Writing code can improve your workflow.

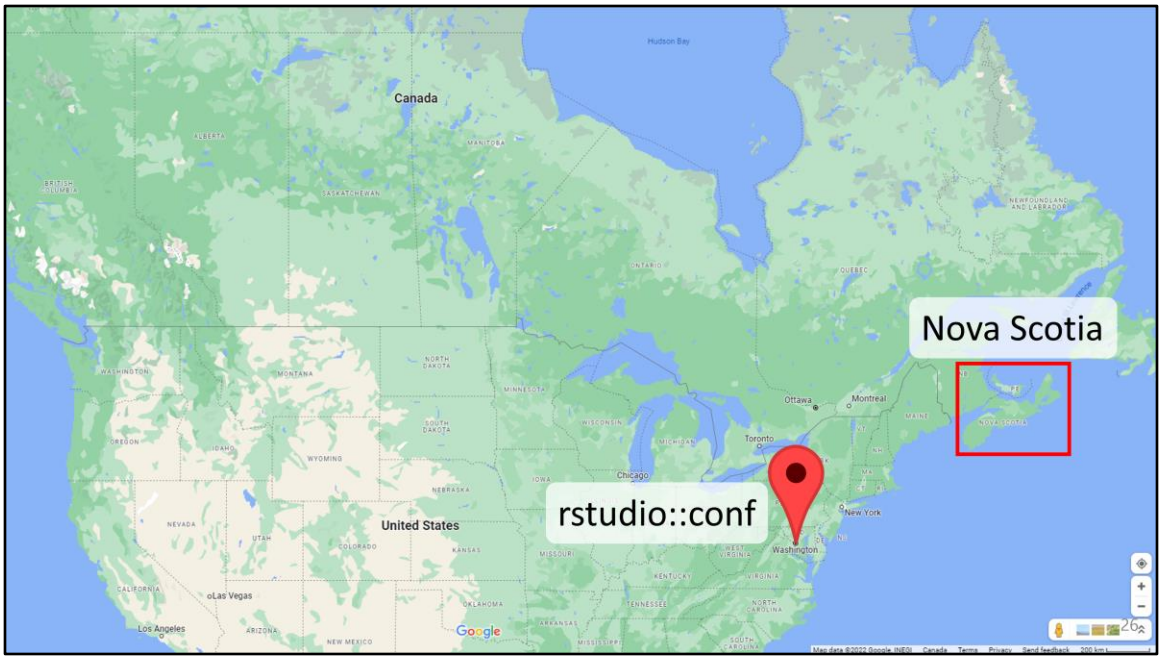
But coding can also be tricky, frustrating, and time consuming.

In my experience, it can be difficult to convince yourself – or more often your managers – that it is worth spending the time up front to write code to automate repetitive tasks like data wrangling, especially if there is an existing process that is “good enough”.

Image
Photo taken by DD

My goal today is to demonstrate that it is worth spending time up front to write the code, because in the long run this will save time, reduce errors, and improve the workflow.

Images
Clock: made by DD
Garbage Can: modified from image designed by Freepik.com
Workflow: modified from image on Flaticon.com



Research Scientist at a small company called the Centre for Marine Applied Research in Nova Scotia, Canada.

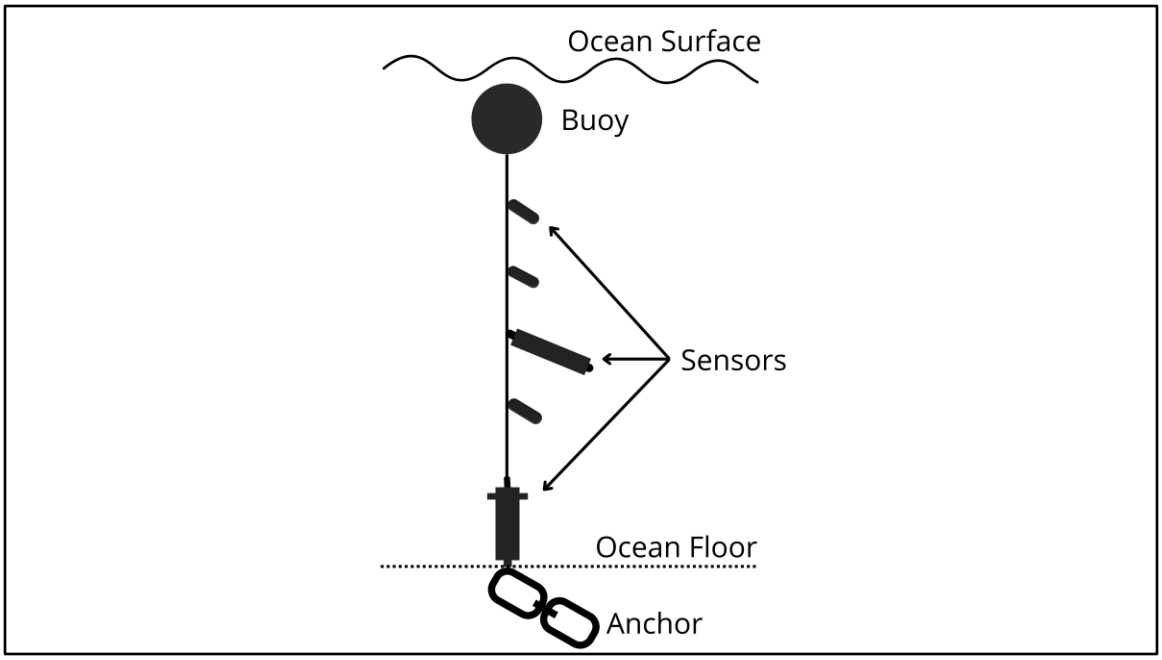
Images
Map modified from GoogleMaps

Coastal Monitoring Program: to measure ocean variables like temperature, salinity, and dissolved oxygen from around the province.

Data can be used by scientists to study coastal ecosystems, industry to inform site selection, government to inform management decisions, etc.

Was rarely used because it was stored in files that were difficult and time consuming to compile into a useful format.

Image
Photo taken by DD

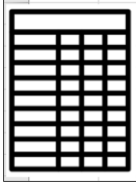


Sensor string: a rope that is anchored to the sea floor and suspended by a buoy near the surface. There are sensors attached at different depths that record data every 1 minute to 1 hour.

Each string is deployed in one location for several months.

