Bachelorarbeit

Fachbereich 3 - Mathematik und Informatik

zum Thema

Unleashing the Power of GNU/Emacs and Org-Mode
A Cache of Wisdom for Thesis Writing
(DRAFT version:`v1.0.0-0-g070f143820a`)

---

von Max Musterman
Matrikelnummer 1234567

Studiengang:	Informatik
Semester:	SoSe 2057
Abgabedatum:	25th August 2024
Erstprüfer:	Prof. Dr. Alice
Zweitprüfer:	Dr. Bob

Abstract

The fusion of Doom/Emacs, Org Mode, and LaTeX presents a potent platform for crafting sophisticated thesis papers.[see ML13, p. 5] This article provides a detailed guide on how to seamlessly integrate these tools, facilitating a smooth transition between generating HTML and LaTeX documents, as well as PDFs, all from a single Org Mode file. Furthermore, a Nix shell, accompanied by a shell.nix file, is introduced to ensure efficient management of the development environment.

Zusammenfassung

Die Vereinigung von Doom/Emacs, Org Mode und LaTeX bietet eine leistungsstarke Plattform zur Verfassung eleganter Abschlussarbeiten. Dieser Artikel erläutert detailliert, wie man diese Werkzeuge optimal kombiniert, um nahtlos zwischen der Erstellung von HTML- und LaTeX-Dokumenten sowie PDFs aus einer einzigen Org-Mode-Datei zu wechseln. Zusätzlich wird eine Nix-Shell mit einer begleitenden shell.nix-Datei vorgestellt, um eine effiziente Verwaltung der Entwicklungs- und Arbeitsumgebung zu gewährleisten.

Contents

1. Introduction / Motivation:

Embarking on the journey of using GNU/Emacs with Org-Mode for thesis writing may seem like a daunting task, especially if you're unfamiliar with the intricate workings of GNU/Emacs itself. Some may question the need to delve into Org-Mode when alternatives like LaTeX are readily available and widely embraced. However, dismissing the potential of GNU/Emacs and Org-Mode prematurely might mean overlooking a transformative experience that can redefine your approach to document creation.

1.1. Why bother writing Org-Mode when I can use LaTeX instead?

While LaTeX is a formidable typesetting system, Org-Mode complements it by offering a more versatile and flexible platform for document creation. Org-Mode's simplicity and ease of use, coupled with the ability to seamlessly integrate LaTeX, provide a unique writing experience that combines the best of both worlds.

1.1.1. The possibilities with Org-Mode are endless!

1. Include and evaluate code:

   Integrate code seamlessly within your document, making it a dynamic and interactive platform for technical content.

2. Write LaTeX in a more relaxed surrounding:

   Org-Mode simplifies the LaTeX integration, providing a more user-friendly and intuitive environment for crafting your documents.

1.2. summery

For this tutorial, we will be using a distribution of GNU/Emacs called DOOM Emacs. Additionally, on our GNU/Linux platform, we'll leverage a package manager known as Nix to generate an environment for building both PDF and HTML versions of your thesis. This powerful combination of tools will enhance your writing experience and streamline the process of creating, editing, and compiling your thesis documents.

1.2.1. motivating videos

- Literate Documentation with Emacs and Org Mode

1. *Introduction / Motivation:*

    - Consistent Technical Documents Using Emacs and Org Mode
    - EmacsConf 2023: Authoring and presenting university courses with Emacs and a full libre software …
    - Emacs Tips - How to Give Presentations with Org Mode
    - EmacsConf 2022: Health data journaling and visualization with Org Mode and GNUplot - David O'Toole

2. (Doom) GNU/Emacs setup and configuration

Start by installing Doom Emacs.
`https://youtu.be/37H7bD-G7nE?feature=shared`

2.1. packages `~/.config/doom/package.el`

After installing Doom Emacs - follow up by installing useful packages. If you are not familiar with Doom Emacs's `evil` mode (which is a vim style set of key bindings in Emacs), it might be a good idea to install `evil-tutor` and check out the tutorial for `evil` mode.

The `pdf-tools` (optional for this tutorial) will give you additional functionality for working with PDF documents in Doom Emacs, the most important being the support for viewing the PDF document in a Emacs buffer.

Finlay install `org-special-block-extras` for additional `org-mode` blocks for math and other usefull stuff.

