Python for Data Science

Introduction to UNIX

Dr. Ilkay Altintas

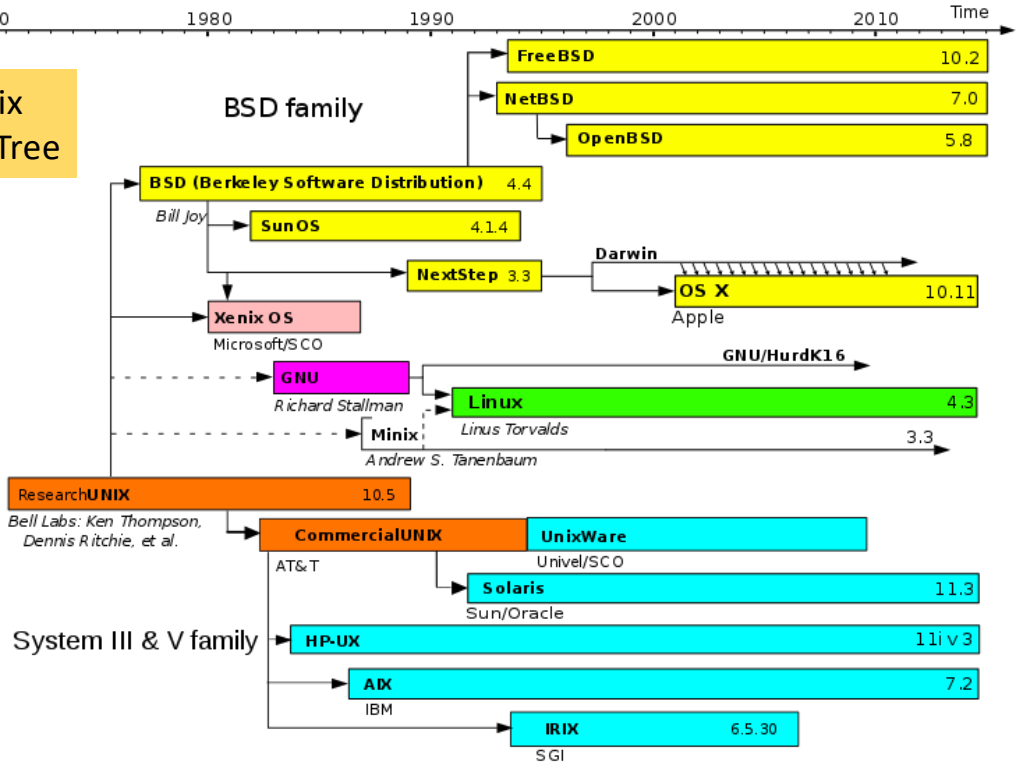
Getting started…

- Explain what UNIX is and why you need to know about it
- Understand files and processes
- Visualize the UNIX directory structure
- Start a UNIX terminal and type your first command

Why UNIX?

- Most operating systems are based on Unix
- Well adopted in industry
  - Back end of many data and compute systems
- Powerful development environment
- Expected competence for most industry jobs

A Unix Family Tree



Why is UNIX relevant to you as a Data Scientist?

UNIX provides many commands that can be used for data analysis.

- File management
- Data manipulation
- Simple scripting using command line pipelines

Engagement with other command line tools and applications.

