DS Class Notes

Notes for CSC482/DSC205 Introduction to Advanced Data Science Spring 2022

README

Instead of bugging you with emails, I opt to summarize my course observations regarding content, process, in this file. These often contain additional links, articles, and musings.

I usually update it after each class - it also contains the homework (if any). The first point of call for any questions should be the FAQ. There are two FAQs - a general one (for all my courses), and a FAQ for CSC 100.

You find the whiteboard photos here in GDrive.

The companion file to this file, with the agenda and much of the course content, is the agenda.org file.

Course introduction - w1s1 (01/12/22)

See also: Google Meet chat

Homework (until Tuesday, 18 Jan, 11:59 PM)

IF YOU DID NOT COMPLETE DSC101	IF YOU COMPLETED DSC101
Complete "Datascience for everyone" on DataCamp	
Complete "Introduction to R" on DataCamp	
Pass the Entry Quiz (Schoology) > 50%	Complete the Entry quiz (Schoology)

Stuff

Data science pipeline

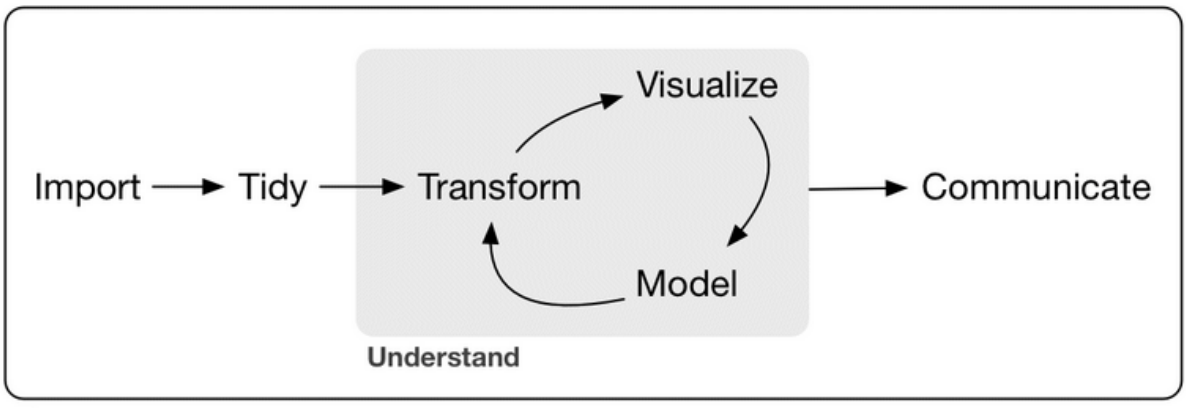


Figure 1: Data science pipeline (Source: Wickham/Grolemund 2017)

Books

None of which will be an exclusive, but I may use stuff from these books. They're all good in their own way but a little hard on one's own.

