

preparing a better-looking thesis
in \LaTeX , R, and open-source software
in less time

Session slides

Andreï V. KOSTYRKA

27/11/2023 – 11/12/2023

Presentation structure

1. Administrative formalities
2. Setting up the computer for research
3. Windows-specific tweaks
4. Research workflow best practices
5. T_EX, L^AT_EX, and related tools

Administrative formalities

Why this course exists

- Every Ph. D. student or an academic researcher has to write papers and submit a thesis at the end of their studies
- There are too many tips on the Internet on so many topics – which ones should one read first? Which ones are obsolete? Which ones are unsound?
- Complements the TS course ‘Job market seminars in economics and finance’
 - No dedicated \LaTeX / open-source course \Rightarrow fills the gap
- Promotion of open science and reproducible research at Uni.LU

How this course was shaped

- Over the course of my doctoral studies, I received dozens of questions from fellow students
- They were struggling with tables and plots, manually searching for changes, losing the additions in dozens of versions, using a broken \LaTeX distribution...
 - Ph. D. time is limited, and one should not spend it struggling with the tools
- As a result, this course is based on DSEFM Ph.D. students' concrete questions and their past struggles with research articles, presentations, and media content thereof

Showcase

1. Thesis
2. Presentation
3. Poster
4. Book

Centrepiece: \LaTeX is everywhere

- Top economic journals (Ecta, QJE, JoFi, JoE, JPE, ...)
- Academic and technical literature, university textbooks
- R manuals, Stata manual, EViews manuals
- If you are seeing this, you are seeing \LaTeX output!
 - Many other presentations by FDEF members as well

How to pronounce ‘ \LaTeX ’

Donald Knuth, the creator of \TeX :

*The name \TeX is the uppercase form Greek root beginning with the letters $\tau\epsilon\chi$. It's the 'ch' sound in German words like **ach**. When you say it correctly to your computer, the terminal may become slightly moist.*

Leslie Lamport, the creator of \LaTeX :

One of the hardest things about LaTeX is deciding how to pronounce it. <...> lah-teck, and lay-teck are logical choices.

L^AT_EX and yours truly

- Discovered L^AT_EX in 2010, immediately fell in love – since then, a true passion
- Helped me not only in Ph.D., but in real life (raised my æsthetic standards)
- Daily coding since then (not only for science – for invoices, reports, poems as well)

What awakened my desire to learn \LaTeX

1. Sergueï Lobanov and Dmitriy Schwarz produced astonishingly beautiful textbooks that made us, 17-year-old bachelor students, speechless (**show!**)
2. Word crashing with 300-page documents (work stalled)
3. *Street cred*: sharing lecture notes in PDF (and needing a good tool for that) instead of hundreds of hand-written scribbles shot on a crummy Nokia E63 camera
 - I heard a story of a student who was attending lectures with a laptop, typing everything down, and selling the print-outs – and decided that knowledge should be free and accessible to every student (I would prepare the full digital notes for myself any way) – and typed all lectures and session on an Asus EeePC 1001HA

My first \LaTeX document (2 hours in 2010)

ЗАЧЁТНАЯ РАБОТА ПО ТЕОРИИ ВЕРОЯТНОСТЕЙ. ВАРИАНТ 3.

1. Совместная плотность распределения случайных величин X и Y задана формулой:

$$f(x, y) = \frac{1}{2\pi} \frac{1}{\sqrt{1 - \rho^2}} e^{-\frac{1}{2(1 - \rho^2)}(x^2 - 2\rho xy + y^2)}$$

Найти $E(X)$, $D(Y)$, $\text{Cov}(X, Y)$, $P(\{X > Y - 1\})$.

2. Случайные величины X , Y , Z независимы и стандартно нормально распределены. Вычислите $P(\{X < \sqrt{2}\})$, $P(\{\frac{|X|}{\sqrt{Y^2 + Z^2}} > 1\})$, $P(\{X^2 + Y^2 > 4\})$.
3. Доходности акций двух компаний являются случайными величинами X и Y , имеющими совместное нормальное распределение с математическим ожиданием $\begin{pmatrix} 2 \\ 2 \end{pmatrix}$ и ковариационной матрицей $\begin{pmatrix} 4 & -2 \\ -2 & 9 \end{pmatrix}$.
Найти $P(\{X > 0\} \mid \{Y = 0\})$. В каком соотношении нужно приобрести акции этих компаний, чтобы риск (дисперсия) получившегося портфеля был минимальным? *Подсказка:* если R — доходность портфеля, то $R = \alpha X + (1 - \alpha)Y$. Можно ли утверждать, что случайные величины $X + Y$ и $7X - 2Y$ независимы?
4. Пусть X_1, \dots, X_n — независимые одинаково распределённые случайные величины с плотностью распределения $f(x) = \frac{3}{x^2}$, $x \geq 1$. Применим ли к данной последовательности закон больших чисел? С помощью неравенства Чебышёва определить, сколько должно быть наблюдений в выборке, чтобы $P(\{|\bar{X} - E(X)| > 0,1\}) \leq 0,02$.
5. В большом-большом городе N 80 % аудиокиосков торгуют контрафактной продукцией. Какова вероятность того, что в наугад выбранных 90 киосках более 60 будут торговать контрафактной продукцией? Каким должен быть объём выборки, чтобы выборочная доля отличалась от истинной менее чем на 0,02 с вероятностью 0,95?

