Emacs for Nix-R

Rohit Goswami · 2020-06-10 · tools · nix · workflow · R · emacs

> *A short post on my current set-up for `R` with `nixpkgs` and `emacs` to auto-compile my system configuration.*

Background

This is my third post on working with `nixpkgs` and `R`.

- Part I covered ways of working effectively with `R` and `nixpkgs`
- Part II dealt with composing dependent `devtools` packages in a per-package environment, with a focus on `rethinking` and `tidybayes.rethinking`

This final part is about automating the system-wide configuration using `emacs`. Specifically `doom-emacs`. Naturally, this is the most optimal way to work with `nix` packages as well.

System Configuration

After experimenting with a per-project layout, I decided to use the full system configuration instead of the per-project layout. So I simply set:

```
# $HOME/.config/nixpkgs/config.nix
{
  packageOverrides = super:
    let
      self = super.pkgs;
      rethinking = with self.rPackages;
        buildRPackage {
          name = "rethinking";
          src = self.fetchFromGitHub {
            owner = "rmcelreath";
            repo = "rethinking";
            rev = "d0978c7f8b6329b94efa2014658d750ae12b1fa2";
            sha256 = "1qip6x3f6j9lmcmck6sjrj50a5azqfl6rfhp4fdj7ddabpb8n0z0";
          };
          propagatedBuildInputs = [ coda MASS mvtnorm loo shape rstan dagitty ];
        };
      tidybayes_rethinking = with self.rPackages;
        buildRPackage {
          name = "tidybayes.rethinking";
          src = self.fetchFromGitHub {
            owner = "mjskay";
            repo = "tidybayes.rethinking";
            rev = "df903c88f4f4320795a47c616eef24a690b433a4";
            sha256 = "1jl3189zdddmwm07z1mk58hcahirqrwx211ms0i1rzbx5y4zak0c";
          };
          propagatedBuildInputs =
            [ dplyr tibble rlang MASS tidybayes rethinking rstan ];
        };
    in {
      rEnv = super.rWrapper.override {
        packages = with self.rPackages; [
          tidyverse
          devtools
          modelr
          purrr
          forcats
          ####################
          # Machine Learning #
          ####################
          # MLR3
          mlr3
          mlr3viz
          mlr3learners
          mlr3pipelines
          # Plotting tools
          ggplot2
          cowplot
          ggrepel
          RColorBrewer
          # Stan Stuff
          rstan
          tidybayes
          # Text Utilities
          orgutils
          latex2exp
          kableExtra
          knitr
          data_table
          printr
          # Devtools Stuff
          rethinking
          tidybayes_rethinking
        ];
      };
```

```
    };
}
```

If any of these look strange, refer to the earlier posts.

Automation Pains

`direnv`, `lorri` and `niv` (the heroes of Part II) are not really useful for working with the system-wide configuration, but an elegant solution still exists, which leverages `firestarter` and `after-save-hooks` in `emacs`.

Firestarter

Firestarter is my favorite method of working with shell commands after saving things. My setup is simply:

```
; packages.el
(package! firestarter)
```

This is coupled with a simple configuration.

```
; config.el
(use-package! firestarter
  :ensure t
  :init
  (firestarter-mode)
  :config
  (setq firestarter-default-type t)
)
```

The default type corresponds to demanding the shell outupt for the commands.

Nix-R Stuff

To finalize this setup, we will need to modify our system configuration slightly. For brevity, we simply note the following local variables.

```
# $HOME/.config/nixpkgs/config.nix
# Local Variables:
# firestarter: "nix-env -f '<nixpkgs>' -iA rEnv"
# firestarter-default-type: (quote failure)
# End:
```

The `firestarter-default-type` used here is to ensure that errors are displayed in a buffer.

To check what is being installed (if anything) simply run:

```
nix-env -f "<nixpkgs>" -iA rEnv --dry-run
```

Conclusion

This is my current setup. It works out better than most of my other attempts and seems to be an optimal approach. The packages are versioned, everything is automated, and I can reproduce changes across all my machines. Will stick with this.