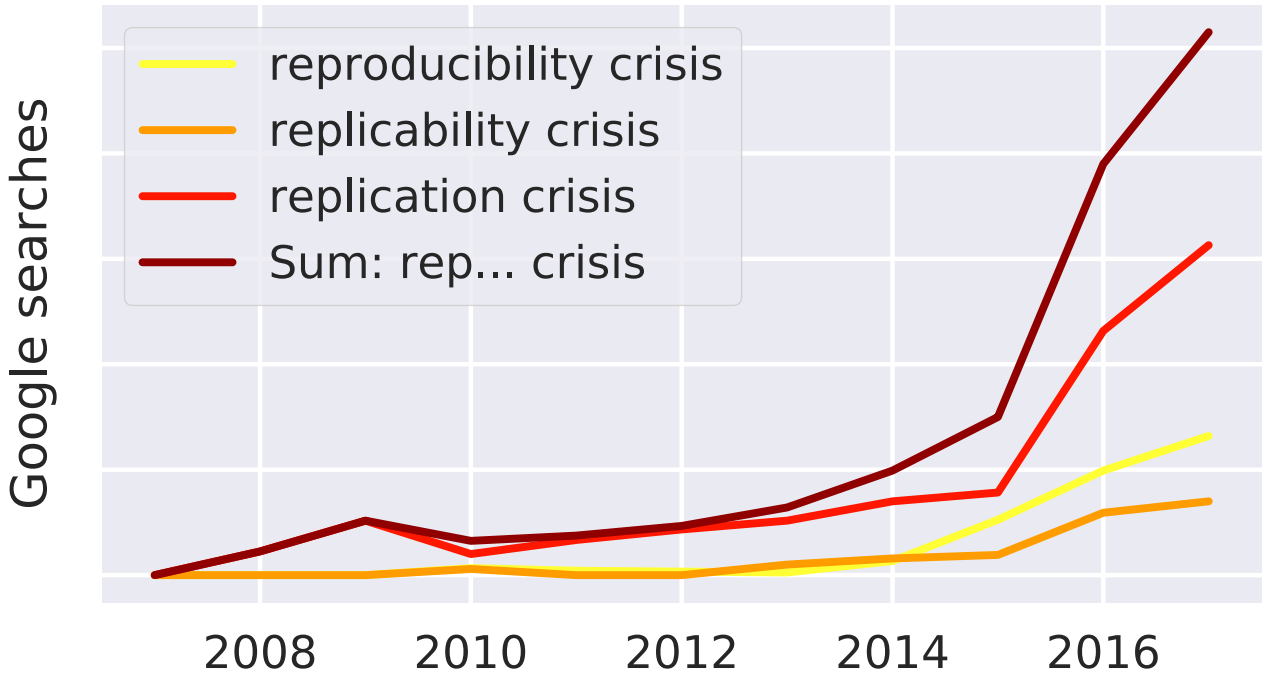
Towards *Efficient & Reproducible* Science

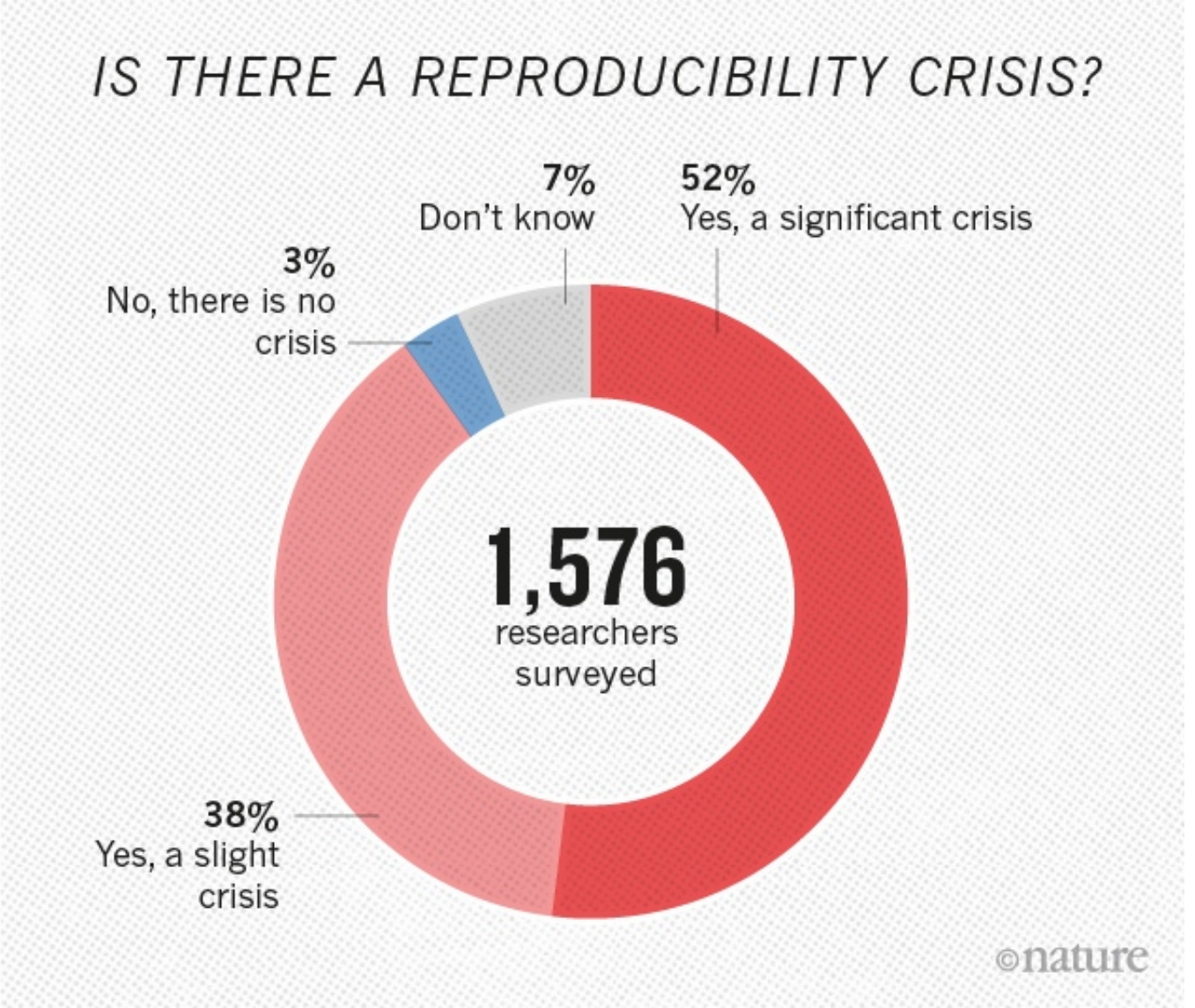
Willi Rath | wrath@geomar.de

Thanks: *Martin Claus, Claus Böning, Torge Martin, Markus Scheinert, Klaus Getzlaff, Franziska Schwarzkopf, Christina Roth, Rafael Abel, Arne Biastoch, Kristin Burmeister, Julia Getzlaff, Carsten Schirnick, Claas Faber, Kai Grunau, Stefan Jöhnke, Lutz Griesbach, Thomas Grunert, Knut Günther, Friedrich Althausen, GEOMAR Data-Management Team, GEOMAR IT Department, ...*

slides — https://willirath.gitlab.io/towards_reproducible_science/
Git repo — https://gitlab.com/willirath/towards_reproducible_science/



This notebook has details.

