Improving reproducibility of geospatial conference papers

Lessons learned from a first implementation of reproducibility reviews

Daniel Nüst (AGILE 2020 Reproducibility Chair), Frank O. Ostermann, Carlos Granell, Alexander Kmoch (all Reproducibility Committee)

https://doi.org/10.7557/5.5601

AGILE Association of Geographic Information Laboratories in Europe

REPRODUCIBLE AGILE

https://agile-online.org/

AGILE council | annual conference | PhD schools | initiatives
GIScience teaching/research @ European research agendas

https://reproducible-agile.github.io/

Workshops on reproducibility in 2017, 2018, 2019

Reproducible publications at AGILE conferences Initiative in 2019: guidelines AGILE reproducibility review 2020

AGILE Reproducible Paper Guidelines: Contents & First Revision

AGILE Reproducible Paper Guidelines

https://doi.org/10.17605/OSF.IO/CB7Z8

Created by AGILE Initiative in 2019, see report at https://osf.io/hupxr/

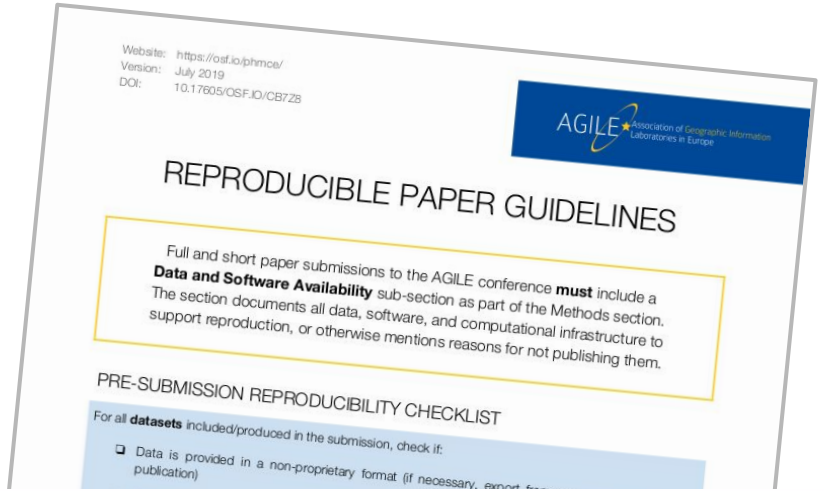
Transparency & Reproducibility

GIScience

https://osf.io/phmce/wiki/home/

Promotion

Acknowledge spectrum


Website: https://osf.io/phmce/
Version: July 2019
DOI: 10.17605/OSF.IO/CB7Z8

AGILE Association of Geographic Information Laboratories in Europe

REPRODUCIBLE PAPER GUIDELINES

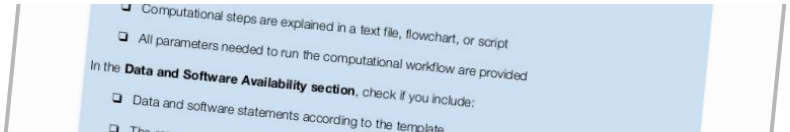
Full and short paper submissions to the AGILE conference **must** include a **Data and Software Availability** sub-section as part of the Methods section. The section documents all data, software, and computational infrastructure to support reproduction, or otherwise mentions reasons for not publishing them.

PRE-SUBMISSION REPRODUCIBILITY CHECKLIST

For all **datasets** included/produced in the submission, check if:

❑ Data is provided in a non-proprietary format (if necessary, export from publication)

Full and short paper submissions to the AGILE conference **must** include a **Data and Software Availability** sub-section as part of the Methods section. The section documents all data, software, and computational infrastructure to support reproduction, or otherwise mentions reasons for not publishing them.


❑ Computational steps are explained in a text file, flowchart, or script

❑ All parameters needed to run the computational workflow are provided

In the **Data and Software Availability section**, check if you include:

❑ Data and software statements according to the template

The guidelines

https://doi.org/10.17605/OSF.IO/CB7Z8

Author guidelines

Data in Research Papers
Computational workflows
in Research Papers
Pre-submission checklist
Writing DASA section

Rationale/Motivation/Vision

Reviewer guidelines

Reproducibility reviewer guidelines (WIP)

Website: https://osf.io/phmce/
Version: July 2019
DOI: 10.17605/OSF.IO/CB7Z8

AGILE Association of Geographic Information Laboratories in Europe

REPRODUCIBLE PAPER GUIDELINES

Full and short paper submissions to the AGILE conference **must** include a **Data and Software Availability** sub-section as part of the Methods section. The section documents all data, software, and computational infrastructure to support reproduction, or otherwise mentions reasons for not publishing them.

PRE-SUBMISSION REPRODUCIBILITY CHECKLIST

For all **datasets** included/produced in the submission, check if:

- ❑ Data is provided in a non-proprietary format (if necessary, export from proprietary format for publication)
- ❑ Data is documented (at least description of collection query and field or column names, ideally using complete metadata following established standards)
- ❑ Data is accessible in a public repository
- ❑ Data has a clear licence

For any **software tool/library/package** used or produced, check if:

- ❑ Computational environment (including hardware) is documented or provided in the most appropriate format given its complexity
- ❑ The versions of relevant software components (libraries, packages) are provided
- ❑ Software is available in a public repository
- ❑ Software has a clear license
- ❑ Computational steps are explained in a text file, flowchart, or script
- ❑ All parameters needed to run the computational workflow are provided

In the **Data and Software Availability section**, check if you include:

- ❑ Data and software statements according to the template
- ❑ The reasons, if any, for not being able to share (parts of) data or code.

For properly **acknowledging data and software** by both you and others check that:

- ❑ All datasets and code used or mentioned are cited throughout the paper and included in the references with DOIs.

