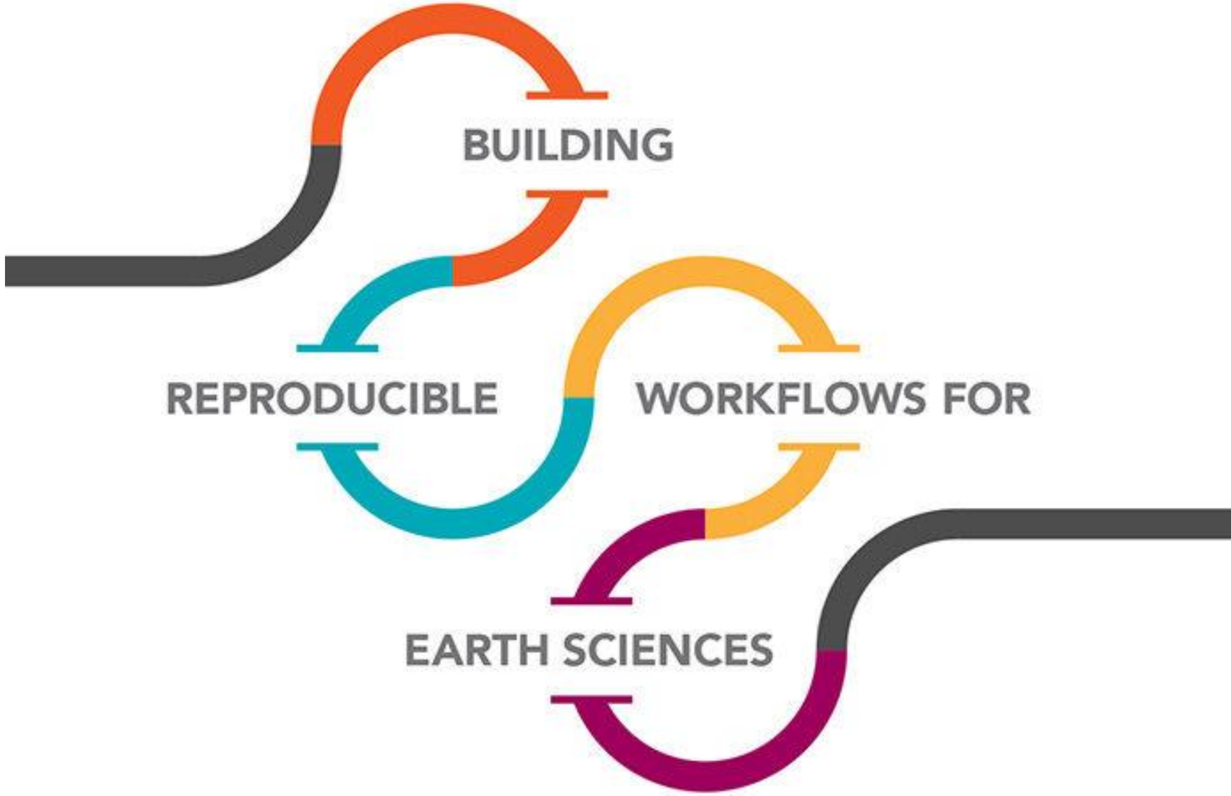
Reproducible workflows – setting the scene

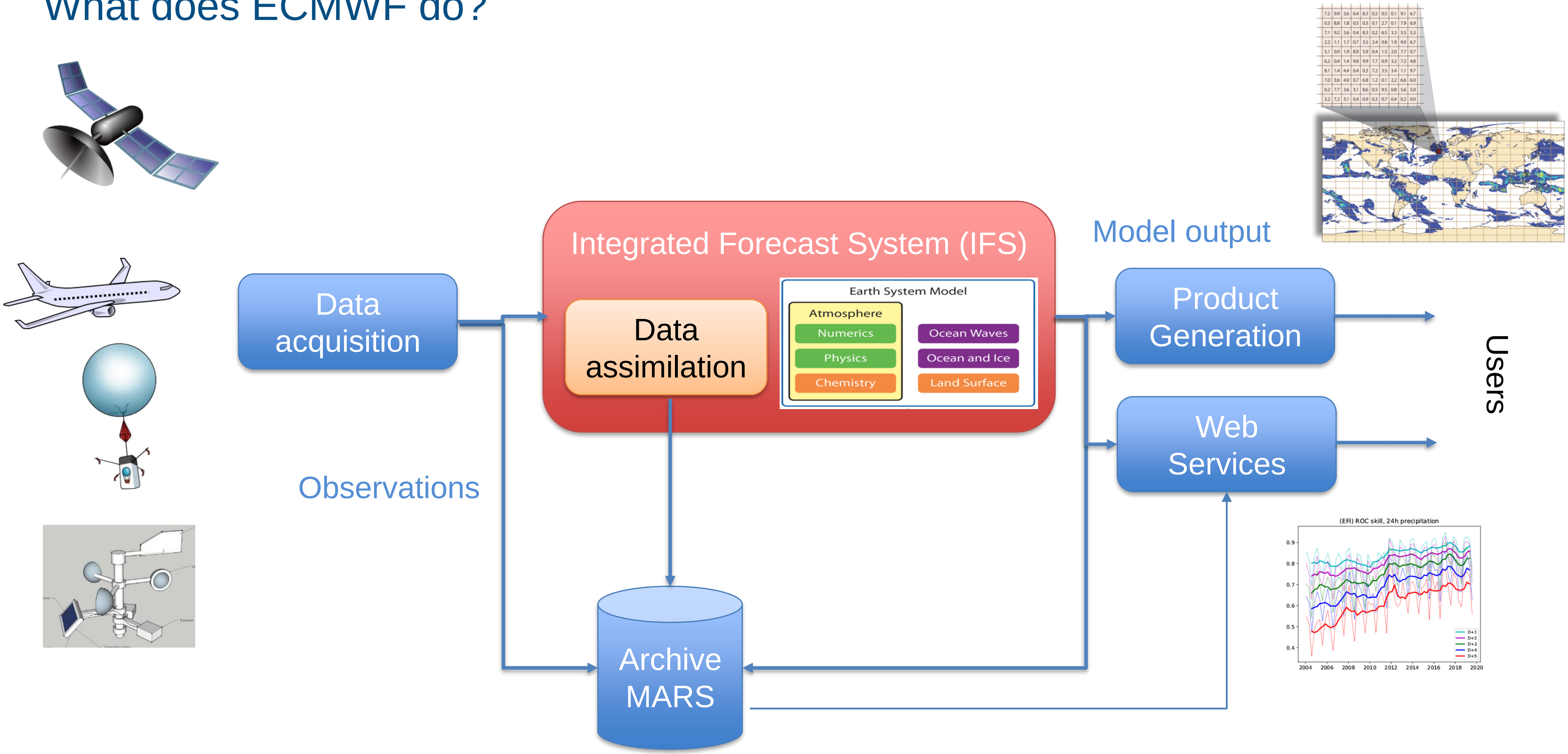
Why are we here?