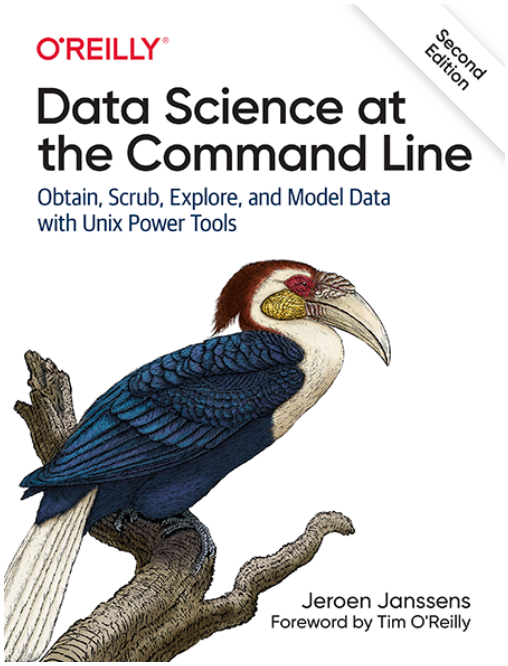


Figure 2: Data Science at the Command Line by Jeroen Janssens

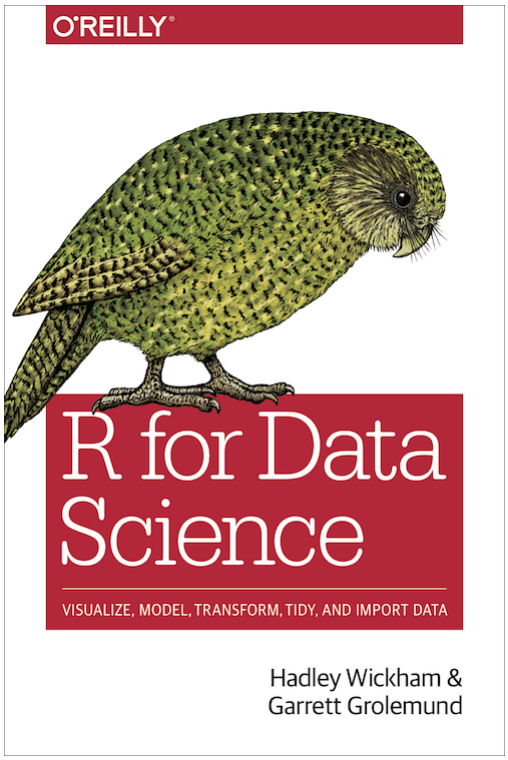


Figure 3: R for Data Science by Wickham/Grolemund

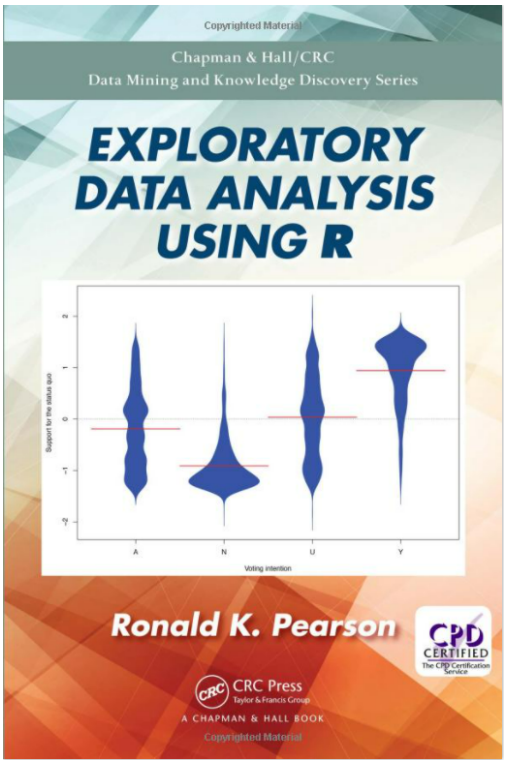


Figure 4: Exploratory Data Science Using R by Ron Pearson

Regular expressions

Important for efficient text mining and string manipulation, e.g. when doing data science on the command line, regexp are search patterns. Here is a complete, free, online tutorial (RegexOne, 2021), and here is a free book chapter explaining regexp as part of automating stuff with Python (Sweigart, 2019).

Examples for such regular expressions are the `*` in an SQL command like `SELECT * FROM t` to query all columns of the table `t`, or `^x` that matches any string starting with `x` etc.