Image
Designed by NLT and DD

	A	B	C
1	Plot Title: 10194899		
2	#	Date Time, GMT+00:00	Temp, °C
3	1	2018-02-20 18:00	1.751
4	2	2018-02-20 18:15	1.724
5	3	2018-02-20 18:30	1.724
6	4	2018-02-20 18:45	1.67
7	5	2018-02-20 19:00	1.67
8	6	2018-02-20 19:15	1.67
9	7	2018-02-20 19:30	1.643
		2018-02-20 19:45	1.615
		2018-02-20 20:00	1.615
		2018-02-20 20:15	1.615
		2018-02-20 20:30	1.588
		2018-02-20 20:45	1.561
		2018-02-20 21:00	1.561
		2018-02-20 21:15	1.561
		2018-02-20 21:30	1.588



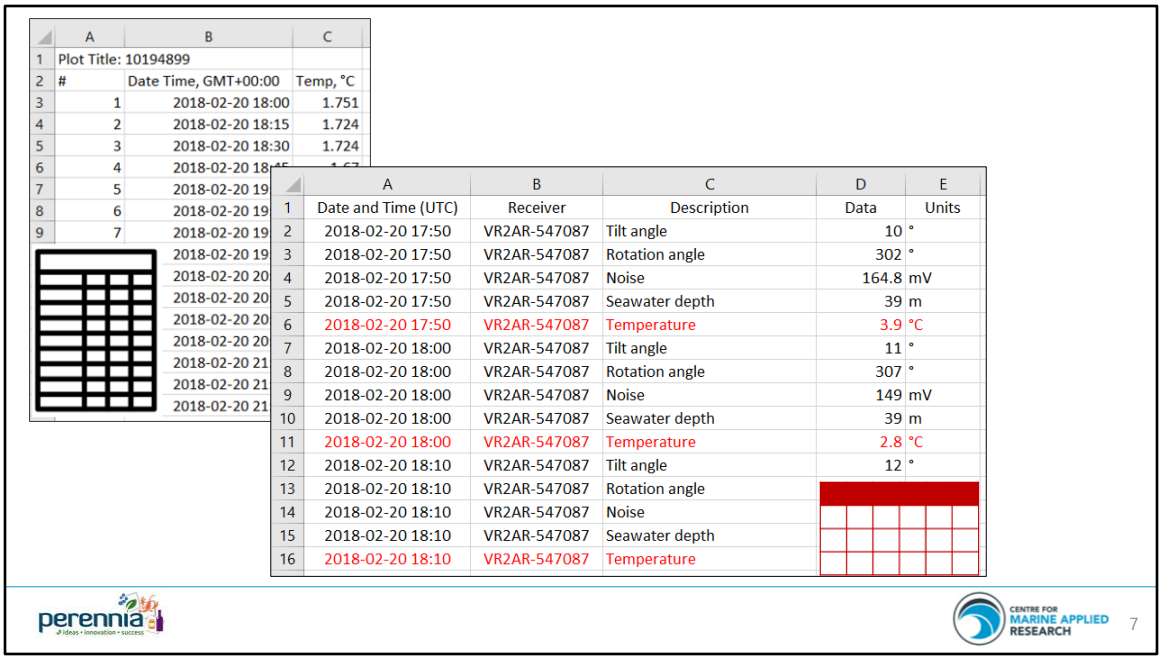
perennia ideas • innovation • success

CENTRE FOR MARINE APPLIED RESEARCH

6

Three different types of sensors that report the data in different formats.
3 to 7 files for each string; 10's of thousands of rows long.

Images
Black spreadsheet icon: https://icon-icons.com/download/127014/PNG/512/
Green spreadsheet icon: https://www.visualpharm.com/free-icons/set/spreadsheets
Red spreadsheet icon: designed by DD
Spreadsheet screen shots by DD

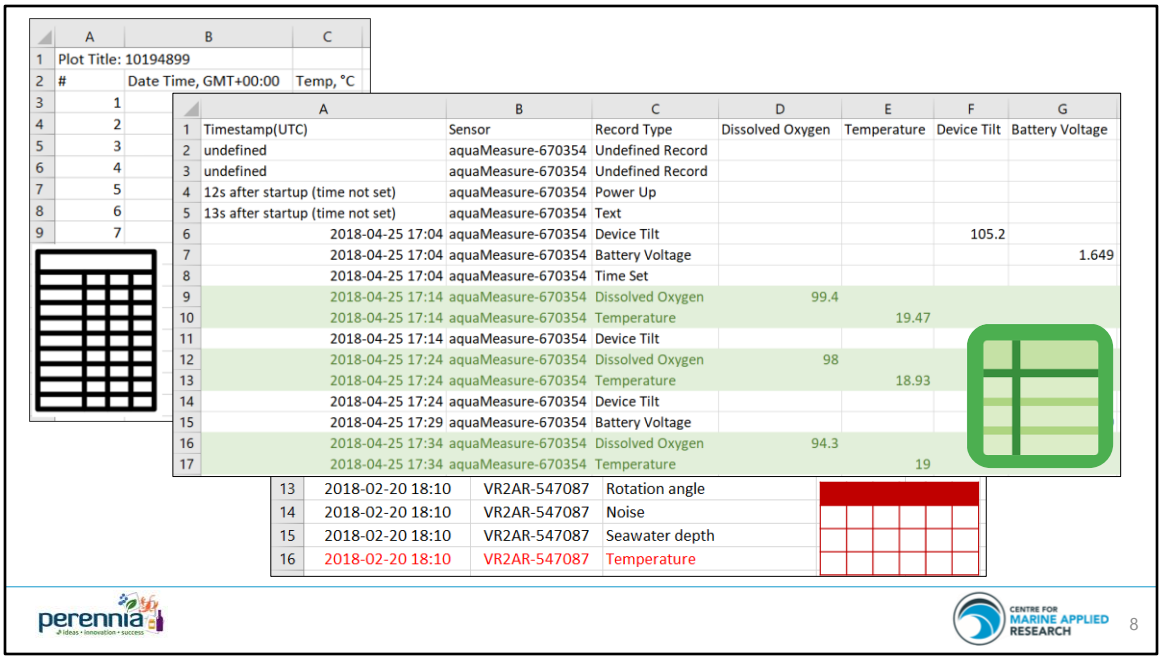


	A	B	C
1	Plot Title: 10194899		
2	#	Date Time, GMT+00:00	Temp, °C
3	1	2018-02-20 18:00	1.751
4	2	2018-02-20 18:15	1.724
5	3	2018-02-20 18:30	1.724
6	4	2018-02-20 18:45	
7	5	2018-02-20 19	
8	6	2018-02-20 19	
9	7	2018-02-20 19	
		2018-02-20 19	
		2018-02-20 20	
		2018-02-20 20	
		2018-02-20 20	
		2018-02-20 20	
		2018-02-20 21	
		2018-02-20 21	
		2018-02-20 21	