The guidelines for reproducibility reviewers (WIP)

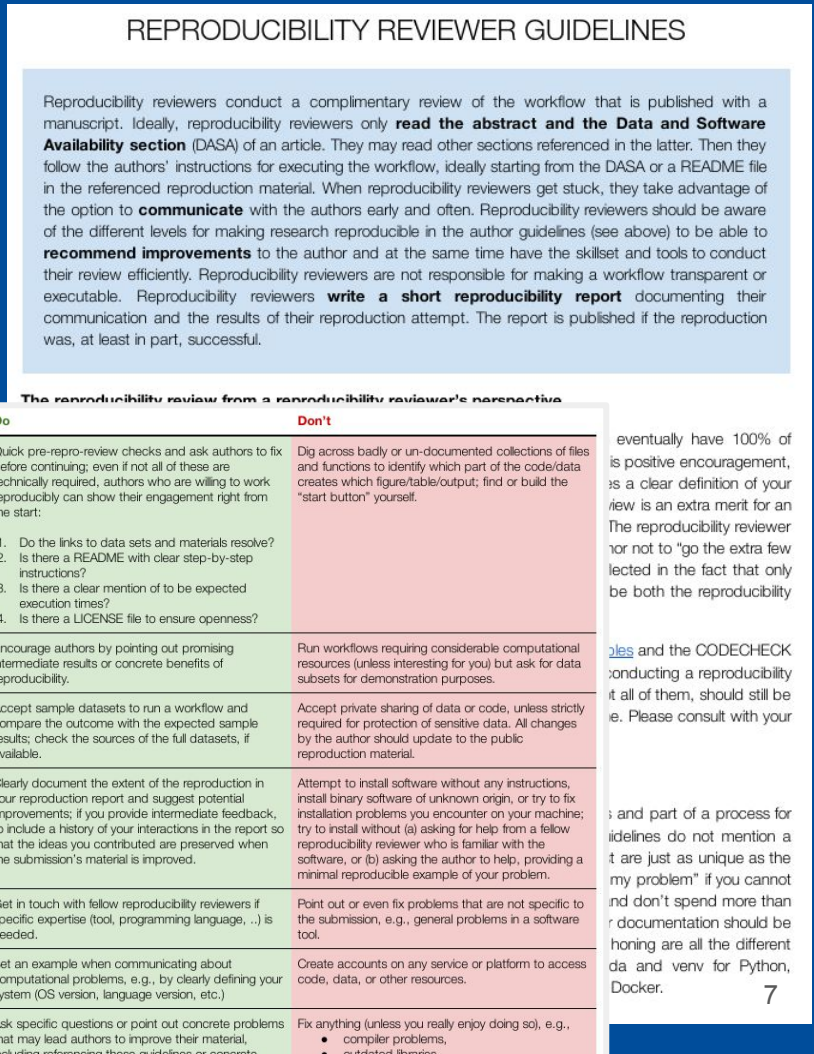
Ideal vs. realistic

Role

Skills

Do's & dont's

https://docs.google.com/document/d/1Kc-ToUVcrdsq6aB8Qy2J_rIluFwDniv6GHGtZuPvIEo/edit#



REPRODUCIBILITY REVIEWER GUIDELINES

Reproducibility reviewers conduct a complimentary review of the workflow that is published with a manuscript. Ideally, reproducibility reviewers only **read the abstract and the Data and Software Availability section** (DASA) of an article. They may read other sections referenced in the latter. Then they follow the authors' instructions for executing the workflow, ideally starting from the DASA or a README file in the referenced reproduction material. When reproducibility reviewers get stuck, they take advantage of the option to **communicate** with the authors early and often. Reproducibility reviewers should be aware of the different levels for making research reproducible in the author guidelines (see above) to be able to **recommend improvements** to the author and at the same time have the skillset and tools to conduct their review efficiently. Reproducibility reviewers are not responsible for making a workflow transparent or executable. Reproducibility reviewers **write a short reproducibility report** documenting their communication and the results of their reproduction attempt. The report is published if the reproduction was, at least in part, successful.

The reproducibility review from a reproducibility reviewer's perspective

Do	Don't
Quick pre-repro-review checks and ask authors to fix before continuing; even if not all of these are technically required, authors who are willing to work reproducibly can show their engagement right from the start: 1. Do the links to data sets and materials resolve? 2. Is there a README with clear step-by-step instructions? 3. Is there a clear mention of to be expected execution times? 4. Is there a LICENSE file to ensure openness?	Dig across badly or un-documented collections of files and functions to identify which part of the code/data creates which figure/table/output; find or build the "start button" yourself.
Encourage authors by pointing out promising intermediate results or concrete benefits of reproducibility.	Run workflows requiring considerable computational resources (unless interesting for you) but ask for data subsets for demonstration purposes.
Accept sample datasets to run a workflow and compare the outcome with the expected sample results; check the sources of the full datasets, if available.	Accept private sharing of data or code, unless strictly required for protection of sensitive data. All changes by the author should update to the public reproduction material.
Clearly document the extent of the reproduction in your reproduction report and suggest potential improvements; if you provide intermediate feedback, to include a history of your interactions in the report so that the ideas you contributed are preserved when the submission's material is improved.	Attempt to install software without any instructions, install binary software of unknown origin, or try to fix installation problems you encounter on your machine; try to install without (a) asking for help from a fellow reproducibility reviewer who is familiar with the software, or (b) asking the author to help, providing a minimal reproducible example of your problem.
Get in touch with fellow reproducibility reviewers if specific expertise (tool, programming language, ..) is needed.	Point out or even fix problems that are not specific to the submission, e.g., general problems in a software tool.
Set an example when communicating about computational problems, e.g., by clearly defining your system (OS version, language version, etc.)	Create accounts on any service or platform to access code, data, or other resources.
Ask specific questions or point out concrete problems that may lead authors to improve their material,	Fix anything (unless you really enjoy doing so), e.g., • compiler problems,

eventually have 100% of
is positive encouragement,
s a clear definition of your
iew is an extra merit for an
The reproducibility reviewer
or not to "go the extra few
lected in the fact that only
be both the reproducibility

les and the CODECHECK
conducting a reproducibility
t all of them, should still be
e. Please consult with your

and part of a process for
idelines do not mention a
t are just as unique as the
my problem" if you cannot
nd don't spend more than
documentation should be
honing are all the different
da and venv for Python,
Docker.