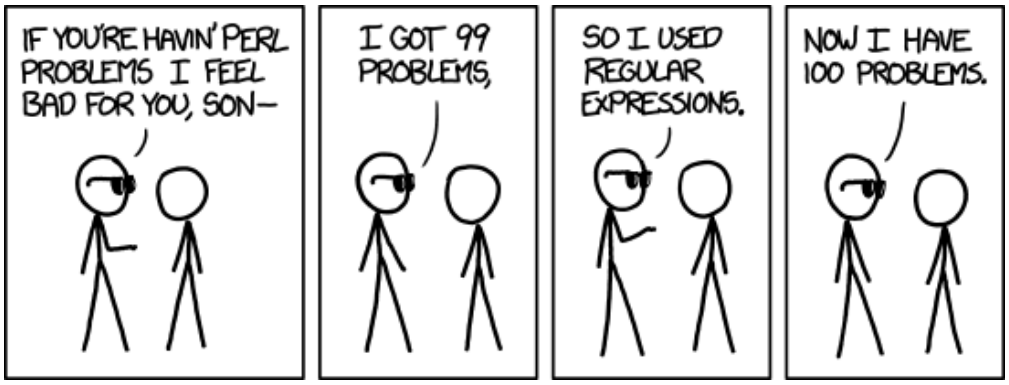
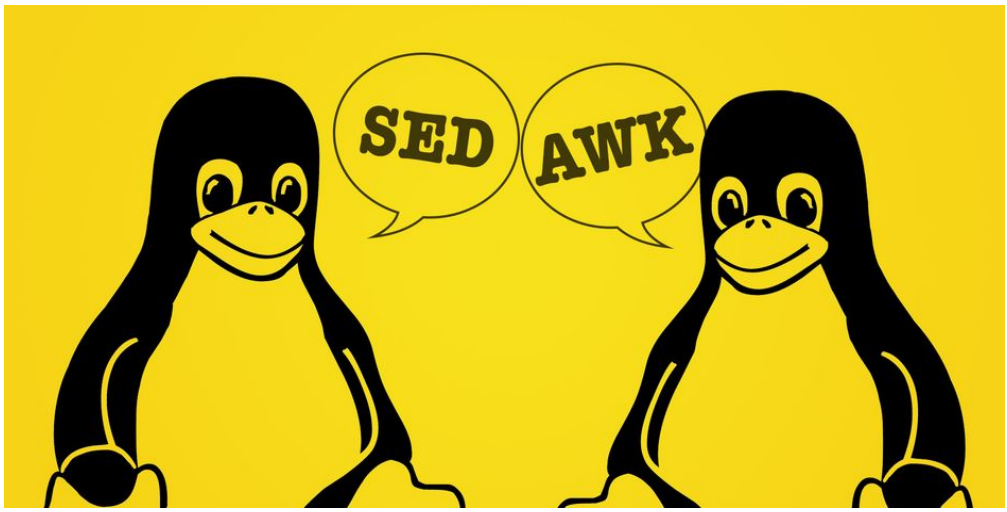


Figure 5: Perl problems (Source: xkcd).

`Perl` from the cartoon title, is another powerful language whose strength is pattern matching and manipulation. It's more high level than `awk` or `sed` though and lives on all operating systems.

`awk` (and `sed`)

This is an "awkward" language on GNU/Linux. They're natural languages for regexp use. Makes data wrangling on the command line reall easy. Not hard to learn, and we might take a look at it - I plan to present it in another class (operating systems) as part of the Linux layout. Here's a tutorial for `awk`, and an opinion piece (Hughes, 2015). Something else for a rainy afternoon.



Getting started with GNU Emacs

GNU Emacs is going to be our IDE and our environment for literate programming. This is an experiment that I'm running this term in all my courses - but this course (R) and the intro class on C/C++ are the two classes where Emacs should really pay off.

I suggested two short videos to get started while munching a bagel:

- First Steps With Emacs (Eddelbuettel, 2021). This is especially for RStats people (like you), with a focus on ESS ('Emacs Speaks Statistics').
- Literate programming demo (Birkenkrahe, 2022). Here I contrast Emacs Org-mode with an interactive shell using SQLite, an RDBMS.

We'll get deeply into this soon as we set up our infrastructure.

Notebooks and notebook platforms

There are many interactive notebooks and notebook platforms - they're especially popular in data science (and perhaps data science is so popular, and easier to learn because of them).

Some examples: Jupyter (originally only for Python), Google Colaboratory (for Python and R - though with "magic" commands, one can use other languages, but it's not straightforward), and Kaggle (owned by Google). Kaggle serves notebooks, datasets and (most importantly) data science competitions (strong focus on machine

learning). These are often quite ideological ("Save the whales with data science") but what isn't these days? Which is why data science needs strong bias monitoring[1].