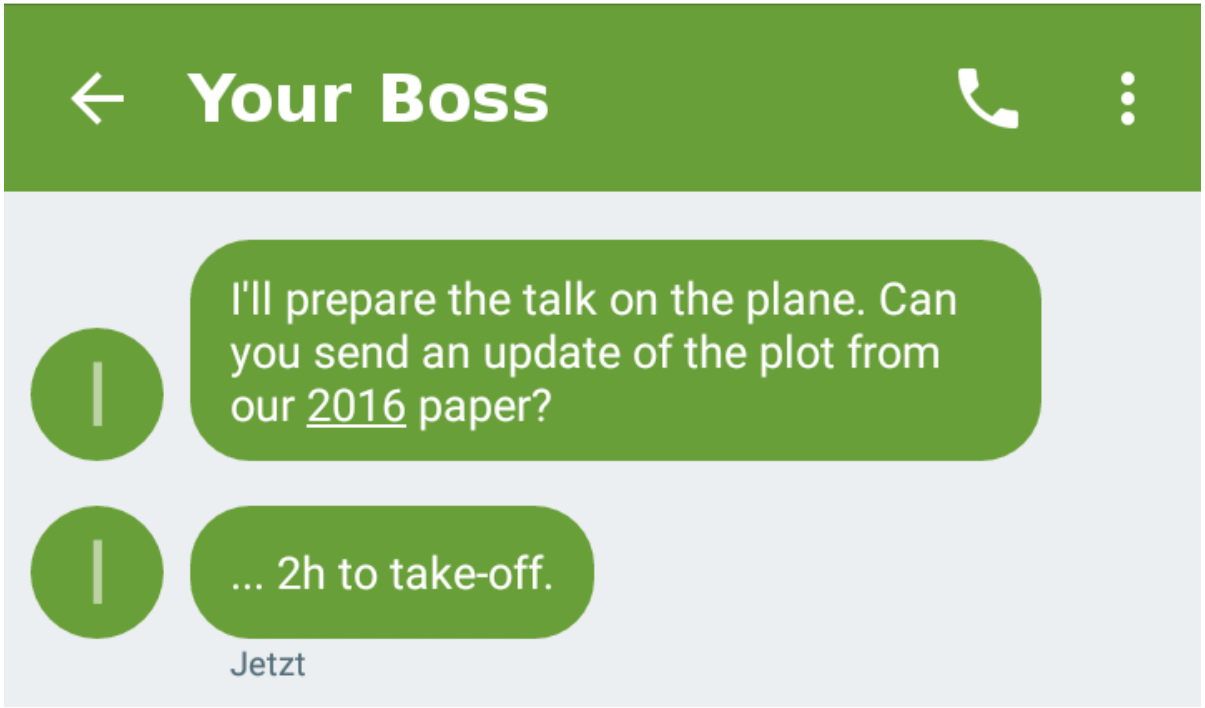


From Baker (2016)

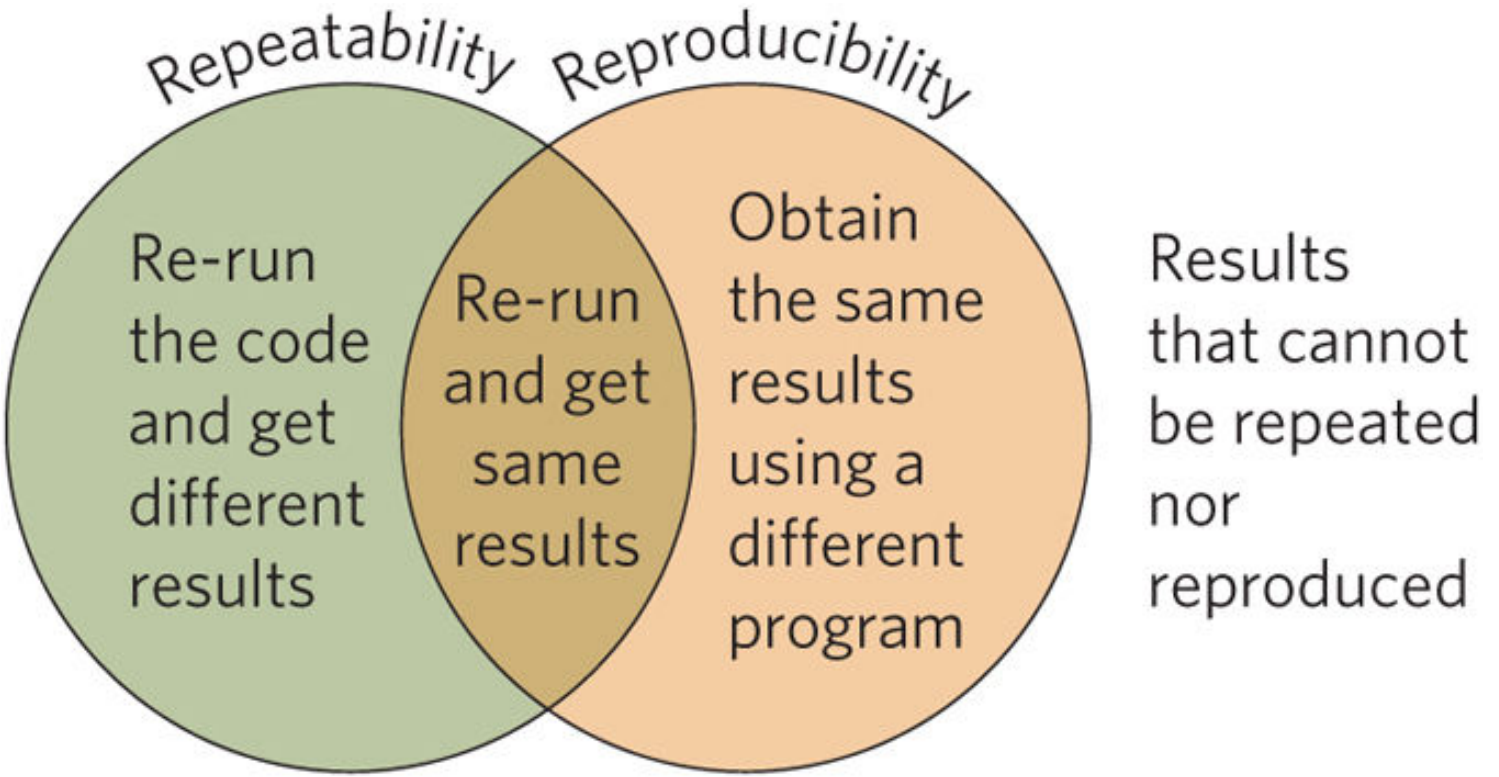
… not my department?

Public debate mostly focused on *fraud prevention* in the medical sciences.

— I'll argue that it's ***you*** who'd benefit from reproducibility.


Your Boss
I'll prepare the talk on the plane. Can you send an update of the plot from our 2016 paper?
... 2h to take-off.
Jetzt

> ***You:*** "Can you check this sea-level trend against satellite data?"
>
> ***Student:*** "… sure …"
>
> *(student about to leave for two weeks of googling for data)*
>
> ***You:*** "Hey wait, here's a script where I did a similar thing with the old AVISO data. Maybe it's good to start there? When you're familiar with this one, adapt it to the new SLTAC product."



From Easterbrook (2014)

Repeatability ~~*Reproducibility*~~

> Let's say an analysis is *repeatable*, if for any *sufficiently skilled* reader it is *in principle* possible to *completely understand* and *repeat all steps* the authors took from their initial idea to the final conclusions.