Reproducibility Review at AGILE Conference 2020

Review process

Proceedings:
https://www.agile-giscience-series.net/review_process.html

Process documentation:
https://osf.io/7rjpe/

Reproducibility review *after* accept/reject decisions, triggered by regular reviewer

Reproducibility review & communication

Community conference & coronavirus

Badges on proceedings page

Presentation at conference

Reproducibility review results

6 reproducibility reports published

16 not possible/not attempted

(5 of which after communication with authors):

- **no starting point** in the paper
- **documentation insufficient** for third party
- **sensitive/confidential/commercial data**
- **proprietary** software
- software paper
- (conceptual papers)

Reproducibility review of: Integrating cellular automata and discrete global grid systems: a case study into wildfire modelling
Nüst
Reproduction report and material.

Reproducibility review of: Extracting interrogative intents and concepts from geo-analytic questions
Nüst
Reproduction report and material.

Reproducibility review of: Tracking Hurricane Dorian in GDELT and Twitter
Ostermann & Nüst
Reproduction report and material.

Reproducibility Review of: Comparing supervised learning algorithms for Spatial Nominal Entity recognition
Ostermann & Nüst
Reproduction report and material.

Reproducibility review of: Window Operators for Processing Spatio-Temporal Data Streams on Unmanned Vehicles
Nüst & Ostermann
Reproduction report and material.

Reproducibility review of: What to do in the Meantime: A Service Coverage Analysis for Parked Autonomous Vehicles
Nüst & Granell
Reproduction report and material.

https://osf.io/6k5fh/

Reproducibility review reports


Reproducibility review of: Integrating cellular automata and discrete global grid systems: a case study into wildfire modelling

Daniel Nüst

2020-06-20

REPRODUCIBLE AGILE

This report is part of the reproducibility review at the AGILE conference. For more information see https://reproducible-agile.github.io/. This document is published on OSF at https://osf.io/ztc7m/. To cite the report use:

Nüst, D. (2020, June 5): Reproducibility review of: Integrating cellular automata and discrete global grid systems: a case study into wildfire modelling. https://doi.org/10.17605/OSF.IO/ZTC7M

Reviewed paper

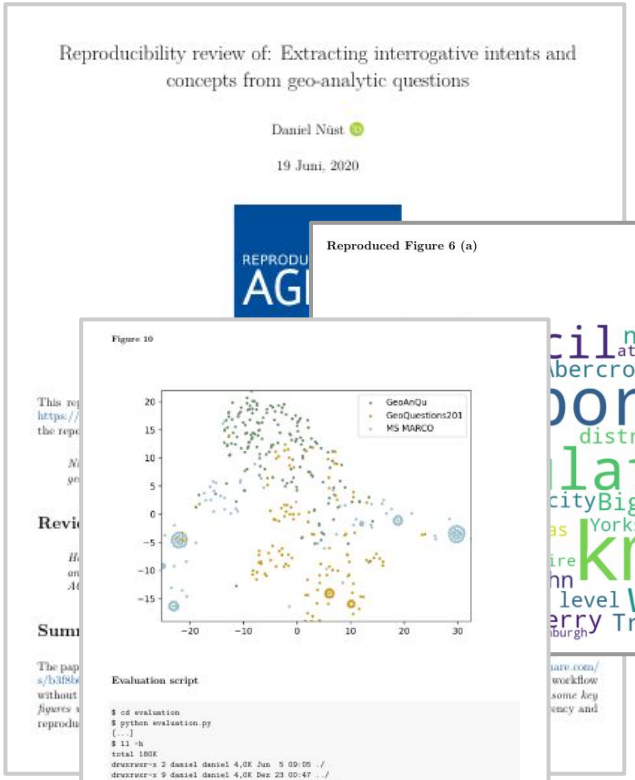
Hojati, Majid and Roberts a case study into wildfire yax-1-5-2020, 2020.

Summary

The paper code and a sample da working Binder link. The workf the workflow does not create a the repository while further da *successful*, the authors demonst therefore include some commen authors after sending them a fi

wind_0.3_50.png

model-output-example.png


Reproducibility review of: Extracting interrogative intents and concepts from geo-analytic questions

Daniel Nüst

19 Juni, 2020

REPRODU AGI

This rep https:// the repo

Revie

Summ

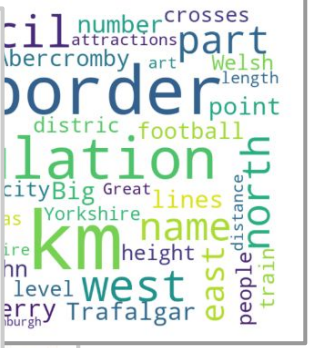
The pap s/b3f8b without *figures* reprodu

Reproduced Figure 6 (a)