Installing R / Windows PATH - w1s2 (01/14/22)

R

TO DO	WINDOWS
Download base R from CRAN	R 4.1.2 "base"
Run installer	
Check files	C:/Program Files/R
Go to the binary folder	c:/Program Files/R/R-4.1.2/bin/x64
Open R GUI	Rgui.exe
Open R terminal	Rterm.exe
Check Rscript	Rscript test.R
Check PATH	

Log

- Short rant about Python vs R and why you learn R (vonjd)
- Showed R console and Rscript in DataCamp
- Showed R in a Windows (CMD) terminal
- Showed R inside Emacs in a terminal (no syntax highlighting)
- At CRAN, we want "base R" (without packages)
- The current version of R (Jan'22) is 4.1.2 "Bird Hippie"
- Normally, before running executables: check the "checksum" (Hoffman,2019)
- Run the installer, accept standard suggestions
- Start the launcher from the desktop
- GUI appears (`Rgui.exe`)
- Saving the workspace image stores `.RData`, `.Rhistory`, and `.Rplots` files containing (binary) data, command history, and PDF plots, respectively
- Update the `PATH` variable (search for PATH) using the string from the file explorer that contains the path to `bin/`
- Apparently, you don't have to do this in Windows 11 (but don't rely on it - better find out how to drive with stick shift!)
- Open a Windows terminal ("CMD")
- Start R (enter R)
- Test R with some commands like in the 1 code block.

```
plot(rnorm(100))
3 + 4
x <- rnorm(100)
str(x)
plot(x)
q()     # you can save your workspace image (don't)
```

- If you have any installation issues: check the R FAQ first

Installing and setting up GNU Emacs - w2s3 (01/19/22)

Emacs+ESS

TO DO	WINDOWS
Download Emacs+ESS	Download Installer
Run installer	Standard config
	Desktop shortcut
Check README	*Opens after installation*
Check Emacs	`emacs -nw` in terminal / desktop shortcut
Set PATH	*requires admin privileges*

Log

- If you don't have the modified GNU Emacs (with ESS already installed), you need to install and load the `ess` package
- See install.org (+ PDF) in the org/emacs GitHub repo for installation instructions if you want to put this on your own PC
- GNU Emacs layout: buffer window + modeline + minibuffer
- Commands begin with C-x (CTRL+x) or M-x (ALT+x)
- C-g interrupts any process
- List of open buffers: C-x C-b
- Change to other buffer: C-x o
- Close all visible buffers except one: C-x 1
- Start R (if installed and PATH set correctly): M-x R
- This opens an R session in the current directory (iESS mode)

Understanding Emacs Org-mode - w2s4 (01/21/22)

This class will get the most intense exposure and training for GNU Emacs, because of the need to work with interactive notebooks in data science. Getting to play around in Emacs in other courses (Databases, Operating Systems) will only improve your editor skills.

What we did using the instructions from tutor.org:

- Downloaded GitHub directory with `.org` files
- Opened `.org` files permanently with GNU Emacs
- We covered:
  - header options in Org-mode
  - moving around in Emacs buffers
  - opening/closing/suspending Emacs (also from the cmd line)
  - reading a file into Emacs, and saving it
  - opening buffer list and directory
  - switching buffers
  - creating a region, killing and yanking it

    - changing the font
    - opening the onboard tutorial
    - aborting commands
- We'll rehearse these in our weekly quiz on Monday!

- To get better, work through the tutorial (C-h t)

  See also the article "Getting started with Emacs" (Kenlon, 2020), and the video "The Absolute Beginner's Guide to Emacs" (System Crafters, 2020) with my notes.

