Experiences from developing and maintaining the surveillance R-package

Michael Höhle*

*Department of Mathematics
Stockholm University, Sweden

Making R packages (and) Shiny
Stockholm R useR group meetup, 23 Apr 2013

Outline

1 Introduction

2 The R package surveillance
- Surveillance time series examples
- Two component modelling of epidemic phenomena
- Overview of package functionality

3 Experiences
- Things to think about while making a package
- Experiences from getting the package into circulation

4 Summary

Introduction – just a package among many

Currently, the Comprehensive R Archive Network (CRAN) package repository contains 4457\(^1\) packages

This is the “story” of one package seen through the eyes of its package maintainer

> install.packages("surveillance")
> library("surveillance")

Aim of this talk: Shortly present the package and then discuss experiences of creating and maintaining a package on CRAN.

---

\(^{1}\) As of 23-Apr-2013.
What is surveillance? (1)

surveillance is an open source R package for the visualization, modeling and monitoring of routinely collected public health surveillance data.

- Motivation: Provide data structure and implementational framework for methodological developments in outbreak detection.
- Spin-off: Tool for epidemiologists and others working in applied disease monitoring.
- Availability: CRAN, current development version from http://surveillance.r-forge.r-project.org/

The R package surveillance

- Surveillance time series examples
- Two component modelling of epidemic phenomena
- Overview of package functionality

Experiences

Summary

Outline

1 Introduction

2 The R package surveillance

- Surveillance time series examples
- Two component modelling of epidemic phenomena
- Overview of package functionality

3 Experiences

4 Summary

Surveillance data as multivariate time series of counts (1)

- Data from surveillance systems is, after suitable preprocessing, available as multivariate time series of counts $\{y_{it}; i = 1, \ldots, m, t = 1, \ldots, n\}$.
- The surveillance class for such data is the sts class.

```r
> data("measlesDE")
> measlesDE
-- An object of class sts --
freq: 52
start: 2005 1
dim(observed): 156 16
```

Head of observed:

Baden-Wuerttemberg Bavaria Berlin Brandenburg Bremen Hamburg Hesse

```
[1,] 0 0 0 0 0 1 3
```

Mecklenburg-Western Pomerania Lower Saxony North Rhine-Westphalia

```
[1,] 0 0 0
```

Rhineland-Palatinate Saarland Saxony Saxony-Anhalt Schleswig-Holstein

```
[1,] 0 0 1 0 0
```

Thuringia

```
[1,] 0
```

Surveillance data as multivariate time series of counts (2)

```r
> plot(measlesDE, type = observed ~ time | unit)
```
Monitoring of univariate surveillance time series (1)

Outbreak detection in univariate time series while adjusting for reporting delays – shown by the example of listeriosis cases in Germany 2001-2013

Monitoring of univariate surveillance time series (2)

Manitz and H. (2013) develop boda to adjust detection for simultaneous covariate processes:

Epidemic Curve of the O104:H4 outbreak in Germany

Werber et al. (2013) analyse the O104:H4 incubation period using an Weibull interval censored regression model in 114 symptomatic adults and use this for back-projecting the epidemic curve of diarrhea onsets.

Use of surveillance by others

A number of public health institutions and projects use the package, especially for outbreak detection:

- Computer Assisted Search For Epidemics (CASE) project by the Swedish Institute for Infectious Disease Control (SMI) – Cakici et al. (2010)
- Project on understanding Disease Risks from Livestock Movement in the Greater Mekong Subregion (Anonymous, 2011)
- Governmental Institute of Public Health, Lower Saxony, Germany, Finish National Institute for Health and Welfare, French National Reference Centre for Salmonella, Austrian Agency for Health and Food Safety
Example: Modelling two specific finetypes of invasive meningococcal disease (IMD) as space-time point processes using `twinstim`

```r
> with(imdepi, { plot(W) ; plot(events,add=TRUE)})
```

Spatio-temporal visualization of disease occurrence using the `animation` package Xie (2010). Produces animated GIF files or Flash animations:

```r
> animate(imdepi)
```

What is `surveillance`? (2)

- Prospective monitoring for univariate count data time series:
  - farrington – Farrington et al. (1996)
  - improvedFarrington – Noufaily et al. (2013)
  - cusum – Rossi et al. (1999) and extensions

- Prospective changepoint detection for categorical time series:
  - pairedbinCUSUM – surgical performance (Steiner et al., 2000)
What is surveillance? (3)

- Retrospective count data time series models:
  - hhh – Held et al. (2005); Paul et al. (2008)
  - hhh4 – Paul and Held (2011)
  - twine – Held et al. (2006)