Figure 10

Evaluation script

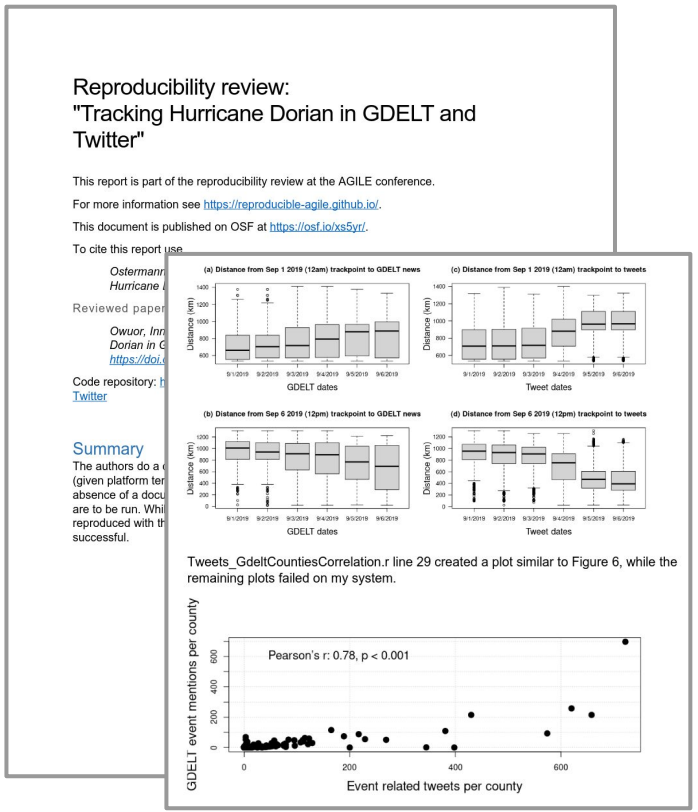
```
$ cd evaluation
$ python evaluation.py
[...]
$ ll -h
total 180K
drwxrwxr-x 2 daniel daniel 4,0K Jun  5 09:05 ./
drwxrwxr-x 9 daniel daniel 4,0K Dez 23 00:47 ../
-rw-rw-r-- 1 daniel daniel  22K Dez 23 00:47 1.csv
-rw-rw-r-- 1 daniel daniel  21K Dez 23 00:47 2.csv
-rw-rw-r-- 1 daniel daniel  22K Dez 23 00:47 3.csv
-rw-rw-r-- 1 daniel daniel  21K Dez 23 00:47 4.csv
```



https://doi.org/10.17605/OSF.IO/ZTC7M

https://doi.org/10.17605/OSF.IO/7XRQG

Reproducibility review reports



Reproducibility review:
"Tracking Hurricane Dorian in GDELT and Twitter"

This report is part of the reproducibility review at the AGILE conference.

For more information see https://reproducible-agile.github.io/.

This document is published on OSF at https://osf.io/xs5yr/.

To cite this report use

Ostermann
Hurricane

Reviewed paper

Owuor, Inn
Dorian in G
https://doi.

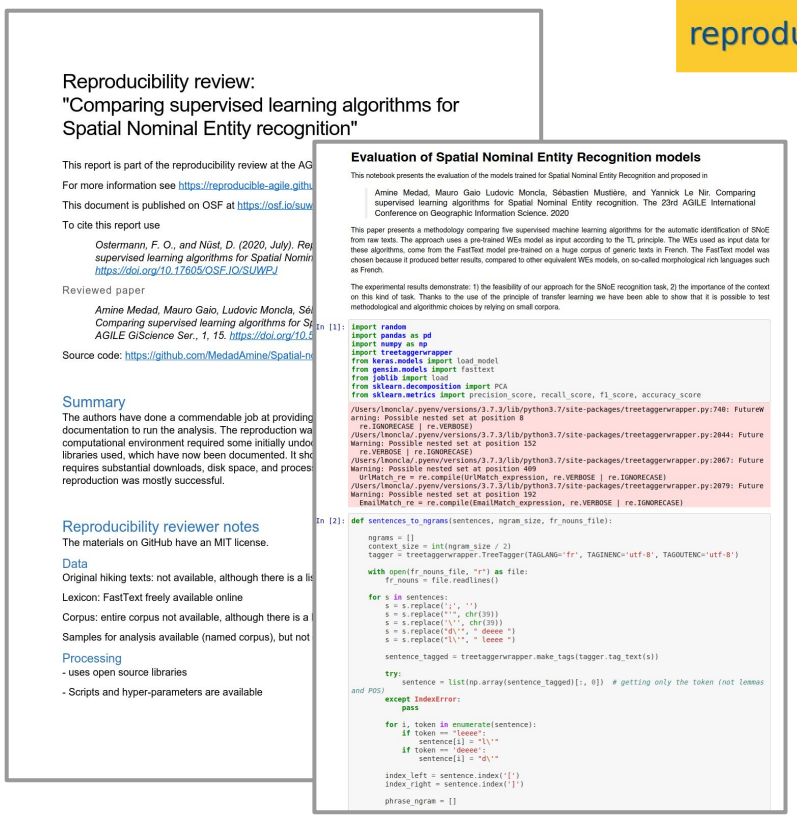
Code repository: h
Twitter

Summary

The authors do a
(given platform ter
absence of a docu
are to be run. Whi
reproduced with th
successful.

Tweets_GdeltCountiesCorrelation.r line 29 created a plot similar to Figure 6, while the remaining plots failed on my system.

https://doi.org/10.17605/OSF.IO/XS5YR



Reproducibility review:
"Comparing supervised learning algorithms for Spatial Nominal Entity recognition"

This report is part of the reproducibility review at the AG

For more information see https://reproducible-agile.githu

This document is published on OSF at https://osf.io/suw

To cite this report use

Ostermann, F. O., and Nüst, D. (2020, July). Re
supervised learning algorithms for Spatial Nomin
https://doi.org/10.17605/OSF.IO/SUWPJ

Reviewed paper

Amine Medad, Mauro Gaio, Ludovic Moncla, Sé
Comparing supervised learning algorithms for Sp
AGILE GiScience Ser., 1, 15. https://doi.org/10.5

Source code: https://github.com/MedadAmine/Spatial-n

Summary

The authors have done a commendable job at providing
documentation to run the analysis. The reproduction wa
computational environment required some initially undo
libraries used, which have now been documented. It sh
requires substantial downloads, disk space, and proces
reproduction was mostly successful.

Reproducibility reviewer notes

The materials on GitHub have an MIT license.

Data

Original hiking texts: not available, although there is a li

Lexicon: FastText freely available online

Corpus: entire corpus not available, although there is a

Samples for analysis available (named corpus), but not

Processing

- uses open source libraries

- Scripts and hyper-parameters are available

Evaluation of Spatial Nominal Entity Recognition models

This notebook presents the evaluation of the models trained for Spatial Nominal Entity Recognition and proposed in

> Amine Medad, Mauro Gaio Ludovic Moncla, Sébastien Mustière, and Yannick Le Nir. Comparing supervised learning algorithms for Spatial Nominal Entity recognition. The 23rd AGILE International Conference on Geographic Information Science. 2020

This paper presents a methodology comparing five supervised machine learning algorithms for the automatic identification of SNoE from raw texts. The approach uses a pre-trained WEs model as input according to the TL principle. The WEs used as input data for these algorithms, come from the FastText model pre-trained on a huge corpus of generic texts in French. The FastText model was chosen because it produced better results, compared to other equivalent WEs models, on so-called morphological rich languages such as French.

