How the R-spatial evolution project affects spatial econometrics workflows

[SEA Plenary Session] Spatial Econometrics Software Roundtable, Friday 1:30 pm-3:30 pm, Santa Fe

Roger Bivand

17 November 2023

Outline

Introduction

- Spatial econometrics workflows using the R statistical language may be impacted as the representation of spatial data undergoes modernization.
- The `sp` package and the classes for spatial data there defined will continue to be supported.
- Infrastructure packages `maptools`, `rgdal` and `rgeos` were archived on the Comprehensive R Archive Network (CRAN) 16 October 2023.
- Archiving means that while the source packages are retained in the archive for manual download and installation, they are no longer available for direct installation for users or for other packages.

Introduction

- The functionality provided by the retiring packages, such as reading and writing data and handling geometries may largely be replaced by `sf` and `terra`.
- The `spdep` package providing functions for creating weights matrices and carrying out tests for spatial autocorrelation has already been adapted, as have spatial econometrics packages.
- The R-spatial evolution project tries to facilitate the mitigation of these changes for workflows using retiring packages, including uses in scripts underlying published papers, books, etc.
- This presentation will cover steps that can be taken to reduce impacts on existing workflows, and indicate how new workflows may be constructed to use available data representations in `sp`, `sf` and `terra` going forward.

R-spatial evolution project

R-spatial evolution project

- It became clear from about ten years ago that the representation of spatial data objects could not simply stay with the initial `sp` object definitions made ten years before that.
- The `sp` spatial vector objects were again based on simpler structures for ESRI Shapefiles in the `maptools` package.
- ESRI Shapefiles do not satisfy the OGC Simple Features standard, do not support modern representations of coordinate reference systems, nor do they properly support multi-byte string encodings.
- `sp` objects have twice been tweaked, once to smuggle some Simple Features structures into polygon representations (exterior/interior rings), later to add a concealed WKT2-2019 coordinate reference system representation.

R-spatial evolution project

- Modernizing and replacing infrastructure packages in R-spatial workflows affects routines using `sp` (Pebesma and Bivand 2023a), `sf` (Pebesma 2023) and `raster` (Hijmans 2023a) and `terra` (Hijmans 2023b) taken together.
- `sf` (since 2017, Pebesma (2018)) and `terra` (since 2020) have interfaced OSGeo libraries PROJ, GDAL and GEOS directly, using the `Rcpp` (Eddelbuettel et al. 2023) framework.
- This means that they no longer need to use `rgdal` (Bivand, Keitt, and Rowlingson 2023), `maptools` (Bivand and Lewin-Koh 2023) or `rgeos` (Bivand and Rundel 2023) as `sp` and `raster` used to do.
- Using `Rcpp` is more efficient and easier to maintain. Rather than leaving `rgdal` and `rgeos` to decay, they were archived on CRAN 16 October 2023, as first announced in Edzer Pebesma's useR! plenary in July 2021.

R-spatial evolution project i

- The `sf` package was based on the Simple Features standard, and consequently can interface directly with GDAL and GEOS, which use Simple Features internally.
- Of course, GEOS only works with planar geometries, so `sf` uses `s2` when coordinates are spherical/ellipsoidal for topological operations and predicates.
- `sf` does not provide class definitions for spatial rasters; `sf`-based `stars` (spatiotemporal arrays) does this, as does `terra`.
- `sf` uses unit metric definitions for measurement (Pebesma, Mailund, and Hiebert 2016; Pebesma et al. 2023).

R-spatial evolution project ii

- `sf` supports `"agr"` attributes (Stasch et al. 2014), related to field domains as used by the GeoPackage format http://www.geopackage.org/guidance/extensions/schema.html, see also https://r-spatial.org/book/05-Attributes.html#sec-extensiveintensive.
- The next step is to work with spatial and spatiotemporal data in the cloud without downloading input data ... so plenty to do going forward, and maintaing outdated packages could not be a priority