My first \LaTeX document (2 hours in 2010)

6. У входа в музей в корзине лежат 20 пар тапочек 36–45 размера (по 2 пары каждого размера). Случайным образом из корзины вытаскивается 2 тапочка. Пусть X_1 — размер первого тапочка, X_2 — размер второго. Являются ли случайные величины X_1 и X_2 зависимыми? Какова их ковариация? Найти математическое ожидание и дисперсию среднего размера $\frac{X_1 + X_2}{2}$.
7. В страховой компании «Ай» застрахованные автомобили можно условно поделить на 3 группы: недорогие (40 %), среднего класса (50 %) и дорогие (10 %). Из предыдущей практики известно, что средняя стоимость ремонта автомобиля зависит от его класса следующим образом:

	Недорогие	Среднего класса	Дорогие
Математическое ожидание	1	2,5	5
Стандартная ошибка	0,3	0,5	1

В каком соотношении в выборке объёма n должны быть представлены классы автомобилей, чтобы оценка средней стоимости ремонта (стратифицированное среднее) была наиболее точной?

8. Реализацией выборки $X = X_1; \dots; X_6$ являются следующие данные: $-0,8; 2,9; 4,4; -5,6; 1,1; -3,2$. Найти выборочное среднее и выборочную дисперсию, вариационный ряд и построить эмпирическую функцию распределения.
9. По выборке $X_1; \dots; X_n$ из равномерного распределения $\mathcal{U} \sim [0; \theta]$ с неизвестным параметром $\theta > 0$ требуется оценить θ . Будут ли оценки $T_1 = 2\bar{X}$, $T_2 = (n+1)X_{(1)}$ несмещёнными? Какая из них является более точной (эффективной)? Являются ли эти оценки состоятельными?

Дополнительная задача (не является обязательной). Случайные величины X и Y независимы, причём $\mathbb{P}(\{X = k\}) = \mathbb{P}(\{Y = k\}) = pq^{k-1}$, $0 < p < 1$, $q = 1 - p$, $k = 1; 2; \dots$. Найти $\mathbb{P}(\{X = k\} \mid \{X + Y = n\})$, $\mathbb{P}(\{Y = k\} \mid \{X = Y\})$.

Open-source and yours truly

1. Installed Linux in 2012
 - Hopped many 'distros': Ubuntu, Mint, Debian, Arch, Manjaro
2. Discovered the vast world of nice tools that *just work*
3. Set up dual-boot Windows + Linux (Mint) on 4 computers of family members, have not received a *single* bug / virus / feature complaint in 9 years
 - Digital reliability is the #1 priority if one cannot physically assist with PC troubleshooting
4. Learned regular expressions in 2013, started compiling software from sources in 2014
 - A lot of convenient tools are written in source code and require compilation – expanded the arsenal with them (although the first steps were not easy)

What this course is not

- Not a comprehensive \LaTeX programming tutorial
 - Just what you need for a thesis / presentation / poster
- Not computer science or algorithm analysis
- Not a UNIX / open-source / Windows / Mac / Linux / computer repair tutorial
 - The HPC school often includes introduction to command-line tools for use under Linux on the Uni.LU HPC

What this course is

- Basic understanding of the core \LaTeX syntax
 - Writing macros to save time
 - Using good-looking templates and showing how to change them (provided at the end of the course)
- Self-help: Handling errors and warnings, getting help, troubleshooting compilation failures
- Learning how to chain together open-source utilities that can perform specialised tasks really well
- Creating simple uncluttered plots and tables

Course goals

- Help researchers become fluent in the technical language of modern academic publishing
- Teach how to improve their papers and theses in terms of typographic quality
- Guide how to proceed in case of troubles or errors
- Open up the world of time-tested open-source tools ('one-trick-ponies') and Unix paradigm in general

Tools mastered in this course

- Focal point: $\text{T}_{\text{E}}\text{X}$ and its extension, \LaTeX
 - Offline distribution ($\text{T}_{\text{E}}\text{XLive}$), online compiler (Overleaf), bibliography processor (biber and $\text{BibT}_{\text{E}}\text{X}$)
- Library management with JabRef
- Proof-reading, spell-checking
 - LanguageTool, $\text{ChkT}_{\text{E}}\text{X}$, online resources
- Plain text editors and version control
 - Regular expression support, version management, change tracking, GitHub repositories
- Working with tables and images
 - Basics of raster image editing
 - R scripting for graphics and table generation (you can get a full set of slides for the R course)

Tasks you will be able to complete

1. Typeset an article that would make their advisor happy / not angry / less angry
2. Implement journal style requirements with minimal pain
3. Automate certain routines (like formatting the output of scientific software)
4. Produce non-standard graphics
5. Solve that one critical error on the night of submission

During the course

- 10-minute break in the middle
- The schedule is not as intense, but the time is limited (only 5 sessions)
 - Study at home, write down questions about what is unclear, ask them during the sessions
- Having a laptop is completely optional (you can follow the screen), but running the code at least once in your spare time is a must

The syllabus

- Contains the intended agenda
- Contains links to openly published learning resources (books, online tutorials etc.)
 - Extra material will be provided on Moodle
 - Warning: many \LaTeX learning resources are outdated (newer best practices will be shown; 2010+ should be good)
 - Your suggestions of learning resources are welcome
- Contains homework descriptions and full final project proposals

Grading

- 5 · 2% attendance + 30% small assignment + 60% project
- The assignment is very short, and its result is infinitely reusable afterwards
- Final project: **choose the task that is the most relevant for your research** or the one that can be later reused in other projects
 - You may reuse your existing material in the assignment

Technical requirements for all assignments

1. No restrictions on the compiler (pdf \LaTeX or Lua \LaTeX) or CTAN packages, but the submissions must be compilable with full \TeX Live 2023 or on Overleaf **without errors**
2. The amount of warnings and overfull / underfull messages must be brought to a minimum (ideally zero)
3. File formats:
 - .tex, .bib, .R, .py, .rb for the documents / data bases / content generators
 - .png / .pdf / TikZ for graphics
 - .bat or .sh for toolchains
 - .docx, .odt, external .pdf are **not accepted**;
 - .jpg graphics are strongly discouraged

Any questions on the formalities?