Customizing Emacs (init file) - w3s5 (01/24/22)

Planned:

Practice	GNU Emacs Tutorial cont'd (tutor.org)
- Package manager	`M-x package-list-packages RET`
- Start R shell in Emacs	`M-x R` (R must be installed & in the `PATH`)
- Add init file	.emacs sample file (GitHub)
- Create `first.org` file	`C-x C-f ob.org RET`
- Create R code block	`#+begin_src R :session :results output ...#+end_src`
- Run R code block	`C-c C-c`

Captain's Log

See tutor.org for details:

- We added .emacs file in the ~/ HOME directory and discussed its content and structure (Emacs-Lisp) - especially the Org-babel packages.
- We talked about the Org-mode file assignment.
- After restarting Emacs (to load the configuration file), we opened the package manager with `M-x package-list-packages`. If the `.emacs` file is in the right location, the package manager should refresh its content.
- The package manager lists many downloadable packages. You downloaded the `org-beautify-theme` and `org-bullet` - both packages to improve the appearance of Emacs.
- Here is the Emacs documentation on the initialization file `.emacs` in the GNU Emacs manual: "How Emacs finds your init file".
- By default, Emacs will open to `default-directory`. This is a variable that you can set in your `.emacs` file. Here is an example where the working directory is set to `C:\Users\birkenkrahe\Emacs`

```
(setq default-directory "c:/Users/birkenkrahe/Emacs")
```

Notice how Windows requires backslashes, while Emacs (and Unix/Linux) use forward slashes.

Running code in Org-mode 1 - w3s6 (01/26/22)

- When you look at an Org file as a PDF or on GitHub, you will not see the meta data starting with `#+`. Org-mode files are meant to be edited/viewed in Emacs.

- The code block header has the following arguments:

HEADER ARGUMENT	MEANING
`:session *R*`	Run R in a session in the Emacs buffer `*R*`
`:results output`	insert output directly in the org file
`:tangle first.R`	export source code as R file `first.R` ("tangle")
`:exports both`	both result and source code will be exported
`:comments both`	link source code and org files, add comments to source

Running code in Org-mode 2 - w3s7 (01/28/22)

- Feel free to bring your own laptop to future sessions. If you want me to check installation because something did not work, come a little earlier or stay a little later.
- This concludes our "Emacs week". To get more practice in GNU Emacs, complete the onboard tutorial (`C-h t`), and of course there's still one (simple, text-only) Org-mode assignment outstanding.
- Solutions to the Org-mode assignment are posted here on GitHub. Note that submissions of programs as Org-mode files should always also be accompanied by references and sources.
- I told you an inaccuracy in class: when rendering the Org-mode file on GitHub, the `#+TITLE` meta information is displayed as the title of the file. If no such header is present, only the `README` file is displayed (with the file name as title).

Org-mode lab session - w4s8 (01/31/22)

- Setting the default directory (the folder where Emacs "wakes up" when you open Dired with `C-x d`):

```
;; set default working directory to c:/Documents/GitHub/
(setq default-directory "c:\\Users\\birkenkrahe\\Documents\\GitHub\\")
```

2022 Data Trends - w4s9 (02/02/22)

Notes from the DataCamp webinar (DataCamp, 2022)

