

README.md

A Tutorial on Reproducible Visualization Research

Velitchko Filipov, Tobias Isenberg, Alexander Lex

Part II

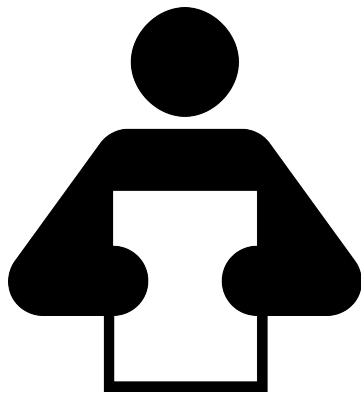
How to Approach Reproducible Research?

Tobias Isenberg

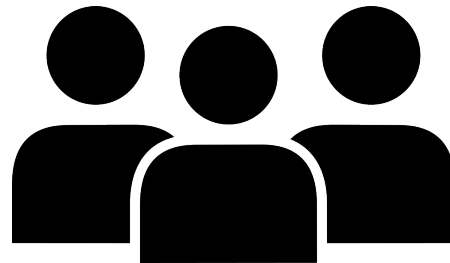
Reproducibility, but for whom?



yourself

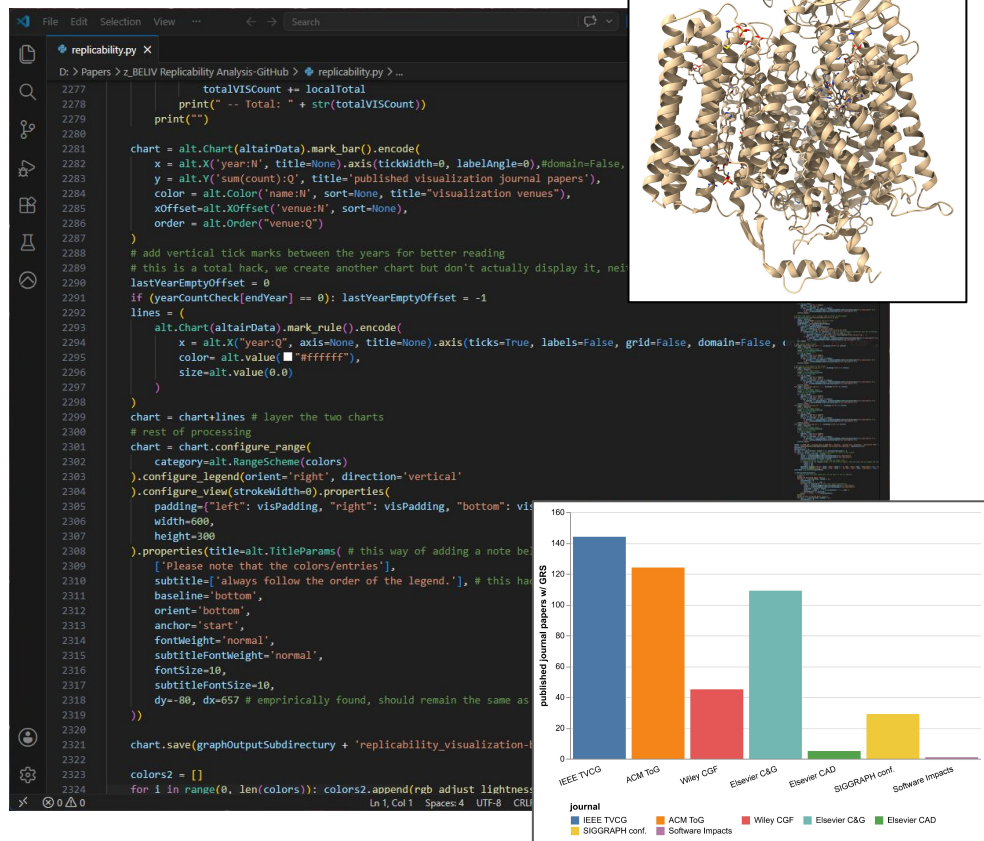


paper reviewers



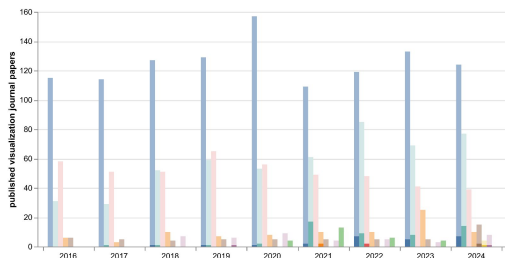
others later-on

Script everything!