Setting up the computer for research

Open-source software

Open source: access to the source code + distribution conditions:

- Any can give away or sell the software (even as a component)
- There must be a well-publicized means of obtaining the source code
- The license must allow modifications and derived works
- Extra legal fluff: no discrimination, distribution of licence, no restrictions on other software.

Perfectly aligns with the Code of Conduct of the University of Luxembourg.

Free software

‘Free software’ = libre software = FOSS software = slightly more than OSS = 4 freedoms:

1. To use for any purpose
2. To study how the programme works, and change it to make it do what one wishes
3. To redistribute and make copies
4. To improve the programme, and release your improvements and modified versions to the public

Free software \neq freeware: non-commercial software can be proprietary / free-download closed blobs (requiring disassembly for debugging).

UNIX systems and POSIX

UNIX: family of multi-tasking operating systems popular in the 1970s – 1980s.

POSIX: Portable Operating System Interface = family of standards for maintaining compatibility between operating systems.

- The backbone of almost every server or mobile device.
Unix + C has held for 40+ years
- The tools developed for UNIX are still perfectly relevant on modern systems

UNIX design philosophy

- Make each programme / function do one thing well
 - Make it easy to test and run programmes
- Write programmes to work together / handle streams
 - Expect the programme output to become input to another (unknown) programme
 - Use plain text for data storage
- Do not hesitate to throw away and rebuild clumsy parts
- Use tools in preference to unskilled help, even if you have to build them for a single run
 - Your inputs may change at any time, requiring re-calculation

Toolchains

Title image for this course: [cow aerodynamics](#). No manual actions – 3 lines with commands:

1. `yt-dlp` to fetch a video from YouTube
`yt-dlp -f bestvideo[ext=mp4] -o cow.mp4`
`https://www.youtube.com/watch?v=goucqTkRAP0`
2. `ffmpeg` to convert the frames to PNG
`ffmpeg -i cow.mp4 -ss 01:37 -to 02:26 %06d.png`
3. `imagemagick` to average thousands of frames into one
`convert -verbose -limit memory 200MiB`
`-limit map 400MiB 00*.png -average average.png`

Most modern data providers provide APIs that *have to* be used via scripts – because otherwise a human life is not long enough to click the buttons manually.

GNU core utilities

GNU coreutils: a package of many basic tools (cat, ls, rm) that are used on Unix-like operating systems.

- Delete all 10 000 JPG files even when the 'Estimated time' is growing with zero sign or progress
`rm *.jpg`
- Change the file modification date by using a different file as a reference:
`touch -r photo-old.jpg photo-edited.jpg`
- Preview the first 10 lines of an 8-GB CSV data base
`head data.csv`
- Check if two files are identical via check sums
`sha256sum data.csv data2.csv`

bash

bash: the most popular interpreter (Unix shell) of commands in Unix-like operating systems

Use shell scripts to run multiple commands automatically (SH on Mac and Linux, BAT on Windows):

```
#!/bin/sh
```

```
latexdiff thesis.tex thesis-v2.tex >> diff.tex  
latexmk -f -pdf diff.tex
```

This redirects (>>) the output of the comparator (`latexdiff`) to a new file and compiles it with `pdf \LaTeX` (full toolchain).

Modern-day UNIX compatibility

Un*x or ***nix**: Mac OS, Linux, BSD variants, Solaris, HP-UX.

What it means: commonly shared commands and shells (interpreters). One command compatible across multiple OSes!..

...except for Windows.

- On Windows: use [Cygwin](#) to run the most common free software

Learn UNIX commands / GNU coreutils / to harness the full power of HPC or cloud solutions.

Mac and UNIX

macOS is UNIX-certified:

- The core of macOS is POSIX-compliant
- Standard Unix facilities are available from the command line (use *Launchpad* → *Terminal* to access it)
- Apple is using a less powerful and capable version (BSD) of the core utilities
 - Remedied via `brew install coreutils` or getting [MacPorts](#)
 - For text operations, one may need `brew install gnu-sed` and `brew install grep`

Linux and UNIX

Linux: kernel of the family of operating systems that replicate UNIX as closely as possible (and for free!)

Kernel + software = distribution (Ubuntu, **Mint**, Rocky...)

- Used by all supercomputers and clusters
- Favoured by developers, IT professionals, engineers, geeks, tinkerers
 - If your granny needs a stable system with Internet, text editors, media players, messengers and no sluggish updates for her 2009 PC, get her Linux

Linux bootable USBs

Too try Linux, one may use a portable USB stick with **no installation** (to do certain checks or get accustomed). Will be shown *live*!

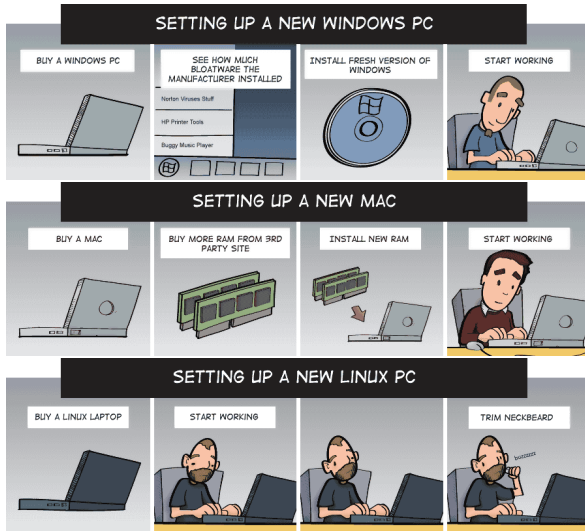
[Story time: faulty laptop of a colleague.]

