Introduction to Data Science

1MS041, 2024

01. BASH Unix Shell

1. Dropping into BASH (Unix Shell) and using basic Shell commands
   - `pwd` --- print working directory
   - `ls` --- list files in current working directory
   - `mkdir` --- make directory
   - `cd` --- change directory
   - `man ls` --- manual pages for any command
   - `head` --- show the first lines of a file
2. Grabbing files from the internet using `curl`

In [ ]:
```
def showURL(url, ht=500):
    """Return an IFrame of the url to show in notebook with height ht"""
    from IPython.display import IFrame
    return IFrame(url, width='95%', height=ht)
showURL('https://en.wikipedia.org/wiki/Bash_(Unix_shell)',400)
```

1. Dropping into BASH (Unix Shell)

Using `%%sh` in a code cell we can access the BASH (Unix Shell) command prompt.

Let us `pwd` or print working directory.

In [ ]:
```
%%sh
pwd
```

In [ ]:
```
%%sh
# this is a comment in BASH shell as it is preceeded by '#'
ls # list the contents of this working directory
```

In [ ]:
```
%%sh
mkdir -p mydir
```

In [ ]:
```
%%sh
cd mydir
```

```
pwd
ls -al
```

In [ ]:
```
%%sh
pwd
```

"Use the source" by `man`-ning the unknown `command`

By evaluating the next cell, you are using the `man`ual pages to find more about the command `ls`. You can learn more about any command called `command` by typing `man command` in the BASH shell.

The output of the next cell with command `man ls` will look something like the following:

```
LS(1)                            User Commands
LS(1)

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List  information  about  the FILEs (the current
directory by default).
       Sort entries alphabetically if none of -cftuvSUX
nor --sort  is  speci-
       fied.

       Mandatory  arguments  to  long  options are
mandatory for short options
       too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..
...
...
...
   Exit status:
       0      if OK,
```

```
       1      if minor problems (e.g., cannot access
    subdirectory),

       2      if serious trouble (e.g., cannot access
    command-line argument).

    AUTHOR
       Written by Richard M. Stallman and David MacKenzie.

    REPORTING BUGS
       GNU coreutils online help:
    <http://www.gnu.org/software/coreutils/>
       Report ls translation bugs to
    <http://translationproject.org/team/>

    COPYRIGHT
       Copyright © 2017 Free Software Foundation, Inc.
    License  GPLv3+:  GNU
       GPL version 3 or later
    <http://gnu.org/licenses/gpl.html>.
       This  is  free  software:  you  are free to change
    and redistribute it.
       There is NO WARRANTY, to the extent permitted by
    law.

    SEE ALSO
       Full documentation at:
    <http://www.gnu.org/software/coreutils/ls>
       or available locally via: info '(coreutils) ls
    invocation'

    GNU coreutils 8.28              January 2018
    LS(1)
```

In [ ]:
```
%%sh
## uncomment by removing '#' in the next line and try executing this cell
# man ls
```

2. Grabbing files from internet using curl

In [ ]:
```
%%sh
cd mydir
curl -O http://lamastex.org/datasets/public/SOU/sou/20170228.txt
```

In [ ]:
```
%%sh
ls mydir/
```

In [ ]:
```
%%sh
cd mydir/
```

```
head 20170228.txt
```

To have more fun with all SOU addresses

Do the following:

In [ ]:
```
%%sh
mkdir -p mydir # first create a directory called 'mydir'
cd mydir # change into this mydir directory
rm -f sou.tar.gz # remove any file in mydir called sou.tar.gz
curl -O http://lamastex.org/datasets/public/SOU/sou.tar.gz
```

In [ ]:
```
%%sh
pwd
ls -lh mydir
```

In [ ]:
```
%%sh
cd mydir
tar zxvf sou.tar.gz
```

After running the above two cells, you should have all the SOU (State of Union) addresses. By evaluating the next cell's `ls ...` command you should see the SOU files like the following:

```
total 11M
-rw------- 1 raazesh raazesh 6.6K Feb 18  2016
17900108.txt
-rw------- 1 raazesh raazesh 8.3K Feb 18  2016
17901208.txt
-rw------- 1 raazesh raazesh  14K Feb 18  2016
17911025.txt
...
...
...
-rw------- 1 raazesh raazesh  39K Feb 18  2016
20140128.txt
-rw------- 1 raazesh raazesh  38K Feb 18  2016
20150120.txt
-rw------- 1 raazesh raazesh  31K Feb 18  2016
20160112.txt
```

In [ ]:
```
%%sh
ls -lh mydir/sou
```

In [ ]:
```
%%sh
head mydir/sou/17900108.txt
```

In [ ]:
```
%%sh
```

```
head mydir/sou/20160112.txt
```

An interesting analysis of the textual content of the *State of the Union (SoU)* addresses by all US presidents was done in:

- Alix Rule, Jean-Philippe Cointet, and Peter S. Bearman, Lexical shifts, substantive changes, and continuity in State of the Union discourse, 1790–2014, PNAS 2015 112 (35) 10837-10844; doi:10.1073/pnas.1512221112.