Foundations for archiving retiring packages

- `sf` was well-established on CRAN in 2018, `s2` and `terra` in 2020, `sf` used WKT2-2019 from 2020, followed by `terra` in 2021, and 2022 for updating the GDAL RRASTER driver for WKT2-2019.
- Edzer Pebesma announced that `maptools`, `rgdal` and `rgeos` will be retiring in his online useR! July 2021 plenary 23:42 - 31:20.
- Edzer and I were funded by R Consortium in late 2021 to execute a project to archive the retiring packages minimizing distruption to CRAN packages depending on the retiring packages

Dependencies on retiring packages

- Dependencies between packages can be represented as a directed graph, so dependencies on retiring packages are edges from other packages to the retiring packages.
- Strong dependencies are those packages that must be installed for the packages depending on them to function.
- All dependencies add weak to strong dependencies; weak dependencies are not required to be present for the package to perform its core functions, but the package should not fail checks when the packages it weakly depends on are absent.

Process timeline

- From over 550 dependencies in early 2022, we eroded the count down to under 180 on archiving day 2023-10-16.
- Many dowstream package maintainers were alerted in December 2022 that `raster` no longer used `rgdal` or `rgeos`, using `terra` in their place; of some 250 packages, only about 30 have not updated.
- Guides and blogs were published on https://r-spatial.github.io/evolution/
- The evolution project successfully asked/nudged maintainers of packages depending on retiring packages to update their packages:

```
##        Sep 2019 Aug 2020 Aug 2021 Jan 2022 Dec 2022 Aug 2023 Oct 2023 Nov 2023
## All         487      519      561      563      497      284      175       43
## Strong      301      314      327      324      265      149       80        0
```

Pageranks of R-spatial packages

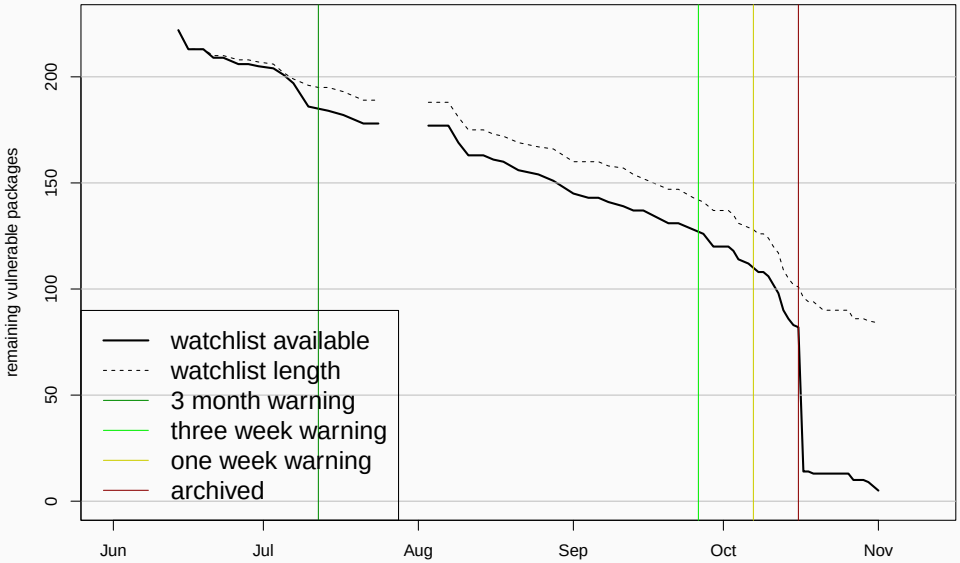
While dependency graph-based pagerank scores sum to unity over the graph of included packages (and these vary), changes over time show success in suppressing usage (all reverse dependencies) of retiring packages:

```
##          Sep 2019 Aug 2020 Aug 2021 Jan 2022 Dec 2022 Aug 2023 Oct 2023 Nov 2023
## sf       0.00078  0.00097  0.00109  0.00115  0.00133  0.00154  0.00168  0.00170
## sp       0.00237  0.00220  0.00190  0.00183  0.00159  0.00141  0.00128  0.00109
## terra    0.00000  0.00002  0.00008  0.00011  0.00029  0.00051  0.00063  0.00067
## raster   0.00149  0.00148  0.00142  0.00139  0.00126  0.00107  0.00097  0.00086
## rgdal    0.00109  0.00105  0.00099  0.00096  0.00079  0.00043  0.00026  0.00006
## rgeos    0.00076  0.00070  0.00061  0.00059  0.00046  0.00027  0.00014  0.00003
## maptools 0.00073  0.00062  0.00055  0.00052  0.00041  0.00022  0.00017  0.00004
```

