

Programs

Contents

1. Python	1
2. LaTeX	2
3. Zotero	4
4. Fresh installation of Ubuntu/Fedora	6
5. Ubuntu and Fedora proxy	6
6. R	8

1. Python

Download Python

1.1. Deploy oTree application to Heroku server To upload the oTree application in a cloud server like Heroku is an important step to conduct the online experiments. Here, I am showing the steps that you need to fulfill to upload your application:

1. Make sure your oTree application is working correctly locally.
2. Create a Heroku account if you don't already have one.
3. Install Heroku in your Mac: If you are a Mac user, you can install Heroku application from the **homebrew** as follows:

```
brew tap heroku/brew && brew install heroku
```

4. Upload an app in Heroku: To upload your oTree application to Heroku, first create an *application name* inside your Heroku account as below:

New... > Create New App

5. Enter to the folder of your Pycharm which contains the application you are interested to upload to Heroku server:

```
cd appname
```

6. Proxy settings if necessary Sometime institutional networks (e.g.; universit, office, business) require some additional proxy settings to upload an application to Heroku. I am sharing the example of my institution.

```
export http_proxy="http://proxy.noc.kochi-tech.ac.jp:3128"
export https_proxy="http://proxy.noc.kochi-tech.ac.jp:3128"
```

7. Log in to Heroku Now it is time to push your application to your Heroku account. To do so, you need to go through the following steps.

```
heroku login
git init
git add .
git commit -am "command message"
heroku git:remote -a appname
git push heroku master
heroku config:set OTREE_PRODUCTION=1
```

Your oTree application should now be deployed to your Heroku server. You can access it by going to the URL provided by Heroku.

Note: Sometime due to the inconsistency in version of Python in your computer and reported Python version in runtime.txt might be different. In that case, you might not be able to rung the oTree program in computer. To get rid from it, you can remove the runtime.txt from your Pycharm.

2. LaTeX

For writing an academic manuscript by using Latex, I prefer to use GNU Emacs environment. One of the main reason to use Emacs is the freedom to customize the codes. It is compatible with Linux (Ubuntu, Fedora), MacOS and Windows. So I can create same environment of writing my manuscript across the board.

2.1 For Ubuntu/Fedora operating system

2.1.1 Install texlive-full What is texlive-full

The TeX Live software distribution offers a complete TeX system. It encompasses programs for typesetting, previewing and printing of TeX documents in many different languages, and a large collection of TeX macros and font libraries. The distribution also includes extensive general documentation about TeX, as well as the documentation accompanying the included software packages.

Install texlive-full Using apt:

We can install `texlive-full` using `apt` (for Ubuntu) / `dnf` (for Fedora) by running the following command:

```
sudo apt -y install texlive-full
```

```
sudo dnf -y install texlive-full
```

2.1.2 Install Jabref What is jabref

JabRef is a graphical Java application for editing BibTeX (.bib) databases. JabRef lets you organize your entries into overlapping logical groups, and with a single click limit your view to a single group or an intersection or union of several groups. You can customize the entry information shown in the main window, and sort by any of the standard BibTeX fields. JabRef can auto-generate BibTeX keys for your entries. JabRef also lets you easily link to PDF or web sources for your reference entries. JabRef can import from and export to several formats, and you can customize export filters. JabRef can be run as a command line application to convert from any import format to any export format.

Install jabref using apt:

We can install `jabref` using `apt` (for Ubuntu) / `dnf` (for Fedora) by running the following command:

```
sudo apt -y install jabref
```

```
sudo dnf -y install jabref
```

2.1.3 Install Latexmk What is latexmk

Latexmk runs LaTeX the correct number of times to resolve cross references, etc; it also runs auxiliary programs (bibtex, makeindex if necessary, and dvips and/or a previewer as requested). It has a number of other useful capabilities, for example to start a previewer and then run LaTeX whenever the source files are updated, so that the previewer gives an up-to-date view of the document.

