

## Research Article

# How to use orgmode for reproducible research, rough cut version 1

Arindam Basu<sup>1</sup>

1. University of Canterbury, New Zealand

In this article we discuss how to use orgmode for writing and weaving data analysis.

## A brief tutorial on org-mode: introduction

Org-mode is a notetaking, time-management, and content authoring and thinking tool for emacs, a free and open source text editor, as a matter of fact, almost an operating system in itself that uses elisp programming language. However, you will not need to know anything about programming languages to use Emacs. In order to use org-mode, while it can be used in any text editor, it is best to first install Emacs and then use org-mode.

Here are some use cases how orgmode can be used:

- Use org-mode as an outliner.
- Use org-mode as a planner
- Use org-mode to conduct data analyses
- Use org-mode to weave data analyses with paper writing
- Use org-mode to write academic documents
- Then export that document to various formats after parsing with pandoc, a universal document parser.
- You can also upload the org document on a git repository such as github and can share the document widely with text and code.

This implies that in itself and with pandoc, with orgmode, you have a truly reproducible research tool. Besides, if you provide the document name to pandoc command, you can convert the document to other formats and get published in formats such as Word/LaTeX and so on. You can even construct

pipelines to write once and get outputs many times and in many formats. For example, you can write in org and then use pandoc to convert the document from org to markdown and then use the markdown document to be exported further. Or, you can write the document in org mode and then use pandoc to convert the document to latex and then use something like overleaf to convert the document to latex and pdf and use it that way. Or you can transform the document to word and then further process it.

In this tutorial, my aim is to write the simplest workflow that we can use to work with emacs and org-mode to write data-driven documents that can be converted to a number of different formats. A popular format is to write in jupyter notebooks.

## **Mechanics of content authoring in orgmode**

### **Adding headers and converting different types of headers**

You add header levels by adding an asterisk at the beginning of a line. The number of asterisks will determine the level of headers you want to insert. A second level header will be nested within a first level header. I usually do not use more than two levels of headers and that seems to work well for me. YMMV. By default, orgmode does not number the headers. Suppose you have written a large document or you are writing a large document and you want to jump from one header to another. You do C-c C-n which is the next heading. Or C-c C-p to jump to the previous heading or if you want to jump one level up, then you do C-c C-u. If you do C-c C-j, then emacs will ask you to move the cursor to a particular header and then hit enter and the header with its content will open up. This is referred to as drawer in org-mode.

### **How to add headers**

If you do M-RET or press the alt key and enter, then it will insert a header at the same level next to the header in which you are working. But if you want to insert a new header BEFORE, then you will take the cursor to the beginning of the header and then hit M-RET and it will insert a new header of the same level but above the current header. Same thing if you do C-RET, that is hit control and enter at the same time. If you want to convert a second level header to a first level header, you do M-S-LEFT that is alt key plus shift key plus left key. Likewise, if you want to convert a second level header to a third

level header, you do M-S-> If you want to move your headers up and down, then do M-up or M-down. This will move the header and the associated content up or down depending on what you did.

## Cutting and pasting

There is no concept of selecting a chunk of text and then cutting or pasting. Pasting in Emacs world is referred to as "yanking" and therefore the symbol is y. Cutting and copying are two different things. In copying you copy the content in the header with C-c C-x M-w, this will leave the text as is and then copy it to something called a "kill ring". On the other hand, if you want to cut the text, you do C-c C-x C-w and it will 'cut' the text and place it in the kill ring. Then, wherever you want, you do C-c C-x C-y.

## What if you want a line to be converted to a heading?

Place the cursor in the beginning and hit M-RET. Basically, wherever you want the heading to begin, place the cursor there and hit M-RET and it will insert the heading in the same level. The same effect if you write a single line and hit C-c \* and it will create a subheading. So same level heading is M-RET, subheading is C-c\*

*This line was converted to a header by c-c star*

I was a headline once, C-c \* turned me to a simple line

These are lines

This is a second line

All these were turned into headers by c-c\* by selecting them first of course

## What if you want a line to be converted to a heading?

Place the cursor in the beginning and hit M-RET. Basically, wherever you want the heading to begin, place the cursor there and hit M-RET and it will insert the heading in the same level.