Note

If you want to use the special blocks in you current org mode session - do not forget to enable it with: `M-x org-special-block-extras-mode`

```
grep "^(package" ~/.config/doom/packages.el | sed 's/).*//; s/$/)/'
```

```
(package! evil-tutor)
(package! pdf-tools)
(package! org-special-block-extras)
(package! gnuplot)
(package! gnuplot-mode)
(package! smex)
(package! fireplace)
```

Note

Do not forget to `doom sync` after adding new packages to your `package.el` file and reload Emacs.

2.2. config ~/.config/doom/config.el

To complete the Doom Emacs setup / configuration you only have to addapt your `config.el` file. We will now add the `koma` latex document classes and add support for code highlighting.

```
grep -v '^\s*;' ~/.config/doom/config.el | awk 'NF {print $0; blank=0} !NF && !blank {print ""; blank=1}'
# sed '/^$/N;/^\n$/D;P;D'
```

```

(global-set-key (kbd "C-x <up>") 'windmove-up)
(global-set-key (kbd "C-x <down>") 'windmove-down)
(global-set-key (kbd "C-x <left>") 'windmove-left)
(global-set-key (kbd "C-x <right>") 'windmove-right)

(ido-mode 1)
(ido-everywhere 1)

(setq tramp-auto-save-directory "/tmp")

(advice-add 'highlight-indent-guides-auto-set-faces :override #'ignore)

(setq debug-on-error t)

(remove-hook 'prog-mode-hook #'yas-minor-mode)
(remove-hook 'text-mode-hook #'yas-minor-mode)

(set-face-attribute 'default nil :family "Fira Code" :height 120)

(after! org

  (setq org-babel-min-lines-for-block-output 0)
  (setq org-babel-default-header-args
        (cons '(:session . "none")
              (assq-delete-all :session org-babel-default-header-args)))

  (add-hook #'org-mode-hook #'org-special-block-extras-mode)

  (setq org-ascii-text-width 120)

  (setq org-latex-listings 'minted
        org-latex-packages-alist '(("" "minted"))
        org-latex-pdf-process
        '("pdflatex -shell-escape -interaction nonstopmode -output-directory %o %f"
```

```
36            "pdflatex -shell-escape -interaction nonstopmode -output-directory %o
              ↪  %f"))
37
38    (setq org-latex-minted-options '(("frame" "lines")
39                                     ("linenos" "true")
40                                     ("fontsize" "\\small")
41                                     ("breaklines" "true")
42                                     ("breakanywhere" "true")
43                                     ("tabsize" "4")))
44
45    (setq org-latex-classes
46          '(("article" "\\documentclass[11pt]{article}"
47             ("\\section{%s}" . "\\section*{%s}")
48             ("\\subsection{%s}" . "\\subsection*{%s}")
49             ("\\subsubsection{%s}" . "\\subsubsection*{%s}")
50             ("\\paragraph{%s}" . "\\paragraph*{%s}")
51             ("\\subparagraph{%s}" . "\\subparagraph*{%s}"))))
52
53    (add-to-list 'org-latex-classes
54                 '("koma-book" "\\documentclass{scrbook}"
55                   ("\\chapter{%s}" . "\\chapter*{%s}")
56                   ("\\section{%s}" . "\\section*{%s}")
57                   ("\\subsection{%s}" . "\\subsection*{%s}")
58                   ("\\subsubsection{%s}" . "\\subsubsection*{%s}")
59                   ("\\paragraph{%s}" . "\\paragraph*{%s}")
60                   ("\\subparagraph{%s}" . "\\subparagraph*{%s}")))
61
62    (add-to-list 'org-latex-classes
63                 '("koma-article"
64                   "\\documentclass{scrartcl}"
65                   ("\\section{%s}" . "\\section*{%s}")
66                   ("\\subsection{%s}" . "\\subsection*{%s}")
67                   ("\\subsubsection{%s}" . "\\subsubsection*{%s}")
68                   ("\\paragraph{%s}" . "\\paragraph*{%s}")
69                   ("\\subparagraph{%s}" . "\\subparagraph*{%s}")))
70    )
71
72  (setq doom-theme 'doom-one)
73
74  (setq display-line-numbers-type 'relative)
75
76  (setq org-directory "~/org/")
77
```

Note

Before running `SPC-h-r-r` (M-x doom/reload) - either wait for the `nix env` setup or install `pygments` yourself via `pip3 install pygments`.