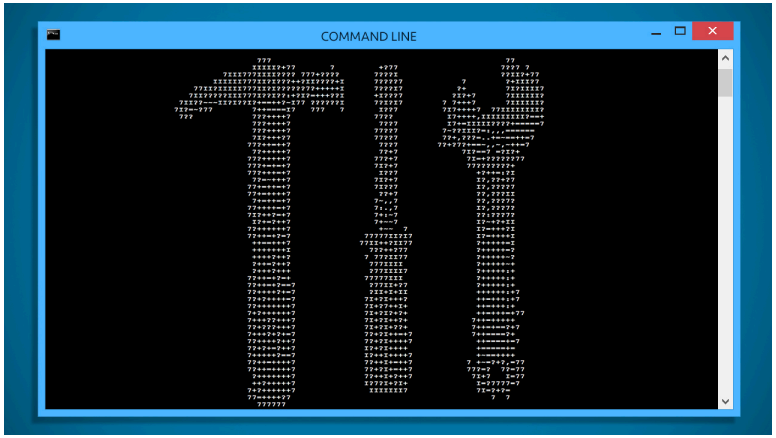
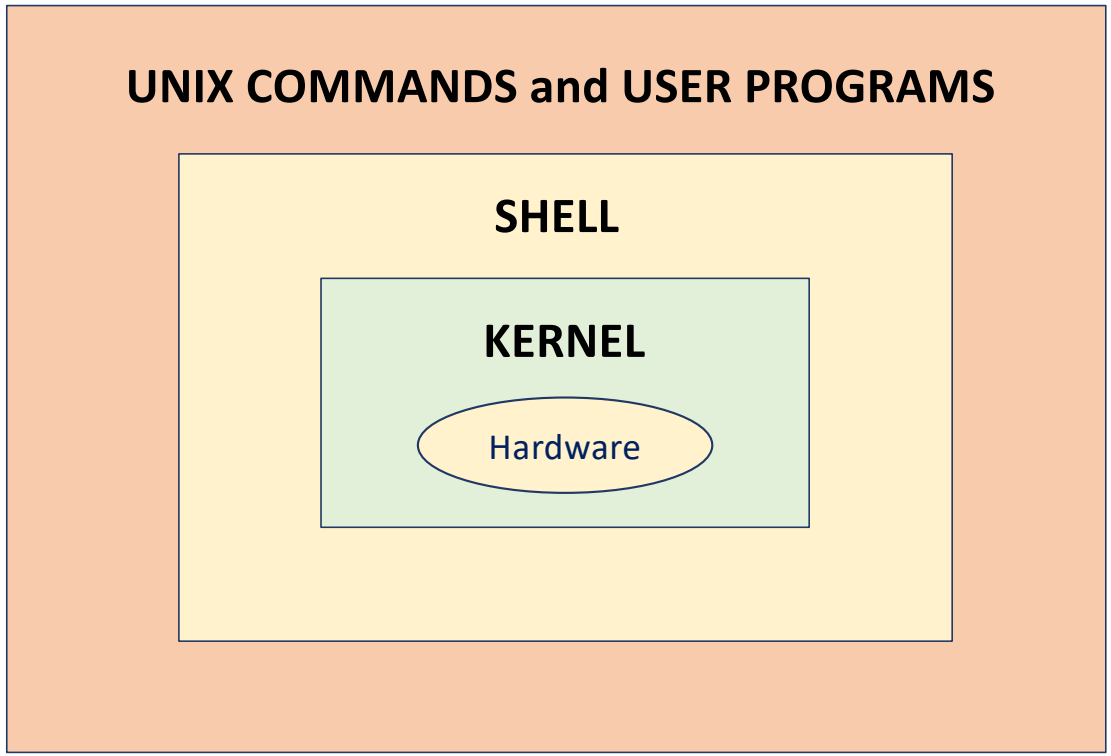