Stephan Siemen

Development Section, Forecast Department



What does ECMWF do?



Observations

Data acquisition

Integrated Forecast System (IFS)

Data assimilation

Earth System Model

Atmosphere

Numerics

Physics

Chemistry

Ocean Waves

Ocean and Ice

Land Surface

Archive MARS

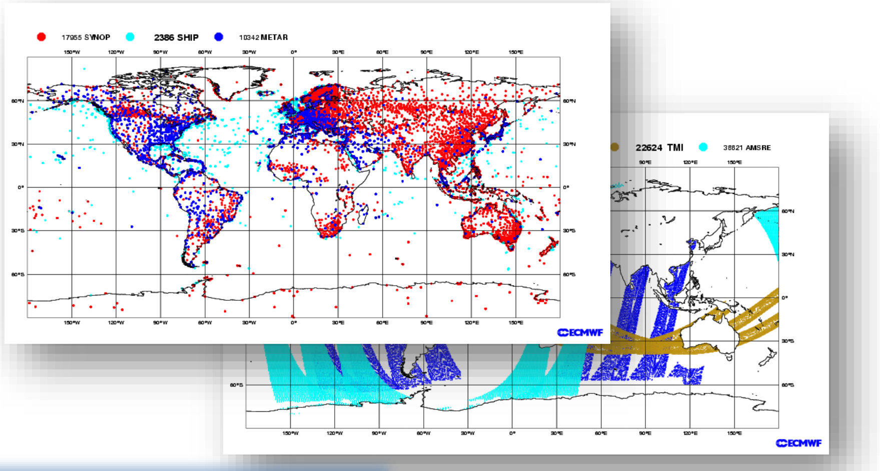
Model output

Product Generation

Web Services

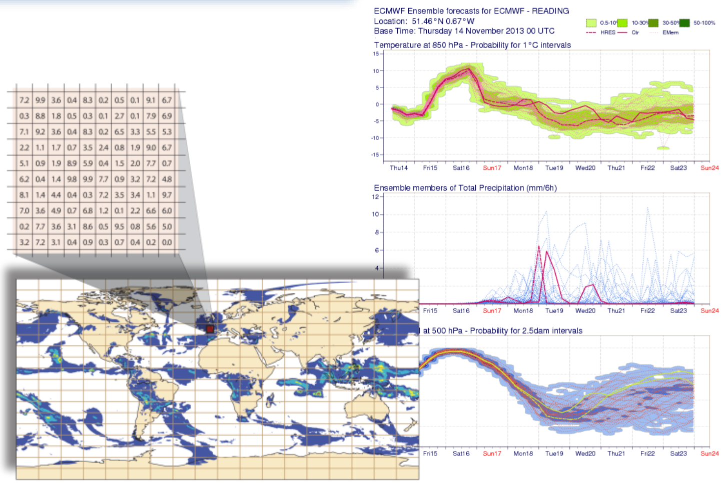
Users

The three stages of a research workflow



1. data acquisition & cleaning & filtering

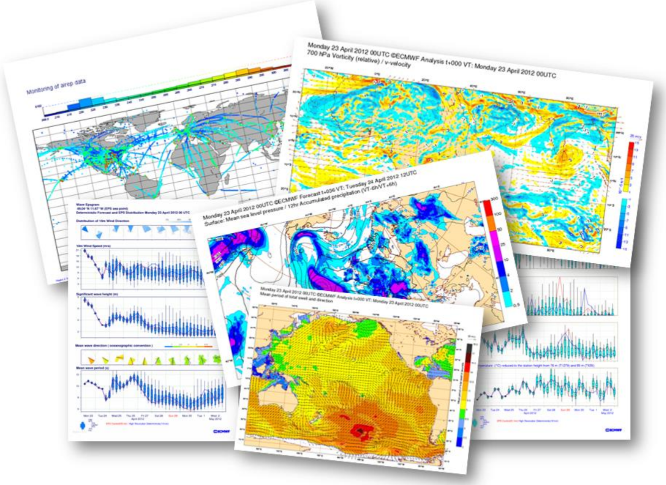
2. data processing – running models



7.2	9.9	3.6	0.4	8.3	0.2	0.5	0.1	9.1	6.7
0.3	8.8	1.8	0.5	0.3	0.1	2.7	0.1	7.9	6.9
7.1	9.2	3.6	0.4	8.3	0.2	6.5	3.3	5.5	5.3
2.2	1.1	1.7	0.7	3.5	2.4	0.8	1.9	9.0	6.7
5.1	0.9	1.9	8.9	5.9	0.4	1.5	2.0	7.7	0.7
6.2	0.4	1.4	9.8	9.9	7.7	0.9	3.2	7.2	4.8
8.1	1.4	4.4	0.4	0.3	7.2	3.5	3.4	1.1	9.7
7.0	3.6	4.9	0.7	6.8	1.2	0.1	2.2	6.6	6.0
0.2	7.7	3.6	3.1	8.6	0.5	9.5	0.8	5.6	5.0
3.2	7.2	3.1	0.4	0.9	0.3	0.7	0.4	0.2	0.0

3. data analysis – presenting outcomes

What is reproducibility anyway?



- Not only getting the same result
  - Hard to achieve anyway with different architectures and environments

- Having a recipe / procedure to follow to get to the same result
  - Allow to evaluate changes to input data and algorithm on output → do science

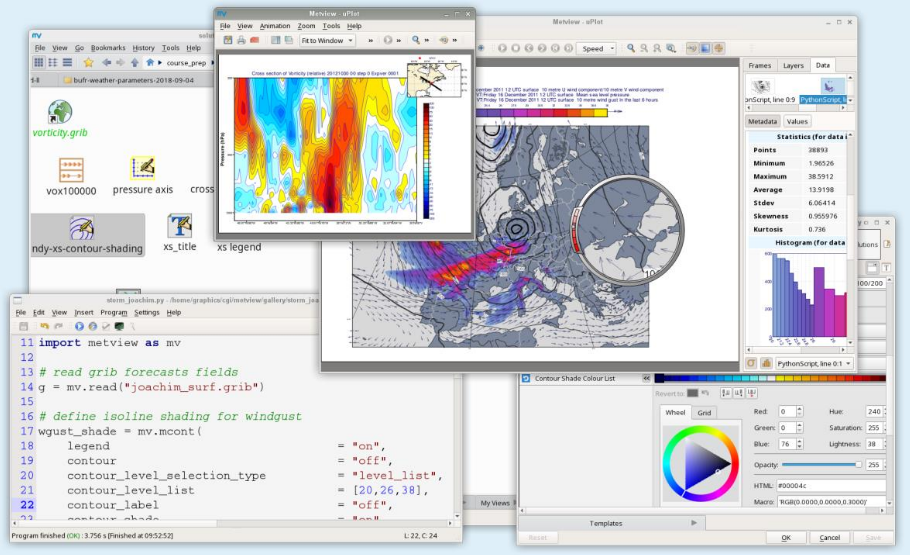
- Having workflow documented allows sharing and scrutiny