Example — A Simple Time Series?

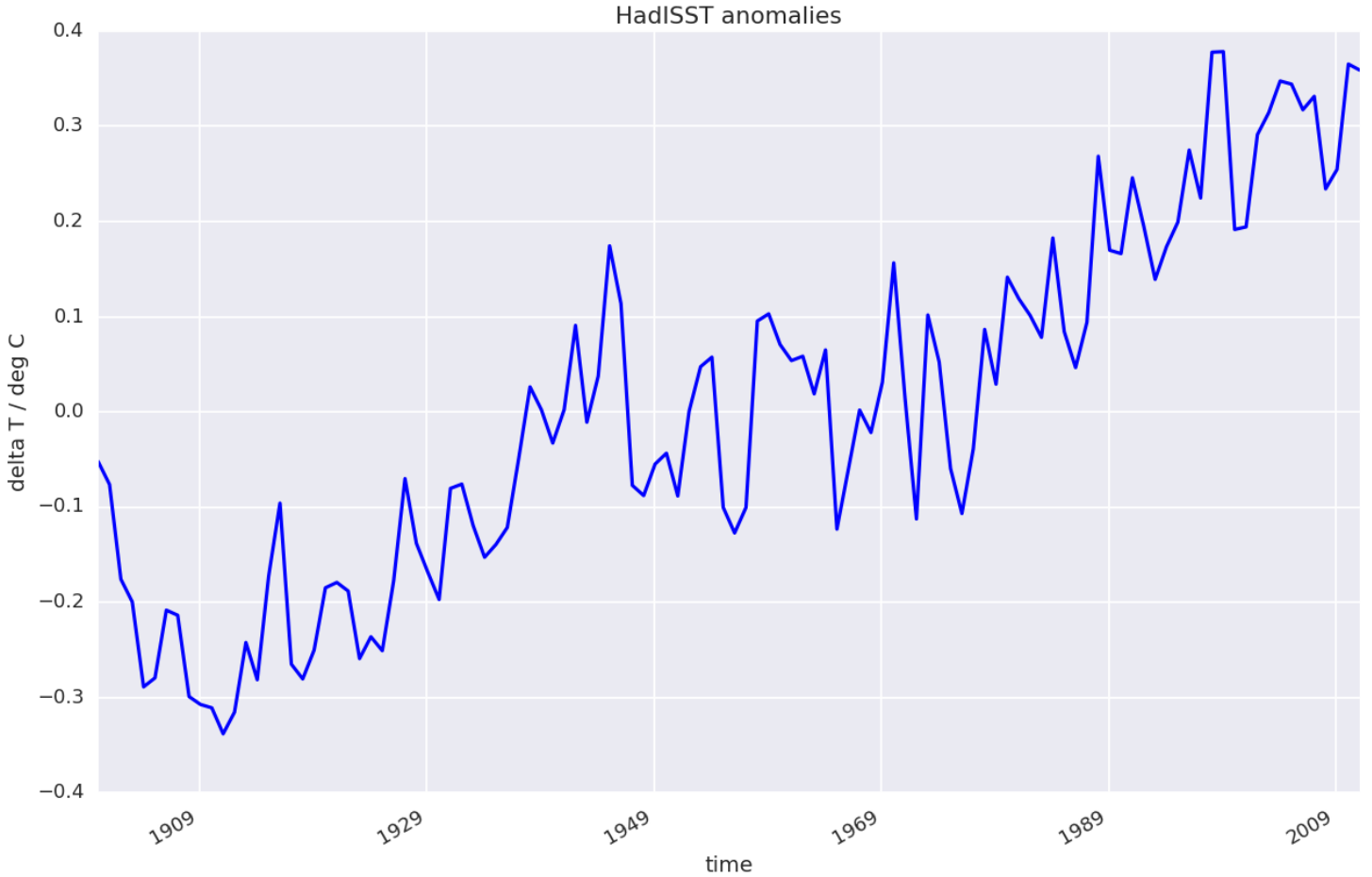


Figure 01. Annual-mean HadISST anomalies.

The Sloppy Way

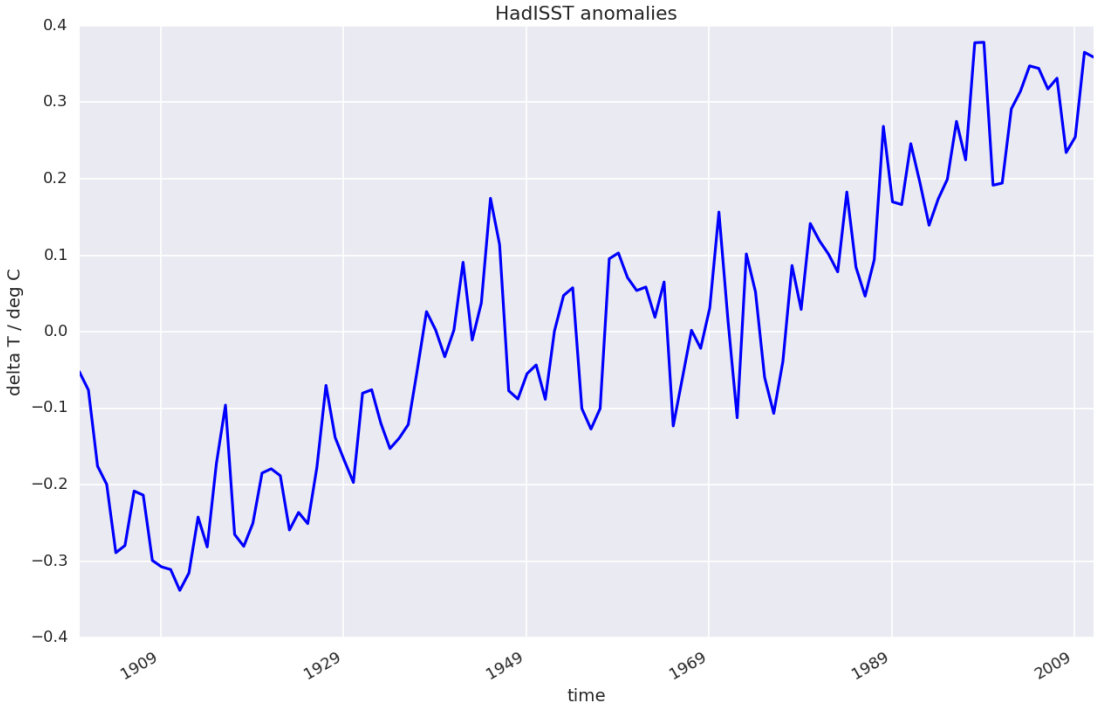


Figure 01. Annual-mean HadISST anomalies.