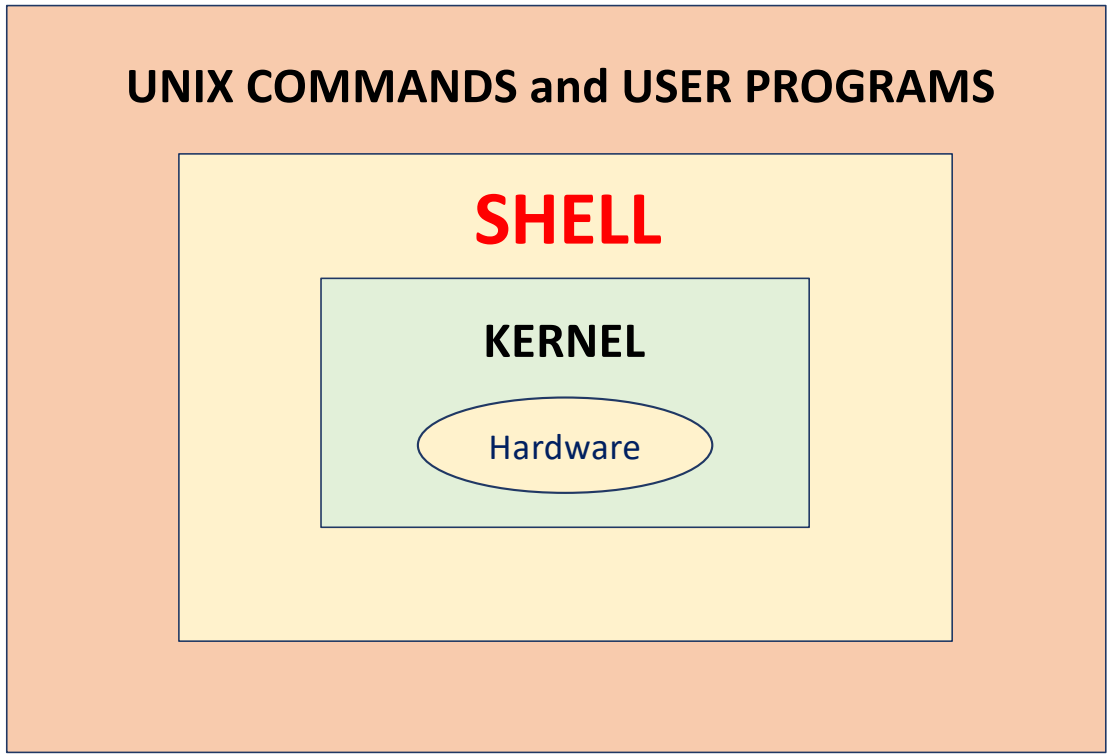
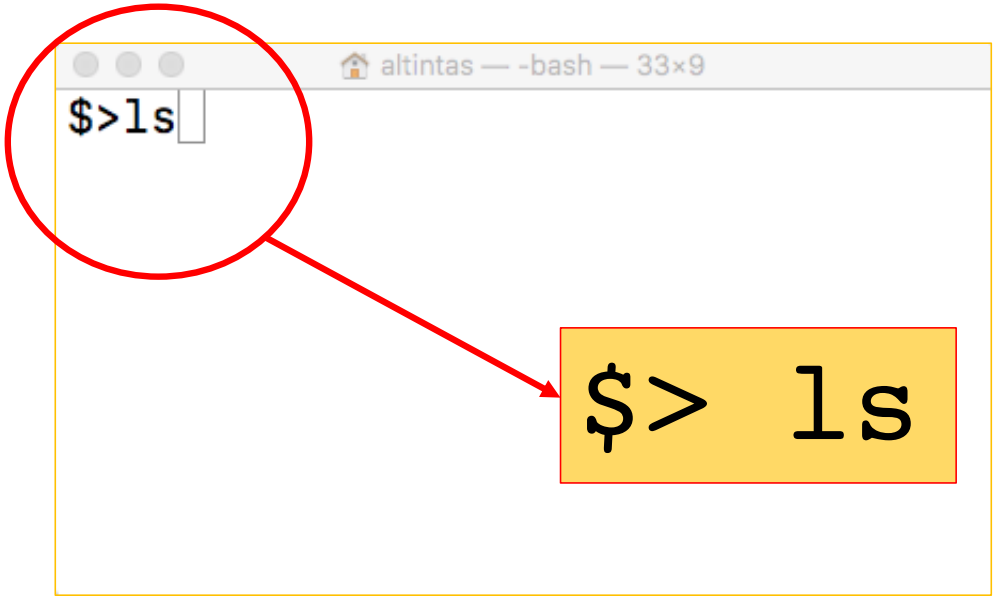


