

Wiki, TidyVerse, Incdtools

February 17, 2023

Outline

- 1 Wiki
- 2 Base R
- 3 dplyr/tidyr
- 4 Shell w/Incdtools
- 5 Bash test driven development

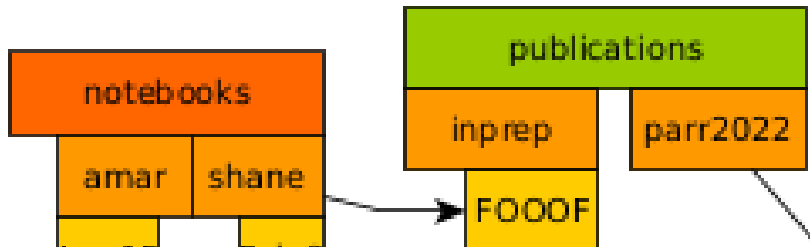
in two parts: demo, structure discussion

Dokuwiki

- self-hosted PHP w/o DB: fits adjacent to wordpress on pitt's free EWI
 - available anywhere ... even Hawaii at no cost to lab
- lindy effect (software life expediency age)
- decent built-in search
- ugly but giant props to Shane for organizing and prettifying
- xml-rpc (emacs, vim, etc) plugins
- zim-deskop-wiki cross over (emacs packages)

Nomenclature

- (forward) links on a page. hierarchical "namespaces"
(:notebooks:shane)
 - `[[[:namespace:page]]]` or `[[[:page]]]`
- Orphan vs wanted, backlinks



Motivation: Zotero tags

"backlinks" a la tags (eg. as used by zotero)

The screenshot shows the Zotero interface. On the left, the 'Group Libraries' pane is open, showing a tree view with folders: 'eyetracking_ref' (selected), 'lncd_ref', and '7T renewal'. Below the tree, there are search filters: 'AltImages', 'multipleLiqhts', 'checkout!', 'review/compare' (highlighted in a blue button), and 'multipleCameras'. On the right, the 'Title' pane displays a list of items:

- ▶ [document icon] [tag icon] A Review and Analysis of Eye-Gaze
- ▶ [document icon] [tag icon] General Theory of Remote Gaze Est
- ▶ [document icon] [tag icon] Eye gaze tracking techniques for int
- ▶ [document icon] [tag icon] A free geometry model-indepe
- ▶ [document icon] [tag icon] A Survey on Eye-Gaze Tracking Tecl
- ▶ [document icon] [tag icon] Development of an Eye Tracking-

The screenshot shows the Zotero interface with a different library structure. On the left, the 'Group Libraries' pane shows 'eyetracking_ref' and 'lncd_ref' (selected). Below the tree, there are search filters: 'Finn SamElliott' (highlighted in a blue button). On the right, the 'Title' pane displays a list of items:

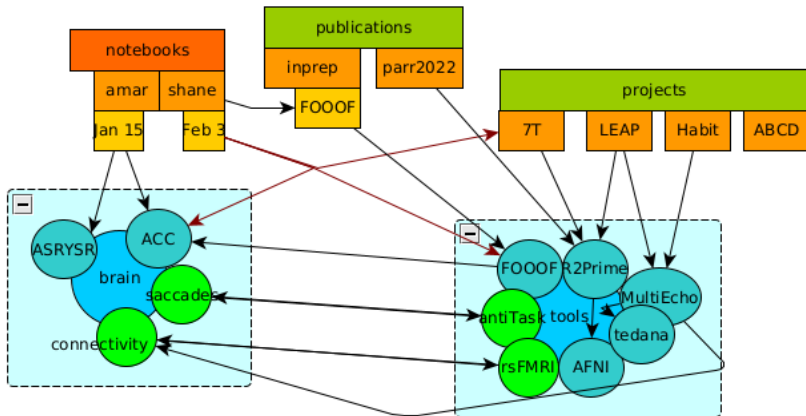
- ▶ [document icon] [tag icon] Compulsivity is linked to reduced adolescent develop...
- ▶ [document icon] [tag icon] Endogenous fluctuations in the dopaminergic midbra...
- ▶ [document icon] [tag icon] Searching for Rewards Like a Child Means Less Gene...