Install latexmk using apt:

We can install `latexmk` using `apt` (for Ubuntu) / `dnf` (for Fedora) by running the following command:

```
sudo apt -y install latexmk
```

```
sudo dnf -y install latexmk
```

2.1.4 Install Gnu Emacs What is emacs

Emacs is an amazing text editor available on all major operating systems from BSD to Linux to macOS and Windows. One of the biggest reasons people love Emacs is that it comes loaded with a wide range of commands designed for the simplest and the most complex tasks.

Install emacs using apt:

We can install the `emacs-gtk` package using `apt` (for Ubuntu) / `dnf` (for Fedora) by running the following command:

```
sudo apt -y install emacs-gtk
```

```
sudo dnf -y install emacs-gtk
```

- To operational use of Emacs, we need to confirm that the following three files/folder (1) `.emacs`, (2) `.latexmkrc` and (3) `.emacs.d` are existing in the Home directory of the Ubuntu/Fedora operating system.
- Customize `.emacs`
- Customize `.emacs.d`
- Customize `.latexmkrc`

By using your text editor in Ubuntu/Fedora, create `.latexmkrc` text file. Then you can write the following codes inside this text file:

```
$dvi_previewer = 'start xdvi -watchfile 1.5';
$ps_previewer  = 'start gv --watch';
$pdf_previewer = 'start evince';

# Generate pdf using pdflatex (-pdf)
$pdf_mode = 1;

# Use bibtex if a .bib file exists
$bibtex_use = 1;

# Define command to compile with pdfsync support and nonstopmode
$pdflatex = 'pdflatex -synctex=1 --interaction=nonstopmode -file-line-error';

# Continous compiling
$preview_continuous_mode=1;
```

- Load necessary packages listed on the `.emacs` file from ELPA/melpa/GNU or from other available repositories.
- Check your `.emacs.d` folder to confirm the existing packages are fully operational or not.
- Regularly update your emacs packages.

2.1.5 LaTeX environment inside Emacs

3. Zotero

Download Zotero

Zotero with Quarto and RMarkdown

1. Create an **account on Zotero** website. It will be used later on to sync the database of bibliographic metadata.
2. Download and install the two plugins we'll need:
 - a. **ZotFile** (organizes the database of PDFs)
 - b. **Better BibTeX** (exports the library to `.bib`, we'll use it later with `rmarkdown`).

The plugins for Zotero are `.xpi` archives. To install the plugins open Zotero and click Tools → Add-ons. A separate window for Add-ons manager will pop-up.

- c. Select Install Add-on from **File** option. Finally navigate to the `.xpi` file and install.
4. **Zotero preferences.** Let's walk through Zotero Preferences. To edit them click Edit → Preferences. A window with several tabs pops up.

5. **Zotero sync.** Here we need to specify the account details to sync our database. It is important to uncheck the option of full-text sync otherwise the 300MB storage will quickly get filled. We'll have the solution for full text a bit later.
6. **Advanced.** Here we are most interested in the sub-tab Files and Folders. This is the most important step to separate the storage of metadata and files.
7. Set your **Base directory** inside your Dropbox

```
/Users/Owner/Dropbox/zotero_library
```

8. **Better BibTeX.** This tab appears after we install the Better BibTeX extension. The extension is needed to export the whole bibliographic library (or some of its parts) as a plain .bib text file. This step is needed to use Zotero in RStudio while writing academic papers with Quarto and RMarkdown.
9. The most important option here is to define the rules for creating citation keys. There are almost infinite number of ways one can define these keys (check the manual). My personal choice is a key consists of the **first author's name_publication year**.

```
[auth:capitalize] '_' [year]
```

10. **ZotFile Preferences.** To open the setup window click Tools -> ZotFile Preferences. Again, the window has several tabs.
11. *General.* Here we define two paths.

