

# Methods for unstructured data

Lecture 1: Tools for Scientific Programming

---

Helge Liebert

# What is this lecture about?

- Most research in economics now involves scientific programming.
- Introduce tools and ideas that may make daily research tasks easier.
- Many of these have been developed by other scientists or IT professionals. Some are decades old, others are novel.
- Focus on data processing and reproducible research.
- Text processing is a common task in research and industry.
- Working with text data requires understanding of some practical techniques.

# Contents

1. Economics and computation
2. Programming languages
3. Version control
4. Command line interface
5. Regular expressions

## Economics and computation

---

# Economics and computer science

- Technical aspects rarely part of the economics curriculum.
- Data management is not taught in introductory econometrics.
- The days of entering data in spreadsheets are largely gone.
- Even before the recent advent of “big data”.
- Scientific programming common in other sciences before it gained a more prominent role in economics.

# Economics and computer science

- Computer science often involves processing data.
- Most problems you are likely to encounter have been solved.
- Tools and concepts that economists can profit from.
  - Remote servers, databases, version control, accessing APIs for data, text processing and analysis, geospatial analysis, OCR, automation, ...
  - Time complexity of algorithms, computational cost, databases, ...

# Survey

- Operating systems - Windows, MacOS, Linux?
- Scientific programming - Stata, R, Python, Matlab/Octave, Julia, ...?
- General purpose programming languages?
- Servers, shell scripting, CLI?
- (Relational) databases - SQL?
- Version control - Git?

# Topics

- General points about scientific programming and working with computers.
- Superficial and incomplete. A stand-alone course on Programming Practices for Research could be devoted to these topics.
- Brief intro to get you going and help identify appropriate tools for a problem.
- Point you towards resources and explain their general concepts.
- Leave you marginally more computer literate.
- Languages, tools, version control.
- Unix shell and the command line.
- Regular expressions.

## Programming languages

---

# Which language to choose?

- *It depends.* Choice is use-case- and taste-specific.
- Anything can be done in *any* language. Convenience varies.
- Concepts and toolkits transfer easily most of the time.
- Trade-off: Prior knowledge vs. task suitability.
- Languages which are great for specific scientific purposes may not be well-suited for other tasks.
- Ease of exploratory analysis vs. ease of deployment in production.
- Never re-invent the wheel.

# Possible options

- Specialized languages: R, MATLAB/Octave, Stata, Gauss, Julia, ...
- General-purpose languages: Python, Perl, Ruby, C, ...
- Choose a high-level, dynamic, interpreted language unless you are sure you require the extra speed of a compiled language.
- Ideally free and open source. Popular is typically better.
- Research ex ante which libraries are mature and best for solving your specific problem.

- R is the major statistical programming language.
- It is free, used in many sciences and in industry. Good documentation.
- New models are frequently published and implemented first in R.
- Having data processing and analysis in the same language is nice.
- Good library support for common tools (e.g. databases, regular expressions).
- Specific tasks for which high-level wrapper functions are not available may be very cumbersome.
- In recent years, R development has been very active and libraries exist for almost anything.

# Python

- General-purpose programming language, supports object-oriented programming.
- Reads like english. Explicit and clear. Whitespace matters, no braces.  
*(“There should be one obvious way to do it”.)*
- Used extensively in industry and sciences. Good documentation.
- Libraries for almost anything.
- Many science-related libraries exist for other languages, but rarely are they as mature.
- Good and growing support for statistical modeling.
- A bit less suited for interactive data work (but more so for deployment in production).

# This lecture

- The lab sessions utilize R.
- Any task covered by this lecture can be accomplished using R or Python (augmented by shell programs).
- R, Python, SQL and knowing your way around a terminal are highly valued skills on the job market.
- Rule-of-thumb recommendation:
  - Simple data analysis/small text corpora:  
Stick with R. Augment with other tools where required.
  - More involved data processing/larger text corpora:  
Go with Python. You can still analyze data in R.
  - ... and whatever program your colleagues are using.

## Why not Stata, Matlab, Gauss or similar?

- Advantage: Many domain-specific models supported.
- Less support for almost anything else.
- Much less flexible for anything not to do with data analysis or numerics.
- Difficult to deploy on a server. Often tied to a GUI.
- Less popular, smaller userbase. Proprietary and expensive.
- You can still rely on them for estimation after your data is clean.

## A note on text editing

- A script is a set of *plain text* instructions, fed to an interpreter.
- Editing is independent from running code.
- R scripts usually have the suffix `.r`, Python `.py`, Shell `.sh`.
- Proficiency in a text editor makes working with text easier and faster.
- Too many options to list. All are better than Notepad.
- A few options: VS Code, Sublime Text, Atom, Notepad++, ...
- Learning Vim or Emacs requires you to invest some time.
- Text editors allow you to integrate your work and edit text efficiently.
- Sometimes IDEs with GUI may be more convenient (e.g. Rstudio, PyCharm).
- Features: Regex search and replace, grep search, diff, syntax checking, formatting, completion, persistent undo, documentation lookup ...