3. nix shell environment

Now that Emacs is ready - we can procued by declaring an `shell.nix` environment file, in which we will specify all the packages we will use for compiling the `org-mode` file and all library's and other stuff used in the thesis (e.g. `numpy python haskell`).

We will use `gnuplot` for plotting stuff, `graphics` for generating nice graphs in the `dot` language. In addition we need `texlive` and therefore can use `tikz` to generate nice plots/graphs. Last but not least, we install the `pygments` lib we are using for highlighting the code blocks in addition to the `python` packages used in the source code in the thesis.

If you want to install `haskell` or `R` for your thesis and haven't done so system wide - this might be a good opportunity to add the software.

Install Nix (the package manager)

```
sh <(curl -L https://nixos.org/nix/install) --daemon
```

https://nixos.org/download

```
cat shell.nix
```

Listing 1: `nix-shell` config for generating a build environment

```
{ pkgs ? import <nixpkgs> {} }:

pkgs.mkShell {
  buildInputs = with pkgs; [
    (python312.withPackages(ps: with ps; [
      numpy          # for math
      # latexify-py   # for converting python funcs to latex
      scipy          # for math and physics
      pygments       # for syntax highlighting
      matplotlib     # for plotting
      seaborn        # for plotting
      scikit-learn   # for ml stuff
    ]))
    # texlive.combined.scheme-full
```

```
    graphviz
    gnuplot
  ];

  shellHook = ''
    echo "entering dev environment"
  '';
}
```

Run `nix-shell` to enter the now defined development environment.

After entering our specified environment - you can run `make doom` to start Doom Emacs.

4. start messing around

Since everything should be ready by now - please feel free to mess around with the `thesis.org` file within Emacs.

1. start the nix development environment

```
nix-shell  # schould open a nix shell with all the packages defined in the
↪  local shell.nix config
```

2. open doom Emacs form within the development environment

```
doom run &
```

3. open `thesis.org`

   in doom Emacs, you could either open a file with `M-x dired` or by using `SPC-.`.

4. mess around

   Finlay adapt the `thesis.org` file for your needs or just explore the file from within Emacs.

Export to HTML and PDF

You can export your derivative with the key binding: `C-c C-e`

5. convert python 2 latex

```
import math
import numpy as np
import sys
import scipy
# import latexify
```

Table 5.1.: table of library's with corresponding version

python: sys.version	'3.12.3 (main, Apr 9 2024, 08:09:14) [GCC 13.2.0]'
np.__version__	'1.26.4'
scipy.__version__	'1.13.0'

With the imported library's, we can now convert a `python` function to `latex`

```
def recHelper(x: int, d: int = 3) -> int:
    # forall x in [3, infty)
    if x % 2 == 0:
        return 0
    elif d > math.floor(math.sqrt(x)):
        return 1
    elif x % d == 0:
        return 0
    else:
        return recHelper(x, d + 2)

def isPrime(x: int) -> int:
    if x <= 1:  # forall x in (infty, 1] : 1
        return 0
    elif x == 2:
        return 1
    else:  # forall x in (2, infty)
        return recHelper(x, 3)
```

Listing 2: code block containing prime checking functions

In line 12 is the main definition of the recursive basecase. Line 1 starts by declaring a recursive helper for the main prime checking function.

5. convert python 2 latex

$$\text{isPrime}(x) = \begin{cases} 0, & \text{if } x \leq 1 \\ 1, & \text{if } x = 2 \\ \text{recHelper}\left(x, 3\right), & \text{otherwise} \end{cases}$$

$$\text{recHelper}(x, d) = \begin{cases} 0, & \text{if } x \,\%\, 2 = 0 \\ 1, & \text{if } d > \lfloor \sqrt{x} \rfloor \\ 0, & \text{if } x \,\%\, d = 0 \\ \text{recHelper}\left(x, d + 2\right), & \text{otherwise} \end{cases}$$

Now we can call the `isPrime` function with the parameter `5` and `6`:

- is `5` a prime number? result of `isPrime` function: `True`
- is `6` a prime number? result of `isPrime` function: `False`

6. Function composition

Graph using the `dot` language

```
digraph G {
    node [shape=circle, fontname="Courier", fontsize=16]
    edge [fontname="Courier", fontsize=16]
    A:nw -> A [label="id"]
    A -> B [label="f"]
    B -> C [label="g"]
    A -> C [xlabel="g . f "]
    {rank=same; A B}
}
```