Giving More Details

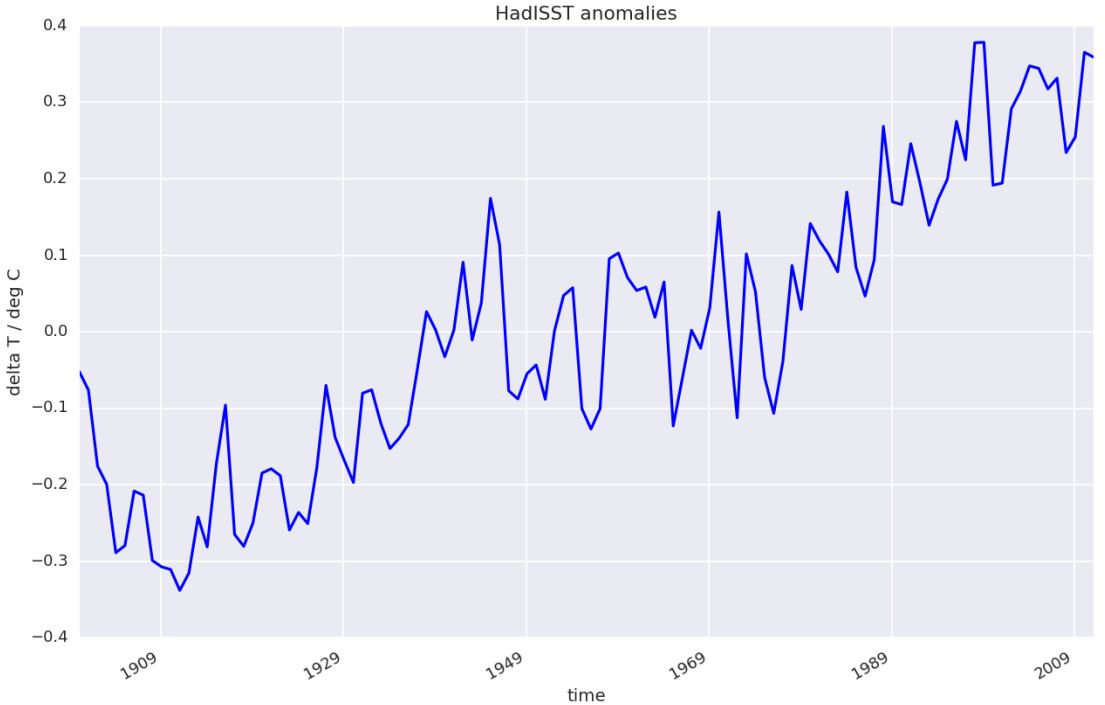


Figure 01. Global-mean and annual-mean HadISST anomalies relative to the full period from 1900 to 2010.

Towards Full Repeatability

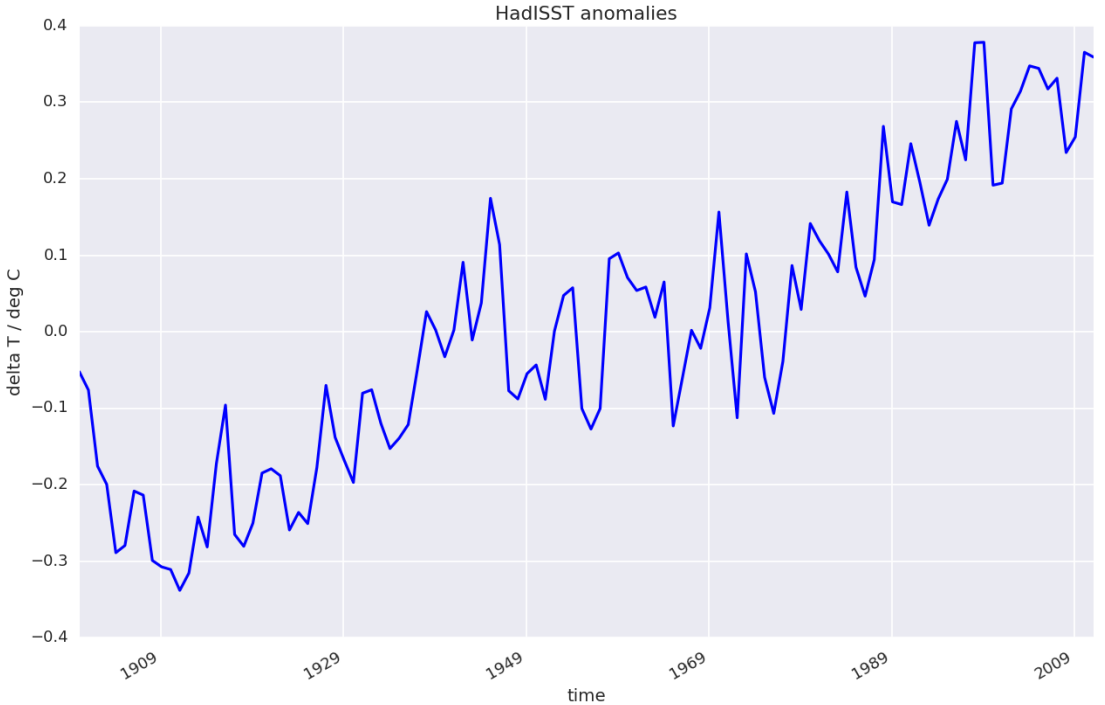


Figure 01. Global-mean and annual-mean HadISST anomalies relative to the full period from 1900 to 2010. There are a Jupyter notebook and a data file with all the details in the *supplementary materials*.

Towards Full Repeatability

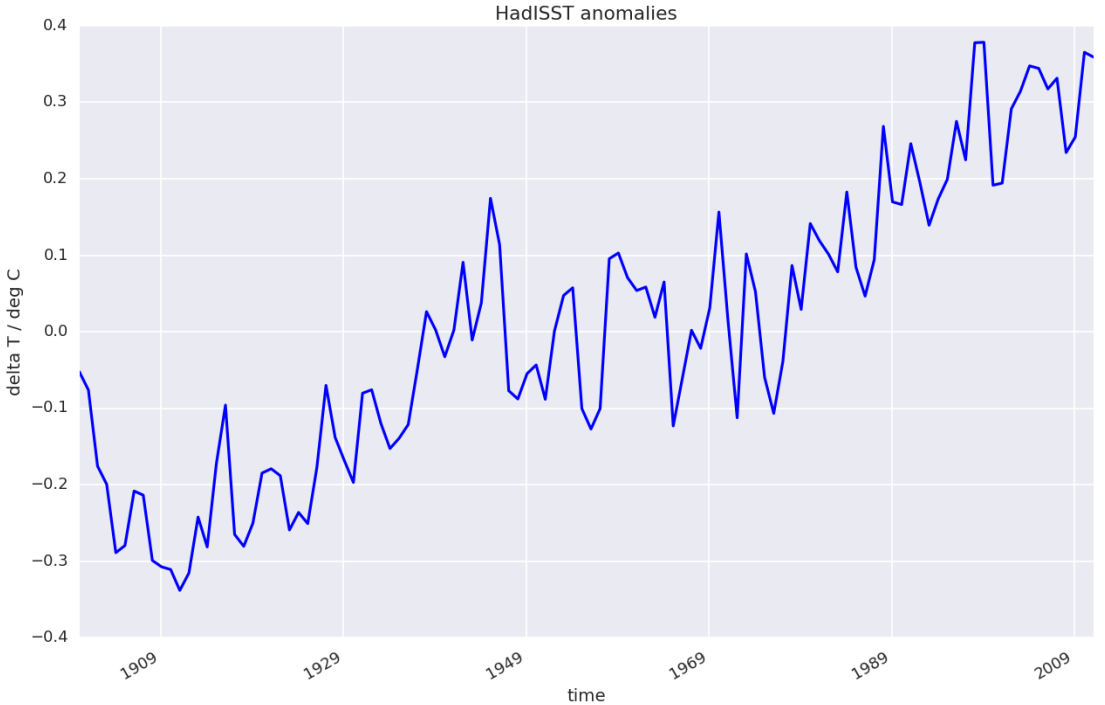


Figure 01. Global-mean and annual-mean HadISST anomalies relative to the full period from 1900 to 2010. There are a Jupyter notebook and a data file with all the details in the *supplementary materials*.