Image source: https://i.kinja-img.com/gawker-media/image/upload/t_original/gjnvguvdl3455rotbwg4.jpg


UNIX COMMANDS and USER PROGRAMS
SHELL
KERNEL
Hardware


UNIX COMMANDS and USER PROGRAMS
SHELL
KERNEL
Hardware

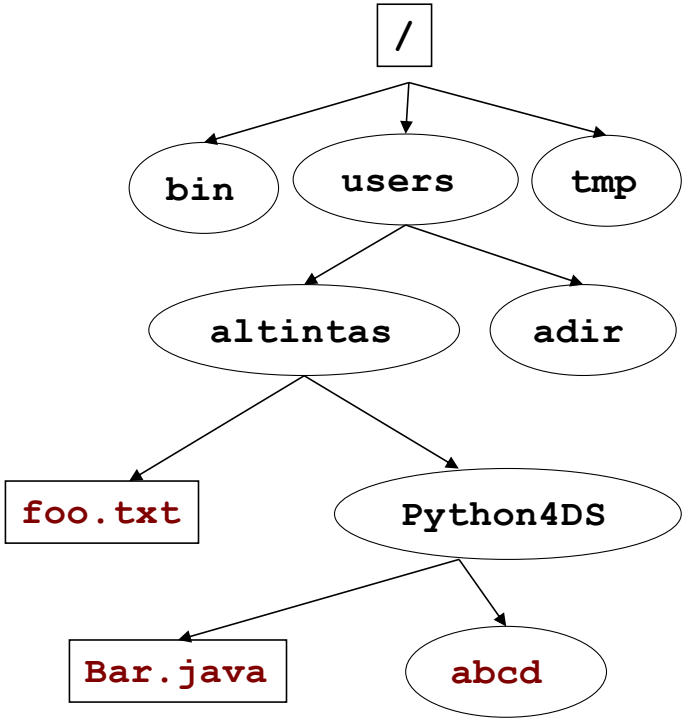
The UNIX Shell



Everything is a file or a process!

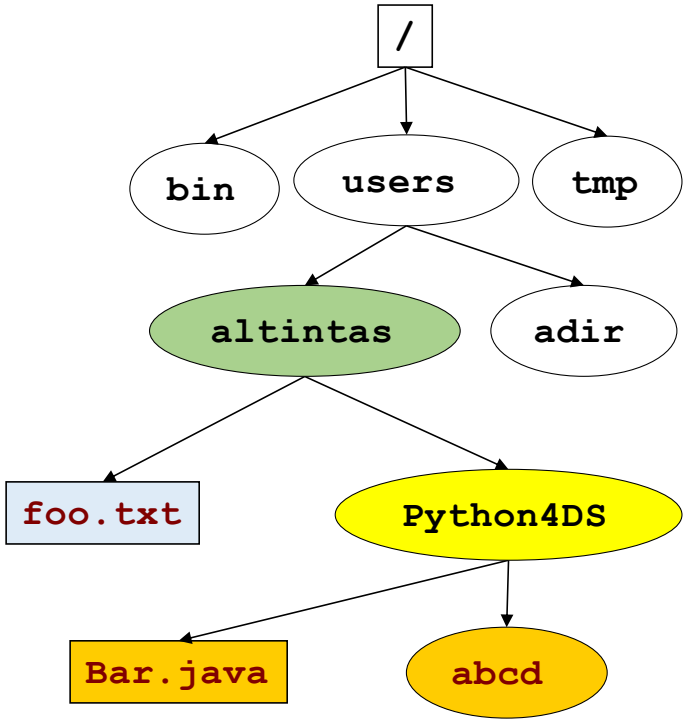
UNIX File Hierarchy

- Starts with the Root directory: /
- Directories may contain files or other directories
- Leads to a tree structure for the filesystem