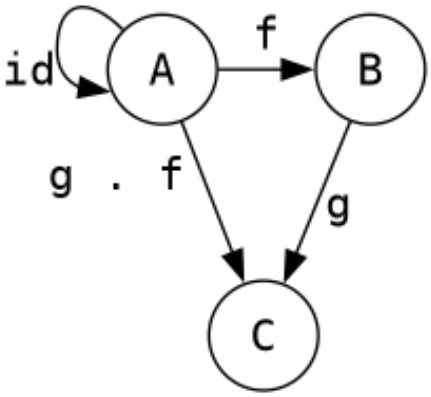


Figure 6.1.: function composition and identity

Graph using latex `tikz picture`

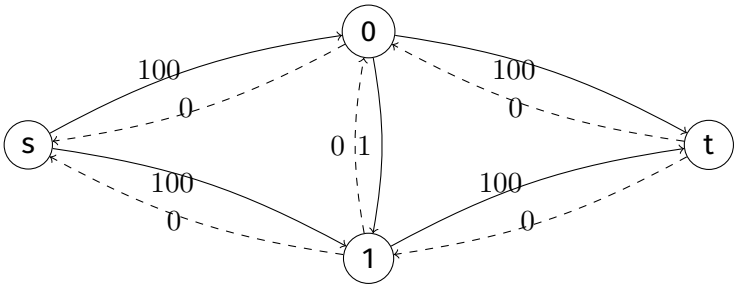


Figure 6.2.: Some random Graph

7. Some random Proof

Let

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \in \mathbb{R}^{3\times 3}.$$

Proof that the following applies to all $\beta \in \mathbb{N}_{>0}$:

$$A^\beta = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \beta & 0 & 1 \end{pmatrix}.$$

'Matrix multiplication as composition | Chapter 4, Essence of linear algebra' [1] The Matrix A can also be interpreted as a linear transformation in $\mathbb{R}^3$.

This interpretation makes it easier to understand the theorem to be proven, and makes it seem almost trivial.

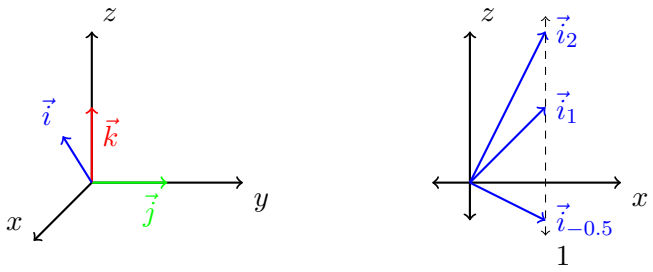


Figure 7.1.: $\mathbb{R}^3$ with linearer transformation A^1

With $\vec{i} = (1, 0, \beta)^T$ and $\vec{j} = (0, 1, 0)^T$ and $\vec{k} = (0, 0, 1)^T$.

This means that only the x coordinate is transformed. No matter how often you exponentiate this transformation, the y and z axes will not change, but the x axis will increasingly approach the y axis.

ℤ

$$\forall \beta \in \mathbb{N}^* \mid \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}^\beta = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \beta & 0 & 1 \end{pmatrix}$$

[1] see https://youtu.be/XkY2DOUCWMU

7. Some random Proof

Proof of important theorem. **Beweis** (durch vollständige Induktion nach β):
Induktionsvoraussetzung ($\star$):

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}^{\beta} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \beta & 0 & 1 \end{pmatrix}$$

Bemerkung 1.
Induktionsbeginn ($\beta = 0$): is also a valid starting point ...

Induktionsbeginn ($\beta = 1$):

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}^{1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

Induktionsschritt ($\beta \rightsquigarrow \beta + 1$):

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}^{\beta+1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ (\beta+1) & 0 & 1 \end{pmatrix} \tag{7.1}$$

$$\Leftrightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}^{\beta} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ (\beta+1) & 0 & 1 \end{pmatrix} \tag{7.2}$$

$$\overset{\star}{\Leftrightarrow} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \beta & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ (\beta+1) & 0 & 1 \end{pmatrix} \tag{7.3}$$

$$\Leftrightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \beta+1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \beta+1 & 0 & 1 \end{pmatrix} \tag{7.4}$$