# A possible setup

```
emacs@helge-x250 ~
1 \begin{frame}[A note on operating systems]
2 \begin{itemize}
3 \item Mac OS or Linux offer built-in access to a Unix shell (Bash).
4 \item Further software is managed via a package management system and
5 distributed via software repositories.
6 \item On Linux, use your package manager to install anything you require.
7 \item On Mac OS, familiarize yourself with Homebrew. Install iterna if you
8 want a fancier terminal.
9 \end{itemize}
10 \begin{itemize}
11 \item For Windows, many tools are not available or cumbersome to use.
12 \item Using a terminal on Windows is a nightmare.
13 \item Windows does not provide proper access to a Unix shell.
14 \item Even reliably installing Python was a chore until recently (now use
15 Anaconda.)
16 \end{itemize}
17 \end{frame}
382 [
```

```
2 \begin{frame}[Fragile](Command line interpreters and shells)
3 \begin{itemize}
4 \item An interface that lets you interact with your computer.
5 \item A CLI using a programming language that allows
6 command-line interaction.
7 \item Unix-based operating systems (Linux, Mac OS) have Bash pre-installed.
8 \item Windows has cmd (or PowerShell). These are not a viable
9 replacement. Cygwin or WSL may be. Git Bash is incomplete.
10 \end{itemize}
11 \begin{itemize}
12 \item Some examples:
13 \begin{minted}[fontsize=footnotesize]{bash}
14 cd somewhere\subdir # navigate to a folder
15 cd .. # navigate to parent directory
16 cd # return to your home folder
17 ls # list directory contents
18 R & start R's console
19 \end{minted}
20 \item \textit{textit}(\textit{CTRL + c}) aborts a process, \textit{textit}(\textit{CTRL + d}) quits.
21 \end{itemize}
22 \end{frame}
23 [
```

```
25 \begin{frame}[Fragile](Examples)
26 \begin{itemize}
27 \item Some examples:
28 \begin{minted}[fontsize=footnotesize]{bash}
29 vim myscript.r # edit your R script with vim
30 R -f myscript.r # execute your R script
31 python myscript.py # execute your python script
32 git add myscript.py # stage file for version control
33 git commit myscript.py # stage file for version control
34 man ssh # display manual pages for the ssh program
35 ssh myusername@19.438.14.673 # secure shell login to your remote server
36 \end{minted}
37 \end{itemize}
38 \begin{itemize}
39 \item Sounds tedious? It is. But it can also be extremely powerful.
40 \end{itemize}
41 \end{frame}
42 [
```

```
* 151-152-153-154-155-156-157-158-159-160-161-162-163-164-165-166-167-168-169-170-171-172-173-174-175-176-177-178-179-180-181-182-183-184-185-186-187-188-189-190-191-192-193-194-195-196-197-198-199-200-201-202-203-204-205-206-207-208-209-210-211-212-213-214-215-216-217-218-219-220-221-222-223-224-225-226-227-228-229-230-231-232-233-234-235-236-237-238-239-240-241-242-243-244-245-246-247-248-249-250-251-252-253-254-255-256-257-258-259-260-261-262-263-264-265-266-267-268-269-270-271-272-273-274-275-276-277-278-279-280-281-282-283-284-285-286-287-288-289-290-291-292-293-294-295-296-297-298-299-299-300-301-302-303-304-305-306-307-308-309-310-311-312-313-314-315-316-317-318-319-320-321-322-323-324-325-326-327-328-329-330-331-332-333-334-335-336-337-338-339-340-341-342-343-344-345-346-347-348-349-350-351-352-353-354-355-356-357-358-359-360-361-362-363-364-365-366-367-368-369-370-371-372-373-374-375-376-377-378-379-380-381-382-383-384-385-386-387-388-389-390-391-392-393-394-395-396-397-398-399-399-400-401-402-403-404-405-406-407-408-409-410-411-412-413-414-415-416-417-418-419-420-421-422-423-424-425-426-427-428-429-430-431-432-433-434-435-436-437-438-439-440-441-442-443-444-445-446-447-448-449-4410-4411-4412-4413-4414-4415-4416-4417-4418-4419-4420-4421-4422-4423-4424-4425-4426-4427-4428-4429-4430-4431-4432-4433-4434-4435-4436-4437-4438-4439-4440-4441-4442-4443-4444-4445-4446-4447-4448-4449-44410-44411-44412-44413-44414-44415-44416-44417-44418-44419-44420-44421-44422-44423-44424-44425-44426-44427-44428-44429-44430-44431-44432-44433-44434-44435-44436-44437-44438-44439-44440-44441-44442-44443-44444-44445-44446-44447-44448-44449-444410-444411-444412-444413-444414-444415-444416-444417-444418-444419-444420-444421-444422-444423-444424-444425-444426-444427-444428-444429-444430-444431-444432-444433-444434-444435-444436-444437-444438-444439-444440-444441-444442-444443-444444-444445-444446-444447-444448-444449-4444410-4444411-4444412-4444413-4444414-4444415-4444416-4444417-4444418-4444419-4444420-4444421-4444422-4444423-4444424-4444425-4444426-4444427-4444428-4444429-4444430-4444431-4444432-4444433-4444434-4444435-4444436-4444437-4444438-4444439-4444440-4444441-4444442-4444443-4444444-4444445-4444446-4444447-4444448-4444449-44444410-44444411-44444412-44444413-44444414-44444415-44444416-44444417-44444418-44444419-44444420-44444421-44444422-44444423-44444424-44444425-44444426-44444427-44444428-44444429-44444430-44444431-44444432-44444433-44444434-44444435-44444436-44444437-44444438-44444439-44444440-44444441-44444442-44444443-44444444-44444445-44444446-44444447-44444448-44444449-444444410-444444411-444444412-444444413-444444414-444444415-444444416-444444417-444444418-444444419-444444420-444444421-444444422-444444423-444444424-444444425-444444426-444444427-444444428-444444429-444444430-444444431-444444432-444444433-444444434-444444435-444444436-444444437-444444438-444444439-444444440-444444441-444444442-444444443-444444444-444444445-444444446-444444447-444444448-444444449-4444444410-4444444411-4444444412-4444444413-4444444414-4444444415-4444444416-4444444417-4444444418-4444444419-4444444420-4444444421-4444444422-4444444423-4444444424-4444444425-4444444426-4444444427-4444444428-4444444429-4444444430-4444444431-4444444432-4444444433-4444444434-4444444435-4444444436-4444444437-4444444438-4444444439-4444444440-4444444441-4444444442-4444444443-4444444444-4444444445-4444444446-4444444447-4444444448-4444444449-44444444410-44444444411-44444444412-44444444413-44444444414-44444444415-44444444416-44444444417-44444444418-44444444419-44444444420-44444444421-44444444422-44444444423-44444444424-44444444425-44444444426-44444444427-44444444428-44444444429-44444444430-44444444431-44444444432-44444444433-44444444434-44444444435-44444444436-44444444437-44444444438-44444444439-44444444440-44444444441-44444444442-44444444443-44444444444-44444444445-44444444446-44444444447-44444444448-44444444449-444444444410-444444444411-444444444412-444444444413-444444444414-444444444415-444444444416-444444444417-444444444418-444444444419-444444444420-444444444421-444444444422-444444444423-444444444424-444444444425-444444444426-444444444427-444444444428-444444444429-444444444430-444444444431-444444444432-444444444433-444444444434-444444444435-444444444436-444444444437-444444444438-444444444439-444444444440-444444444441-444444444442-444444444443-444444444444-444444444445-444444444446-444444444447-444444444448-444444444449-4444444444410-4444444444411-4444444444412-4444444444413-4444444444414-4444444444415-4444444444416-4444444444417-4444444444418-4444444444419-4444444444420-4444444444421-4444444444422-4444444444423-4444444444424-4444444444425-4444444444426-4444444444427-4444444444428-4444444444429-4444444444430-4444444444431-4444444444432-4444444444433-4444444444434-4444444444435-4444444444436-4444444444437-4444444444438-4444444444439-4444444444440-4444444444441-4444444444442-4444444444443-4444444444444-4444444444445-4444444444446-4444444444447-4444444444448-4444444444449-44444444444410-44444444444411-44444444444412-44444444444413-44444444444414-44444444444415-44444444444416-44444444444417-44444444444418-44444444444419-44444444444420-44444444444421-44444444444422-44444444444423-44444444444424-44444444444425-44444444444426-44444444444427-44444444444428-44444444444429-44444444444430-44444444444431-44444444444432-44444444444433-44444444444434-44444444444435-44444444444436-44444444444437-44444444444438-44444444444439-44444444444440-44444444444441-44444444444442-44444444444443-44444444444444-44444444444445-44444444444446-44444444444447-44444444444448-44444444444449-444444444444410-444444444444411-444444444444412-444444444444413-444444444444414-444444444444415-444444444444416-444444444444417-444444444444418-444444444444419-444444444444420-444444444444421-444444444444422-444444444444423-444444444444424-444444444444425-444444444444426-444444444444427-444444444444428-444444444444429-444444444444430-444444444444431-444444444444432-444444444444433-444444444444434-444444444444435-444444444444436-444444444444437-444444444444438-444444444444439-444444444444440-444444444444441-444444444444442-444444444444443-444444444444444-444444444444445-444444444444446-444444444444447-444444444444448-444444444444449-4444444444444410-4444444444444411-4444444444444412-4444444444444413-4444444444444414-4444444444444415-4444444444444416-4444444444444417-4444444444444418-4444444444444419-4444444444444420-4444444444444421-4444444444444422-4444444444444423-4444444444444424-4444444444444425-4444444444444426-4444444444444427-4444444444444428-4444444444444429-4444444444444430-4444444444444431-4444444444444432-4444444444444433-4444444444444434-4444444444444435-4444444444444436-4444444444444437-4444444444444438-4444444444444439-4444444444444440-4444444444444441-4444444444444442-4444444444444443-4444444444444444-4444444444444445-4444444444444446-4444444444444447-4444444444444448-4444444444444449-44444444444444410-44444444444444411-44444444444444412-44444444444444413-44444444444444414-44444444444444415-44444444444444416-44444444444444417-44444444444444418-44444444444444419-44444444444444420-44444444444444421-44444444444444422-44444444444444423-44444444444444424-44444444444444425-44444444444444426-44444444444444427-44444444444444428-44444444444444429-44444444444444430-44444444444444431-44444444444444432-44444444444444433-44444444444444434-44444444444444435-44444444444444436-44444444444444437-44444444444444438-44444444444444439-44444444444444440-44444444444444441-44444444444444442-44444444444444443-44444444444444444-44444444444444445-44444444444444446-44444444444444447-44444444444444448-44444444444444449-444444444444444410-444444444444444411-444444444444444412-444444444444444413-444444444444444414-444444444444444415-444444444444444416-444444444444444417-444444444444444418-444444444444444419-444444444444444420-444444444444444421-444444444444444422-444444444444444423-444444444444444424-444444444444444425-444444444444444426-444444444444444427-444444444444444428-444444444444444429-444444444444444430-444444444444444431-444444444444444432-444444444444444433-444444444444444434-444444444444444435-444444444444444436-444444444444444437-444444444444444438-444444444444444439-444444444444444440-444444444444444441-444444444444444442-444444444444444443-444444444444444444-444444444444444445-444444444444444446-444444444444444447-444444444444444448-444444444444444449-4444444444444444410-4444444444444444411-4444444444444444412-4444444444444444413-4444444444444444414-4444444444444444415-4444444444444444416-4444444444444444417-4444444444444444418-4444444444444444419-4444444444444444420-4444444444444444421-4444444444444444422-4444444444444444423-4444444444444444424-4444444444444444425-4444444444444444426-4444444444444444427-4444444444444444428-4444444444444444429-4444444444444444430-4444444444444444431-4444444444444444432-4444444444444444433-4444444444444444434-4444444444444444435-4444444444444444436-4444444444444444437-4444444444444444438-4444444444444444439-4444444444444444440-4444444444444444441-4444444444444444442-4444444444444444443-4444444444444444444-4444444444444444445-4444444444444444446-4444444444444444447-4444444444444444448-4444444444444444449-44444444444444444410-44444444444444444411-44444444444444444412-44444444444444444413-44444444444444444414-44444444444444444415-44444444444444444416-44444444444444444417-44444444444444444418-44444444444444444419-44444444444444444420-44444444444444444421-44444444444444444422-44444444444444444423-44444444444444444424-44444444444444444425-44444444444444444426-44444444444444444427-44444444444444444428-44444444444444444429-44444444444444444430-44444444444444444431-44444444444444444432-44444444444444444433-44444444444444444434-44444444444444444435-44444444444444444436-44444444444444444437-44444444444444444438-44444444444444444439-44444444444444444440-44444444444444444441-44444444444444444442-44444444444444444443-44444444444444444444-44444444444444444445-44444444444444444446-44444444444444444447-44444444444444444448-44444444444444444449-444444444444444444410-444444444444444444411-444444444444444444412-444444444444444444413-444444444444444444414-444444444444444444415-444444444444444444416-444444444444444444417-444444444444444444418-444444444444444444419-444444444444444444420-444444444444444444421-444444444444444444422-444444444444444444423-444444444444444444424-444444444444444444425-444444444444444444426-444444444444444444427-444444444444444444428-444444444444444444429-444444444444444444430-444444444444444444431-444444444444444444432-444444444444444444433-444444444444444444434-444444444444444444435-444444444444444444436-444444444444444444437-444444444444444444438-444444444444444444439-444444444444444444440-444444444444444444441-444444444444444444442-444444444444444444443-444444444444444444444-444444444444444444445-444444444444444444446-444444444444444444447-444444444444444444448-444444444444444444449-4444444444444444444410-4444444444444444444411-4444444444444444444412-4444444444444444444413-4444444444444444444414-4444444444444444444415-4444444444444444444416-4444444444444444444417-4444444444444444444418-4444444444444444444419-4444444444444444444420-4444444444444444444421-4444444444444444444422-4444444444444444444423-4444444444444444444424-4444444444444444444425-4444444444444444444426-4444444444444444444427-4444444444444444444428-4444444444444444444429-4444444444444444444430-4444444444444444444431-4444444444444444444432-4444444444444444444433-4444444444444444444434-4444444444444444444435-4444444444444444444436-4444444444444444444437-4444444444444444444438-4444444444444444444439-44
```