### *Visibility cycling*

using tab and shift-tab, the headers can be cycled to be visible. So, if you hit tab next to a header, that header will be shown and the document will fold up. If that header contains any text associated with it or nested headers, then you will see an ellipsis in the form of three dots to the left of this header. This

is ONLY going to work if you place your cursor next to a header, it is not going to work in a paragraph like where you are reading it. So the way it works is something like as follows. You place the cursor next to the header and then hit tab key. The header only shows and the content of the header (known as tree) hides itself. Then you hit tab again and the header with the content shows up. This is useful when you are editing a large document where you do not want to be distracted with too many headers and you only want to work on some part of the document, not all of it. This works like a toggle button.

*Writing tables*

*Adding images to orgmode documents*

*Adding hyperlinks*

*Adding citations*

*Adding codes*

*Exporting orgmode documents to other formats*

## **Visibility cycling**

using tab and shift-tab, the headers can be cycled to be visible. So, if you hit tab next to a header, that header will be shown and the document will fold up. If that header contains any text associated with it or nested headers, then you will see an ellipsis in the form of three dots to the left of this header. This is ONLY going to work if you place your cursor next to a header, it is not going to work in a paragraph like where you are reading it. So the way it works is something like as follows. You place the cursor next to the header and then hit tab key. The header only shows and the content of the header (known as tree) hides itself. Then you hit tab again and the header with the content shows up. This is useful when you are editing a large document where you do not want to be distracted with too many headers and you only want to work on some part of the document, not all of it. This works like a toggle button.

## Concept of a sparse tree

So let's say we are working on a large document and I just want to work on this particular segment where I am fully immersed. I do not want to see anything else. Sparse tree will enable that. To enable that, use C-c /

## Writing paragraphs

## Copying and pasting or cutting and pasting

## Writing lists

Start with a dash sign, so

first item

☐ this checkbox was inserted with M-S-RET

test

test 2 this was done with first typing 1. then M-RET

and this with M-RET and so on

third item was place ABOVE second item with M-up

second item

third item again was indented

then unindenttd with

with M-> indented again

and M-RET inserted this in the same level

*I was a list item once, C-c \* turned me to a header*

First item

Second item

Third item

the following is an unsorted list

Fifth item

First item

Third item

But doing a C-c ^ sorted them to an alphabetically sorted list

And then to get out of the list, hit RET twice

## Concept of Blocks

HashPlusBEGIN and HashPlusEND begins and closes blocks. Usually used for codes etc.

## Tables

Start with a | symbol to begin a table

Start a table with C-c |

Then use | as a column separator

Then use | - as a horizontal line

This will turn the first row a header row

Put cell contents between | and |

Realign table with RET or TAB or C-c C-c

Within the table, TAB moves to next cell

Within the table RET moves to next row

Put the cursor outside the table and RET to end the table

Exposure	Cases	Controls
Exposed	120	20
Non-exposed	60	100
Total	180	120

## How to edit tables

What to do	Instructions
Sort table	C-c^
Move between columns	M-a or M-e
Move row down	M-DN
Move current columnn to left or right	M-> or M-<-
Insert new column to the left	M-S->
Delete column	M-S-<-
Insert row above the current	M-S-DN
Move row up	M-UP
Delete Row	M-S-UP
Insert horizontal line	c-c -
Insert horizontal line and move cursor	C-c RET

## Other miscellaneous things with tables

What do you want to do	Instructions
Copy table region	C-c C-x M-w
Cut table region	C-c C-x C-w
Paste region	C-c C-x C-y
Split table	M-RET
Sum numbers	C-c + then C-y
Copy Down	S-RET
Edit in separate window	C-c ` and finish with C-c C-c
Import tab separated table	M-x org-table-import
Convert a region into table	Select then C-c
Export Table	M-x org-table-export
For long tables, display header	M-x org-table-header-line-mode
Transpose Table	M-x org-table-transpose-table-at-point

