

Development Environment Setting and Project Management

MinDong Sung, M.D.

DHLab, Yonsei University College of Medicine

2021/07/02

Disclosure

- Main language - python, R, go
- Using Visual Studio code, R-Studio.
- Using both Mac and Windows
- I'm not a developer. I'm a doctor.
- 제가 하는 방식은 완벽하지 않으며, 반드시 이렇게 해야한다는 것은 아닙니다.

Development Environment

Introduction

<https://www.notion.so/mindong/DHLab-4c974b1303634ac9962191fe67ecf81a>

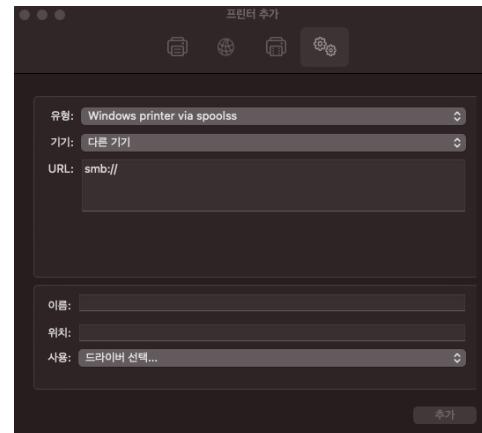
Wireless Lan

YUHS-external

- 그룹웨어 ID / Password 이용해서 접속 가능

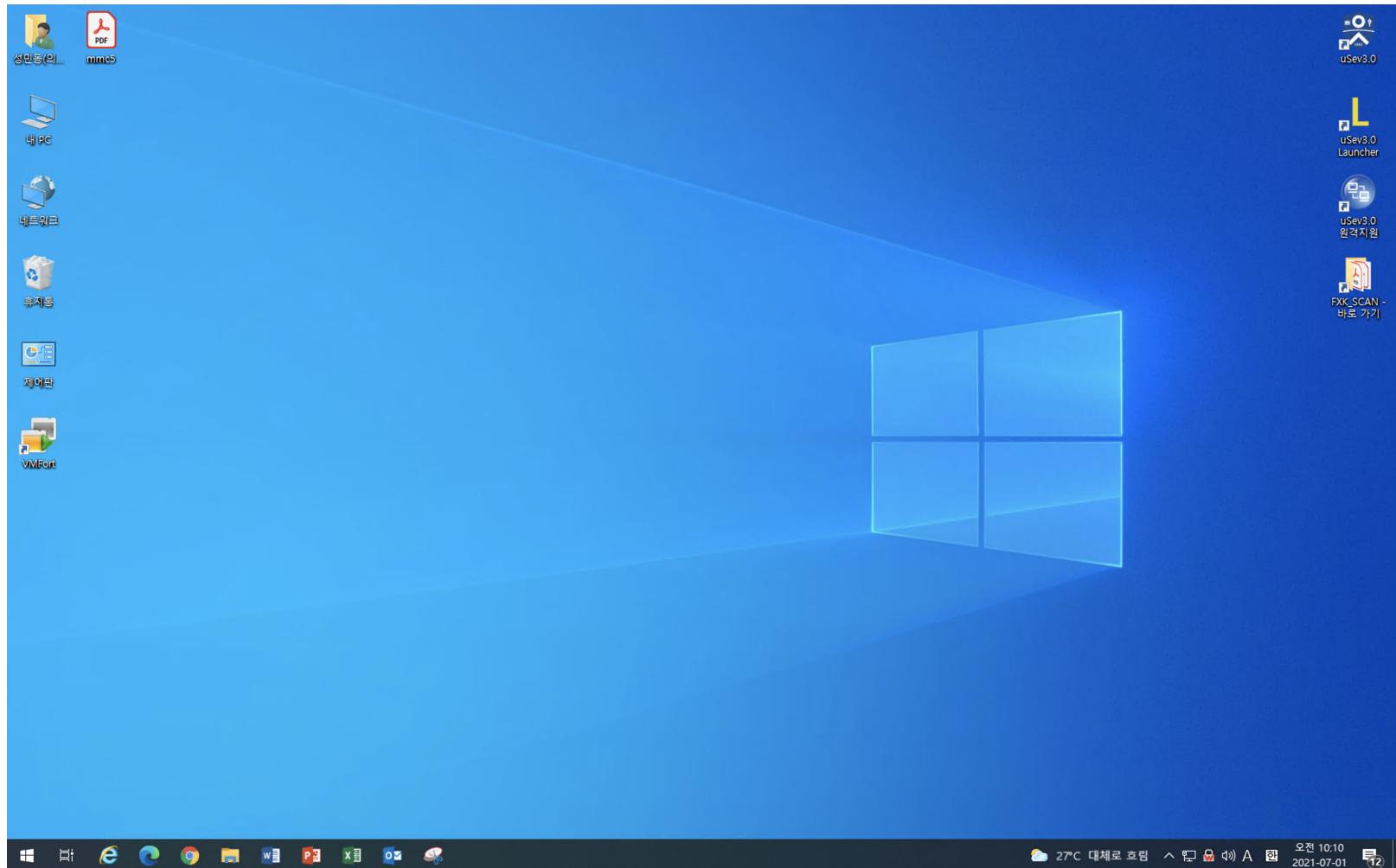
Connect to Print(for Mac)