The experimental results demonstrate: 1) the feasibility of our approach for the SNoE recognition task, 2) the importance of the context on this kind of task. Thanks to the use of the principle of transfer learning we have been able to show that it is possible to test methodological and algorithmic choices by relying on small corpora.

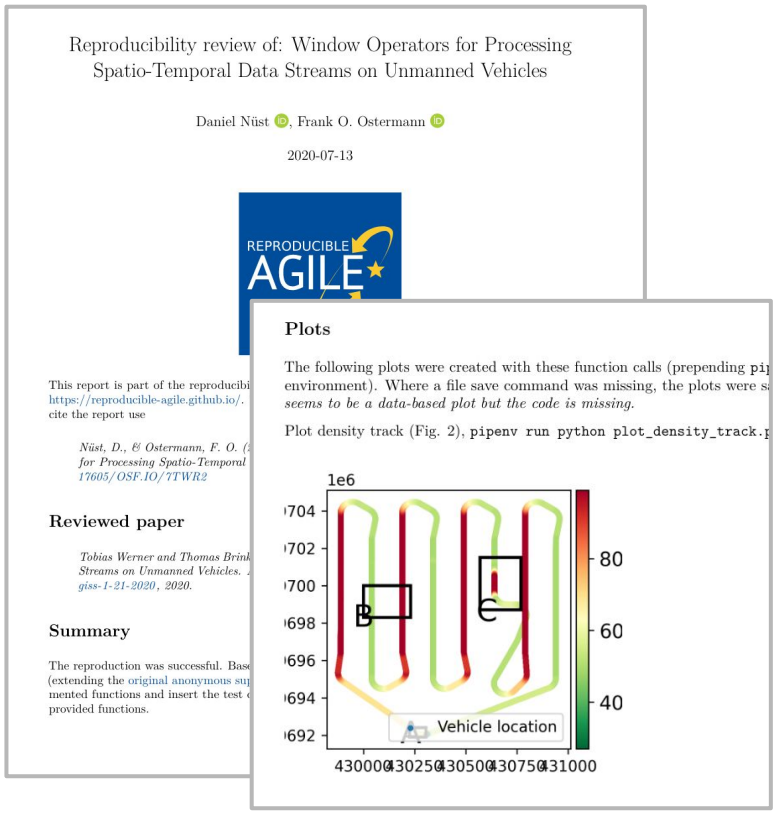
In [1]:
```
import random
import pandas as pd
import numpy as np
import treetaggerwrapper
from keras.models import load_model
from gensim.models import fasttext
from joblib import load
from sklearn.decomposition import PCA
from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score
```

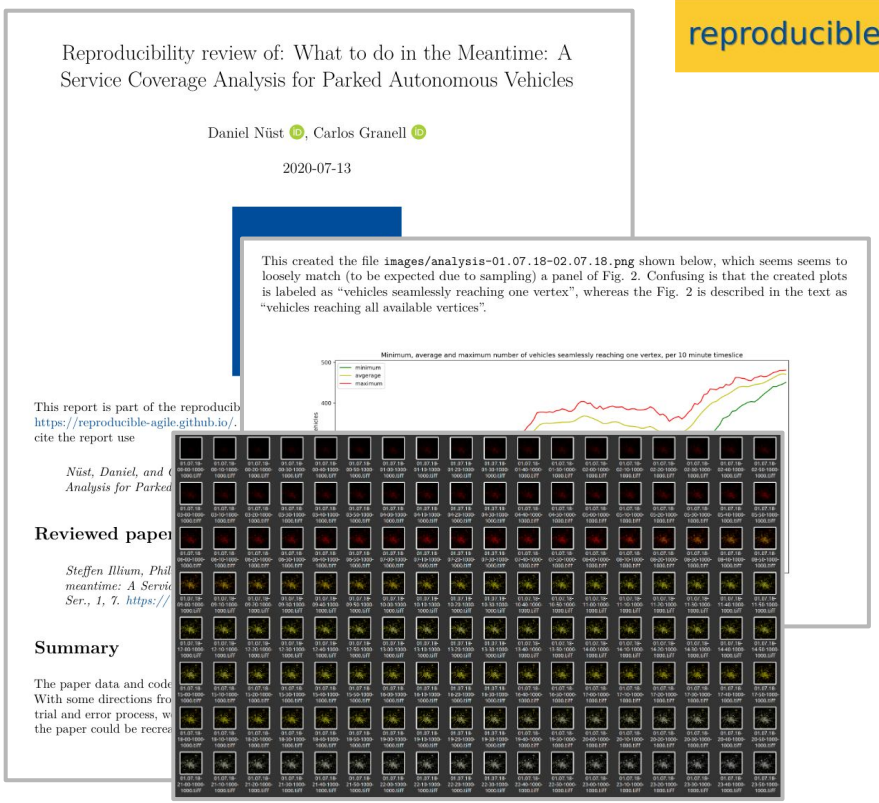
```
/Users/lmoncla/.pyenv/versions/3.7.3/lib/python3.7/site-packages/treetaggerwrapper.py:740: FutureW
arning: Possible nested set at position 8
  re.IGNORECASE | re.VERBOSE)
/Users/lmoncla/.pyenv/versions/3.7.3/lib/python3.7/site-packages/treetaggerwrapper.py:2044: Future
Warning: Possible nested set at position 152
  re.VERBOSE | re.IGNORECASE)
/Users/lmoncla/.pyenv/versions/3.7.3/lib/python3.7/site-packages/treetaggerwrapper.py:2067: Future
Warning: Possible nested set at position 409
  UrlMatch_re = re.compile(UrlMatch_expression, re.VERBOSE | re.IGNORECASE)
/Users/lmoncla/.pyenv/versions/3.7.3/lib/python3.7/site-packages/treetaggerwrapper.py:2079: Future
Warning: Possible nested set at position 192
  EmailMatch_re = re.compile(EmailMatch_expression, re.VERBOSE | re.IGNORECASE)
```

