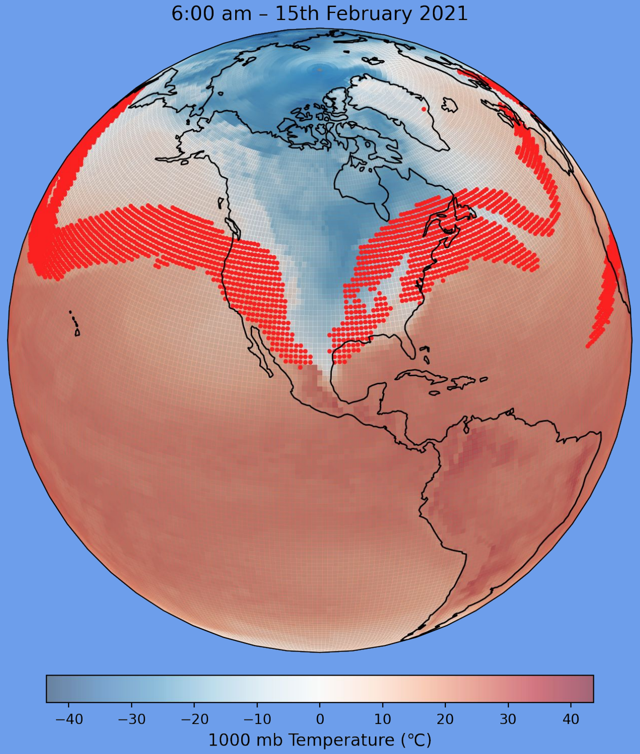
Approaching meaningful insight into the climate system. My experiences so far.



UCL
KING'S College LONDON

Tom Keel

About me

Sept 2015 to June 2018 – Geography & Geocomputation (KCL GEOG)

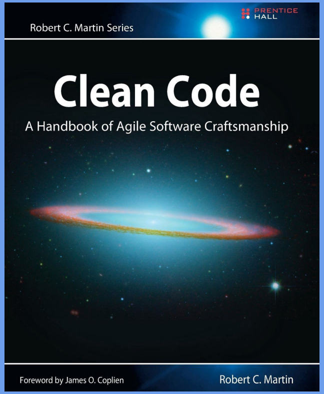
Sept 2018 to Sept 2019 – Spatial Data Science and Visualisation (UCL CASA)

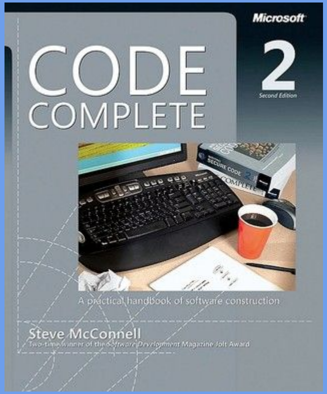
Sept 2019 to Sept 2020 – *Bits and Bobs*

Sept 2020 to right now – London NERC DTP PhD (UCL GEOG)

Experience:

Data scientist, GIS software developer.





About this presentation

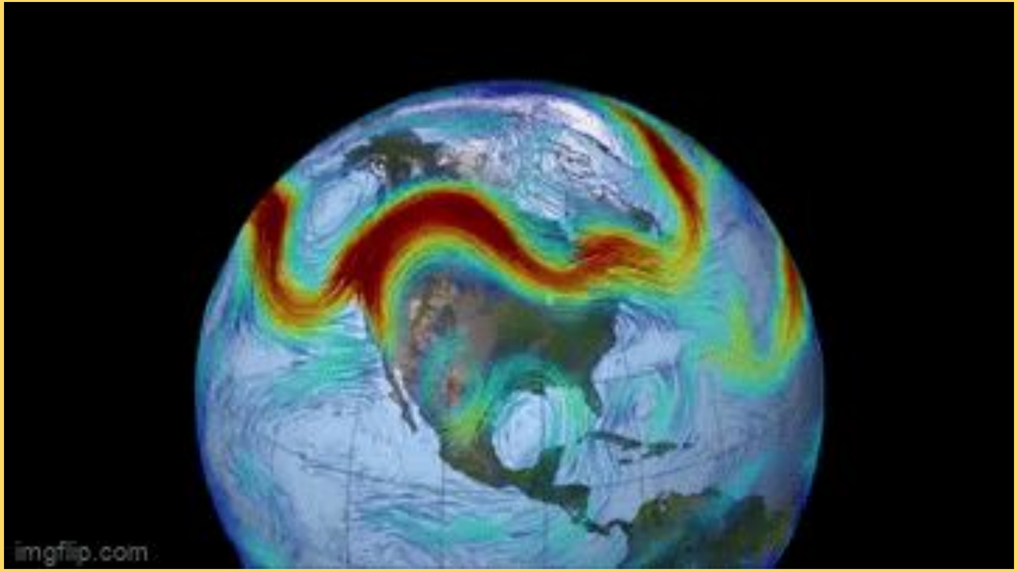
- ❏ About my research
- ❏ About climate
- ❏ About meaningful insight
- ❏ My approach to my project
- ❏ Role of open-source
- ❏ Summary of key points

About my research

Title: A shifting jet-stream in a changing climate: Exploring the response of the polar jet-stream in the Northern Hemisphere to various climate futures.

What are jet-streams?

- **Fast and fluid:** Streams of fast wind which occur in regions known as jet streaks
- **High and broad:** 8-12 km in between the troposphere and stratosphere
- **Complex and unknowable?**



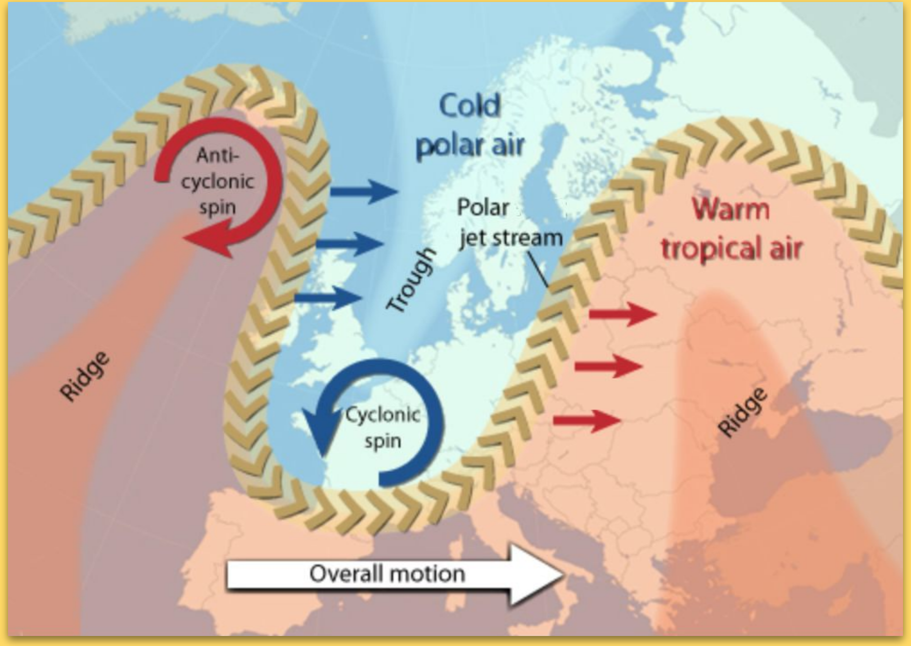
Jet-streams – Why are they important?

Dynamical properties:

- Form at the boundaries between “air-masses”
- Create cyclones and anticyclones at surface

Jet-streams as patterns:

- Transport moisture and heat across latitudes
- Important proxy for location of weather systems at any instant.
- [Let's look!]



Jet-stream – Link to weather

Storm tracks!

