Lecture 5

Brooke Anderson

2023-07-27

Sharing code and data

Sharing code and data

There are a few options for how you share the code and data for the paper, including:

- Supplemental files through the journal's website
- GitHub
- Scientific repository, like [examples, NIH]
- Personal / academic website
- R package plus script

Sharing code and data

There are advantages to solutions that let you share everything as a directory:

- Relative pathnames work in scripts (because directory structure is preserved)
- If you set up your directory as an R Project, it's designed to be self-contained (can interact with other things in the directory but doesn't rely on the computer's file directory outside of that)

Leveraging a sharing system before you publish

Some of these methods are platforms that you can use before you publish, and they can help you work on code-heavy projects within your team. GitHub is one example, but some agency-funded platforms now combine repositories with workspaces for coding.

For GitHub (and some of the agency-funded platforms), you can toggle between having your data and code be public or private. This allows you to work privately before you publish, but then switch the repository to public when you publish the paper.

This can be much easier than trying to figure out how to post everything after publication. It also provides some helpful tools as you work on the paper.

Considerations about data

Processed data versus raw data

A lot of health-related data can require extensive pre-processing to extract the information you need to answer a scientific question.

It is helpful to include the original, raw data and the code used to get to the processed data.

Extremely large data

- Repositories built for data curation might be better suited for storing data that methods with data size limits ([example])
- Some repositories might have APIs—data can be downloaded programatically (from the code script), so the code could be connected to the data in this way
- Some repositories allow options to toggle private versus public (similar to GitHub)

Sensitive or protected data

- Can include mock dataset (same format, different numbers) to use to test and demonstrate the function of the code
- Can use .gitignore to include the data in a git repo on a local computer (or a secure server you're working on) but not push it to remote versions of that repo, like one on GitHub
- When working on a server for security reasons, it can be helpful to have RStudio Server installed, as this provides a friendlier interface for working with git version control on the server (avoids having to do much from the terminal)

Saving intermediate data

It can be useful to save intermediate versions of the data, as the code moves from raw data to the final figures, tables, and other results. Intermediate data can include:

- Processed data (e.g., if raw data are from flow cytometry, the results after gating and counting the data)
- Data points that are plotted in figures (i.e., data at the point immediately before it is graphed)

Note that this is in addition to the raw data.

For these intermediate data sets, it's helpful if you save them in plain text formats (e.g., ".csv" or ".txt" file).

(Sandve et al.)

Considerations about Code

Navigating the code

Can a reader or new user:

- Find where the code is?
- Find what order to run the code?
- Use the filepaths that are included in the code? (Or are they absolute paths that will need to change on a new user's computer?)
- Figure out which code was used to create specific figures and tables in the paper?
- Figure out how to start from a raw dataset similar to the data used for the paper and get to the final results?

Code License

Are others allowed to reuse your code? What are the rules for how they reuse it?

License choices

Versions used in original analysis

R base code changes from one version to another.

The code within packages can change even more.

Seems to be a particular source of difficulty in reproducing for work that involves Bioconductor packages.

Versions used in original analysis

You can record the information on all versions of the R software you used in your analysis (base R and packages).

`session_info` from `devtools` (or `sessionInfo` from base R)

Include this as a line in an RMarkdown file (often at the end), and it will print out the information on versions.