Intermediate option: install a second SSD, install Windows to the first one and Linux to the second one, choose the system on boot with F12.

Why is Windows workflow so different

- Researchers have traditionally been working on UNIX systems (AT&T, Bell Labs)
 - Designed as a convenient platform for programmers developing software
 - Adopted in academic circles, experts sharing good tools
- Windows is still relying on MS-DOS conventions
 - Compatibility with 1979 86-DOS (QDOS) for 8086 kits
 - Backslash: 1970 TOPS-10 mainframe quirks (reserved /) + 1981 IBM model F keyboard ergonomics + 1983 MS-DOS 2.0
- Pragmatism in 2023: big-data frameworks (servers, CPU clusters, supercomputers) are only UNIX-compatible
 - Even on MS Azure servers, Linux dominates (since 2019)

OS out-of-the-box experience



Credit: Brad Colbow.

OS comparison (updates)

AN UPDATE IS AVAILABLE
FOR YOUR COMPUTER



Credit: [Sticky comics](#).

Your old laptop is fine

In 2016, Windows update started acting like spyware and was trying to force upgrade all the computers. As a consequence, one often thinks, 'should I buy a new PC?'

The answer is **no** in 90% cases.

- Any non-Mac computer: do a fresh Windows reinstall, add Linux (dual boot)
 - Will be shown *live* at the end of this session (or next)
- Mac computer: prepare to pay dearly for an upgrade
 - *Multiple class-action lawsuits* against Apple: violation of the right to repair
 - Company Mac = fine. After 3–4 years, it is often impossible to use the most recent software. After 10 years, it becomes an expensive paper press (unless one is a tinkerer)

Get an SSD

SSD: solid-state drive.

- 11× faster read-write speeds than HDDs
 - Easier to make back-ups
- More reliable, have no moving parts, lower data loss damage
- Makes old computers feel drastically faster, extends their life by 5+ years for a very low price (under €100)

Final advice

Good tools may be expensive, and buying them can be a solid investment.

Choosing hardware / commercial software can be tricky, but bear one thing in mind: **avoid planned obsolescence** like plague.

- Paying for expensive devices that are good only for several years is a bad investment
- Paying for tools that neither the user can customise nor the developer can be made to customise is short-sighted
- Check the reparability index (popular in France), negative user reviews (is there anything that breaks often)

The tomato parable (Max Barry, 2011)



The tomato parable (Max Barry, 2011)

...I decided to grow my own vegetables. A few people I knew were growing their own vegetables, and they kept yakking about how wonderful it was, not depending on manufactured supermarket vegetables, which are evil for some reason, so I thought, 'what the heck'.

The tomato parable



The tomato parable (Max Barry, 2011)

For a while I was intimidated by the idea of growing vegetables. When I reach for a vegetable, I usually just want to eat it. I don't want to be intimately involved with its creation. I worried I would end up spending more time tending to the health of fragile, overly complicated peas than eating them.

The tomato parable (Max Barry, 2011)



The tomato parable (Max Barry, 2011)

Then I saw an ad for genetically modified seeds. These promised to take the hassle out of growing vegetables, which seemed pretty intriguing. The tomatoes would be big and red, and I wouldn't have to do anything. So I got those.

The tomato parable (Max Barry, 2011)



The tomato parable (Max Barry, 2011)

This upset my hippy friends. Especially when I started having problems. My frankenfruit was supposed to be simple, but after a few weeks, the whole garden stopped growing. My cabbages were flaccid. My carrots were anaemic. My spinach wouldn't self-seed.

The tomato parable



The tomato parable (Max Barry, 2011)

It wasn't supposed to self-seed. The genetics company had engineered it not to, so I'd have to buy new seeds each season. But I thought there should be a way around that.

The tomato parable (Max Barry, 2011)



The tomato parable (Max Barry, 2011)

I asked my hippy friends for help. Well! You'd think I asked for a kidney. They kept bringing up the fact that I was using GM seeds. Eventually they all got together and said:

– Max... we can't help you any more. We want to. But you brought these problems on yourself. And the thing is, when you ask for help, you're actually asking us to use our skills and knowledge to prop up a corporatised product that's not just practically inferior to the free alternative you ignored, but actually bad for the world. We just can't do that."

And that was how I taught them to stop asking me for help with Windows.

Windows-specific tweaks

Windows 11 is unsafe

- **Network analysis** shows that there is constant outbound traffic from the user computer to MS servers
- Win11 resets the privacy settings back
- A web cam (!) is needed for a Win11 laptop + forced online MS account – red flag
 - There are Windows installers with offline account creation baked in
- It took an EU intervention to create an 'N' edition of Win10 for the EU market (without the media player)

This puts research and the researcher at risk.

Minimising total harm

One cannot insist on radical changes straight away. However, the pain of Windows users can be somewhat alleviated.

This section contains the tips that I have been accumulating and revising the following usability tips since 2008.

A properly set-up Windows machine can run well for years if the user does not commit risky actions.

Example: I installed a lightweight Windows repack (2.5 Gb) on a laptop with a Windows licence in late 2018 and had no bugs until early 2023. No anti-viruses, no software updates – but it helped me throughout the Ph. D. (and more)!

Complete re-installation

A fresh install can prolong the life of a laptop (especially if you are replacing an HDD with an SSD).