Why is it important?

- Important proof that science is solid
  - No one believes a result if it constantly changes
  - With changing input a hypothesis can be confirmed/corrected

- Allows easy sharing of scientific work and allows others to follow and reflect on work

- Big Data workflows need to carefully designed because of their high costs
  - Data amounts are challenging to handle and prone to change
  - Manual work does not scale and needs automation
  - This is especially true for AI / Machine Learning applications


vorticity.grib
vox100000
pressure axis
ndy-xs-contour-shading
xs_title
xs legend
storm_joachim.py - /home/graphics/cgi/metview/gallery/storm_joa
File Edit View Insert Program Settings Help
Metview - uPlot
Contour Shade Colour List
Wheel
Grid
Templates
OK
Cancel

```
import metview as mv

# read grib forecasts fields
g = mv.read("joachim_surf.grib")

# define isoline shading for windgust
wgust_shade = mv.mcont(
    legend                                  = "on",
    contour                                 = "off",
    contour_level_selection_type            = "level_list",
    contour_level_list                      = [20,26,38],
    contour_label                           = "off",
    contour_shade                           = "on",
```

```
Program finished (OK) : 3.756 s [Finished at 09:52:52]
```

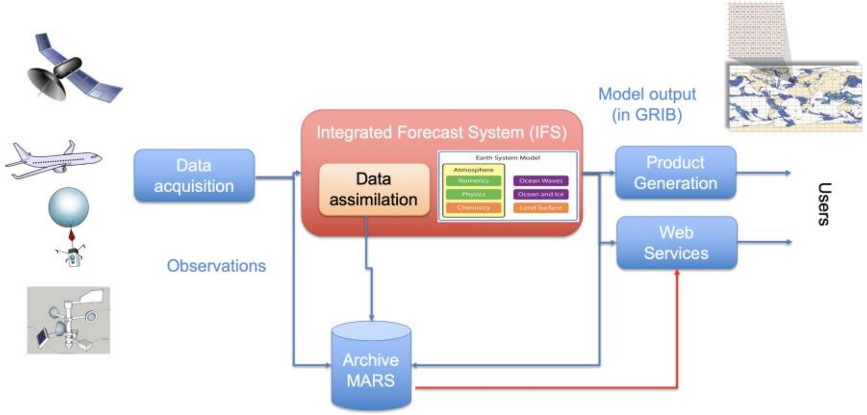
Why is it important to ECMWF?