Motivation: Will's Zim

example zim wiki link map



LNCD



All about the links

- backlinks => treat links like #tags to build network nodes
 - what pubs include PET project data
 - who has used the FOOOF tool
 - where does ACC fit in with lab research?

Even empty links

- "wanted" empty pages no link
- still collects backlinks
- wanted list: `:admin:website:wiki#linkswithout content`

Links without content

wanted (opposite of orphan)

#	ID	Links
1	notebooks:orma	3 : Show backlinks
2	admin:deprecated	2 : Show backlinks
3	brain:dlpfc	2 : Show backlinks
4	deprecated	2 : Show backlinks
5	notebooks:ashley	2 : Show backlinks
6	tools:matlab	2 : Show backlinks

Even poorly named links

- "admin>rename plugin" to re-organization

p?id=admin:website:wiki&do=admin&page=move_main

to the simple form you can manage complex restructuring of your wiki using

Move current page or namespace

Move page

Move namespace

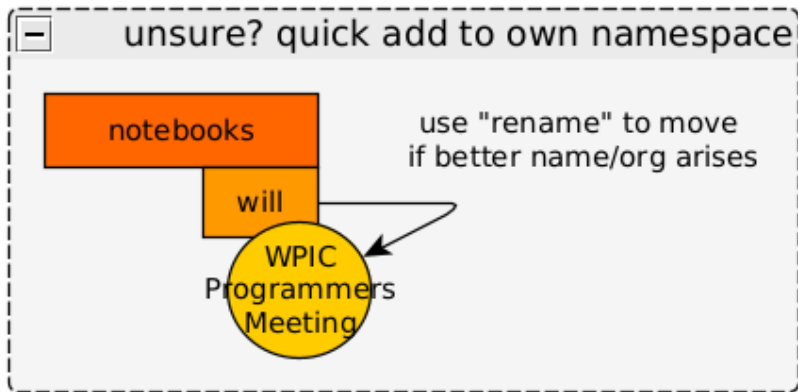
New name:

Ignore errors and skip pages or files that can't be moved

Rewrite links right after the move completed

Even poorly structured links

- use `publications:in_prep:XYZ` with abandon?
- notebook self nest for organization



Example

<https://lncd.pitt.edu/wiki/doku.php?id=notebooks:will>

[du/wiki/doku.php?id=notebooks:will#w07](https://lncd.pitt.edu/wiki/doku.php?id=notebooks:will#w07)

2023w07

- `lcmode` + MRRC `matlab` code for [Magnetic Resonance Spectroscopy Imaging](#) runs on `rhea`. MRRC still syncing older Hc.
- `r2prime` out for LEAP, shared w/Manual. screenshot `palladium` as highest iron/`r2prime` value.
🌐 <https://github.com/LabNeuroCogDevel/r2prime-prisma>
- `mergePet` - quick run for rIVBA update. still works (now 7099 columns)!
- `presentation` expires. used for `AuditorySteadyState` for [Electroencephalogram](#). have WIP python version 🌐 <https://github.com/LabNeuroCogDevel/SteadyState.py>
- `wpicprogammmers` - meeting. Andrew using `deepMREye`. Alex&Tyler like figwheel at bridges, also have docker storage issues. Warren struggling to get MRRC 7T data as well.
- `matlab` code from Kyle to get triggers from `seeg` data during `rIHabitTask`
- wiki updates: [Wiki Map Description](#), [Wiki Tricks](#)
- downloaded `gordonatlas` for `ABCD` share on `NDA`. month old py code no longer works for oracle? had to update syntax for pull

TODO:

- NDA upload for [Habit Study \(R37\)](#)

