Use case publishing guide

A use case is generally created in the form of a jupyter notebook1. It shows how to use the City of Melbourne's Open Data to solve realistic examples of real-world problems. They should inspire, educate, and help to encourage uptake of the open data.

However, it can't inspire or educate anyone if they never even see it – so your use case needs to get published!

Here are the use cases which Chameleon has had published previously (although that link is expected to be changed by the Web Dev team to something which is more user friendly).

Publishing a use case

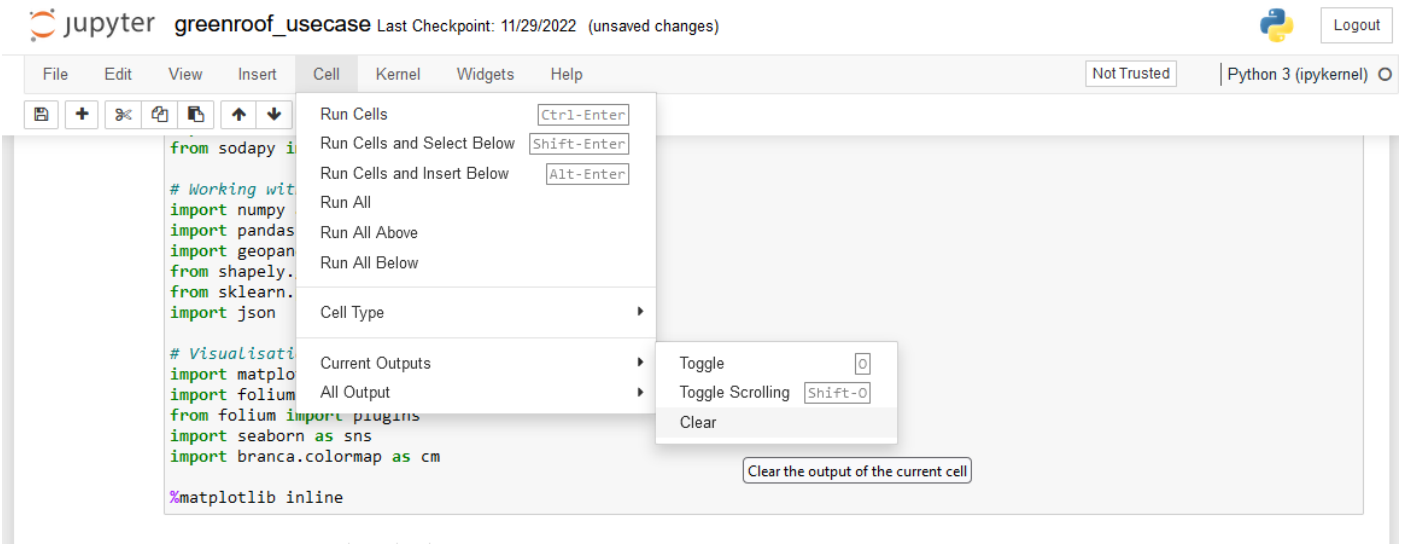
The process of getting a use case published goes like this:

1) Create your use case. (I believe this step is fairly obvious…)

2) Have it peer reviewed. A peer review process has recently been developed. Ask your team leader if you need help with this.

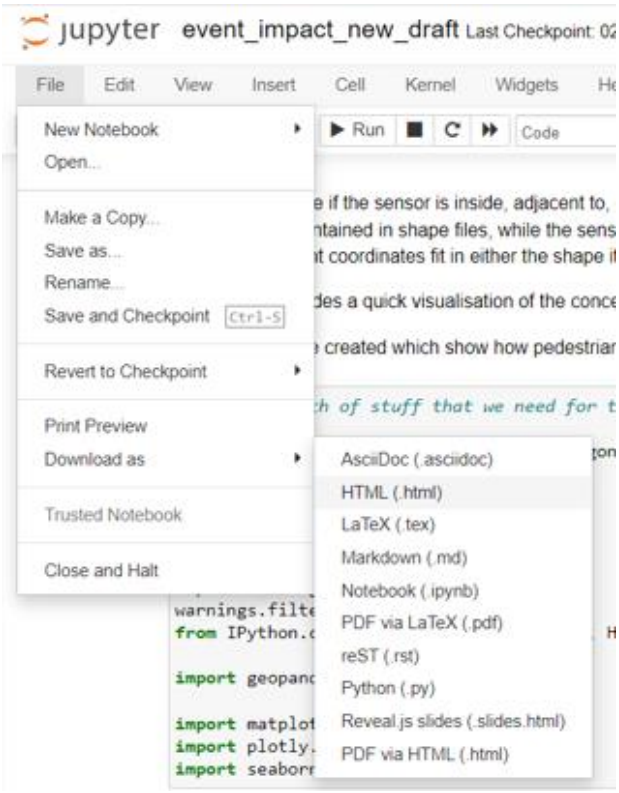
3) Put it into the template provided on Github, following the style guide in the notebook. A reference for the style guide is also included in this document.