## Convert a region to table

First type space separated data in a region

Select the region

Type C-c |

The following table was drawn on the basis of the above notes

Student	test <sub>results</sub>
Tess	100
Kim	200



## Transposition of a time dependent table e.g.

Write a space separated table

Then put the cursor in first row first column

Type M-x org-table-transpose-table-at-point

The following table was produced with the above steps

Individual	xxx	yyy
Score <sub>t1</sub>	100	99
Score <sub>t2</sub>	120	98
Score <sub>t3</sub>	130	49

The org table mode can be turned on with M-x orgtbl-mode

## Using the table as a spreadsheet

The table can be used as a spreadsheet. In order to do so, it is useful to use the concepts of references.

Take the following table

Exposure	Cases	Controls	Total
Exposed	120	40	160
Non-exposed	40	80	120
Total	160	120	280

Let's say we want to work on this as a table

If you want to turn on/off grid display, do C-c }

If you want to find out coordinate of a grid, C-c ?

You can find the reference of a cell with C-c ?

Rule is: @ROW\$COLUMN

@0\$0 refers to the current cell

@2 implies second row of current column

\$1 implies first column of the current row

Ranges are indicated with beginning .. end

The beginning and ends are INCLUDED

\$1..\$3 means the row elements of first, second, and third columns

Otherwise, indicate the row with @ symbol, so

\$1..@2\$3 would mean from the first column to second row of third column

These range references return a VECTOR of values

Feed them into vector functions

If you want to reference another table, set a name to it, by:

{=org} #+NAME: Name of the table

After writing the formula below the table with:

{=org} #+TBLEFM: <expression>, place cursor on the line and

Do, C-c C-c

You can write more than one formulae

If you do so, place the cursor next to each line and do C-c C-c

## Writing the formula

A formula can be any algebraic expression

Example: vsum(@2..@3) will calculate the vertical sum

Use #+TBLEFM: write the formula

## How to insert hyperlinks to a document

Let's say we want to place a link to an external resource such as Medium or Curvenote. We will write like this:

Medium Arin's page

Org will turn the link to a blue hyperlink

<https://www.curvenote.com>

Org will turn the page to a blue hyperlink for Curvenote

But this time it leaves the URL as is as we did not provide a description

For internal links, we can do similar things with:

A section with *Concept of drawers*

Write the header between two squared brackets

Otherwise, for tables, do something like #+NAME: somename and then

Add the somename between the square brackets, so

*longtable* will lead to the long table

You can also insert link with C-c C-l, so

NixOs webpage was inserted with this keystroke

## Concept of drawers

Hiding information so use ::Draw1:: Then write something and then ::END:: ends the drawer

## Orgmode as productivity tool

Orgmode can be used to store ideas and productivity stuff while writing, including todo items and logging processes.

Create a to do item with headline with TODO keyword, or

C-c C-t

The following todo item was created that way

Rotate between TODO, DONE, and Plain header with C-c C-t

Or, do S-> or S-<- shift right or left

Once you have done that, and entered a few stuff, then

Use C-c / to use it as a sparse tree (see *Concept of a sparse tree* )

Add a new todo entry below the present one with S-M-RET

The sparse tree will show the todo items, not the done ones

We can add something like INPROCESS before DONE, so we modify

the ~/.emacs.d/init.el file with

```
(setq org-todo-keywords '((sequence "TODO" "INPROCESS" "|" "DONE")))
```

Then exit Emacs and restart and revisit an org file

How to set up dependencies for todo items

This means, you list what items must be completed before

Another item can be marked as done

and so on

Record timestamp and note when change a todo state

C-u C-c C-t then C-c C-c

This is a good practice as this forces you to recognise

What you did with it!