The first is the default location of the files downloaded by your browser. This option tells ZotFile where to look for the PDFs to process when you import a paper from the publisher's website.

```
/Users/Owner/Downloads
```

The second path leads to the local directory created for the full-text PDFs, the one that I named zotero-library and which is synced with an external cloud solution of our choice.

12. For the second path, set your **Custom Location** inside your Dropbox.

```
/Users/Owner/Dropbox/zotero_library/AUTH
```

13. **Renaming Rules.** Here it's important to make sure that ZotFile is responsible for renaming. Then we define how to rename the PDFs based on the bibliographic metadata available. Again, here we have many many options. My choice is `{%a_}{%y_}{%t}` which yields file names like.
14. If you parse the metadata manually from a PDF, make sure to rename the file using ZotFile. For that right-click the metadata record Manage Attachments -> Rename Attachments. This action explicitly tells to use ZotFile for renaming and will move the renamed PDF to a proper sub-directory.
15. Use Zotero library in RStudio with rmarkdown **Better BibTeX** offers an easy way to export bibliographic records from Zotero as plain .bib text and keep the file updated once the records are changed. Just right-click on the collection in Zotero and choose Export Collection.

4. Fresh installation of Ubuntu/Fedora

4.1 Use fdisk command to delete a hard disk partition: `fdisk` also known as format disk is a dialog-driven command in Linux used for creating and manipulating disk partition table. It is used for the view, create, delete, change, resize, copy and move partitions on a hard drive using the dialog-driven interface. `fdisk` allows you to create a maximum of four primary partitions and the number of logical partition depends on the size of the hard disk you are using.

1. View All Disk Partitions:

The first thing to do before doing any thing with the disks and partition is to view basic details about all available partition in the system.

```
sudo fdisk -l
```

2. View Partition on a Specific Disk:

Below command is used to view all disk partitions on device `/dev/sda`.

```
sudo fdisk -l /dev/sda
```

3. Delete a Hard Disk Partition:

To delete a partition for the hard disk and free up space occupied by that partition for example `/dev/sdb`. Go to the command menu using following:

```
sudo fdisk /dev/sda
```

- And then type `d` to go to the delete partition menu. It will prompt the partition number you want to delete (type the number).
- Run `w` command to write the changes and reboot the system.

5. Ubuntu and Fedora proxy

```
@client:~# sudo nano /etc/profile.d/proxy.sh
```

5.1 Proxy for Ubuntu Create new (set proxy settings to the environment variables)

```
MY_PROXY_URL="proxy.noc.kochi-tech.ac.jp:3128"
```

```
HTTP_PROXY=$MY_PROXY_URL
HTTPS_PROXY=$MY_PROXY_URL
FTP_PROXY=$MY_PROXY_URL
http_proxy=$MY_PROXY_URL
https_proxy=$MY_PROXY_URL
ftp_proxy=$MY_PROXY_URL
```

```
export HTTP_PROXY HTTPS_PROXY FTP_PROXY http_proxy https_proxy ftp_proxy
```

```
@client:~# sudo source /etc/profile.d/proxy.sh
```

It is possible to set proxy settings for each application, not System wide for apt

```
@client:~# sudo nano /etc/apt/apt.conf
```

Create new

```
Acquire::http::proxy "http://proxy.noc.kochi-tech.ac.jp:3128";  
Acquire::https::proxy "http://proxy.noc.kochi-tech.ac.jp:3128";  
Acquire::ftp::proxy "ftp://proxy.noc.kochi-tech.ac.jp:3128";
```

For curl

```
@client ~]# sudo nano ~/.curlrc
```

Create new

```
proxy=proxy.noc.kochi-tech.ac.jp:3128
```

For wget

```
@client:~# sudo nano /etc/wgetrc
```

Add to the end

```
http_proxy = proxy.noc.kochi-tech.ac.jp:3128  
https_proxy = proxy.noc.kochi-tech.ac.jp:3128  
ftp_proxy = proxy.noc.kochi-tech.ac.jp:3128
```

```
@client:~# sudo nano /etc/profile.d/proxy.sh
```