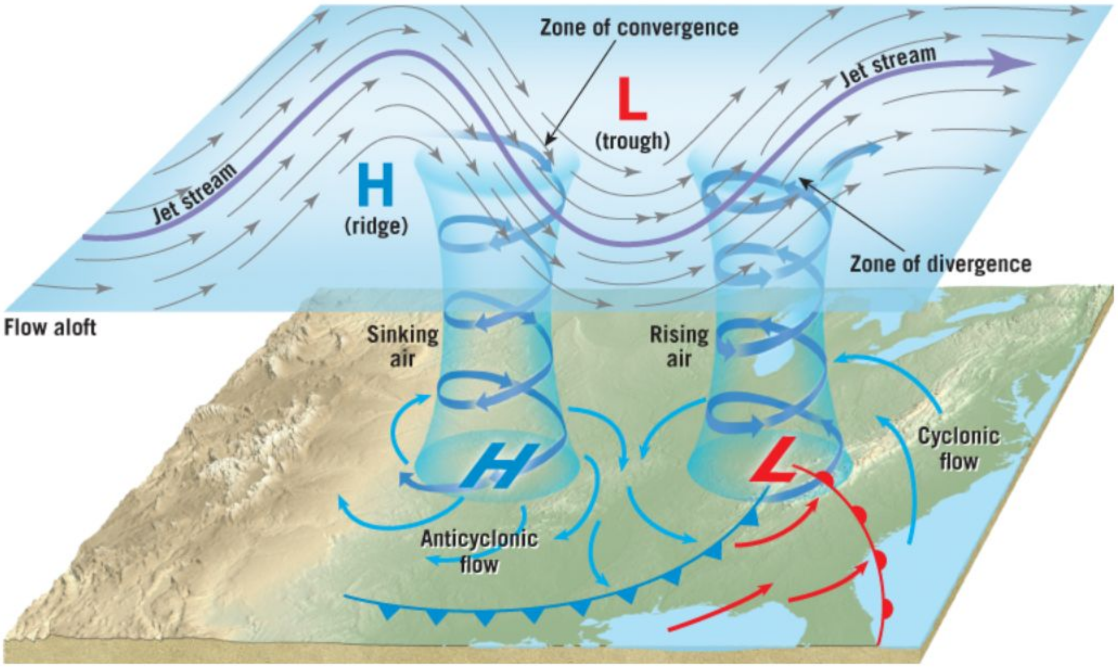


Figure 9.14 Idealized view of divergence and convergence aloft that supports cyclonic and anticyclonic circulation at the surface

Jet-streams – Impacts

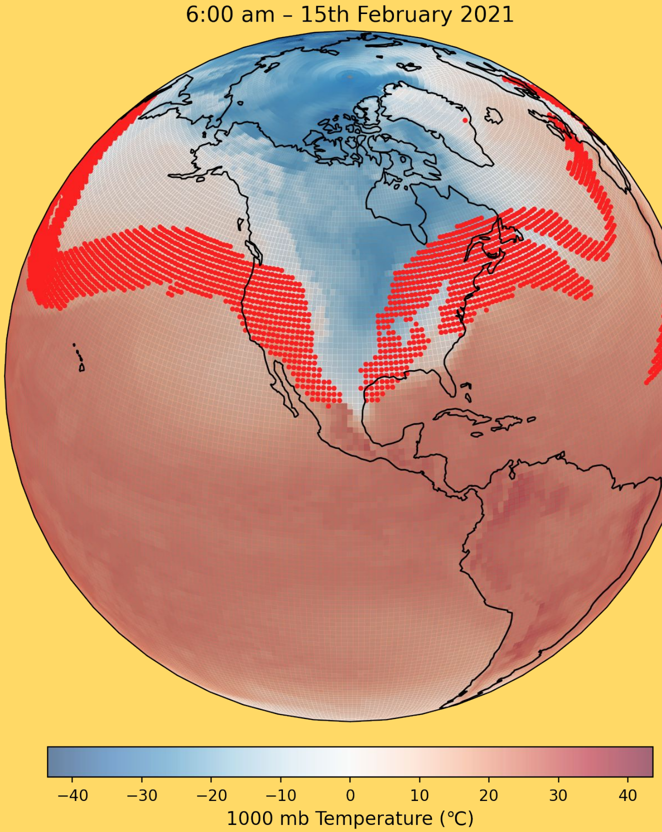
Cold-air transport:

- Texas snowstorm February 2021
- Beast from the East(s)

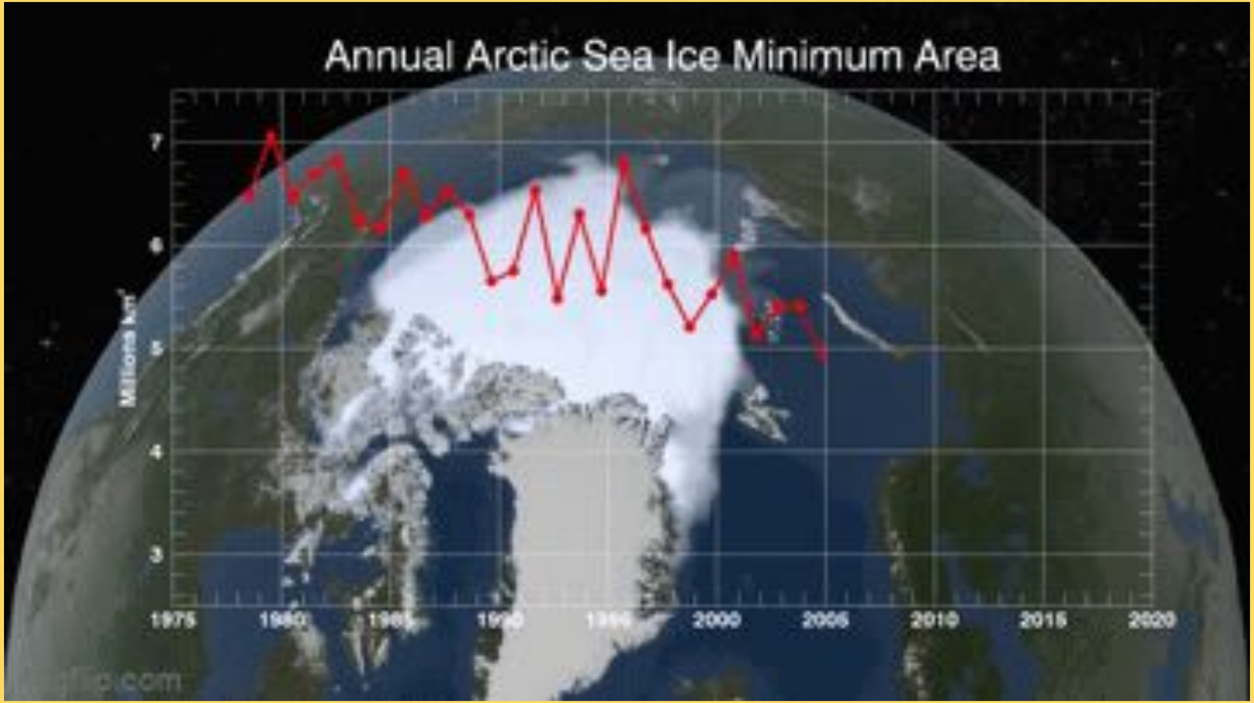
Persistence:

- European heatwaves (2003, 2010, 2021)
- Droughts

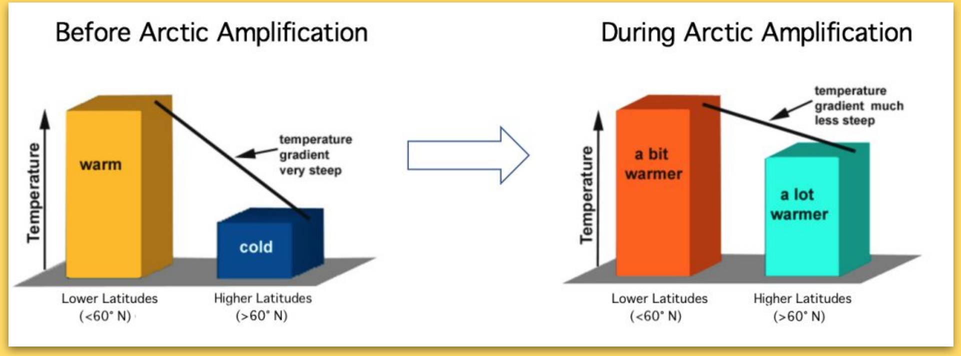
Note: Northern Hemisphere!

