



8. Introduction to R Packages

Ken Rice
Timothy Thornotn

University of Washington

Seattle, July 2013

In this session

Base R includes pre-installed packages that allow for a fully functioning statistical environment in which a variety of analyses can be conducted. Thousands of user-contributed extension packages are available that provide enhanced functionality with R.

- Finding and installing packages
- Loading packages
- Examples using available packages on CRAN

CRAN Packages

The programming environment of R has facilitated rapid development of packages by numerous authors.

R packages are collections of functions, data, and compiled code in a well-defined format.

Thousands of packages are available for download from the Comprehensive R Archive Network (CRAN):

<http://cran.r-project.org>

There are more than 4,500 packages available on CRAN!

Standard or *base* R comes with some standard packages installed for basic data management, analysis, and graphical tools.

CRAN Packages

Below are some of the CRAN packages that come with base R:

KernSmooth

Functions for kernel smoothing (and density estimation) corresponding to the book “Kernel Smoothing” by M. P. Wand and M. C. Jones, 1995.

MASS

Functions and datasets from the main package of Venables and Ripley, “Modern Applied Statistics with S”. (Contained in the `VR` bundle for R versions prior to 2.10.0.)

Matrix

A Matrix package. (Recommended for R 2.9.0 or later.)

boot

Functions and datasets for bootstrapping from the book “Bootstrap Methods and Their Applications” by A. C. Davison and D. V. Hinkley, 1997, Cambridge University Press.

class

Functions for classification (*k*-nearest neighbor and LVQ). (Contained in the `VR` bundle for R versions prior to 2.10.0.)

cluster

Functions for cluster analysis.

codetools

Code analysis tools. (Recommended for R 2.5.0 or later.)

foreign

Functions for reading and writing data stored by statistical software like Minitab, S, SAS, SPSS, Stata, Systat, etc.

lattice

Lattice graphics, an implementation of Trellis Graphics functions.

mgcv

Routines for GAMs and other generalized ridge regression problems with multiple smoothing parameter selection by GCV or UBRE.

nlme

Fit and compare Gaussian linear and nonlinear mixed-effects models.

nnet

Software for single hidden layer perceptrons (“feed-forward neural networks”), and for multinomial log-linear models. (Contained in the `VR` bundle for R versions prior to 2.10.0.)

rpart

Recursive PARTitioning and regression trees.

spatial

Functions for kriging and point pattern analysis from “Modern Applied Statistics with S” by W. Venables and B. Ripley. (Contained in the `VR` bundle for R versions prior to 2.10.0.)

survival

Functions for survival analysis, including penalized likelihood.

Installing packages

We often need additional functionality beyond the packages offered by the base core R library.

In order to install an extension package, we can invoke the `install.packages()` function which typically downloads the package from CRAN and installs it for use.

For example, the following command can be used to install the `hexbin` package:

```
install.packages("hexbin")
```