	A	B	C	D	E
1	Date and Time (UTC)	Receiver	Description	Data	Units
2	2018-02-20 17:50	VR2AR-547087	Tilt angle	10	°
3	2018-02-20 17:50	VR2AR-547087	Rotation angle	302	°
4	2018-02-20 17:50	VR2AR-547087	Noise	164.8	mV
5	2018-02-20 17:50	VR2AR-547087	Seawater depth	39	m
6	2018-02-20 17:50	VR2AR-547087	Temperature	3.9	°C
7	2018-02-20 18:00	VR2AR-547087	Tilt angle	11	°
8	2018-02-20 18:00	VR2AR-547087	Rotation angle	307	°
9	2018-02-20 18:00	VR2AR-547087	Noise	149	mV
10	2018-02-20 18:00	VR2AR-547087	Seawater depth	39	m
11	2018-02-20 18:00	VR2AR-547087	Temperature	2.8	°C
12	2018-02-20 18:10	VR2AR-547087	Tilt angle	12	°
13	2018-02-20 18:10	VR2AR-547087	Rotation angle		
14	2018-02-20 18:10	VR2AR-547087	Noise		
15	2018-02-20 18:10	VR2AR-547087	Seawater depth		
16	2018-02-20 18:10	VR2AR-547087	Temperature		

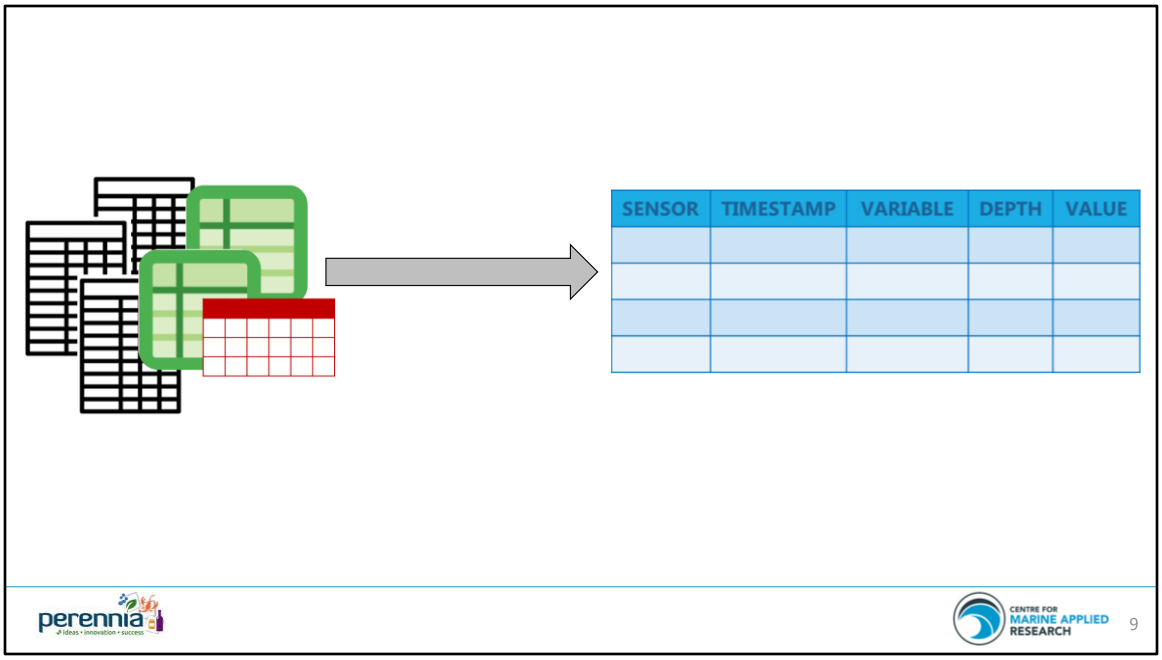
Three different types of sensors that report the data in different formats.
3 to 7 files for each string; 10's of thousands of rows long.

Images
Black spreadsheet icon: https://icon-icons.com/download/127014/PNG/512/
Green spreadsheet icon: https://www.visualpharm.com/free-icons/set/spreadsheets
Red spreadsheet icon: designed by DD
Spreadsheet screen shots by DD



Three different types of sensors that report the data in different formats.
3 to 7 files for each string; 10's of thousands of rows long.

Images
Black spreadsheet icon: https://icon-icons.com/download/127014/PNG/512/
Green spreadsheet icon: https://www.visualpharm.com/free-icons/set/spreadsheets
Red spreadsheet icon: designed by DD
Spreadsheet screen shots by DD

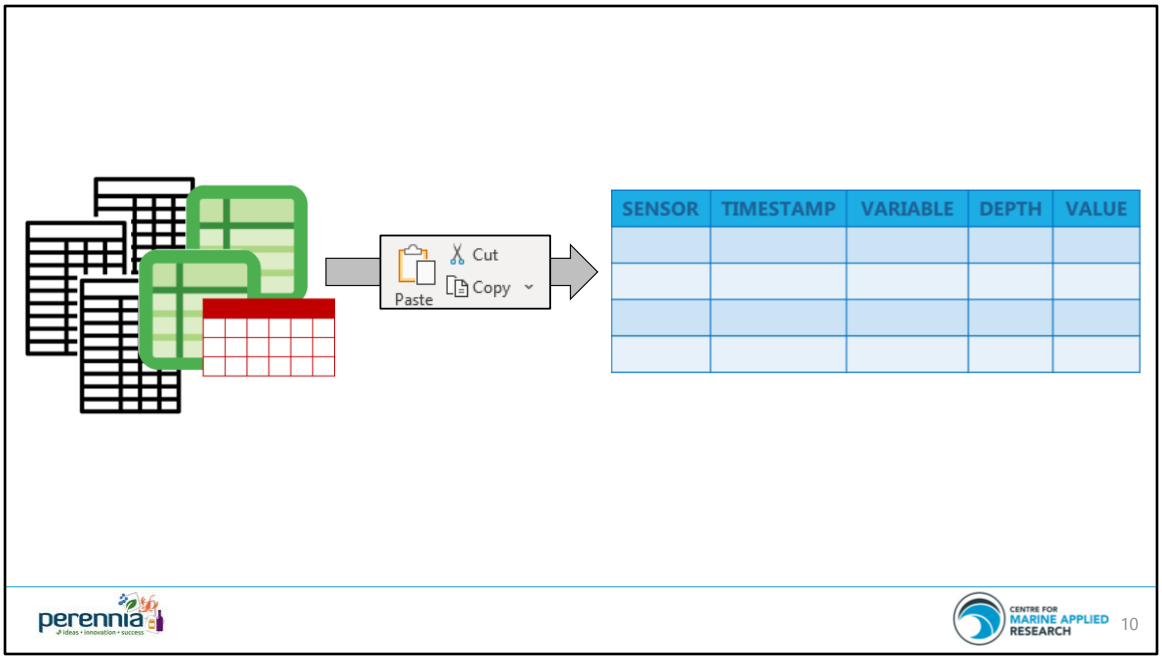


I was hired to compile data into a tidy format so we could put it online for any interested stakeholder to download and use for their own analyses.