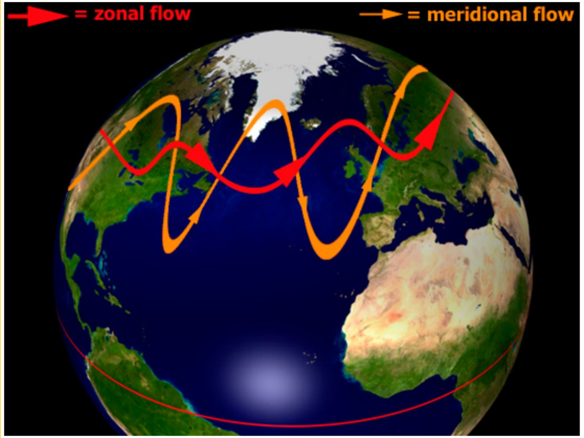


Jet-streams – Changes



Jet-streams – Changes





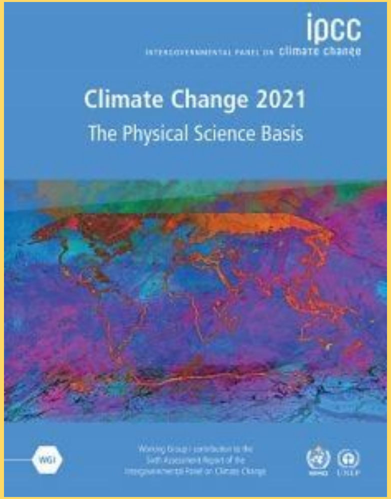
Problems:

- Trends dependent on start–end year (Blackport & Screen, 2020)
- Trends dependent on metrics used (Cohen et al. 2020)

Jet-streams – IPCC AR6

“The extratropical jets and cyclone tracks have likely been shifting poleward in both hemispheres since the 1980s with marked seasonality in trends (medium confidence)”. (IPCC AR6 2.3.1.4.3)

“There is low confidence in projected poleward shifts of the Northern Hemisphere mid-latitude jet and storm tracks due to large internal variability and structural uncertainty in model simulations”. (IPCC AR6 TS-38)



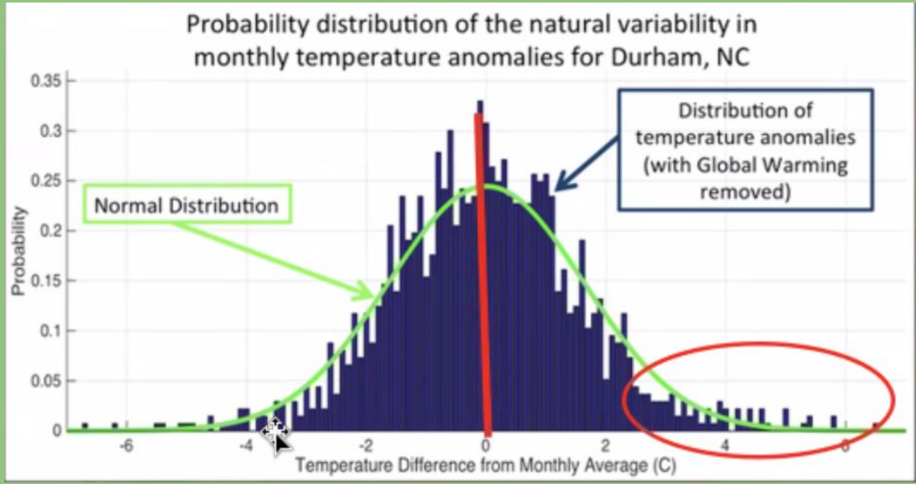
What is change in climate?

What is weather?:

- Short-term variations in atmospheric variables.

What is climate?:

- An average of weather conditions over a particular region.



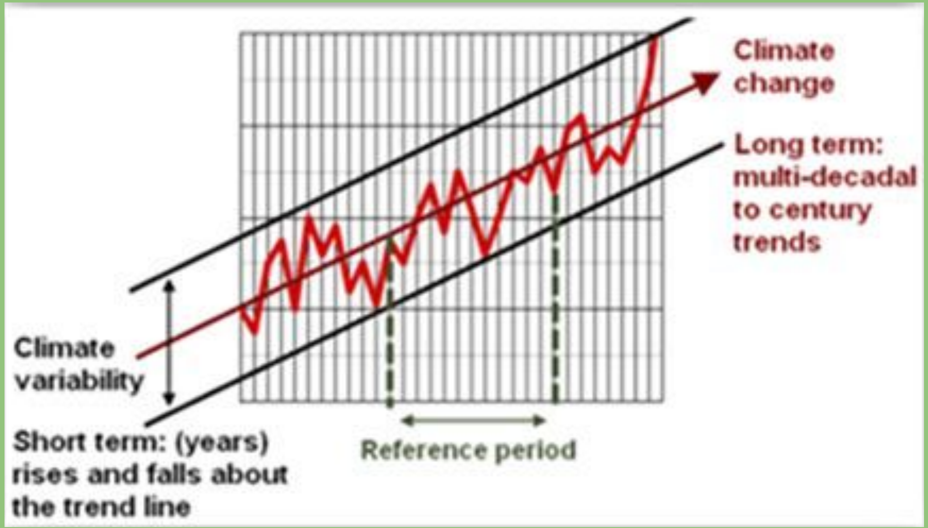
Obstacles for characterising change:

- **Space-time continuity:** no separation between <u>scales</u>.
- **Interactions:** between different parts of climate system.

What is variability in climate?

Variability:

- Deviations around a mean or trendline in a reference period and area.
- **Non-stationarity**: We are already living in an era of record-breaking weather conditions in the Northern Hemisphere.



Cool part about the climate problem:

- Future projection!
- Magnitude of variation about a trend line may be changing.

What are climate phenomena?

Phenomenon: An occurrence, circumstance, or fact that is perceptible by the senses.

Climate phenomenon: An observable event perceptible in data.

Examples of climate phenomena:

- Monsoons
- Beast from the East
- El Nino, La Nina

What makes them real?

What are climate metrics?

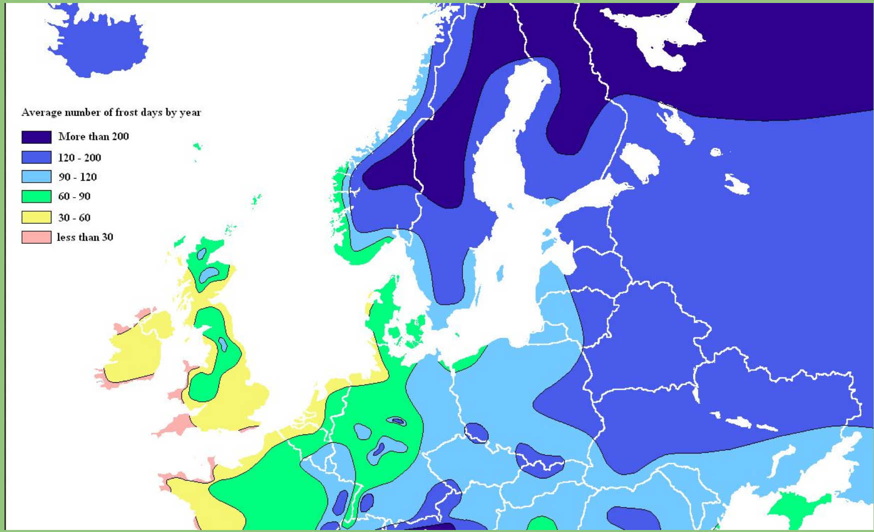
Metrics: A set of numbers that give you information about a particular process or activity.

Climate metrics: indices, statistics and algorithms used to isolate and characterise a given climate phenomena over a given time period and location.

Examples: days with temperatures above 35°C; frost-days; ENSO index

Problems:

- You can show anything with metrics *“all metrics are wrong, but some are useful”*



Gaining meaningful insight

It is a research *community* working together that can get through to ‘meaningful’ insight (*hint: open-source*).