1. 원도우 프린트 공유가 필요하고 공유 위치(ip)와 공유 프린트 이름이 필요하다.
 - IP: 10.19.12.58
 - Printer Name: Kyocera FS-4025DN KX
2. smb://[IP address of Printer]로 접속하여 인증을 한다.
3. 프린트 추가 - 고급
 - 유형: windows printer via spoolss
 - 기기: 다른기기
 - URL: smb://10.19.12.58/Kyocera%20FS-4025DN%20KX
 - 빈칸에 **%20**을 입력한다.
 - 사용은 소프트웨어가 있으면 좋겠지만 없으면 일반 postscript 프린트로 설정



Connect to PC (feat. Microsoft Remote Desktop)

Connect to PC (feat. Microsoft Remote Desktop)



Development Environment

Terminal

Shell

tmux

Integrated Development Environment

Terminal

Terminal (Windows)

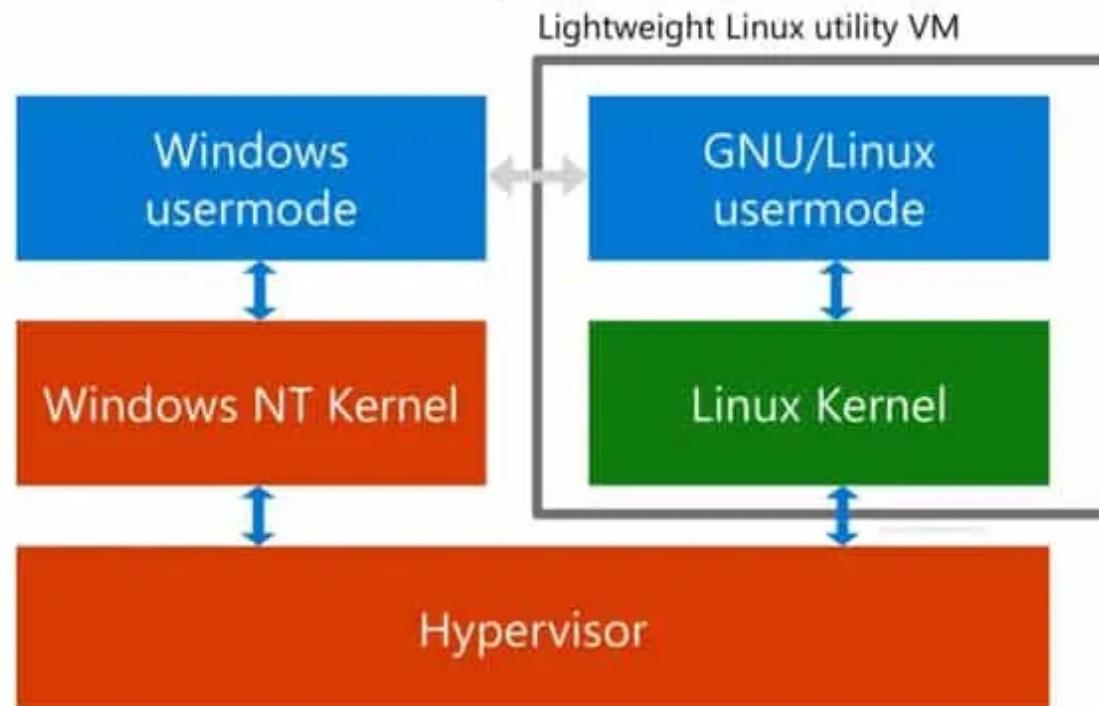
- putty? Terminal!
- 이전에는 open ssh가 지원이 되지 않아서 putty가 필요했으나 최근에는 필요가 없음
- 예시)

```
ssh mindong@103.22.220.153 -p10022
```

WLS2 (Windows Subsystem Linux 2)

- 윈도우 내에서 linux를 사용할 수 있습니다!!
- 윈도우 터미널 환경이 불편하다면 WSL2가 대안이 될 수 있습니다.