In [2]:
```
def sentences_to_ngrams(sentences, ngram_size, fr_nouns_file):

    ngrams = []
    context_size = int(ngram_size / 2)
    tagger = treetaggerwrapper.TreeTagger(TAGLANG='fr', TAGINENC='utf-8', TAGOUTENC='utf-8')

    with open(fr_nouns_file, "r") as file:
        fr_nouns = file.readlines()

    for s in sentences:
        s = s.replace(':', '')
        s = s.replace('"', chr(39))
        s = s.replace('\'', chr(39))
        s = s.replace("d\'", " deeee ")
        s = s.replace("l\'", " leeee ")

        sentence_tagged = treetaggerwrapper.make_tags(tagger.tag_text(s))

        try:
            sentence = list(np.array(sentence_tagged)[:, 0])  # getting only the token (not lemmas
and POS)
        except IndexError:
            pass

        for i, token in enumerate(sentence):
            if token == "leeee":
                sentence[i] = "l\'"
            if token == 'deeee':
                sentence[i] = "d\'"

        index_left = sentence.index('[')
        index_right = sentence_index(']')

        phrase_ngram = []
```

https://doi.org/10.17605/OSF.IO/SUWPJ

Reproducibility review reports





https://doi.org/10.17605/OSF.IO/7TWR2

https://doi.org/10.17605/OSF.IO/5SVMT

Independent execution of computations underlying research articles.

CODECHECK Register

Certificate	Repository	Type	Issue	Report	Check date
2020-001	codecheckers/Piccolo-2020	journal (GigaScience)	NA	http://doi.org/10.5281/zenodo.3674056	2019-02-14
2020-002	codecheckers/Reproduction-Hancock	community	2	http://doi.org/10.5281/zenodo.3750741	2020-04-13
2020-003	codecheckers/Hopfield-1982	community	1	https://doi.org/10.5281/zenodo.3741797	2020-04-06
2020-004	codecheckers/Barto-Sutton-Anderson-1983	community	4	https://doi.org/10.5281/zenodo.3827371	2020-05-14
2020-005	codecheckers/Larisch-reproduction	community	5	https://doi.org/10.5281/zenodo.3959175	2020-07-23
2020-006	codecheckers/Detorakis-reproduction	community	6	https://doi.org/10.5281/zenodo.3948353	2020-07-16
2020-007	codecheckers/Hathway-Goodman-2018	community	7	NA	NA
2020-008	codecheckers/covid-uk	community (preprint)	8	http://doi.org/10.5281/zenodo.3746024	2020-04-09
2020-009	codecheckers/2020-cov-tracing	community (preprint)	9	http://doi.org/10.5281/zenodo.3767060	2020-04-26
2020-010	codecheckers/covid-report9	community (preprint)	14	https://doi.org/10.5281/zenodo.3865491	2020-05-29
2020-011	codecheckers/covid19model-nature	community (in press)	18	https://doi.org/10.5281/zenodo.3893138	2020-06-13
2020-012	codecheckers/covid19model-report23	community (preprint)	19	https://doi.org/10.5281/zenodo.3893617	2020-06-14
2020-013	codecheckers/Spitschan2020_bioRxiv	community (preprint)	20	https://doi.org/10.5281/zenodo.3947959	2020-07-14
2020-018	reproducible-agile/AGILECA	conference (AGILEGIS)	25	https://doi.org/10.17605/OSF.IO/ZTC7M	2020-07-13
2020-019	5SVMT	conference (AGILEGIS)	25	https://doi.org/10.17605/OSF.IO/5SVMT	2020-07-13
2020-020	7TWR2	conference (AGILEGIS)	25	https://doi.org/10.17605/OSF.IO/7TWR2	2020-07-13
2020-021	reproducible-agile/Spatial-nominal-entity-recognition	conference (AGILEGIS)	25	https://doi.org/10.17605/OSF.IO/SUWPJ	2020-07-13
2020-022	7XRQG	conference (AGILEGIS)	25	https://doi.org/10.17605/OSF.IO/7XRQG	2020-07-13
2020-023	reproducible-agile/Tracking-Hurricane-Dorian-in-GDELT-and-Twitter	conference (AGILEGIS)	25	https://doi.org/10.17605/OSF.IO/XS5YR	2020-07-13

https://codecheck.org.uk/register/

CODECHECK is a process for independent execution of computations underlying scholarly research articles.

Findings

Read full report at https://osf.io/7rjpe/

Overall

- Saw full ***spectrum*** of reproducibility
- Compared to previous years' submissions, the guidelines and increased community awareness markedly **improved reproducibility**
- ⅝ reproduced papers have DASA; all embrace guidelines
- Reproducibility reports with many **recommendations** for improvement, well received by authors, even included in revision before publication > reward!
- Good practices spread slowly
- Process

Findings

Read full report at https://osf.io/7rjpe/

Challenges for reproducibility reviewer:

- Inconsistencies (identifiers, links) between paper and code
- Lack of connections between artefacts (code <> figure)
- Workspaces layout: no documentation, absolute paths
- Unknown runtime and no demo subsets of data
- No guidance on efforts and stop points

All efforts beyond mere workflow execution

How to put your community on a path towards more reproducibility in 5 ~~easy~~ hard steps

1. Build a team of enthusiasts (workshop, social events)
2. Assess the current state and raise awareness (workshop, paper)
3. Institutional support (🙏 AGILE Council 🙏 + committee chairs)
4. Positive encouragement (no reproduction != bad science)
5. Keep at it!