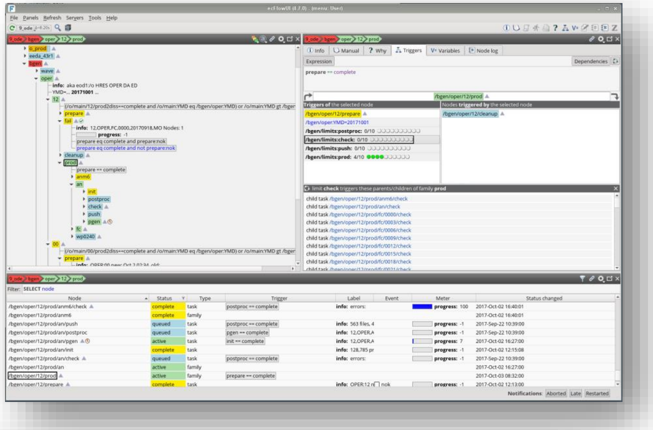
- Reproducibility has always been important to drive innovation on model developments

- As an operational centre users rely on ECMWF to provide robust results in a controlled and stable environment

- With data amounts continuing to grow complex processing jobs need to be moved to the data
  - Require flexible and high-level workflows

- Being a central part of a large scientific community, ECMWF is keen to share workflows and offer training





ecFlowUI (4.7.0) - (menu: User)
File Panels Refresh Servers Tools Help
9_ode
9_ode bgen oper 12 prod
Info Manual Why Triggers Variables Node log
Expression
Dependencies

o_prod
eeda_43r1
bgen
wave
oper
info: aka eod1:/o HRES OPER DA ED
YMD=... 20171001 ...
12
(/o/main/12/prod2diss==complete and /o/main:YMD eq /bgen/oper:YMD) or /o/main:YMD gt /bgen
prepare
fail
info: 12.OPER.FC.0000.20170918.MO Nodes: 1
progress: -1
prepare eq complete and prepare:nok
prepare eq complete and not prepare:nok
cleanup
prod
prepare == complete
anm6
an
init
postproc
check
push
pgen
fc
wp0240
00
(/o/main/00/prod2diss==complete and /o/main:YMD eq /bgen/oper:YMD) or /o/main:YMD gt /bgen
prepare
info: OPER:00 new: Oct 2 02:34 old:

prepare == complete

/bgen/oper/12/prod

Triggers of the selected node
/bgen/oper/12/prepare
/bgen/oper:YMD=20171001
/bgen/limits:postproc: 0/10
/bgen/limits:check: 0/10
/bgen/limits:push: 0/10
/bgen/limits:prod: 4/10

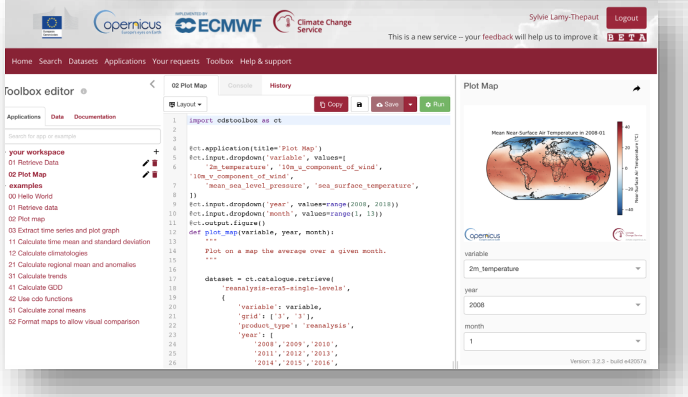
Nodes triggered by the selected node
/bgen/oper/12/cleanup

limit check triggers these parents/children of family prod
child task /bgen/oper/12/prod/anm6/check
child task /bgen/oper/12/prod/an/check
child task /bgen/oper/12/prod/fc/0000/check
child task /bgen/oper/12/prod/fc/0003/check
child task /bgen/oper/12/prod/fc/0006/check
child task /bgen/oper/12/prod/fc/0009/check
child task /bgen/oper/12/prod/fc/0012/check
child task /bgen/oper/12/prod/fc/0015/check
child task /bgen/oper/12/prod/fc/0018/check
child task /bgen/oper/12/prod/fc/0021/check

Filter: SELECT node

Node	Status	Type	Trigger	Label	Event	Meter	Status changed
/bgen/oper/12/prod/anm6/check	complete	task	postproc == complete	info: errors:		progress: 100	2017-Oct-02 16:40:01
/bgen/oper/12/prod/anm6	complete	family					2017-Oct-02 16:40:01
/bgen/oper/12/prod/an/push	queued	task	postproc == complete	info: 563 files, 4		progress: -1	2017-Sep-22 10:39:00
/bgen/oper/12/prod/an/postproc	queued	task	pgen == complete	info: 12.OPER.A		progress: -1	2017-Sep-22 10:39:00
/bgen/oper/12/prod/an/pgen	active	task	init == complete	info: 12.OPER.A		progress: 7	2017-Oct-02 16:27:00
/bgen/oper/12/prod/an/init	complete	task		info: 128,785 pr		progress: -1	2017-Oct-02 12:15:08
/bgen/oper/12/prod/an/check	queued	task	postproc == complete	info: errors:		progress: -1	2017-Sep-22 10:39:00
/bgen/oper/12/prod/an	active	family					2017-Oct-02 16:27:00
/bgen/oper/12/prod	active	family	prepare == complete				2017-Oct-03 08:32:00
/bgen/oper/12/prepare	complete	task		info: OPER:12 n nok		progress: -1	2017-Oct-02 12:13:00