Paths in UNIX

Separate directories by /

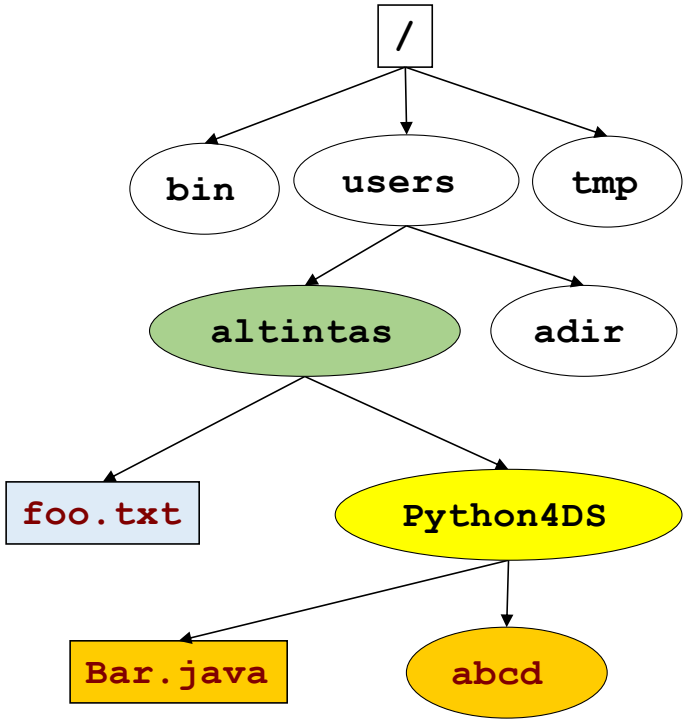


- Absolute path names
  - start at root and follow the tree

```
/users/altintas/foo.txt
```

Paths in UNIX

Separate directories by /



- Relative path
  - start at working directory
  - . refers to working directory

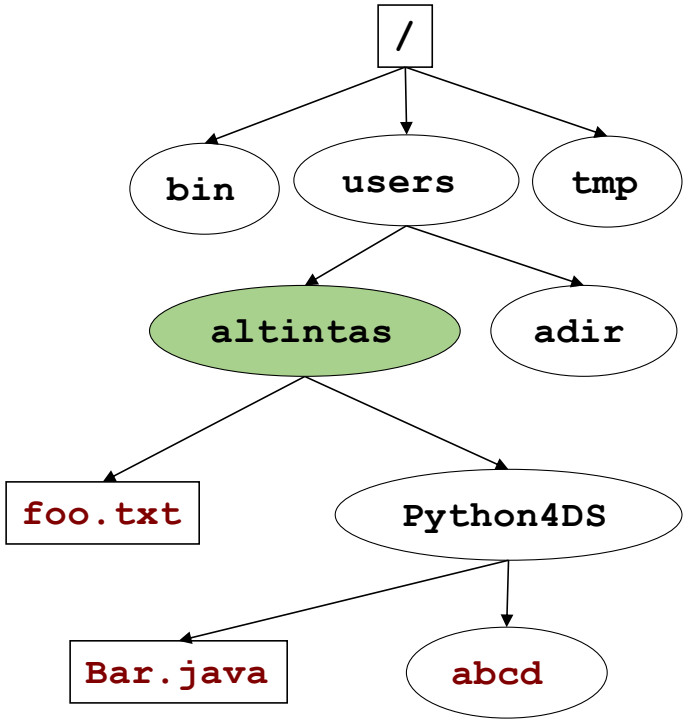
foo.txt ./foo.txt

  - . . refers to level above

../foo.txt

Paths in UNIX

- ~ (TILDE) in path names
  - `~/foo.txt`
  - `~altintas/foo.txt`



Caution when working with relative paths:

Make sure you check which directory you are in.

```
pwd
```

Let’s start a UNIX Shell and review what we learned…

Which of the following is an absolute path?

a) `/Users/altintas/temp/in.txt`
b) `temp/in.txt`
c) `../in.txt`
d) None of the above
e) More than one of the above

Basic UNIX Commands

- List files and directories in UNIX
- Change to a directory in UNIX
- Make new directories
- Explain the terms “standard in” and “standard out”
- Name six simple UNIX commands

Common Unix Commands

- **`ls`**
  - run with `-a` **option** to list hidden files (.*)
- **`mkdir`**
- **`cd`**
  - "." -- current directory,
  - ".." -- parent directory,
  - "~" -- home directory
- `*` -- wildcard

`stdin`, `stdout`, and `stderr`

- Standard input (`stdin`)
  - Usually from the keyboard
- Standard output (`stdout`)
  - Usually to the terminal
- Standard error (`stderr`)
  - Usually to the terminal

Common Unix Commands

- **`pwd`**
- **`cp`**
- **`mv`**
- **`rm`**
- **`clear`**
- **`cat`**
- **`man`**

REDIRECTION

stdout: TRY

- ls > MyFiles.text
- cat > temp1.txt
- cat >> temp1.txt

stdin:

- mail user@domain.com < message

```
bash-2.05$ cp temp.txt temp1.txt
bash-2.05$ mkdir dir1
bash-2.05$ cp temp.txt dir1
bash-2.05$ mkdir dir2
bash-2.05$ mv temp.txt dir2
```

How many copies of a file named **`temp.txt`** exist after running the commands above?

a) 1
b) 2
c) 3
d) 0
e) None of the above

Redirecting Standard IO

- Name all three UNIX file descriptors
- Utilize the redirect operators
- Explain the difference between “cat” and “more” commands

Standard Output and Standard Error

File Descriptor

FD 0: standard input, "`stdin`"

FD 1: standard output, "`stdout`"

FD 2: standard error, "`stderr`"

Redirecting Standard IO

- Redirect standard input to read from a file

```
$  command < somefile
```

- Redirect standard output to write to a file

```
$  command > afile1
```

- Redirect standard ouput explicitly as FD 1

```
$  command 1> afile2
```

- Redirect standard error to write to a file

```
$  command 2> afile3
```

Redirecting Standard IO

- Redirect standard error and standard output to different files

```
$ command 2> afile4 > afile5
```

- Redirect everything!

```
$ command > onefile 2> anotherfile < yetanotherfile
```

Redirecting Standard IO

- Redirect standard output to append to a file

```
$  command >> afile1
```

- Redirect standard error to append to a file

```
$  command 2>> afile3
```

- To redirect stderr and stdout to the same file

```
$  command > afile 2>&1
```

Let's review these concepts…

Which of the following is NOT correct?

a) `wc < shakespeare.txt > count.txt 2> log.txt`

b) `> count.txt 2> log.txt wc < shakespeare.txt`

c) `wc < shakespeare.txt 1> count.txt 2> log.txt`

d) None of the above

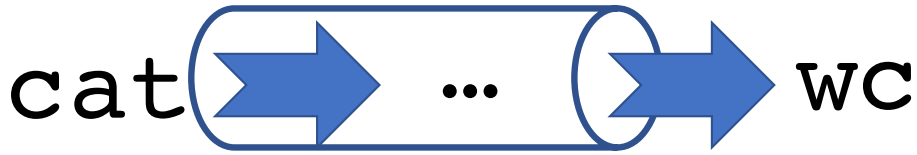
e) More than one of the above

Pipes and Filters

- Describe what a UNIX pipe is with an example
- Name five filter commands
- Differentiate between redirecting the standard out to a file and piping it to another UNIX command

A UNIX Pipe

```
cat foo.txt | wc
```



Filter Commands

- **`grep`**
- **`more`**
- **`less`**
- **`sort`**
- **`uniq`**

All transform their input!