Next steps

Do it again in 2021 🎉

Revise guidelines 🛠️🇮🇹🇫🇷🇨🇳

Grow reproducibility reviewer team

ECRs, credit @ ORCID, skills

Continue research 🕵️

Ostermann, F., Nüst, D., Granell, C., Hofer, B., & Konkol, M. (2020). *Reproducible Research and GIScience: an evaluation using GIScience conference papers*. EarthArXiv. https://doi.org/10.31223/x5zk5v

Continue community engagement towards opening scholarship

Scope

Requirements

Acceptance condition?

Open review if tenured

Format-free submission

...

Thank you!

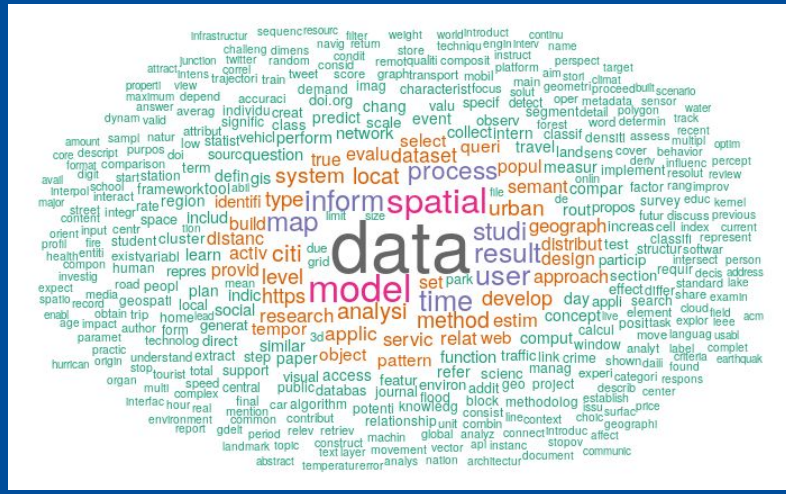
I look forward to your questions!
@nordhomen | d.n@wwu.de

Slides: **https://doi.org/10.7557/5.5601**

Reproducibility Committee 2020 + Initiative
Daniel Nüst (University of Münster, GER)
Frank Ostermann (University of Twente, NEL)
Carlos Granell (Universitat of Jaume I, ESP)
Alexander Kmoch (University of Tartu, EST)
Barbara Hofer (University of Salzburg, AUT)
Rusne Sileryte (TU Delft)
Markus Konkol (University of Twente, NEL)

REPRODUCIBLE AGILE

https://reproducible-agile.github.io/



Word-stem cloud of all AGILE 2020 submissions (full/short/poster)

Bonus slides

The guidelines for data

REPRODUCIBLE AGILE

DATA IN RESEARCH PAPERS

	Minimum	Ideal
What?	Publish all input data + data description / documentation	Publish all data and adhere to standardised, discipline-specific metadata[2] to describe your data
Where?	Use a data repository providing a DOI[3]	Use a discipline-specific repository[4] with a DOI
How?	Use open data formats + specify a license	Make your data FAIR (Findable, Accessible, Interoperable and Reusable) and as open as possible

"What if…" and Examples (not shown)

https://doi.org/10.17605/OSF.IO/CB7Z8

The guidelines for workflows

REPRODUCIBLE AGILE

Examples (not shown)

COMPUTATIONAL WORKFLOWS IN RESEARCH PAPERS

	Minimum	Intermediate	Ideal
What? Computational environment	Describe the environment and computational infrastructure, e.g. computer specs, operating system + software versions	Provide live documents (structured configuration files with dependency information, e.g. a Binder[9])	Provide the actual environment, e.g. a container created by a Dockerfile[10] or a Virtual Machine (VM, e.g. OSGeo-Live)
Computation steps	Document the detailed steps in a text file and/or flowchart (every action/click)	Provide scripts / models and a README file that explains their use	Provide a software package with structured metadata[11], tests/CI[12], and an automated workflow[13] + If applicable: Add link to running instance of software
Where?	Repository providing a DOI, such as Zenodo, OSF, b2share, or FigShare		Minimum + versioned code repository, such as GitHub or GitLab
How? Tools used	Use generally available proprietary tools (avoid tools that are not available to reviewers and other researchers)		Use (and create) open source tools; cite core modules/tools/language used, including your own
Development practices	Use clear licenses[14] that fit your environment	Follow "Good enough practices" for scientific computing software[15]	Use development guidelines for your environment / language of choice (e.g. for R[16])

https://doi.org/10.17605/OSF.IO/CB7Z8

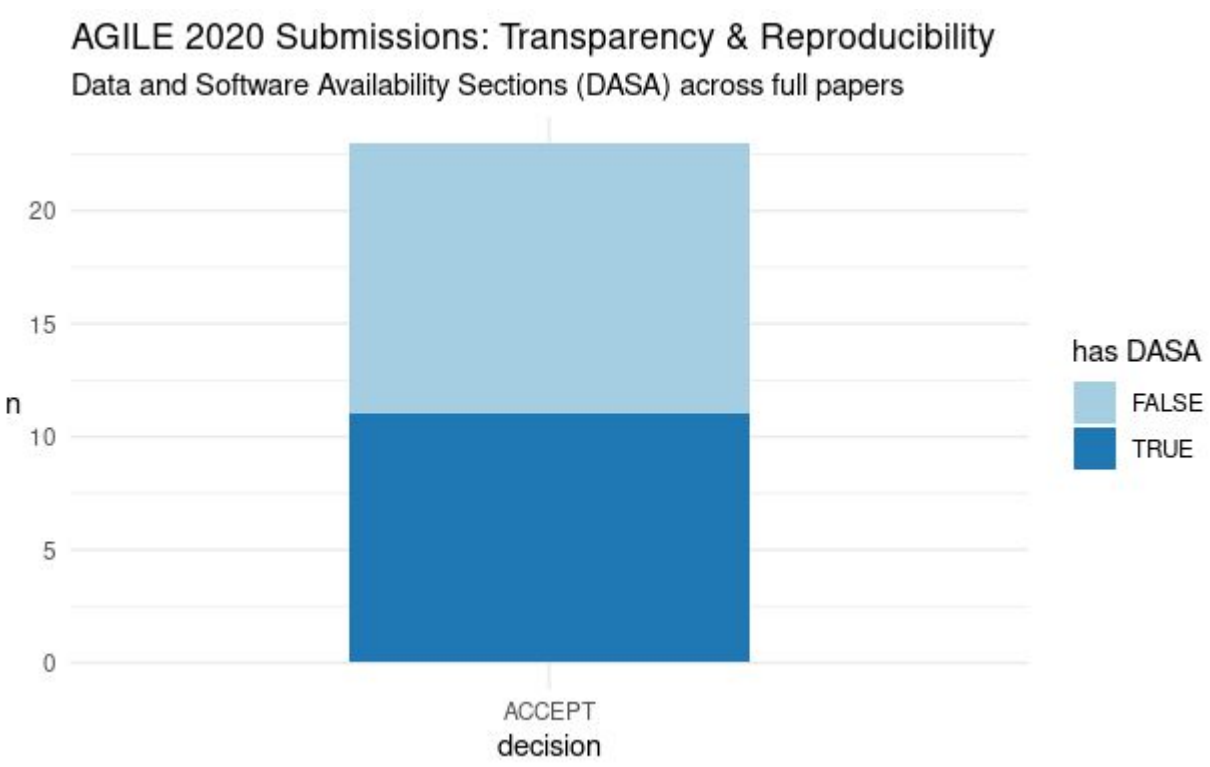
The guidelines for reproducibility reviewers (WIP)

