CGS600A: Computational tools in Cog. Sc.

Classes: W, F 12.00-13.15, TB201

Harish Karnick (hk@cse.iitk.ac.in or hk@iitk.ac.in), KD-318 (Old CS bldg.)

About the course: Content

- Use of the command line in Linux/Unix systems and introduction to some useful tools.
- Programming in Python - so as to reach a reasonable level of proficiency.
- Very basic probability and statistics.
- Collecting and analyzing data.
- Testing hypotheses.
- Simple models for some cognitive phenomena.

About the course: Logistics

Programming/data analysis/modelling is like

About the course: Logistics

Programming/data analysis/modelling is like swimming you can only learn it by doing it.

- Lab assignments almost every week. You can do them on your laptop or using the systems in the CGS lab or the institute. They will have to be submitted and will be graded.
- Material discussed in class, assignments, other reference material, links, etc. will be put up on the course website. `www.cse.iitk.ac.in/users/hk/cgs600`.
- The main references we will use for programming and the basics of statistics and hypothesis testing are:
  - John Guttag, Introduction to Computation and Programming Using Python with Application to Understanding Data, MIT Press, 2016.
  - D S Moore, G P McCabe, B A Craig, Introduction to the Practice of Statistics, 8th Ed., WH Freeman and Co., 2014.
  - Other internet based resources.

About the course: Evaluation

Item	Weight
Labs	20%
Midsem	20%
Endsem	30%
Project	30%

- Project: build a computational model for categorization - see course website for details.
- It will involve choosing one of the models/formal theory and write a program that implements the model/theory. Your model should do one or more of the following: a) explains existing behavioural data directly concerned with categorization b)
- Evaluation will be based on presentation cum demo of your project and two written reports the first roughly in the middle of the semester and the second at the end of the sem.
- Credit breakup: 30=5 (mid-term report) + 20 (presentation+demo) + 5 (final report).

Linux 1: command line/shell[1]

- The command line or shell (called Bash) is a text based command interpreter. (Other shells are also available - Bash is the default in Linux and widely used).
- The shell can also be used to execute commands in a file - typically called scripts. It has basic programming constructs like assignment, conditionals and looping so scripts can be full fledged programs. You can study shell scripting on your own using the references. It is not part of the course but very useful to know.

[1] V Gedris, An Introduction to the Linux Command Shell For Beginners, `http://vic.dyndns.org`

Linux 2: file system/directory structure

The directory structure or filesystem is organized as a tree.

Directory	Description
/	The base or root of the filesystem. Other dirs, files, drives, devices are children of root. .
/bin	Essential command binaries (programs) are stored here (bash, ls, mount, tar, etc.)
/boot	Static files of the boot loader.
/dev	Device files. In Linux, hardware devices are acceessd just like other files, and they are kept under this directory.
/etc	Host-specific system configuration files.
/home	Location of users' personal home directories (e.g. /home/hk).
/lib	Essential shared libraries and kernel modules.
/proc	Process information pseudo-filesystem. An interface to kernel data structures.
/root	The root (superuser) home directory.
/sbin	Essential system binaries (fdisk, fsck, init, etc).
/tmp	Temporary files. All users have permission to place temporary files here.
/usr	The base directory for most shareable, read-only data (programs, libraries, documentation, and much more).
/var	Variable data. Logs, spools, lock files etc.

Linux 3: special characters

Character	*Description*
\	Escape character. If you want to reference a special character, you must "escape" it with a backslash first. Example: `touch /tmp/filename\*`
/	Directory separator, used to separate a string of directory names. Example: `/usr/src/linux`
.	Current directory. Can also "hide" files when it is the first character in a filename.
..	Parent directory
~	User's home directory
*	Represents 0 or more characters in a filename, or by itself, all files in a directory. Example: `pic*2002` can represent the files `pic2002`, `picJanuary2002`, `picFeb292002`, etc.
?	Represents a single character in a filename. Example: `hello?.txt` can represent `hello1.txt`, `helloz.txt`, but not `hello22.txt`
[ ]	Can be used to represent a range of values, e.g. `[0-9]`, `[A-Z]`, etc. Example: `hello[0-2].txt` represents the names `hello0.txt`, `hello1.txt`, and `hello2.txt`
|	"Pipe". Redirect the output of one command into another command. Example: `ls | more`
>	Redirect output of a command into a new file. If the file already exists, over-write it. Example: `ls > myfiles.txt`
>>	Redirect the output of a command onto the end of an existing file. Example: `echo "Mary 555-1234" >> phonenumbers.txt`
<	Redirect a file as input to a program. Example: `more < phonenumbers.txt`
;	Command separator. Allows you to execute multiple commands on a single line. Example: `cd /var/log ; less messages`
&&	Command separator as above, but only runs the second command if the first one finished without errors. Example: `cd /var/logs && less messages`
&	Execute a command in the background, and immediately get your shell back. Example: `find / -name core > /tmp/corefiles.txt &`