The Jupyter Notebook

```
import numpy as np
import xarray as xr

data_file = "/data/c2/TMdata/git_geomar_de_data/HadISST/v1.x.x/data/HadISST_sst.nc"

sst = xr.open_dataset(data_file).sst
sst = sst.sel(time=slice("1900-01-01", "2011-01-01"))
sst = sst.where(sst != -1000.0)

def weighted_global_mean(data):
    cosine_latitude = np.cos(np.pi / 180.0 * data.coords["latitude"])
    data = ((cosine_latitude * data).sum(dim=["latitude", "longitude"])
            / (cosine_latitude + 0 * data).sum(dim=["latitude", "longitude"]))
    return data

def annual_mean(data):
    data = data.resample(time="12M").mean(dim="time")
    return data

def temporal_anomaly(data):
    data = data - data.mean("time")
    return data

sst_anomalies = temporal_anomaly( weighted_global_mean( annual_mean( sst )))

sst_anomalies.plot()
```

This code shows the essential parts of the analysis. The full notebook is here.

Saving the Plotted Data for Reference

```
[...]
output_data_set = xr.Dataset({"global_and_annual_mean_SST_anomalies": sst_anomalies})
output_data_set.to_netcdf(file_name)
[...]
```

Click here for the full notebook and here for the data file.

Raw Data

We use the HadISST data set from a repository of fully *version-controlled* data sets:

```
[...]
data_file = Path("/data/c2/TMdata/git_geomar_de_data/HadISST/v1.x.x/data/HadISST_sst.nc")
[...]
```

From within the notebook, we find out that at the time of the analysis, `HadISST v1.3.0` was the latest of the `v1.x.x` versions:

```
git --work-tree="/data/c2/TMdata/git_geomar_de_data/HadISST/v1.x.x/" describe
```

```
v1.3.0
```

Tools and Libraries

Within the Jupyter notebook, we list the *complete* Python *environment* that was activated during the analysis:

```
conda list
```

```
# packages in environment at /home/wrath/TM/software/miniconda3_20170727/envs/py3_std:
#
alabaster                 0.7.10                   py35_1    conda-forge
anaconda-client           1.6.5                     py_0    conda-forge
[...]
xarray-0.9.6-51           g25d1855                 <pip>
xz                        5.2.3                        0    conda-forge
yaml                      0.1.6                        0    conda-forge
zeromq                    4.2.1                        1    conda-forge
zict                      0.1.3                     py_0    conda-forge
zlib                      1.2.8                        3    conda-forge
```

… reproduce the env by feeding this list back to `conda`.

Evolution of the Analysis

The development of the analysis (and of this talk) was tracked on Gitlab.com.

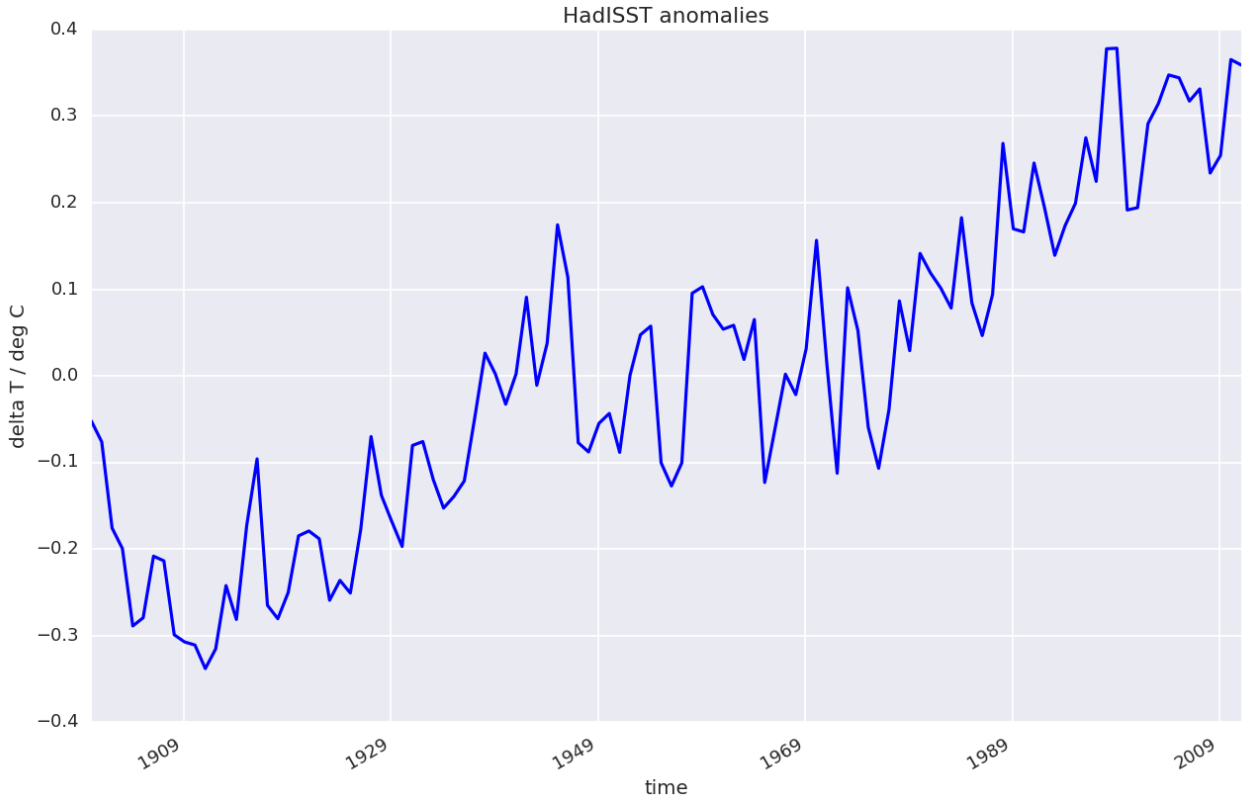
To see how it developed in time, check:
https://gitlab.com/willirath/towards_reproducible_science/commits/master

Suppose, this was a multi-author paper. Then, it would be easy

- to *return* to any *earlier version* of the scripts at any later point, or
- to *compare* scripts between *revisions* sent to the journal.

