How I now write my blog (with Remarkable and Claude Code)

Introduction

I've maintained a blog since 2024 [confirm this, get more specific date], and over that time have written more than 200,000 words. So, around two reasonable novels worth of 'content'. Over these years, the focus of the content has changed, along with my primary motivations for writing. What's also changed is *how* I go about writing a blog post, from initial ideas to completed and published content.

[Mark this as standout/noteline] When I started writing this post, I thought I'd cover not just how I write now, but also my previous writing setup. I also thought I'd discuss my motivations for writing. But, it turns out the current writing setup alone seems a substantial enough piece of content. So consider this post not just a 'post in itself' (sort of Kantian?) but also the start of a short *on writing* series of posts, where I'll cover some of these additional areas in due course.

Blog platform

My blog uses Quarto [ref], so at some level involves maintaining a nice big bag of files with .qmd extensions. For those with a bit of a software background, the clue's in the file extension. A .qmd (Quarto) file is like a markdown (.md) file, but with something added. (A markdown file is like a simpllified way of writing an HTML page, which makes it a bit easier for humans to read and maintain.) The 'something added' by the Q is the ability to include evaluable code chunks, sections of the page which are instructions for various types of computer program - called *engines* - to do something and return what it's done. This was especially important when much of my blog content was focused on statistics and data science, as then the main engine powering the code chunks was R, the open source statistical software. Much of my blog content these days doesn't need these engines, so the ability to include them as chunks is now less important. But anyway, that's why my blog uses Quarto.

Maintaining a Quarto blog means, after setting up the index page, having a folder called _posts/_; this _posts/_ folder then has a series of carefully named subfolders, one for each specific blog post on the site itself. Each of these subfolders determines the specific url for a specific blog post. For example, my most recent post[link to it here] was about the book **The Dilbert Future**[link to book], and was in a folder _posts/dilbert-future-in-retrospect/_, and so has a url ending '/posts/dilbert-future-in-retrospect/' too.

Inside each of these differently-named folders in _posts/_ is an identically named folder, _index.qmd_. This is where both the 'data' (writing) and 'metadata' (post attributes) are contained, in separate parts of the .qmd file.

- At the top of the index.qmd file: the YAML (Yet-another-markup-language) header. This comprises a series of key-value pairs telling Quarto things like: categories, publication date, author(s), title, and subtitle. By default, these YAML contents are always at the start (head) of the file, and fenced off from the rest of the file contents by three hyphens at the start, and three hyphens at the end. These triplets of hyphens 'fence the YAML in', away from the rest of the content, and so are known as 'fences'.
- Below the YAML header: the body text. This is the content itself: the text you're reading right now, and information about attributes that the text should take on. Attributes include: should the text link to something? should the text be **bold**, or *in italics*, or be a header? Should the text link to some footnote text [footnote: like this], and so on and so on. For a technical post, as mentioned, the body can also comprise some code chunks, evaluated by engines, an interleaving of contents for people (plus formatting) and contents for specific computer programs. [footnote: this interleaving borrows from a related writing paradigm, literate programming and scientific notebooks, which became popular in Jupyter notebooks in python]

If you think having to think about .qmd file structures, metadata, and folder organisation is maybe a bit of a distraction from the act of *writing*, especially *freely and creatively*, you're right. For me, at least, the act of having to decide exactly which words to write, what arguments to make, which references to mention, which analogies and metaphors to employ... all of this is demanding and challenging in its own terms. Having to think about how to express formatting and structure to computer programs too feels like too much overhead.

That's why, when I'm trying to write freely and creatively, I try to *split* and *separate* the tasks of *writing* from the tasks of *putting contents into a blog post*. Keeping these tasks separate is perhaps the single most important factor in my current workflow.

So, how does this workflow (when it does flow) work in practice?