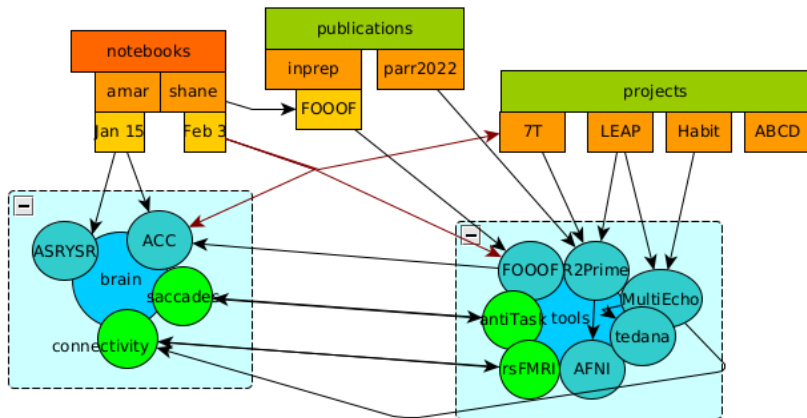
Editing pages

notebooks:will

- create new page by linking from another
- paste images
- auto-complete links
- rename page
- view changes

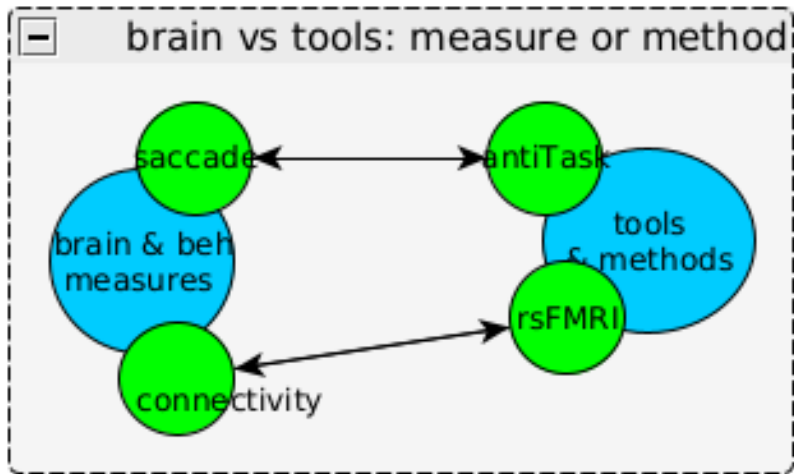
Organization

- top level namespaces to encapsulate free-for-all underneath



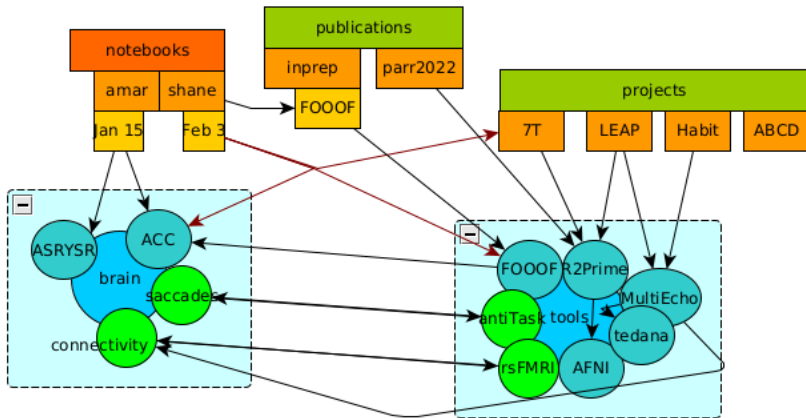
Brain Vs Tools

- :brain for measures, :tools for methods
- worth the distinction?



Organization

- does this work? suggestions, critiques? any utility?
- ":funding" under projects? notebooks? own top level namespace
- :admin:myapps vs :tools:magicwormhole



nested new page VS new headlines?

back linking VS overview

- `tools:r2prime` and `tools:tat2` are better separate
- weekly updates in notebook make sense to have scroll?
 - blog plugin

Notebook template/standard

- standard on weekly headings?
- date formats: ~February 13, 2023", "November 14th", "11/14/22", "2023w07"
- prefer: yyyy-mm-dd (for sorting)

Not mentioned

- Access Control List – private vs public `?do=admin&page=acl`
- recent changes, page diff `?do=recent`
- wiki formatting, esp `<code></code>`
- specials includes `{{simplenav>notebooks}}`, `{{ORPHANSWANTED}}`
- sidebar, emoji headers