# ... that is universal

```
emacs@helge-x250 ~
  1 ## Load packages and other information for iteration
  2 response$paging$pages
  3 maxpages <- response$paging$pages
  4 records <- response$paging$total
  5 columns <- ncol(response$loans)
  6
  7
  8
  9
 10 ## Open csv, write header
 11 header <- names(response$loans)
 12 write.table(t(header), file = "Data/kiva.csv", sep = ";",
 13             col.names = FALSE, row.names = FALSE)
 14
 15
 16 ## Or collect in data frame (Don't do this for large jobs)
 17 ## data <- data.frame(matrix(nrow = 0, ncol = columns))
 18 ## names(data) <- header
 19
 20 ## Single helper function to flatten columns
 21 unnest <- function(col) paste(unlist(col), collapse = ", ")
 22
 23
 24
 25 ## Iterate over pages, limit to first three
 26 for (p in seq(1, maxpages, by = 1)[1:3]) {
 27
 28   ## Info
 29   print(paste0(p, "/", maxpages))
 30
 31   ## Append page to url
 32   query <- paste0(url, "&page=", p)
 33
 34   ## Get data, assert completeness
 35   loans <- fromJSON(query, flatten = TRUE)$loans
 36   stopifnot(nrow(loans) == pagelength)
 37   stopifnot(ncol(loans) == columns)
 38
 39   ## Fix nested list columns ... or just use data.table::fwrite()
 40   ## str(loans)
 41   loans$tags <- sapply(loans$tags, unnest)
 42   ## loans$themes <- sapply(loans$themes, unnest) # missing for older records
 43   loans$description$languages <- sapply(loans$description$languages, unnest)
 44   ## str(loans)
 45
 46   ## Collect loans in data frame
 47   ## data <- rbind(data, loans)
 48
 49   ## Append to file
 50   write.table(loans, "Data/kiva.csv", sep = ";", append = TRUE,
 51               col.names = FALSE, row.names = FALSE)
 52 }
 53
 54
 55 ## head(data)
 56 ## dtm(data)
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
 100
 101
 102
 103
 104
 105
 106
 107
 108
 109
 110
 111
 112
 113
 114
 115
 116
 117
 118
 119
 120
 121
 122
 123
 124
 125
 126
 127
 128
 129
 130
 131
 132
 133
 134
 135
 136
 137
 138
 139
 140
 141
 142
 143
 144
 145
 146
 147
 148
 149
 150
 151
 152
 153
 154
 155
 156
 157
 158
 159
 160
 161
 162
 163
 164
 165
 166
 167
 168
 169
 170
 171
 172
 173
 174
 175
 176
 177
 178
 179
 180
 181
 182
 183
 184
 185
 186
 187
 188
 189
 190
 191
 192
 193
 194
 195
 196
 197
 198
 199
 200
 201
 202
 203
 204
 205
 206
 207
 208
 209
 210
 211
 212
 213
 214
 215
 216
 217
 218
 219
 220
 221
 222
 223
 224
 225
 226
 227
 228
 229
 230
 231
 232
 233
 234
 235
 236
 237
 238
 239
 240
 241
 242
 243
 244
 245
 246
 247
 248
 249
 250
 251
 252
 253
 254
 255
 256
 257
 258
 259
 260
 261
 262
 263
 264
 265
 266
 267
 268
 269
 270
 271
 272
 273
 274
 275
 276
 277
 278
 279
 280
 281
 282
 283
 284
 285
 286
 287
 288
 289
 290
 291
 292
 293
 294
 295
 296
 297
 298
 299
 300
 301
 302
 303
 304
 305
 306
 307
 308
 309
 310
 311
 312
 313
 314
 315
 316
 317
 318
 319
 320
 321
 322
 323
 324
 325
 326
 327
 328
 329
 330
 331
 332
 333
 334
 335
 336
 337
 338
 339
 340
 341
 342
 343
 344
 345
 346
 347
 348
 349
 350
 351
 352
 353
 354
 355
 356
 357
 358
 359
 360
 361
 362
 363
 364
 365
 366
 367
 368
 369
 370
 371
 372
 373
 374
 375
 376
 377
 378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 420
 421
 422
 423
 424
 425
 426
 427
 428
 429
 430
 431
 432
 433
 434
 435
 436
 437
 438
 439
 440
 441
 442
 443
 444
 445
 446
 447
 448
 449
 450
 451
 452
 453
 454
 455
 456
 457
 458
 459
 460
 461
 462
 463
 464
 465
 466
 467
 468
 469
 470
 471
 472
 473
 474
 475
 476
 477
 478
 479
 480
 481
 482
 483
 484
 485
 486
 487
 488
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 500
 501
 502
 503
 504
 505
 506
 507
 508
 509
 510
 511
 512
 513
 514
 515
 516
 517
 518
 519
 520
 521
 522
 523
 524
 525
 526
 527
 528
 529
 530
 531
 532
 533
 534
 535
 536
 537
 538
 539
 540
 541
 542
 543
 544
 545
 546
 547
 548
 549
 550
 551
 552
 553
 554
 555
 556
 557
 558
 559
 560
 561
 562
 563
 564
 565
 566
 567
 568
 569
 570
 571
 572
 573
 574
 575
 576
 577
 578
 579
 580
 581
 582
 583
 584
 585
 586
 587
 588
 589
 590
 591
 592
 593
 594
 595
 596
 597
 598
 599
 600
 601
 602
 603
 604
 605
 606
 607
 608
 609
 610
 611
 612
 613
 614
 615
 616
 617
 618
 619
 620
 621
 622
 623
 624
 625
 626
 627
 628
 629
 630
 631
 632
 633
 634
 635
 636
 637
 638
 639
 640
 641
 642
 643
 644
 645
 646
 647
 648
 649
 650
 651
 652
 653
 654
 655
 656
 657
 658
 659
 660
 661
 662
 663
 664
 665
 666
 667
 668
 669
 670
 671
 672
 673
 674
 675
 676
 677
 678
 679
 680
 681
 682
 683
 684
 685
 686
 687
 688
 689
 690
 691
 692
 693
 694
 695
 696
 697
 698
 699
 700
 701
 702
 703
 704
 705
 706
 707
 708
 709
 710
 711
 712
 713
 714
 715
 716
 717
 718
 719
 720
 721
 722
 723
 724
 725
 726
 727
 728
 729
 730
 731
 732
 733
 734
 735
 736
 737
 738
 739
 740
 741
 742
 743
 744
 745
 746
 747
 748
 749
 750
 751
 752
 753
 754
 755
 756
 757
 758
 759
 760
 761
 762
 763
 764
 765
 766
 767
 768
 769
 770
 771
 772
 773
 774
 775
 776
 777
 778
 779
 780
 781
 782
 783
 784
 785
 786
 787
 788
 789
 790
 791
 792
 793
 794
 795
 796
 797
 798
 799
 800
 801
 802
 803
 804
 805
 806
 807
 808
 809
 810
 811
 812
 813
 814
 815
 816
 817
 818
 819
 820
 821
 822
 823
 824
 825
 826
 827
 828
 829
 830
 831
 832
 833
 834
 835
 836
 837
 838
 839
 840
 841
 842
 843
 844
 845
 846
 847
 848
 849
 850
 851
 852
 853
 854
 855
 856
 857
 858
 859
 860
 861
 862
 863
 864
 865
 866
 867
 868
 869
 870
 871
 872
 873
 874
 875
 876
 877
 878
 879
 880
 881
 882
 883
 884
 885
 886
 887
 888
 889
 890
 891
 892
 893
 894
 895
 896
 897
 898
 899
 900
 901
 902
 903
 904
 905
 906
 907
 908
 909
 910
 911
 912
 913
 914
 915
 916
 917
 918
 919
 920
 921
 922
 923
 924
 925
 926
 927
 928
 929
 930
 931
 932
 933
 934
 935
 936
 937
 938
 939
 940
 941
 942
 943
 944
 945
 946
 947
 948
 949
 950
 951
 952
 953
 954
 955
 956
 957
 958
 959
 960
 961
 962
 963
 964
 965
 966
 967
 968
 969
 970
 971
 972
 973
 974
 975
 976
 977
 978
 979
 980
 981
 982
 983
 984
 985
 986
 987
 988
 989
 990
 991
 992
 993
 994
 995
 996
 997
 998
 999
 1000
 1001
 1002
 1003
 1004
 1005
 1006
 1007
 1008
 1009
 1010
 1011
 1012
 1013
 1014
 1015
 1016
 1017
 1018
 1019
 1020
 1021
 1022
 1023
 1024
 1025
 1026
 1027
 1028
 1029
 1030
 1031
 1032
 1033
 1034
 1035
 1036
 1037
 1038
 1039
 1040
 1041
 1042
 1043
 1044
 1045
 1046
 1047
 1048
 1049
 1050
 1051
 1052
 1053
 1054
 1055
 1056
 1057
 1058
 1059
 1060
 1061
 1062
 1063
 1064
 1065
 1066
 1067
 1068
 1069
 1070
 1071
 1072
 1073
 1074
 1075
 1076
 1077
 1078
 1079
 1080
 1081
 1082
 1083
 1084
 1085
 1086
 1087
 1088
 1089
 1090
 1091
 1092
 1093
 1094
 1095
 1096
 1097
 1098
 1099
 1100
 1101
 1102
 1103
 1104
 1105
 1106
 1107
 1108
 1109
 1110
 1111
 1112
 1113
 1114
 1115
 1116
 1117
 1118
 1119
 1120
 1121
 1122
 1123
 1124
 1125
 1126
 1127
 1128
 1129
 1130
 1131
 1132
 1133
 1134
 1135
 1136
 1137
 1138
 1139
 1140
 1141
 1142
 1143
 1144
 1145
 1146
 1147
 1148
 1149
 1150
 1151
 1152
 1153
 1154
 1155
 1156
 1157
 1158
 1159
 1160
 1161
 1162
 1163
 1164
 1165
 1166
 1167
 1168
 1169
 1170
 1171
 1172
 1173
 1174
 1175
 1176
 1177
 1178
 1179
 1180
 1181
 1182
 1183
 1184
 1185
 1186
 1187
 1188
 1189
 1190
 1191
 1192
 1193
 1194
 1195
 1196
 1197
 1198
 1199
 1200
 1201
 1202
 1203
 1204
 1205
 1206
 1207
 1208
 1209
 1210
 1211
 1212
 1213
 1214
 1215
 1216
 1217
 1218
 1219
 1220
 1221
 1222
 1223
 1224
 1225
 1226
 1227
 1228
 1229
 1230
 1231
 1232
 1233
 1234
 1235
 1236
 1237
 1238
 1239
 1240
 1241
 1242
 1243
 1244
 1245
 1246
 1247
 1248
 1249
 1250
 1251
 1252
 1253
 1254
 1255
 1256
 1257
 1258
 1259
 1260
 1261
 1262
 1263
 1264
 1265
 1266
 1267
 1268
 1269
 1270
 1271
 1272
 1273
 1274
 1275
 1276
 1277
 1278
 1279
 1280
 1281
 1282
 1283
 1284
 1285
 1286
 1287
 1288
 1289
 1290
 1291
 1292
 1293
 1294
 1295
 1296
 1297
 1298
 1299
 1300
 1301
 1302
 1303
 1304
 1305
 1306
 1307
 1308
 1309
 1310
 1311
 1312
 1313
 1314
 1315
 1316
 1317
 1318
 1319
 1320
 1321
 1322
 1323
 1324
 1325
 1326
 1327
 1328
 1329
 1330
 1331
 1332
 1333
 1334
 1335
 1336
 1337
 1338
 1339
 1340
 1341
 1342
 1343
 1344
 1345
 1346
 1347
 1348
 1349
 1350
 1351
 1352
 1353
 1354
 1355
 1356
 1357
 1358
 1359
 1360
 1361
 1362
 1363
 1364
 1365
 1366
 1367
 1368
 1369
 1370
 1371
 1372
 1373
 1374
 1375
 1376
 1377
 1378
 1379
 1380
 1381
 1382
 1383
 1384
 1385
 1386
 1387
 1388
 1389
 1390
 1391
 1392
 1393
 1394
 1395
 1396
 1397
 1398
 1399
 1400
 1401
 1402
 1403
 1404
 1405
 1406
 1407
 1408
 1409
 1410
 1411
 1412
 1413
 1414
 1415
 1416
 1417
 1418
 1419
 1420
 1421
 1422
 1423
 1424
 1425
 1426
 1427
 1428
 1429
 1430
 1431
 1432
 1433
 1434
 1435
 1436
 1437
 1438
 1439
 1440
 1441
 1442
 1443
 1444
 1445
 1446
 1447
 1448
 1449
 1450
 1451
 1452
 1453
 1454
 1455
 1456
 1457
 1458
 1459
 1460
 1461
 1462
 1463
 1464
 1465
 1466
 1467
 1468
 1469
 1470
 1471
 1472
 1473
 1474
 1475
 1476
 1477
 1478
 1479
 1480
 1481
 1482
 1483
 1484
 1485
 1486
 1487
 1488
 1489
 1490
 1491
 1492
 1493
 1494
 1495
 1496
 1497
 1498
 1499
 1500
 1501
 1502
 1503
 1504
 1505
 1506
 1507
 1508
 1509
 1510
 1511
 1512
 1513
 1514
 1515
 1516
 1517
 1518
 1519
 1520
 1521
 1522
 1523
 1524
 1525
 1526
 1527
 1528
 1529
 1530
 1531
 1532
 1533
 1534
 1535
 1536
 1537
 1538
 1539
 1540
 1541
 1542
 1543
 1544
 1545
 1546
 1547
 1548
 1549
 1550
 1551
 1552
 1553
 1554
 1555
 1556
 1557
 1558
 1559
 1560
 1561
 1562
 1563
 1564
 1565
 1566
 1567
 1568
 1569
 1570
 1571
 1572
 1573
 1574
 1575
 1576
 1577
 1578
 1579
 1580
 1581
 1582
 1583
 1584
 1585
 1586
 1587
 1588
 1589
 1590
 1591
 1592
 1593
 1594
 1595
 1596
 1597
 1598
 1599
 1600
 1601
 1602
 1603
 1604
 1605
 1606
 1607
 1608
 1609
 1610
 1611
 1612
 1613
 1614
 1615
 1616
 1617
 1618
 1619
 1620
 1621
 1622
 1623
 1624
 1625
 1626
 1627
 1628
 1629
 1630
 1631
 1632
 1633
 1634
 1635
 1636
 1637
 1638
 1639
 1640
 1641
 1642
 1643
 1644
 1645
 1646
 1647
 1648
 1649
 1650
 1651
 1652
 1653
 1654
 1655
 1656
 1657
 1658
 1659
 1660
 1661
 1662
 1663
 1664
 1665
 1666
 1667
 1668
 1669
 1670
 1671
 1672
 1673
 1674
 1675
 1676
 1677
 1678
 1679
 1680
 1681
 1682
 1683
 1684
 1685
 1686
 1687
 1688
 1689
 1690
 1691
 1692
 1693
 1694
 1695
 1696
 1697
 1698
 1699
 1700
 1701
 1702
 1703
 1704
 1705
 1706
 1707
 1708
 1709
 1710
 1711
 1712
 1713
 1714
 1715
 1716
 1717
 1718
 1719
 1720
 1721
 1722
 1723
 1724
 1725
 1726
 1727
 1728
 1729
 1730
 1731
 1732
 1733
 1734
 1735
 1736
 1737
 1738
 1739
 1740
 1741
 1742
 1743
 1744
 1745
 1746
 1747
 1748
 1749
 1750
 1751
 1752
 1753
 1754
 1755
 1756
 1757
 1758
 1759
 1760
 1761
 1762
 1763
 1764
 1765
 1766
 1767
 1768
 1769
 1770
 1771
 1772
 1773
 1774
 1775
 1776
 1777
 1778
 1779
 1780
 1781
 1782
 1783
 1784
 1785
 1786
 1787
 1788
 1789
 1790
 1791
 1792
 1793
 1794
 1795
 1796
 1797
 1798
 1799
 1800
 1801
 1802
 1803
 1804
 1805
 1806
 1807
 1808
 1809
 1810
 1811
 1812
 1813
 1814
 1815
 1816
 1817
 1818
 1819
 1820
 1821
 1822
 1823
 1824
 1825
 1826
 1827
 1828
 1829
 1830
 1831
 1832
 1833
 1834
 1835
 1836
 1837
 1838
 1839
 1840
 1841
 1842
 1843
 1844
 1845
 1846
 1847
 1848
 1849
 1850
 1851
 1852
 1853
 1854
 1855
 1856
 1857
 1858
 1859
 1860
 1861
 1862
 1863
 1864
 1865
 1866
 1867
 1868
 1869
 1870
 1871
 1872
 1873
 1874
 1875
 1876
 1877
 1878
 1879
 1880
 1881
 1882
 1883
 1884
 1885
 1886
 1887
 1888
 1889
 1890
 1891
 1892
 1893
 1894
 1895
 1896
 1897
 1898
 1899
 1900
 1901
 1902
 1903
 1904
 1905
 1906
 1907
 1908
 1909
 1910
 1911
 1912
 1913
 1914
 1915
 1916
 1917
 1918
 1919
 1920
 1921
 1922
 1923
 1924
 1925
 1926
 1927
 1928
 1929
 1930
 1931
 1932
 1933
 1934
 1935
 1936
 1937
 1938
 1939
 1940
 1941
 1942
 1943
 1944
 1945
 1946
 1947
 1948
 1949
 1950
 1951
 1952
 1953
 1954
 1955
 1956
 1957
 1958
 1959
 1960
 1961
 1962
 1963
 1964
 1965
 1966
 1967
 1968
 1969
 1970
 1971
 1972
 1973
 1974
 1975
 1976
 1977
 1978
 1979
 1980
 1981
 1982
 1983
 1984
 1985
 1986
 1987
 1988
 1989
 1990
 1991
 1992
 1993
 1994
 1995
 1996
 1997
 1998
 1999
 2000
 2001
 2002
 2003
 2004
 2005
 2006
 2007
 2008
 2009
 2010
 2011
 2012
 2013
 2014
 2015
 2016
 2017
 2018
 2019
 2020
 2021
 2022
 2023
 2024
 2025
 2026
 2027
 2028
 2029
 2030
 2031
 2032
 2033
 2034
 2035
 2036
 2037
 2038
 2039
 2040
 2041
 2042
 2043
 2044
 2045
 2046
 2047
 2048
 2049
 2050
 2051
 2052
 2053
 2054
 2055
 2056
 2057
 2058
 2059
 2060
 2061
 2062
 2063
 2064
 2065
 2066
 2067
 2068
 2069
 2070
 2071
 2072
 2073
 2074
 2075
 2076
 2077
 2078
 2079
 2080
 2081
 2082
 2083
 2084
 2085
 2086
 2087
 2088
 2089
 2090
 2091
 2092
 2093
 2094
 2095
 2096
 2097
 2098
 2099
 2100
 2101
 2102
 2103
 2104
 2105
 2106
 2107
 2108
 2109
 2110
 2111
 2112
 2113
 2114
 2115
 2116
 2117
 2118
 2119
 2120
 2121
 2122
 2123
 2124
 2125
 2126
 2127
 2128
 2129
 2130
 2131
 2132
 2133
 2134
 2135
 2136
 2137
 2138
 2139
 2140
 2141
 2142
 2143
 2144
 2145
 2146
 2147
 2148
 2149
 2150
 2151
 2152
 2153
 2154
 2155
 2156
 2157
 2158
 2159
 2160
 2161
 2162
 2163
 2164
 2165
 2166
 2167
 2168
 
