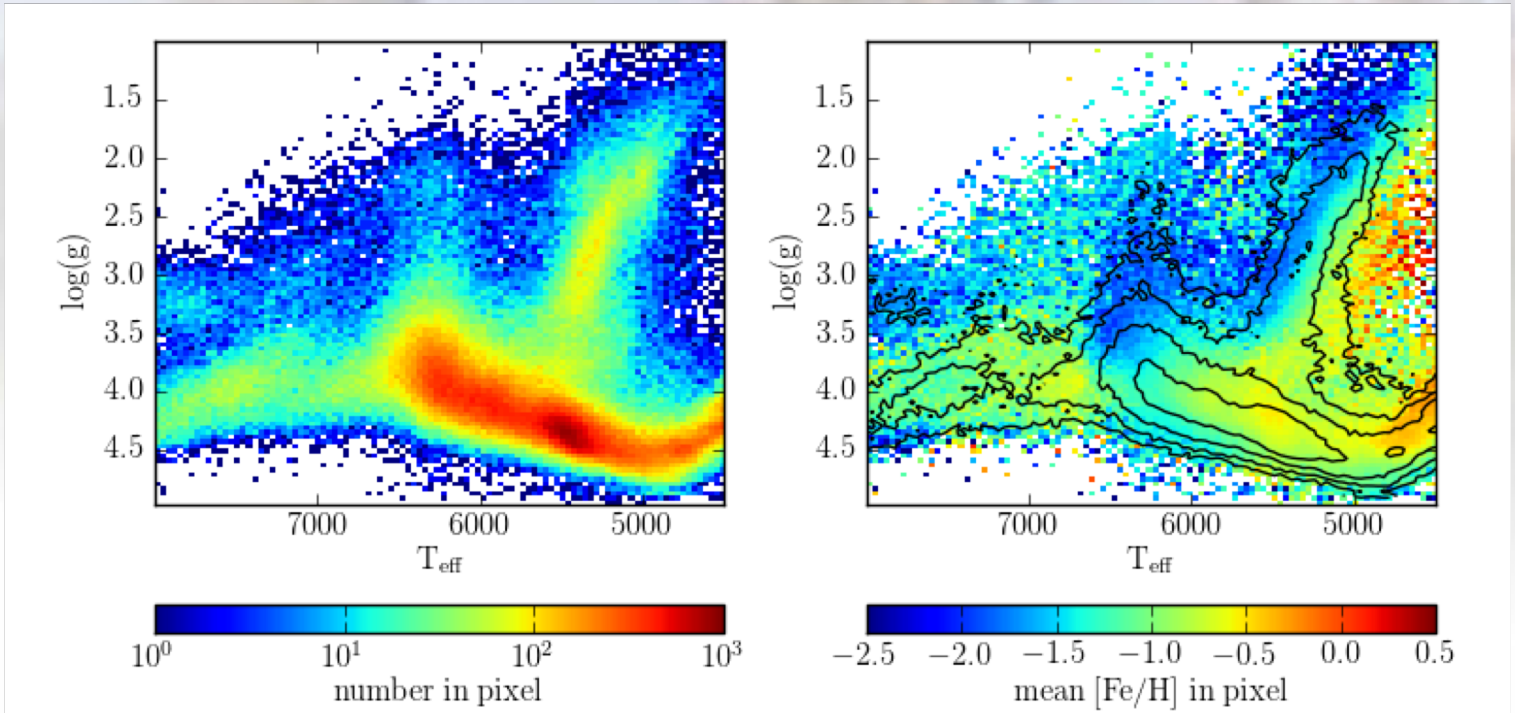
Astropy A Community Python Library for Astronomy
http://www.astropy.org/

LMFIT Non-Linear Least-Squares Minimization and Curve-Fitting for Python
http://lmfit.github.io/lmfit-py/

PyMC Markov chain Monte Carlo estimation
http://code.google.com/p/pymc/


log([OIII]/Hβ)
log([NII]/Hα)


Input PDF and sampled data
true distribution
best fit normal
$\mu_1 = 0;\ \sigma_1 = 0.3$
$\mu_2 = 1;\ \sigma_2 = 1,0$
ratio = 1.5
p(x)
x


log(g)
T_{eff}
number in pixel
mean [Fe/H] in pixel

Excellent tools

Starlink http://starlink.jach.hawaii.edu/starlink

Data reduction/analysis/programming tools
The Starlink Project was a long running UK Project supporting astronomical data processing. It was shut down in 2005 but the software continues to be developed at the Joint Astronomy Centre and is open source.

Peter Draper is the local expert.

Includes many noteworthy tools include GAIA,
TOPCAT, KAPPA, etc.

Data analysis recipes: Fitting a model to data*

David W. Hogg
Center for Cosmology and Particle Physics, Department of Physics, New York University
Max-Planck-Institut für Astronomie, Heidelberg

Jo Bovy
Center for Cosmology and Particle Physics, Department of Physics, New York University

Dustin Lang
Department of Computer Science, University of Toronto
Princeton University Observatory

Abstract

We go through the many considerations involved in fitting a model to data, using as an example the fit of a straight line to a set of points in a two-dimensional plane. Standard weighted least-squares fitting is only appropriate when there is a dimension along which the data points have negligible uncertainties, and another along which all the uncertainties can be described by Gaussians of known variance; these conditions are rarely met in practice. We consider cases of general, heterogeneous, and arbitrarily covariant two-dimensional uncertainties, and situations in which there are bad data (large outliers), unknown uncertainties, and unknown but expected intrinsic scatter in the linear relationship being fit. Above all we emphasize the importance of having a "generative model" for the data, even an approximate one. Once there is a generative model, the subsequent fitting is non-arbitrary because the model permits direct computation of the likelihood of the parameters or the posterior probability distribution. Construction of a posterior probability distribution is indispensible if there are "nuisance parameters" to marginalize away.

It is conventional to begin any scientific document with an introduction that explains why the subject matter is important. Let us break with tradition and observe that in almost all cases in which scientists fit a straight line to their data, they are doing something that is simultaneously *wrong* and *unnecessary*. It is wrong because circumstances in which a set of two