

::

What's in a name?

Matt Pocernch

Denver R Users Group May 23rd,
2012

Motivation

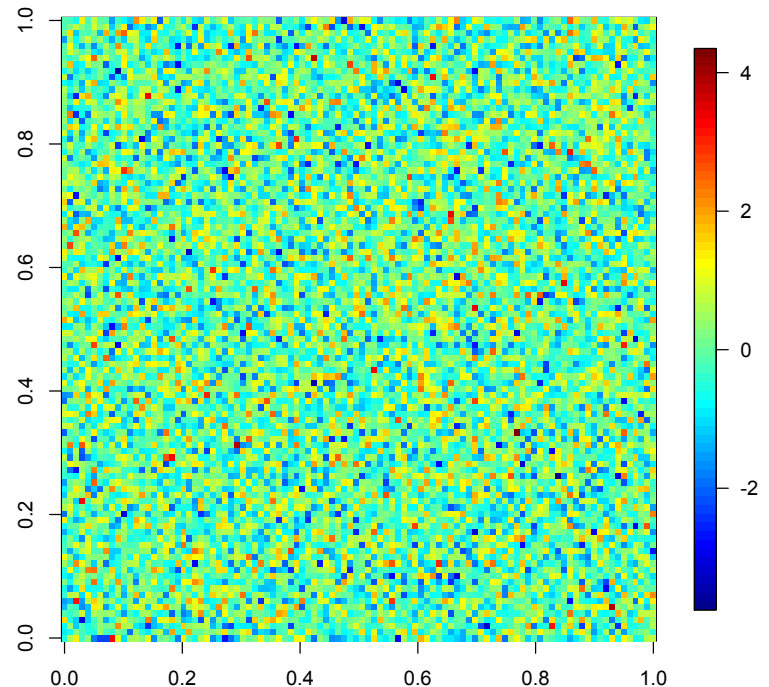
- `googleVis::gvisTable(basemap, chartid, ...)`
- “If a NAMESPACE file is not present, then one is generated automatically when the package is built or installed, all objects are exported, and all packages listed in the Imports or Depends fields in the DESCRIPTION file are imported. **This is only intended as a temporary measure whilst packages are converted to have a NAMESPACE file and will be removed in due course. A hand-crafted NAMESPACE should be added to any existing package which does not have one.**”

::

- If a package only needs a few objects from another package it can use a fully qualified variable reference in the code instead of a formal import. A fully qualified reference to the function `f` in package **foo** is of the form `foo::f`

An Illustration

```
> X <- matrix(rnorm(10000), nrow = 100)  
> image.plot(X)  
> Error: could not find function "image.plot"  
> fields::image.plot(X)
```



More

```
> tim.colors(10)
```

```
Error: could not find function "tim.colors"
```

```
> library(fields)
```

```
> tim.colors(10)
```

```
[1] "#00008F" "#0000FF" "#0070FF" "#00DFFF"  
"#50FFAF" "#BFFF40" "#FFCF00" "#FF6000"  
"#EF0000" "#800000"
```

:: Advantages

- This is slightly less efficient than a formal import and also loses the advantage of recording all dependencies in the NAMESPACE file, so this approach is usually not recommended. Evaluating `foo::f` will cause package **foo** to be loaded, but not attached, if it was not loaded already—this can be an advantage in delaying the loading of a rarely used package.

NAMESPACE

- Name spaces are sealed. This means that , once a package with a name space is loaded, it is not possible to add or remove variables or to change the values of variables defined in a name space.
- Allows control over what get exported and is visible to the global environment.

NAMESPACE files in packages

```
exportPattern("^[^\\.].")
```

Or

```
export(vaggregate)
```

```
importFrom(stats,setNames)
```

Or

```
exportClasses(SpatialCollections)
```

```
export(SpatialCollections)
```

```
exportMethods(row.names)
```

By default

Default

```
# Default NAMESPACE created by R  
# Remove the previous line if you edit this file
```

```
# Export all names  
exportPattern(". ")
```

```
# Import all packages listed as Imports or Depends  
Import( methods, waveslim,  
        fields, boot, CircStats,  
        MASS )
```

Example

- In the package foo

```
x<-1
```

```
f <- function(y) c(x,y)
```

Assume in the namespace

```
export(f)
```

Namespaces are *sealed* once they are loaded. Sealing means that imports and exports cannot be changed and that internal variable bindings cannot be changed.

References

- Extensions manual
- Luke Tierney. Name space management for R. *R News*, 3(1):2-6, June 2003.