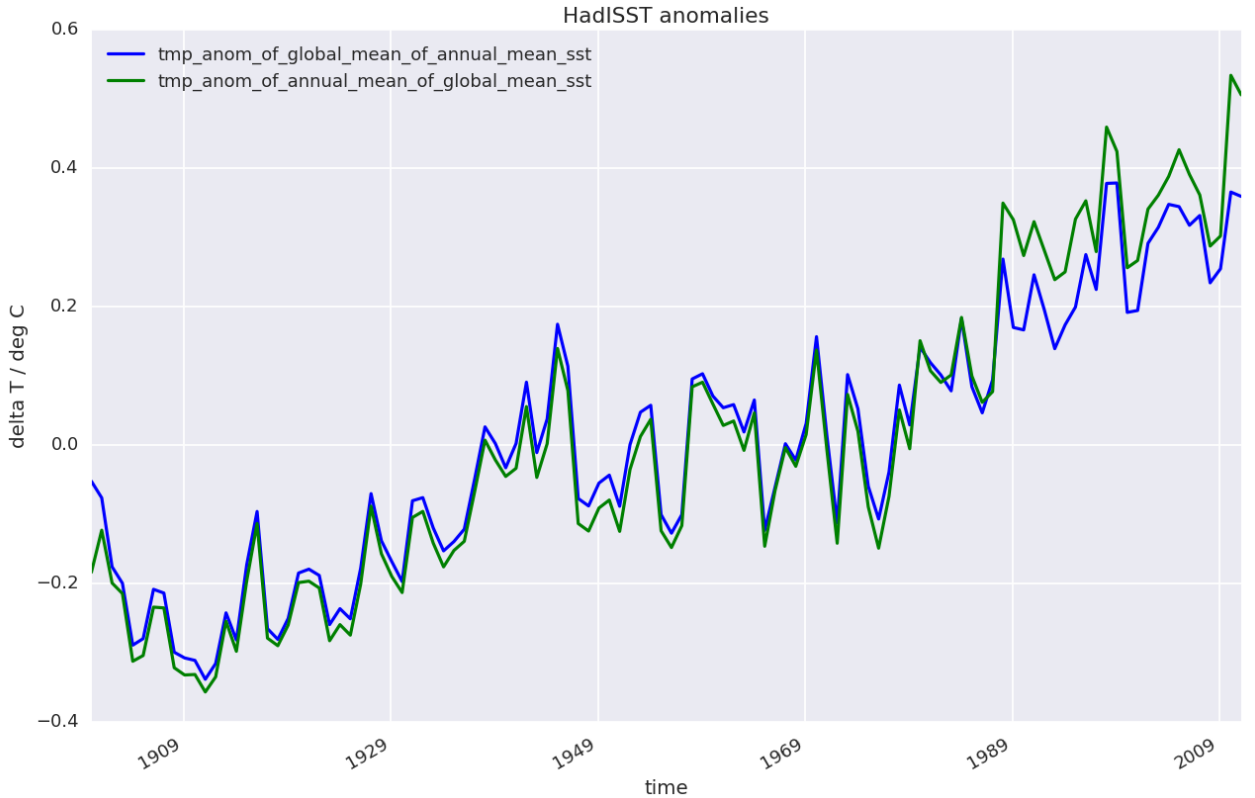
Interlude

Let's compare different ways to calculate the SST anomalies:



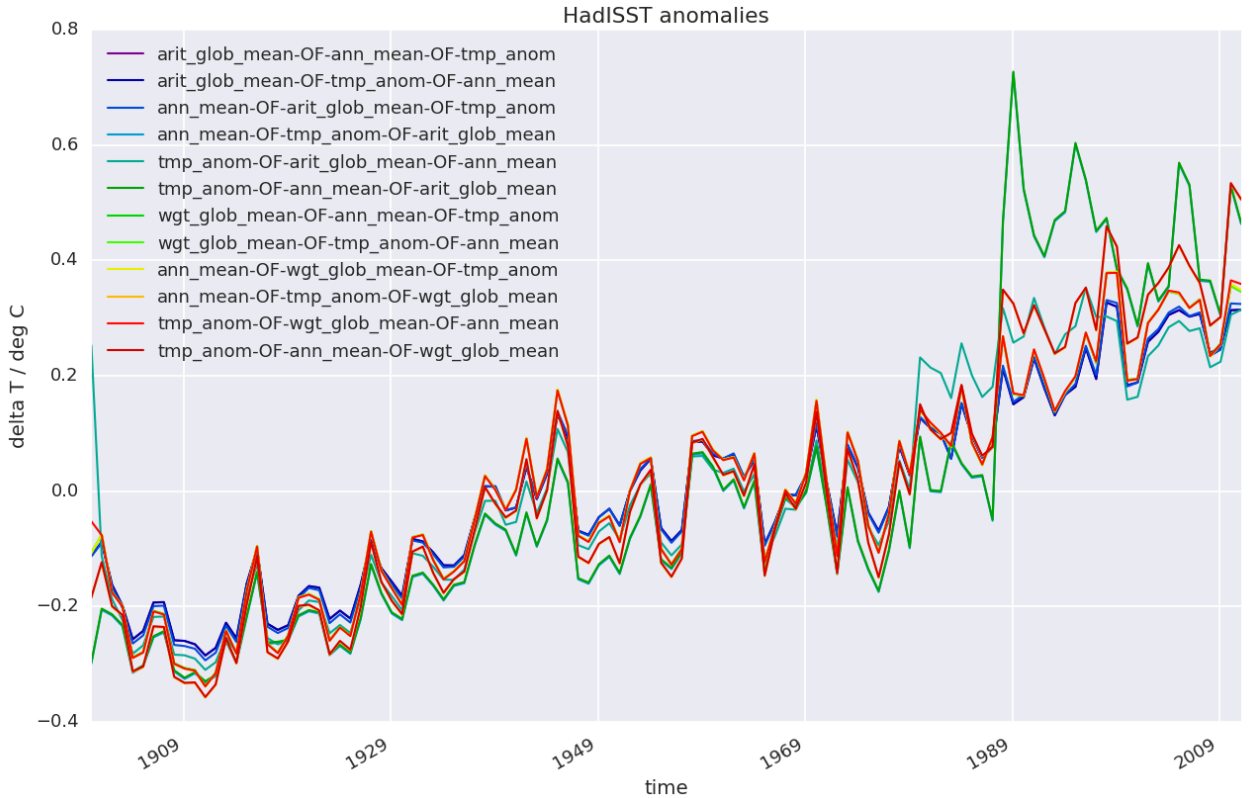
Interlude - Two Lines

This notebook details a subtlety with the *order* of averaging:



Interlude - Twelve Lines

This notebook adds arithmetic averages and shows all 12 variants:



... note that we're still weighting all months equally.

Building Repeatable Work Flows

1. Provide a data set containing ***all the numbers*** necessary to re-plot and compare the data presented in the analysis.
2. Provide fully ***documented steps*** from the original data to the final presentation.
3. Provide an overview of all the ***tools*** and ***libraries*** used in the analysis and of their exact versions.
4. Provide a pointer to the full ***raw data*** used in the analysis.
5. Provide a full ***time line*** of the development of the analysis. ← *That's more of a bonus.*

Building Repeatable Work Flows

1. *all the numbers*
2. *documented steps*
3. *tools & libraries*
4. *raw data*
5. *time line*

Building Repeatable Work Flows

1. *all the numbers* ← already required by many journals
2. *documented steps* ← already required by many journals
3. *tools & libraries*
4. *raw data*
5. *time line*

Building Repeatable Work Flows

1. *all the numbers* ← already required by many journals
2. *documented steps* ← already required by many journals
3. *tools & libraries*
4. *raw data* ← already required by *some* journals
5. *time line*

Building Repeatable Work Flows

1. *all the numbers* ← already required by many journals
2. *documented steps* ← already required by many journals
3. *tools & libraries* ← be prepared for those!
4. *raw data* ← already required by *some* journals
5. *time line* ← be prepared for those!

"all the numbers"

Generally speaking:

- *"checkpoints"* allowing for repetition of parts of an analysis
- data contained in ***figures*** or ***tables*** allowing for later comparison
- ...

"all the numbers" ← data.geomar.de

- *Stable* point of *first contact* for anybody looking for a dataset from Geomar
- talk to datamanagement(at)geomar.de

Alternatives:

- https://zenodo.org — storage and a DOI for data
- https://www.pangaea.de/ — storage and a DOI for geo-referenced data
- https://figshare.com/ — general supplementary-materials platform
- …
- At TM, we have data-tm(at)geomar.de


cc-by-sa/2.0 - Stepping stones across Afon... by David Purchase - geograph.org.uk/p/5134739

"documented steps"

Generally speaking:

- *script* your analyses / *avoid* un-documented *interactive* work
- use *consistent naming* ← think: `figure_01.m`, `figure_01.mat`, `figure_01.png`, `figure_01.log`

Consider Jupyter Notebooks.