Problems for my research area (jet-streams):

- Jet-streams show signal in various measured variables.
- BUT: We cannot reliably collect observations about them
- AND: We have multiple sources of information about them (past & future).

Sources of information (i.e. what data can we use)

Climate Reanalysis:

- **Data assimilation strategy:** combination of observations, climate models and interpolation
- Uses various filters Kalman & Particle filters

Climate models:

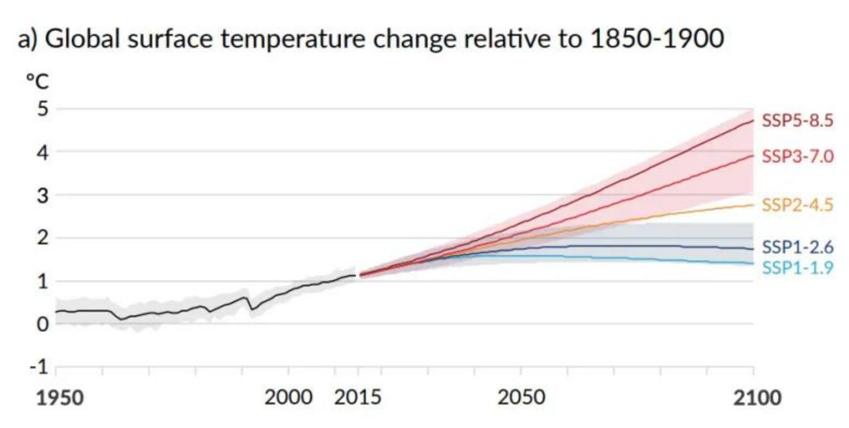
- **Code that solves equations** based on our understanding of the climate system and newtonian laws of physics
- Coupled Model Intercomparison Project 6 (CMIP6) has 33 modelling groups in 16 countries

Do we (always) have best opportunity now?

- Higher resolution, most modelling groups

Reading between the lines

We never use one model, one scenario, one metric *INSTEAD* we use an ensemble and read between the lines.



Finding trends:

- **Model agreement:** Probability of X occurring given the ensemble.
- Visualisation is our story-telling device

My process

To approach solving problem we first need to define the bounds of the problem:

- What is the jet? What is a change in the jet?
- Which space-time context is most useful for understanding a jet?

To approach the software required, we first need to define what the community might find useful:

Philosophy A: to create a solution using the least amount of components.

Philosophy B: to create a solution that is decoupled.

Philosophy C: take a reductive view of the problem we are trying solve (it is just data at the end of the day)

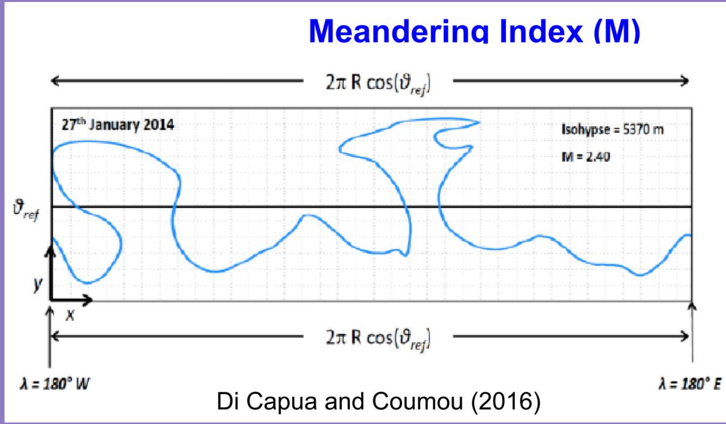
First steps: Look at literature

Context from literature

30 metrics found!

Storylines from literature:

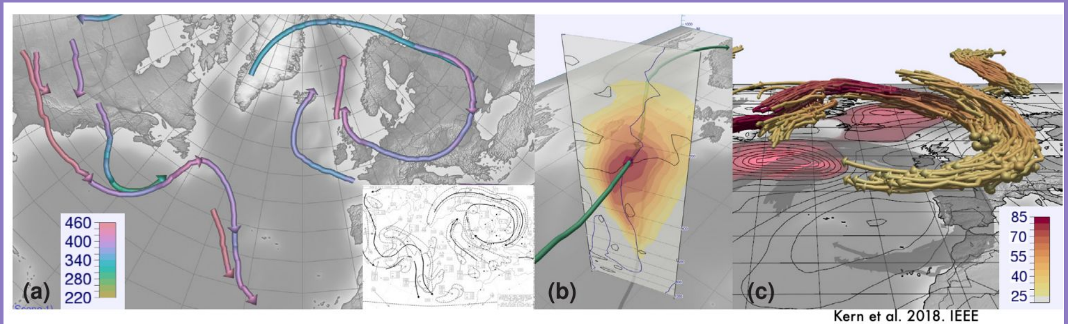
1. *Mean* latitude.
2. Waviness.
3. Preferred positions.



Di Capua and Coumou (2016)

Contexts of understanding:

1. Jet as continuous.
2. Jet as segmented.
3. Jet as emergent.



Kern et al. 2018. IEEE

Example metric – Woolings et al. 2010

A metric for diagnosing jet-stream latitude over the North Atlantic.

Reductive view:

- Computationally constrained?
- Enough information?
- Which context does this metric make sense in?

2. Diagnosing jet latitude and speed

The latitude and speed of the eddy-driven jet stream is identified in daily ERA-40 wind data over the period 1 December 1957–28 February 2002. This provides 45 complete winter seasons (DJF) of data but only 44 complete seasons in spring (MAM), summer (JJA) and autumn (SON). The algorithm proceeds as follows:

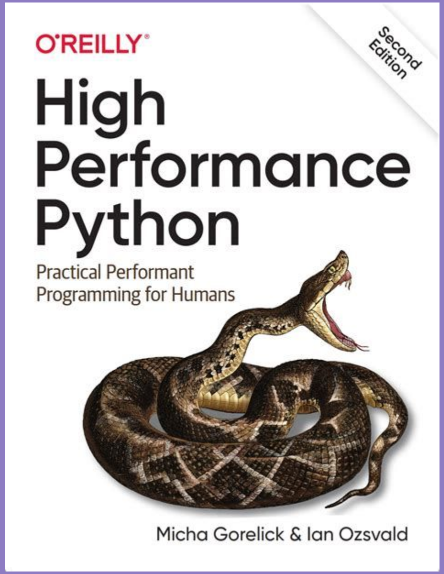
1. The daily mean zonal wind is averaged over the levels 925, 850, 775 and 700 hPa.
2. The resulting field is then zonally averaged over a longitudinal sector (0–60°W for the North Atlantic), neglecting winds poleward of 75° and equatorward of 15°.
3. The resulting field is then low-pass filtered to remove the features associated with individual synoptic systems. This is done using a 10 day Lanczos filter with a window of 61 days (Duchon, 1979).
4. The maximum westerly wind speed of the resulting profile is then identified and this is defined as the jet speed. The jet latitude is defined as the latitude at which this maximum is found.
5. Smooth seasonal cycles of the jet latitude and speed are defined by averaging over all years and then Fourier filtering, retaining only the mean and the two lowest frequencies. The jet latitude and speed as presented here are anomalies from the seasonal cycle.