```

# ... that is universal

each school-track, classroom formation is conditionally ignorable with respect to SN status, as students from different primary school districts are mixed and their SN status is not observed by secondary school administrators.<sup>\footnote{Using PISA data from secondary schools in Switzerland, \citet{Vardardottir2015} shows that track-by-school fixed effects render peer group composition conditionally uncorrelated with a large set of students' characteristics, while track fixed effects and school fixed effects do not. Neither primary schools nor the SPS share information with secondary schools for equity reasons and to avoid stigma when transitioning between schools.}</sup>

174 175 % NEU: BEu bitte lesen

176 % changed some small things, ok for me.

177 Beyond this anecdotal evidence, we formally test the validity of the identification strategy with four balancing tests, which are presented and discussed extensively in Appendix B. First, we examine whether the proportion of SN peers predicts individual baseline characteristics (gender, native speaker, and age). The aim of this test is to detect potential selection into classrooms. We also conduct this test separately for SN and non-SN students. None of the baseline characteristics are statistically significant at conventional levels, either considered individually or jointly. Second, we regress the indicator for SN status on class fixed effects, which should be jointly insignificant if assignment to `Classrooms` is ignorable with respect to SN status <sup>\citet{Chettyetal2011}</sup>. We also conduct this test to check for ignorable assignment of SN students to teachers. We find no evidence for systematic assignment of SN students to either classes or teachers. Third, we conduct a simulation exercise in the spirit of <sup>\citet{Carrel2010}</sup>. We re-sample classes and thereby assign SN students `randomly` to `Classrooms`, and test whether the observed distribution of SN students differs from the simulated one. In addition, we compute the interquartile range of the proportion SN students across classes for each simulation, and compare these simulated interquartile ranges with the one we observe in the data. Neither simulation procedure uncovers any worrisome pattern in the assignment of SN students to classes. Fourth, we decompose the variation in the fraction of SN peers across and within schools. To do so, we examine the residual variation in the proportion of SN peers after partialling out the school-track-year fixed effects. We find that the residual distribution in the proportion of SN peers is consistent with variation from a random process. Overall, the balancing tests we performed indicate that the key identification assumption of (conditionally) ignorable assignment of SN students to classes is plausible.

178  
179 Our identification relies on variation between classes within school-track-years. Although families can potentially choose their district of residence and thereby influence schooling options for their children, possible selection into schools does not confound our results.<sup>\footnote{Endogenous class formation could still occur if parents request to transfer their children to a class with a lower SN fraction. To investigate this potential threat, we acquired the official education statistics from the Swiss Federal Statistical Office (SDO, \textit{Statistik der Lernenden} in German) for the years 2012–2015. Importantly, the SDO has a classroom ID which allows us to reconstruct the classes within each school-year-track. For the state of St. Gallen, we find that no</sup>

different education tracks, and classes are strictly separated between tracks. Within each school-track, classroom formation is conditionally ignorable with respect to SN status, as students from different primary school districts are mixed and their SN status is not observed by secondary school administrators.<sup>\footnote{Using PISA data from secondary schools in Switzerland, \citet{Vardardottir2015} shows that track-by-school fixed effects render peer group composition conditionally uncorrelated with a large set of students' characteristics, while track fixed effects and school fixed effects do not. Neither primary schools nor the SPS share information with secondary schools for equity reasons and to avoid stigma when transitioning between schools.}</sup>

172

173 % NEU: BEu bitte lesen

174 Beyond this anecdotal evidence, we formally test the validity of the identification strategy with four balancing tests, which are presented and discussed extensively in Appendix B. First, we examine whether the proportion of SN peers predicts individual baseline characteristics (gender, native speaker, and age). The aim of this test is to detect potential selection into classrooms. We also conduct this test separately for SN and non-SN students. None of the baseline characteristics are statistically significant at conventional levels, either considered individually or jointly. Second, we regress the indicator for SN status on class fixed effects, which should be jointly insignificant if assignment to `Classroom` is ignorable with respect to SN status <sup>\citet{Chettyetal2011}</sup>. We also conduct this test to check for ignorable assignment of SN students to specific teachers. We find no evidence for systematic assignment of SN students to either classes or teachers. Third, we conduct a simulation exercise in the spirit of <sup>\citet{Carrel2010}</sup>. We re-sample classes, randomly assign SN students to `classes`, and test whether the observed distribution of SN students differs from the simulated one. In addition, we simulate random classroom assignment within schools, compute the interquartile range of the proportion SN across classes, and compare the simulated interquartile range with the one we observe in the data. Neither simulation procedure uncovers any worrisome pattern in the assignment of SN students to classes. Fourth, we decompose the variation in the fraction of SN peers across and within schools. To do so, we examine the residual variation in the proportion of SN peer after partialling out the school-track-year fixed effects. We find that the residual distribution in the proportion of SN peers is consistent with variation from a random process. Overall, the balancing tests we performed indicate that the key identification assumption of (conditionally) ignorable assignment of SN students to classes is plausible.

175

176 Our identification relies on variation between classes within school-track-years. Although families can potentially choose their district of residence and thereby influence schooling options for their children, possible selection into schools does not confound our results.<sup>\footnote{Endogenous class formation could still occur if parents request to transfer their children to a class with a lower SN fraction. To investigate this potential threat, we acquired the official education statistics from the Swiss Federal Statistical Office (SDO, \textit{Statistik der Lernenden} in German) for the years 2012–2015. Importantly, the SDO has a classroom ID which allows us to reconstruct the classes within each school-year-track. For the state of St. Gallen, we find that no</sup>

A: ● 104k xSource/manuscript\_R1\_v3.tex ??:0 37%

LF UTF-8 LaTeX/FPS

B: ● 104k xOld/manuscript\_R1\_v2.tex ??:0 37%

LF UTF-8 LaTeX/FPS

Type ? for help

--Ediff Control Panel\* diff 4 of 15

Quick Help

Auto-refining is ON

# Rstudio

RStudio  
Project: (None) ▾

Halyan-13.R x

Source on Save | Go to file/function

```
library(XML)
url <- "http://www.gdacs.org/Cyclones/report.aspx?eventid=41058&episodeid=28&
dat <- readHTMLTable(readLines(url), which=5)
dat$latlon <- dat[,8]
levels(dat$latlon) <- sapply(
  strsplit(levels(dat[,8]), ",\n"), 
  function(x) paste(x[2], x[1], sep=""))
dat$Category <- factor(dat$Category, levels=levels(dat$Category)[c(6,7,1:5]),
  ordered=TRUE)
dat$cat <- as.numeric(dat$Category)
dat$Gust_kmh <- dat[,6]
levels(dat$Gust_kmh) <- sapply(strsplit(levels(dat[,6]), "km"),
  function(x) gsub(" ", "", x[1]))
dat$Gust_kmh <- as.numeric(as.character(dat$Gust_kmh))
M <- gvisGeoChart(dat, "latlon", sizevar="cat",
  colorvar="Gust_kmh",
  options=list(region='035',
    backgroundColor="lightblue",
    datalessRegionColor="grey"))
plot(M)
```

10:37 (Top Level) ▾

R Script ▾

Environment History Presentation ▾

Import Dataset Clear Global Environment ▾

Data dot 33 obs. of 11 variables

Values M List of 3

url "http://www.gdacs.org/Cyclones/report.aspx?eventid=41058&e...