Notifications: Aborted Late Restarted



opernicus
IMPLEMENTED BY ECMWF
Climate Change Service
Sylvie Lamy-Thepaut
Logout
BETA
Home Search Datasets Applications Your requests Toolbox Help & support
Toolbox editor
Applications Data Documentation
Search for app or example
02 Plot Map Console History
Layout
Copy Save Run
Plot Map
Mean Near-Surface Air Temperature in 2008-01
Near-Surface Air Temperature (°C)
40
20
0
-20
-40
variable
year
month

This is a new service -- your feedback will help us to improve it

your workspace
01 Retrieve Data
02 Plot Map

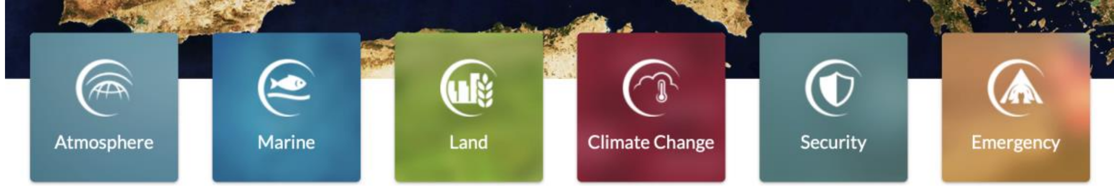
examples
00 Hello World
01 Retrieve data
02 Plot map
03 Extract time series and plot graph
11 Calculate time mean and standard deviation
12 Calculate climatologies
21 Calculate regional mean and anomalies
31 Calculate trends
41 Calculate GDD
42 Use cdo functions
51 Calculate zonal means
52 Format maps to allow visual comparison

```
import cdstoolbox as ct


@ct.application(title='Plot Map')
@ct.input.dropdown('variable', values=[
    '2m_temperature', '10m_u_component_of_wind',
'10m_v_component_of_wind',
    'mean_sea_level_pressure', 'sea_surface_temperature',
])
@ct.input.dropdown('year', values=range(2008, 2018))
@ct.input.dropdown('month', values=range(1, 13))
@ct.output.figure()
def plot_map(variable, year, month):
    """
    Plot on a map the average over a given month.
    """

    dataset = ct.catalogue.retrieve(
        'reanalysis-era5-single-levels',
        {
            'variable': variable,
            'grid': ['3', '3'],
            'product_type': 'reanalysis',
            'year': [
                '2008','2009','2010',
                '2011','2012','2013',
                '2014','2015','2016',
```

variable: 2m_temperature
year: 2008
month: 1

Version: 3.2.3 - build e42057a

EU Copernicus programme



- Large programme to make earth observations and monitoring data (freely) available
- Data is validated and from verified sources
- Sponsored by the European Commission
- ECMWF is operating two services on behalf of the EC

- Copernicus Atmospheric Monitoring Service (CAMS)
  - Air pollution, chemicals, sand & dust
  - https://atmosphere.copernicus.eu

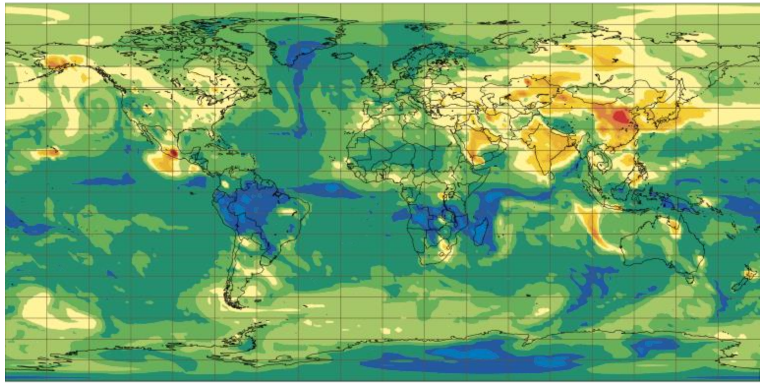
- Copernicus Climate Change Services (C3S)
  - Reanalysis data, seasonal forecasts, climate scenarios
  - https://climate.copernicus.eu