Script everything! ➡ Data analysis

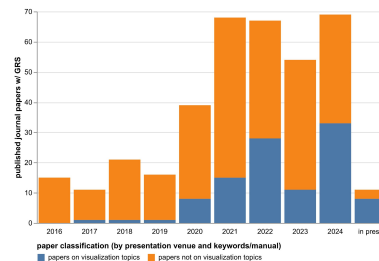
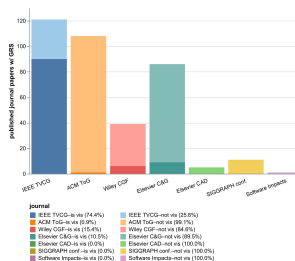
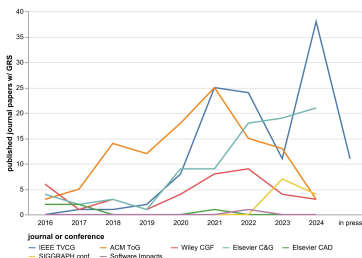
- primarily for yourself: fixing mistakes, update data, etc.
- create candidate examples to (potentially) be used in paper



visualization venues

- IEEE VIS: w/ GRS
- IEEE VIS: w/o GRS
- journal pres. @ IEEE VIS: w/ GRS
- journal pres. @ IEEE VIS: w/o GRS
- EuroVis: w/ GRS
- EuroVis: w/o GRS
- journal pres. @ EuroVis: w/ GRS
- journal pres. @ EuroVis: w/o GRS
- PacificVis TVCG: w/ GRS
- PacificVis TVCG: w/o GRS
- journal pres. @ PacificVis: w/ GRS
- journal pres. @ PacificVis: w/o GRS
- VCBM C&G: w/ GRS
- VCBM C&G: w/o GRS
- C&G special issue: w/ GRS
- C&G special issue: w/o GRS

Please note that the colors/entries always follow the order of the legend.



- use versioning/external repositories (also for backup)



GitLab

Script everything! ➡ Data visualization

- tools: Python, R, etc.
- many visualization libraries



- non-chart visualizations:
parameterizable command-line calls/APIs
- interactive tools:
still parameterizable command-line calls/APIs
(not-great alternative: textual descriptions of what to do)
- visual output can then directly be embedded into your papers
- scripts (+data) can serve as additional material for reviewers



Go to extremes: Injecting results into LaTeX

Problem: talking about your analysis results in the paper (“We found that 42 of our experiments were successful.”)

- use your script to inject the numbers into LaTeX!
`\newcommand{\KeyMetric}{42}`
- in your LaTeX file:
`\input{numbersFromScript.tex}`

```
#!/usr/bin/python3

latexOutputFile =
    "numbersFromScript.tex"
latexOutputString = ""

# ... compute keyMetric

latexOutputString +=
    "\\newcommand{\\KeyMetric}{\" +
    str(keyMetric) + "}\\n"

with open(latexOutputFile, "w") as
text_file:
    text_file.write(latexOutputString)
```

Go to extremes: Injecting results into LaTeX

Problem with LaTeX macros: cannot contain digits

```
# ... compute keyMetric
```

```
latexOutputString += "\\newcommand{\\KeyMetric}{\" + str(keyMetric) + \"}\\n\"
```

```
# ... macro with number in name (e.g., a year)
```

```
latexOutputString += "\\newcommand{\\KeyMetricIn\" + intToRoman(year) + \"}{\"  
    + str(keyMetric) + \"}\\n\"
```

resulting LaTeX input:

```
\newcommand{\KeyMetric}{42}
```

```
\newcommand{\KeyMetricInMMXXVI}{42}
```



code in our repo

Full paper example: <https://doi.org/10/nt3t> (including doing math via `pgf`)