Installing packages

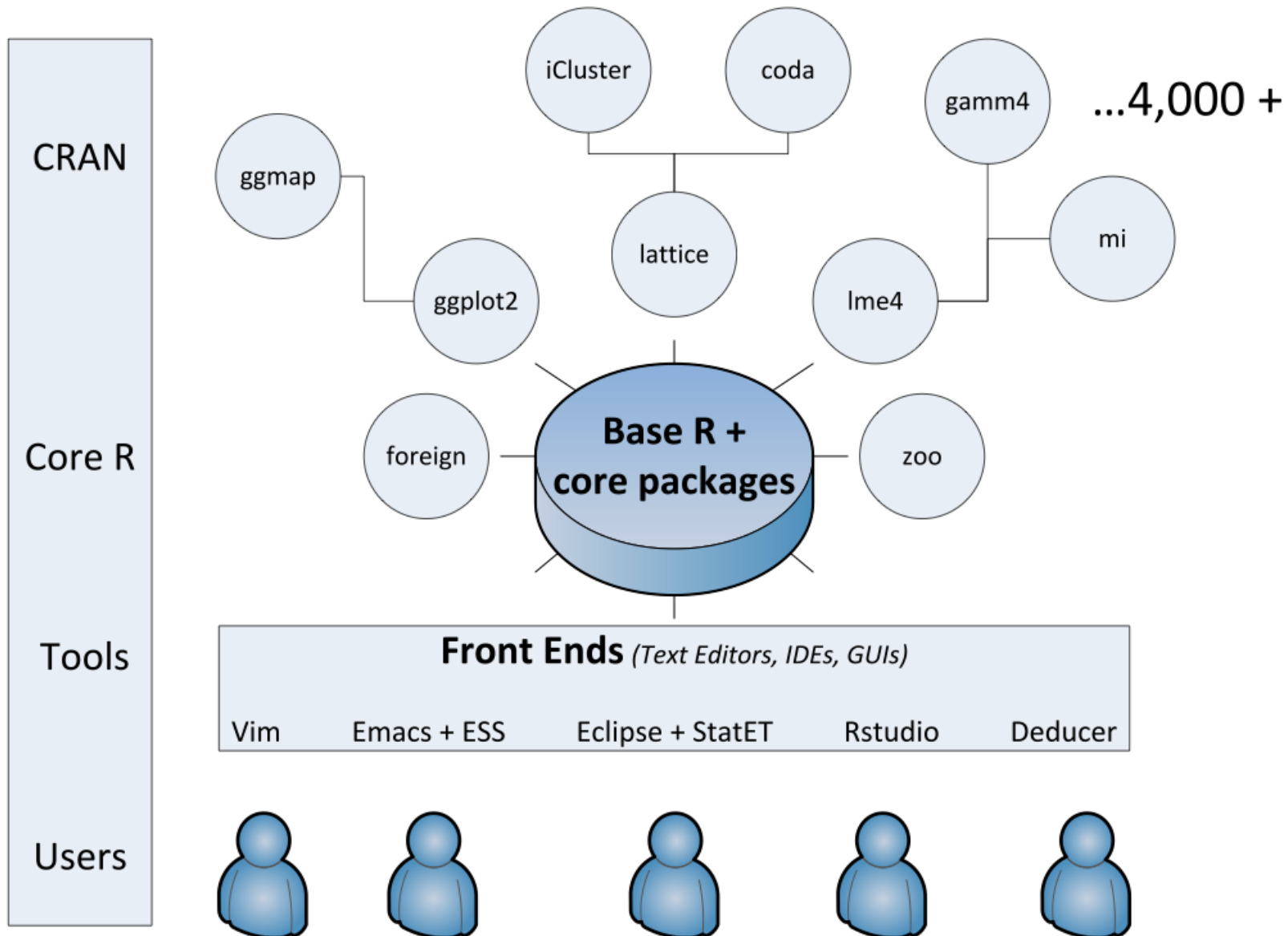
Alternatively, an R package can be installed by

- using the command `install.packages()` without any arguments
- selecting “Install package(s)” from the “Packages” menu (for Windows versions) or “Packages and Data” menu (for Apple versions) near the top of the R console.

This will bring up a list of packages that are available for installation.

You can then choose the package that you would like to install from that list.

Installing packages



Loading Packages

To use an installed package for an R session, the package must first be loaded.

The `library` function can be used to load a package. For example, to load the `foreign` package for an R session, use the following command:

```
library("foreign")
```

To see which packages are installed for your session, issue the command `library()` without any arguments.

Users connected to the Internet can use the `update.packages()` function to update packages.

Vignettes

Many packages include "vignettes". A package vignette gives an overview of the package and sometimes includes examples.

Package vignettes are not a required component of an R package, so some packages will not have them.

For those packages which contain vignettes, you can find them by the `browseVignettes()` function. For example, the vignette for the `hexbin` package can be found using the following command:

```
browseVignettes("hexbin")
```

The `vignette()` function can also be used for finding package vignettes.

Info about your R Session

To get a description of the version of R and packages that have been loaded for the current session, the `sessionInfo` function can be used:

```
> sessionInfo()
R version 2.13.0 (2011-04-13)
Platform: x86_64-apple-darwin9.8.0/x86_64 (64-bit)

locale:
[1] en_US.UTF-8/en_US.UTF-8/C/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] grid      stats      graphics  grDevices  utils
datasets  methods   base
```

Info about your R Session

other attached packages:

```
[1] hexbin_1.26.0    lattice_0.19-23  foreign_0.8-43
```

loaded via a namespace (and not attached):

```
[1] tools_2.13.0
```

The command `search()`, without any arguments, can also be used to for information on which packages are currently loaded.