Console ~ /Downloads/IdR-dev/IdR/ ↵

> out\$latlon <- dat[,8]
> levels(out\$latlon) <- sapply(
+ strsplit(levels(dat[,8]), ",\n"), 
+ function(x) paste(x[2], x[1], sep=""))
+
> dat\$Category <- factor(dat\$Category, levels=levels(dat\$Category)[c(6,7,1:5]),
+ ordered=TRUE)
> dat\$cat <- as.numeric(dat\$Category)
> dat\$Gust\_kmh <- dat[,6]
> levels(dat\$Gust\_kmh) <- sapply(strsplit(levels(dat[,6]), "km"),
+ function(x) gsub(" ", "", x[1]))
> dat\$Gust\_kmh <- as.numeric(as.character(dat\$Gust\_kmh))
> M <- gvisGeoChart(dat, "latlon", sizevar="cat",
+ colorvar="Gust\_kmh",
+ options=list(region='035',
+ backgroundColor="lightblue",
+ datalessRegionColor="grey"))
> plot(M)
>

Files Plots Packages Help Viewer

56 380

Data: dat • Chart ID: GeoChartD926a2a743168  
R version 3.0.2 (2013-09-25) • googleVis-0.4.7 • Google Terms of Use • Data Policy

# Spyder

Spyder - Spyder (Python 3.6)

File Edit Search Source Run Debug consoles Projects Tools View Help

Project explorer Editor C:\Users\testUser\Downloads\16740\156420f67bbf8ab7c3ee3aae+55140feca17b09f7a9fb5b7f4dbd1b953679\16740\156420f67bbf8ab7c3ee3aae+55140feca17b09f7a9fb5b7f4dbd1b953679 README.md

temp.py interpolation.py \_run\_py und\_helper.py und\_main.py

6

```
7 import pylab
8 from numpy import cos, linspace, pi, sin, random
9 from scipy.interpolate import splprep, splev
10
11 # %% Generate data for analysis
12
13 # Make ascending spiral in 3-space
14 t = linspace(0, 1.75 * 2 * pi, 100)
15
16 x = sin(t)
17 y = -cos(t)
18 z = t
19
20 # Add noise
21 x += random.normal(scale=0.1, size=x.shape)
22 y += random.normal(scale=0.1, size=y.shape)
23 z += random.normal(scale=0.1, size=z.shape)
24
25
26 # %% Perform calculations
27
28 # Spline parameters
29 smoothness = 3.0 # Smoothness parameter
30 k_param = 2 # Spline order
31 nests = -1 # Estimate of number of knots needed (-1 = maximal)
32
33 # Find the knot points
34 knot_points, u = splprep([x, y, z], s=smoothness, k=k_param, nests=-1)
35
36 # Evaluate spline, including interpolated points
37 xnew, ynew, znew = splev(linspace(0, 1, 400), knot_points)
38
39
40 # %% Plot results
41
42 # TODO: Rewrite to avoid code smell.
43 pylab.subplot(2, 2, 1)
44 data, = pylab.plot(x, y, 'bo-', label='Data with X-Y Cross Section')
45 fit, = pylab.plot(xnew, ynew, 'r-', label='Fit with X-Y Cross Section')
46 pylab.legend()
47 pylab.xlabel('x')
48 pylab.ylabel('y')
49
50 pylab.subplot(2, 2, 2)
51 data, = pylab.plot(x, z, 'bo-', label='Data with X-Z Cross Section')
52 fit, = pylab.plot(xnew, znew, 'r-', label='Fit with X-Z Cross Section')
53 pylab.legend()
54 pylab.xlabel('x')
```

7 import pylab
8 from numpy import cos, linspace, pi, sin, random
9 from scipy.interpolate import splprep, splev
10
11 # %% Generate data for analysis
12
13 # Make ascending spiral in 3-space
14 t = linspace(0, 1.75 \* 2 \* pi, 100)
15
16 x = sin(t)
17 y = -cos(t)
18 z = t
19
20 # Add noise
21 x += random.normal(scale=0.1, size=x.shape)
22 y += random.normal(scale=0.1, size=y.shape)
23 z += random.normal(scale=0.1, size=z.shape)
24
25
26 # %% Perform calculations
27
28 # Spline parameters
29 smoothness = 3.0 # Smoothness parameter
30 k\_param = 2 # Spline order
31 nests = -1 # Estimate of number of knots needed (-1 = maximal)
32
33 # Find the knot points
34 knot\_points, u = splprep([x, y, z], s=smoothness, k=k\_param, nests=-1)
35
36 # Evaluate spline, including interpolated points
37 xnew, ynew, znew = splev(linspace(0, 1, 400), knot\_points)
38
39
40 # %% Plot results
41
42 # TODO: Rewrite to avoid code smell.
43 pylab.subplot(2, 2, 1)
44 data, = pylab.plot(x, y, 'bo-', label='Data with X-Y Cross Section')
45 fit, = pylab.plot(xnew, ynew, 'r-', label='Fit with X-Y Cross Section')
46 pylab.legend()
47 pylab.xlabel('x')
48 pylab.ylabel('y')
49
50 pylab.subplot(2, 2, 2)
51 data, = pylab.plot(x, z, 'bo-', label='Data with X-Z Cross Section')
52 fit, = pylab.plot(xnew, znew, 'r-', label='Fit with X-Z Cross Section')
53 pylab.legend()
54 pylab.xlabel('x')

variable explorer

Name	Type	Size	Value
array_int8	Int8	(2, 3)	Min: -7 Max: 6
array_uint32	UInt32	(2, 2, 3)	Min: 1 Max: 7
bars	Container-BarContainer	20	BarContainer object of matplotlib.container
df	DataFrame	(3, 2)	Column names: bools, ints
filename	str	1	C:\ProgramData\Anaconda\lib\site-packs-
list_test	list	2	[DataFrame, Numpy array]
nrows	int	1	544
r	float64	1	7.61180259334796
radii	float64	(20,)	Min: 0.4863016038516867 Max: 9.85684897842551
region	tuple	2	(slice, slice)
rgb	Float64	(45, 45, 4)	Min: 1.0 Max: 1.0
series	Series	(1,)	Series object of pandas.core.series.mod-
text_name	NoneType	1	NoneType object of builtins module

Python console

```
...: ls = LightSource(270, 45)
...: # To use a custom hillshading mode, override the built-in shading
...: # in the rgb colors of the shaded surface calculated from "shade".
...: # rgb = ls.shade(z, cmap=cms.gist_earth, vert_exag=0.1, blend_mode='soft')
...: ...: surf = ax.plot_surface(x, y, z, rstride=1, cstride=1, facecolors=rgb,
...: ...:                      linewidth=0, antialiased=False, shade=False)
...: ...
...: plt.show()
```

Figure 1: A 3D surface plot showing a spiral in 3D space. The x-axis ranges from -84.41 to 36.73, the y-axis from -84.40 to 36.71, and the z-axis from 400 to 900. The plot shows a blue-shaded surface representing a fitted spline to the data points.

Figure 2: A polar plot showing data points and a fitted curve. The radial axis represents values from 1 to 10, and the angular axis represents angles from 0 to 360 degrees. The plot shows a red-shaded area representing the fitted curve.

In [12]:

Python console History log Internal console

Permissions: RW End-of-lines LF Encoding: UTF-8 Line: 26 Column: 4 Memory: 49 % CPU: 15 %

## Version control

---

Familiar?

## "FINAL".doc



FINAL.doc!



FINAL\_rev.2.doc



FINAL\_rev.2.doc



FINAL\_rev.6.COMMENTS.doc



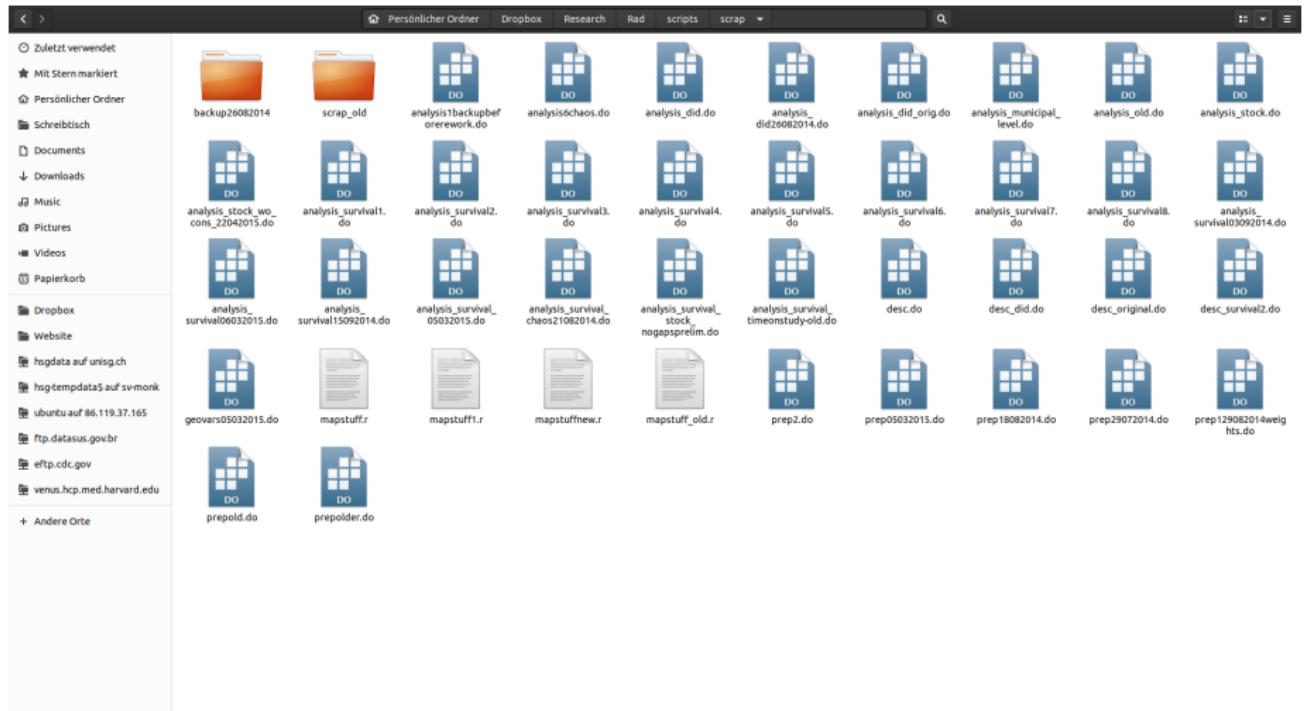
FINAL\_rev.8.comments5.  
CORRECTIONS.doc



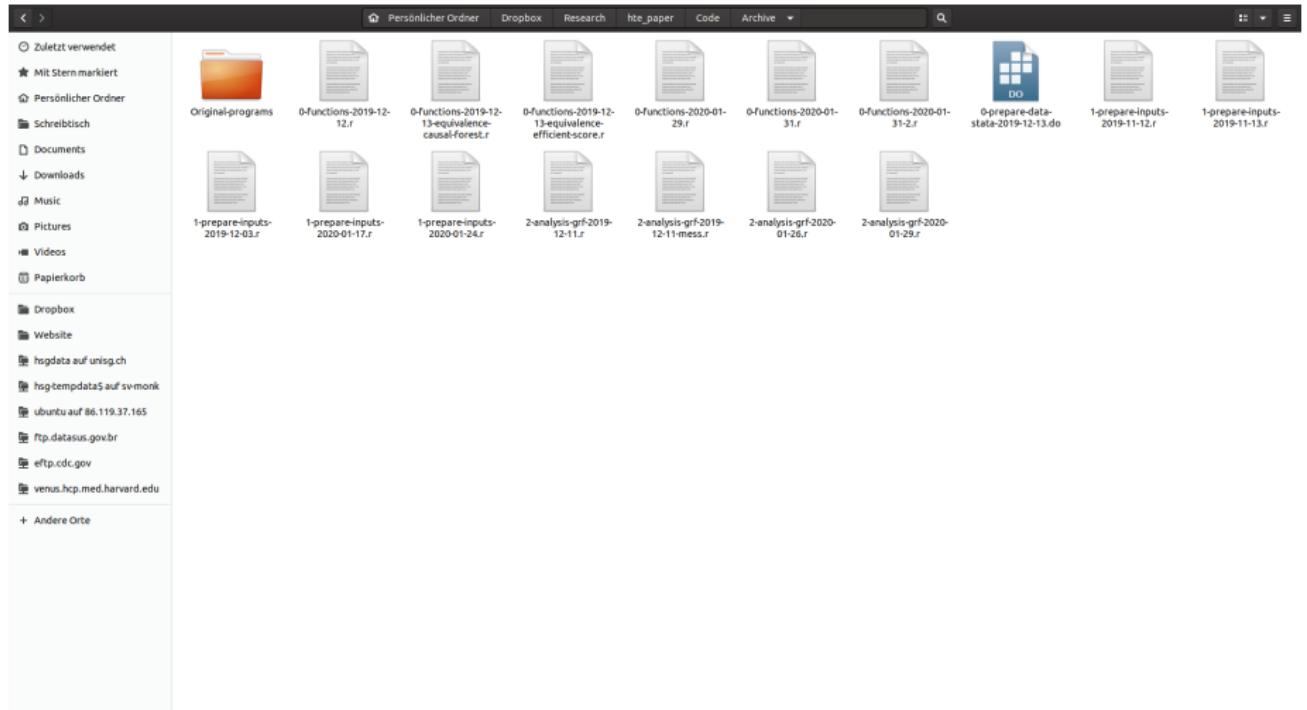
FINAL\_rev.18.comments7.  
corrections9.MORE.30.doc      FINAL\_rev.22.comments49.  
corrections.10.#@\$%WHYDID  
ICOMETOGRAD SCHOOL????.doc