Go to extremes: Injecting results into LaTeX

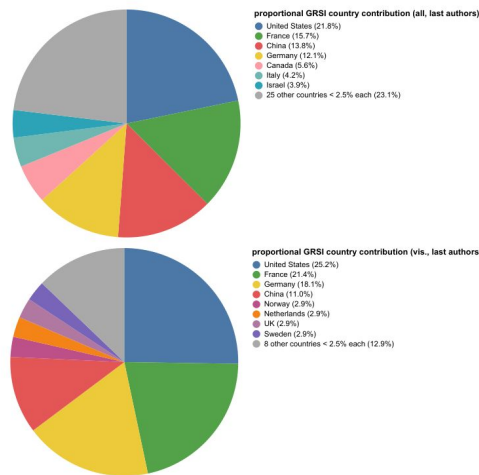


Figure 12: Country contribution of paper last authors to GRS awards. Top: overall; bottom: visualization papers. These plots are thresholded to 2.5% per-country contribution (for clarity), for full versions see Figure 24 and Figure 26 in Appendix C, respectively.

GRSI-awarded authors, the bottom one shows the version for only visualization-themed papers. We can see that four countries played the major role in both cases. For all papers with GRS, the United States has the largest proportion with 21.1%, followed by China with 15.3%, France with 15.1%, and Germany with 11.6%. For visualization-themed papers with GRS, the United States also has the largest proportion with 25.2%, followed by France with 19.7%, Germany with 18.1%, and China with 11.5%. I also asked myself if this distribution would change drastically if we would only consider

6 DISCUSSION

So how are we doing, then? Visualization papers currently only represent 29.2% of all papers with a GRS, somewhere between a quarter and a third. And out of the 485 full papers at IEEE VIS conferences 2021–2024, only 21 (or 4.3%) have received a stamp as of today,¹⁹ and 48 out of the 242 TVCG journal presentations at IEEE VIS (or 19.8%) in the same time frame.²⁰ After all, there is a difference between, on the one hand, some code being available as open source for a paper (which we do see often in today's papers) that, with little or (usually much) more effort [3] can be made to run and, on the other hand, the actual certification of a sufficiently documented and runnable code—the latter being ensured by a GRSI certification. Research in psychology [7], for instance, found that “just” awarding badges to papers for the fact that data and/or code is available is not sufficient to guarantee an exact reproducibility of the reported results—which, in contrast, the certification by initiatives such as the GRSI does guarantee (at least for those visuals of the paper that are being checked). Also within visualization we thus need more certification of reproducibility because only this certification forces authors to check that there shared code actually runs out of the box and has a sufficient level of documentation. The 29.2%, the 19.8%, and especially the 4.3% are too low—we can and we should do better than that. Several calls for more reproducibility exist in our field as I have reviewed in Section 3 (e.g., [10, 15, 21])—yet why are the numbers of papers at our main venues that have received a GRS still that low then? Speculating about possible reasons for this situation, a few reasons could explain the low numbers.

First, there may be an **issue with (too) few people knowing about the GRSI**. In this case we would need more publicity for this service. Yes, the EICs of the affected journals repeatedly mention the initiative in their messages to the reader (e.g., [9, 19]), but maybe these messages are not as well read by the community? Also, while the GRSI is mentioned in the acceptance letter at least of some journals (such as TVCG) and conferences (such as IEEE VIS), maybe we could more clearly announce this initiative and its benefits for our community in the opening sessions of our conferences, so

```
\section{Discussion}
```