Existing process: copy-paste in Excel, which took hours for each deployment.
150 + deployments to compile.

I wrote an R package called strings to help automate this process.

Images
Black spreadsheet icon: https://icon-icons.com/download/127014/PNG/512/
Green spreadsheet icon: https://www.visualpharm.com/free-icons/set/spreadsheets
Red spreadsheet icon: designed by DD
Blue spreadsheet icon: designed by DD
strings hex sticker: designed by DD
Copy/Paste: screenshot from Microsoft Excel

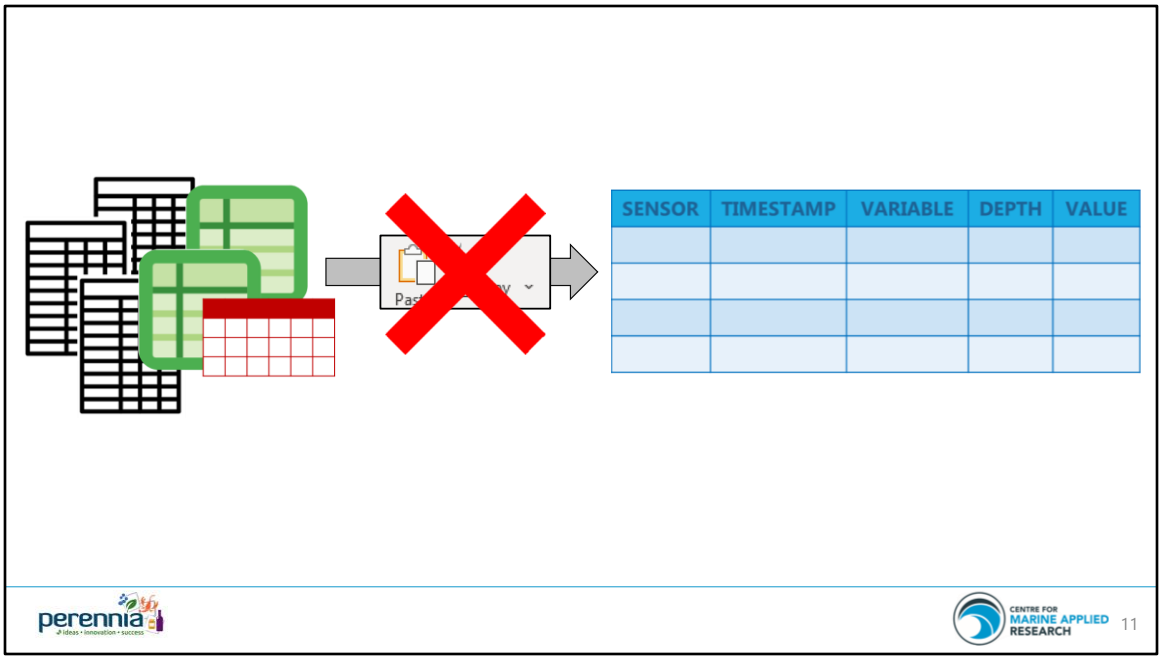


I was hired to compile data into a tidy format so we could put it online for any interested stakeholder to download and use for their own analyses.

Existing process: copy-paste in Excel, which took hours for each deployment.
150 + deployments to compile.

I wrote an R package called strings to help automate this process.

Images
Black spreadsheet icon: https://icon-icons.com/download/127014/PNG/512/
Green spreadsheet icon: https://www.visualpharm.com/free-icons/set/spreadsheets
Red spreadsheet icon: designed by DD
Blue spreadsheet icon: designed by DD
strings hex sticker: designed by DD
Copy/Paste: screenshot from Microsoft Excel

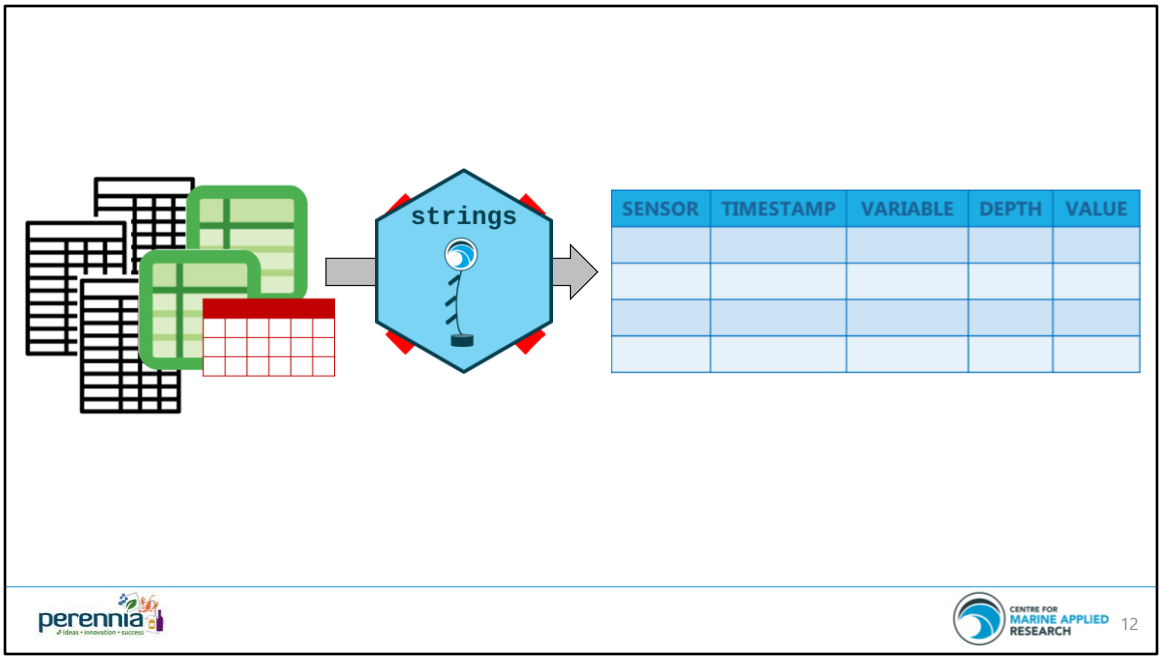


I was hired to compile data into a tidy format so we could put it online for any interested stakeholder to download and use for their own analyses.