# Familiar?



# Familiar?

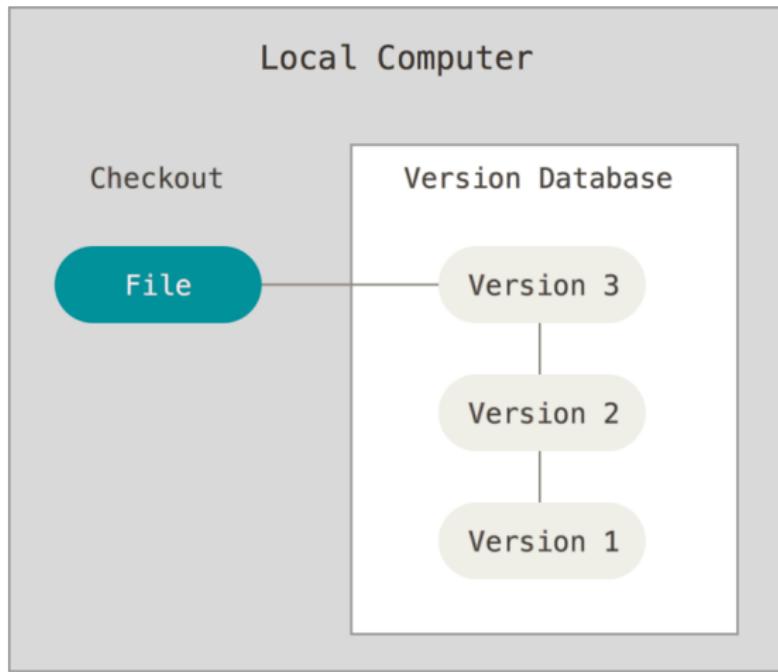


# Familiar?

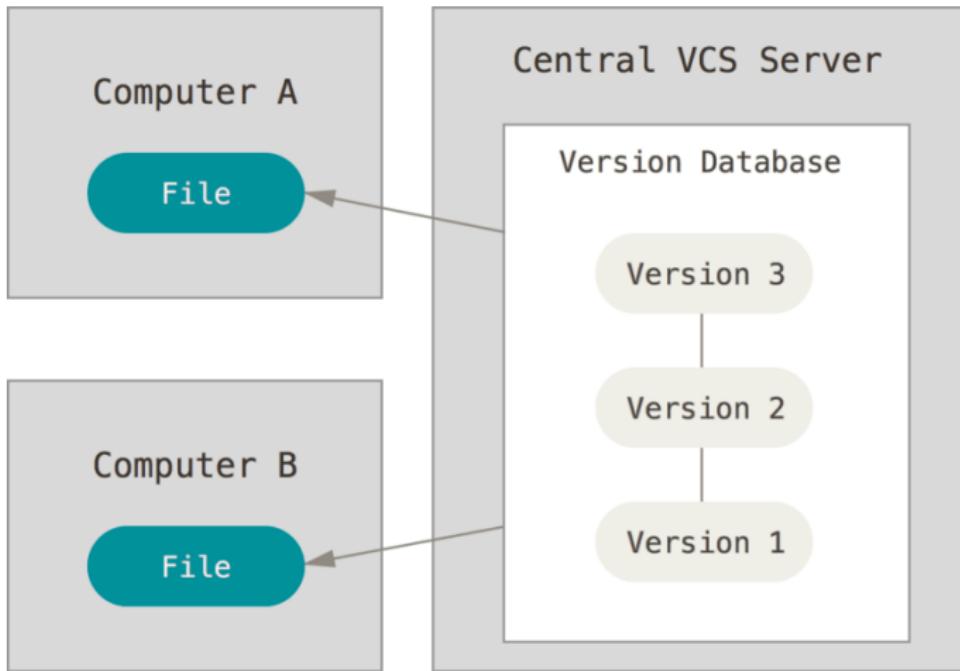
# Tracking different versions

- Different options:
  - Don't keep track.
  - Keep file copies. Adhere to a versioning format. Write a changelog. Diff files.
  - Formal version control.
- Formal version control:
  - Ubiquitous in software development, extremely useful for many purposes.
  - Less established for statistical data management and analysis.
  - Workflow in data analysis differs. More effort to use properly.
  - Viewpoints on usefulness for data analysis vary.

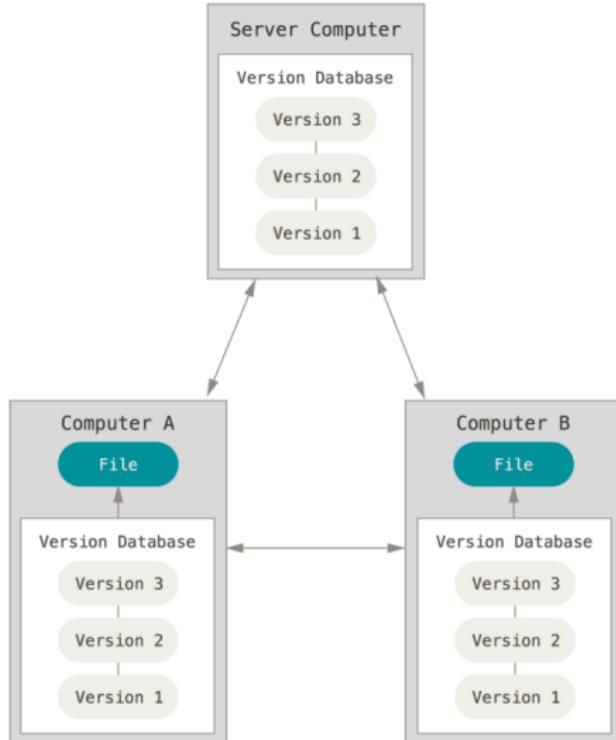
# Local version control



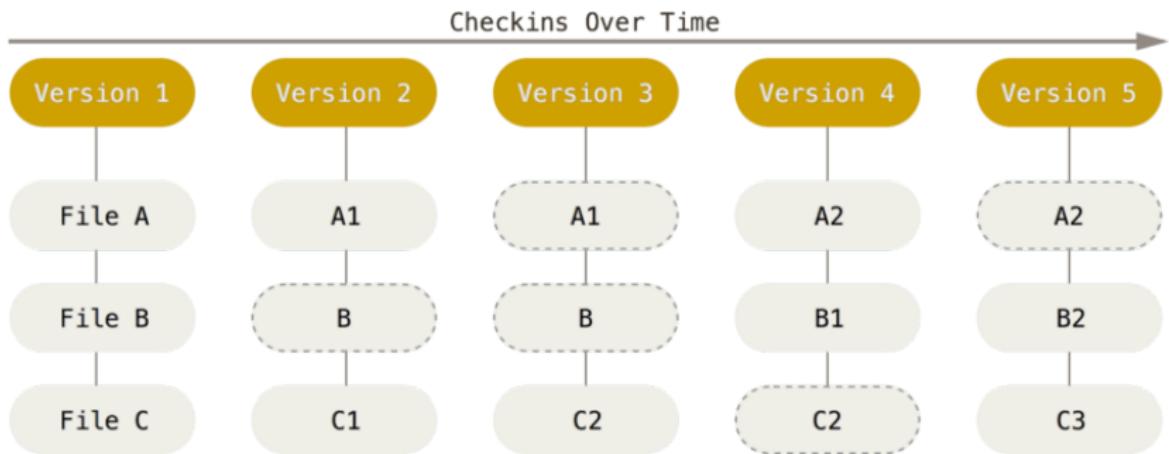
# Centralized version control



# Distributed version control



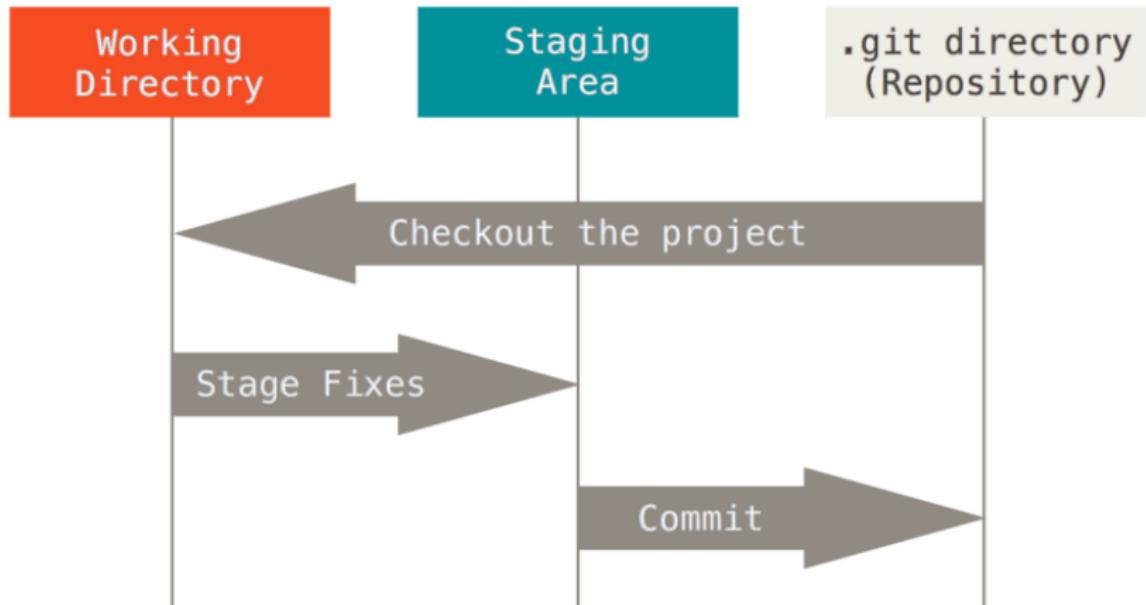
# Distributed version control - snapshots



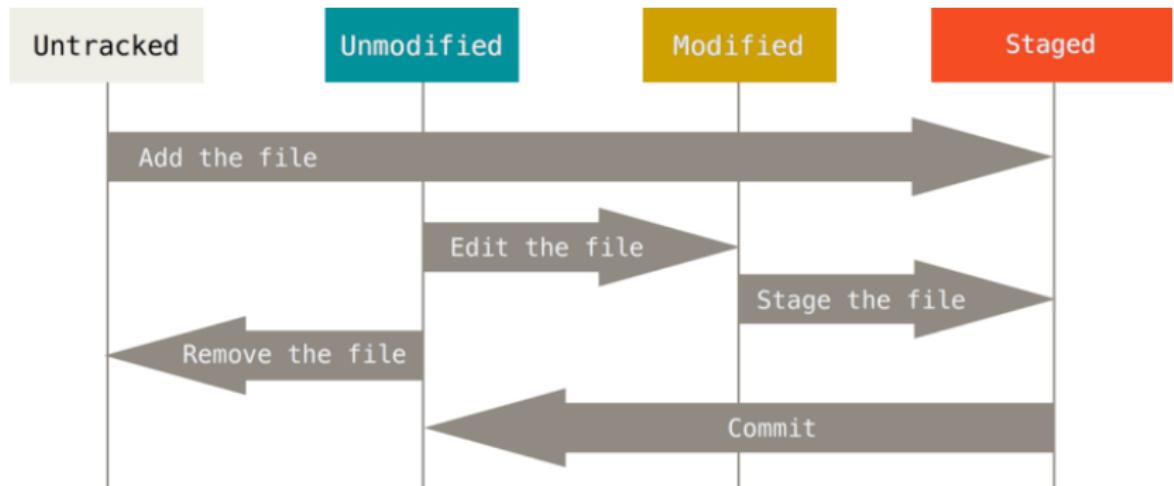
# Git

- Git is the dominant *distributed* version control system today.
- Developed by Linus Torvalds, the creator of the Linux kernel.
- Can be used on the command line. GUIs/editor plugins available.  
Integrated into RStudio.
- Fast, good support for non-linear development (thousands of parallel branches).
- Tracks any content (but mostly plain text files).
  - Source code, manuscripts, websites, presentations, ...
  - As a rule, do not version control data.