Woollings, T., Hannachi, A., & Hoskins, B. (2010). *Variability of the North Atlantic eddy-driven jet stream. April*, 856–868. https://doi.org/10.1002/qj.625

Step 1. Explore in Jupyter notebooks

- Explore solutions
- Check outputs

```
[18]: mean_data = data.sel(plev=slice(92500, 75000))
      mean_data = mean_data.mean(['lon','plev'])

[19]: # TODO: add to code
      if mean_data.lat[0] > mean_data.lat[-1]:
          mean_data = mean_data.reindex(lat=list(reversed(mean_data.lat)))

[20]: ## Only lats between 15 and 75
      lat_min = 15
      lat_max=75
      mean_data = mean_data.sel(lat=slice(lat_min, lat_max))

[21]: filtered_mean_data = apply_lancoz_filter(mean_data, filter_freq=10, window_size=61)

[22]: ## before
      mean_data['ua'][100]

[22]: <xarray.DataArray 'ua' (lat: 49)>
      array([-3.9038632 , -3.5604527 , -3.5367    , -3.5146356 , -3.053179  ,
             -2.1034822 , -1.0978746 , -0.32219973,  0.49564856,  1.34152   ,
              2.2609496 ,  3.3286147 ,  4.5936337 ,  5.804064  ,  7.078275  ,
              8.045278  ,  8.6514845 ,  8.5648985 ,  8.458412  ,  8.2538185 ,
              7.869816  ,  7.5115047 ,  6.7492895 ,  5.635572  ,  4.620459  ,
              3.6683176 ,  2.7055445 ,  1.7739418 ,  0.76496196, -0.15904112,
             -0.8327735 , -1.3357099 , -1.7940115 , -2.2432022 , -2.7942348 ,
             -3.2484474 , -3.486374  , -3.3994331 , -3.1952105 , -2.9751742 ,
             -2.9037502 , -3.2213612 , -3.5094094 , -3.6368015 , -3.7498631 ,
             -3.769252  , -3.6677792 , -3.0857055 , -1.7986805 ], dtype=float32)
      Coordinates:
          time     object 2015-04-11 12:00:00
        * lat      (lat) float64 15.0 16.25 17.5 18.75 20.0 ... 71.25 72.5 73.75 75.0

[23]: ## after
      filtered_mean_data[100]

[23]: <xarray.DataArray (lat: 49)>
      array([-4.20956129, -3.73016963, -3.33366258, -2.88158888, -2.31081294,
             -1.67817857, -0.98678979, -0.25856305,  0.55889753,  1.39777087,
              2.34299108,  3.25080116,  4.1233274 ,  4.93521054,  5.97292744,
              6.93619479,  7.59767942,  7.54263457,  7.4899679 ,  7.30749696,
              6.97105213,  6.5363142 ,  5.8496134 ,  4.97220355,  4.19216775
```



Step 2. Write de-coupled functions for metric

- Stick to Single Responsibility Principle (SRP)
- Each function takes same argument: *data*
- Refactor, Refactor, Refactor!

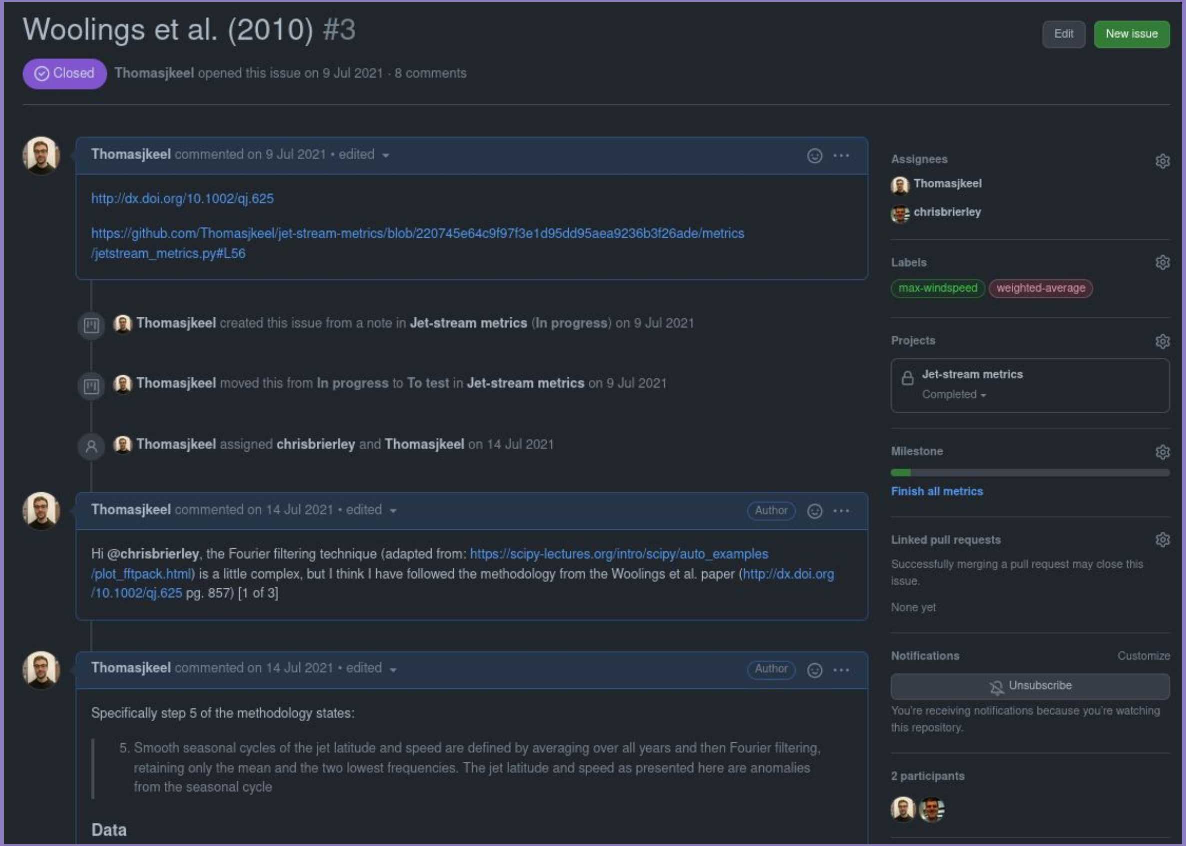
```python
# Step 1: Calculate long and/or plev mean
zonal_mean = jetstream_metrics_utils.get_zonal_mean(data)

# Step 2: Apply n-day lancoz filter
lancoz_filtered_mean_data = jetstream_metrics_utils.apply_lanczos_filter(
    zonal_mean["ua"], filter_freq, window_size
)

# Step 3: Calculate max windspeed and lat where max ws found
max_lat_ws = np.array(
    list(
        map(
            jetstream_metrics_utils.get_latitude_and_speed_where_max_ws,
            lancoz_filtered_mean_data[:],
        )
    )
)
zonal_mean_lat_ws = jetstream_metrics_utils.assign_lat_and_ws_to_data(
    zonal_mean, max_lat_ws
)
```

```python
def woolings_et_al_2010(data, filter_freq=10, window_size=61):
    """
    Method from Woolings et al (2010) http://dx.doi.org/10.1002/qj.625

    Follows an in-text description of 4-steps describing the algorithm of jet-stream identification from Woolings et al. (2010).
    Will calculate this metric based on data (regardless of pressure level of time span etc.).

    Parameters
    ----------
    data : xarray.Dataset
        Data containing u- component wind
    filter_freq : int
        number of days in filter (default=10 timeunits)
    window_size : int
        number of days in window for Lancoz filter (default=61 timeunits)

    Returns
    ----------
    fourier_filtered_data : xarray.Dataset
        Data containing maximum latitudes and maximum windspeed at those lats and fourier-filtered versions of those two variables
    """
    if isinstance(data, xarray.DataArray):
        data = data.to_dataset()
    # Step 1: Calculate long and/or plev mean
    zonal_mean = jetstream_metrics_utils.get_zonal_mean(data)

    # Step 2: Apply n-day lancoz filter
    lancoz_filtered_mean_data = jetstream_metrics_utils.apply_lanczos_filter(
        zonal_mean["ua"], filter_freq, window_size
    )
    # TODO make way of assuring that a dataarray is passed

    # Step 3: Calculate max windspeed and lat where max ws found
    max_lat_ws = np.array(
        list(
            map(
                jetstream_metrics_utils.get_latitude_and_speed_where_max_ws,
                lancoz_filtered_mean_data[:],
            )
        )
    )
    zonal_mean_lat_ws = jetstream_metrics_utils.assign_lat_and_ws_to_data(
        zonal_mean, max_lat_ws
    )
    # Step 4: Make climatology
    climatology = general_utils.get_climatology(zonal_mean_lat_ws, "month")

    # Step 5: Apply low-freq fourier filter to both max lats and max ws
    fourier_filtered_lats = (
        jetstream_metrics_utils.apply_low_freq_fourier_filter(
            climatology["max_lats"].values, highest_freq_to_keep=2
        )
    )
    fourier_filtered_ws = (
        jetstream_metrics_utils.apply_low_freq_fourier_filter(
            climatology["max_ws"].values, highest_freq_to_keep=2
        )
    )

    # Step 6: Join filtered climatology back to the data
    time_dim = climatology["max_ws"].dims[0]
    fourier_filtered_data = (
        jetstream_metrics_utils.assign_filtered_lats_and_ws_to_data(
            zonal_mean_lat_ws,
            fourier_filtered_lats.real,
            fourier_filtered_ws.real,
            dim=time_dim,
        )
    )
    return fourier_filtered_data
```