Last five months of process

- After a further extensive round of emails and where possible github issues in April 2023, a watchlist of still vulnerable packages was started in June 2023 and more nudges sent.
- From June 2023, `sp` began to use `sf` in place of `rgdal`; from October 2023, `sp` dropped use of retiring packages.
- If a package on the watchlist is updated on CRAN, it is checked without retiring packages on the library path; if it passes checks, it is removed from the checklist; if no longer available on CRAN it is retained.
- Further warnings were sent to still-vulnerable package maintainers three months, three weeks and one week before archiving.

Last five months of process



Using ASSE (Kopczewska 2021) as an example

Using ASSE (Kopczewska 2021) as an example

- Although ASSE (Kopczewska 2021) was largely written five years ago, `sf` is only used in chapters 3 and 10, `terra` is not used, `raster` is used in chapters 2-4 and 9-10, `maptools` is used in chapters 2-6 and 8-9, `rgeos` in chapters 2, 7 and 10, and `rgdal` in chapters 2-6 and 8-10 (thankfully, the code is available); this contrasts with Lovelace, Nowosad, and Muenchow (2019), which does not use retiring packages.
- Packages `spdep` (chapters 2 and 4-5) and `spatialreg` (chapter 5) were a single package until early 2019, so book code post-dates that split; `sf` was added as a strong dependency of `spdep` and a weak dependency of `spatialreg` in early 2019.
- `sphet` is not used, and `pspatreg`, `spsur`, `spldv`, `hspm` post-date the publication of the book.
- We can use a project blog to indicate how the code in ASSE chapters 2, 4-5 might be updated.

ASDAR second edition code

- On the evolution project website the code of ASDAR (Bivand, Pebesma, and Gomez-Rubio 2013) is shown as by-chapter diffs between the original code, and code that works without the retiring packages.
- The updated code is available from https://github.com/rsbivand/sf_asdar2ed.
- Because the ASDAR first and second edition code is executed nightly, problems emerging from changes in packages used are detectable quite quickly.
- For example, a transient problem in `stars` (spatio-temporal arrays) turned up this week in line 118 of sf_asdar2ed/05_cm2/cm2_mod.R, and was addressed by updating `stars` to the development version (and raising a github issue).

ASSE code

- The ASSE code is not run regularly, and was provided as-is, with no aim to be runnable.
- So first a number of changes had to be made to permit it to run to completion, including the commenting out of `SDMTools` that is archived on CRAN and was not on my desktop machine.
- Other changes included adding those data files that could be located (thanks to Krzysztof Dyba https://github.com/kkopczewska/spatial_book/issues/5), and corrections of obvious typos.
- Calls to interactive functions were commented out, and code added to record warnings and dump output from running the scripts to file.

Updating ASSE code to current R

- From R 4.0.0 (April 2020), `stringsAsFactors = FALSE`, but the ASSE code pre-dated that change; adding `stringsAsFactors = TRUE` to all file-reading functions was necessary.
- R on macOS and Linux (`unix` systems) have handled multibyte `UTF-8` characters natively for a long time.
- `UTF-8` was also supported on Windows systems if declared as the preferred encoding, but until R 4.2.0 this was not general; R 4.2.0 for Windows and later are based on Microsoft's universal C runtime (UCRT), rather than many differing CRT versions, and provides system-wide `UTF-8` support.
- Data provided for ASSE is encoded as `CP1250`, so needed conversion where string comparisons or graphics output encountered non-ASCII characters.