Example: the foreign package

The `foreign` package comes with base R and is quite useful for importing and exporting data.

To use the functions that are available in this package, we must first load `foreign`:

```
library("foreign")
```

Datasets from other statistical analysis software using functions in the `foreign` package.

The function `read.spss` can be used to read in an SPSS data file:

```
dat.spss <- read.spss("http://faculty.washington.edu/tathornt/sisg2013/hsb2.sav",  
to.data.frame=TRUE)
```

The function `read.dta` can be used to read in a Stata data file:

```
dat.dta <- read.dta("http://faculty.washington.edu/tathornt/sisg2013/hsb2.dta")
```

Example: the survey package

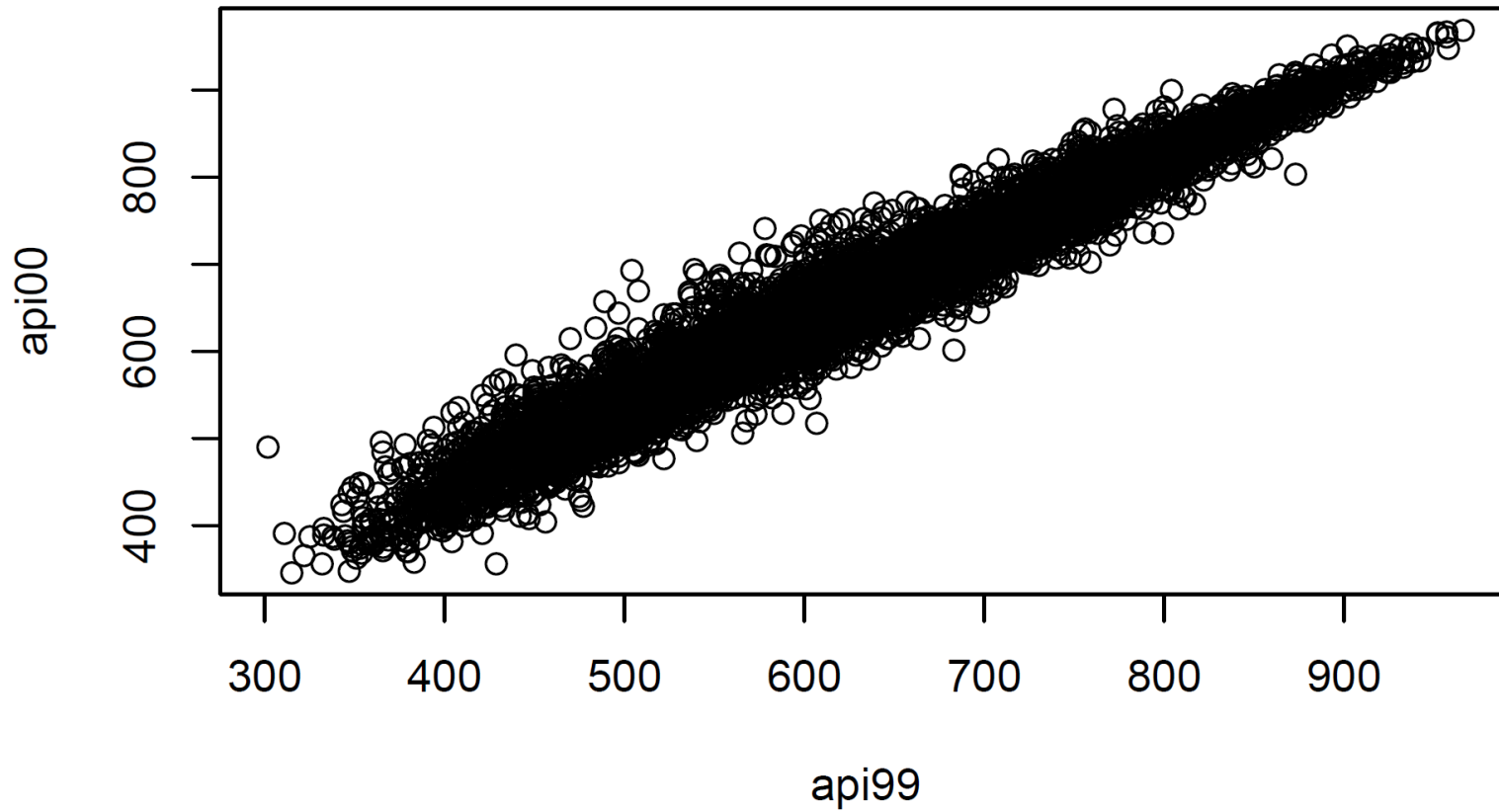
The `survey` package includes a data set named "api" containing California Academic Performance Index is reported on 6194 schools

```
library("survey")  
data(api, package="survey")
```