R >= 4.1 lambda

- for newer versions of R: `|>` pipe and `\()` lambda syntax
- all versions: `invisible()` and `var <-` to suppress output

```
dbl <- function(v){  
  return(v*2)  
}
```

```
dbl <- function(v) return(v*2)
```

```
dbl <- function(v) v*2
```

```
dbl <- \ (v) v*2
```

bonus: don't print value if run in console

```
dbl <- function(v) invisible(v*2)
```

```
dbl <- function(v) rval <- v*2
```

R \geq 4.1 pipes



R>=4.1 pipes

```
library(magrittr) # %>%  
  
dbl(dbl(2))  
dbl(2) %>% dbl  
2 %>% dbl %>% dbl      # magrittr %>%  
2 |> dbl() |> dbl()   # base R |> req func have parrens
```

8

Base R: Backticks

using backticks to make infix notation functional

```
1 + 2  
`+`(1,2)
```

3

Backticks and piping

```
d %>% `names<-` (letters[1:4]) %>%  
  mutate(bc=b+c) %>%  
  head(n=3)
```

a	b	c	d	bc
a	1	6	10	7
b	2	5	12	7
c	3	4	14	7

Backticks for DIY infix

```
`%wf%` <- \(a,b) a+b  
1 %wf% 2
```

3

example dataframe

example data frame. i column repeats once (grouping variable).

```
d <- data.frame(i=letters[1:3],  
                x=1:6, y=6:1, z=seq(10,20,2))
```

i	x	y	z
a	1	6	10
b	2	5	12
c	3	4	14
a	4	3	16
b	5	2	18
c	6	1	20

magrittr pipes, basic mutate

```

# loading dplyr is noisy. silence just the startup messages
suppressPackageStartupMessages({
  library(dplyr)})

# pipes (from magrittr) can be broken across lines
d %>%
  select(i, x, y) %>%
  # ----- order matters -----
  mutate(x=x*2, a=x,
         b=y,   y=y*2) %>%
  head(n=3)

```

i	x	y	a	b
a	2	12	2	6
b	4	10	4	5
c	6	8	6	4

dplyr – version check

- frequent changes and deprecation! tidyselect from 1.1.2 (2022-02-21) doesn't have 'pick'

```
paste(packageVersion("tidyselect"),  
       packageDate("tidyselect"))  
# also packageDescription()
```

1.2.0 2022-10-10

dplyr vocab

- `<tidy-select>` (select, across) - find a column
- `<data-masking>` (mutate, group_by, filter) - column into env like 'with()'

tidy-select

```
d %>% select(contains('x'))      # string compare
d %>% select(matches('x'))       # regular expression
d %>% select(starts_with('x'))
d %>% select(ends_with('x'))
d %>% select(any_of(c('x', 'a')) # 'all_of' would error b/c no 'a' column
```

x

1

2

3

4

5

6

replaces mutate_{xx} functions w/across

```

# deprecated 'mutate_*'
d %>% mutate_if(is.numeric,           function(v) v*2)
d %>% mutate_at(vars(matches('x|y|z')), function(v) v*2)

# new 'across' w/tidymodels: where, matches, starts_with, all_of, any_of
d %>% mutate(across(where(is.numeric),   function(v) v*2))
d %>% mutate(across(matches('x|y|z'),    function(v) v*2))
d %>% mutate(across(all_of(c('x','y','z')), function(v) v*2))

```

i	x	y	z
a	2	12	20
b	4	10	24
c	6	8	28
a	8	6	32
b	10	4	36
c	12	2	40

across using more than one function

```
d %>% mutate(across(matches('x|y'),
                      list(dbl=\(v) v*2,
                           trp=\(v) v*3)))
```

same as

```
dbl <- function(v) v*2
```

```
trp <- function(v) v*3
```

```
my_funs <- list(dbl=dbl, trp=trp) # without names, get x_1 and x_2
```

```
d %>% mutate(across(matches('x|y'), my_funs))
```

i	x	y	z	xdbl	xtrp	ydbl	ytrp
a	1	6	10	2	2	12	12
b	2	5	12	4	4	10	10
c	3	4	14	6	6	8	8
a	4	3	16	8	8	6	6
b	5	2	18	10	10	4	4
c	6	1	20	12	12	2	2

c_across

```
d %>% rowwise %>% mutate(s=sum(c_across(matches('x|y'))))
```

i	x	y	z	s
a	1	6	10	7
b	2	5	12	7
c	3	4	14	7
a	4	3	16	7
b	5	2	18	7
c	6	1	20	7

c_across TRAP

(hint `sum(1:6) == 21`)

```
d %>% mutate(s=sum(c_across(matches('x|y'))))
```

i	x	y	z	s
a	1	6	10	42
b	2	5	12	42
c	3	4	14	42
a	4	3	16	42
b	5	2	18	42
c	6	1	20	42