The Steps

Stage One: Ideating, Freewriting and First Pass Editing

I start by *going somewhere*, and *not bringing too much*. i.e. I head to a cafe (if during the day/weekend) or a bar or pub (if in the evenings), with my Remarkable, its folio cover, and some semblance of an idea. Then, with a table and libation sourced, I aim to draw, write, or type materials - in a remarkable Notebook in a 'blogs' folder - for at least an hour. Having a timer is crucial: if I'm travelling very lightly, I use my wristwatch (a fitbit) or phone to keep time. If I'm travelling with a small backpack, I might bring a 'time timer' too, such as that shown in the photo below

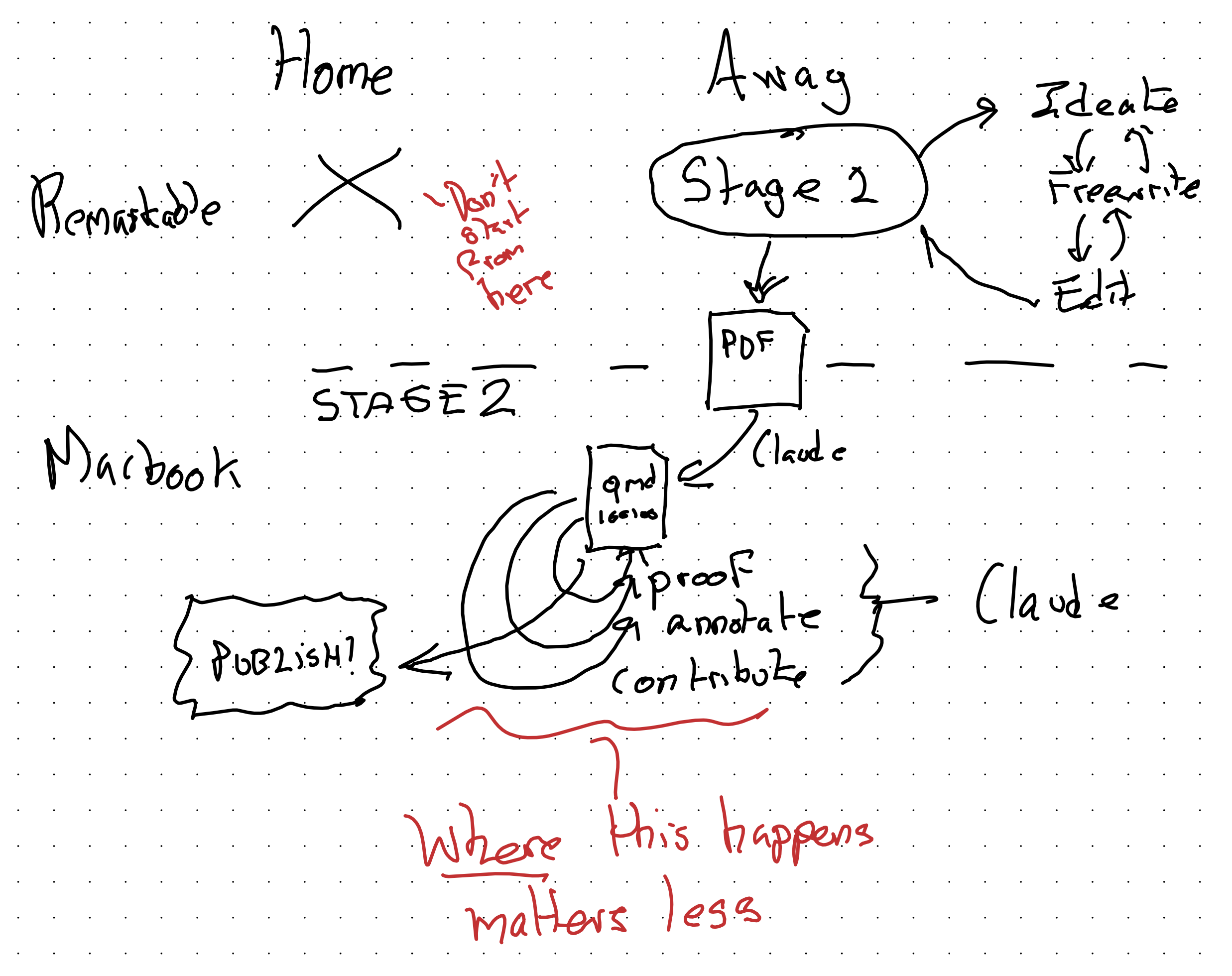
[PHOTO OF remarkable setup]

I both work at home, and I rest from work at home. This is why getting some kind of separation for writing is crucial for me. Another space: another headspace.

As implied by the subheading, this stage one really comprises some distinct substeps: Ideating, free-writing, editing.

Ideating is having that semblance of something potentially coherent, comprehensive and cohesive to think and write about. Good ideation might involve very little writing, or none at all. Thinking itself, in this third space, can be a valid form of ideation. So can making notes, and so can rearranging and linking notes.