- Spatio-Temporal point process modelling and monitoring:
  - twinSIR – discrete space - continuous time modelling (H., 2010)
  - twine – continuous space - continuous time modelling (Meyer et al., 2012)
  - stcd – continuous space - continuous time cluster detection (Assunção and Correa, 2009)

- Interpreting the epidemiological curve of an outbreak:
  - backprojNP – Non-parametric back-projection (Becker et al., 1991)
  - nowcast – Now-casting to adjust for reporting delays during an outbreak (H. and an der Heiden, 2013)

Outline

1. Introduction
2. The R package surveillance
3. Experiences
   - Things to think about while making a package
   - Experiences from getting the package into circulation
4. Summary

Things to think about before making a package (1)

- Why write a package at all?
  - Structuring R code as a package is a useful part of the documentation and code re-factoring process.
  - A package is a standardized way of collecting of code, data and documentation into a bundle.
  - A package is easy to distribute and easy for others to install.

- Why share your code?
  - Others can use it & improve it → collaborative software development
  - It's a way to get your new statistical methodology applied in practice (...and might even boost your citation count)
  - You get to talk at UserR meetings or give tutorials...
  - Allows others to reproduce the results of your analyses → reproducible research

Getting your package into circulation

- Distribution of a package through the Comprehensive R Archive Network (CRAN) repository is subject to the repository policy.
- This includes legal requirements as well as technical aspects. Probably the most important practical requirement is:
  In principle, packages must pass R CMD check without warnings or significant notes to be admitted to the main CRAN package area.
- At submission the CRAN team verifies that policies are adhered to.
- As package maintainer one is amazed by their efforts and patience to maintain CRAN. They deserve a large credit for what R is today!
Experiences from getting the package into circulation

Things to think about before making a package (2)

- Sharing code is great, but are you up for the challenge of maintaining a package on CRAN?
- Find the right license to distribute your free software
  - GNU General Public License is the license of choice
  - ...but it might be more complicated than you think
- Think carefully about which packages you want to depend on
  - if they change, your package might have to change
  - your license model may depend on it

Package Design

- What's going in the package (one package fits them all?)
- Just documenting classes, methods and functions is not enough. How are you going to document
  - the data structure?
  - package applicability for an entire analysis?
- Vignettes written with Sweave/knitr are a good way to bring more context into your documentation.

Organization of Package Maintenance

- How to organize the files to accommodate access & revisions for all users & developers
  - R-forge (svn, overnight package building, issue trackers, etc.)
  - github (collaboration on github might be superior, easier to branch)
- How to make others aware that your software is available?
  - Post in R forums, CRAN Task Views,
  - Present at R User meetings,
  - Write an article, e.g., for RNews or JSS.
  - Solve a real world problem...
- How to deal with user and developer feedback?

Reflecting the experiences...

Q: What questions were asked before making surveillance?
A: Few of the above. I just wanted to try it.

Q: How was it decided, which functionality goes into the package?
A: Little or no structure. Toolbox idea: Code was created as part of methodological developments or when reading someone else's paper

Q: Writing a package is just software design, or not?
A: Not quite. Explaining statistical models by R model formula syntax is an alternative way to abstract than using equations
Outline

1 Introduction
2 The R package surveillance
3 Experiences
4 Summary

Outlook (incl. reality check)

- Current works:
  - Integrate use of outbreak detection algorithms into the epidemiologist’s workflow at the RKI including automatic report generation
  - Improve documentation for the modelling of epidemic phenomena
- A quote worth remembering:
  *It is frightful that someone who is no one ... can set any error into circulation with no thought of responsibility and with the aid of this dreadful disproportioned means of communication*²

Take home message:

Have fun writing your own package and making your own experiences!

> q()


Acknowledgements

Persons:

- Sebastian Meyer, Michaela Paul, Andrea Riebler and Leonhard Held, Institute of Social and Preventive Medicine, University of Zurich, Switzerland
- Maëlle Salmon and SurvStat@RKI, Robert Koch Institute, Berlin
- T. Correa, M. Hofmann, C. Lang, J. Manitz, A. Riebler, D. Sabanés Bové, Steiner, M. Virtanen, V. Wimmer, and The R Core Team

Financial Support:

- German Science Foundation (DFG, 2003-2006)
- Munich Center of Health Sciences (2007-2010)
- Swiss National Science Foundation (SNF, since 2007)
- Robert Koch Institute (RKI, since 2012)

Literature I

http://trade.animalhealthresearch.asia.


Literature II


Literature III