WSL 2 architecture overview



- Install protocol: https://github.com/mdsung/wsl2_install_tutorial

iTerm2 (Mac) (vs. Terminal.app)

Shell

- bash
- zsh
- (fish)

cf. <https://medium.com/harrythegreat/oh-my-zsh-iterm2%EB%A1%9C-%ED%84%B0%EB%AF%B8%EB%84%90%EC%9D%84-%EB%8D%94-%EA%B0%95%EB%A0%A5%ED%95%98%EA%B2%8C-a105f2c01bec>

cf. .bashrc/.zshrc

- 필요한 PATH나 alias 등록 가능

```
# Add Visual Studio Code (code)
export PATH="$PATH:/Applications/Visual Studio Code.app/Contents/Resources"

alias phd="cd work/phd_thesis_2021; source env/bin/activate"
alias mimic="cd work/mimiciv"
alias wd='cd ~/work'

## server
alias 149ssh='ssh mindong@103.22.220.149 -p10022'
alias 153ssh='ssh mindong@103.22.220.153 -p10022'
alias 67ssh='ssh mindong@61.252.54.67 -p10022'
alias 76ssh='ssh mindong@61.252.54.76 -p10022'
alias 79ssh='ssh mindong@61.252.54.79 -p10022'
```

Commonly using shell command

- head / tail / cat / less - reader
- ls / ls - lash
- cd - change directory
 - cd ~ - home directory
 - cd / - root directory
- mkdir - make directory
- rm / rm -r - remove
- cp / mv - copy / move
- scp - ssh copy
- vim -> code
- !! - lastest command
 - sudo !!

ssh (without password)

Client

- ssh key를 만든다. - ~/ .ssh 폴더에 id_rsa(private key), id_rsa.pub(public key)가 생성된다.

```
ssh-keygen -t rsa
```

- client에서 만든 ssh public key를 서버로 복사를 한다.

```
scp -P [port] id_rsa.pub [id]@[host]:~/
```

Server

- 서버에 복사된 id_rsa.pub을 .ssh 폴더에 authorized_keys로 옮긴다.

```
mkdir ~/.ssh  
chmod 700 ~/.ssh  
cat ~/id_rsa.pub >> ~/.ssh/authorized_keys
```