So how are we doing, then? Visualization papers currently only represent $\text{\GrsiPercentageVisPapers}\%$ of all papers with a GRS, somewhere between a quarter and a third. And out of the $\text{\pgfmthparse{\TotalIeeeVisPapersInMMXXIV+\TotalIeeeVisPapersInMMXXIII+\TotalIeeeVisPapersInMMXXII+\TotalIeeeVisPapersInMMXXI}\pgfmthprintnumber[fixed, precision=0]\pgfmthresult}$ full papers at IEEE VIS conferences 2021--2024, only $\text{\pgfmthparse{\GrsiIeeeVisPapersInMMXXIV+\GrsiIeeeVisPapersInMMXXIII+\GrsiIeeeVisPapersInMMXXII+\GrsiIeeeVisPapersInMMXXI}\pgfmthprintnumber[fixed, precision=0]\pgfmthresult}$ (or $\text{\percentageRounded{\GrsiIeeeVisPapersInMMXXIV+\GrsiIeeeVisPapersInMMXXIII+\GrsiIeeeVisPapersInMMXXII+\GrsiIeeeVisPapersInMMXXI}\pgfmthprintnumber[fixed, precision=0]\pgfmthresult}$) have received a stamp as of today,¹⁹ and $\text{\pgfmthparse{\GrsiIeeeVisPapersInMMXXIV+\GrsiIeeeVisPapersInMMXXIII+\GrsiIeeeVisPapersInMMXXII+\GrsiIeeeVisPapersInMMXXI}\pgfmthprintnumber[fixed, precision=0]\pgfmthresult}$ out of the $\text{\pgfmthparse{\TotalIeeeVisPapersInMMXXIV+\TotalIeeeVisPapersInMMXXIII+\TotalIeeeVisPapersInMMXXII+\TotalIeeeVisPapersInMMXXI}\pgfmthprintnumber[fixed, precision=0]\pgfmthresult}$ TVCG journal presentations at IEEE VIS (or $\text{\percentageRounded{\GrsiIeeeVisTVCGJournalPapersInMMXXIV+\GrsiIeeeVisTVCGJournalPapersInMMXXIII+\GrsiIeeeVisTVCGJournalPapersInMMXXII+\GrsiIeeeVisTVCGJournalPapersInMMXXI}\pgfmthprintnumber[fixed, precision=0]\pgfmthresult}$) in the same time frame.

Script everything! ➡ “Scripting” evaluations

- several aspects of research are not computer code: evaluations
- making evaluations reproducible: clear instructions
 - ➡ pre-registrations of planned procedures
 - research questions, hypotheses, general study design/approach
 - your apparatus (your software & hardware setup)
 - planned tasks, counter-balancing, ...
 - participant recruitment, constraints, payment, ...
 - data analysis method/approach
 - anything else
- you can deviate from your pre-registrations (document it!)
- more on “scripting” studies in Part 3 ...



Share your code/materials anonymously

- beneficial for paper reviewers
- need trusted services that guarantee anonymity
- throw-away GitLab/GitHub accounts with code copy
- anonymized repos: [anonymous.4open.science](#)

Anonymous GitHub

- anonymous OSF repos:

https://osf.io/abcde/overview?view_only=abcdefghijklmnpqrstuvwxyz123456



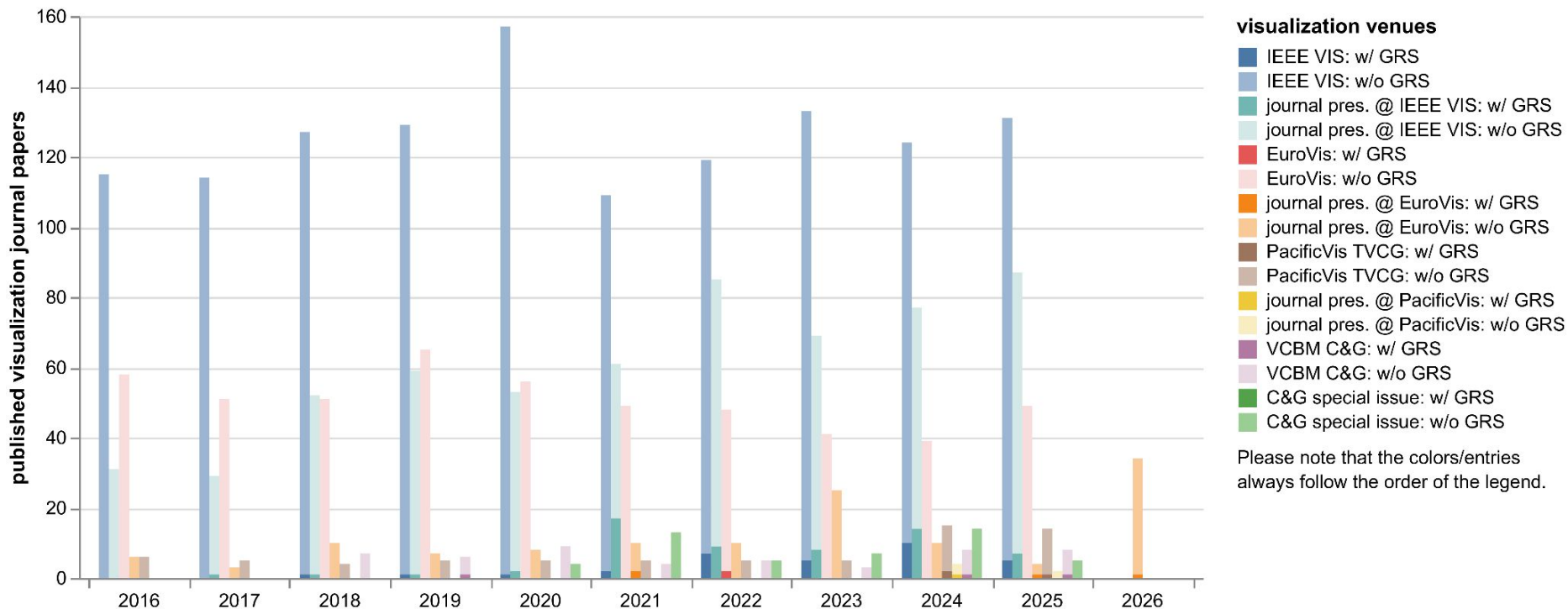
- additional material uploads (e.g., PCS)