- Back up all the data
- Export explicitly or sync all favourite tabs from the browser
- Check: are there any web sites with log-ins that require a password that one forgot / authentication method that is unavailable?
- Install from a lightweight Windows image with spy / privacy-violating / bloatware tools cut out and the possibility of disabling / enabling updates and Windows Defender (just like Windows 7 was)
 - Read other users' feedback

After re-installation

- Avoid spaces, accents, and non-Latin characters in the user name; use something short and simple
 - Sergueï Fauré = bad; lumi = good
- Disable *System Restore*
 - Almost impossible to make it work; false sense of security and wasted space
- Install the drivers via the update centre once and disable the updates
 - Security updates for Windows = bullet-proof jacket for real bullets, whereas under Linux, the bullets are blanks
 - Windows is notoriously insecure in general – remember it
- Disable Windows Defender (too many false positives)

Miscellaneous changes

- Unhide the file extensions
 - Egregious security flaw: `OpenMe.docx.exe` is shown as `OpenMe.docx` – a known attack vector since the 1990s!
- Show all tray icons to always see which software is running (easier to shut down the unwanted processes)
- Press Shift 5 times and disable Sticky Keys
 - If dual boot is planned, use `regedit` to add a `DWORD 32` registry key `RealTimeIsUniversal=1` to the branch `HKLM\SYSTEM\CurrentControlSet\Control\TimeZoneInformation`
- Use [O&O ShutUp10++](#) to disable most data harvesting by Microsoft

My personal free set-up

Archiver: [7-Zip](#)

Cleaner: [BleachBit](#)

Password manager: [BitWarden](#)

Eye protection: [f.lux](#)*

Start button: [Open Shell](#)

System specs: [Speccy](#)*

Cloud sync: [Dropbox](#)*

Image editor: [GIMP](#)

Screenshots: [GreenShot](#)

* Free but not open-source

Image viewer: [IrfanView](#)*

Office: [LibreOffice](#)

Privacy: [O&O ShutUp 10](#)*

Web browser: [Mozilla Firefox](#)

Text editors: [Notepad++](#)
and [VSCodium](#)

PDF viewer: [SumatraPDF](#)
and (bloated!) [Adobe Reader](#)*

Media: [VLC Mediaplayer](#)

Firefox extensions

Your means of research: finding information quickly.

Your obstacle: adverts and slow page-loading times.

Solution: Firefox + [uBlock origin](#). Unlike Chrome (with the recent 'manifest v3' scandal), Firefox gives its user much more browser configurability.

- Consider installing isolated container support to limit cookie tracking
- If not Firefox, install a browser that respects your privacy and is not trying to change the way the Internet works to maximise corporate profits (Vivaldi, Midori...)

Tip for software search

Do not search for 'free JPG to PDF converter'. Use these prompts:

- **open source** convert JPG to PDF
avoids dubious data-harvesting web sites and shady free utilities
- **Linux** convert JPG to PDF
will suggest many powerful cross-platform tools (like ImageMagick)

This tool will be shown later.

Demonstration time

Deception that software developers use to scare the user or trick the user into buying the product.

Research workflow best practices

Use good tools

- Most top academic software is free and open-source
- Automate: write toolchains, scripts, and routines
- Learn new handy utilities and show them to your advisor
- Abandon the software that hurts your research
 - Even if it is widespread or commercial, it is not necessarily good

Plain text

Plain text: unformatted readable text with no images, media, binary data etc.

- ‘Worse is better’: simplicity, correctness, consistency, completeness.
- Gets more popular: Markdown, minimalist web sites etc.

Examples of non-plain-text files: JPG images, DOCX documents (really a zipped XML + media), PDF files (collections of elements and their page placement).

More on the topic: [‘Unreasonable effectiveness of plain text’](#).

Write in plain text

- Waste 0 seconds on formatting drafts, focus on the ideas, not the interface
 - Easily formatted with `**basic**` mark-up
- Text-processing toolchains rely *exclusively* on plain-text streams
 - Including natural-language processing
 - Cannot lose formatting if there is none
- Never ages (plain text from the 1970s is readable today), no hidden meta-data lost
- Remains plain text when printed, fits into anything

Back-ups

If you store 1 copy of your data, consider it lost.

The second best time is now! Do it today – right after this lecture!

3-2-1 rule: 3 copies of data, at least 2 on different devices, at least one off-site.

Example: a computer (laptop) + external HDD with a backup-up of weekly incremental snapshots (diffs) + the base and 'patches' are copied to the cloud storage (Dropbox).

Alternatives: HPC cluster with their own back-up infrastructure, two HDDs...

Check sums

Check sum: an algorithm to calculate a small digital fingerprint (a short string of bytes) from arbitrary inputs to detect errors or verify integrity.

Hash function: a one-way function that maps arbitrary-size inputs to fixed-size strings.

Requirements: deterministic (non-random), uniform (any string from 000...000 to FFF...FFF should be likely), avalanche effect (small changes of inputs create totally different outputs), collision-resistant (hard to concoct an input that would produce the 'desired' hash)...

Check sum calculation

```
echo "University of Luxembourg" | sha256sum  
# e1b3bc3d...c9b6b807  
echo "University of Luxembourg" | sha256sum  
# 245509c4...62f1fe90
```

Can be done with 7-Zip on Windows, but much easier to use sha256sum from GNU coreutils.

Backing up with rsync + check sums

Back-ups themselves can become corrupted. How does one know if the original or back-up is corrupted?

1. Create check sums of all files + modification times
 - Assuming that everything is fine at creation time
2. After some time, verify if the modification times are identical but check sums are different

To avoid creating dozens of heavy back-ups, use rsync to transfer the differences between source and destination files.

rsync examples

Update the full back-up:

```
rsync -zavh /home/avk/ /media/USBHDD/
```

Copy only DOCX and TEX files, ignoring everything else:

```
rsync -zarv --prune-empty-dirs --include "*" \
--include="*.DOCX" --include="*.docx" \
--include="*.TEX" --include="*.tex" \
--exclude="*" \
"$from" "$to"
```

Replace \$from and \$to with full paths (e.g. /home/avk/ and /media/USBHDD/).