tmux

- Terminal 창으로 여러개 띄울 수 있다. (한꺼번에 보는 것도 가능)
 - pane
 - window
- Terminal이 닫혀도 session이 종료되지 않는다.

```
Press ? For Help
... (up a dir)
/home/don/dev/c/stroustrup/
part1-introductory-material/
chapter2-a-tour-of-cpp-the-
chapter3-a-tour-of-cpp-abs-
chapter4-a-tour-of-cpp-con-
bin/
  4-2-strings.cpp
  4-3-stream-lo.cpp
  4-4-containers-1-vector>
  4-4-containers-1-vector>
  4-4-containers-1-map.cpp
  4-4-containers-4-unorder-
  4-5-algorithms-1-use-of->
  4-5-algorithms-3-stream-
  4-5-algorithms-4-predica-
  4-5-algorithms-6-contain-
  4-5-algorithms.cpp
  Makefile
chapter5-a-tour-of-cpp-con-
bin/
  5-2-resource-management>
  5-3-concurrency-1-passin-
  5-3-concurrency-2-passin-
  5-3-concurrency-3-return-
  5-3-concurrency-4-sharin-
  Makefile
README.md
int main() {
    Vec<Entry> phone_book;
    input_book(phone_book);
    checked(phone_book);
    return 0;
}

istream & operator>>(istream& is, Entry& e);

void input_book(Vec<Entry>& book) {
    for (Entry e; cin >> e);
    book.push_back(e);
}

// corresponding input operator
istream & operator>>(istream& is, Entry& e) {
    <ter4-a-tour-of-cpp-containers-and-algorithms-4-4-containers-1-vector-2-range-checking.cpp 1,1 Top>
    || make[1]: Entering directory '/home/don/dev/c/stroustrup/part1-introductory-material/chapter4-a-tour-of-cpp-con-
    || bin/4-4-1-vector.out 4-4-containers-1-vector.cpp
    || g++ -std=c++11 -o bin/4-4-1-vector.out 4-4-containers-1-vector.cpp
    || /bin/4-4-1-vector.out > bin/book-entries.txt
    || ("John Marwood Cleese",123456)
    || ("Michael Edward Palin",98754)
    || ("David Hume",123456)
    || ("Kurt Vonnegut",123456)
    || ("Bertrand Arthur William Russell",345678)
[QuickFix List] :Dispatch make 4-4-1-vector (tmux@10466) 6,1 Top
/home/don/dev/c/stroustrup  tmux@10466: ~$ vimc 3:bash 3:python 4:bash
execute pathogen#infect()
let napleader = ""
let mapleader = "\\""
" .vimrc -- {{{ $MAKE -C $(CH3) $0
nnoremap <leader>ev :vsplit $MY_VIMRC
nnoremap <leader>sv :source $MY_VIMRC
" }})
" Mappings to disable non-vim keymaps {{{ $MAKE -C $(CH3) $0
cyrillic {
    inoremap jk <esc>
    inoremap <esc> <nop>
    inoremap <Up> <nop>
    inoremap <Down> <nop>
    inoremap <Left> <nop>
    inoremap <Right> <nop>
}
-
" 4-4-1-vector {{{ $MAKE -C $(CH4) $0
-
" 4-4-1-2-range-check {{{ $MAKE -C $(CH4) $0
-
" 4-4-1-list {{{ $MAKE -C $(CH4) $0
-./.vimrc  1,1  All <ertal/Makefile 66,0-1  58%
top - 14:54:07 up 3:10,  8 users,  load average: 0.37, 0.39, 0
Tasks: 225 total, 1 running, 224 sleeping, 0 stopped, 0 z
KCPUs(s): 1.4 us, 0.5 sy, 0.0 nl, 98.0 ld, 0.0 wa, 0.0 hi,
KLB Men: 7589068 total, 3664668 free, 1787728 used, 213667
KLB Swap: 7794684 total, 7794684 free, 0 used. 504645
PID USER PR NI VIRT RES SHR S %CPU %MEM
16828 don 20 0 935860 71328 51628 S 2.0 0.9
2783 don 20 0 2733312 429656 197896 S 1.7 5.7
2251 don 20 0 1520188 162544 72396 S 1.0 2.1
1110 root 20 0 447080 114744 91152 S 0.7 1.5
2950 don 20 0 2699072 462364 188596 S 0.7 6.1
7057 root 20 0 0 0 0 S 0.3 0.0
10631 don 20 0 41416 6608 2364 S 0.3 0.0
1449 root 20 0 452424 45624 28944 S 0.3 0.6
2321 don 20 0 936548 40984 40776 S 0.3 0.7
3482 don 20 0 685476 47284 35120 S 0.3 0.6
9770 root 20 0 0 0 0 S 0.3 0.0
10343 don 20 0 43084 3992 3356 R 0.3 0.1
10550 root 20 0 0 0 0 S 0.3 0.0
10343 don 20 0 185460 6104 3952 S 0.3 0.0
2 root 20 0 0 0 0 S 0.0 0.0
4 root 20 0 -20 0 0 0 S 0.0 0.0
6 root 20 0 0 0 0 0 S 0.0 0.0
8 root 20 0 0 0 0 0 S 0.0 0.0
9 root 20 0 0 0 0 0 S 0.0 0.0
10 root 20 0 0 0 0 0 S 0.0 0.0
11 root 20 0 0 0 0 0 S 0.0 0.0
12 root 20 0 0 0 0 0 S 0.0 0.0
13 root 20 0 0 0 0 0 S 0.0 0.0
tmux@10466: ~$
```