Overview

- Great acceleration (2020) - reaction
- Great transition (2021) - recognition

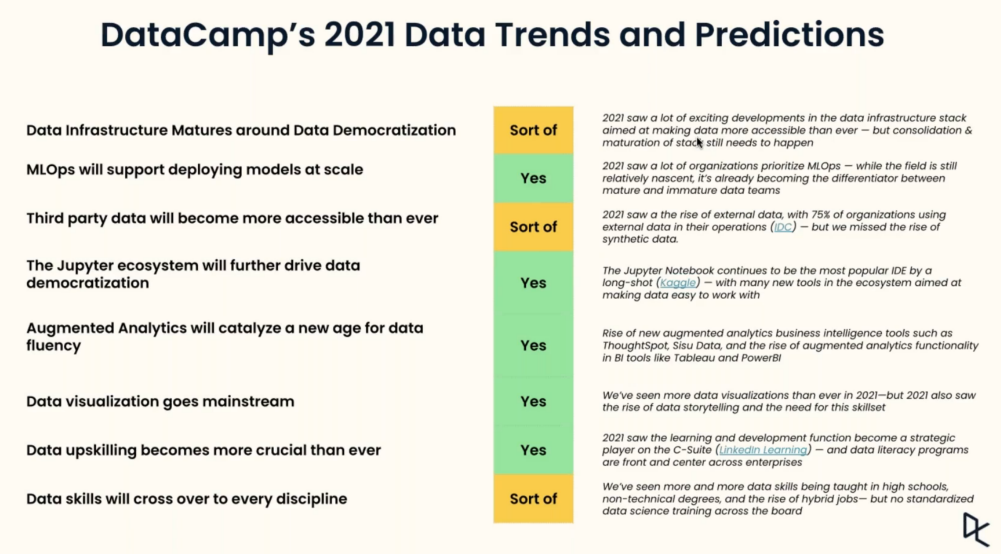


Figure 7: Prediction check 2021-2022

"Jupyter ecosystem"?

"Augmented analytics"?

- Operationalization of large language models: dashboards
- "Innovations in the data tooling stack" (more and better tools)

Trends

1. 100-fold increase prediction in data generated (2021-2025) according to Accenture
2. Data mesh vs data lake - Data PaaS - the issue is speed01234 (infrastructure is complex and slow-moving)
3. MLOps mature - report: mostly startups (= economically irrelevant)
4. Data tool stack grows
5. Learning & Development - "upskilling becomes a mandate". Cotton: "People on this call are weird. Most people do not voluntarily join a webinar on data trends." Hairdresser asked him about his job…"does this mean that you work with computers and stuff." The knowledge divide is huge.

6. Data governance and quality

    - data **catalog**
    - data **observability** (freshness)

    Documentation aids analysis (compare with Andrew Ng's initiative for more data tools and transparency). Independence of technical skill (no-coders).

7. NLP - e.g. PowerBI allows NLP descriptions of something you want to calculate, and it will auto-generate code/graphs for you. OpenAI: Codex allows for Python-from-description coding.

    Reverse: replit.com - don't have to read code anymore because the platform explains it to you.

8. Culture focus shift intensifies
9. Talent pool and talent generation will expand

Discussion / Groupwork results

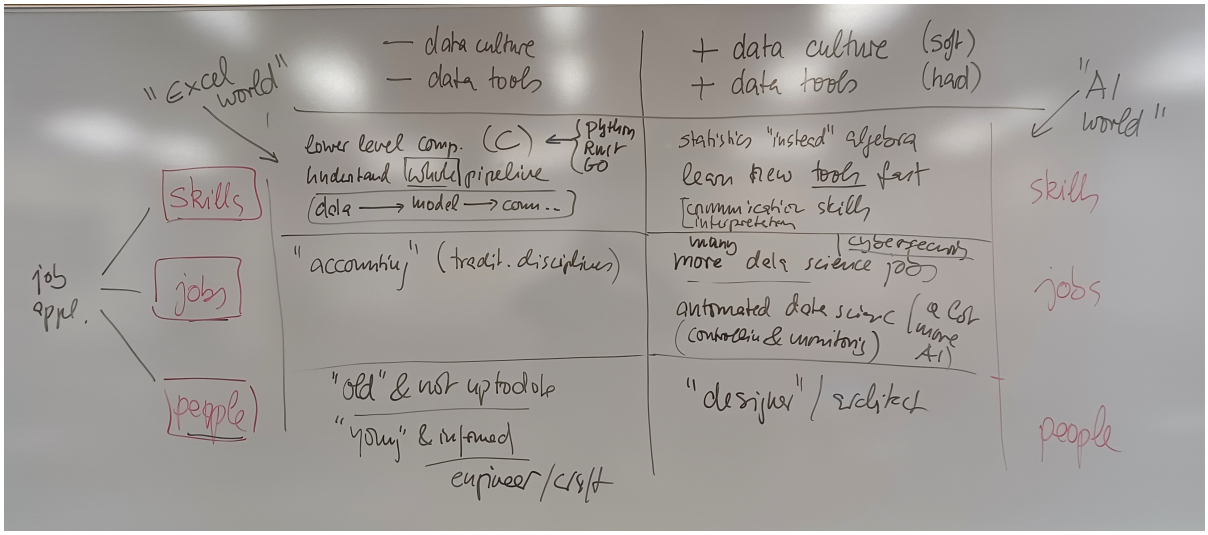


Figure 8: data trend scenarios

Studying with DataCamp - w5s10 (02/07/22)

