

# All Deployments

## Have

- Model `./pkgs/mymodel/data/fit.RDA`
- Scoring function(s) `./pkgs/mymodel/R/score.R`
- Invoking application(s)
  - CL/Rscript `./app/score-things.r`
  - WebAPI `./app/plummer.R`
  - ShinyApp `./app/server.R`
  - Embedded Application varies

## May Have

- Add'l assets `./pkgs/mymodel/R/plot_roc.R`
- Data fetching funs `./pkgs/mymodel/R/fetch.R`

# Good Practice



- Decoupled model scoring and asset generation from deployment endpoint.
- Allows multiple model consumers
  - Allows use of multiple deployments
  - Allows hand-off deployment to  
Developers / DevOps / DecisionOps
  - Provides deployment standard

# Requirements

Deployment Environment Has

- Invoker(s) installed
- R installed and executable

All R dependencies installed ...  
often a vary long list.

# Solution 1

Create a list of all packages used  
Log onto the deployment environment

```
$ R
R> pack <- installed.packages()
R> saveRDS(pack, file='packs.RDS' )
...
R> packs <- readRDS('packs.RDS')
R> install.packages(packs)
```

# Limitation of Solutions 1

All applications are dependent on the global list of R packages;

- Deployment is hard
- Multiple models cannot use different versions of packages

You need application encapsulation ...

# packrat

Isolated  
Portable  
Reproducible

Made for projects, but works with applications as well.

# What packrat does

Isolates execution environ (a directory)

```
> .libPaths(...) # .Rprofile
```

**Tracks/capture** package dependencies

```
> packrat::snapshot()
```

**Bundles** complete execution environ

```
> packrat::bundle(...)
```

**Installs** in platform indep new environ

```
> packrat::unbundle(...)
```

# Warning!

Packrat projects can become ridiculously large.

- Your invoker(s)
- Your package(s)
- Package Dependencies
  - Platform specific R files

Installation and unbundling can take a long time.



# Warning!

Packrat works great ... if you are just using CRAN repositories and the latest version of packages.

Packrat is tricky to use with “local” repositories.

# Warning!

Packrat works great ... if you are just using CRAN repositories and the latest version of packages.

Packrat is tricky to use with “local” repositories.

# Packrat Deployment Recipe

```
$ cp ./pkgs -> ./app/pkgs
```

```
# Create Rstudio project for ./app
```

```
> packrat::init() # In ./app/
```

```
> packrat::set_opts( local.repos = './pkgs/' )
```

```
> packrat::install_local( "pkg_name" )
```

```
> packrat::snapshot()
```

```
> packrat::bundle()
```

# Alternatives

Use drat or miniCRAN to create a local CRAN-like repository