□

8. plotting functions and their root function / derivation

8.1. python

```python
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Set seed for reproducibility
np.random.seed(42)

# Generate random 5x5 covariance matrix
cov_matrix = np.random.rand(5, 5)
cov_matrix = np.dot(cov_matrix, cov_matrix.T)

# Generate random data with the specified covariance matrix
data = np.random.multivariate_normal(mean=[0, 0, 0, 0, 0], cov=cov_matrix,
↪  size=100)

# Create scatter plot
plt.figure(figsize=(12, 6))

# Scatter plot
plt.subplot(1, 3, 1)
sns.scatterplot(x=data[:, 0], y=data[:, 1])
plt.title('Scatter Plot')

# Box plot
plt.subplot(1, 3, 2)
sns.boxplot(data=data)
plt.title('Box Plot')

# Violin plot
plt.subplot(1, 3, 3)
sns.violinplot(data=data, inner='quartile', palette='muted')
plt.title('Violin Plot')

plt.tight_layout()
# plt.show()
```

```
35 plt.savefig("./img-gen-dirrandom-py-plot.png")
```

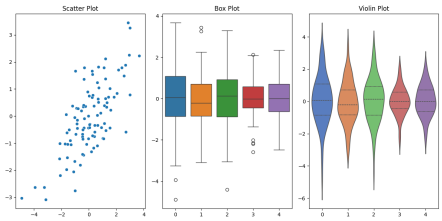
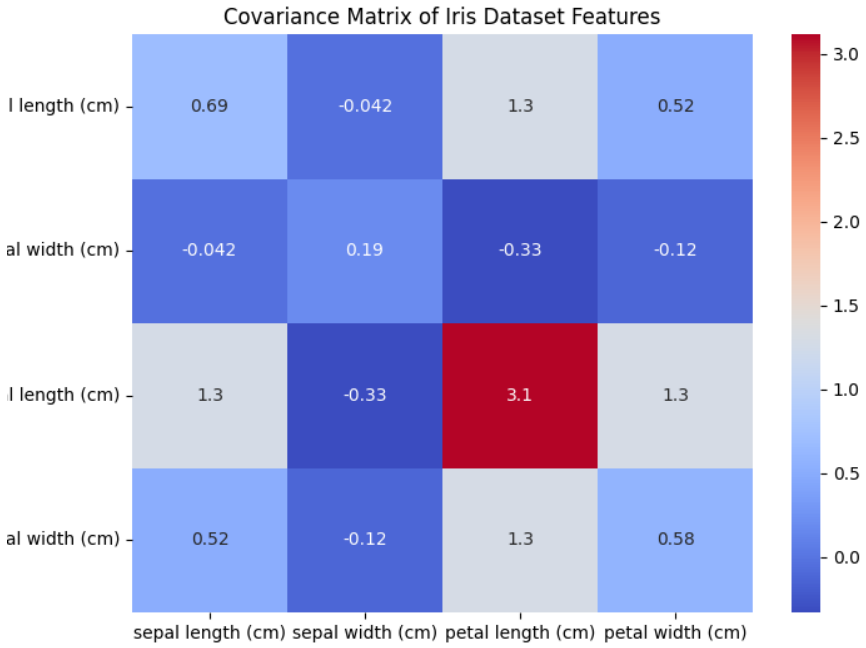
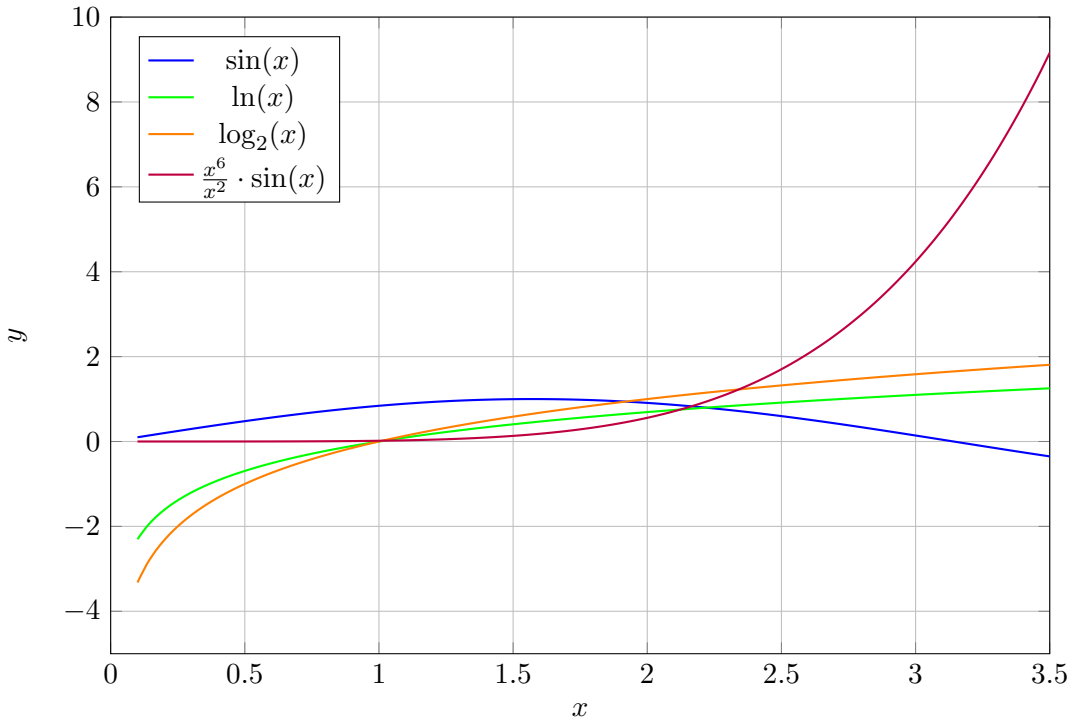