Versions used in original analysis

Help user recreate the "environment" that code ran in originally

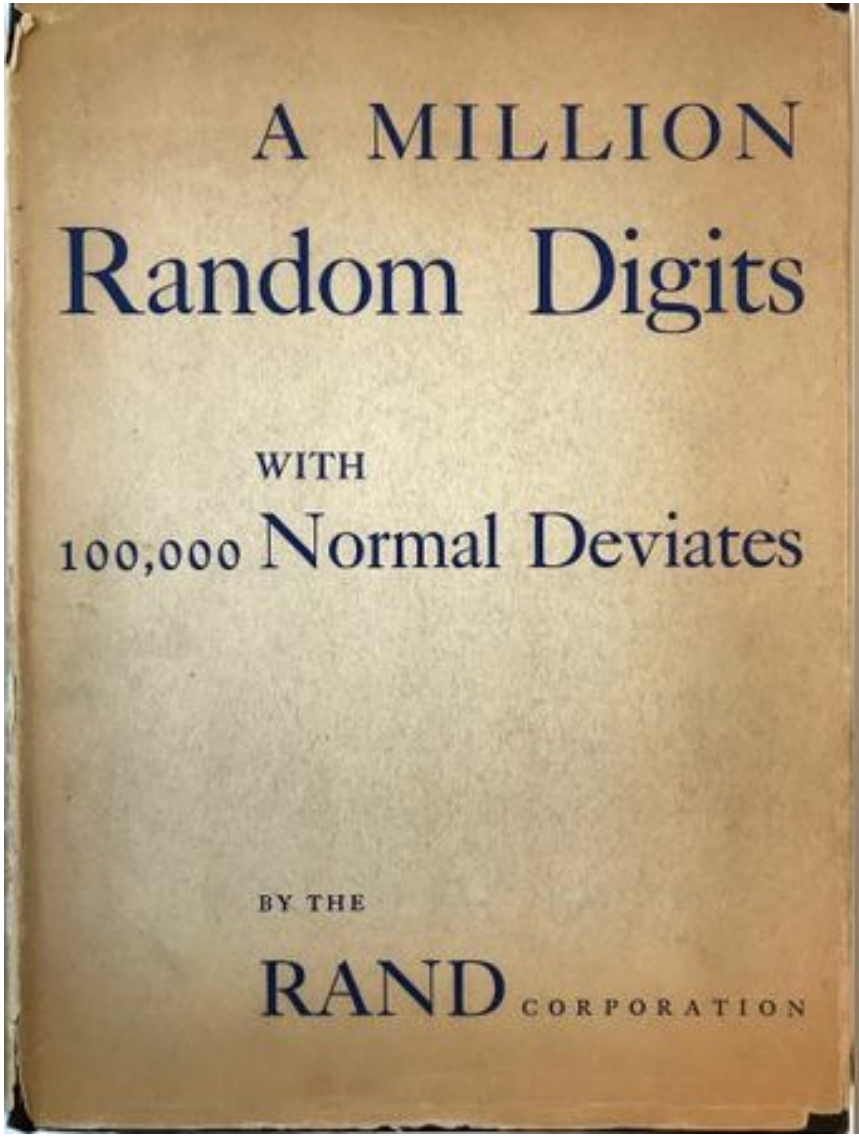
If you have version numbers, you could do it by hand (all old versions of CRAN packages, for example, are archived and avaiable for you to install, although it takes a bit more work that installing the current version of the package).

[Packrat alternatives]

Reproducing randomness

[Dice, roulette wheel]