Prioritise tasks with #A #B #C where A = highest priority

## INPROCESS Orgmode as a productivity tool

State "INPROCESS" from "TODO" [2021-11-04 Thu 13:45]

Still learning,

*TODO Complete the tutorial by today [1/2]*

*DONE Find out how to rename files [2/2]*

## How to use tags in orgmode learning tutorial editing

Headline can have tags at the end of the headline

Tags are added with :tagword:

Two or more tags are written as :tag1:tag2:

Tags have same colour as headlines

Subheadings i.e., heading level 2 from heading level 1 inherits the tags

So in the following example, "Adding images ..." has tag learning

Adding images also has two subheadings "Adding images from files" and "Adding images from websites"

Adding images from websites does not have a separate tag but it can be said that this one has inherited the learning tag

You can specify tags for files so in the preamble, do

```
{=org} #+FILETAGS: :learning:tutorial:
```

See, same as tags for headings

Tags are searchable

Search tags with C-c / m

How to set tags:

After headline, colon tagword colon

C-c C-q

C-c C-c works only when you are in a headline

In the preamble of your document with #+TAGS: keyword

## What are properties? learning

You can assign properties to headlines or agenda items using drawers where the drawer name is PROPERTIES between colons and assign attributes and values or keys and values. So for example an article can be assigned properties such as title, author, year as follows

\* Article Collection

If you have entered properties anywhere in the document, you can search for them using the C-c / m and specifying the name of the property that you created. As with the tags, properties can be inherited.

## Using orgmode for universal idea capture productivity uses

First, define a directory where you will keep notes

```
(setq org-default-notes-file (concat org-directory "/notes.org"))
```

Write the above in ~/.emacs.d/init.el file

So, notes.org is the directory where I am going to store my notes

Hit M-x org-capture

Select a template, for the first time it will show t for todo

Write something and then hit C-c C-c

When you do it for the first time, it will ask you to create org folder, say yes

It will create /home/arini/org directory and will place a notes.org in it

We often want to capture more than todo items, for this, create a template

```
Template (setq org-capture-templates '(("j" "Journal" entry (file-date-tree "~/org/journal.org") "* %\n\n %U %i %a")))
```

## How to write rich text documents in orgmode?

### Writing paragraphs

At least one empty line separate paragraphs

Use blocks to insert special elements such as VERSE and QUOTE

**Bold** is like this, *italic* is like this, and underlined

verbatim and code

~~strikethrough~~

Superscript: x<sup>2</sup>

Subscript x<sub>1</sub>

Toggle them with C-c C-x \

Symbols, say alpha is  $\alpha$  (view with C-c C-x \)

For equations, set #+OPTIONS: tex: t then

for single line equations, and for multiline equations,

...
$$equation$$
...

Preview with `#+STARTUP: latexpreview` and then `C-c C-x C-l`

Horizontal rules are with five dashes

Footnotes are given as `[fn:1]` and then `[fn:1]` Description

View footnotes with `C-c C-x f`

## Adding images to orgmode document

Do the following, so:

Write a caption of the image

Give the image a name

Insert the image with `[[image url]]`

then view it inline with `C-c C-x C-v` and toggle the view

Then view it with `C-c C-x C-v`

## Adding citations

Before adding citations, prepare a bibtex file

Say the bibtex file is named `test.bib`

Add it to the preamble with `#+bibliography: test.bib` in the block

Wherever you want to insert, type `M-x org-cite-insert` then type the first letter of the citation key or something

Then hit tab key

It will show the matched name, accept it with `RET RET`

Or manually with `[cite: @citation_id]`

Make sure you have orgmode version 9.5 or above

find out with `M-x org-version`

Egger wrote `@Egger1997-wq`

## Adding and evaluating codes

Codes can be added with blocks

Codes can also be evaluated that way

this is a test

But that was a trivial example. We will show two examples; one for R and the other for python.

### *What to do*

Start a block with `#+begin_src R (or Python) ... #+end_src`

Write your code

Place cursor within the code block

C-c C-c

### *Here is an example with R*

```
x <- 2
```

```
y <- 9
```

```
print(x * y)
```

```
x = 1
```

```
return x
```

**Supplementary data:** available at <https://doi.org/10.32388/B5RVO7>

## Declarations

**Funding:** The author(s) received no specific funding for this work.

**Potential competing interests:** The author(s) declared that no potential competing interests exist.