Existing process: copy-paste in Excel, which took hours for each deployment.
150 + deployments to compile.

I wrote an R package called strings to help automate this process.

Images
Black spreadsheet icon: https://icon-icons.com/download/127014/PNG/512/
Green spreadsheet icon: https://www.visualpharm.com/free-icons/set/spreadsheets
Red spreadsheet icon: designed by DD
Blue spreadsheet icon: designed by DD
strings hex sticker: designed by DD
Copy/Paste: screenshot from Microsoft Excel



I was hired to compile data into a tidy format so we could put it online for any interested stakeholder to download and use for their own analyses.

Existing process: copy-paste in Excel, which took hours for each deployment.
150 + deployments to compile.

I wrote an R package called strings to help automate this process.

Images
Black spreadsheet icon: https://icon-icons.com/download/127014/PNG/512/
Green spreadsheet icon: https://www.visualpharm.com/free-icons/set/spreadsheets
Red spreadsheet icon: designed by DD
Blue spreadsheet icon: designed by DD
strings hex sticker: designed by DD
Copy/Paste: screenshot from Microsoft Excel

Writing code will SAVE YOU TIME.
May have to spend time up front to develop the code but it will be worth it.

Images
Background: image by Jan Vasek from Pixaby
Clock: made by DD

I spent a few weeks developing the backbone of the strings package.
I spent some time investigating different packages to help with these challenges.

These some of the packages that I ended up working with.

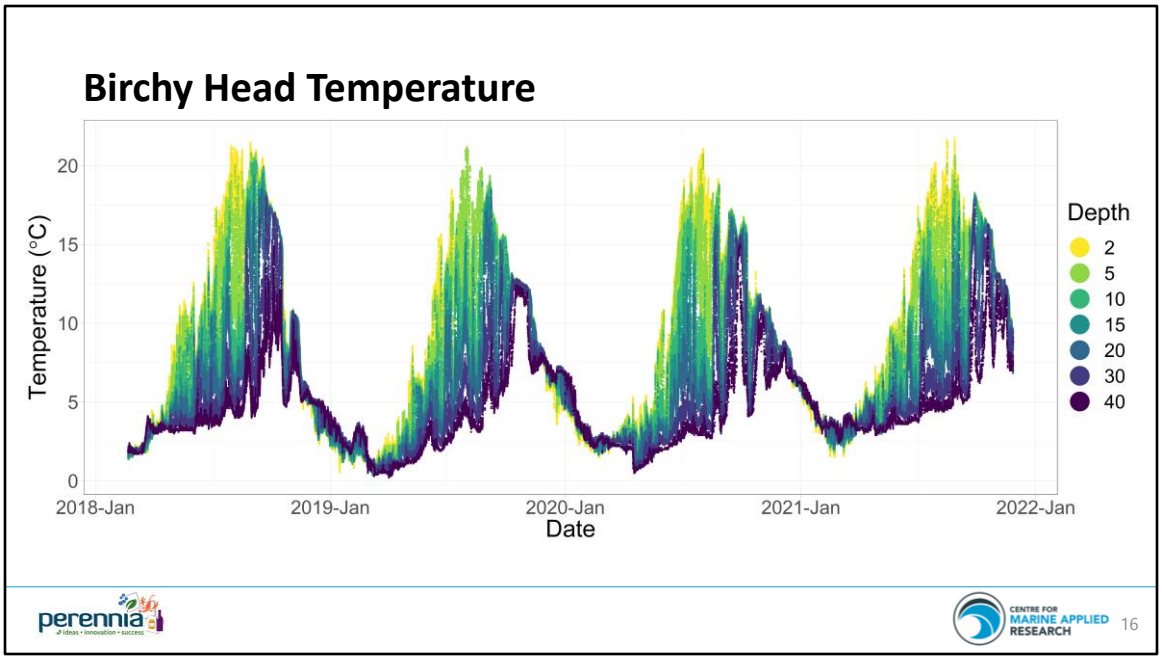
Many of these packages are now my go-to packages BECAUSE of the time I spent learning about them for the strings package.

Image
Compiled by DD
Tidyverse stickers : https://github.com/rstudio/hex-stickers
janitor sticker: https://github.com/sfirke/janitor
data.table sticker: https://github.com/Rdatatable/data.table

And time upfront HAS saved us time for processing the Coastal Monitoring Program data.

We calculated that following the old, horrendous copy-paste method, it would have taken TWO YEARS of work hours to process all of the data.