Updating ASSE code to absence of retiring packages

- Minimal updating means keeping to `sp` object representations as far as possible; proper updating would move to `sf` objects.
- Using https://r-spatial.org/r/2023/04/10/evolution3.html#splitting-r_libs, the scripts running successfully with retiring packages on the library path were next re-run without retiring packages on the library path to detect failures.
- Because `GISTools` both was itself archived last year for unconnected maintenence oversights and it strongly depends on retiring packages, it had to be dropped; possibly if others use it, a new maintainer could adopt and update it.
- My fork is at https://github.com/rsbivand/kk_spatial_book, with diffs of the scripts and their output at https://rsbivand.github.io/kk_spatial_book/

Coordinate reference system definitions

- `rgeos` always assumed that geometries were planar (as does `GEOS`), but `sf` uses `s2` for spherical coordinates; for whatever reason, ASSE code transforms from `"ETRS89 / Poland CS92"` to `"NAD83"`, which I corrected to `"OGC:CRS84"` as used by geojson; NAD83 is strictly only for North America, but with spherical coordinates, `s2` will be in play for most topological operations and predicates.
- The provided WKT1 CRS definition in the ESRI Shapefile formatted data is out of date (Bivand 2021), but also note that the more recent definition is in northing-easting axis order; respecting data source axis order is becoming much more important, as are correct datum definitions and their representation.

Coordinate reference system: OGC:CRS84 i

```
sf::st_crs("OGC:CRS84")
```

```
## Coordinate Reference System:
##   User input: OGC:CRS84
##   wkt:
## GEOGCRS["WGS 84 (CRS84)",
##     DATUM["World Geodetic System 1984",
##         ELLIPSOID["WGS 84",6378137,298.257223563,
##             LENGTHUNIT["metre",1]]],
##     PRIMEM["Greenwich",0,
##         ANGLEUNIT["degree",0.0174532925199433]],
##     CS[ellipsoidal,2],
##         AXIS["geodetic longitude (Lon)",east,
##             ORDER[1],
##             ANGLEUNIT["degree",0.0174532925199433]],
##         AXIS["geodetic latitude (Lat)",north,
##             ORDER[2],
##             ANGLEUNIT["degree",0.0174532925199433]],
##     USAGE[
##         SCOPE["unknown"],
##         AREA["World"],
##         BBOX[-90,-180,90,180]],
##     ID["OGC","CRS84"]]
```

Coordinate reference system: legacy EPSG:2180 i

```
sf::st_crs(sf::st_read("../../kk_spatial_book/Panstwo.shp", quiet=TRUE))
```

```
## Coordinate Reference System:
##   User input: ETRS89 / Poland CS92
##   wkt:
## BOUNDCRS[
##     SOURCECRS[
##         PROJCRS["ETRS89 / Poland CS92",
##             BASEGEOGCRS["ETRS89",
##                 DATUM["European Terrestrial Reference System 1989",
##                     ELLIPSOID["GRS 1980",6378137,298.257222101,
##                         LENGTHUNIT["metre",1]]],
##                 PRIMEM["Greenwich",0,
##                     ANGLEUNIT["Decimal Degree",0.0174532925199433]]],
##             CONVERSION["unnamed",
##                 METHOD["Transverse Mercator",
##                     ID["EPSG",9807]],
##                 PARAMETER["Latitude of natural origin",0,
##                     ANGLEUNIT["Decimal Degree",0.0174532925199433],
##                     ID["EPSG",8801]],
##                 PARAMETER["Longitude of natural origin",19,
##                     ANGLEUNIT["Decimal Degree",0.0174532925199433],
##                     ID["EPSG",8802]],
```

Coordinate reference system: legacy EPSG:2180 ii

```
##                 PARAMETER["Scale factor at natural origin",0.9993,
##                     SCALEUNIT["unity",1],
##                     ID["EPSG",8805]],
##                 PARAMETER["False easting",500000,
##                     LENGTHUNIT["Meter",1],
##                     ID["EPSG",8806]],
##                 PARAMETER["False northing",-5300000,
##                     LENGTHUNIT["Meter",1],
##                     ID["EPSG",8807]]],
##             CS[Cartesian,2],
##                 AXIS["(E)",east,
##                     ORDER[1],
##                     LENGTHUNIT["Meter",1]],
##                 AXIS["(N)",north,
##                     ORDER[2],
##                     LENGTHUNIT["Meter",1]],
##             ID["EPSG",2180]]],
##     TARGETCRS[
##         GEOGCRS["WGS 84",
##             DATUM["World Geodetic System 1984",
##                 ELLIPSOID["WGS 84",6378137,298.257223563,
##                     LENGTHUNIT["metre",1]]],
##             PRIMEM["Greenwich",0,
```