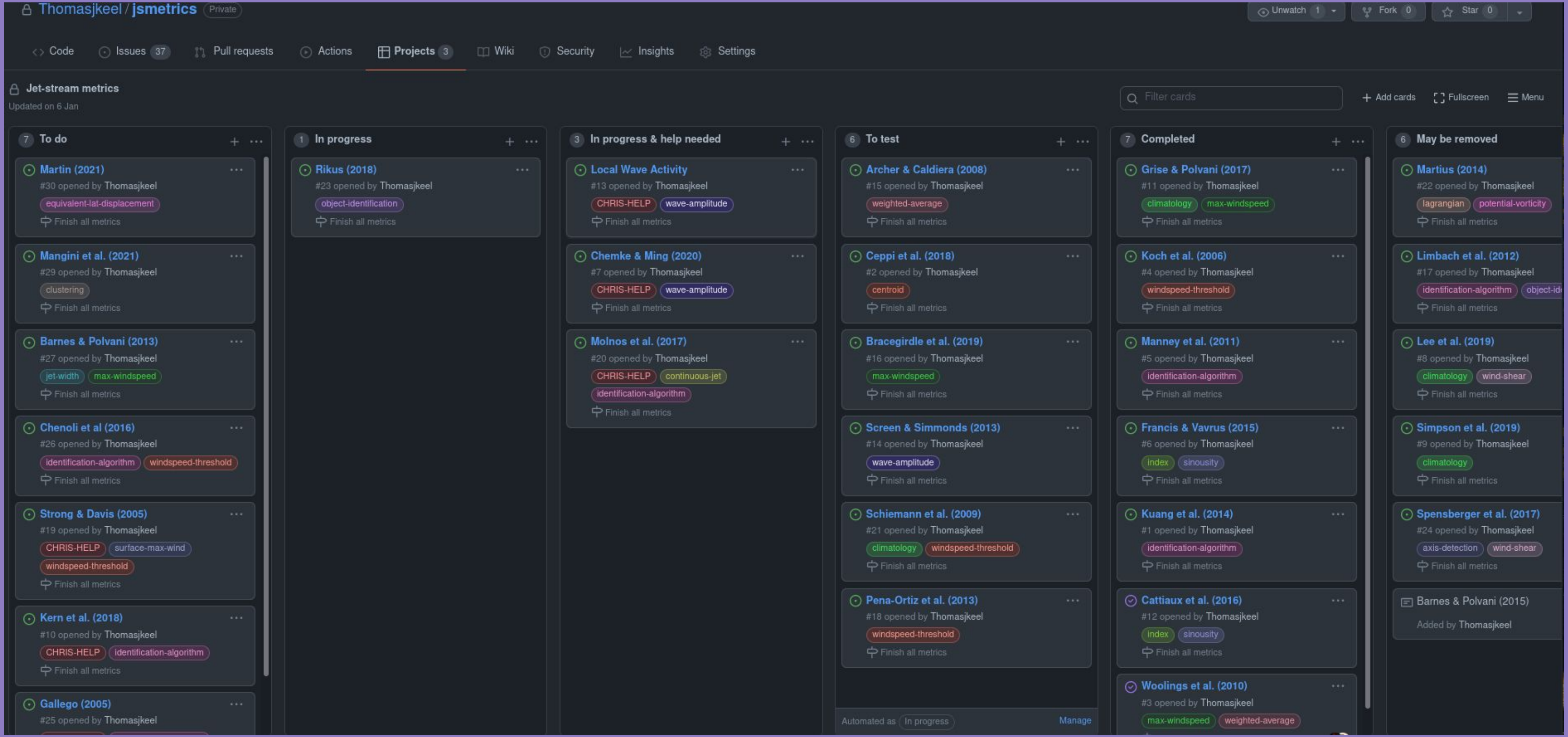
Step 3. Add to Python Module

- Keep it organised, keep it scalable
- Standard inputs, Standard outputs
- Version, manage dependencies (xarray), test

Managing the module with GitHub



Managing the module with GitHub



Initial experiment on JASMIN supercomputer

When: mid-October 2021

Runtime: 40 hours

Number of metrics: 10

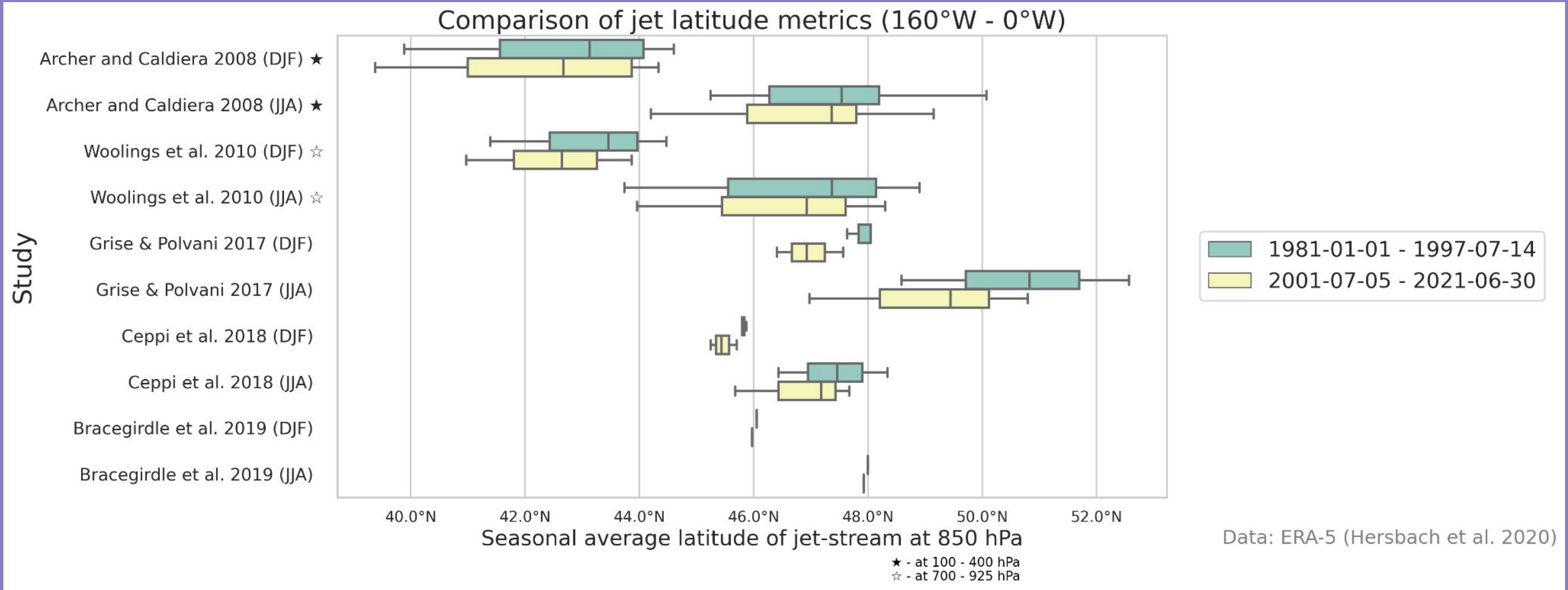
Data: ECMWF’s ERA-5 global climate reanalysis for Jan 1981 to Jun 2021