Image
Image by theTrueMikeBrown from Pixabay

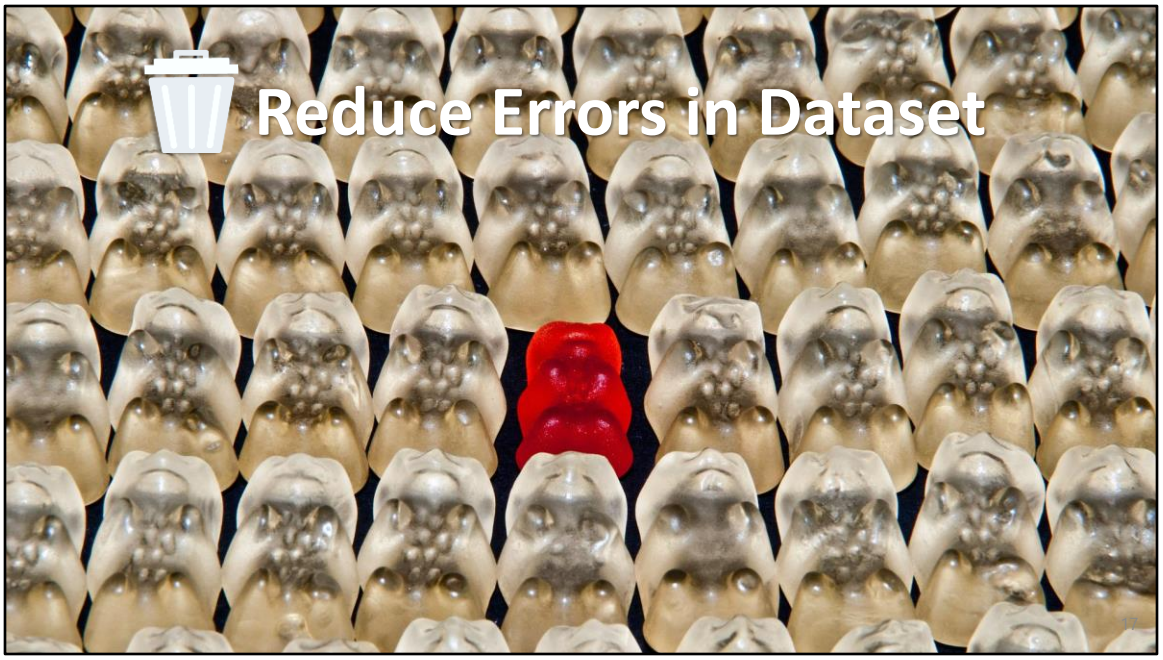


Within a few months, I had written the package and myself and our student at the time had compiled and formatted the data, put it online, and generated summary reports through RMarkdown.

Easy to train new people how to compile the data with the package using templates I've written.
Our current intern has compiled 71 deployments on top of his other work in the 11 weeks he's been with us. With the old method, this would have taken about 9.5 months of work hours!

Temperature data from 6 deployments – about 4 years – at Birchy Head

Image
Generated from CMAR Coastal Monitoring Program data by DD

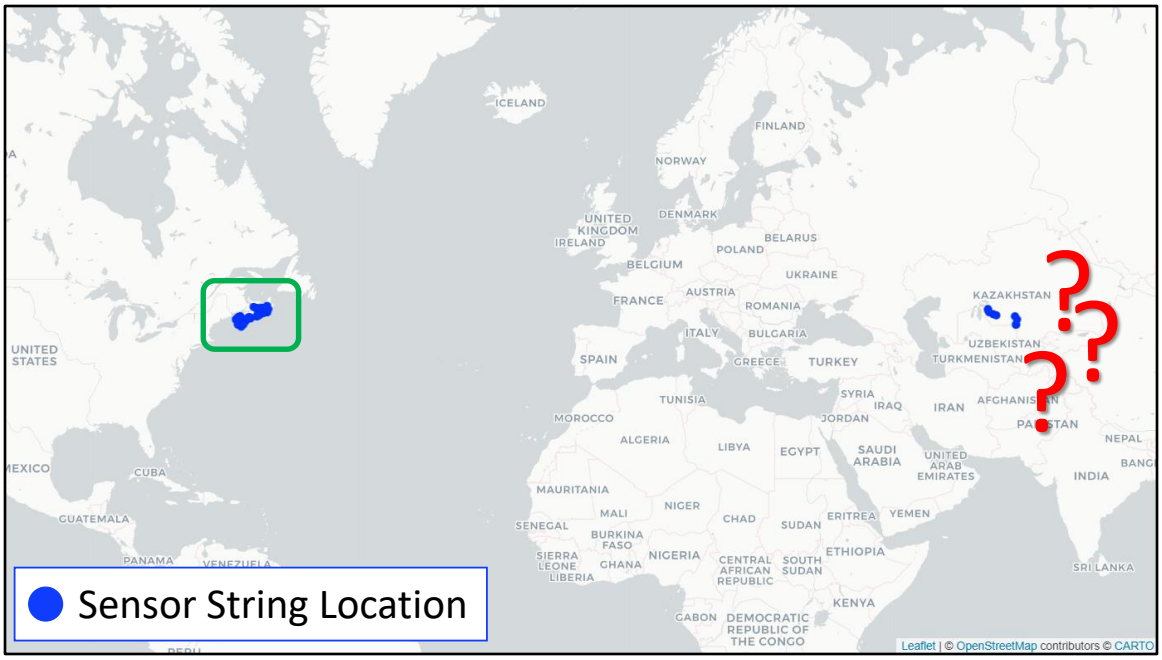


Writing code to automate data wrangling can help find and reduce errors in your dataset. Will give you and your clients more confidence in the dataset and improve the reliability of your data analysis.

Errors can creep in at different parts of the data pipeline, especially when humans are involved.

I've learned to write code to anticipate these errors.

Images
Background: image by Ronile from Pixabay
Garbage Can: modified from image designed by Freepik.com



For example, after I processed the strings data, I plotted the lat/long of all the deployments to get a sense of our spatial coverage.

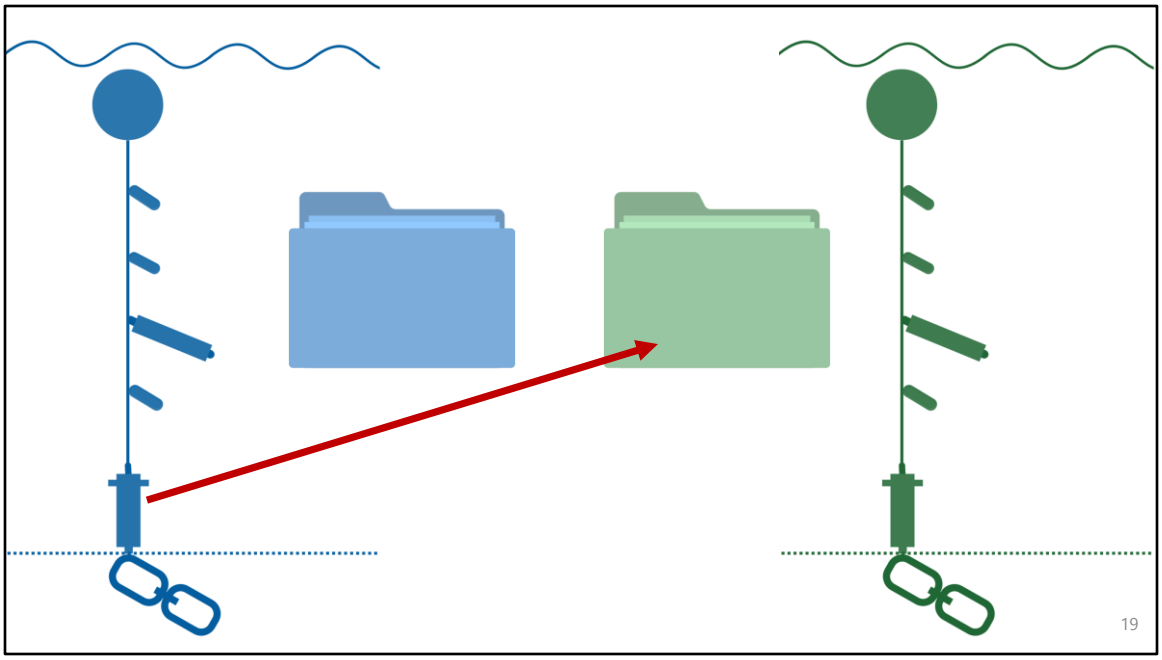
Most of the sensor locations were clustered around Nova Scotia, as expected.
But there were a few deployments off in Khazakstan!