Instant cloud synchronisation

Good to have all intermediate versions upon each `ctrl + S`!

- Self-hosted solutions (Nextcloud, ownCloud), but they require a second working PC
- Google One (15 GB free; 100 GB = €2/month)
- Dropbox (2 GB free; Plus = 2 TB, €10/month, Family = 2 TB + sharing between 6 users, €17/month total)
 - I have been using Dropbox since 2011
- Microsoft OneDrive (5 GB free; 100 GB = €2/month)

Uni.LU offers DropIT, but its implementation was buggy in the past (e. g. it was writing to protected system directories; the developers refused to change this behaviour).

Collaboration tools

Co-authors exist, and their time is precious. Sending `paper-v12-rev3-edit2-GT-AC-AK-AC.docx` via email for is preposterous in 2023.

- Easiest: Google Docs / Sheets
- Moderate: Overleaf (1 co-author in the free version)
- Most advanced: GitHub

Hacker culture and tinkering

*'What software do you use <...> and how do I get it?'
I think the answer is: receive a Commodore 64 for your
tenth birthday and no good games.*

Max Barry (2011).

Academic hacker culture (MIT, 1970s) may spawn and publicise brilliant original problem-solving ideas, or more efficient solutions. It is similar to DIY and maker cultures.

The driving force: enthusiasm and passion (unlike salaries in corporate culture). If corporate software is not working, (1) enthusiasts will not buy it to help you, (2) its developers are unlikely to baby-sit you.

Avoid black-box apps / 'converter' web sites

Around 2015, many web sites, services, and pieces of software were replaced by 'apps' (often mobile-only, with no PC version) or web interfaces.

As much as they save time for certain jobs, over-reliance on them is detrimental.

- Learn the principles and the method
- The black box may disappear from the software centre at any time
 - Install offline software can be used at any time
- Never process confidential data or documents with the tools that you do not own

Own your data

In the increasingly subscription-based world, data ownership is extremely important.

Have physical drives (recall the 3-2-1 back-up principle). Regularly pull the data from the cloud lest you should lose it in an accident (they happen).

Do not blindly trust the ephemeral 'cloud' – distribute your data across multiple media that you own.

'I created a sticker pack!' – 'Nice, in which format? Can you send me the graphics?' – 'No, they are within this application...' ⇒ the creator does not own their creation!

Emergency disassembly skills

You will make a thousand dives with a knife, and will not use it even once. However, on the thousand-and-first dive, it will save your life. (Scuba-diver wisdom.)

- Researchers often consume beverages whilst working
- Their peripheral vision or hand-eye co-ordination might be imperfect
- \Rightarrow Oh no, the laptop has drunk my cup of (*drink name*)
 - Oh no, I wanted to finish something before backing up and ended up not backing up
- It had been a bad idea to place liquids near electronic devices any way

Spilled liquid

1. Hard shutdown: hold the *Power* button for 10 seconds
 - Sometimes, 5 s are enough
 - Make sure that it really switches off, not hibernates; disable hibernation via *Power* button press
2. Remove the back panel and disconnect the battery (or other components if they are wet)
3. Dab the liquid with a paper towel / toilet paper + gently absorb small drops with a cotton bud / blow way with compressed air
 - Sticky liquids: dissolve with isopropyl alcohol + cotton bud
4. Blow some cold or warm (not hot! use your skin as a temperature gauge) and leave to dry for 24 hours
5. Reconnect everything

Spillage aftermath

Keyboard keys might become stuck (if the contacts oxidise).
New keyboards are available.

- No immersion into rice
- No leaving in the sun
- If the battery is soaked, replace it
- No luck turning on? Liquids with ions (caustic / corrosive / acidic) might have shorted it – contact a repair service
 - Do not let them trick you into paying for the replacements that you do not need – ask if the damage components can be re-soldered first

Emergency drill

Practice removing screws and opening the bottom cover.

First step: try finding an online video where a trained professional is doing the disassembly.

- Get a screwdriver (with multiple tips). There might be hidden screws (e. g. under a sticker / rubber foot).
 - Do not strip the screws! Do not apply excessive force!
 - Pick the appropriate tip (Phillips, hexagonal etc.) and size
- Release the plastic hooks of the cover. Pry open with plastic components only (e. g. old credit card)!
 - Do not use metal lest you should deform the softer plastic
 - Push the plastic edges outwards to widen the gap, but check if all screws are removed first
- Never bend the hardware, especially the motherboard!

Know your hardware specs

Windows: use Speccy to export your configuration to a text file (*File* → *Save as Text file*)

Mac: `sysctl -A, ioreg -l, system_profiler`

Linux: `sudo lshw, sudo inxi -Fxz`

Pipe the output of command-line tools to a text file, e. g.

```
sudo lshw > my-pc-specs.txt
```

T_EX, L^AT_EX, and related tools

Today, we start from scratch together

- Finish configuring the fresh installation of Windows 10
- Install T_EXLive on a Windows machine
- Install MacT_EX on a Mac OS machine
- During the installation, we explore Overleaf
 - Your requests: what you want to see in a document – accomplishment of desires
- At the end, we boot into a live Linux distribution (Linux Mint) and install it as the second OS (20 minutes)

L^AT_EX place in the ecosystem

- Standard in the academia
 - Most top journals **require** L^AT_EX and may charge money for conversion from Word
- Separates creation from formatting
 - Write first, focus on the content / ideas, prettify later
- Very stable (core features frozen), but *not rigid*: yet thousands (6500+) of extensions in active development
 - High-quality plots (even 3D!) and support for many external input objects
- Is a Turing-complete programming language \Rightarrow one can achieve **anything**
 - Blazingly fast, runs on any hardware, processes plain text into any output
 - Integrates nicely into script and pipelines with other tools