Figure 8.1.: python function plot

```
1  import seaborn as sns
2  import matplotlib.pyplot as plt
3  from sklearn.datasets import load_iris
4  import numpy as np
5
6  # Load the Iris dataset from scikit-learn
7  iris = load_iris()
8  iris_data = iris.data
9  iris_feature_names = iris.feature_names
10
11 # Create a covariance matrix
12 cov_matrix = np.cov(iris_data, rowvar=False)
13
14 # Create a heatmap using Seaborn
15 plt.figure(figsize=(8, 6))
16 sns.heatmap(cov_matrix, annot=True, cmap='coolwarm',
   ↪  xticklabels=iris_feature_names, yticklabels=iris_feature_names)
17 plt.title('Covariance Matrix of Iris Dataset Features')
18 # plt.show()
19 plt.savefig("./img-gen-dir/random-covariance-fig.png")
```


Covariance Matrix of Iris Dataset Features
l length (cm)
al width (cm)
l length (cm)
al width (cm)
0.69
-0.042
1.3
0.52
-0.042
0.19
-0.33
-0.12
1.3
-0.33
3.1
1.3
0.52
-0.12
1.3
0.58
sepal length (cm)
sepal width (cm)
petal length (cm)
petal width (cm)
3.0
2.5
2.0
1.5
1.0
0.5
0.0