4) If there are any output cells you don't wish to include in the published version, clear them before exporting. This might include warnings, long outputs, anything you think could clutter up your use case.



[1] If you want to create your use case in a different format, propose this to your team leader. They will always be interested in cool ideas.

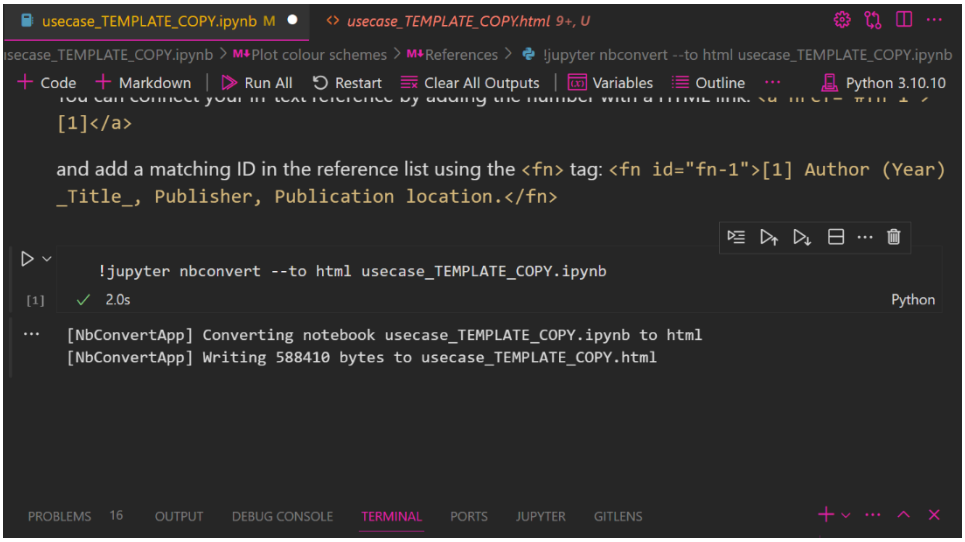
5) Export your notebook as a HTML file using jupyter:



Export your .ipynb file as a HTML file using VS code:

1. Install nbconvert
2. And install nbconvert dependency modules
   [Installation — nbconvert 7.12.0 documentation]
3. Add a new cell at the end of your use case and use the code:
   !jupyter nbconvert --to html <usecase<name>.ipynb

When you run this code, a new html file of your use case will be made.



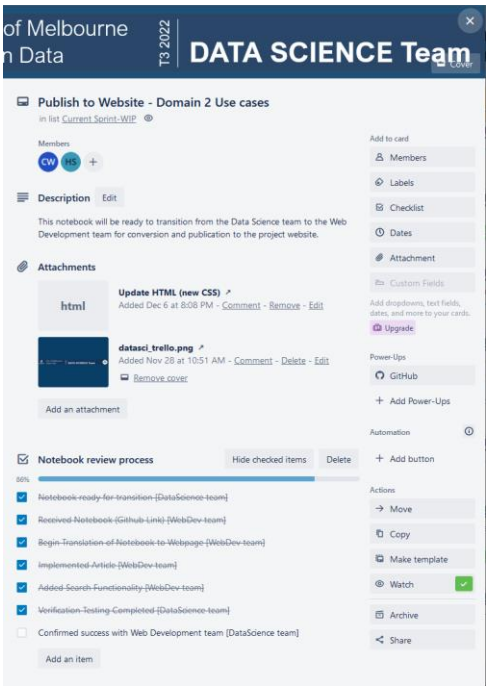
([Explanation if you use VS Code without .ipynb extension)](#)