<p>[tmux sessions]</p> <ul style="list-style-type: none"> _ new sessions <pre>tmux tmux new tmux new-session tmux new -s sessionname</pre> <hr/> <ul style="list-style-type: none"> _ attach sessions <pre>tmux a tmux att tmux attach tmux attach-session tmux a -t sessionname</pre>	<p>linuxacademy.local</p> <ul style="list-style-type: none"> _ remove sessions <pre>tmux kill-ses tmux kill-session -t sessionname</pre> <hr/> <ul style="list-style-type: none"> _ key bindings <table border="0"> <tr> <td>Ctrl + B \$</td> <td>rename session</td> </tr> <tr> <td>Ctrl + B D</td> <td>detach session</td> </tr> <tr> <td>Ctrl + B)</td> <td>next session</td> </tr> <tr> <td>Ctrl + B (</td> <td>previous session</td> </tr> </table>	Ctrl + B \$	rename session	Ctrl + B D	detach session	Ctrl + B)	next session	Ctrl + B (previous session	<p>[tmux windows]</p> <p>linuxacademy.local</p> <ul style="list-style-type: none"> _ windows are like tabs in a browser. Windows exist in sessions and occupy the space of a session screen. <hr/> <ul style="list-style-type: none"> _ key bindings <table border="0"> <tr> <td>Ctrl + B C</td> <td>create window</td> </tr> <tr> <td>Ctrl + B N</td> <td>move to next window</td> </tr> <tr> <td>Ctrl + B P</td> <td>move to previous window</td> </tr> <tr> <td>Ctrl + B L</td> <td>move to window last used</td> </tr> </table>	Ctrl + B C	create window	Ctrl + B N	move to next window	Ctrl + B P	move to previous window	Ctrl + B L	move to window last used	<table border="0"> <tr> <td>Ctrl + B 0 .. 9</td> <td>select window by number</td> </tr> <tr> <td>Ctrl + B ‘</td> <td>select window by name</td> </tr> <tr> <td>Ctrl + B .</td> <td>change window number</td> </tr> <tr> <td>Ctrl + B ,</td> <td>rename window</td> </tr> <tr> <td>Ctrl + B F</td> <td>search windows</td> </tr> <tr> <td>Ctrl + B &</td> <td>kill window</td> </tr> </table>	Ctrl + B 0 .. 9	select window by number	Ctrl + B ‘	select window by name	Ctrl + B .	change window number	Ctrl + B ,	rename window	Ctrl + B F	search windows	Ctrl + B &	kill window																								
Ctrl + B \$	rename session																																																						
Ctrl + B D	detach session																																																						
Ctrl + B)	next session																																																						
Ctrl + B (previous session																																																						
Ctrl + B C	create window																																																						
Ctrl + B N	move to next window																																																						
Ctrl + B P	move to previous window																																																						
Ctrl + B L	move to window last used																																																						
Ctrl + B 0 .. 9	select window by number																																																						
Ctrl + B ‘	select window by name																																																						
Ctrl + B .	change window number																																																						
Ctrl + B ,	rename window																																																						
Ctrl + B F	search windows																																																						
Ctrl + B &	kill window																																																						
<p>[tmux panes]</p> <ul style="list-style-type: none"> _ panes are sections of windows that have been split into different screens – just like the panes of a real window! <hr/> <ul style="list-style-type: none"> _ key bindings <table border="0"> <tr> <td>Ctrl + B %</td> <td>vertical split</td> </tr> <tr> <td>Ctrl + B “</td> <td>horizontal split</td> </tr> <tr> <td>Ctrl + B →</td> <td>move to pane to the right</td> </tr> <tr> <td>Ctrl + B ←</td> <td>move to pane to the left</td> </tr> </table>	Ctrl + B %	vertical split	Ctrl + B “	horizontal split	Ctrl + B →	move to pane to the right	Ctrl + B ←	move to pane to the left	<p>linuxacademy.local</p> <ul style="list-style-type: none"> _ key bindings <table border="0"> <tr> <td>Ctrl + B ↑</td> <td>move up to pane</td> </tr> <tr> <td>Ctrl + B ↓</td> <td>move down to pane</td> </tr> <tr> <td>Ctrl + B 0</td> <td>go to next pane</td> </tr> <tr> <td>Ctrl + B ;</td> <td>go to last active pane</td> </tr> <tr> <td>Ctrl + B }</td> <td>move pane right</td> </tr> <tr> <td>Ctrl + B {</td> <td>move pane left</td> </tr> <tr> <td>Ctrl + B !</td> <td>convert pane to window</td> </tr> <tr> <td>Ctrl + B X</td> <td>kill pane</td> </tr> </table>	Ctrl + B ↑	move up to pane	Ctrl + B ↓	move down to pane	Ctrl + B 0	go to next pane	Ctrl + B ;	go to last active pane	Ctrl + B }	move pane right	Ctrl + B {	move pane left	Ctrl + B !	convert pane to window	Ctrl + B X	kill pane	<p>[tmux copy mode]</p> <p>linuxacademy.com</p> <ul style="list-style-type: none"> _ key bindings <table border="0"> <tr> <td>Ctrl + B [</td> <td>enter copy mode</td> </tr> <tr> <td>Ctrl + B]</td> <td>paste from buffer</td> </tr> </table> <hr/> <ul style="list-style-type: none"> _ copy mode commands <table border="0"> <tr> <td>space</td> <td>start selection</td> </tr> <tr> <td>enter</td> <td>copy selection</td> </tr> <tr> <td>Esc</td> <td>clear selection</td> </tr> <tr> <td>g</td> <td>go to top</td> </tr> </table>	Ctrl + B [enter copy mode	Ctrl + B]	paste from buffer	space	start selection	enter	copy selection	Esc	clear selection	g	go to top	<table border="0"> <tr> <td>G</td> <td>go to bottom</td> </tr> <tr> <td>h</td> <td>move cursor left</td> </tr> <tr> <td>j</td> <td>move cursor down</td> </tr> <tr> <td>k</td> <td>move cursor up</td> </tr> <tr> <td>l</td> <td>move cursor right</td> </tr> <tr> <td>/</td> <td>search</td> </tr> <tr> <td>#</td> <td>list paste buffers</td> </tr> <tr> <td>q</td> <td>quit</td> </tr> </table>	G	go to bottom	h	move cursor left	j	move cursor down	k	move cursor up	l	move cursor right	/	search	#	list paste buffers	q	quit
Ctrl + B %	vertical split																																																						
Ctrl + B “	horizontal split																																																						
Ctrl + B →	move to pane to the right																																																						
Ctrl + B ←	move to pane to the left																																																						
Ctrl + B ↑	move up to pane																																																						
Ctrl + B ↓	move down to pane																																																						
Ctrl + B 0	go to next pane																																																						
Ctrl + B ;	go to last active pane																																																						
Ctrl + B }	move pane right																																																						
Ctrl + B {	move pane left																																																						
Ctrl + B !	convert pane to window																																																						
Ctrl + B X	kill pane																																																						
Ctrl + B [enter copy mode																																																						
Ctrl + B]	paste from buffer																																																						
space	start selection																																																						
enter	copy selection																																																						
Esc	clear selection																																																						
g	go to top																																																						
G	go to bottom																																																						
h	move cursor left																																																						
j	move cursor down																																																						
k	move cursor up																																																						
l	move cursor right																																																						
/	search																																																						
#	list paste buffers																																																						
q	quit																																																						