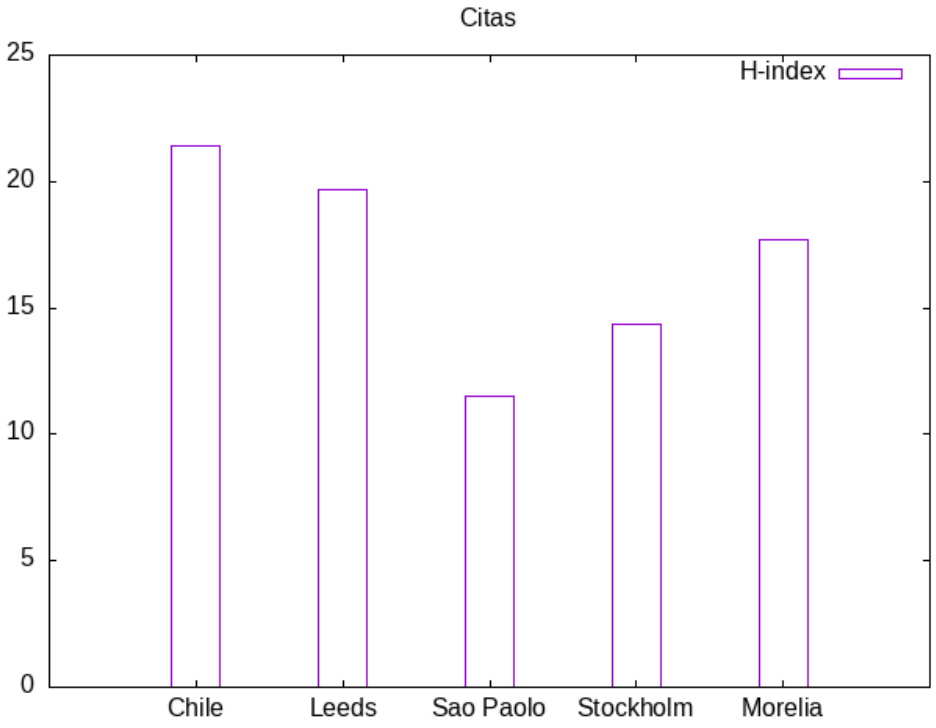
8.2. tikzpicture

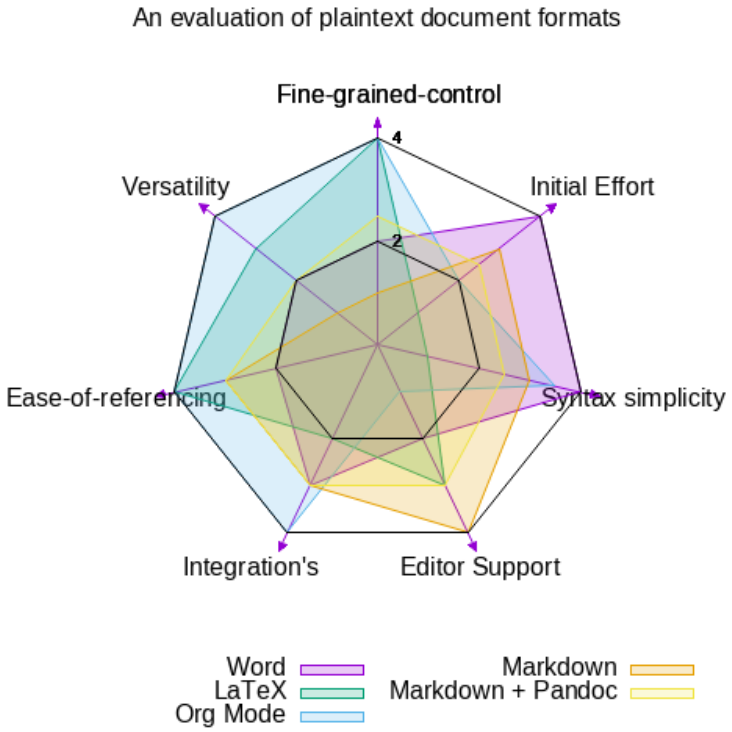


8.3. org plot

see `https://orgmode.org/manual/Org-Plot.html`

Sede	Max cites	H-index
Chile	257.72	21.39
Leeds	165.77	19.68
Sao Paolo	71.00	11.50
Stockholm	134.19	14.33
Morelia	257.56	17.67


Citas
H-index
25
20
15
10
5
0
Chile
Leeds
Sao Paolo
Stockholm
Morelia


An evaluation of plaintext document formats
Fine-grained-control
Initial Effort
Syntax simplicity
Editor Support
Integration's
Ease-of-referencing
Versatility
4
2
Word
LaTeX
Org Mode
Markdown
Markdown + Pandoc

Sede	Max cites	
Chile	257.72	WWWWWWWWWWW
Leeds	165.77	WWWWWWWh
Sao Paolo	71.00	WWW;
Stockholm	134.19	WWWWWW:
Morelia	257.56	WWWWWWWWWWW
Rochefourchat	0.00	
test …	42.0	WW

8.4. gnu plot

see `https://gitlab.com/dto/health-template/-/tree/main?ref_type=heads` by David O'Toole