Coordinate reference system: legacy EPSG:2180 iii

```
##                 ANGLEUNIT["degree",0.0174532925199433]],
##             CS[ellipsoidal,2],
##                 AXIS["latitude",north,
##                     ORDER[1],
##                     ANGLEUNIT["degree",0.0174532925199433]],
##                 AXIS["longitude",east,
##                     ORDER[2],
##                     ANGLEUNIT["degree",0.0174532925199433]],
##             ID["EPSG",4326]]],
##     ABRIDGEDTRANSFORMATION["Transformation from ETRS89 to WGS84",
##         METHOD["Position Vector transformation (geog2D domain)",
##             ID["EPSG",9606]],
##         PARAMETER["X-axis translation",0,
##             ID["EPSG",8605]],
##         PARAMETER["Y-axis translation",0,
##             ID["EPSG",8606]],
##         PARAMETER["Z-axis translation",0,
##             ID["EPSG",8607]],
##         PARAMETER["X-axis rotation",0,
##             ID["EPSG",8608]],
##         PARAMETER["Y-axis rotation",0,
##             ID["EPSG",8609]],
##         PARAMETER["Z-axis rotation",0,
```

Coordinate reference system: legacy EPSG:2180 iv

```
##             ID["EPSG",8610]],
##         PARAMETER["Scale difference",1,
##             ID["EPSG",8611]]]]
```

Coordinate reference system: current EPSG:2180 i

```
sf::st_crs("EPSG:2180")
```

```
## Coordinate Reference System:
##   User input: EPSG:2180
##   wkt:
## PROJCRS["ETRF2000-PL / CS92",
##     BASEGEOGCRS["ETRF2000-PL",
##         DATUM["ETRF2000 Poland",
##             ELLIPSOID["GRS 1980",6378137,298.257222101,
##                 LENGTHUNIT["metre",1]]],
##         PRIMEM["Greenwich",0,
##             ANGLEUNIT["degree",0.0174532925199433]],
##         ID["EPSG",9702]],
##     CONVERSION["Poland CS92",
##         METHOD["Transverse Mercator",
##             ID["EPSG",9807]],
##         PARAMETER["Latitude of natural origin",0,
##             ANGLEUNIT["degree",0.0174532925199433],
##             ID["EPSG",8801]],
##         PARAMETER["Longitude of natural origin",19,
##             ANGLEUNIT["degree",0.0174532925199433],
##             ID["EPSG",8802]],
##         PARAMETER["Scale factor at natural origin",0.9993,
```

Coordinate reference system: current EPSG:2180 ii

```
##             SCALEUNIT["unity",1],
##             ID["EPSG",8805]],
##         PARAMETER["False easting",500000,
##             LENGTHUNIT["metre",1],
##             ID["EPSG",8806]],
##         PARAMETER["False northing",-5300000,
##             LENGTHUNIT["metre",1],
##             ID["EPSG",8807]]],
##     CS[Cartesian,2],
##         AXIS["northing (x)",north,
##             ORDER[1],
##             LENGTHUNIT["metre",1]],
##         AXIS["easting (y)",east,
##             ORDER[2],
##             LENGTHUNIT["metre",1]],
##     USAGE[
##         SCOPE["Topographic mapping (medium and small scale)."],
##         AREA["Poland - onshore and offshore."],
##         BBOX[49,14.14,55.93,24.15]],
##     ID["EPSG",2180]]
```

Main updating points i

- Obviously `rgdal::readOGR()` has to be replaced by `as(sf::st_read(), "Spatial")` for reading spatial vector files.
- ASSE scripts make extensive use of `sp::over()`, which in some input combinations used `rgeos`; in those cases, `sf` has to be used instead.
- `maptools::unionSpatialPolygons` using `rgeos` can sometimes be replaced by the `sf::aggregate` method for `sf` objects, sometimes by `sf::st_union`.
- `rgeos` binary predicates can be replaced by equivalent `sf` predicates, note that returned objects from `sf` by default are sparse not dense logical matrices.
- `maptools::pointLabel` is now `car::pointLabel`.