IDE

- Jupyter Vs. VS Code (Vs. Pycharm)
- 대부분 Jupyter에서 하는 것을 VS Code에서 할 수 있다.
 - 원격 개발이 가능
 - 필요시 jupyter notebook도 실행 가능
- 파일 관리가 쉽다.
- Web browser 형태가 아닌 프로그램 형태로 프로그램을 띄우기 쉽다.
- 여러가지 extension이 있어 개발할 때 편하다.
- Debugging
- Test Automation

VS code

- python
- python jupyter
- python integration shell(like jupyter) - #%%
- 원격 접속 개발
 - ssh key 이용 시 password 없이 접속 가능

```
Host 103.22.220.153
  HostName 103.22.220.153
  User mindong
  Port 10022
  IdentityFile ~/.ssh/id_rsa
```

- 여러가지 단축키
 - alt + 위/아래
 - alt + shift 위/아래
 - ctrl/cmd + d - 같은 것 선택
 - ctrl/cmd + / - 주석 처리

Project Management

(feat. Reproducible Research)

Project structure

- code/
- data/
 - raw/
 - processed/
- explore/
- (optional) proposal/
- (optional) manuscript/
- (optional) presentation/
- .gitignore
- Makefile
- README.md

c.f. https://github.com/mdsung/2020_KISA_blockchain_survey

[R] Project

- 프로젝트별로 데이터 및 파일을 관리할 수 있어 여러 프로젝트를 동시에 진행할 때 organize하게 진행할 수 있다.
- library here
 - make your path absolute and unified based on the project folder
- library renv
 - save your dependencies
 - `renv::init()`
 - `renv::snapshot()`
 - `renv::restore()`