Reproducing randomness



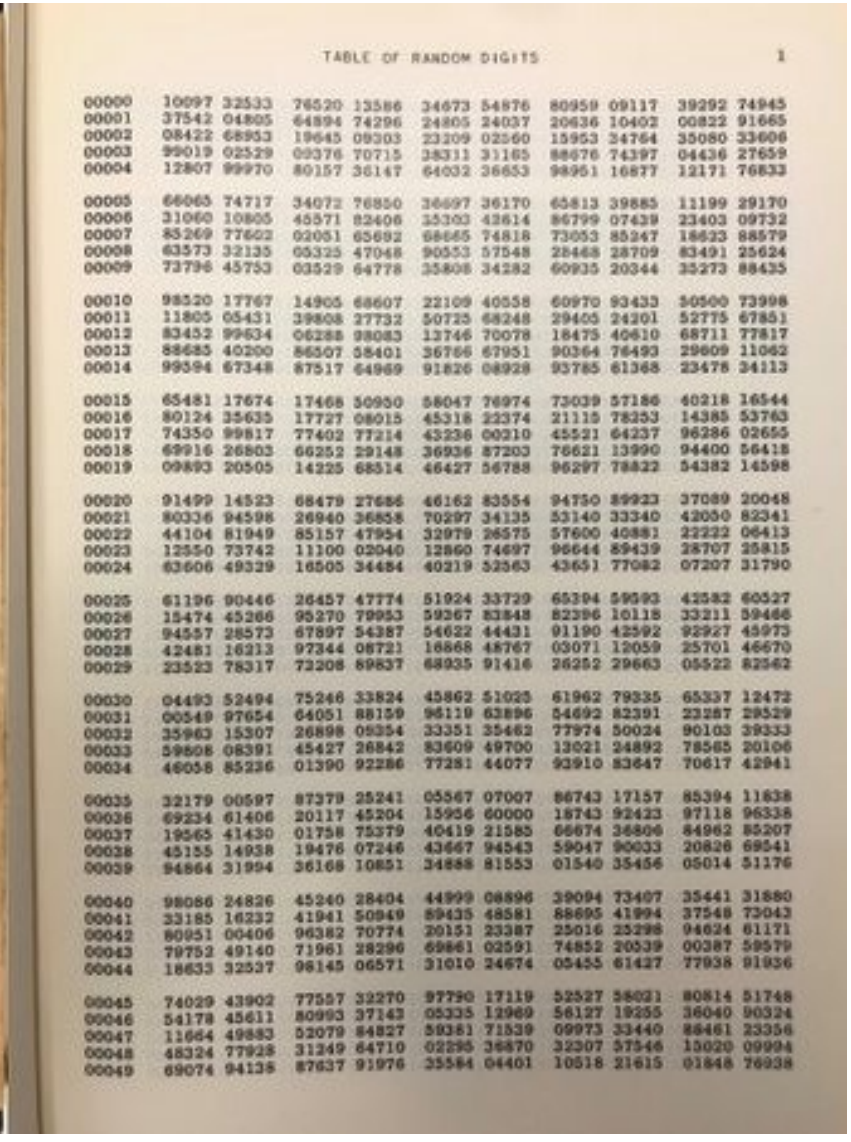


TABLE OF RANDOM DIGITS 1

00000	10097	32533	76520	13586	34673	54876	80959	09117	39292	74945
00001	37542	04805	64894	74296	24805	24037	20636	10402	00822	91665
00002	08422	68953	19645	09303	23209	02560	15953	34764	35080	33606
00003	99019	02529	09376	70715	38311	31165	88676	74397	04436	27659
00004	12807	99970	80157	36147	64032	36653	98951	16877	12171	76833
00005	66065	74717	34072	76850	36697	36170	65813	39885	11199	29170
00006	31060	10805	45571	82406	35303	42614	86799	07439	23403	09732
00007	85269	77602	02051	65692	68665	74818	73053	85247	18623	88579
00008	63573	32135	05325	47048	90553	57548	28468	28709	83491	25624
00009	73796	45753	03529	64778	35808	34282	60935	20344	35273	88435
00010	98520	17767	14905	68607	22109	40558	60970	93433	50500	73998
00011	11805	05431	39808	27732	50725	68248	29405	24201	52775	67851
00012	83452	99634	06288	98083	13746	70078	18475	40610	68711	77817
00013	88685	40200	86507	58401	36766	67951	90364	76493	29609	11062
00014	99594	67348	87517	64969	91826	08928	93785	61368	23478	34113
00015	65481	17674	17468	50950	58047	76974	73039	57186	40218	16544
00016	80124	35635	17727	08015	45318	22374	21115	78253	14385	53763
00017	74350	99817	77402	77214	43236	00210	45521	64237	96286	02655
00018	69916	26803	66252	29148	36936	87203	76621	13990	94400	56418
00019	09893	20505	14225	68514	46427	56788	96297	78822	54382	14598
00020	91499	14523	68479	27686	46162	83554	94750	89923	37089	20048
00021	80336	94598	26940	36858	70297	34135	53140	33340	42050	82341
00022	44104	81949	85157	47954	32979	26575	57600	40881	22222	06413
00023	12550	73742	11100	02040	12860	74697	96644	89439	28707	25815
00024	63606	49329	16505	34484	40219	52563	43651	77082	07207	31790
00025	61196	90446	26457	47774	51924	33729	65394	59593	42582	60527
00026	15474	45266	95270	79953	59267	83848	82396	10118	33211	59466
00027	94557	28573	67897	54387	54622	44431	91190	42592	92927	45973
00028	42481	16213	97344	08721	16868	48767	03071	12059	25701	46670
00029	23523	78317	73208	89837	68935	91416	26252	29663	05522	82562
00030	04493	52494	75246	33824	45862	51025	61962	79335	65337	12472
00031	00549	97654	64051	88159	96119	63896	54692	82391	23287	29529
00032	35963	15307	26898	09354	33351	35462	77974	50024	90103	39333
00033	59808	08391	45427	26842	83609	49700	13021	24892	78565	20106
00034	46058	85236	01390	92286	77281	44077	93910	83647	70617	42941
00035	32179	00597	87379	25241	05567	07007	86743	17157	85394	11838
00036	69234	61406	20117	45204	15956	60000	18743	92423	97118	96338
00037	19565	41430	01758	75379	40419	21585	66674	36806	84962	85207