Main updating points ii

- `rgeos::gDelaunayTriangulation` can be replaced by `deldir::deldir`.
- `rgeos::gArea` can be replaced by `sf::st_area`, but the area units will be given correctly by `sf` rather than wrongly for spherical coordinates by `rgeos`.
- Coercion to classes defined in `spatstat` was in `maptools` but is now in `sf`; tesselations could be coerced by stepping through lists of tiles.
- Upgrading spatial vector files from `ESRI Shapefile` to `GPKG` (GeoPackage) would permit proper storage of `UTF-8` characters and `WKT2-2019` coordinate reference system declarations.
- The `jpt_powier` field of `powiaty.dbf` appears to be corrupted in the upstream data source.

Main updating points iii

- `sf::st_read` does not use the `FID` component of vector layers as a `row.name`.
- `sp` is no longer attached by `spdep`, so may need to be attached explicitly.

Further steps

- It should be obvious (and expected) that there were no impacts on `spdep` for creating spatial weights or carrying out tests for spatial autocorrelation, or on `spatialreg`.
- Indeed, packages for spatial econometrics generally have been pro-actively adapting to the archiving of the retiring packages.
- Chapters 14-17 of Pebesma and Bivand (2023b) were written without the retiring packages, and adaptations of code for creating spatial weights in `spdep` have tracked `s2` and `sf` closely.
- In some cases, `spdep` converts input `sp` objects to `sf` to benefit from its more efficient approaches.
- Consequently, converting ASSE from an `sp` to an `sf` workflow could be considered were a second edition anticipated.

Conclusions i

- The mitigations applied here use `sf`, but `terra` could have been employed as it also links to GDAL, GEOS and PROJ, and remains closer to the `sp` legacy GIS position of pretending that most coordinates are planar for topological predicates and operations.
- Data should no longer be shared in ESRI Shapefile format, but the GeoPackage format should be used; recent and impending changes in GDAL drivers will make GPKG even more efficient (column-based access more generally in vector drivers, SOzip).
- Increasingly, data are accessed through APIs (ASSE chapter 3), but both the data and the API definitions are seldom stable; computing in the cloud is now realistic for earth observation data.

Conclusions ii

- Workflow impacts on spatial econometrics from the archiving of retiring R-spatial packages are not difficult to mitigate, but older workflows, courses, tutorials, etc., would benefit from active maintenance.
- Both R itself and key programming infrastructure components are evolving, native pipes came in R 4.1.0.
- S7 is a forward-looking proof-of-concept class/object system hoping to combine S3 (as used by `sf`) and S4 (as used by `sp`), see https://cran.r-project.org/package=S7.
- Changes in class representations will also affect how we organise model output, like methods such as `impacts`, `summary`, `predict`, etc.

sessionInfo i

```
sessionInfo()

## R version 4.3.2 (2023-10-31)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Fedora Linux 38 (Workstation Edition)
##
## Matrix products: default
## BLAS:   /home/rsb/topics/R/R432-share/lib64/R/lib/libRblas.so
## LAPACK: /home/rsb/topics/R/R432-share/lib64/R/lib/libRlapack.so;  LAPACK version 3.11.0
##
## locale:
##  [1] LC_CTYPE=en_GB.UTF-8       LC_NUMERIC=C               LC_TIME=en_GB.UTF-8
##  [4] LC_COLLATE=en_GB.UTF-8     LC_MONETARY=en_GB.UTF-8    LC_MESSAGES=en_GB.UTF-8
##  [7] LC_PAPER=en_GB.UTF-8       LC_NAME=C                  LC_ADDRESS=C
## [10] LC_TELEPHONE=C             LC_MEASUREMENT=en_GB.UTF-8 LC_IDENTIFICATION=C
##
## time zone: Europe/Oslo
## tzcode source: system (glibc)
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## loaded via a namespace (and not attached):
```

