LaTeX Experiments

AeAeA

January 23, 2020

1 TeX distributions

1.1 MacTeX

The best for Mac.

```
$ brew cask install mactex
```

1.2 Visual Studio Code LaTeX Workshop Extension

LaTeX Workshop is an extension for Visual Studio Code, aiming to provide core features for LaTeX typesetting with Visual Studio Code.

- `https://github.com/James-Yu/LaTeX-Workshop`
- `https://github.com/James-Yu/LaTeX-Workshop/wiki/Compile`

Build LaTeX file by calling the command `Build LaTeX project` from the Command Palette or from the TeX badge. This command is bound to
`Cmd+Ctrl+b`

You can change VS Code settings by opening Settings tab:
`Cmd+, -> Extensions -> LaTeX`
or, alternatively, by directly editing settings.json file:
`~/Library/Application\ Support/Code/User/settings.json`
Recommended settings for LaTeX Workshop:

```
{
    "latex-workshop.view.pdf.viewer": "tab",
    "latex-workshop.latex.outDir": "%DIR%/texout",
    "latex-workshop.latex.autoBuild.run": "never",
    "latex-workshop.latex.autoClean.run": "onBuilt"
}
```

1.3 MiKTeX

Not for Mac. Old MiKTeX installation:
`/usr/local/bin/`
`/Applications/MiKTeX\ Console.app/`

1.4 TinyTeX

TinyTeX - a lightweight, cross-platform, portable, and easy-to-maintain LaTeX distribution based on TeX Live.

Currently TinyTeX works best for R users. Installing and maintaining TinyTeX is easy for R users, since the R package tinytex has provided wrapper functions.

For other (non-R) users:

- See TinyTeX docs here.
- In the directory
  `~/Library/TinyTeX/texmf-dist/tex/latex/`
  you can find all LaTeX packages installed for TinyTeX.
- If you compile a LaTeX document and run into an error message like this:
  ```
  ! LaTeX Error: File ‘times.sty’ not found.
  ```
  It basically indicates a missing LaTeX package.

  Use the command `tlmgr search` to find the name of the missing package:
  ```
  $ tlmgr search --global --file "/times.sty"
  psnfss: texmf-dist/tex/latex/psnfss/times.sty
  ```
  In this case, the missing package is `psnfss`, and we can install a package via `tlmgr install`, e.g.,
  ```
  $ tlmgr install psnfss
  ```
  If you still see error messages that you don’t understand, you may need to update everything:
  ```
  $ tlmgr update --self --all
  $ tlmgr path add
  $ fmtutil-sys --all
  ```
- To uninstall TinyTeX use command line:
  ```
  $ tlmgr path remove
  $ rm -r "~/Library/TinyTeX"
  ```

2 Epigraph

> *In doing what we ought we deserve no praise, because it is our duty.*
>
> — Saint Augustine

2.1 Online docs

TeXdoc Online is TeX and LaTeX documentation lookup system.

2.2 verbatim

```
Text enclosed inside
    \begin{verbatim} ... \end {verbatim}
environment                      is printed directly
and all \LaTeX{} commands are ignored.
```

```
Text␣enclosed␣inside␣\begin{verbatim*}␣environment
␣␣␣␣is␣printed␣directly
and␣all␣\LaTeX{}␣commands␣are␣ignored,
and␣white␣spaces␣are␣emphasized␣with␣a␣special␣symbol.
```

Use `\verb+<inline verbatim text>+` like this:
The `\ldots` command produces …

2.3 listings: Source code printing

- listings package documentation
- `https://www.overleaf.com/learn/latex/Code_listing`

2.3.1 minimal setup

Example of using the `\begin{lstlisting}[language=Python]` environment from the `\usepackage{listings}` package to highlight Python code:

```python
import numpy as np

def incmatrix(genl1,genl2):
    m = len(genl1)
    n = len(genl2)
    M = None #to become the incidence matrix
    VT = np.zeros((n*m,1), int)  #dummy variable

    #compute the bitwise xor matrix
    M1 = bitxormatrix(genl1)
    M2 = np.triu(bitxormatrix(genl2),1)

    for i in range(m-1):
        for j in range(i+1, m):
            [r,c] = np.where(M2 == M1[i,j])
            for k in range(len(r)):
                VT[(i)*n + r[k]] = 1;
                VT[(i)*n + c[k]] = 1;
                VT[(j)*n + r[k]] = 1;
                VT[(j)*n + c[k]] = 1;

                if M is None:
                    M = np.copy(VT)
                else:
                    M = np.concatenate((M, VT), 1)

                VT = np.zeros((n*m,1), int)

    return M
```

2.3.2 with code styles and colours

You need `\usepackage{xcolor}` package for the code colouring.
Just like in floats (tables and figures), captions can be added to a listing for a more clear presentation. This caption can be later used in the list of Listings `\lstlistoflistings`.

```python
import numpy as np

def incmatrix(genl1,genl2):
    m = len(genl1)
    n = len(genl2)
    M = None #to become the incidence matrix
    VT = np.zeros((n*m,1), int)  #dummy variable

    s = "codepurple"

    #compute the bitwise xor matrix
    M1 = bitxormatrix(genl1)
    M2 = np.triu(bitxormatrix(genl2),1)

    for i in range(m-1):
        for j in range(i+1, m):
            [r,c] = np.where(M2 == M1[i,j])
            for k in range(len(r)):
                VT[(i)*n + r[k]] = 1;
                VT[(i)*n + c[k]] = 1;
                VT[(j)*n + r[k]] = 1;
                VT[(j)*n + c[k]] = 1;

                if M is None:
                    M = np.copy(VT)
                else:
                    M = np.concatenate((M, VT), 1)

                VT = np.zeros((n*m,1), int)

    return M
```

Listing 1: Python example

Listings