After a little bit of digging, I realized that our contractors who deploy the sensor strings manually record the latitude and longitude of where the sensor string was deployed.

We live in the Western Hemisphere, so by convention the longitude is measured as negative degrees, but every now and then it gets entered as a positive value.

So now, when the package first reads in coordinate data, there will be an Error if any of the Longitudes are positive, so we can go in and fix them before the deployment is compiled.

Images
Leaflet map made by DD



Another error that I have seen is raw data from the wrong string in a folder of data ready to be compiled.

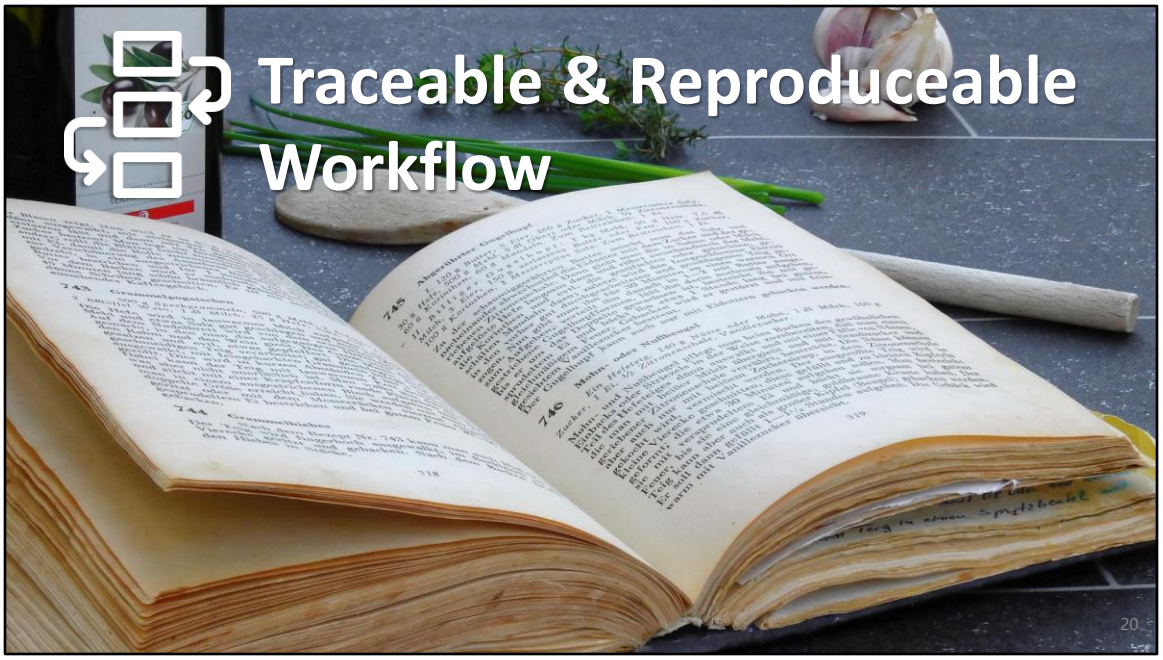
Here raw data from one of the sensors on the blue string was put into the folder with data for the green string.

The dataset for the blue string will be incomplete, and there will be false information about the area where the green string was deployed.

Raw files are automatically named with the sensor serial number, and we keep track of the serial numbers that are on each string in deployment logs.

So as part of the package, I wrote code that will give an error if there are any missing or extra files compared to what was recorded in the log.

Images
Sensor sting: designed by NLT and DD
Folder: modified from image on Flaticon.com



Writing code for your repetitive data-wrangling task can help make your workflow traceable and reproduceable, especially compared to compared to copying and pasting

Image
Background: image by Bru-nO from Pixabay
Workflow icon: modified from image on Flaticon.com

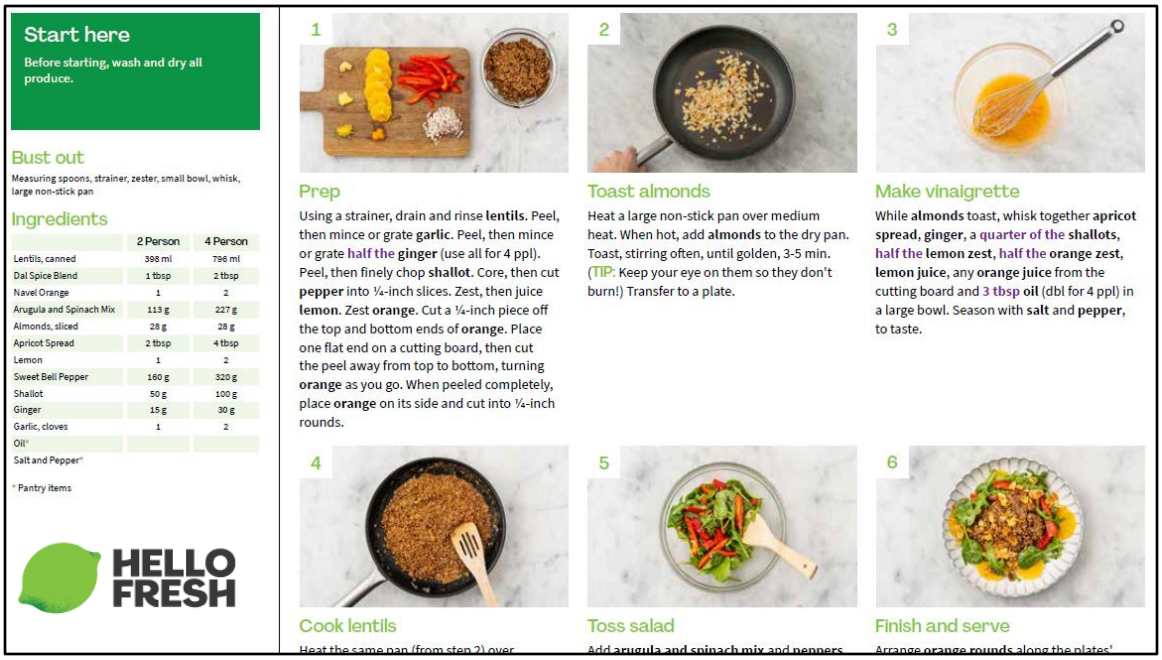
Copying and pasting in Excel is like making a salad from whatever is left in your fridge.