Variables: u-component wind; v-component wind

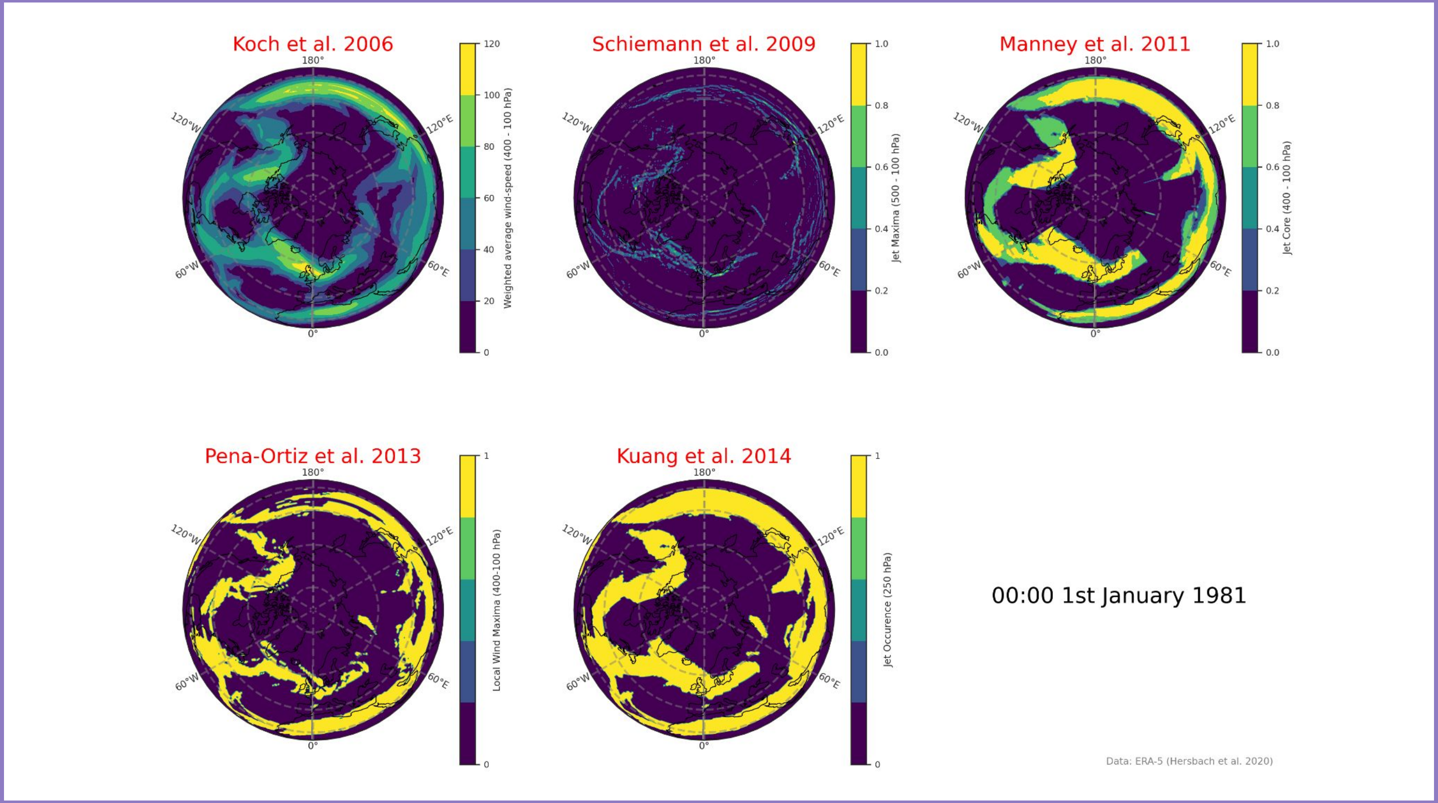
Size: ~11 GB

Outputs: Simple log, new .nc file to plot results locally

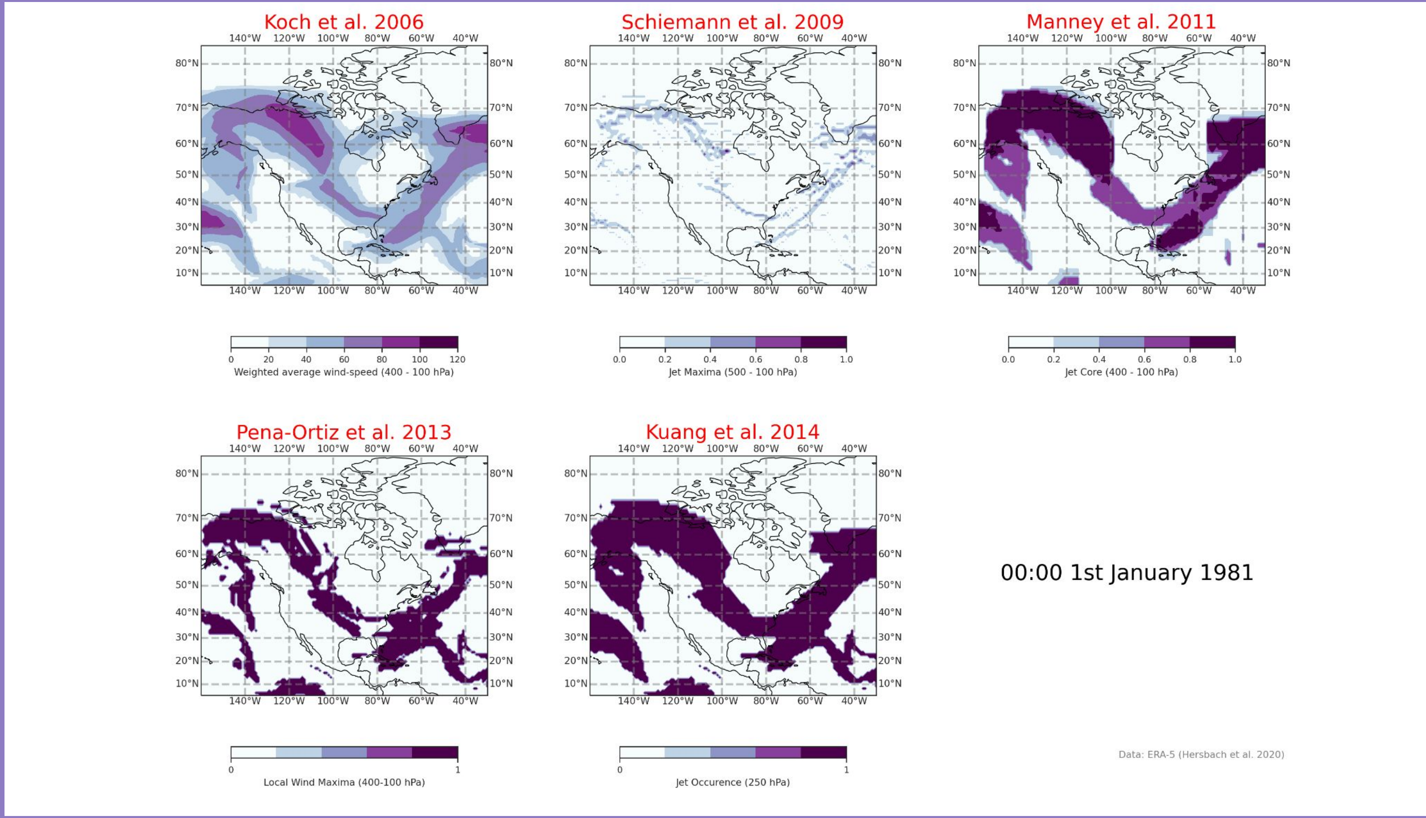
Initial experiment results



Initial experiment results



Initial experiment results



Running with the Research-atron

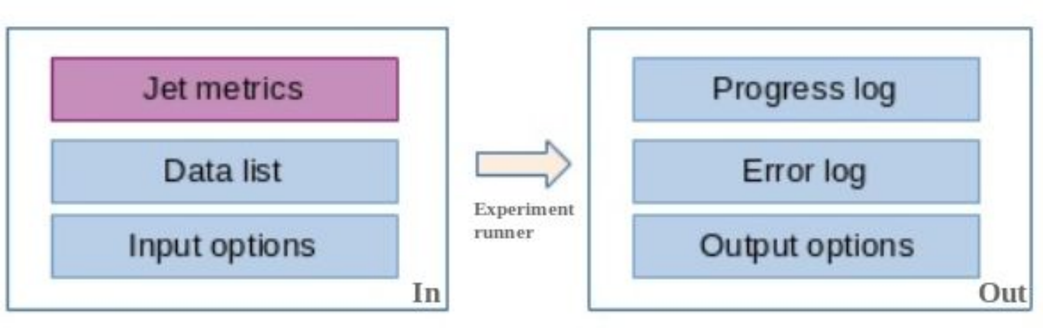
My needs after first experiments:

- Effective logging
- Doing data analysis in a stream
- Running on lots more data
- Ability to fail and fix itself

Running with the Research-atron

Input options:

- Where to find data
- Time-out limit



Output options:

- Which visualisations to save?
-

Initial research-atron experiment on JASMIN

When: *2nd* February 2022

Runtime: 1 hours

Number of metrics: 1 (Jet-latitude metric)

Data: 187 datasets from 7 modelling groups of projections between Jan 2020 and Jan 2040

Variables: u-component wind

Size: between 0.1-10 GB per dataset

Outputs: Various log, plot on JASMIN

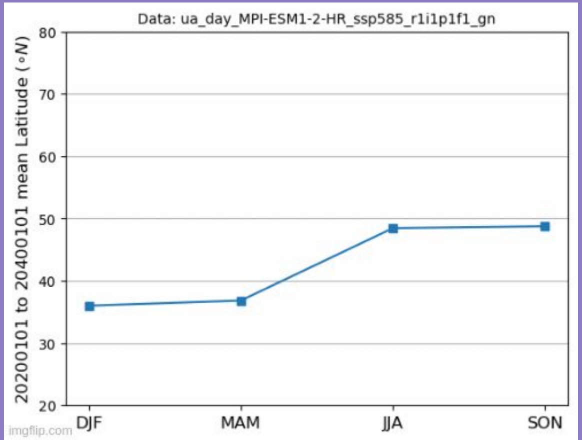
```
JETSTREAM_METRIC_DICT = {
    "Bracegirdle2018NH": {
        "variables": ["ua"],
        "coords": {
            "plev": [85000, 85000],
            "lat": [20, 80],
        },
        "metric": jetstream_metrics.bracegirdle_et_al_2018,
        "name": " Bracegirdle et al. 2018",
        "decription": "",
        "doi": "https://doi.org/10.1175/JCLI-D-17-0320.1",
    },
}
```

Bracegirdle, T. J., Hyder, P., & Holmes, C. R. (2018). CMIP5 diversity in Southern Westerly jet projections related to historical sea ice area: Strong link to strengthening and weak link to shift. *Journal of Climate*, *31*(1), 195–211. https://doi.org/10.1175/JCLI-D-17-0320.1

Running with the Research-atron

Findings:

- Data is not always standard
- Slight discrepancy between modelling groups




```
Number of datasets found: 1078
Number of datasets after date range subset (20200101 to 20400101): 187
```

```
02/11/2022 03:34:01 PM : (main.py): INFO: main_grouped_no_progress_log Line: 182 - Starting ua_day_AWI-CM-1-1-MR_ssp585_r1i1p1f1_gn. 1 out of 31. Total datsets in group: 20
02/11/2022 03:45:03 PM : (main.py): INFO: main_grouped_no_progress_log Line: 194 - 1 sucessfully loaded
02/11/2022 03:45:26 PM : (main.py): INFO: main_grouped_no_progress_log Line: 209 - subset for  Bracegirdle et al. 2018
02/11/2022 03:45:26 PM : (main.py): INFO: main_grouped_no_progress_log Line: 210 - Subset data coords: Coordinates:
   plev     float64 8.5e+04
 * time     (time) object 2020-01-01 12:00:00 ... 2039-12-31 12:00:00
 * lat      (lat) float64 20.1 21.04 21.97 22.91 ... 77.14 78.08 79.01 79.95
 * lon      (lon) float64 0.0 0.9375 1.875 2.812 ... 356.2 357.2 358.1 359.1
02/11/2022 03:45:31 PM : (main.py): INFO: main_grouped_no_progress_log Line: 222 -  Bracegirdle et al. 2018 run
02/11/2022 03:45:31 PM : (main.py): INFO: main_grouped_no_progress_log Line: 223 - Output data variables: Data variables:
   time_bnds      (time, bnds) object 2020-01-01 00:00:00 ... 2040-01-01 00:...
   lat_bnds       (time, lat, bnds) float64 19.64 20.57 20.57 ... 79.48 80.41
   lon_bnds       (time, lon, bnds) float64 -0.4688 0.4688 ... 358.6 359.5
   ua             (time, lat, lon) float32 -2.871 -3.109 ... -9.024 -9.398
   seasonal_JPOS  (season) float64 43.2 48.53 42.68 52.2
   annual_JPOS    (year) float64 46.13 43.88 44.7 45.0 ... 45.15 44.55 44.63
   seasonal_JSTR  (season) float64 6.613 4.265 4.952 6.467
   annual_JSTR    (year) float64 5.631 4.671 5.397 4.964 ... 5.464 5.169 5.38
02/11/2022 03:45:31 PM : (main.py): INFO: main_grouped_no_progress_log Line: 237 -  Bracegirdle et al. 2018 output saved to experiments/ScenarioMIP/outputs
```

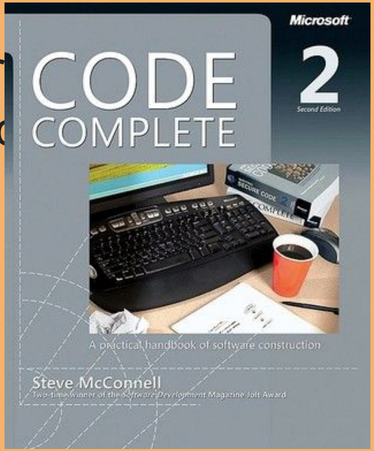
Open-source

Wait a minute, that's socialism!

For climate research:

- Is about getting enough people the right tools (think computer, then think software)
- We have an opportunity to pool problem solving in a new and exciting way.

Programming languages have always facilitated/enhanced climate r we are not at the ceiling of possibility, or have all the tools we can pc for research.



Example: xclim

xclim

`xclim` is a library of functions to compute climate indices from observations or model simulations.

- Takes metrics that already exist (in literature), but Pythonises them and makes them run fast.
- Huge inspiration for my own-code and this presentation.
- Thanks to Raquel Alegre, Jamie Quinn and Clair Barnes for getting me involved

One issue with this form of open-source:

- Hiding too much complexity from those who like problem solving

Data-driven vs Theory-driven research

Comments from my own experience:

- Solving complex problems with data when being reductive.
- "Some solution exists!"
- When can we side step theory?

Example of need for theory:

- Woolings et al. 2010. Where knowing something about the jet-streams helps.

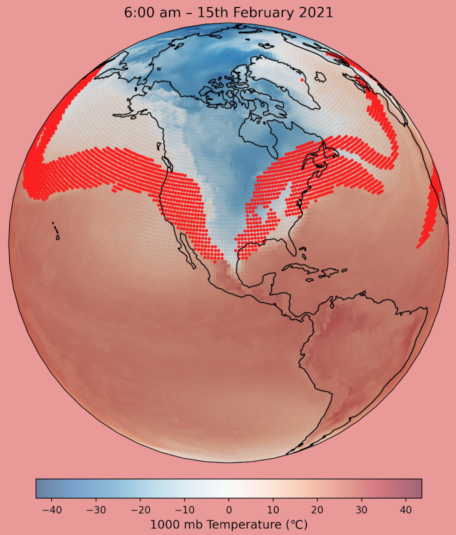
BUT: No theory needed when:

- Software used as a tool
- Machine learning algorithms and 'black-box' methodologies

Summary of key points

- All metrics are wrong, some are useful.
- Reading between the lines with climate information.
- (Climate) scientists are often self-confessed gate-keepers of (climate) science knowledge but there is a big opportunity to use open-source is an opportunity to get the tools in front of more people.

Any questions?



github: @Thomasjkeel
email: thomas.keel.18@ucl.ac.uk