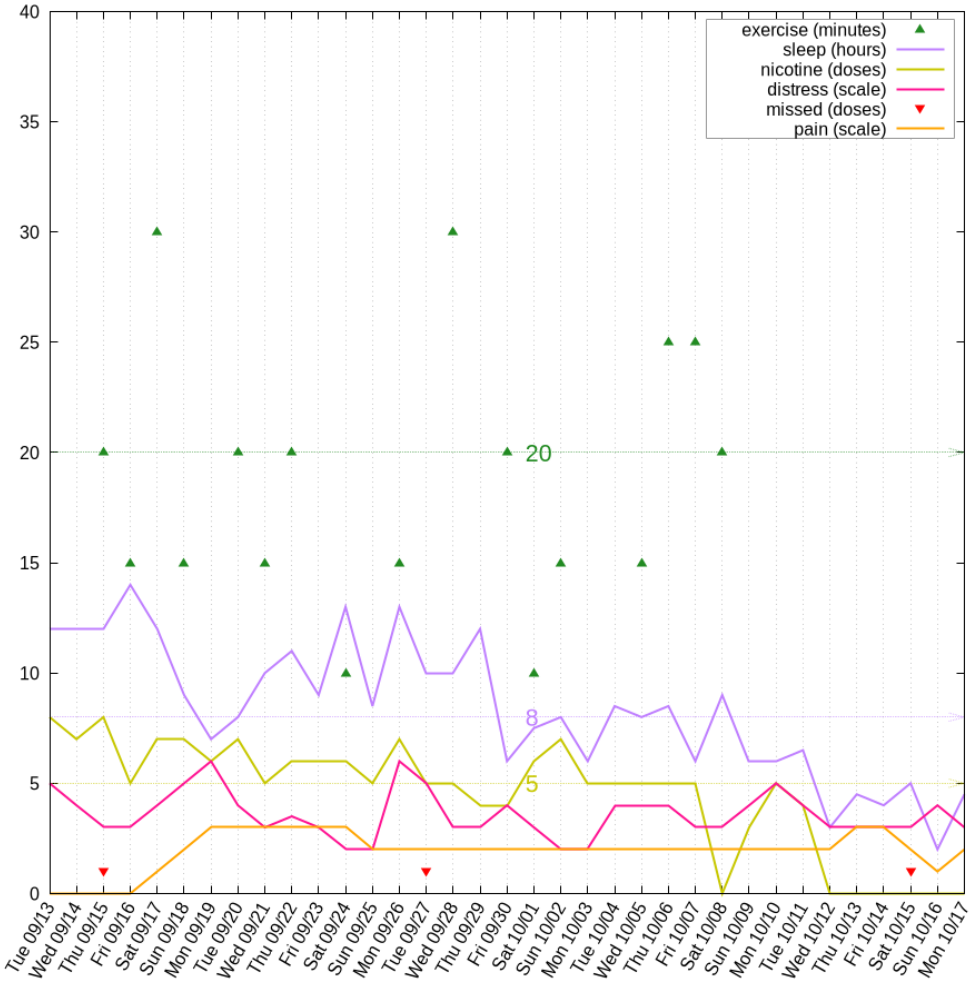


Figure 8.2.: This is the caption

9. logic resolution proof

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

9.1. forward direction ($\rightarrow$)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[A]^3 \qquad [A \to B]^1}{B}[\text{mp}] \qquad [\neg B]^2}{\bot}[\neg E]}{\neg A}[\neg I^3]}{\neg B \to \neg A}[\to I^2]}{(A \to B) \to (\neg B \to \neg A)}[\to I^1]$$

9.2. back direction ($\leftarrow$)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[A]^{5} \qquad \dfrac{[\neg B]^{6} \qquad [\neg B \to \neg A]^{4}}{\neg A}[\mathsf{mp}]}{\bot}[\neg E]}{B}[\mathsf{raa}^{6}]}{A \to B}[\to I^{5}]}{(\neg B \to \neg A) \to (A \to B)}[\to I^{4}]$$

9.3. both directions ($\leftrightarrow$)

$$\dfrac{(A \to B) \to (\neg B \to \neg A) \qquad (\neg B \to \neg A) \to (A \to B)}{\vdash (A \to B) \leftrightarrow (\neg B \to \neg A)}[\leftrightarrow \mathsf{I}]$$

A. some random appendix

maybe add a document or other related but not that important info, listings plots etc. here …

TOC’s

Abkürzungsverzeichnis

ABA	alternierender Büchi-Automat	engl. alternating Büchi automaton
AFA	alternierender endlicher Automat	engl. alternating f inite automaton
BA	Büchi-Automat	engl. Büchi automaton
BNF	Normalform kontextfreier Grammatiken	engl. Backus–N aur f orm
DFA	endlicher Automat	engl. deterministic f inite automaton

Definitions- und Theoremverzeichnis

Bibliography

Book Sources

[ML13] Saunders Mac Lane. *Categories for the working mathematician*. Vol. 5. Springer Science & Business Media, 2013.

List of Listings

List of Tables

List of Figures