Most of the time it will probably be fine, but eventually you are going to get something that doesn't taste quite right.

But you never 100 % sure what went in it, and there is no way to go back and tell where you went wrong.

You can't take the dressing back out of the salad.

Image
Photo by Ello on Unsplash



Start here

Before starting, wash and dry all produce.

Bust out

Measuring spoons, strainer, zester, small bowl, whisk, large non-stick pan

Ingredients

	2 Person	4 Person
Lentils, canned	398 ml	796 ml
Dal Spice Blend	1 tbsp	2 tbsp
Navel Orange	1	2
Arugula and Spinach Mix	113 g	227 g
Almonds, sliced	28 g	28 g
Apricot Spread	2 tbsp	4 tbsp
Lemon	1	2
Sweet Bell Pepper	160 g	320 g
Shallot	50 g	100 g
Ginger	15 g	30 g
Garlic, cloves	1	2
Oil*		
Salt and Pepper*		

* Pantry items

HELLO FRESH

1

Prep

Using a strainer, drain and rinse **lentils**. Peel, then mince or grate **garlic**. Peel, then mince or grate **half the ginger** (use all for 4 ppl). Peel, then finely chop **shallot**. Core, then cut **pepper** into ¼-inch slices. Zest, then juice **lemon**. Zest **orange**. Cut a ¼-inch piece off the top and bottom ends of **orange**. Place one flat end on a cutting board, then cut the peel away from top to bottom, turning **orange** as you go. When peeled completely, place **orange** on its side and cut into ¼-inch rounds.

2

Toast almonds

Heat a large non-stick pan over medium heat. When hot, add **almonds** to the dry pan. Toast, stirring often, until golden, 3-5 min. (TIP: Keep your eye on them so they don't burn!) Transfer to a plate.

3

Make vinaigrette

While **almonds** toast, whisk together **apricot spread**, **ginger**, a **quarter of the shallots**, **half the lemon zest**, **half the orange zest**, **lemon juice**, any **orange juice** from the cutting board and **3 tbsp oil** (dbl for 4 ppl) in a large bowl. Season with **salt** and **pepper**, to taste.

4

Cook lentils

Heat the same pan (from step 2) over

5

Toss salad

Add **arugula and spinach mix** and **peppers**

6

Finish and serve

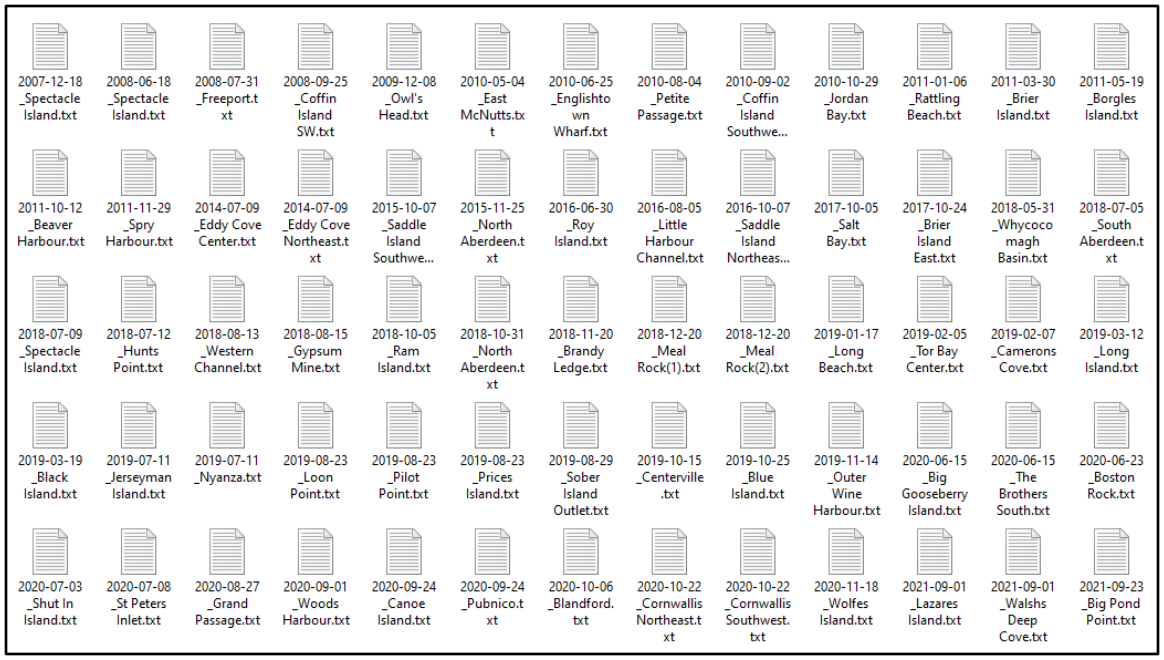
Arrange **orange rounds** along the plates'

Writing code to automate the data wrangling is more like writing and following a recipe.

The steps are written out, so you can follow it to make a delicious salad again and again.

Easy to share with a friend.

Image
Hello Fresh recipe card



For another part of our Coastal Monitoring Program, we had 65 current datasets that needed to be compiled from these text files into tidy datasets.

Manager decided to have a few people manually copy, paste, and format in Excel. It took hours and hours of several people's time.

Since learned that there was a mistake in how the original text files were exported, so all of the data would have to be re-compiled.

This time I was asked to write a package, and did so within a few weeks.
NOW, if we need to modify the process again, we just need to tweak the code, press a button, and we can re-generate all of the processed data in a few minutes.

Image
Screenshot by DD

If you have a repetitive task that you have been thinking about automating, I encourage you to go home after the conference and start writing some code!

You don't have to write a package in a day – start with a function or two, and see where things go from there.

If you get some pushback from your R-skeptic, remind them that writing code will save time, catch and reduce errors, and result in a more traceable and reproducible workflow.

Images
Clock: made by DD
Garbage Can: modified from image designed by Freepik.com
Workflow: modified from image on Flaticon.com

Finally, I will say that my previously skeptical manager now is totally on board with my packages.

She even sewed me this “Super R”, and says that I’ve convinced her that R can do anything we put our minds to.

Image
Photo taken by DD