- keep documentation, discussion, code and figures in one place
- very convenient to share an analysis
- available on all "our" *large machines* ← nb.geomar.de

CCo-licenced

"tools & libraries"

Generally speaking:

- keep track of versions of software
- actively *decide* which version to use
- prefer *stable* environments

Conda Environments

- Anaconda (*Python* and *R*) and Conda-Forge (far beyond)
- *identical* working environments *across different machines*

CCo-licenced

"raw data"

- *Referring* to data is hard !
- Work towards a *"single source of truth"* and a *clear* (central?) *structure* .
- Plan for ***evolution*** of each data set right from the start !
- ...

"raw data" ← git.geomar.de/data/

> *"We used v1.3.0 of the HadISST data set from our internal mirror."*

- fully *version controlled* data sets ← Git LFS
- *growing* collection of external data sets (*today* ≃ 1 TB)
- available on in-house computers and on external data centers
- available on Geomar *thredds server*

… to learn more, check: https://git.geomar.de/data/docs/

CCo-licenced

“time line”

- commented overview of evolution of scripts etc.
- tracking when (and *why*!) sth. was done
- often solved by *“version control”*

"time line" ← git.geomar.de

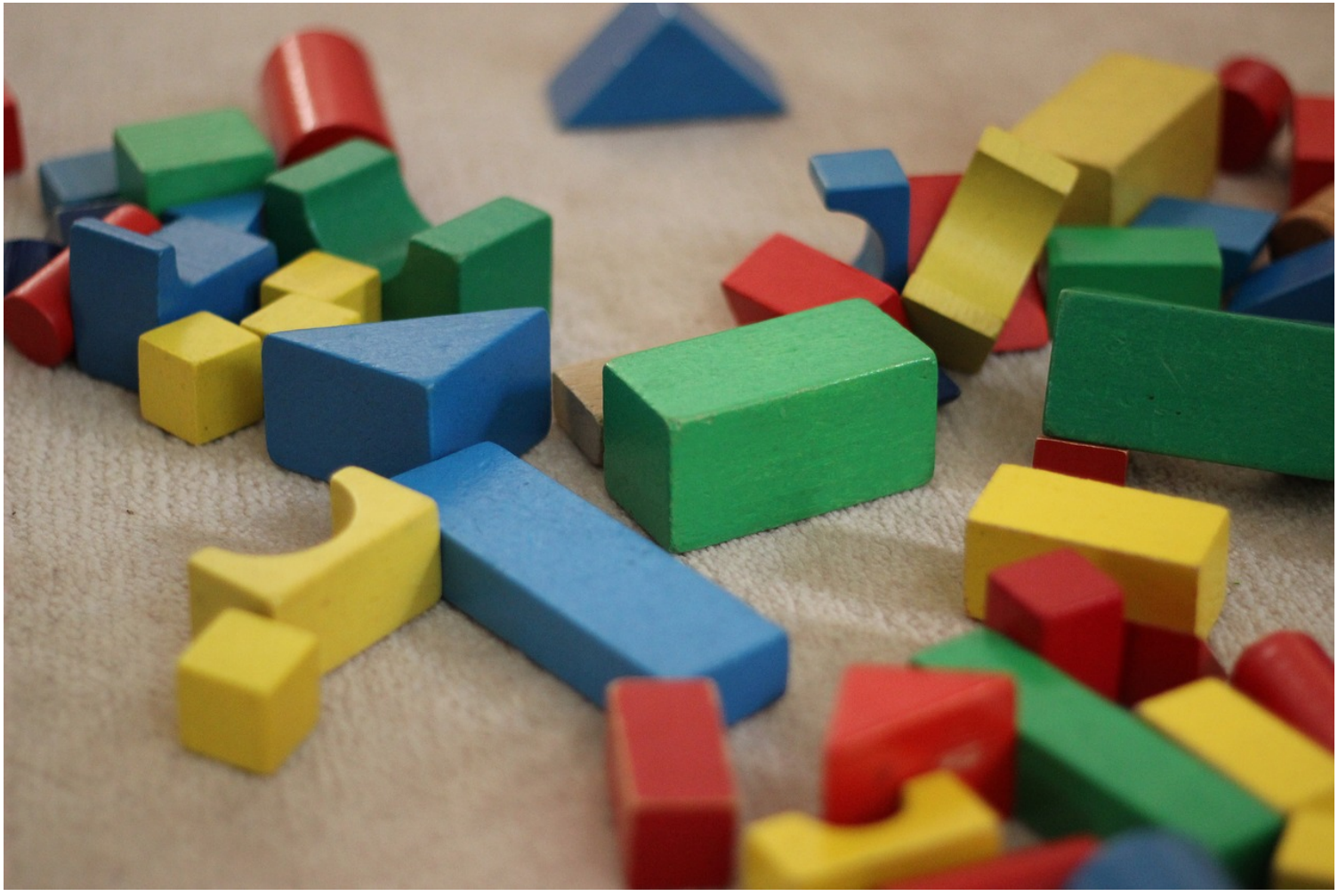
- *full-blown* version-control environment
- for *Geomar members* and for *external collaborators*
- easy *project management* and *collaboration*
- Easy-to-search *archive* comes for free !

Alternatives:

- https://github.com ← *the* industry standard
- https://gitlab.com ← evolving competitor
- SVN ← if joining an existing work flow

Repeatable Workflows ← at Geomar

1. *all the numbers* ← https://data.geomar.de
2. *documented steps* ← https://nb.geomar.de
3. *tools & libraries* ← https://git.geomar.de/python/conda_environments/
4. *raw data* ← https://git.geomar.de/data/docs
5. *time line* ← https://git.geomar.de



CCo-licenced

What Can *You* Do Now?

- Have a *mental framework* for repeatability.
- Talk to *each other*.
- *Script* all your analyses. / *Avoid undocumented interactive* work.
- Use a *version-control system*, in your *daily routine* work.
- Keep *track* of *your data*.
- Ping the data-management team, me, colleagues who might know …

What Can *You* Do Now?

- Have a *mental framework* for repeatability. ← this talk
- Talk to *each other*. ← this talk ?
- *Script* all your analyses. / *Avoid undocumented interactive* work.
- Use a *version-control system*, in your *daily routine* work. ← Git.
- Keep *track* of *your data*.
- Ping the data-management team, me, colleagues who might know …

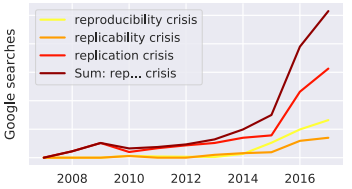
What Can *We* Do Now?