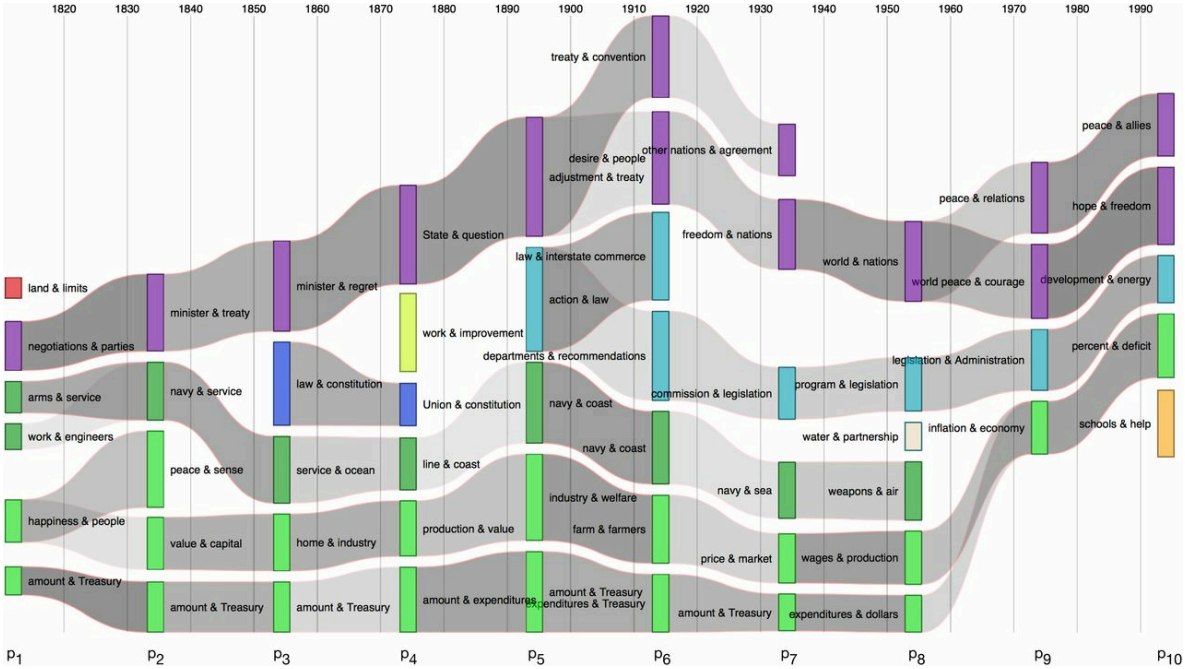

Image source

Fig. 5. A river network captures the flow across history of US political discourse, as perceived by contemporaries. Time moves along the x axis. Clusters on semantic networks of 300 most frequent terms for each of 10 historical periods are displayed as vertical bars. Relations between clusters of adjacent periods are indexed by gray flows, whose density reflects their degree of connection. Streams that connect at any point in history may be considered to be part of the same system, indicated with a single color.

You *will be able to carry out such analyses and/or critically reflect on the mathematical statistical assumptions made in such analyses, as you learn more during your programme of study after successfully completing this course.*

How was the `sou.tgz` file created?

If you are curious, read: http://lamastex.org/datasets/public/SOU/README.md.

Briefly, this is how a website with SOU was scraped by Paul Brouwers and adapted by

Raazesh Sainudiin. A data scientist, and more generally a researcher interested in making statistical inference from data that is readily available online in a particular format, is expected to be comfortable with such *web-scraping tasks* (which can be done in more gracious and robust ways using specialised Python libraries). Such tasks also known as *Extract-Load-Transform (ELT)* operations are often time-consuming, expensive and the necessary first step towards extracting value from data.

A bit of bash and lynx to achieve the scraping of the state of the union addresses of the US Presidents,

by Paul Brouwers

The code below is mainly there to show how the text content of each state of the union address was scraped from the following URL:

- http://stateoftheunion.onetwothree.net/texts/index.html

Such data acquisition tasks is usually the first and cucial step in a data scientist's workflow.

We have done this and put the data in the distributed file system for easy loading into our notebooks for further analysis. This keeps us from having to install unix programs like `lynx`, `sed`, etc. that are needed in the shell script below.

```
for i in $(lynx --dump
http://stateoftheunion.onetwothree.net/texts/index.html |
grep texts | grep -v index | sed 's/.*http/http/') ; do
lynx --dump $i | tail -n+13 | head -n-14 | sed 's/^\s\+//'
| sed -e ':a;N;$!ba;s/\(.\)\n/\1 /g' -e 's/\n/\n\n/' >
$(echo $i | sed 's/.*\([0-9]\{8\}\).*/\1/').txt ; done
```

Or in a more atomic form:

```
for i in $(lynx --dump
http://stateoftheunion.onetwothree.net/texts/index.html \

        | grep texts \

        | grep -v index \

        | sed 's/.*http/http/')

do

        lynx --dump $i \
```

```
                    | tail -n+13 \

                    | head -n-14 \

                    | sed 's/^\s\+//' \

                    | sed -e ':a;N;$!ba;s/\(.\)\n/\1 /g' -e
's/\n/\n\n/' \

                   > $(echo $i | sed 's/.*\([0-9]\
{8\}\).*/\1/').txt

done
```

If you have time and are curious how each of the components in the above pipeline via `|` operators work, try to read `man echo`, `man sed`, `man grep`, `man head`, `man tail`, and `man lynx` or `lynx --help`. If a command like `lynx` is not in your system, then you can install it with some work (mostly googling).

In [ ]:
```
%%sh
## uncomment by removing '#' in the next line and try executing this cell
#lynx --help
```

So using `lynx` is not that difficult. Suppose you want to dump the contents of https://lamastex.github.io/research/#available-student-projects to `stdout` or standard out, we can do the following:

In [ ]:
```
%%sh
## uncomment by removing '#' in the next line and try executing this cell
#lynx --dump https://lamastex.github.io/research/#available-student-proje
```

Hopefully, you had fun with BASH! Now let us put BASH to use.