## Workflow: Areas



# Workflow: File lifecycle



# Git

- Git stores the complete history of files and all changes you or others made to them in a local repository on your computer.
- You can have different ‘branches’ with different features, switch between them or merge them.
- You can keep a ‘remote’ copy of the repository on a server, but this is not required.
- You can ‘clone’ repositories from others and ‘push’ back changes you made to them.
- Public repositories facilitate collaboration and development.

## Advantages and disadvantages

- Easy to track changes over time (from multiple people) and simple to revert them.
- Helps structure thoughts and makes you write cleaner code.
- Reproducible. Eases outside contributions and helps hunting down bugs.
- Requires work (to learn and to use), especially in the short term.
- More so if you have to convince your colleagues.
- Integrates less well with iterative back-and-forth workflow.

## Advantages and disadvantages

- Easy to track changes over time (from multiple people) and simple to revert them.
  - Helps structure thoughts and makes you write cleaner code.
  - Reproducible. Eases outside contributions and helps hunting down bugs.
  - Requires work (to learn and to use), especially in the short term.
  - More so if you have to convince your colleagues.
  - Integrates less well with iterative back-and-forth workflow.
- ➔ Useful for many projects.
- ➔ Good to understand the basics, even if you do not use it. Lots of programs are hosted on public git repositories, ubiquitous in industry.

# Resources

- ProGit is a good and free resource. Skim the first few chapters.
- Software carpentry has a good introductory course. Many more resources online.
- Github is a site for online storage of Git repositories (Git ≠ Github!). You can create a remote repository there and push code to it. Public repositories are free.
- Github also offers student and educator accounts including free private repositories.
- Some editors offer *persistent undo* with branching, which can be very helpful (as long as you are aware of its limits—it is not a replacement for version control).

## Command line interface

---

# Command line interfaces

- A *command-line interface* (CLI) processes commands to a computer program in the form of lines of text.
- Different from *graphical user interfaces* (GUIs).
- The program which handles the interface is called a *command-line interpreter* or *shell*.
- A shell provides interactive or programmatic access to operating system functions and other services.
- CLI shells became the primary access to computer terminals starting in the mid-1960s. Also used by mainstream personal computers (Unix, MS-DOS, ...).

# Terminals



```
SLMN .S15 12P 25-Nov-95 1K .S15 3P 13-Aug-95  
XL .S15 4P 13-Aug-95 BL .S15 3P 13-Aug-95  
SP .S15 7P 13-Aug-95 SP .S15 3P 13-Aug-95  
BT100J.S15 7W 13-Aug-95 TT .S15 3P 13-Aug-95  
M .S15 2P 13-Aug-95 BLANG .S15 7ZP 25-Nov-95  
DE .S15 9W 24-May-79 BACKUP .S15 13 25-Nov-95  
DATIME .S15 4 28-Nov-95 SIS .S15 13 25-Nov-95  
DUMP .S15 7W 27-Nov-95 FORTNITE .S15 20 25-Nov-95  
TSDR001 .S15 4 23-Jun-95 LET .S15 9 25-Nov-95  
START .P20 2 23-Jun-95 TSDR01 .S15 15ZP 16-May-95  
EMLA .S15 13P 15-Aug-95 TSDR02 .S15 15 16-May-95  
STARD .L20 12P 15-Aug-95 TSDR03 .S15 15 16-May-95  
EMER .S15 13 15-Aug-95 TSDR04 .S15 20 08-May-95  
JASLP .P10 20 18-Sep-95 TSDR05 .S15 25 08-May-95  
JASLP .P10 20 18-Sep-95 TSDR06 .S15 25 08-May-95  
JASLP .P10 20 18-Sep-95 TSDR07 .S15 7 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR08 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR09 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR10 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR11 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR12 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR13 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR14 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR15 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR16 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR17 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR18 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR19 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR20 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR21 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR22 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR23 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR24 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR25 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR26 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR27 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR28 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR29 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR30 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR31 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR32 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR33 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR34 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR35 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR36 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR37 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR38 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR39 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR40 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR41 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR42 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR43 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR44 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR45 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR46 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR47 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR48 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR49 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR50 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR51 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR52 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR53 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR54 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR55 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR56 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR57 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR58 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR59 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR60 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR61 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR62 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR63 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR64 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR65 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR66 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR67 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR68 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR69 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR70 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR71 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR72 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR73 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR74 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR75 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR76 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR77 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR78 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR79 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR80 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR81 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR82 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR83 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR84 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR85 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR86 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR87 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR88 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR89 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR90 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR91 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR92 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR93 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR94 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR95 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR96 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR97 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR98 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR99 .S15 1 16-Jul-95  
JASLP .P10 20 18-Sep-95 TSDR100 .S15 1 16-Jul-95  
14432 Files, 9549 Blocks  
14432 Free Blocks
```

# Terminals

The screenshot shows a terminal window with a dark blue background and white text. The title bar reads "helge@helge-x250: ~". The window contains the following command-line session:

```
helge@helge-x250:~$ ssh -X liebert@venus.hcp.med.harvard.edu
liebert@venus.hcp.med.harvard.edu's password:
Last login: Fri Feb 14 12:33:14 2020 from 10.0.34.90
venus:/home/liebert$ ll
insgesamt 13
drwxr-xr-x+ 3 liebert users    3  8. Apr 2019  ado
drwx-----+ 2 liebert users    5  5. Mär 2019  Desktop
drwx-----+ 4 liebert users    4  4. Feb 15:10 hte_paper
drwx-----+ 3 liebert users   10 11. Dez 06:26 iqr-simulation
drwx-----+ 2 liebert users    7 11. Feb 10:38 Mortality
drwx-----+ 4 liebert users    5  2. Dez 15:56 Physicians
drwx-----+ 3 liebert users    3  2. Jul 2019  R
-rw-----+ 1 liebert users 1714 15. Jan 2019  Trash Bln.lnk
venus:/home/liebert$ cd Physicians/
venus:/home/liebert/Physicians$ xstata
venus:/home/liebert/Physicians$ logout
Connection to venus.hcp.med.harvard.edu closed.
helge@helge-x250:~$
```

# Operating system shells

- Still in use today. Unix shells most relevant.
- Unix shells are used in Unix-derived systems (Linux, MacOS).
- Common operating system shells: `sh` (Unix), `cmd` (Windows); `bash`, `zsh` (Linux, MacOS), ...
- Unix-based operating systems (Linux, MacOS) have Bash/zsh pre-installed. Just start `Terminal`.
- On Linux, all system maintenance and software installation can be done via the shell.

## Why does this matter?

- In 2019, 100% of the world's TOP500 supercomputers run on Linux.
- 96.3% of the world's top 1 million servers run on Linux.
- 90% of all cloud infrastructure operates on Linux.

## Why does this matter?

- In 2019, 100% of the world's TOP500 supercomputers run on Linux.
  - 96.3% of the world's top 1 million servers run on Linux.
  - 90% of all cloud infrastructure operates on Linux.
  - **sciCORE** runs on Linux and uses CLI access.
  - **Switch engines** provides Linux instances with CLI access.
  - **Amazon web services** (AWS) provide Linux-based computing resources.
  - So do **Swisscom dynamic computing services**.
- ➡ Much of the computing infrastructure you are likely to use will run Linux.

helge@helge-x250:~

```
helge@helge-x250:~$ ssh -X cyfiwa67@login.scicore.unibas.ch  
cyfiwa67@login.scicore.unibas.ch's password:  
Last login: Fri Feb 14 18:49:10 2020 from vpn_pat.med.harvard.edu
```



```
#####  
OS: CentOS 7.5.1804  
Queing System: Slurm  
support contact: scicore-admin@unibas.ch  
#####
```

Check the cluster documentation in these links:  
<https://wiki.biozentrum.unibas.ch/display/scicore/scicore+user+guide>  
<https://wiki.biozentrum.unibas.ch/display/scicore/Slurm+user+guide+at+scicore>

Try "module av" or "module spider" to see the list  
of available software in the cluster.

For details about software management:  
<https://wiki.biozentrum.unibas.ch/display/scicore/scicore+user+guide#scicOREuserguide-Software>  
<https://lmod.readthedocs.io/en/latest/>

```
[cyfiwa67@login10 ~]$ ll  
insgesamt 0  
[cyfiwa67@login10 ~]$ exit  
Abgemeldet  
Connection to login.scicore.unibas.ch closed.  
helge@helge-x250:~$
```

# A note on operating systems

- MacOS or Linux offer built-in access to a Unix shell (Bash or zsh).
- Further software is managed via a package management system and distributed via software repositories.
- On Linux, use your package manager to install anything you require.
- On MacOS, familiarize yourself with Homebrew. Install iterm2 if you want a fancier terminal.
- Windows does not provide proper access to a Unix shell out-of-the-box.
- Many common tools are not available or cumbersome to use.
- Windows has cmd (or PowerShell). These are not Unix shells and not a viable replacement for most use cases.
- Dependency resolution can be a nightmare. Package managers for Windows like scoop (or chocolatey) help.

# What is a package manager?

*“A package manager or package-management system is a collection of software tools that automates the process of installing, upgrading, configuring, and removing computer programs for a computer’s operating system in a consistent manner.”*

- Software is installed and updated from official software repositories.
- System-level package managers: `apt`, `dnf`, `pacman`, `yum`, `homebrew`, `chocolatey`, `scoop`, ...
- Application-level package managers: `pip`, `conda`, ...
- Typically used via the command line.

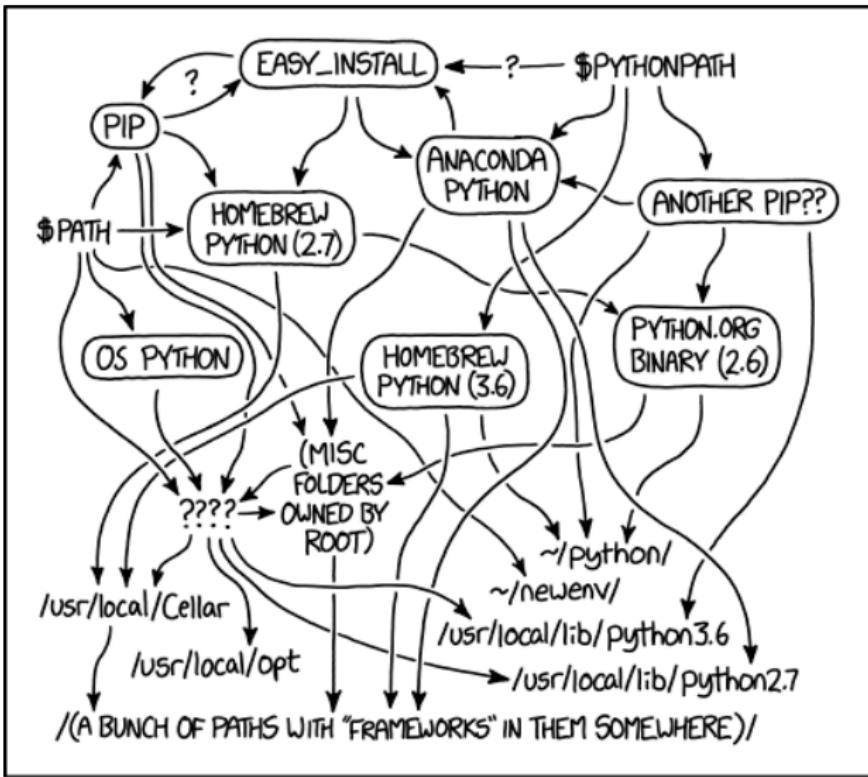
# Reproducible research

- Often researchers write code and then leave it unused for a while.
- Software gets updated over time.
- Breaking changes may be introduced.
- Packages may be unmaintained and fail to work.
- Code you wrote years ago may not produce the same results any more, or fail to run at all.
- Preventing this is crucial for reproducible research.