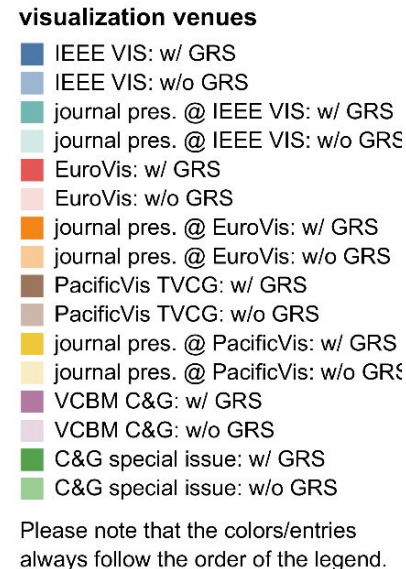
External verification: GRSI

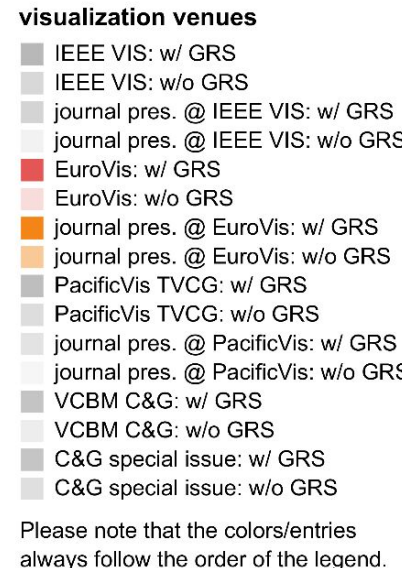


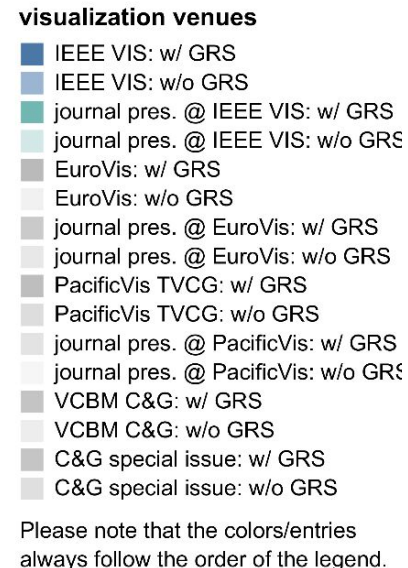
- Graphics Replicability Stamp Initiative: replicabilitystamp.org
- independent verification of **reproducibility**
- TOG; TVCG; CGF; Elsevier's C&G, CAGD, CAD; SIGGRAPH conference
- simple application on website, independent reviewer assigned
- reproduces **at least one visual** of your accepted/published paper
- non-anonymous! (reviewers may contact you for fixes)
- what does it do for you?
 - debugging of your repo & documentation
 - archiving of your code/repo
 - eternal fame and glory ;-)
 - more citations?

External verification: GRSI









External verification: GRSI

- get your paper accepted and published (DOI) ;-)
- put your code in a repo (GitHub, GitLab) [it should be there already!]
- add sufficient documentation
- describe what part of paper will be reproduced, and how
- test your repo code on a virgin machine (Windows, Linux, MacOS)
- fill in form to apply

Replicability Stamp Form

For more information visit <http://www.replicabilitystamp.org>.
For more information specific to the review process, labeling, and badging used by ACM TOG, visit <http://www.acm.org/publications/artifacts>.
For more information specific to the review process, labeling, and badging used by Computer Graphics Forum, visit <http://goo.gl/WnTaGM>.

IMPORTANT: Before completing the form, please make sure that you have followed the **GRSI requirements**. For short, clear instructions must be provided to build the code and replicate some paper results. If specific devices (e.g. GPU), datasets or libraries are required (eg. pytorch/tensorflow/CUDA version for learning-based papers), all details must be specified.
For ML projects, the training procedure (code, datasets) must be documented, and pretrained network weights must be provided for inference.

ABOUT THE CODE

Code URL

Please provide a URL of a git repository that contains the code to reproduce the results in your paper. This URL will be published on the Replicability website if the stamp is approved.
Though there is no strict limit on the size, too large repositories will not be properly processed by this submission system. If your repository is larger than 800 Mb, please do your best to reduce its size. If this is not feasible, please contact the GRSI Technical Manager to arrange an alternative submission procedure.

☐ This is a resubmission of an already submitted URL.

Short Description of the software

External verification: GRSI



- reviewer gets info from form (repo, preprint, description)
- usually via (empty) virtual machine
- several needs
 - installation of prerequisites (versions? scripted!)
 - clear instructions on what to do to reproduce (readme section!)
 - clear descriptions which newly produced images should match which ones from the published paper
 - explain when things may deviate, and how (live/changing data, etc.)
 - video for interactive tools
- two different target audiences for your code repo:
 - future code users in general
 - GRSI reviewer (or authors who want to reproduce exactly what you did)

External verification: GRSI



🔗 How to update the VisPubData dataset

This code will allow you to create an update of the [VisPubData dataset](#). If you have only small fixes of the data to report you might be better off to leave a comment on the [Google spreadsheet with the data](#) and to e-mail petra.isenberg@inria.fr to make the change for you. If, however, you would like to, for example, add a new year to the dataset or create a local update for yourself, then read on.

🔗 Prerequisites

- A Python 3 installation, we recommend [Anaconda](#).
- Several additional Python packages:
 - [lxml](#)
 - [requests](#)
 - [crossrefapi](#)
 - [tqdm](#)
 - The file `requirements.txt` includes all of these requirements, install them all in one go with `pip3 install -r requirements.txt`.

Hack for keeping your own images reusable

- keep the copyright, make images CC-BY like this:

IMAGES/GRAPHS/PLOTS/TABLES/DATA LICENSE/COPYRIGHT

We as authors state that all of our figures, graphs, plots, and data tables in this article are and remain under our own personal copyright, with the permission to be used here. We also make them available under the [Creative Commons Attribution 4.0 International \(CC BY 4.0\)](#) license and share them at osf.io/hfu6e.

REFERENCES

➡ add something like this at the end of all your papers