Examples of Filtering

```
ls -la | more
cat filename | wc
man cat | grep file
ls -l | grep txt | wc
who | sort > current_users
```

Example filter commands on the UNIX shell…

Useful UNIX Commands for Data Science

- Sort, clean, cut and explore text data using Unix commands
- Plot data on the UNIX shell
- Use pipes and filters for quick data exploration in Unix

Exploratory Data Analysis in UNIX

- **cat**
- **grep**
- **wc**
- **sort**
- **uniq**

- **head**
- **tail**
- **cut**
- **sed**
- **find**

Exploring Data on the UNIX Shell

- What are the top words used in Shakespeare's works?
- Which users run the most processes on my unix system?
- Transform `fruits.txt` into all caps for further processing.

What are the top 15 words used in Shakespeare's works?

```
sed -e 's/ /\'$'\n/g' < shakespeare.txt | sort | sed '/^$/d' | uniq -c | sort -nr | head -15 > count_vs_words
```

```
sed -e 's/ /\'$'\n/g'
    < shakespeare.txt |
      sort |
        sed '/^$/d' |
          uniq -c |
            sort -nr |
              head -15
                  > count_vs_words
```

```
sed -e 's/\s/\n/g'
```

Which users run the most processes on my UNIX system?

```
ps -aef | cut -c3-5 | sort | uniq –c | sort -nr | head -3
```

```
ps -aef |
    cut -c3-5 |
        sort |
            uniq –c |
                sort -nr |
                    head -3
```

Transform fruits.txt into all caps for further processing.

```
tr '[a-z]' '[A-Z]' < fruits.txt > fruits-AllCaps.txt
```

```
tr '[a-z]' '[A-Z]'
    < fruits.txt
        > fruits-AllCaps.txt
```

Plotting the Results of Our Exploration

http://www.gnuplot.info/

gnuplot homepage

FAQ
Documentation
Demos
Download

Contributed scripts
External Links
Tutorials, learning, and help
Books

Gnuplot is a portable command-line driven graphing utility for Linux, OS/2, MS Windows, OSX, VMS, and many other platforms. The source code is copyrighted but freely distributed (i.e., you don't have to pay for it). It was originally created to allow scientists and students to visualize mathematical functions and data interactively, but has grown to support many non-interactive uses such as web scripting. It is also used as a plotting engine by third-party applications like Octave. Gnuplot has been supported and under active development since 1986.

Gnuplot supports many different types of 2D and 3D plots
Here is a Gallery of demos.

Gnuplot supports many different types of output

interactive screen display:	cross-platform (Qt, wxWidgets, x11) or system-specific (MS Windows, OS/2)
direct output to file:	postscript (including eps), pdf, png, gif, jpeg, LaTeX, metafont, emf, svg, ...
mouseable web display formats:	HTML5, svg

Version 5.0 Release
- Download from SourceForge
- Release Notes
- User Manual (PDF)
- version 5.0 demo gallery
- contributed executables for OSX

The Development version is gnuplot 5.1
- New features are being added regularly. You are welcome to build gnuplot from the CVS source code. Instructions here. More instructions here.
- Version 5.1 Documentation (PDF)
- Version 5.1 demo gallery

Version 4.6
- Download from SourceForge
- Release Notes
- User Manual (PDF)
- version 4.6 demo gallery

Release History
- gnuplot 5.0 January 2015
  gnuplot 5.0.6 (latest) March 2017
- gnuplot 4.6 March 2012
  gnuplot 4.6.6 (final) September 2014
- gnuplot 4.4 March 2010

Gnuplot in Action
Second Edition
by Philipp K. Janert

Updated for gnuplot 5
Manning Publications (2016)

gnuplot Cookbook
by Lee Phillips

Packt Publishing (2012)
ISBN : 184951724X
ISBN-13 : 9781849517249

Let’s get started with our live UNIX session…