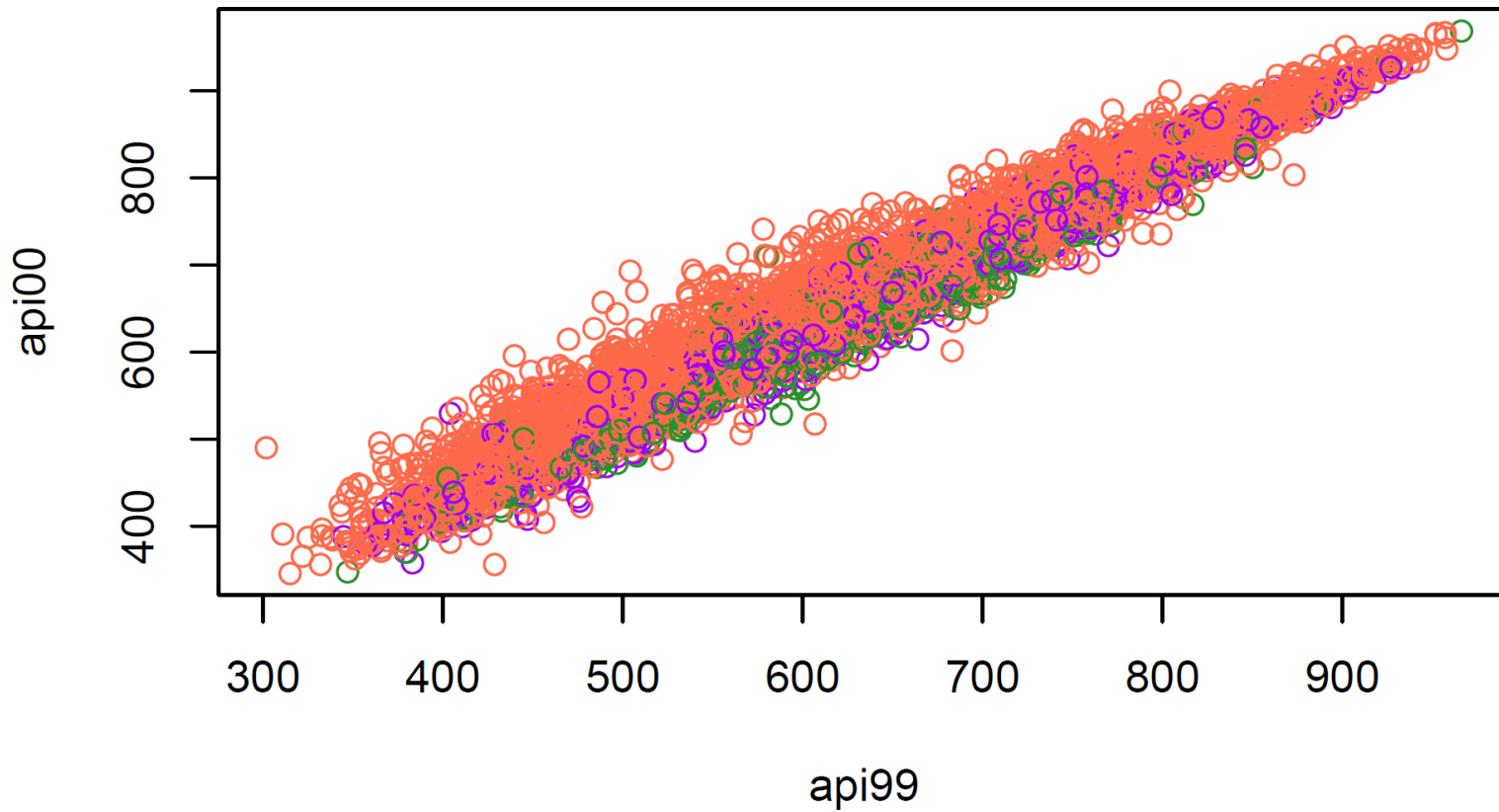
Below are scatterplot commands for this data. The scatterplots illustrate how crowded these plots can be with large data sets.

```
plot(api00~api99,data=apipop)  
colors<-c("tomato","forestgreen","purple")[apipop$type]  
plot(api00~api99,data=apipop,col=colors)
```