Copernicus
Europe's eyes on Earth

European Commission

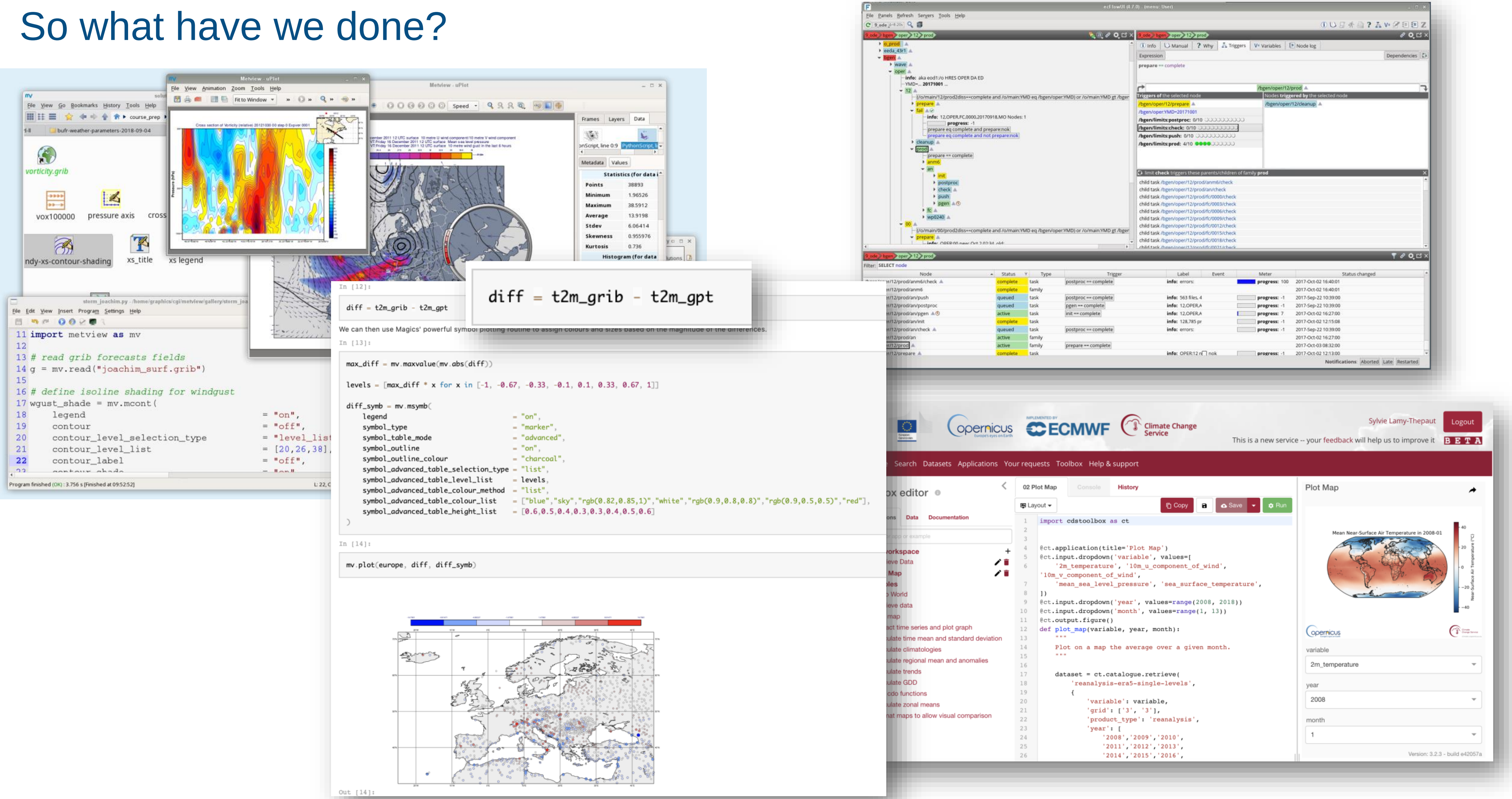
In collaboration with:

esa European Space Agency · EUMETSAT · ECMWF · EEA · MERCATOR OCEAN INTERNATIONAL · FRONTEX · SatCen



Total column of sulphur dioxide
(provided by the Copernicus Atmosphere Monitoring Service)

So what have we done?


Metview - uPlot
File View Animation Zoom Tools Help
Fit to Window
Speed
Frames Layers Data
Metadata Values
Statistics
Points 38893
Minimum 1.96526
Maximum 38.5912
Average 13.9198
Stdev 6.06414
Skewness 0.955976
Kurtosis 0.736
Histogram
File View Go Bookmarks History Tools Help
course_prep
bufr-weather-parameters-2018-09-04
vorticity.grib
vox100000
pressure axis
ndy-xs-contour-shading
xs_title
xs legend
ecflowUI
File Panels Refresh Servers Tools Help
Info Manual Why Triggers Variables Node log
Node Status Type Trigger Label Event Meter Status changed
Notifications: Aborted Late Restarted
Copernicus
ECMWF
Climate Change Service
Sylvie Lamy-Thepaut
Logout
BETA
Search Datasets Applications Your requests Toolbox Help & support
02 Plot Map Console History
Layout Copy Save Run
Plot Map
Mean Near-Surface Air Temperature in 2008-01
Version: 3.2.3 - build e42057a

storm_joachim.py - /home/graphics/cgi/metview/gallery/storm_joa

```
import metview as mv

# read grib forecasts fields
g = mv.read("joachim_surf.grib")

# define isoline shading for windgust
wgust_shade = mv.mcont(
    legend                              = "on",
    contour                             = "off",
    contour_level_selection_type        = "level_list
    contour_level_list                  = [20,26,38],
    contour_label                       = "off",
```

Program finished (OK) : 3.756 s [Finished at 09:52:52]

In [12]:

```
diff = t2m_grib - t2m_gpt
```

```
diff = t2m_grib - t2m_gpt
```

We can then use Magics' powerful symbol plotting routine to assign colours and sizes based on the magnitude of the differences.