# Virtual environments

- If your projects rely on software from your global system environment, they will be prone to breakage and software rot.
- Virtual environment can help prevent this.
- Tool to keep dependencies required by different projects separate by creating isolated virtual environments for them.
- Environments can be exported/shared/saved.
- Exact dependencies incl. version numbers are specified in environment files.
- For Python: `venv`, `pipenv`, `conda env`, ....
- For R: `renv`, `conda env`.
- Most R libraries are now hosted on conda-forge (also easy to add).

# Try to stick with one solution



MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED  
THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

## More on reproducibility

- For setting up a reproducible project workflow, look into **snakemake**.
- Allows specifying all parts of a project (e.g. data management, analysis, input and output files of each step taken).
- If you want to set up a fully self-contained replication environment, look into **docker**.
- The [Rocker project](#) provides docker container images for R.
- Docker containers are also used to deploy models in production as they bundle their own dependencies and configuration and are relatively lightweight.
- [The Turing Way Handbook](#) is a great resource to get a broad overview over different options.

# Simple CLI usage

- Return executes a command, *CTRL + c* aborts a process, *CTRL + d* quits.
- Multiple commands can be run as scripts (suffix `.sh`).
- Comment character is `#`.
- Some examples:

```
sudo apt install vlc # install vlc video player
cd somedir/subdir # Navigate to a folder, (c)hange (d)irectory
cd .. # Navigate to parent directory
cd # Return to your home folder
ls # List directory contents
mv file subdir/file # Move a file
cp file ../file # Copy a file
rm myscript.ch # Delete a file
R # Start the R console
man ssh # Display manual pages for the ssh program
shutdown -h +20 # Shut off computer in 20 minutes
```

# Resources

- Basic knowledge often sufficient. Examples cover some ground.
- Google and O'Reilly/No Starch books and pocket references are handy.
- [Software carpentry](#) has an introductory course.
- [\(Brief\) Introduction to the Bash Command Line](#)
- [Advanced Bash-Scripting Guide](#)

## More examples

```
# Secure shell login to remote server
ssh myusername@13.438.14.673

# Edit your R script with vim
vim myscript.r
# Search for file and open it
vim $(find -name somefile.txt)

# Execute your R script
R -f myscript.r
# Execute your Python script
python myscript.py
# Execute a shell script
sh myscript.sh

# Stage file for version control
git add myscript.py
# Commit file to version control tree
git commit myscript.py
```

# Searching and filtering

- Great capabilities for pattern matching.

```
# Rename files by pattern, e.g. 'data_file1.csv' -> 'file1.csv'  
rename data_ '' data_*.csv
```

- Searching/filtering of text is a very popular application.
- Popular unix filters

- cat, head, tail, tee, wc, ...
- cut, paste
- find
- sort, uniq
- diff, comm, cmp
- grep (or ag, ripgrep), sed, awk
- (perl, python, ...)

# Searching and filtering

- Sounds tedious? It can be. But it can also be extremely powerful.
- `grep` is a popular command line utility for searching plain text files.

```
# Search for keyword in filename  
grep "keyword" filename  
# Search for keyword, (i)gnore case, (r)eursively, all files  
grep -ir "keyword" *  
# Print matching line and line (n)umber to output-file  
grep -n "keyword" *.txt > output-file
```

## Search patterns

- Convert all pdf files in a folder to text and search them.

```
for file in *.pdf; do pdftotext "$file"; done  
grep -ir "keyword" *.txt
```

- Searching can be more complex than just keywords.
- Suppose you need to find all documents with sentences expressing proficiency in Spanish or Italian, but not English.

```
grep -ir  
"(fluent|proficient)(?!.*?english)[^\n]*?(spanish|italian).*\n*.txt
```

## Regular expressions

---

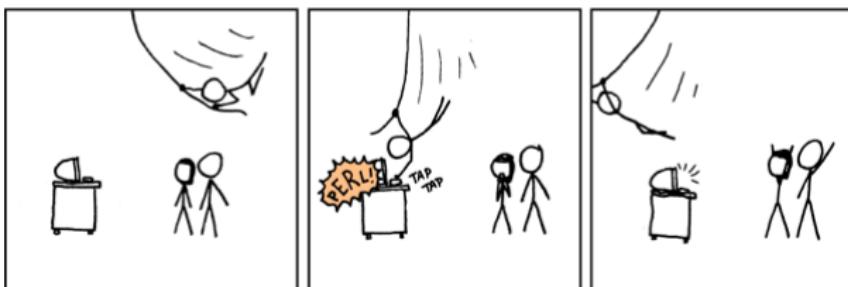
# Regular expressions

- Working with text requires understanding **regular expressions**.
- **grep**: globally search a regular expression and **print**.
- A regular expression is a character sequence that describes a set of strings.
- Regular expressions are constructed analogously to arithmetic expressions, by using various operators to combine smaller expressions.
- Usually used for find/replace operations on strings, or for validation.
- Pervasive in Unix text processing programs (**grep** was originally written by Ken Thompson).
- Language-independent concept. Not limited to command line search tools.

# Regular expressions

- *Pattern matching:* Find one of a specified set of strings in text.
- Examples:
  - Diagnoses in medical records.
  - Addresses or zip codes in concatenated admin records.
  - Sequences within a genome, e.g. a virus signature.
  - Validate data-entry fields (URL, date, email, credit card #).
  - Example using a regex tester.

# Obligatory xkcd



## Examples

AHVN 13:	$756\.[0-9]\{4\}\.[0-9]\{4\}\.[0-9]\{4\}\.[0-9]\{2\}$
Matches:	756.1234.5678.90
Does not match:	123.45.678.675
US-SSN:	$[0-9]\{3\}-[0-9]\{2\}-[0-9]\{4\}$
Matches:	166-11-4433
Does not match:	11-55555555
Email addresses:	$[a-z]+\@[([a-z]+\.)+(ch edu com)$
Matches:	someone@unibas.ch
Does not match:	someone@invalid.domain

# Screening job candidates

“ [First name]! and pre/2 [last name] w/7  
bush or gore or republican! or democrat! or charg!  
or accus! or criticiz! or blam! or defend! or iran contra  
or clinton or spotted owl or florida recount or sex!  
or controvers! or fraud! or investigat! or bankrupt!  
or layoff! or downsiz! or PNTR or NAFTA or outsourc!  
or indict! or enron or kerry or iraq or wmd! or arrest!  
or intox! or fired or racis! or intox! or slur!  
or controvers! or abortion! or gay! or homosexual!  
or gun! or firearm! ”

— *LexisNexis search string used by Monica Goodling  
to illegally screen candidates for DOJ positions*



<http://www.justice.gov/oig/special/s0807/final.pdf>

# Regular expressions

- Characters in a regular expression are either regular characters (literal meaning) or metacharacters (special meaning).
- Generally, letters and numbers match themselves.
- Normally case sensitive, but can be set to ignore case.
- Careful with punctuation, most of it has special meanings.
- To match metacharacters literally, they need to be *escaped*, i.e. preceded by a backslash \.

## Matching string literals

Regular expression	Input string
input	This input string is short.
15	The due date is 15.12.

## Matching string literals

Regular expression	Input string
input	This <b>input</b> string is short.
15	The due date is <b>15.12</b> .
15.12	The due date is <b>15.12</b> .

# Matching string literals

Regular expression	Input string
input	This input string is short.
15	The due date is 15.12.
15.12	The due date is 15.12.
but:	
15.12	The due date is 15712.
match . literal:	
15\.12	The due date is 15712.
15\(.12	The due date is 15.12.

# Regex basics

Operation	Regular expression	Input string
concatenation	foobar	Matches foobar but not foo or bar.
disjunction	this that	Matches this or that.
closure	like.* apples	I like apples, Peter likes apples.
	like. apples	Mary also likesALKFHEDL apples.
	like. apples	Mary also likesALKFHEDL apples.
parentheses	(He She) likes	She likes apples. He likes apples.
	(He She).*(very )*much\.	She likes apples very very much.

- More complicated patterns can be expressed via concatenation, disjunction, repetition and scope.
- Precedence in descending order.

# Quantifiers

Character	Matches
*	0 or more instances of preceding char
+	1 or more instances of preceding char
?	0 or one instance of preceding char
{m}	exactly m instances of preceding char
{m,n}	m through n instances of preceding char
{m,}	m or more instances of preceding char
{,n}	up to n instances of preceding char
?	add to a quantifier to match ungreedy

- Quantifiers match greedily by default (i.e. the longest string possible).

Ex: `^begin.*end` will match ‘begin bla bla end bla end’.

`^begin.*?end` will match ‘begin bla bla end bla end’.

# Groups, ranges and character classes

Character	Matches	Example RE	Matches
.	Any character, except \n	like.	likes like! like like
(a b)	a or b	(you me)	you or me
[ab]	Character range	202[01]	2019 2020 2021 2022
[a-z]	Character range	[A-Z][a-z]*	Capitalized words
[0-9]	Digit range	20[0-9]{2}	Years in the 21st century
[^ab]	Any character but (negation)	20[^0][01]	2000 2010 2020 2025 2031

- Quantifiers, ranges and other shortcuts improve expressiveness.  
Ex: [A-E]⁺ is shorthand for (A|B|C|D|E)(A|B|C|D|E)\*.
- More character classes (sometimes) available.  
Ex: \w for words ([A-Za-z0-9\_]), or \d for digits ([0-9]), \a, \s, ...

# Anchors and other special characters

Character	Matches	Example RE	Matches
^	Beginning of a string	^New	New research in this field
\$	End of a string	[A-Za-z]+:\$	A breakthrough! Finally!
\n	Newline		
\t	Tab		
...	...		

- Strings can stretch multiple lines.
- Character encodings can sometimes cause problems. Stick to UTF-8.

# Regex syntaxes

- More elaborate regex syntaxes also support positive and negative lookahead/lookbehind, conditionals and group references.  
Ex: `^(?!.*word).*$` matches lines not containing a word.
- Different syntaxes (basic, extended, perl, vim, ...) mostly similar with regard to basic features.
- Perl-compatible regular expressions (PCRE) is the de-facto standard.
- Most regex implementations feature switches to invert the search pattern, to ignore case, and more.

## Example: Valid RFC-822 email addresses

<http://www.ex-parrot.com/pdw/Mail-RFC822-Address.html>

## Remarks

- Writing a regular expression is like writing a program.
  - Requires understanding the programming model.
  - Can be easier to write than read.
  - Can be difficult to debug.
- ➡ Break up problems into smaller pieces. Try not to do everything in one large regex. Comment liberally.

# Resources

- Pin a [cheat sheet](#) to your office wall.
- Regular expression tools help (e.g. [regex101.com](#)).
- Simple interactive tutorial: [regexone.com](#). Also learn regex the easy way.
- For regular expression syntax specific to R, look up [this short tutorial](#).
- [This guide](#) also provides a more detailed overview of working with strings in R.

## Remarks

- Regexes are a powerful tool.
- Easy to grasp, complex to master.
- Using them in applications can be complex and error-prone.
- Regular expressions are not parsers.

*“Some people, when confronted with a problem, think ‘I know, I’ll use regular expressions.’ Now they have two problems.”*

*Emacs newsgroup*

Next lecture: Web scraping basics

## References

---

## References

-  Barrett, D.J. (2016). *Linux Pocket Guide*. Third edition. Beijing ; Boston: O'Reilly.
-  Chacon, S. and B. Straub (2014). *Pro Git*. Apress.
-  Fitzgerald, M. (2012). *Introducing Regular Expressions: Unraveling Regular Expressions, Step-by-Step*. 1. ed. Beijing: O'Reilly.
-  Matloff, N. (2011). *The Art of R Programming: A Tour of Statistical Software Design*. No Starch Press.
-  Robbins, A. (2000). *Sed & Awk Pocket Reference*. 1st ed. Beijing ; Sebastopol, CA: O'Reilly & Associates.
-  Robbins, A. (2010). *Bash Pocket Reference*. 1st ed. Beijing ; Sebastopol, CA: O'Reilly.
-  Shotts, W. E. (2019). *The Linux Command Line: A Complete Introduction*. Second edition. San Francisco: No Starch Press.