{{var}} interpolation

```
my_vars <- c('x', 'y', 'i')  
d %>% select({my_vars}) %>% head(n=3)  
  
# doesn't work as expected  
# d %>% mutate(a={my_vars[1]}) # see .data[[]]
```

x	y	i
1	6	a
2	5	b
3	4	c

pick escape

'pick()' is also useful as a bridge between data-masking functions (like 'mutate()' or 'group_by()') and functions with tidy-select behavior (like 'select()')

```
d %>% group_by(i) %>% summarise(xm=mean(x))
d %>% group_by(pick({"i"})) %>% summarise(xm=mean(x))
d %>% group_by(pick({my_vars[3]})) %>% summarise(xm=mean(x))
```

i	xm
a	2.5
b	3.5
c	4.5

pick as opposite of across

quick aside, pick is designed to send partial dataframes

```
addxy <- function(d) d$x+d$y
d %>% mutate(xy = addxy(data.frame(x,y)))
d %>% mutate(xy = addxy(pick(x,y)))
```

i	x	y	z	xy
a	1	6	10	7
b	2	5	12	7
c	3	4	14	7
a	4	3	16	7
b	5	2	18	7
c	6	1	20	7

mutate({my_vars})

curly interpolation wont work in mutate

```
d %>% mutate(a={my_vars[1]}) %>% select(x,a) %>% head(n=3)
#d %>% mutate(a={{my_vars[1]}}) # "arg must be a symbol"
```

x	a
1	x
2	x
3	x

`.data[[var]]`

```
d %>% mutate(a=x*2) %>% head(n=3)
d %>% mutate(a=.data[['x']]*2) %>% head(n=3)
d %>% mutate(a=.data[[ my_vars[1] ]]*2) %>% head(n=3)
```

i	x	y	z	a
a	1	6	10	2
b	2	5	12	4
c	3	4	14	6

"Note that `.data` is not a data frame; it's a special construct, a pronoun, that allows you to access the current variables either directly, with `.data$x` or indirectly with `.data[[var]]`. Don't expect other functions to work with it."

walrus for variable names $\{var\} := exp$

```
d %>% mutate(x_2 = x*2)
d %>% mutate("{x'}_2" := x*2)
d %>% mutate("{my_vars[1]}_2" := x*2)
d %>% mutate("{my_vars[1]}_2" := .data[["x"]]*2)

dbl_col <- function(d, v){
  d %>%
    mutate("{v}_2" := .data[[v]]*2)
}

d %>% dbl_col('x')
```

i	x	y	z	x ₂
a	1	6	10	2
b	2	5	12	4
c	3	4	14	6
a	4	3	16	8
b	5	2	18	10

nonstandard evaluation w/ `{{v}}` and `enquo(v)`

if you really don't want to quote your variable names provided to functions.

```
dbl_col_q <- function(d, v){
  d %>%
    mutate("{{v}}_2" := .data[[enquo(v)]]*2)
}
d %>% dbl_col_q(x) # NB. not "x", just x
```

i	x	y	z	x ₂
a	1	6	10	2
b	2	5	12	4
c	3	4	14	6
a	4	3	16	8
b	5	2	18	10
c	6	1	20	12

Not mentioned

- qassert_m in LNCDR
- tidyr pivot
- split/lapply model

<https://github.com/LabNeuroCogDevel/lncdtools/>

- dryrun, D (DRYRUN=1)
- verb, V; VERBOSE=99 verb -level 2 hi
- eval "\$ (iffmain)"
- niinote, 3dNotes_{each},
- waitforjobs
- mkmissing, mkls, mkifdiff, mkstat
- mknii, dcmdirtab