- Simplify your Org-mode setup with `PROPERTY` settings
  - Add this at the top of your Org-mode file with code:

```
#+PROPERTY: header-args:R :session
#+PROPERTY: header-args:R :results output
```

  Restart the file or refresh with `C-c C-c` on the `PROPERTY` line, and now a code chunk like this should work fine:

```
str(mtcars)
```

  Try it now! (Documentation)

- Type the DataCamp exercises out as Org-mode files to get practice - both in Emacs Org-mode, but also in R (example)

The Anti-IF Campaign

You may have been mystified by my mentioning Cirillo and the Pomodoro time management technique but also IF-THEN-ELSE as a No-No in software engineering. This last issue was related to the "Anti-IF Campaign" launched (not tongue-in-cheek) by my friend Francesco Cirillo of Berlin (Cirillo, 2022):

#+begin_quote "The Campaign is against the use of the IF statement as a regular design strategy to deal with growth, change and complexity ("Let's Put an IF Syndrome") in an evolutionary context. Despite being an "easy," and apparently effective, way of delivering the value requested by the customer, this "design strategy"

has negative repercussions when applied regularly as the main strategy to deal with change, growth and complexity. By applying the "IF Strategy" in an evolutionary context, software systems becomes more complex to be read, tested, even debugged. It becomes easier to duplicate code, accumulate technical debt and spend more time fixing bugs. In the result, the software system become more complex. New features and changes will cost more and more." #+end_quote.

References

- Birkenkrahe (Jan 11, 2022). Interactive shell vs. interactive notebook (literate programming demo). URL: youtu.be/8HJGz3IYoHI.
- DataCamp (January 2022). Data Trends and Predictions 2022 [webinar]. URL: www.datacamp.com.
- Cirillo (2022). The Anti-IF Campaign [website}. URL: francescocirillo.com.
- Emacs Speaks Statistics (Mar 19, 2021). First Steps With Emacs [video]. URL: youtu.be/1YOrd7NCGkg.
- Hoffman (Sep 30, 2019). What is a checksum (and why should you care)? [blog]. URL: www.howtogeek.com.
- Hughes (Oct 30, 2015). Every Linux Geek Needs To Know Sed and Awk. Here's Why…[blog]. URL: www.makeuseof.com.
- Kenlon (March 10, 2020). Getting started with Emacs [blog]. URL: opensource.com.
- Pearson (2019). Exploratory Data Analysis Using R. CRC Press. URL: routledge.com.
- RegexOne (2021). Lesson 1: An Introduction, and the ABCs [tutorial]. URL: regexone.com.
- Sweigart (2019). Automating the boring stuff with Python. NoStarch. URL: nostarch.com/automatestuff2.
- System Crafters (March 8, 2021). The Absolute Beginner's Guide to Emacs [video]. URL: youtu.be/48JlgiBpw_I.
- vonjd (n.d). Why R for Data Science – and *Not* Python! [blog]. URL: blog.ephorie.de.
- Wickham/Grolemund (2017). R for Data Science. O'Reilly. URL: r4ds.had.co.nz.
- xkcd (n.d.). Perl Problems [cartoon]. URL: xkcd.com.

Footnotes:

[1] Who wouldn't want to save the whales! Still, even a seeminly harmless ideological thrust can lead to conflict. E.g. what if you only have enough project budget to either save the whales or starving children? That used to be a question for philosophy class - in data science, it's everybody's task - because data science is decision science.

Author: Marcus Birkenkrahe
Created: 2022-02-08 Tue 20:59
Validate