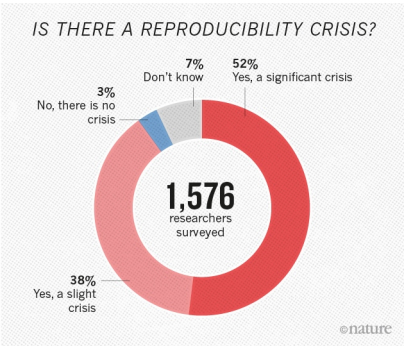
Develop ***Culture:***

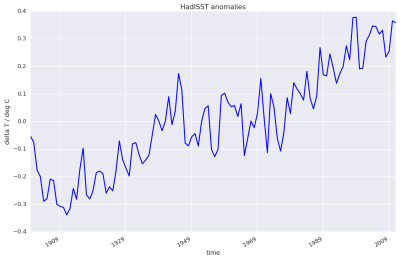
- Be *confident to publish* code and data.
- Develop *ethics* of using code and data published by others.
- *"Nobody is entitled to demand technical support for freely provided code."*

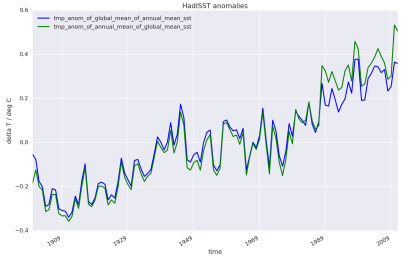
Develop ***Best Practices:***

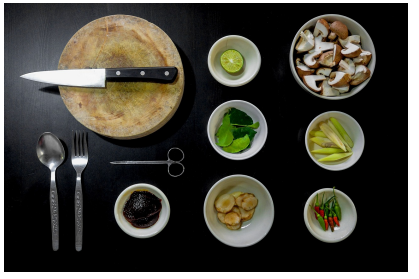
- *How much* to document?
- *Where* to document?
- What to *expect* from collaborators?

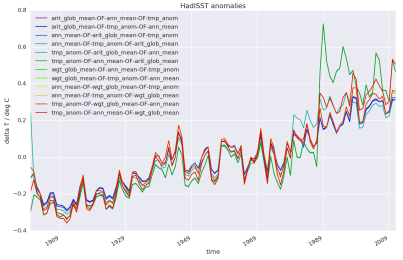

reproducibility crisis
replicability crisis
replication crisis
Sum: rep... crisis
Google searches
2008
2010
2012
2014
2016

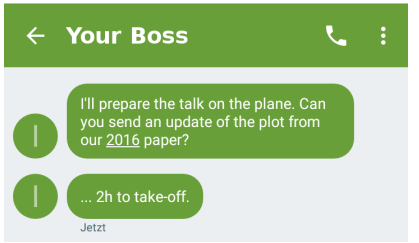

IS THERE A REPRODUCIBILITY CRISIS?
7%
Don't know
52%
Yes, a significant crisis
3%
No, there is no crisis
1,576
researchers surveyed
38%
Yes, a slight crisis
©nature


HadISST anomalies
delta T / deg C
time


HadISST anomalies
tmp_anom_of_global_mean_of_annual_mean_sst
tmp_anom_of_annual_mean_of_global_mean_sst
delta T / deg C
time




HadISST anomalies
delta T / deg C
time


Your Boss
I'll prepare the talk on the plane. Can you send an update of the plot from our 2016 paper?
... 2h to take-off.
Jetzt

Resources for Repeatable Work Flows

Cheat Sheets:

- Sandve (2013) has the *"10 Repeatability Commandments"*.
- Wilson (2012) has a reference sheet to *be prepared for coding*.

Resources at Geomar:

- Geomar Git server: https://git.geomar.de
- central Jupyter notebook server:
  - https://nb.geomar.de
  - https://git.geomar.de/python/doc/blob/master/nb_user_guide.md
- central data repository:
  - start here: https://git.geomar.de/data/docs/
  - and: https://data.geomar.de/thredds/catalog/tmdata/git_geomar_de_data/catalog.html
- conda environments:
  - our standard envs: https://git.geomar.de/python/conda_environments/
  - Conda Forge: https://conda-forge.org/
  - Anaconda: https://www.anaconda.com/distribution/
- publish data on: https://data.geomar.de
- how to number versions:
  - for data: https://git.geomar.de/data/docs/blob/master/versioning.md
  - semantic versioning (the meaning behing `v12.14.7` et al.): http://semver.org

Nice tools and software:

- Anaconda's Python distribution: https://www.anaconda.com/distribution/
- Conda Forge: https://conda-forge.org/
- Jupyter notebooks: https://github.com/jupyter/jupyter

Supplementaries from this Talk

- Notebook for the Google-Trends Figure 00:
  https://nbviewer.jupyter.org/url/willirath.gitlab.io/towards_reproducible_science/notebooks/fig_00_google_trends.ipynb
- Data for the Google-Trends Figure 00:
  https://willirath.gitlab.io/towards_reproducible_science/data/fig_00_google_trends.csv
- Notebook for Figure 01:
  https://nbviewer.jupyter.org/url/willirath.gitlab.io/towards_reproducible_science/notebooks/fig_01_HadISST_global_and_annual_mean_SST_anomalies.ipynb
- Data for Figure 01:
  https://willirath.gitlab.io/towards_reproducible_science/data/fig_01_HadISST_global_and_annual_mean_SST_anomalies.nc
- Notebook for Figure 02:
  https://nbviewer.jupyter.org/url/willirath.gitlab.io/towards_reproducible_science/notebooks/fig_02_HadISST_global_and_annual_mean_SST_anomalies_two_variants.ipynb
- Data for Figure 02:
  https://willirath.gitlab.io/towards_reproducible_science/data/fig_02_HadISST_global_and_annual_mean_SST_anomalies_two_variants.nc
- Notebook for Figure 03:
  https://nbviewer.jupyter.org/url/willirath.gitlab.io/towards_reproducible_science/notebooks/fig_03_HadISST_global_and_annual_mean_SST_anomalies_all_variants.ipynb
- Data for Figure 03:
  https://willirath.gitlab.io/towards_reproducible_science/data/fig_03_HadISST_global_and_annual_mean_SST_anomalies_all_variants.nc
- HadISST data set:
  https://git.geomar.de/data/HadISST/
- Vollkorn typeface:
  http://vollkorn-typeface.com/

Reading List

- *"1,500 scientists lift the lid on reproducibility"*: https://www.nature.com/news/1-500-scientists-lift-the-lid-on-reproducibility-1.19970
- *"Publish your computer code: it is good enough"* : https://www.nature.com/news/2010/101013/full/467753a.html
- *"Open code for open science?"* : http://www.nature.com/ngeo/journal/v7/n11/full/ngeo2283.html
- *"Why bitwise reproducibility matters"* : https://khinsen.wordpress.com/2015/01/07/why-bitwise-reproducibility-matters/
- *"Which mistakes do we actually make in scientific code?"* : http://blog.khinsen.net/posts/2017/05/04/which-mistakes-do-we-actually-make-in-scientific-code/
- *"A Minimum Standard for Publishing Computational Results in the Weather and Climate Sciences"* : http://journals.ametsoc.org/doi/full/10.1175/BAMS-D-15-00010.1
- *"Good Scientific Practice at MPI-M"* : http://www.mpimet.mpg.de/en/science/publications/good-scientific-practice.html
- *"Nature - Code share"* : https://www.nature.com/news/code-share-1.16232
- *"Ten Simple Rules for Reproducible Computational Research"* : http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1003285
- *"Best Practices for Scientific Computing"* : https://arxiv.org/abs/1210.0530
- *"Most computational hydrology is not reproducible, so is it really science?"*: http://onlinelibrary.wiley.com/doi/10.1002/2016WR019285/full
    - first comment: http://onlinelibrary.wiley.com/doi/10.1002/2016WR020190/full
    - first reply: http://onlinelibrary.wiley.com/doi/10.1002/2017WR020480/full
    - second comment: http://onlinelibrary.wiley.com/doi/10.1002/2016WR020208/full
    - second reply: http://onlinelibrary.wiley.com/doi/10.1002/2017WR020476/full