The remarkable can help markedly with some forms and varieties of ideation. For example, it makes it very straightforward to draw simple mindmaps and concept maps [footnotes: mindmaps can be considered concept maps with one central node; conversely concept maps can be considered mindmaps with multople central nodes], like the following:



Remember that markdown-plus-code-engine interleaving I mentioned previously? That which separates .qmd files from mere .md? Well, as with this figure above (which in a sense gives the game away about where this blog post is going), the Remarkable allows me to do something similar, but with content for and from different parts of my brain. Text like this, which is fairly easy to bring into the strictures of a .qmd file, is something I can generate fairly easily with the keyboard extension to the Remarkable folio 'cover'. But some ideas are best expressed graphically: that's where the Remarkable stylus comes into its own. Yet another reason why I start with the Remarkable is because I want to be able to switch between drawing, writing, mapping, and typing as the specific tasks of expression demand it.

Stage 1.5: The Interface

After I have produced some initial content on my Remarkable, which I consider 'complete enough' for a post, I make use of the following (ahem) Remarkable features:

- Feature one: The Remarkable syncs across devices, including to my mobile phone and, crucially, my Macbook Pro
- Feature two: Anything within the Remarkable Macbook Pro can be exported as a PDF.

So, after syncing my Remarkable contents to my Macbook Pro, I then export the blog material I've written into a PDF.

That's where my new workflow (or writeflow?) gets 'interesting'.

Next, I open up Claude - with both my blog codebase and the new PDF file as environment/context - and instruct it to *turn the pdf into a .qmd file in an appropriate location in the blog codebase*. [Footnote: Usually I do this in Claude Code, with the codebase as the active folder, adding the PDF either within the project, or

And usually - almost always, in fact - it manages to do so very effectively, properly inferring how my formatting in the PDF links to markdown formatting, identifying the title and subtitle for the YAML, and even producing appropriate topic categories.

So, what this means is - thanks to **Porter Claude** - that a lot of the technical overhead is not just moved to a different step in the workflow, but abstracted away almost completely!

Stage 2 : MaClauding in VSCode

Because of RemarkaClaude, I now, as if by magic, have my first draft of my post - started on the Remarkable - transformed into a perfectly viable .qmd style blog post. I could of course then just publish it at this stage, with Claude doing only the interfacing work. However, I'm happy for Claude to get a bit more involved.

Typically, this means (usually through the Claude Code extension in VSCode):

- **Proofreader Claude*: Asking Claude to carefully proofread my text. I give it instructions to spot any obvious typos, and edit rights to change them, while making clear I don't want my words to be re-written any further.
- **Gopher Claude**: Guiding Claude through implementing any ambiguous notes and instructions I left for it in the Remarkable PDF. (I can think ahead. At the moment I usually suggests Claude tasks by putting them in square brackets [like this]). This might include, for instance, finding links to other blog posts or external pages, or clarifying details after web searches which appear in the main text. (In [the previous blog post] this involved, for instance, finding the exact date of Scott Adams' death.)
- **Formatter Claude**: Wrangling over blog topic categories, titles, and formatting used. (For example, in the last post I started off thinking that each of the 'predictions' should be a level 2 header, but then moved each to a level 3 header with a broader level 2 'Predictions' grouping. Exciting stuff!)
- **Scriptdoctor Claude**: Asking Claude to make *suggestions only* about whether there are any awkward passages or phrasing that could be improved. For this I ask Claude to show suggested before/after phrases, for me to approve or not, rather than just editing the .qmd file without my express permission.

There are then some more optional phases, which I've tended to employ in some recent posts:

- **Fact Checker Claude**: Asking Claude to add, as clearly self-identifying footnotes, fact checking and contextualisation materials.
- **Claude the Author**: Finally, *sometimes*, asking Claude to make distinct, self-identified, contributions to the body of the text itself. (For example, int his last post this took the form of the Judge Claude section.)

Summing up

Wow! That's a lot of Claude!

I was going to write a longer piece, comparing my current writing and publishing workflow, with how it used to be in the past; comparing my earlier content with my most recent content; and comparing my initial motivation to start, with my current motivation to continue writing.

But maybe this is as good a place as any to finish. This is a more applied and technically focused post than I was initially planning. So maybe it's best to stop here and turn this from a single post to a short series of posts *on writing*.