In [13]:

```
max_diff = mv.maxvalue(mv.abs(diff))

levels = [max_diff * x for x in [-1, -0.67, -0.33, -0.1, 0.1, 0.33, 0.67, 1]]

diff_symb = mv.msymb(
    legend                                  = "on",
    symbol_type                             = "marker",
    symbol_table_mode                       = "advanced",
    symbol_outline                          = "on",
    symbol_outline_colour                   = "charcoal",
    symbol_advanced_table_selection_type    = "list",
    symbol_advanced_table_level_list        = levels,
    symbol_advanced_table_colour_method     = "list",
    symbol_advanced_table_colour_list       = ["blue","sky","rgb(0.82,0.85,1)","white","rgb(0.9,0.8,0.8)","rgb(0.9,0.5,0.5)","red"],
    symbol_advanced_table_height_list       = [0.6,0.5,0.4,0.3,0.3,0.4,0.5,0.6]
)
```

In [14]:

```
mv.plot(europe, diff, diff_symb)
```

Out [14]:

```
import cdstoolbox as ct


@ct.application(title='Plot Map')
@ct.input.dropdown('variable', values=[
    '2m_temperature', '10m_u_component_of_wind',
'10m_v_component_of_wind',
    'mean_sea_level_pressure', 'sea_surface_temperature',
])
@ct.input.dropdown('year', values=range(2008, 2018))
@ct.input.dropdown('month', values=range(1, 13))
@ct.output.figure()
def plot_map(variable, year, month):
    """
    Plot on a map the average over a given month.
    """

    dataset = ct.catalogue.retrieve(
        'reanalysis-era5-single-levels',
        {
            'variable': variable,
            'grid': ['3', '3'],
            'product_type': 'reanalysis',
            'year': [
                '2008','2009','2010',
                '2011','2012','2013',
                '2014','2015','2016',
```

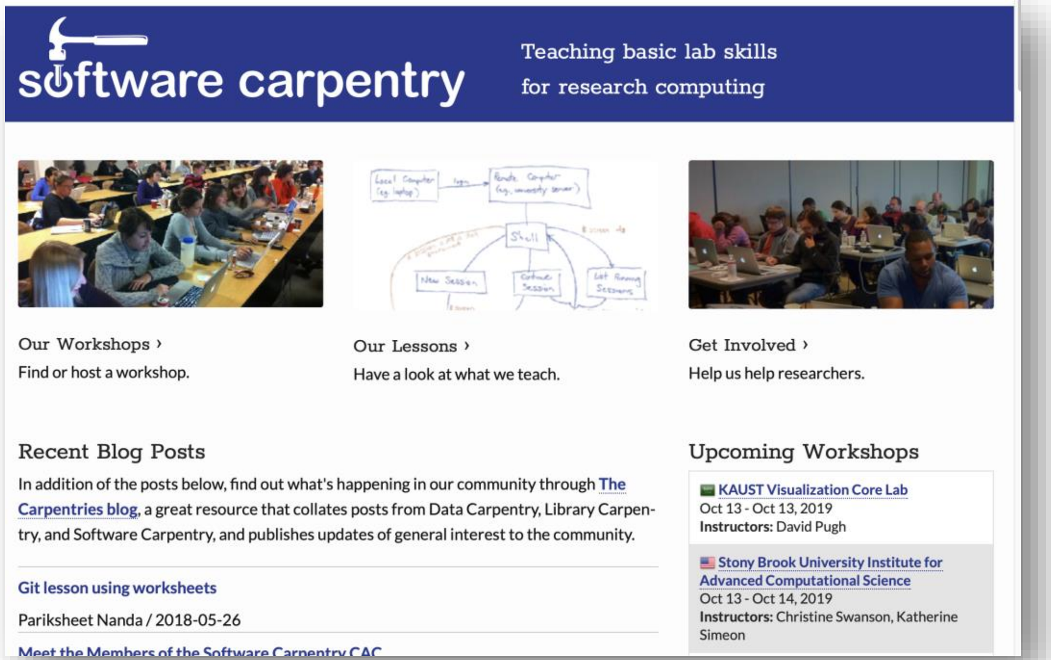
variable: 2m_temperature

year: 2008

month: 1

What makes a workflow reproducible?

REPRODUCIBILITY
METHODS
RESULTS
INFERENCE
RIGOUR



Teaching basic lab skills
for research computing

Our Workshops ›
Find or host a workshop.

Our Lessons ›
Have a look at what we teach.

Get Involved ›
Help us help researchers.

Recent Blog Posts

In addition of the posts below, find out what's happening in our community through The Carpentries blog, a great resource that collates posts from Data Carpentry, Library Carpentry, and Software Carpentry, and publishes updates of general interest to the community.