5.2 Proxy for Fedora Create new (set proxy settings to the environment variables for system wide)

```
MY_PROXY_URL="proxy.noc.kochi-tech.ac.jp:3128"
```

```
HTTP_PROXY=$MY_PROXY_URL  
HTTPS_PROXY=$MY_PROXY_URL  
FTP_PROXY=$MY_PROXY_URL  
http_proxy=$MY_PROXY_URL  
https_proxy=$MY_PROXY_URL  
ftp_proxy=$MY_PROXY_URL
```

```
export HTTP_PROXY HTTPS_PROXY FTP_PROXY http_proxy https_proxy ftp_proxy
```

```
@client:~# sudo source /etc/profile.d/proxy.sh
```

Or it is possible to set proxy settings for each application, not system wide for dnf

```
@client:~# sudo nano /etc/dnf/dnf.conf
```

Create new

```
Acquire::http::proxy "http://proxy.noc.kochi-tech.ac.jp:3128";  
Acquire::https::proxy "http://proxy.noc.kochi-tech.ac.jp:3128";  
Acquire::ftp::proxy "ftp://proxy.noc.kochi-tech.ac.jp:3128";
```

For curl

```
@client ~]# sudo nano ~/.curlrc
```

Create new

```
proxy=proxy.noc.kochi-tech.ac.jp:3128
```

For wget

```
@client:~# sudo nano /etc/wgetrc
```

Add to the end

```
http_proxy = proxy.noc.kochi-tech.ac.jp:3128  
https_proxy = proxy.noc.kochi-tech.ac.jp:3128  
ftp_proxy = proxy.noc.kochi-tech.ac.jp:3128
```

6. R

6.1 Basic Information About the pacman Package

- Install pacman Package in R. In order to use the functions that are contained in the pacman package, we have to install and load the package to R:

```
install.packages("pacman")    # Install pacman package  
library("pacman")             # Load pacman package
```

- After running the previous lines of code, we can use the functions provided by the pacman package.

Example 1: Install & Load Multiple R Packages Using p_load Function In this example, I'll illustrate how to apply the p_load function to load multiple add-on packages in one line of code.

Have a look at the following R syntax:

```
p_load(ggplot2, dplyr, stringr) # Install & load packages
```

After executing the previous code, we have loaded the three packages ggplot2, dplyr, and stringr.

Note that the p_load function also checks whether a package is installed already. If the package is not installed yet, it is installed automatically by the pacman package.

Example 2: Unload Multiple R Packages Using p_unload Function Another useful function of the pacman package is the p_unload function. The p_unload function can be used to detach one or multiple loaded packages from within R.

Let's apply the p_unload command to unload the three packages ggplot2, dplyr, and stringr that we have loaded before:

```
p_unload(ggplot2, dplyr, stringr) # Unload packages
```

The following packages have been unloaded:ggplot2, dplyr, stringr The previous output of the RStudio console tells us that our three packages were detached.

Example 3: Update Outdated R Packages Using p_update Function In this example, I'll show how to use the p_update function of the pacman package to check for outdated packages AND to update all of these packages.

- First, let's check which of my packages are outdated:

```
p_update(update = FALSE) # Check for outdated packages
```

- Second, the pacman package provides an easy solution to how we can update all outdated packages with only one line of R code.
- Make sure that you have some time before running the following R code. This may take some time, depending on the number of packages you need to update:

```
p_update() # Update all packages
```

- Quick Reference Table

p_load	#Loads and Install Packages
p_install	#Install Packages from CRAN
p_load_gh	#Loads and Install GitHub Packages
p_install_gh	#Install Packages from GitHub
p_temp	#Install a Package Temporarily
p_unload	#Unloads Packages from the Search Path
p_update	#Update Out-of-Date Packages