T_EX similarities

- MS Word 2007+ supports restricted T_EX-like formulæ
- R + RStudio + CRAN \approx T_EX + T_EXStudio + CTAN
- In essence, T_EX really combines symbols into lines, optimises the line flow on a page, and breaks the page – *like a real typesetter in the past*

L^AT_EX strong points

- Beautiful output, very good default parameters
 - Tables, figures, sources easily referenced
- Generates bibliography, lists of tables, indices etc. with one (!) command
 - `\printbibliography`, `\listoftables`, `\printindex`
- Less menial work in general, automation with macros and packages, fast and efficient workflow
- Creates PDFs with cool features (hyperlinks, interactivity...)
- Easy integration with other plain-text-based tools (i. e. anything except MS Word)

L^AT_EX shortcomings

- Harder at first (real programming language!)
 - Error messages are often cryptic
 - That's what you are here for – to start early
- Requires a compiler to see changes (real programming language!)
 - Overleaf solves that problem – compile from anywhere
- Collaborator compatibility: some professors might refuse to learn it
 - Partial conversion solutions exist (will bw shown!)
- Some packages may cause incompatibility errors
 - Like R or any other serious programming language
- Some minuscule changes can be non-intuitive
 - Remedied by a web search and an excellent dedicated StackExchange

Example: unhelpful error message

```
! Missing \endcsname inserted.  
<to be read again> \protect  
1.42 \section
```

```
! Paragraph ended before \command was complete.  
<to be read again>  
                                \par
```

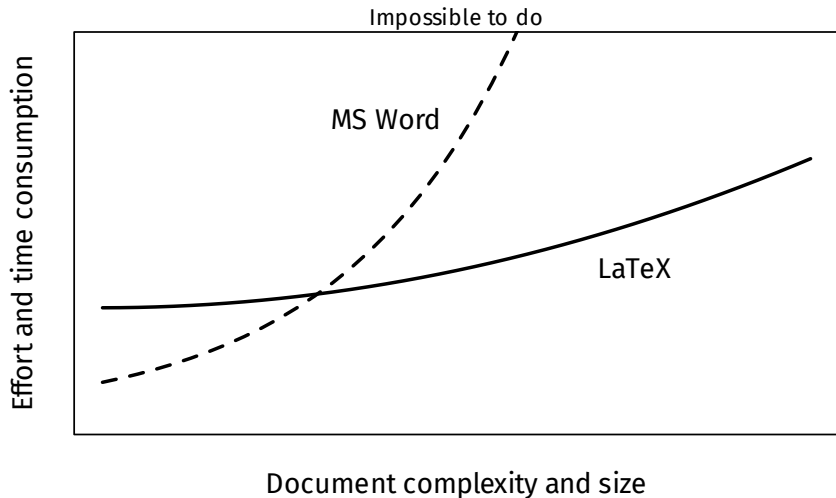
```
1.50
```

Reason: forgotten }.

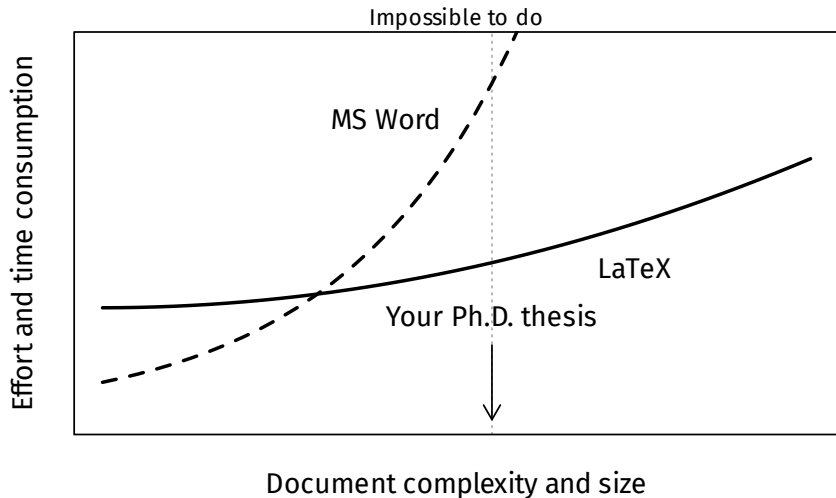
```
! Extra alignment tab has been changed to \cr.  
1.42 \begin{tabular}{|c|c|}
```

Reason: extra column separator & in a table.