sessionInfo ii

```
##  [1] vctrs_0.6.4        cli_3.6.1          knitr_1.45         rlang_1.1.1
##  [5] xfun_0.40          DBI_1.1.3          KernSmooth_2.23-22 generics_0.1.3
##  [9] sf_1.0-14          glue_1.6.2         htmltools_0.5.6.1  e1071_1.7-13
## [13] fansi_1.0.5        rmarkdown_2.25     grid_4.3.2         tibble_3.2.1
## [17] evaluate_0.22      classInt_0.4-10    fastmap_1.1.1      yaml_2.3.7
## [21] lifecycle_1.0.3    compiler_4.3.2     dplyr_1.1.3        pkgconfig_2.0.3
## [25] Rcpp_1.0.11        rstudioapi_0.15.0  digest_0.6.33      R6_2.5.1
## [29] tidyselect_1.2.0   utf8_1.2.4         class_7.3-22       pillar_1.9.0
## [33] magrittr_2.0.3     tools_4.3.2        proxy_0.4-27       units_0.8-4
```

Aftermatter

Bivand, Roger. 2021. “Progress in the R Ecosystem for Representing and Handling Spatial Data.” *Journal of Geographical Systems* 23: 515–46. https://doi.org/10.1007/s10109-020-00336-0.

Bivand, Roger, Tim Keitt, and Barry Rowlingson. 2023. *rgdal: Bindings for the ’Geospatial’ Data Abstraction Library*. https://cran.r-project.org/package=rgdal.

Bivand, Roger, and Nicholas Lewin-Koh. 2023. *Maptools: Tools for Handling Spatial Objects*. https://cran.r-project.org/package=maptools.

Bivand, Roger, Edzer Pebesma, and Virgilio Gomez-Rubio. 2013. *Applied Spatial Data Analysis with R, Second Edition*. Springer, NY. https://asdar-book.org/.

Bivand, Roger, and Colin Rundel. 2023. *Rgeos: Interface to Geometry Engine - Open Source (’GEOS’)*. https://cran.r-project.org/package=rgeos.

References ii

Eddelbuettel, Dirk, Romain Francois, JJ Allaire, Kevin Ushey, Qiang Kou, Nathan Russell, Inaki Ucar, Douglas Bates, and John Chambers. 2023. *Rcpp: Seamless R and C++ Integration*. https://CRAN.R-project.org/package=Rcpp.

Hijmans, Robert J. 2023a. *raster: Geographic Data Analysis and Modeling*. https://cran.r-project.org/package=raster.

———. 2023b. *terra: Spatial Data Analysis*. https://cran.r-project.org/package=terra.

Kopczewska, Katarzyna. 2021. *Applied Spatial Statistics and Econometrics, Data Analysis with R*. Abingdon, UK: Routledge.

Lovelace, Robin, Jakub Nowosad, and Jannes Muenchow. 2019. *Geocomputation with R*. Chapman & Hall/CRC. https://r.geocompx.org/.

Pebesma, Edzer. 2018. “Simple Features for R: Standardized Support for Spatial Vector Data.” *The R Journal* 10 (1): 439–46. https://doi.org/10.32614/RJ-2018-009.

References iii

———. 2023. *sf: Simple Features for R*. https://cran.r-project.org/package=sf.

Pebesma, Edzer, and Roger Bivand. 2023a. *Sp: Classes and Methods for Spatial Data*. https://CRAN.R-project.org/package=sp.

———. 2023b. *Spatial Data Science with Applications in R*. Chapman & Hall. https://r-spatial.org/book.

Pebesma, Edzer, Thomas Mailund, and James Hiebert. 2016. "The r Journal: Measurement Units in R." *The R Journal* 8: 486–94. https://doi.org/10.32614/RJ-2016-061.

Pebesma, Edzer, Thomas Mailund, Tomasz Kalinowski, and Iñaki Ucar. 2023. *Units: Measurement Units for r Vectors*. https://CRAN.R-project.org/package=units.

Stasch, Christoph, Simon Scheider, Edzer Pebesma, and Werner Kuhn. 2014. "Meaningful Spatial Prediction and Aggregation." *Environmental Modelling & Software* 51: 149–65. https://doi.org/10.1016/j.envsoft.2013.09.006.