[Python] Project

- Make virtual environment to set your project
 - `python -m venv env`
 - `env/bin/activate`
 - `deactivate`
- Save your dependencies in the txt file
 - `pip freeze > requirements.txt`

[R] R notebook or R markdown

- Rscript에서 작업을 많이 하지만, notebook이나 document를 이용하면, 기록하고 공유할 때 편하다.
- output: 여러가지 형태로 변환이 가능하다.
 - html_notebook
 - html_document
 - github_document
 - word_document
 - pdf_document

[Python] Jupyter notebook

- jupyter nbconvert
- 공유시 html 파일로 변환 후 공유하면 좀 더 편하다

git / github

- version 관리
- code 백업
- (협업?)
- project별로 repository를 구성하면 webpage에서 분석 결과를 쉽게 확인할 수 있다.

c.f. https://github.com/mdsung/2021_scRNA_PMBC_replicate

code 관리

- code는 하나의 일을 하는 것이 좋다. (input -> output)

- naming은 항상 중요하다.

e.g. create_preprocess_dataset.R e.g. create_figure1.R

data/raw 관리

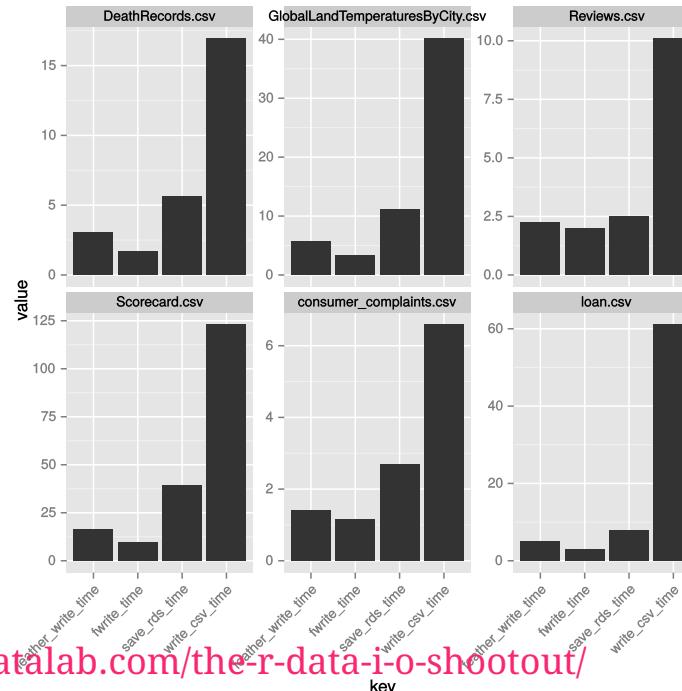
- raw 파일은 절대로 무순일이 있더라도 건드리지 않는다.
- 필요하다면 복사해서 진행을 하고,
- 코드로 항상 기록을 남기자
- 가능하다면 raw file은 저장소에서 가져오는 방식을 이용한다.

e.g.

https://github.com/mdsung/2021_scRNA_PMBC_replicate/blob/master/code/cngb_download.sh

data/processed 관리

- Intermediate, result files
- Use feather file
 - feather format can be read in both R and Python.
 - c.f RDS format is the way faster than feather.



<https://blog.dominodatalab.com/the-r-data-i-o-shootout/>

.gitignore

- github에 모든 파일을 올릴 수는 없다.
- data 특히 raw는 올리지 않을 것이 좋다.
- 민감한 정보를 분석할 때에는 code만 올리는 것을 고려
- 예시)

```
env/  
data/raw/  
*/__pycache__/  
.DS_Store
```

e.g. https://github.com/mdsung/2021_ICU_Sepsis_Prospective/blob/main/.gitignore

github 기본 이용법

- git init/git clone
- git add
- git commit -m "first commit"
- git push
- git pull

Git 기본 환경 설정하기

```
git config --global user.email "mdskylover@gmail.com"  
git config --global user.name "mdsung"  
git config --global core.editor "code --wait" # git 에서 사용하는 기본 editor
```

Others...

- **Makefile** - save your work process (from download raw data to write manuscript)

e.g. https://github.com/mdsung/2021_ICU_Sepsis_Prospective/blob/main/Makefile

e.g. https://github.com/mdsung/2020_KISA_blockchain_survey/blob/main/Makefile

Lets Start the Project