1) Send the web dev team the following details for your use case to create the search metadata:
    - title
    - file name or a link to the html you exported
    - description
    - tags
    - difficulty
    - technology (name of main packages used)

You can do this using a .json file or txt. The below screenshot shows you an example of what this will look like. Reach out to the Web Dev team for support with this. They are the experts on this sort of thing!

```
{
    "title": "Melbourne Bicycle Network Routes and Road Safety Part 1",
    "name": "bicyclenetworkroadsafety-part1",
    "description": "As a cyclist, I want a safe transport journey in the city of Melbourne. Which
roads are safest to cycle on? As a council, we seek to invest in road safety initiatives which reduce the
occurrences of accidents resulting in serious injuries of citizens using our road network. Where are
accident hotspots for cyclists occurring?",
    "tags": ["GeoJSON", "bicycles", "road safety", "route network", "accidents"],
    "difficulty": "Intermediate",
    "technology": [{
        "name": "numpy",
        "code": "python"
    },{
        "name": "pandas",
        "code": "python"
    },{
        "name": "sodapy",
        "code": "python"
    },{
        "name": "folium",
        "code": "python"
    }]
}
```

2) Submit the html and the json to the Web Dev team using a Trello card – make sure to add the web dev team leader to the card, as well as links to your html file and the metadata json. Example:



3) Sit back and enjoy the glory of having your use case published for the world to see.

Style guide for use cases

Headers

For styling within your markdown cells, there are two choices you can use for headers.

1) You can use HTML classes specific to the use case styling:

```
<p class="usecase-subsection-header">This is a subsection header.</p>
<p class="usecase-subsection-blurb">This is a blurb header.</p>
```

2) Or if you like you can use the markdown header styles:

```
# for h1
## for h2
### for h3
#### for h4
##### for h5
```

Plot colour schemes

General advice:

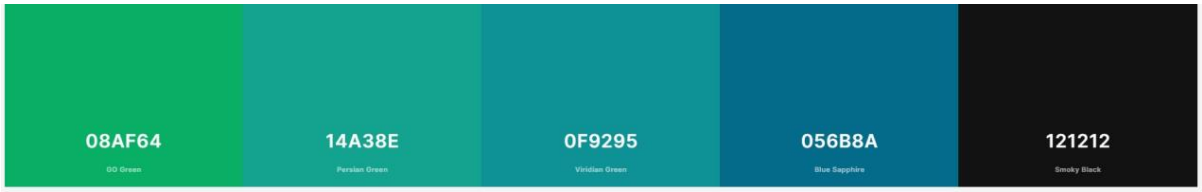
1. Use the same colour or colour palette throughout your notebook, unless variety is necessary
2. Select a palette based on the type of data being represented
3. Consider accessibility (colourblindness, low vision)

If all of your plots only use 1-2 colors use one of the company style colors:

Light theme

2af598 22e4ac 1bd7bb 14c9cb 0fbed8 08b3e5

Dark theme



08af64 14a38e 0f9295 056b8a 121212

2) If your plot needs multiple colors, choose an appropriate palette using either of the following tutorials:

- https://seaborn.pydata.org/tutorial/color_palettes.html
- https://matplotlib.org/stable/tutorials/colors/colormaps.html

3) Consider accessibility as well.

For qualitative plotting Seaborn's 'colorblind' palette is recommended. For maps with sequential or diverging it is recommended to use one of the Color Brewer schemes which can be previewed at https://colorbrewer2.org/ .

If you want to design your own colour scheme, it should use the same principles as Cynthia Brewer's research (with variation not only in hue but also, saturation or luminance).

References

Be sure to acknowledge your sources and any attributions using links or a reference list.

If you have quite a few references, you might wish to have a dedicated section for references at the end of your document, linked using footnote style numbers.

You can connect your in-text reference by adding the number with a HTML link:

```
<a href="#fn-1">[1]</a>
```

and add a matching ID in the reference list using the `<fn>` tag:

```
<fn id="fn-1">[1] Author (Year) _Title_, Publisher, Publication location.</fn>
```

Author

Brendan Richards (first draft)

Hannah Smith (added style guide and some details to publishing guide)

Alison Collins (added export .ipynb in VS code)