00038	45155	14938	19476	07246	43667	94543	59047	90033	20826	69541
00039	94864	31994	36168	10851	34888	81553	01540	35456	05014	51176
00040	98086	24826	45240	28404	44999	08896	39094	73407	35441	31880
00041	33185	16232	41941	50949	89435	48581	88695	41994	37548	73043
00042	80951	00406	96382	70774	20151	23387	25016	25298	94624	61171
00043	79752	49140	71961	28296	69861	02591	74852	20539	00387	59579
00044	18633	32537	98145	06571	31010	24674	05455	61427	77938	91936
00045	74029	43902	77557	32270	97790	17119	52527	58021	80814	51748
00046	54178	45611	80993	37143	05335	12969	56127	19255	36040	90324
00047	11664	49883	52079	84827	59381	71539	09973	33440	88461	23356
00048	48324	77928	31249	64710	02295	36870	32307	57546	15020	09994
00049	69074	94138	87637	91976	35584	04401	10518	21615	01848	76938

Reproducing randomness

[Random lava lamps, Geiger counters]

Reproducing randomness

[xckd random number generator]

pseudorandom number generator

Reproducing randomness

You might be using random numbers if you:

- are sampling
- are using the Monte Carlo method / simulations
- are doing Bayesian statistics

Reproducing randomness

Setting seeds when code includes random number generation

Seed can be any integer

Pseudorandom number generator—depends on an initial value (the seed)

Reproducing randomness

If you don't set a seed, you will get different results when you run code that involves random number generation, because of the randomness involved.

```
sample(1:5)
```

```
## [1] 5 4 2 1 3
```

```
sample(1:5)
```

```
## [1] 3 4 5 1 2
```

Reproducing randomness

If you set the same seed each time before you run that code, you will get the same "random" results:

```
set.seed(100)
sample(1:5)
```

```
## [1] 2 3 5 4 1
```

```
set.seed(100)
sample(1:5)
```

```
## [1] 2 3 5 4 1
```

Reproducing randomness

```
set.seed(100)
sample(1:5)
```

```
## [1] 2 3 5 4 1
```

```
sample(1:5)
```

```
## [1] 1 2 4 3 5
```

Considerations when writing the journal article

RMarkdown for journal articles

rticles templates

How to collaborate when using RMarkdown

More advanced and complex approaches

Considering containers

[Docker. Galaxy? Workspaces on agency platforms?]

Considering containers

Containers can get very large—stores not just info on the versions of each piece of software used, but full source code (?)

Works somewhat like a black box—harder for users to explore, change, adapt

Can be useful, though, if using a large collection of different open-source software (command line tools, R, Python, etc., all in the same pipeline)

Considering making a package