Bashism - { }

```
a_123 b_123 c_123
```

Bash - repl

- alt-. - previous argument
- ctrl-r - fzf enhanced history)
- alt-r - wf special fzf alt-.
- ctrl-/ - undo
- ctrl-a/e
- ctrl-w/k ctrl-y

quick inspect

```
which verb
```

```
/home/foranw/src/work/incdtools/verb
```

```
head -n 5 $(which verb)
```

```
#!/usr/bin/env sh
```

```
# USAGE:
```

```
# VERBOSE=1 verb debug msg # prints "debug msg"
```

```
# verb not printed # no print w/o VERBOSE set
```

```
# VERBOSE=1 verb -level 99 not shown # can set -level
```

ld8

extract luna_date8 from stdin (pipe) or arguments. useful for loops

```
echo a b 10931_20201231 c 11234_20230130 c | ld8
```

```
10931_20201231
```

```
11234_20230130
```

```
ld8 a b 10931_20201231 c 11234_20230130 c
```

```
10931_20201231
```

```
11234_20230130
```


selld8

```
selld8 l | grep Brain | head -n3
```

10202_20230130	33.5	M	eeg	BrainMechR01
11561_20230125	21.4	M	eeg	BrainMechR01
11910_20230119	26.6	F	Scan	BrainMechR01

selld8: ctrl-x ctrl-l in rhea command line

20230217/lunaid_select_CxCl.gif

ld8info

ld8	age	sex	dob
10143_20230217	30.74	F	1992-05-22
10568_20230217	32.57	M	1990-07-23

dryrun

dryrun wraps an expensive or destructive operation with the ability to just echo instead
useful for debugging before actually committing to the action

```
expensive_cmd(){ echo "running $* ..."; } # e.g. recon-all
export -f expensive_cmd # not needed for external commands/progs
for subj in 10921 10921; do
    dryrun expensive_cmd $subj
done
```

dryrun in clean environ

without doing anything special, the script runs as if dryrun wasn't there

```
20230217/expensive.bash
```

```
running 10921 ...
```

```
running 10921 ...
```

dryrun with DRYRUN set

```
DRYRUN=1 20230217/expensive.bash
```

```
expensive_cmd 10921
```

```
expensive_cmd 10921
```

D

D is a shortcut for "DRYRUN=1"

```
D 20230217/expensive.bash
```

```
expensive_cmd 10921
```

```
expensive_cmd 10921
```

warn

a shortcut for printing to stderr instead of stdout (avoid sub-shell capture or writing to file)

```
# echo "$*" >&2  
warn "ERROR: thing went wrong"
```

```
ERROR: thing went wrong
```


verb

- only print if VERBOSE is set or set and higher than '-level N'.
- V is a shortcut for VERBOSE=1

```
VERBOSE=1 verb "is printed"
```

```
V verb "is printed again"
```

```
VERBOSE=99 verb -level 2 "is printed despite higher level"
```

```
verb "not printed"
```

```
V verb -level 2 "not printed again"
```

```
is printed
```

```
is printed again
```

```
is printed despite higher level
```

Not Mentioned

- maskvolume, 4dBrikConcat, 3dmaskave_{grp} (support Datatable files!)
- dcmnii, dcmniidi
- mkls, mkdiff, mkstat

bats

```
foo() { echo "foobar"; }  
function test_foo { # @test  
  run foo  
  [[ "$output" == "foobar" ]]  
}
```

```
function test_bar { # @test  
  run foo  
  [[ "$output" == "bar" ]]  
}
```

```
bats --tap 20230217/testme.bash
```

1..2

ok 1 test_foo

not ok 2 test_bar

iffmain

iffmain outputs a bash template for running a function as the entrypoint with strict mode bash and an exit trap

```
iffmain myfunction
```

```
if [[ "$(caller)" == "0 *" ]]; then
  set -euo pipefail
  trap 'e=$?; [ $e -ne 0 ] && echo "$0 exited in error $e"' EXIT
  myfunction "$@"
  exit $?
fi
```

eval iffmain

we just need to add a single line to the bottom of our all functions bash file.

```
eval "$(iffmain foo)"
```

then when we run the script, we get the output of the 'foo' function

```
20230217/testme.bash
```

```
foobar
```

Examples in allscripts

tedana, db age, 7t decon, mrqa