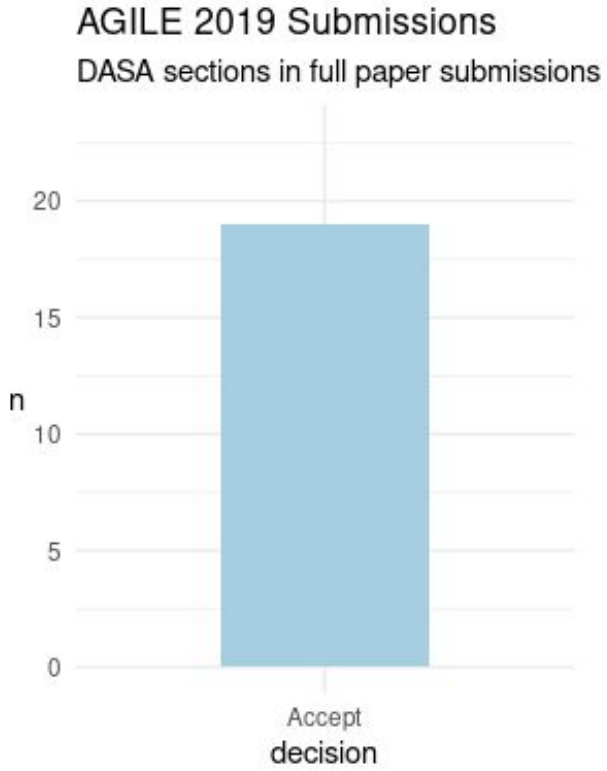
Examples for “Do’s and Don’ts”:

- Do shift burden to author
- Do encourage and set examples
- Do *not* accept private data sharing
- Document your work in report (impact)
- Be kind (career stage, knowledge, privileges)
- No rummaging

https://docs.google.com/document/d/1Kc-ToUVcrdsq6aB8Qy2J_rIluFwDniv6GHGtZuPvIEo/edit#

Do	Don't
Quick pre-repro-review checks and ask authors to fix before continuing; even if not all of these are technically required, authors who are willing to work reproducibly can show their engagement right from the start: 1. Do the links to data sets and materials resolve? 2. Is there a README with clear step-by-step instructions? 3. Is there a clear mention of to be expected execution times? 4. Is there a LICENSE file to ensure openness?	Dig across badly or un-documented collections of files and functions to identify which part of the code/data creates which figure/table/output; find or build the “start button” yourself.
Encourage authors by pointing out promising intermediate results or concrete benefits of reproducibility.	Run workflows requiring considerable computational resources (unless interesting for you) but ask for data subsets for demonstration purposes.
Accept sample datasets to run a workflow and compare the outcome with the expected sample results; check the sources of the full datasets, if available.	Accept private sharing of data or code, unless strictly required for protection of sensitive data. All changes by the author should update to the public reproduction material.
Clearly document the extent of the reproduction in your reproduction report and suggest potential improvements; if you provide intermediate feedback, to include a history of your interactions in the report so that the ideas you contributed are preserved when the submission’s material is improved.	Attempt to install software without any instructions, install binary software of unknown origin, or try to fix installation problems you encounter on your machine; try to install without (a) asking for help from a fellow reproducibility reviewer who is familiar with the software, or (b) asking the author to help, providing a minimal reproducible example of your problem.
Get in touch with fellow reproducibility reviewers if specific expertise (tool, programming language, ..) is needed.	Point out or even fix problems that are not specific to the submission, e.g., general problems in a software tool.
Set an example when communicating about computational problems, e.g., by clearly defining your system (OS version, language version, etc.)	Create accounts on any service or platform to access code, data, or other resources.
Ask specific questions or point out concrete problems that may lead authors to improve their material, including referencing these guidelines or concrete tools/methods that you already (!) know about, especially if you suspect that the author might now be familiar with them (e.g., version pinning/dependency management, absolute paths).	Fix anything (unless you really enjoy doing so), e.g., • compiler problems, • outdated libraries, • broken paths, or • Incomplete computing environment specifications, especially if the author can fix them even quicker.
Make sure that you are aware of any templates or specific resources provided for reproducibility reviewers from the reproducibility committee chair before starting your review.	
Consider the author’s background, career stage, and position to be aware of (a lack of) privileges or institutional power to decide how much support you provide and how you communicate; your reproducibility review can be a contribution to improve equity and inclusion in academia.	Be a bro.





0% **of rejected papers have a DASA section (correlation, not cause)**

48% **of accepted full papers have DASA section**

Reproducible research and GIScience: an evaluation using AGILE conference papers
https://doi.org/10.7717/peerj.5072