Git lesson using worksheets

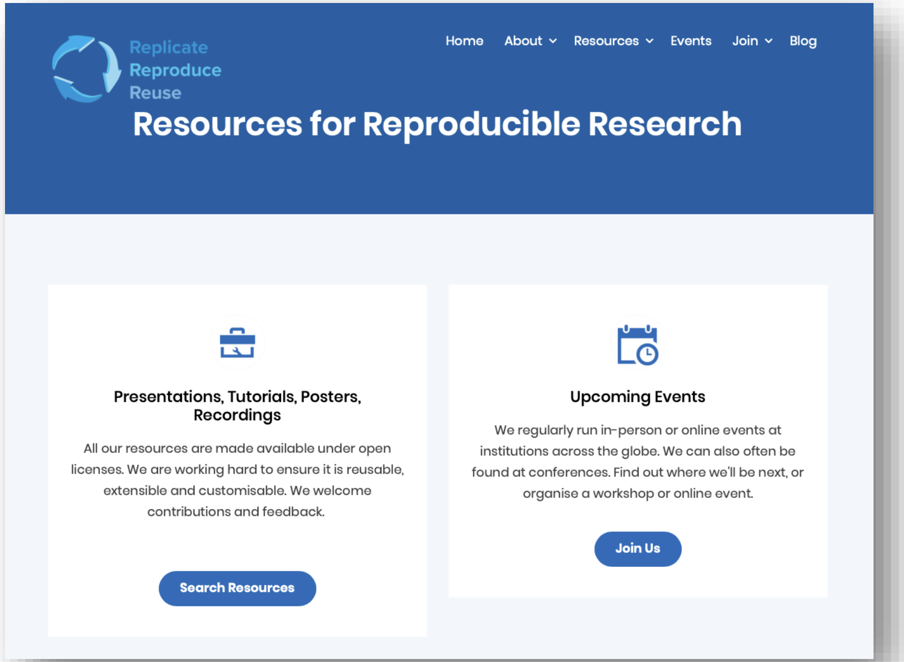
Pariksheet Nanda / 2018-05-26

Meet the Members of the Software Carpentry CAC

Upcoming Workshops

KAUST Visualization Core Lab
Oct 13 - Oct 13, 2019
Instructors: David Pugh

Stony Brook University Institute for Advanced Computational Science
Oct 13 - Oct 14, 2019
Instructors: Christine Swanson, Katherine Simeon



Resources for Reproducible Research

Presentations, Tutorials, Posters, Recordings

All our resources are made available under open licenses. We are working hard to ensure it is reusable, extensible and customisable. We welcome contributions and feedback.

Upcoming Events

We regularly run in-person or online events at institutions across the globe. We can also often be found at conferences. Find out where we'll be next, or organise a workshop or online event.

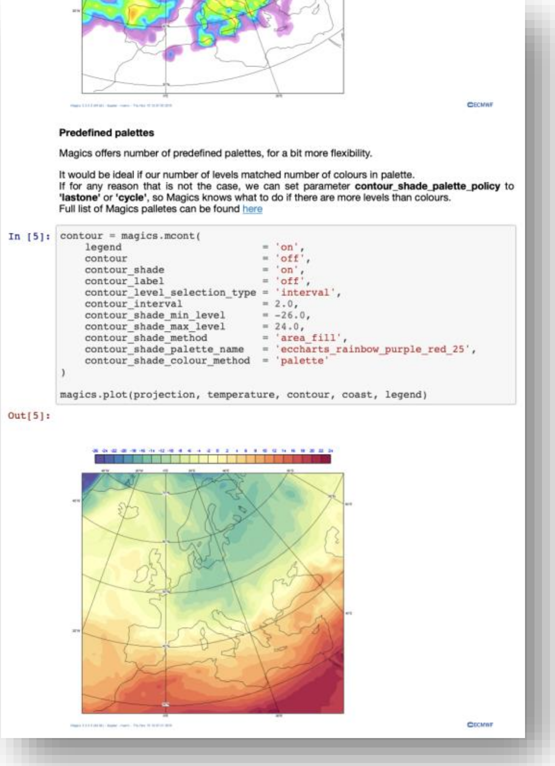
General advice

- Document all data, files, and operations that occur on data and files. Keep meta data of all components and operations

- Create the workflow as a sequence of small steps that are connected. Use intermediate outputs from a step input for the next step

- Automate all work as much as possible, and avoiding manual intervention in the workflow – let the automation (scripts) be the documentation

- Key are clear interfaces between components - APIs



Predefined palettes

Magics offers number of predefined palettes, for a bit more flexibility.

It would be ideal if our number of levels matched number of colours in palette.
If for any reason that is not the case, we can set parameter **contour_shade_palette_policy** to **'lastone'** or **'cycle'**, so Magics knows what to do if there are more levels than colours.
Full list of Magics palletes can be found here

In [5]:
```
contour = magics.mcont(
    legend                       = 'on',
    contour                      = 'off',
    contour_shade                = 'on',
    contour_label                = 'off',
    contour_level_selection_type = 'interval',
    contour_interval             = 2.0,
    contour_shade_min_level      = -26.0,
    contour_shade_max_level      = 24.0,
    contour_shade_method         = 'area_fill',
    contour_shade_palette_name   = 'eccharts_rainbow_purple_red_25',
    contour_shade_colour_method  = 'palette'
)

magics.plot(projection, temperature, contour, coast, legend)
```

Out[5]:

Technical developments of recent years help

- Git made version control main stream
  - Big ecosystem of tools → GitHub enables community for developers
- Python and its large eco system of packages
- Conda & pip help to define environments
  - Let’s user easily recreate environments
- Jupyter notebooks made it popular to document codes
  - Easy to share interactive environments
- Containers make it easier to ”freeze” environments
  - Next steps up are Kubernetes and service meshes to define whole services

We have an interesting workshop in front of us

- Three days of great talks and demonstrations
- Discussions to exchange experiences