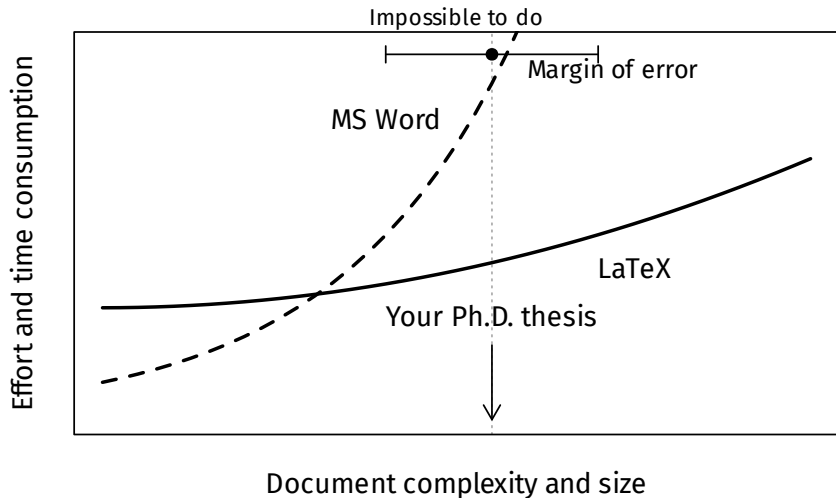
Learning curves



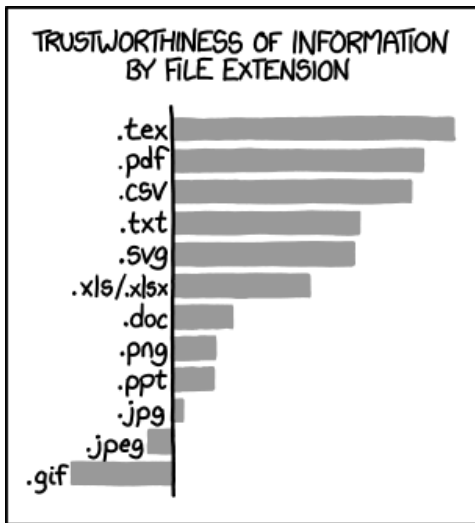
Learning curves



Learning curves



Professional use T_EX



Credit: xkcd. [Technical explanation.](#)

Not using $\text{T}_{\text{E}}\text{X}$ is bad

Scott Aaronson, 'Ten Signs a Claimed Mathematical Breakthrough is Wrong' (2008):

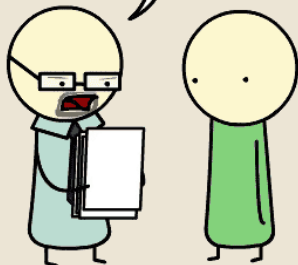
1. The authors don't use $\text{T}_{\text{E}}\text{X}$. This simple test (suggested by Dave Bacon) already catches at least 60% of wrong mathematical breakthroughs. D. Deutsch and L. Grover are among the only known false positives.

Joke comment by John Sidles:

Suppose you are a young researcher who wants to avoid being tripped-up by the 'Ten signs'. Then your biggest return-for-investment—as all posters have agreed—is to learn to write in $\text{T}_{\text{E}}\text{X}$.

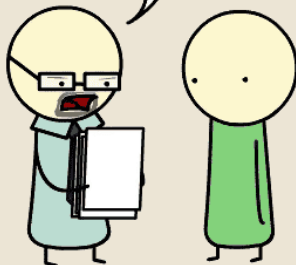
PYTHON

THIS IS PLAGIARISM.
YOU CAN'T JUST "IMPORT ESSAY."



LATEX

YOUR PAPER MAKES NO GODDAMN SENSE,
BUT IT'S THE MOST BEAUTIFUL THING
I HAVE EVER LAID EYES ON.






Credit: somethingofthatilk comics (now defunct).

Why not MS Word



ellewasamistake

@ellewasamistake

gotta respect the longevity of microsoft word. nothing about it works and it's still the standard. want to move an image? go to . edit a pdf? edit your expectations. ignore a spelling mistake? how about  my . that'll be 150\$

5:04 pm · 5 Jul 2021

32.3K Reposts **1,340** Quotes **210.8K** Likes **3,078** Bookmarks

MS products are a privacy nightmare

Microsoft 365 banned in German schools over privacy concerns

German schools cannot legally use Microsoft Office 365 over lack of clarity about how data is collected, shared and used, as well as the potential for unlawful transfer of European citizens' personal data to the US

This article can also be found in the Premium Editorial Download: [CW EMEA: CW EMEA: Protecting the privacy of schoolchildren](#)



By **Sebastian Klovig Skelton**, Senior reporter

Published: 30 Nov 2022 16:39

Federal German data protection authorities have banned the use of [Microsoft Office 365](#) in schools due to privacy concerns around the use of US cloud providers.

The German Data Protection Conference (DSK) – which consists of the German Federal Data Protection Authority and 16 state regulators – said that, given the lack of transparency around how Microsoft collects and processes personal data, as well as the potential for third-party access to it, the use of O365 is not legally compliant with the [General Data Protection Regulation \(GDPR\)](#).

"Microsoft does not fully disclose which processing operations take place in detail. In addition, Microsoft does not fully disclose which processing operations are carried out on behalf of the customer or which are carried out for its own purposes," said a [report by the DSK working group](#) looking at the issue.

Latest News

UK police plan national roll-out of facial-recognition phone app

Ooredoo Qatar claims milestone for private networks in Middle East

Elr Ireland transforms contact centres to enhance customer experience

Typical \LaTeX workflow

\LaTeX is a set of macros extending the capabilities of \TeX .

Typically, the document is created in 2 steps, and the process resembles writing programmes for other compiled languages:

1. The user creates a plain-text file and intuitive commands (called control sequences) to add sections, formulæ, images in the text editor that they like.
2. The user presses the *Compile* button in the editor (or calls `pdflatex myfile.tex` in the command line), and a PDF file is produced.
 - Multiple compilation iterations are necessary to get the page numbers / references right because they are saved to external files and used later.

Open resource review

[Syllabus sources.]

Structure of a T_EX document

Typically, T_EX documents consist of 2 parts

1. **Preamble** – a set of definitions, settings, imported packages, and macros (not exported to the final document)
2. **Body** – text, formulæ, tables, illustrations, commands that make up the document content

Minimal working example

```
% This is myfirstfile.tex
\documentclass[a4paper]{article}
\usepackage[T1]{fontenc}
\newcommand{\mycmd}{cruel}

\begin{document}
Hello, \mycmd\ \textbf{world}!

New paragraph.
\end{document}
```

In the terminal, run:

```
pdflatex myfirstfile.tex
```

This is what myfirstfile.pdf should look like.

Hello, cruel **world**!

New paragraph.

1

Further reading

- Dedicated StackExchange (sorted by popularity)

Thank you for your attention!