Example: the survey package



Example: the survey package



Example: the hexbin package

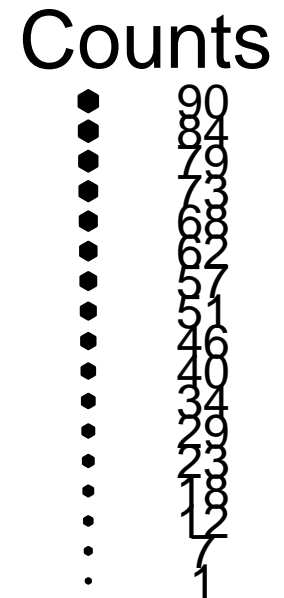
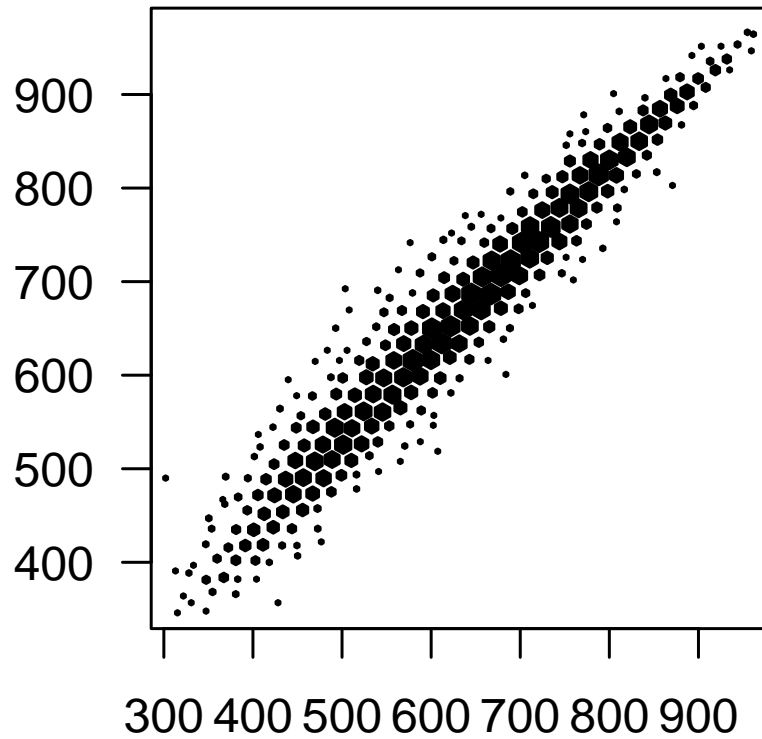
When there are many data points and significant overlap, scatterplots become less useful.

The `hexbin()` function in the `hexbin` package provides a way to aggregate the points in a scatterplot. It computes the number of points in each hexagonal bin.

```
library("hexbin")  
with(apipop, plot(hexbin(api99,api00), style="centroids"))
```

The `style="centroids"` option plots filled hexagons, at the centroid of each bin. The sizes of the plotted hexagons are proportional to the number of points in each bin.

Example: the hexbin package



Summary

- Many functions in R live in optional **packages**.
- Thousands of packages are available on CRAN for downloading.
- The `install.packages()` function is used for installing an extension package
- The `library()` function lists packages, shows help, or loads packages from the package library.
- The **foreign** package is in the standard distribution. It handles import and export of data.